Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions

Ryan Abbott,[1,2] Michael S. Albergo,[3] Denis Boyda,[4,1,2] Kyle Cranmer,[3] Daniel C. Hackett,[1,2] Gurtej Kanwar,[5,1,2] Sébastien Racanière,[6] Danilo J. Rezende,[6] Fernando Romero-López,[1,2] Phiala E. Shanahan,[1,2] Betsy Tian,[1] and Julian M. Urban[7]

[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]*Center for Cosmology and Particle Physics, New York University, New York, New York 10003, USA*
[4]*Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, Illinois 60439, USA*
[5]*Albert Einstein Center, Institute for Theoretical Physics, University of Bern, 3012 Bern, Switzerland*
[6]*DeepMind, London, United Kingdom*
[7]*Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany*

(Received 28 July 2022; accepted 30 September 2022; published 18 October 2022)

This work presents gauge-equivariant architectures for flow-based sampling in fermionic lattice field theories using pseudofermions as stochastic estimators for the fermionic determinant. This is the default approach in state-of-the-art lattice field theory calculations, making this development critical to the practical application of flow models to theories such as QCD. Methods by which flow-based sampling approaches can be improved via standard techniques such as even/odd preconditioning and the Hasenbusch factorization are also outlined. Numerical demonstrations in two-dimensional U(1) and SU(3) gauge theories with $N_f = 2$ flavors of fermions are provided.

DOI: 10.1103/PhysRevD.106.074506

I. INTRODUCTION

Lattice quantum field theory (LQFT), particularly lattice quantum chromodynamics, has become an ubiquitous tool in high-energy and nuclear theory [1–4]. Given the extraordinary computational cost of state-of-the-art LQFT studies, advances in the form of more efficient algorithms are of great value [5,6]. Recently, significant efforts have been made to design novel algorithms incorporating machine-learned components to accelerate the first stage of LQFT calculations, which involves sampling contributions to the high-dimensional discretized path integral [7–19]. In particular, bespoke generative flow models [20–22] tailored to the sampling of LQFT field configurations have been developed and applied in a variety of ways [23–46] and have been found to address key sampling challenges such as critical slowing-down and topological freezing in some two-dimensional (2D) theories [24–28,30].

In order to use flow-based sampling for LQFTs with fermions, such as QCD, suitable flow architectures to treat both gauge and fermionic degrees of freedom are needed. Several pieces of this puzzle are already in place. First, gauge-equivariant architectures for gauge fields have been developed and applied for 2D Abelian and non-Abelian theories [25,26,47]. Second, different methods to incorporate fermions in flow architectures have been discussed in Ref. [28], with numerical demonstrations provided in the context of the 2D Yukawa theory. Among the various proposals, integrating out the fermion fields and directly evaluating the resulting fermion determinant is the most straightforward approach; in this case, the target probability distribution is given only in terms of gauge variables. This procedure has found success in proof-of-principle applications to the Schwinger model at criticality [30]. However, this approach is not scalable. Specifically, computing the probability density after the fermionic integration via direct methods is not feasible for at-scale studies of theories such as QCD, as such methods scale cubically with the spacetime volume. The usual approach to this challenge is to introduce auxiliary degrees of freedom, named pseudofermions, which function as stochastic determinant estimators for which the cost of evaluation scales more favorably with the lattice volume. It seems natural to follow a similar approach

in flow-based sampling. This was considered in Ref. [28] for Yukawa theory, but architectures combining both gauge equivariance and pseudofermion degrees of freedom have not previously been presented.

This paper develops and presents flow architectures to model fermionic lattice gauge theories using pseudofermions. These architectures are based on "joint models," where the action defining the target probability distribution is split into a "marginal" part, which depends only on the gauge variables, and a "conditional" part, which depends on the pseudofermionic variables given fixed gauge fields. The function of the conditional part is to efficiently estimate the fermion determinant. The new developments of this work are architectures to model the conditional component, and the introduction of a parallel transporter convolutional network, which is the central piece of the architecture. In addition, it is outlined how standard approaches such as even/odd or Hasenbusch preconditioning can be combined with pseudofermion modeling.

Numerical demonstrations of the joint flow model architectures are provided in two toy gauge theories. One is the Schwinger model, which contains pseudo-Goldstone bosons, is confining, and has distinct topological sectors. The other is a 2D SU(3) gauge theory with $N_f = 2$ flavors of fermions. While this system is topologically trivial, it shares the gauge group of QCD and can have large correlation lengths.

This paper is organized as follows. First the lattice actions and the pseudofermion approach are summarized in Sec. II. Flow-based sampling is reviewed in Sec. III A, joint models are constructed in Sec. III B, the pseudofermion architectures are presented in Sec. III C, preconditioning is discussed in Sec. III D, and a method to use multiple samples of pseudofermions given fixed gauge fields is explored in Sec. III E. Numerical demonstrations are provided in Sec. IV A for the Schwinger model and in Sec. IV B for 2D SU(3). Section V presents a summary and conclusions. Appendix A is a derivation of the scaling of the model quality with the number of pseudofermions at fixed gauge fields, and further details of the numerical exploration are provided in Appendix B. Finally, Appendix C compares distributions of observables using two examples of flow models and hybrid Monte Carlo (HMC) for sampling.

II. LATTICE GAUGE THEORIES WITH PSEUDOFERMIONS

The action of a lattice gauge theory with N_f fermion degrees of freedom can be factorized into gauge and fermionic components as

$$S(U,\psi,\bar{\psi}) = S_g(U) + S_f(U,\psi,\bar{\psi}), \tag{1}$$

where the gauge links $U_\mu(x)$ are elements of the gauge group $\mathcal{G}$ and the fermion fields are collectively denoted as $\psi(x)$. The Wilson discretization of the gauge part of the action is [48,49]

$$S_g(U) = -\frac{\beta}{N_c}\sum_x \sum_{\mu<\nu} \mathrm{Re}\,\mathrm{Tr}\, P_{\mu\nu}, \tag{2}$$

where $\beta/2N_c = 1/g^2$ is the inverse of the squared gauge coupling, and N_c refers to the dimension of the fundamental representation of the gauge group. The plaquette is defined as

$$P_{\mu\nu}(x) \coloneqq U_\mu(x)U_\nu(x+\hat{\mu})U_\mu^\dagger(x+\hat{\nu})U_\nu^\dagger(x), \tag{3}$$

where $\hat{\mu},\hat{\nu}$ denote unit vectors in direction μ,ν, respectively, and periodic boundary conditions are assumed. For 2D theories, the only contribution is $\mu = 0$, $\nu = 1$. The fermionic part of the action of a 2D gauge theory with N_f degenerate flavors of fermions in the fundamental representation of $\mathcal{G}$ can be written as

$$S_f(U,\psi,\bar{\psi}) = \sum_{f=1}^{N_f}\sum_{x,y}\bar{\psi}_f^\gamma(y)D[U](y,x)^{\gamma\alpha}\psi_f^\alpha(x), \tag{4}$$

where $\psi_f^\alpha(x)$ denotes a fermion field with flavor f and spin index $\alpha \in \{1,2\}$, and the gauge indices are kept implicit. The Wilson discretization [48,49] of the lattice Dirac operator $D[U]$ is given by

$$\begin{aligned} D[U](y,x)^{\gamma\alpha} = {} & \delta(y-x)\delta^{\gamma\alpha} \\ & - \kappa\sum_{\mu=0,1}\{[1-\sigma_\mu]^{\gamma\alpha}U_\mu(y)\delta(y-x+\hat{\mu}) \\ & + [1+\sigma_\mu]^{\gamma\alpha}U_\mu^\dagger(y-\hat{\mu})\delta(y-x-\hat{\mu})\}, \end{aligned} \tag{5}$$

where $\sigma_\mu = (\sigma_x,\sigma_y)$, with $\sigma_{x,y}$ denoting the usual Pauli matrices, and $\kappa = 1/(4+2m_0)$, where m_0 is the bare fermion mass. Antiperiodic boundary conditions in the time direction are incorporated in the definition of the δ functions.

The full action, Eq. (1), is invariant under gauge transformations of the form

$$U_\mu(x) \to \Omega(x)U_\mu(x)\Omega^\dagger(x+\hat{\mu}), \qquad \psi(x) \to \Omega(x)\psi(x), \tag{6}$$

for any choice of $\Omega(x) \in \mathcal{G}$. Observables are computed as expectations over field configurations,

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\int DU\, D\psi\, D\bar{\psi}\, \mathcal{O}e^{-S(U,\psi,\bar{\psi})}, \tag{7}$$

where $p = e^{-S(U,\psi,\bar{\psi})}/\mathcal{Z}$ plays the role of a probability density, with $\mathcal{Z}$ a normalization constant.

It is common practice to integrate out the Grassmann-valued fermionic degrees of freedom,

$$\frac{1}{\mathcal{Z}_f}\int D\psi\, D\bar{\psi}\, e^{-S_f(U,\psi,\bar{\psi})} = (\det\, D[U])^{N_f}, \tag{8}$$

where $\mathcal{Z}_f$ is a normalization constant. This results in an effective action, which in the case of $N_f = 2$ reads

$$S_{\text{eff}}[U] = S_g(U) - \log\ \det\, DD^\dagger[U], \tag{9}$$

where $\det D^2 = \det DD^\dagger$ due to γ_5 Hermiticity of D. This form of the action is given only in terms of gauge variables and can in principle be used for lattice calculations. However, the evaluation cost of the determinant scales poorly with the lattice volume. The scalable approach for state-of-the-art lattice QCD uses stochastic determinant estimators, in particular ones based on the following relation for positive definite matrices M:

$$\det M = \frac{1}{(2\pi)^N}\int d\phi e^{-\phi^\dagger M^{-1}\phi}, \tag{10}$$

where $\phi(x)$ are complex bosonic variables (the pseudofermions), N is the number of pseudofermion variables, and $\int d\phi$ denotes integration over these variables. Adding these auxiliary variables, the $N_f = 2$ theory can be represented with the following action:

$$S(U,\phi,\phi^\dagger) = S_g(U) + S_{\text{pf}}(U,\phi,\phi^\dagger), \tag{11}$$

$$\text{with}\quad S_{\text{pf}}(U,\phi,\phi^\dagger) = \phi^\dagger(D[U]D^\dagger[U])^{-1}\phi. \tag{12}$$

In this form only the inverse of the Dirac operator applied to the pseudofermions is needed to evaluate the action, which can be computed with effectively linear cost scaling[1] with respect to the lattice volume. This comes at the cost of additional stochastic noise. Several common variations on this approach, referred to as schemes for preconditioning the action, are described below.

III. FLOW MODELS FOR PSEUDOFERMIONS

A. Flow-based sampling

Normalizing flows [20–22] have proven to be a promising tool to mitigate critical slowing down and topological freezing in some lattice field theories [24,25,30]. For an in-depth introduction to normalizing flows for lattice field theory, we refer the reader to Ref. [29]. Here, we review the key concepts relevant for this work.

A "flow" is a diffeomorphism f that is applied on a set of samples drawn from an easy-to-sample base (or prior) distribution, $r(z)$. The resulting configurations, $\varphi = f(z)$, are distributed according to the model distribution with density

$$q(\varphi) = r(z)\left|\det\,\frac{\partial f(z)}{\partial z}\right|^{-1}. \tag{13}$$

The flow model is constructed with trainable parameters, which can be optimized to approximate a target probability distribution p, i.e., $q(\varphi) \simeq p(\varphi)$.

In the context of lattice field theory, configurations φ are discretized quantum fields, and the target distribution is given by the Euclidean action of the theory, i.e., $p(\varphi) = e^{-S(\varphi)}/\mathcal{Z}$. Expressive transformations may be obtained by defining the flow f as the composition of n invertible layers

$$f = g_1 \circ g_2 \ldots {}^\circ g_n. \tag{14}$$

Various architectures for the invertible layers g_i tailored for different theories have been proposed in previous work, e.g., for scalar theories [24,27,41], Abelian and non-Abelian pure-gauge theories [25,26], and theories containing fermions [28].

The optimization of a flow model proceeds by minimizing a loss function that quantifies the difference between the model and target distributions. A common choice of loss is the Kullback-Leibler (KL) divergence [51], see also Appendix B 1. A self-training optimization scheme can be used, in which the KL divergence is computed stochastically by drawing samples, or "training data," from the model distribution. A useful measure of model quality during training is the effective sample size per configuration (ESS), as defined, e.g., in Ref. [29]:

$$\text{ESS} = \frac{1}{N}\frac{(\sum_{i=1}^N p(\varphi_i)/q(\varphi_i))^2}{\sum_{i=1}^N (p(\varphi_i)/q(\varphi_i))^2}, \tag{15}$$

where N is the number of samples used to compute the estimate. Larger ESS implies better model quality and $\text{ESS} \in [1/N, 1]$.

Given a trained flow model, the most straightforward approach to sample from p is to build a Markov chain using the independence Metropolis algorithm. Starting from some initial sample φ, the probability to accept an independent proposal φ' generated by the flow model is defined as

$$A(\varphi \to \varphi') = \min\left(1, \frac{p(\varphi')}{p(\varphi)}\frac{q(\varphi)}{q(\varphi')}\right). \tag{16}$$

Since each proposal from the flow model is independent from the previous one, autocorrelations in the Markov chain only arise from rejections in the Metropolis step. Higher quality models therefore result in less autocorrelation.

[1]For instance, the scaling of the conjugate gradient algorithm is $O(n_d\sqrt{k_n})$, where n_d is the dimension of the matrix and k_n is its condition number [50].

Alternatively, one may use a reweighting procedure, where the weight of each configuration is given by

$$w(\varphi) = p(\varphi)/q(\varphi). \tag{17}$$

B. Joint models

We build on Ref. [28] to define models for joint sampling of bosonic and pseudofermionic degrees of freedom. Specifically, we construct "joint autoregressive models" as described in Ref. [28], which we simply refer to as "joint models" in the following. The fundamental idea is to factorize the probability distribution of Eq. (12) as

$$p(U,\phi) = p(U)p(\phi|U), \tag{18}$$

where each piece is defined as

$$\begin{aligned} p(U) &\propto \det DD^\dagger[U] e^{-S_g(U)}, \\ p(\phi|U) &\propto \frac{e^{-S_{\mathrm{pf}}(U,\phi,\phi^\dagger)}}{\det DD^\dagger[U]}, \end{aligned} \tag{19}$$

referred to as the "marginal" and "conditional" distributions, respectively. As noted in Ref. [28], sampling from $p(\phi|U)$ is straightforward, but computing the normalizing constant $\det DD^\dagger$—needed for the ratios in Eqs. (16) and Eq. (17)—is not.

We use two independent flow models to model the respective distributions:

$$q(U,\phi) = q(U)q(\phi|U) \simeq p(U)p(\phi|U), \tag{20}$$

where the first component approximates the marginal distribution:

$$q(U) = r_m(z)\left|\det\frac{\partial f_m(z)}{\partial z}\right|^{-1}, \qquad U = f_m(z), \tag{21}$$

and the other models the conditional distribution:

$$q(\phi|U) = r_c(\chi)\left|\det\frac{\partial f_c(\chi|U)}{\partial \chi}\right|^{-1}, \qquad \phi = f_c(\chi|U), \tag{22}$$

where r_m is taken to be uniform over the Haar measure and r_c is a spherical Gaussian over all components of ϕ. Note that the transformations defining the conditional flow, f_c, act only on the pseudofermions, that is, the gauge fields remain unchanged under the action of f_c.

Figure 1 sketches the sample generation workflow of a joint model. For $f_m(U)$ we use the gauge-equivariant layers described in Ref. [26], while $f_c(\phi|U)$ requires new technology which will be described in the following section; while pseudofermion architectures have already been presented in Ref. [28] for Yukawa theory, these did not treat pseudofermions coupled to gauge variables.

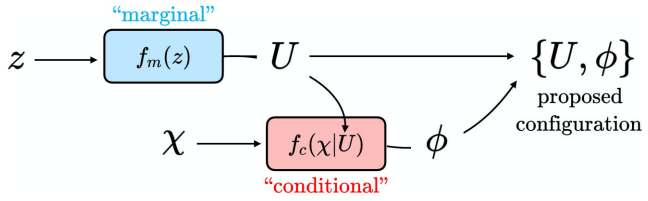


FIG. 1. Sketch of the workflow of a joint model. z and χ are samples from the base distribution which are transformed to produce the gauge and pseudofermion fields, U and ϕ. $f_m(z)$ labels the flow model for the marginal distribution, and $f_c(z|U)$ that for the conditional one. The proposed configuration is distributed according to $q(\phi,U) = q(U)q(\phi|U)$.

To evaluate the quality of a joint model, one can use the "joint ESS," defined by Eq. (15) with $q(\varphi) \to q(U,\phi) = q(U)q(\phi|U)$ and $p(\varphi) \to p(U,\phi) = p(U)p(\phi|U)$. Another useful metric is the "marginal ESS," which evaluates the quality of only the marginal model, cf. the target defined by Eq. (9). This is obtained from Eq. (15) with $q(\varphi) \to q(U)$ and $p(\varphi) \to p(U)$.[2]

C. Gauge-equivariant architectures

In this section, we describe how expressive gauge-equivariant transformations can be constructed for the conditional distribution of pseudofermions. In the approach of Ref. [28], which treats pseudofermions that are not coupled to gauge degrees of freedom, the conditional flow f_c is a function that maps an uncorrelated Gaussian distribution to a correlated one:

$$r(\chi) \propto e^{-\chi^\dagger\chi} \overset{f_c(\chi|U)}{\to} q(\phi|U) \propto e^{-\phi^\dagger(\tilde{D}\tilde{D}^\dagger[U])^{-1}\phi}. \tag{23}$$

That is, the flow is a change of basis constructed as a linear transformation $\tilde{D}$ of the pseudofermions. Naturally, the choice $\tilde{D} = D$ would set $q(\phi|U) = p(\phi|U)$; however, evaluating the Jacobian of such a transformation would require evaluating the fermion determinant exactly, which is expensive. Instead the map f_c can be constructed as a composition of simpler linear transformations, each with a tractable Jacobian.

In constructing such linear transformations, care must be taken to ensure the resulting map f_c is gauge equivariant. A naive linear combination of pseudofermion variables at different lattice sites is not gauge equivariant, since pseudofermions at different positions transform differently. In order to construct a gauge-equivariant linear transformation, the pseudofermions must be parallel transported to a common point. For instance, the linear combination

[2]Note that the marginal ESS requires the evaluation of the fermion determinant, and thus cannot be evaluated on larger volumes.

$$\phi'(x) = a\phi(x) + bU_\mu(x)\phi(x+\hat{\mu}), \tag{24}$$

with $a, b \in \mathbb{C}$, transforms as $\phi'(x) \to \Omega(x)\phi'(x)$, and thus defines an equivariant transformation.

We generalize this idea to introduce a *parallel transport layer*, $\mathbb{L}[U,\phi]$, which collects together the field at a given point y with all of its nearest neighbors parallel transported to y, i.e.,

$$\mathbb{L}[U,\phi](y) = \{\phi(y), \phi^{x_1}(y), \ldots, \phi^{x_{2\times d}}(y)\}, \tag{25}$$

where x_i labels the coordinates of the $2 \times d$ neighbors of y (assuming d space-time dimensions), and $\phi^{x_i}(y)$ is the pseudofermion $\phi(x_i)$ parallel transported to the location y. Note that antiperiodic boundary conditions must be applied when parallel transporting across the temporal boundary.

We use the parallel transport layer to form linear combinations of the pseudofermions in an equivariant way, similar to the technique developed in Ref. [47]. Each input $\phi(y)$ has K features, which in the this context indicate internal indices that are not gauge indices, e.g., spinors in 2D have $K = 2$ features. Thus, the concatenation of all vectors gathered by $\mathbb{L}[U,\phi](y)$ across their features outputs a new vector with $\tilde{K} = (2 \times d + 1)K$ features. This way, we define the *parallel transport convolution*, $\mathrm{PTC}[U,\phi](y)$, as

$$\mathrm{PTC}[U,\phi]_\alpha(y) = \sum_\gamma M_{\alpha\gamma}\mathbb{L}[U,\phi]_\gamma(y), \tag{26}$$

where α, γ denote the feature indices, and the complex-valued matrix M has dimensions $H \times \tilde{K}$. Here, H is the number of features of the output of this layer. In order to build expressive transformations and incorporate further information from the gauge field, one can parametrize $M_{\alpha\gamma}$ as a neural network applied to the gauge field, for instance, a standard convolutional neural network with gauge invariant inputs. Finally, a *parallel transport convolutional network*, $\mathrm{PTCN}[U,\phi](y)$, is defined as the composition of multiple PTCs with $H = 2$ (i.e., the number of spin components in 2D) in the last PTC layer. The choice of H at intermediate layers is arbitrary, and can be varied to increase expressivity. In a PTCN, the number of PTCs is denoted as n_{PT}. We emphasize that PTCNs are linear operators on the input fields $\phi(x)$.

We are now in the position to define *pseudofermion layers*. To ensure that the Jacobian of the transformation can be computed efficiently, we perform variable partitioning into active, ϕ_a, and frozen, ϕ_f, degrees of freedom, encoded via a projector $\mathbb{P}$ defined such that

$$\phi_a(x) = \mathbb{P}(x)\phi(x), \phi_f(x) = (\mathbb{1} - \mathbb{P}(x))\phi(x). \tag{27}$$

Then, a pseudofermion layer is defined by updating the active variables conditioned on the frozen ones as

$$\begin{aligned}\phi'_a(x) &= A(x)\phi_a(x) + \mathrm{PTCN}[U,\phi_f](x),\\ \phi'_f(x) &= \phi_f(x),\end{aligned} \tag{28}$$

where $A(x)$ is a site-dependent complex matrix in spin space (or is a scalar) that, like $M_{\alpha\gamma}$, is also a function of gauge invariant combinations of links parametrized by a neural network.

The projector $\mathbb{P}$ can be defined in different ways. One choice is "spatial partitioning," where the lattice sites are separated into active and frozen partitions, x_a and x_f, respectively. This can be achieved, e.g., by using a checkerboard pattern. In this scheme, the projector takes the form

$$\mathbb{P}(x_a) = \mathbb{1}_s, \qquad \mathbb{P}(x_f) = 0, \tag{29}$$

and $\mathbb{1}_s$ is the identity in the spinor indices. The Jacobian thus factorizes as

$$\log J = 2\sum_{x\in x_a} \log|\det A(x)|, \tag{30}$$

where the factor of 2 accounts for the complex numbers. Another choice of projector implements "spin partitioning," such that the projector is identical on all lattice sites and isolates a certain spinor component. In 2D, this can be defined as

$$\mathbb{P}^{\pm}(x) = \frac{1}{2}(\mathbb{1}_s \pm \sigma_z), \tag{31}$$

where σ_z is the usual Pauli matrix, and $\mathbb{P}^+(\mathbb{P}^-)$ projects to the upper (lower) spinor component. In this case, the Jacobian of the transformation is also trivial to evaluate:

$$\log J = 2\sum_x \log|A(x)|, \tag{32}$$

where $A(x)$ is now interpreted as a complex scalar that acts only on the active spin component.

A flow transformation can finally be defined as a composition of several pseudofermion layers. The parameters of the transformations of all layers—$M_{\alpha\gamma}$ in Eq. (26) and $A(x)$ in Eq. (28)—can be chosen to be the outputs of a "context function" built from neural networks.[3] This function only depends on the gauge links, and it can only take gauge-invariant inputs. We describe our choice in Appendix B. In the numerical demonstrations of Sec. IV we choose to alternate the choice of active and frozen variables such that all variables are updated with the same frequency. In addition, one can build a conditional flow using only

[3]This nested evaluation of neural networks to produce parameters for higher-level networks is similar to previous explorations of nested architectures in the machine learning community [52].

spatial partitioning, only spin partitioning, or a combination of both.

D. Joint models for preconditioned actions

The Wilson-Dirac operator can have a large condition number due to the presence of small eigenvalues. This effect is enhanced when approaching the chiral limit of lattice gauge theories. Fortunately, this situation can be ameliorated by employing preconditioned actions, as is standard in state-of-the-art lattice QCD calculations using the hybrid Monte Carlo algorithm. Two common approaches are even/odd [53] preconditioning and the Hasenbusch factorization [54]. As detailed below, these schemes define modified target distributions cf. Eq. (19), which require different architectures to model.

In the context of flow models, higher condition numbers imply more difficult target distributions, as the variance of the stochastic determinant estimator increases. This intuition is borne out in practice, as it can indeed be observed that it is more difficult to train flow models to model operators with larger condition number. Here, we describe how pseudofermion modeling can be combined with preconditioning techniques to mitigate this effect. As demonstrated in Sec. IV, this approach can be numerically advantageous.

1. Pseudofermion models for even/odd preconditioning

Even/odd (EO) preconditioning [53] is a simple idea that reduces both the condition number and the number of degrees of freedom of the Dirac operator at almost no cost. It is based on the rearrangement of the Wilson-Dirac operator into the form:

$$D = \begin{pmatrix} \mathbb{1} & D_{\mathrm{eo}} \\ D_{\mathrm{oe}} & \mathbb{1} \end{pmatrix}, \tag{33}$$

where D_{oe} and D_{eo} are the terms connecting nearest neighbors ("even" and "odd" sites). This way, the determinant can be calculated as

$$\det D = \det\left(\mathbb{1} - D_{eo}D_{oe}\right) \equiv \det D_{\mathrm{sc}}, \tag{34}$$

where the subscript "sc" stands for Schur complement. Note that while $D \sim \mathbb{1} + O(\kappa)$, the EO preconditioned Dirac operator is $D_{\mathrm{sc}} \sim \mathbb{1} + O(\kappa^2)$, explaining the smaller condition number. The operator D_{sc} acts on only even-site variables.

Pseudofermions can be used to estimate the EO preconditioned fermion determinant in the usual manner:

$$\det DD^\dagger \propto \int d\phi_e e^{-\phi_e^\dagger (D_{\mathrm{sc}}D_{\mathrm{sc}}^\dagger)^{-1}\phi_e}, \tag{35}$$

where ϕ_e represents the pseudofermionic degrees of freedom defined only on the even sites of the lattice.

In order to model the determinant in Eq. (35), we must adapt our architecture to treat a pseudofermion field defined only on even sites. In practice, this can be done using architectures almost identical to those described in the previous section, by retaining odd-site variables but never updating their values from zero. Each PTCN may then populate these fields with values in intermediate states, so that they serve as additional "working memory." Then, the only architectural change must account for the fact that a single application of $\mathbb{L}[U,\phi]$ as defined in Eq. (26) connects only even sites to odd ones, and vice versa. To ensure that information is transferred across the $\phi_e(x)$ variables, one must apply always at least two $\mathrm{PTC}[U,\phi]$ within the PTCN, i.e., $n_{\mathrm{PT}} > 1$. An alternative possibility is to modify the parallel transport layer, $\mathbb{L}[U,\phi]$, to parallel transport directly between even sites. Finally, note that spatial partitioning by checkerboarding is not well suited for EO-preconditioned targets, for which only even sites are defined. Instead, one may for example update even sites in every other row (or equivalently, column), alternating which rows/columns are updated from layer to layer; this approach is used in the numerical investigation of Sec. IV.

2. Hasenbusch factorization in pseudofermion models

Hasenbusch factorization [54] is another common approach, and is a useful trick to separate the modes of the fermionic determinant. This can be achieved by factoring the determinant as

$$\det M = \left[\frac{\det M}{\det\,(M+\mu)}\right] \det\,(M+\mu), \tag{36}$$

where $\mu > 0$. Each of the factors in this equation are referred to as "monomials." Each monomial can be estimated independently with separate pseudofermion fields:

$$\frac{\det M}{\det\,(M+\mu)} \propto \int d\phi_0 e^{-\phi_0^\dagger(\mathbb{1}+\mu M^{-1})\phi_0},$$
$$\det\,(M+\mu) \propto \int d\phi_1 e^{-\phi_1^\dagger (M+\mu)^{-1}\phi_1}, \tag{37}$$

associated with separate conditional target densities $p(\phi_0|U)$ and $p(\phi_1|U)$, respectively. When $\mu = 0$, the first monomial is trivial and the original problem is recovered. It can easily be seen that increasing μ numerically simplifies the evaluation of the second term of Eq. (37), while making the evaluation of the first more difficult.

This procedure can be iterated N_h times:

$$\det M = \det\,(M+\mu_{N_h}) \prod_{i=0}^{N_h-1} \frac{\det\,(M+\mu_i)}{\det\,(M+\mu_{i+1})}, \tag{38}$$

where $\mu_0 = 0$ and $\mu_i < \mu_{i+1}$. In this equation, one has $N_h + 1$ independent monomials for N_h "Hasenbusch

steps." In practice, the values of μ_i must be tuned to achieve the optimal performance; the best choice is typically such that all monomials have similar average condition numbers.

To combine this technique with flow models, one can use the architecture described in Sec. III C, without modification, for each of the determinants to be estimated. This means constructing joint models containing multiple different conditional models:

$$q(U, \phi_0, \ldots, \phi_{N_h+1}) = q(U) \prod_i^{N_h+1} q(\phi_i|U), \tag{39}$$

where $q(\phi_i|U)$ is the density of the flow model for the monomial i. The ESS for the resulting joint model can be computed using Eq. (15) with the weight factor

$$w(U, \phi_0, \ldots, \phi_{N_h+1}) = \frac{p(U)}{q(U)} \prod_i^{N_h+1} \frac{p(\phi_i|U)}{q(\phi_i|U)}. \tag{40}$$

Furthermore, the EO preconditioning introduced in Sec. III D 1 can be easily combined with the Hasenbusch factorization, as is common in state-of-art lattice QCD studies, by simply replacing $M = DD^\dagger \to D_{\mathrm{sc}} D_{\mathrm{sc}}^\dagger$ in Eq. (38). Combinations of both techniques will be the default approach used in the numerical demonstrations discussed in Sec. IV.

E. Improving the stochastic determinant estimate

In order to use joint models more efficiently, one can draw multiple pseudofermion samples for fixed gauge fields. This provides more precise estimators of the determinant of the Dirac operator. The resulting improved weights can be used with the pseudomarginal Markov-chain Monte Carlo algorithm [28,55] to provide better statistical performance. The procedure is outlined below.

For a fixed gauge field U, the determinant of $M(U)$ can be estimated as

$$\begin{aligned} \det M(U) &= \int d\phi e^{-S_{\mathrm{pf}}(U,\phi,\phi^\dagger)}, \\ &= \int d\phi q(\phi|U) \frac{e^{-S_{\mathrm{pf}}(U,\phi,\phi^\dagger)}}{q(\phi|U)}, \\ &= \left\langle \frac{e^{-S_{\mathrm{pf}}(U,\phi,\phi^\dagger)}}{q(\phi|U)} \right\rangle_q, \end{aligned} \tag{41}$$

where $\langle \cdots \rangle_q$ denotes the average over samples distributed as $q(\phi|U)$. This way, one can define the weight factor for the conditional part using N_{pf} pseudofermion samples as

$$w_{N_{\mathrm{pf}}}(\{\phi\}|U) = \frac{1}{N_{\mathrm{pf}}} \sum_{i=1}^{N_{\mathrm{pf}}} \frac{p(\phi^{(i)}|U)}{q(\phi^{(i)}|U)}. \tag{42}$$

Combining this with the weight factor of the marginal model, the weight factor of a single gauge configuration U and corresponding set of N_{pf} pseudofermion samples $\{\phi\}$ can be defined as

$$\begin{aligned} w_{N_{\mathrm{pf}}}(U) &= \frac{p(U)}{q(U)} \times w_{N_{\mathrm{pf}}}(\{\phi\}|U), \\ &= \frac{1}{N_{\mathrm{pf}}} \sum_{i=1}^{N_{\mathrm{pf}}} \frac{p(\phi^{(i)}, U)}{q(\phi^{(i)}, U)}. \end{aligned} \tag{43}$$

This way, one can use the weight factor $w_{N_{\mathrm{pf}}}(U)$ averaged over several pseudofermion draws for a Metropolis accept-reject step, which leads to higher acceptance rates. In the limit of infinitely many pseudofermion samples, this converges to using the marginal flow model with an exact evaluation of the determinant of the Dirac operator. As a metric, one can also define the ESS for several pseudofermions:

$$\mathrm{ESS}(N_{\mathrm{pf}}) = \frac{1}{N} \frac{(\sum_{i=1}^N w_{N_{\mathrm{pf}}}(U^{(i)}))^2}{\sum_{i=1}^N w_{N_{\mathrm{pf}}}(U^{(i)})^2}. \tag{44}$$

As demonstrated in Appendix A, this depends on the number of pseudofermion draws as

$$\mathrm{ESS}(N_{\mathrm{pf}}) = \frac{\mathrm{ESS}(\infty)}{1 + \frac{C}{N_{\mathrm{pf}}}}, \tag{45}$$

where C is a constant, and $\mathrm{ESS}(\infty)$ coincides with the marginal ESS.

Finally, when using this procedure with the Hasenbusch factorization, one must draw N_{pf} independent pseudofermion samples for each monomial, for a total of $N_{\mathrm{pf}}(N_h + 1)$. The weight factor for each monomial, $w_{N_{\mathrm{pf}}}(\{\phi_i\}|U)$, will be computed as in Eq. (42), and the overall weight factor $w_{N_{\mathrm{pf}}}(U)$ will result from multiplying all the weight factors from each monomial with the marginal part:

$$w_{N_{\mathrm{pf}}}(U) = \frac{p(U)}{q(U)} \times \prod_i^{N_h+1} w_{N_{\mathrm{pf}}}(\{\phi_i\}|U). \tag{46}$$

IV. NUMERICAL DEMONSTRATION

This section presents numerical examples of applications of the joint modeling approach to sampling gauge field configurations for two 2D toy theories. A subsection is dedicated to each of the Schwinger model and SU(3) in 2D. More details about the numerical experiments can be found in Appendix B, and examples of distributions of values for observables using two of these models for sampling in Appendix C.

A. $N_f = 2$ Schwinger model

The Schwinger model, that is, 2D QED, shares some features with QCD [56,57] and is often used as a testbed for new algorithms for lattice gauge theories [42,58–67]. In the context of flow-based sampling and related approaches, this theory has already been investigated in Refs. [30,42]. However, these works used the exact determinant action of Eq. (9), while here we use the pseudofermion approach.

For the numerical demonstrations, we use the following parameters for the action: $\beta = 2.0$ and $\kappa = 0.265$ (corresponding to a bare fermion mass of $m_0 = -0.113208$). The pseudoscalar mass is $aM_{\mathrm{PS}} \sim 0.35$. We implement an architecture using the spatial variable partitioning presented in Eq. (29). Furthermore, we use a regulator of the Dirac operator during training in the form of $(DD^\dagger + \mu_0)$, with $\mu_0 = 10^{-5}$. We use this regulated target only during training, for stability. During sampling and when computing observables, we use the unregulated operator and no additional approximation is induced by this procedure.

1. Example of joint model

This section presents the results of a joint model trained for lattice extent $L = 16$. For the pseudofermionic target, we use EO preconditioning with three Hasenbusch iterations and $(\mu_1, \mu_2, \mu_3) = (0.001, 0.01, 0.05)$. We train using a batch size of 1344 configurations per gradient step, and use models with 32 pseudofermion layers. In total, there is one flow model to generate the gauge links, and four independent models for the pseudofermions, one for each Hasenbusch monomial (some weights are shared in the context function of these models, as discussed in Appendix B). We train the joint model as a whole, optimizing all five components simultaneously. The training curve for the joint and marginal ESS is shown in Fig. 2. As can be seen, a ∼5% joint ESS is obtained, while the marginal ESS is ∼35%. When employed for sampling using the independence Metropolis algorithm with a batch of 20k configurations, this flow model provides an acceptance rate of ∼18%.

A way to improve the sampling quality is to follow the procedure outlined in Sec. III E, that is, drawing more pseudofermion samples for fixed gauge fields. This is shown in Fig. 3, where the ESS as a function of the number of pseudofermion draws, N_{pf}, is shown. In that figure, from right to left, every point corresponds to an increase by a factor of 2 in N_{pf}. It can be clearly seen that the ESS increases with larger N_{pf} and approaches the marginal ESS. To guide the eye, a fit to Eq. (45) is also shown. This method is a promising option for increasing the statistical precision of measured observables when using flow-based sampling, as it requires only additional pseudofermion sampling without modifying or generating additional gauge fields. Indeed, training can be done with a small N_{pf}, and then N_{pf} may be increased arbitrarily during evaluation. Because statistical quality can be improved without generating additional gauge field samples, the approach can be particularly advantageous in situations where the evaluation of observables dominates the computational cost. This is in fact the case in many lattice QCD applications.

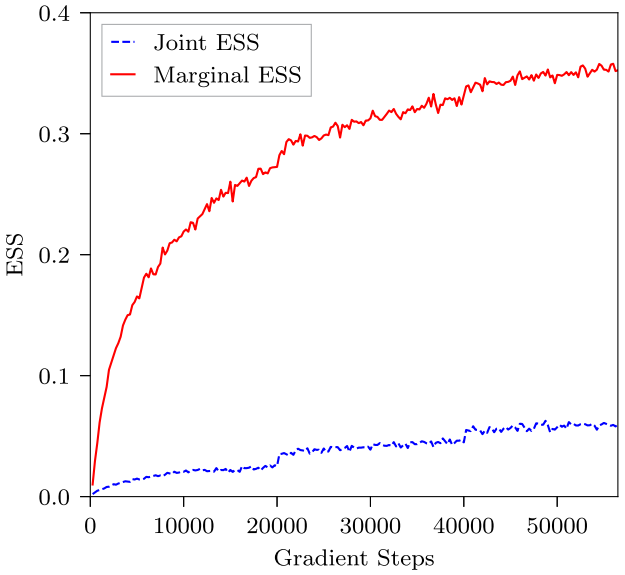


FIG. 2. Joint and marginal ESS for a flow for the Schwinger model with $L = 16$ and model parameters and flow model architecture as described in the text. The jumps in ESS are a result of learning-rate scheduling; see Appendix B for details of the training hyperparameter choices.

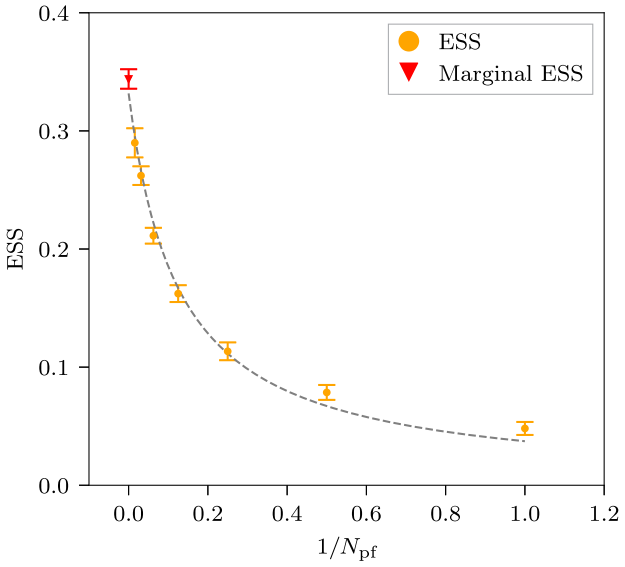


FIG. 3. Scaling of the ESS as defined in Eq. (44) as a function of $1/N_{\mathrm{pf}}$, where N_{pf} is the number of pseudofermions drawn with fixed gauge fields. Each orange point corresponds to a factor of 2 increase in N_{pf}, from 1 to 64. The red triangle corresponds to the marginal ESS of the model. Uncertainties are computed by using 10 independent estimations of the ESS on batches of 10240.

2. Effect of preconditioning

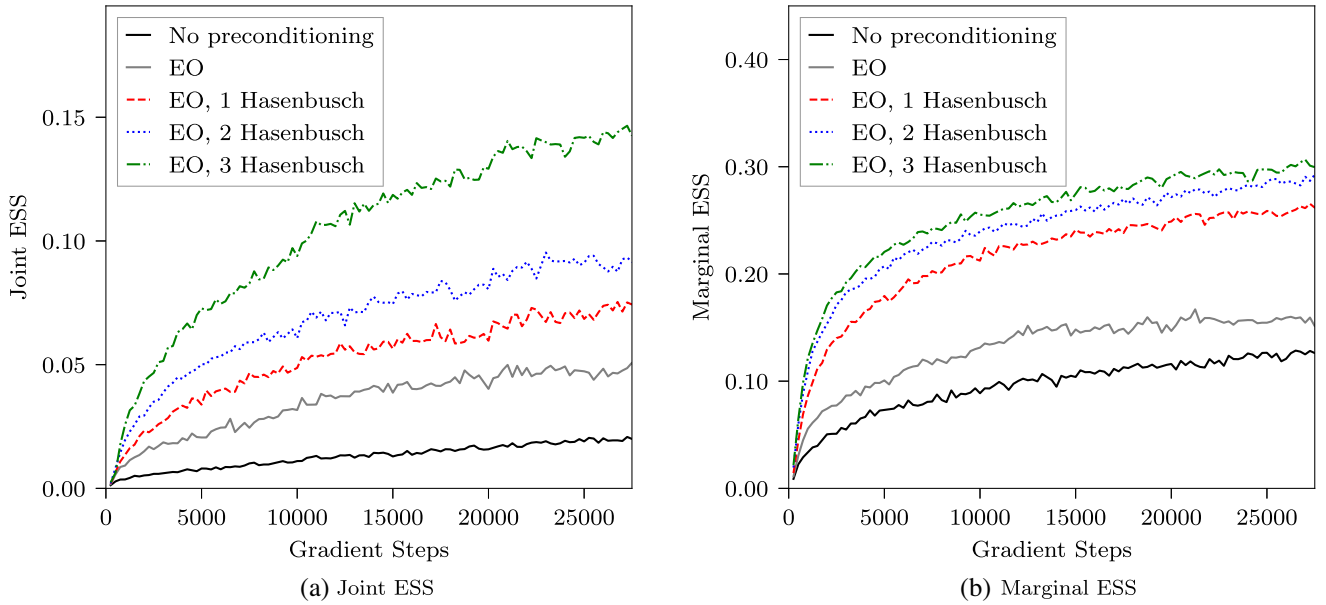
This section provides a numerical demonstration of how different preconditioning schemes interact with flow model quality for the Schwinger model. For these tests, we use a similar model architecture as detailed in the previous section, with 24 pseudofermion layers and the same target parameters as in the previous example, but at a smaller lattice extent, $L = 8$. Figure 4 shows the joint ESS (left) and marginal ESS (right) of models for a set of five different pseudofermion targets: (i) no preconditioning, (ii) even/odd, (iii) even/odd and Hasenbusch with one monomial, $\mu_1 = 0.001$, (iv) even/odd and Hasenbusch with two monomials, $(\mu_1, \mu_2) = (0.001, 0.01)$, and (v) even/odd and Hasenbusch with three monomials, $(\mu_1, \mu_2, \mu_3) = (0.001, 0.01, 0.05)$.

Clearly, EO preconditioning results in significantly increased ESS, as is to be expected from the reduced number of pseudofermion variables. This improvement is also visible in the marginal ESS, which suggests better propagation of information about fermionic effects to the marginal model. Another source of intuition about target complexity is condition number. The unpreconditioned $DD^\dagger$ operator has an average condition number of ∼960, while that with EO preconditioning is ∼195 for test ensembles of 40k gauge fields generated from each flow model. However, EO preconditioning does not prevent the occasional appearance of very large condition numbers—in test ensembles with 40k configurations, condition numbers as large as $O(10^5)$ are found both with and without EO.

Considering the combination of EO with Hasenbusch factorization, one might naively expect the ESS to decrease with more Hasenbusch steps since adding monomials increases the number of pseudofermion variables to model. However, both the joint and marginal ESS improve, with more improvement if more Hasenbusch steps are used. Therefore, the Hasenbusch-preconditioned target must be significantly easier to model, which is confirmed by considering the condition number of the matrices involved; for the case of three factorization steps, in an ensemble of 40k configurations generated from this model the average condition number is $O(10)$ in all four monomials, and the maximum value is less than 100 in all cases. We conclude that while using Hasenbusch requires additional conditional flows and thus larger models overall—with the corresponding increased memory and computing cost—it may provide a systematic approach for increasing the sampling quality.

B. Application to SU(3) in 2D with $N_f = 2$

This section details a demonstration of the gauge-equivariant joint models to sample a 2D SU(3) theory with two fermion flavors. This toy model also has some similar features to QCD, such as the gauge group, confinement, and light mesonlike bound states. It is however intrinsically different than QCD, since 2D SU(N) theories are topologically trivial. For this investigation we choose the following parameters: $\beta = 6$, $\kappa = 0.265$ (corresponding to $m_0 = -0.113208$), and lattice volume L^2 with $L = 16$.



(a) Joint ESS (b) Marginal ESS

FIG. 4. Joint ESS (left) and marginal ESS (right) for flow models trained for the Schwinger model at $L = 8$, $\beta = 2$, and $\kappa = 0.265$ with five different choices of preconditioning. From bottom to top, the curves show results with no preconditioning, using EO preconditioning, and one, two, or three iterations of the Hasenbusch factorization. The training procedure and the marginal architectures (with 24 gauge layers) are the same for all curves. All conditional models have 24 pseudofermion layers. All Hasenbusch monomials are modeled by separate conditional models. The ESS is evaluated with batch size 4096.

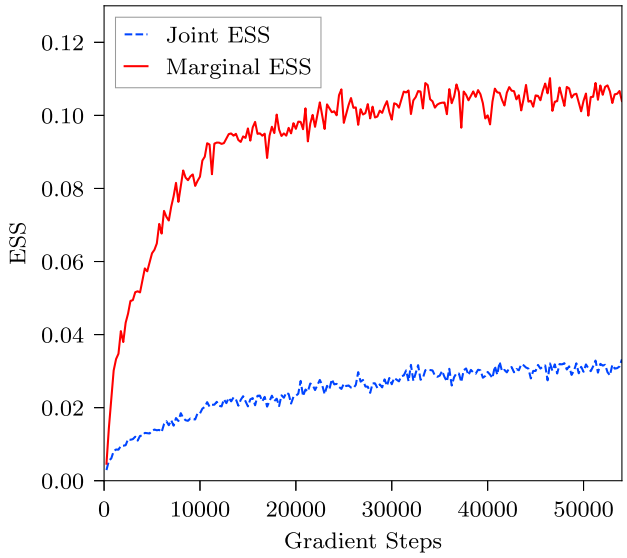


FIG. 5. Marginal and joint ESS of a flow model trained as described in the text for a 2D SU(3) theory with $L = 16$.

Note that for this value of β, the bare gauge coupling is $g_0^2 = 6/\beta = 1$, and so it is in the physically relevant weak-coupling region. This choice of κ results in a pseudoscalar mass of $aM_\pi \simeq 0.72$.

We model a target preconditioned with EO and one step of the Hasenbusch factorization with $\mu_1 = 0.3$. We find that more steps are not necessary, since the Dirac operator has lower condition number than in the Schwinger model. We build the pseudofermion flows alternating spatial and spin variable partitioning, which provides better models than using either scheme alone. We use the PTCN as described in Sec. III C with $n_{\mathrm{PT}} = 6$, and $H = 4$ in the intermediate layers. Increasing the values of the latter parameters does not lead to substantial improvements, while increasing the cost.

Figure 5 shows the joint and marginal ESS along training. A joint ESS of ∼3% is achieved after 54k steps of training. When using this model in sampling, this yields an acceptance rate of ∼8%.

V. CONCLUSION AND OUTLOOK

This work presents a crucial development for scalable flow-based modeling of gauge theories with fermion content. Specifically, we have introduced gauge-equivariant architectures for the pseudofermion approach to the fermion determinant. These architectures are used in the context of joint models, where the probability distribution is split into the marginal and conditional part—the former depends on the gauge variables, and the latter on the pseudofermions with fixed gauge fields. The central piece of technology employed in this work to model the conditional part is the *parallel transport convolutional network*. A PTCN is the composition of several *parallel transport convolutions*, which compute linear combinations of pseudofermions that have been parallel transported to a common site. This is described in depth in Sec. III C.

An important observation is that a large condition number in the Dirac operator makes models for the conditional target harder to optimize. Therefore, we argue that preconditioners should also be used in the context of flow models for pseudofermions. In particular, we describe how the new pseudofermion layers can be easily adapted to be used in conjunction with even/odd preconditioning and the Hasenbusch factorization.

Numerical implementations of the approach are provided in 2D gauge theories, namely the Schwinger model and 2D SU(3). These investigations demonstrate the advantages of preconditioning to obtain higher-quality models, and show how using more pseudofermion draws can systematically improve statistical performance.

The ideas of this work can be applied to four spacetime dimensions without formal complications, enabling flow-based sampling of theories of phenomenological interest such as QCD [68]. In practice, however, significant further engineering will be required to design and train architectures well suited to treating the more complex structures that arise in higher dimensions and in theories with more complex topological features. As discussed at length in Ref. [69], determining whether the approach presented here will provide a viable approach to QCD sampling at state-of-the-art parameters is a critical question which will require direct exploration.

ACKNOWLEDGMENTS

R. A., G. K., D. C. H., F. R. L., and P. E. S. are supported in part by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under grant Contract No. DE-SC0011090. D. C. H., F. R. L., and P. E. S. thank the Institute for Nuclear Theory at the University of Washington for its kind hospitality and stimulating research environment. This research was supported in part by the INT's U.S. Department of Energy Grant No. DE-FG02-00ER41132. P. E. S. is additionally supported by the National Science Foundation under EAGER Grant No. 2035015, by the U.S. DOE Early Career Award No. DE-SC0021006, by a NEC research award, and by the Carl G and Shirley Sontheimer Research Fund. G. K. is additionally supported by the Schweizerischer Nationalfonds. K. C. and M. S. A. are supported by the National Science Foundation under the Award No. PHY-2141336. M. S. A. thanks the Flatiron Institute for their hospitality. D. B. is supported by the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-06CH11357. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy Grant No. EXC 2181/1—390900948 (the

Heidelberg STRUCTURES Excellence Cluster), the Collaborative Research Centre SFB 1225 (ISOQUANT), and the U.S. National Science Foundation under Cooperative Agreement Grant No. PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/). This work 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 No. DEAC02-06CH11357. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center [70] for providing HPC resources that have contributed to the research results reported within this paper. Numerical experiments and data analysis used PYTORCH [71], JAX [72], HAIKU [73], HOROVOD [74], NUMPY [75], and SCIPY [76]. Figures were produced using MATPLOTLIB [77].

APPENDIX A: ESS WITH MULTIPLE PSEUDOFERMIONS SAMPLES

For any gauge field U we can estimate the conditional reweighting factor $w_{N_{\rm pf}}(\{\phi\}|U)$ with $N_{\rm pf}$ samples from the pseudofermion distribution as in Eq. (42). Asymptotically,

$$\begin{aligned}\mathrm{ESS}^{-1}(N_{\rm pf}) &= \langle w(U,\{\phi\})^2\rangle_{q(U,\{\phi\})},\\ &\equiv \int dU\, q(U) w(U)^2 \epsilon^{-1}_{N_{\rm pf}}(U), \end{aligned} \tag{A1}$$

where

$$\begin{aligned}\epsilon^{-1}_{N_{\rm pf}}(U) &= \int \left(\prod_{i=1}^{N_{\rm pf}} d\phi^{(i)} q(\phi^{(i)}|U)\right) w_{N_{\rm pf}}(\{\phi\}|U)^2,\\ &\equiv \langle w_{N_{\rm pf}}(\{\phi\}|U)^2\rangle_{\phi^{(1)}\cdots\phi^{(N_{\rm pf})}}, \end{aligned} \tag{A2}$$

and the second line is simply defining compact notation for the expectations over $q(\cdot|U)$. Using $\langle w\rangle = 1$,

$$\begin{aligned}\epsilon^{-1}_{N_{\rm pf}}(U) &= \frac{1}{N_{\rm pf}^2}\left\langle \sum_i w_i^2 + \sum_{i\neq j} w_i w_j\right\rangle_{\phi^{(1)}\cdots\phi^{(N_{\rm pf})}},\\ &= \frac{1}{N_{\rm pf}}\langle w(\phi|U)^2\rangle_{q(\phi|U)} + \frac{N_{\rm pf}-1}{N_{\rm pf}},\\ &\equiv 1 + \frac{X(U)}{N_{\rm pf}}, \end{aligned} \tag{A3}$$

where $w_i \equiv w(\phi^{(i)}|U)$, and in the last line we have isolated the U and $N_{\rm pf}$ dependence. Inserting back into the expression for the full ESS and evaluating the integrals, we obtain

$$\begin{aligned}\mathrm{ESS}^{-1}(N_{\rm pf}) &= \int dU q(U) w(U)^2 \left[1 + \frac{X(U)}{N_{\rm pf}}\right],\\ &= \mathrm{ESS}^{-1}(\infty) + \frac{C'}{N_{\rm pf}}, \end{aligned} \tag{A4}$$

where $\mathrm{ESS}(\infty)$ is the marginal ESS. The result is identical to Eq. (45) up to algebraic manipulations and redefinition of C'.

APPENDIX B: FURTHER DETAILS OF NUMERICAL EXPERIMENTS

This section provides additional details of the architecture and training scheme for the numerical implementations of the flow models described in Sec. IV.

1. Training and optimization

The self-training scheme uses a loss function that is a stochastic estimate of the Kullback-Leibler divergence [51] computed with q-distributed samples generated by the model

$$\begin{aligned}D_{\rm KL}(q||p) &= \int dU q(U) \log\frac{q(U)}{p(U)},\\ &\approx \frac{1}{B}\sum_{i=1}^{B}\left[\log q(U_i) + S(U_i)\right] + (\text{const}), \end{aligned} \tag{B1}$$

where B is the batch size (i.e., number of field samples generated for each gradient step). The constant does not affect optimization, and it is ignored. We have two independent implementations of the experiments, one using PYTORCH and the other using JAX [72]. In the PYTORCH setup, we train using the ADAMW optimizer [78]. The rest of the parameters correspond to the PYTORCH 1.12 default. We initialize the weights using "Xavier normal" PYTORCH initialization with gain $= 0.5$. In the JAX setup we use the Adam optimizer with gradient clipping and the default Haiku [73] initialization with variance scaling to make the flows closer to the identity map at initialization. The results from both implementations are consistent, but the specific models shown in this paper are trained using the PYTORCH implementation.

2. Context function for the pseudofermion layers

The linear transformation of pseudofermion variables in each PTC is parametrized by the outputs of a context function. To preserve linearity in $\phi(x)$, this context function cannot depend on the pseudofermions. Moreover, using only use gauge-invariant inputs ensures the gauge equivariance of the transformation. In the numerical implementation in this work, in order to respect translation symmetry, we build the context function from 2D convolutions with periodic boundary conditions. The first part of the context function is

shared by all the pseudofermion layers, and maps N_i input channels to N_{hidden} hidden channels. If the Hasenbusch factorization is used, all monomials also share this first part. The second part of the convolution is specific for each pseudofermion layer, and maps the N_{hidden} hidden channels to N_o channels. In intermediate convolutions the ELU [79] activation is applied, and tanh is used in the final one.

We use four input channels to the convolution:

$$\text{Re}\,\text{Tr}\,P_{01}(x), \qquad \text{Im}\,\text{Tr}\,P_{01}(x), \qquad I_0(x), \qquad I_1(x), \tag{B2}$$

where P_{01} is the plaquette and

$$I_i(x) = x_i \mod 4 \tag{B3}$$

is a constant input with periodicity mod 4. We find that including I_i improves the quality, but is nonessential. The number of output channels will depend on the nature of the pseudofermion layer. First, for each PTC in the PTCN, $2 \times 5 \times H \times H'$ real numbers are required to parametrize a generic complex $M_{\alpha\gamma}(x)$ in Eq. (26). Moreover, eight complex numbers are required for $A(x)$ in Eq. (28) if spatial partitioning is used. The neural network outputs are taken as the real and imaginary parts of the entries of all complex matrices, with the exception of $A(x)$ in Eq. (28) to which we add the identity.

3. Hyperparameters for the Schwinger model

First, we list the common hyperparameters for all the flow models in Sec. IV A. For the U(1) gauge layers, we use the architecture and masking pattern described in the Appendix of Ref. [30]. The differences with respect to that work are the number of gauge layers and the number of hidden channels in the context function (here we use 32). We use $\epsilon = 0.02$ in the ADAMW optimizer. The PTCNs have $n_{\text{PT}} = 6$ when EO is used, and $n_{\text{PT}} = 3$ otherwise. $N_{\text{hidden}} = 16$ is used in the context function for the pseudofermion layers. $H = 2$ is used in the intermediate PTCs. These models are constructed only with a spatial partitioning of the pseudofermion variables.

In addition, other hyperparameters vary between Secs. IV A 1 and IV A 2.

(1) The results shown in Fig. 2 and Sec. IV A 1 are generated from a model trained with batch size $B = 1344$, and $L = 16$. The model has 32 pseudofermion layers, as well as 32 gauge layers. The initial learning rate (LR) is $\eta = 7 \times 10^{-4}$, which is reduced by a factor of 0.5 every 20k gradient steps.

(2) All the models in Fig. 4 and Sec. IV A 2 have $B = 4096$, $L = 8$, 24 gauge layers, and 24 pseudofermion layers for each monomial. The LR starts from $\eta = 5 \times 10^{-4}$, and decays by a factor of 0.5 every 10k gradient steps.

4. Hyperparameters for SU(3) in 2D

The flow model described in Sec. IV B has 48 gauge layers and 48 pseudofermion layers in each of the two monomials. The PTCNs have $n_{\text{PT}} = 6$, and $H = 4$ is used in the intermediate PTCs. $N_{\text{hidden}} = 16$ is used in the context function for the pseudofermion layers. Spatial and spin partitioning are alternated in the model. For the SU(3) gauge layers, we use the architecture and masking pattern described in Ref. [26]. The context function for the gauge part has two intermediate standard convolutions, with 32 hidden channels. For training, we use $\epsilon = 0.01$ in the ADAMW optimizer. We use a batch size of 576. The LR scheduling is as follows: beginning from $\eta = 5 \times 10^{-4}$, the LR is decayed by a factor of 0.5 every 10 k gradient steps.

APPENDIX C: HISTOGRAMS WITH OBSERVABLES

In this appendix, we compare distributions of values for observables computed on gauge field configurations generated using HMC to the those computed on configurations from flow models (reweighted). Figures 6 and 7 display histograms comparing two different observables for each of the two fermionic gauge theories studied in this work. In all cases, the observables computed on configurations obtained using both sampling algorithms are statistically consistent.

For the Schwinger model, the flow model of Fig. 2 is used. The observables compared are the average plaquette,

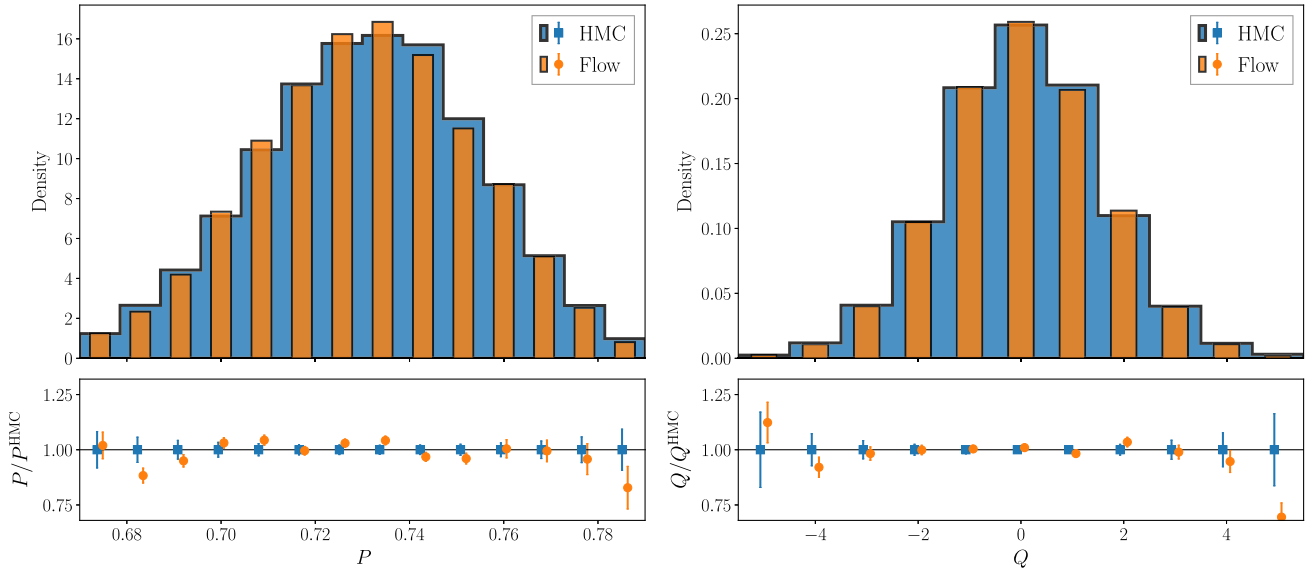
$$P = \frac{1}{L^2}\sum_x \text{Re}\,\text{tr}\,P_{\mu\nu}(x), \tag{C1}$$

and the topological charge,

$$Q = \frac{1}{2\pi}\sum_x \text{Im}\,\log\,P(x). \tag{C2}$$

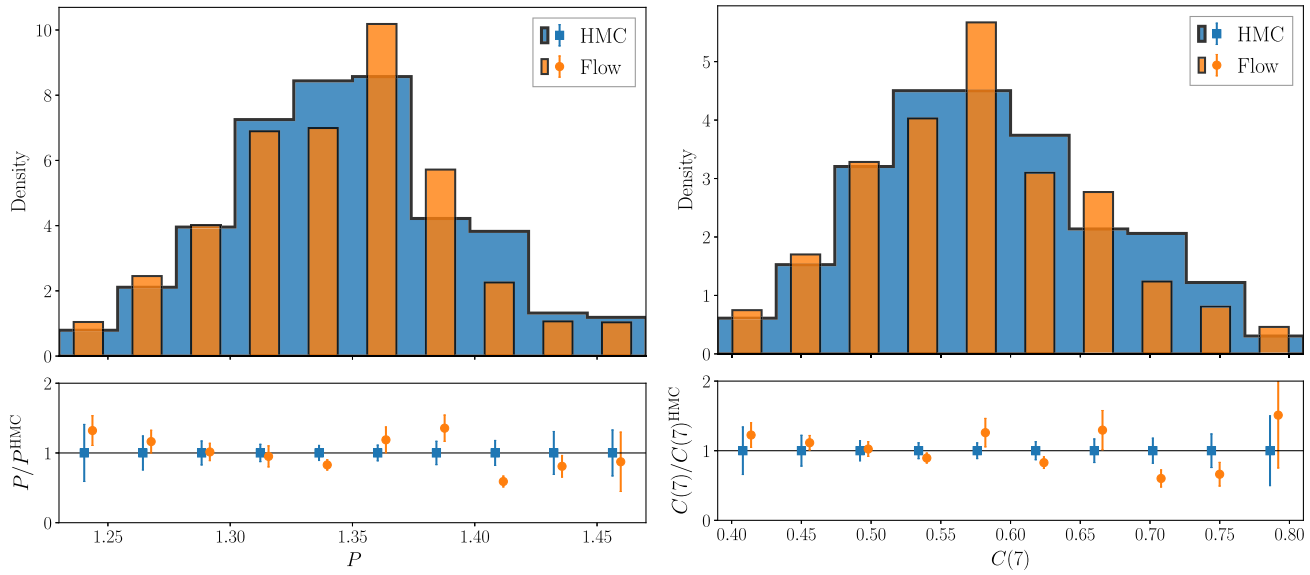
For the SU(3) gauge theory, the flow model of Fig. 5 is used. As an example, two observables are compared: the average plaquette, as in (C1), and the pseudoscalar correlator,

$$C(x_0) = -\sum_{x_1,y_1}\langle[\bar{u}\gamma_5 d](x_0,x_1)[\bar{d}\gamma_5 u](0,y_1)\rangle. \tag{C3}$$



(a) Average plaquette in U(1). (b) Topological charge in U(1).

FIG. 6. Comparison of distributions for two observables in the Schwinger model. The upper panels show density histograms comparing the distributions from HMC and flow-based sampling using reweighting. The lower panels show the ratio of the counts in each bin divided by the central value of the counts obtained in the same bin using the HMC ensemble. The uncertainties are estimated from 200 bootstrap ensembles. Histograms are constructed using comparable statistics, corresponding to approximately $\sim 10\ k$ independent configurations: 52k configurations from the flow model of Fig. 2 using $N_{\mathrm{pf}} = 64$ pseudofermion samples with an ESS $\sim 30\%$, and 130k HMC configurations thinned by a factor of 10 to yield independent samples.



(a) Average plaquette in SU(3). (b) Pseudoscalar correlator at $x_0 = 7$ in SU(3).

FIG. 7. Same as Fig. 6 but for two observables for the 2D fermionic SU(3) theory. Histograms are constructed using comparable statistics, corresponding to approximately ~ 500 independent configurations: 41k configurations from the flow model of Fig. 2 using a single pseudofermion sample with an ESS $\sim 3\%$, and 3200 HMC configurations thinned by a factor of 10 to yield independent samples.

[1] W. Detmold, R. G. Edwards, J. J. Dudek, M. Engelhardt, H.-W. Lin, S. Meinel, K. Orginos, and P. Shanahan (USQCD Collaboration), Eur. Phys. J. A **55**, 193 (2019).
[2] C. Lehner *et al.* (USQCD Collaboration), Eur. Phys. J. A **55**, 195 (2019).
[3] A. S. Kronfeld, D. G. Richards, W. Detmold, R. Gupta, H.-W. Lin, K.-F. Liu, A. S. Meyer, R. Sufian, and S. Syritsyn (USQCD Collaboration), Eur. Phys. J. A **55**, 196 (2019).
[4] P. A. Boyle *et al.*, arXiv:2205.15373.
[5] B. Joó, C. Jung, N. H. Christ, W. Detmold, R. Edwards, M. Savage, and P. Shanahan (USQCD Collaboration), Eur. Phys. J. A **55**, 199 (2019).
[6] D. Boyda *et al.*, arXiv:2202.05838.
[7] L. Wang, Phys. Rev. E **96**, 051301 (2017).
[8] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).
[9] J. Song, S. Zhao, and S. Ermon, in *Advances in Neural Information Processing Systems* (2017), pp. 5140–5150.
[10] A. Tanaka and A. Tomiya, arXiv:1712.03893.
[11] D. Levy, M. D. Hoffman, and J. Sohl-Dickstein, arXiv:1711.09268.
[12] J. M. Pawlowski and J. M. Urban, Mach. Learn. Sci. Tech. **1**, 045011 (2020).
[13] G. Cossu, L. Del Debbio, T. Giani, A. Khamseh, and M. Wilson, Phys. Rev. B **100**, 064304 (2019).
[14] D. Wu, L. Wang, and P. Zhang, Phys. Rev. Lett. **122**, 080602 (2019).
[15] D. Bachtis, G. Aarts, and B. Lucini, Phys. Rev. E **102**, 033303 (2020).
[16] Y. Nagai, A. Tanaka, and A. Tomiya, arXiv:2010.11900.
[17] A. Tomiya and Y. Nagai, arXiv:2103.11965.
[18] D. Bachtis, G. Aarts, F. Di Renzo, and B. Lucini, Phys. Rev. Lett. **128**, 081603 (2022).
[19] D. Wu, R. Rossi, and G. Carleo, Phys. Rev. Res. **3**, L042024 (2021).
[20] D. J. Rezende and S. Mohamed, arXiv:1505.05770.
[21] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv:1605.08803.
[22] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, J. Mach. Learn. Res. **22**, 1 (2021).
[23] S.-H. Li and L. Wang, Phys. Rev. Lett. **121**, 260601 (2018).
[24] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Phys. Rev. D **100**, 034515 (2019).
[25] 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).
[26] 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).
[27] 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.
[28] 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).
[29] M. S. Albergo, D. Boyda, D. C. Hackett, G. Kanwar, K. Cranmer, S. Racanière, D. J. Rezende, and P. E. Shanahan, arXiv:2101.08176.
[30] 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).
[31] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel, Phys. Rev. E **101**, 023304 (2020).
[32] 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).
[33] S. Foreman, X.-Y. Jin, and J. C. Osborn, in *Proceedings of the 9th International Conference on Learning Representations* (2021), arXiv:2105.03418.
[34] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya, Proc. Sci., LATTICE2021 (**2022**) 073 [arXiv:2112.01586].
[35] S. Foreman, X.-Y. Jin, and J. C. Osborn, Proc. Sci., LATTICE2021 (**2022**) 508 [arXiv:2112.01582].
[36] L. Del Debbio, J. M. Rossney, and M. Wilson, Phys. Rev. D **104**, 094507 (2021).
[37] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, Proc. Natl. Acad. Sci. U.S.A. **119**, e2109420119 (2022).
[38] P. de Haan, C. Rainone, M. C. N. Cheng, and R. Bondesan, arXiv:2110.02673.
[39] S. Lawrence and Y. Yamauchi, Phys. Rev. D **103**, 114509 (2021).
[40] X.-Y. Jin, Proc. Sci., LATTICE2021 (**2022**) 600 [arXiv:2201.01862].
[41] J. M. Pawlowski and J. M. Urban, arXiv:2203.01243.
[42] J. Finkenrath, arXiv:2201.02216.
[43] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, arXiv:2207.00283.
[44] A. Singha, D. Chakrabarti, and V. Arora, arXiv:2207.00980.
[45] A. G. D. G. Matthews, M. Arbel, D. J. Rezende, and A. Doucet, arXiv:2201.13117.
[46] M. Caselle, E. Cellini, A. Nada, and M. Panero, J. High Energy Phys. 07 (2022) 015.
[47] M. Favoni, A. Ipp, D. I. Müller, and D. Schuh, Phys. Rev. Lett. **128**, 032003 (2022).
[48] K. G. Wilson, Phys. Rev. D **10**, 2445 (1974).
[49] K. G. Wilson, in *Proceedings of the 13th International School of Subnuclear Physics: New Phenomena in Subnuclear Physics* (1975).
[50] J. R. Shewchuk, Technical Report, USA, 1994.
[51] S. Kullback and R. A. Leibler, Ann. Math. Stat. **22**, 79 (1951).
[52] S. Sabour, N. Frosst, and G. E. Hinton, Adv. Neural Inf. Process. Syst. **30** (2017).
[53] T. A. DeGrand and P. Rossi, Comput. Phys. Commun. **60**, 211 (1990).
[54] M. Hasenbusch, Phys. Lett. B **519**, 177 (2001).
[55] C. Andrieu and G. O. Roberts, Ann. Stat. **37**, 697 (2009).
[56] J. S. Schwinger, Phys. Rev. **128**, 2425 (1962).
[57] S. R. Coleman, R. Jackiw, and L. Susskind, Ann. Phys. (N.Y.) **93**, 267 (1975).
[58] J. Smit and J. C. Vink, Nucl. Phys. **B303**, 36 (1988).
[59] H. Dilger, Phys. Lett. B **294**, 263 (1992).
[60] H. Dilger, Int. J. Mod. Phys. C **06**, 123 (1995).
[61] S. Durr, Phys. Rev. D **85**, 114503 (2012).
[62] L. Funcke, K. Jansen, and S. Kühn, Phys. Rev. D **101**, 054507 (2020).
[63] N. Butt, S. Catterall, Y. Meurice, R. Sakai, and J. Unmuth-Yockey, Phys. Rev. D **101**, 094509 (2020).

[64] M. C. Bañuls *et al.*, Eur. Phys. J. D **74**, 165 (2020).
[65] D. Albandea, P. Hernández, A. Ramos, and F. Romero-López, Eur. Phys. J. C **81**, 873 (2021).
[66] T. Hartung, K. Jansen, F. Y. Kuo, H. Leövey, D. Nuyens, and I. H. Sloan, Proc. Sci., LATTICE2021 (**2022**) 010 [arXiv:2112.05069].
[67] T. Eichhorn and C. Hoelbling, Proc. Sci., LATTICE2021 (**2022**) 573 [arXiv:2112.05188].
[68] R. Abbott *et al.*, arXiv:2208.03832.
[69] R. Abbott *et al.*, Aspects of scaling and scalability for flow-based sampling of lattice QCD (to be published).
[70] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell *et al.*, *Proceedings of the 2018 IEEE High Performance extreme Computing Conference (HPEC)* (2018), pp. 1–6, arXiv:1807.07814.
[71] A. Paszke *et al.*, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), pp. 8024–8035, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
[72] 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).
[73] T. Hennigan, T. Cai, T. Norman, and I. Babuschkin, http://github.com/deepmind/dm-haiku (2020).
[74] A. Sergeev and M. Del Balso, arXiv:1802.05799.
[75] 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).
[76] 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).
[77] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).
[78] I. Loshchilov and F. Hutter, arXiv:1711.05101.
[79] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, arXiv:1511.07289.