Sampling using SU(N) gauge equivariant flows

Denis Boyda,[1,*] Gurtej Kanwar,[1,†] Sébastien Racanière,[2,‡] Danilo Jimenez Rezende,[2,§] Michael S. Albergo,[3]
Kyle Cranmer,[3] Daniel C. Hackett,[1] and Phiala E. Shanahan[1]

[1] *Center for Theoretical Physics, Massachusetts Institute of Technology,*
Cambridge, Massachusetts 02139, USA
[2] *DeepMind, London N1C 4AG, United Kingdom*
[3] *Center for Cosmology and Particle Physics, New York University, New York, New York 10003, USA*

(Received 24 September 2020; accepted 16 March 2021; published 20 April 2021)

We develop a flow-based sampling algorithm for SU(N) lattice gauge theories that is gauge invariant by construction. Our key contribution is constructing a class of flows on an SU(N) variable [or on a U(N) variable by a simple alternative] that respects matrix conjugation symmetry. We apply this technique to sample distributions of single SU(N) variables and to construct flow-based samplers for SU(2) and SU(3) lattice gauge theory in two dimensions.

DOI: 10.1103/PhysRevD.103.074504

I. INTRODUCTION

Gauge theories based on SU(N) or U(N) groups describe many aspects of nature. For example, the Standard Model of nuclear and particle physics is a non-Abelian gauge theory with the symmetry group U(1)× SU(2) × SU(3), candidate theories for physics beyond the Standard Model can be defined based on strongly interacting SU(N) gauge theories [1,2], SU(N) gauge symmetries emerge in various condensed matter systems [3–7], and SU(N) and U(N) gauge symmetries feature in the low energy limit of certain string-theory vacua [8]. In the context of the rapidly developing area of machine-learning applications to physics problems, the incorporation of gauge symmetries in machine learning architectures is thus of particular interest [9–14].

Here, we demonstrate how SU(N) gauge symmetries can be incorporated into *flow-based models* [15]. These models use a parametrized invertible transformation (a “flow”) to construct a variational ansatz for a target probability distribution that can be optimized via machine learning techniques to enable efficient sampling. We detail the application of this approach to lattice field theory calculations, for which such samplers have been found to offer potentially significant advantages over more traditional sampling algorithms [11,16,17].

A general approach to incorporating a symmetry in flow-based sampling models is to construct the models in terms of invertible transformations that are *equivariant* to symmetry operations, meaning that the transformation and symmetry operations commute. For any gauge theory with a continuous gauge group, we showed in Ref. [11] that a gauge equivariant transformation that simultaneously remains equivariant under a large subgroup of spacetime translations can be constructed in terms of a *kernel*: a transformation that acts on elements of the gauge group and is equivariant under matrix conjugation, $U \to XUX^{-1}$, where U and X are elements of the gauge group in the fundamental matrix representation. In Ref. [11], this approach was demonstrated in the context of U(1) gauge theory. Here, we develop a class of kernels for SU(N) group elements (and describe a similar construction for U(N) group elements). We show that if an invertible transformation acts only on the eigenvalues of a matrix and is equivariant under permutation of those eigenvalues, then it is equivariant under matrix conjugation and may be used as a kernel. Moreover, by making a connection to the maximal torus within the group and to the Weyl group of the root system, we show that this is in fact a universal way to define a kernel for unitary groups.

The application of flow-based models to lattice field theory is reviewed briefly in Sec. II A. Methods to impose symmetries in these models are reviewed in Sec. II B, and Sec. II C describes our particular approach to imposing gauge symmetry in flow-based models using single-variable kernels. In Sec. III, we construct kernels for SU(N) variables and investigate sampling from distributions over such variables, including the marginal distributions relevant

[*] boyda@mit.edu
[†] gurtej@mit.edu
[‡] sracaniere@google.com
[§] danilor@google.com

for plaquettes in two-dimensional (2D) lattice gauge theory. Finally, in Sec. IV, we use these kernels to construct gauge-symmetric flow-based samplers for SU(2) and SU(3) lattice gauge theory in two dimensions and demonstrate that observables in these theories are exactly reproduced by the flow-based sampling approach.

II. FLOW-BASED SAMPLING FOR LATTICE GAUGE THEORY

Lattice quantum field theory provides a nonperturbative regularization of the path integral by discretizing the theory onto a spacetime lattice. In Euclidean spacetime, the regularized expectation value of an observable $\mathcal{O}$ is defined in terms of the discretized action $S(U)$ by

$$\langle\mathcal{O}\rangle = \frac{1}{Z}\int \mathcal{D}U\,\mathcal{O}(U)e^{-S(U)}, \quad Z = \int \mathcal{D}U\,e^{-S(U)}, \quad (1)$$

where $\int\mathcal{D}U$ integrates over all degrees of freedom of the discretized quantum field U. We denote by $U_\mu(x) \in G$ the element of U on link $(x, x+\hat{\mu})$, where $\mu \in \{1, 2, \ldots, D\}$ is the spacetime direction of the link, $x \in \mathbb{Z}^D$ indicates a site on the D-dimensional spacetime lattice, and G is the structure group of the gauge theory; for many relevant physical theories, the structure groups are Lie groups. The path integral measure $\mathcal{D}U$ for a lattice gauge theory is a product of the Haar measure of G per link.

Equation (1) can be evaluated numerically by sampling *configurations* from the probability distribution $p(U) = e^{-S(U)}/Z$, which is typically undertaken using Markov chain methods [18]. In Refs. [11,16], we developed an approach to evaluate Eq. (1) for lattice field theories by sampling independent configurations from a flow-based model optimized to approximate $p(U)$, where unbiased estimates of observables can be obtained from this approximate distribution by either a reweighting technique or a Metropolis accept/reject step.[1] Flow-based methods can similarly be applied to statistical theories (with continuous degrees of freedom) by replacing the field configurations U of Eq. (1) with microstates, replacing the action with the Hamiltonian over temperature, $S \to H/k_B T$, and interpreting the distribution as the Boltzmann distribution [21–25].

A. Sampling gauge configurations using flows

A flow-based sampler consists of two components which are as follows:

(1) A *prior distribution*[2] $r(V)$ that is easily sampled

(2) An invertible function, or flow, f that has a tractable Jacobian factor

Here, we restrict discussion to flow-based models targeting distributions $p(U)$ on Lie groups $\mathcal{G}$, for which $U \in \mathcal{G}$ and $f: \mathcal{G} \to \mathcal{G}$. The group could be a product of structure groups $\mathcal{G} = G \otimes G \otimes \ldots$, as in the case of lattice gauge theory, or an unfactorizable group such as SU(N) or U(N). Generating a sample from the model proceeds by first sampling from the prior distribution $r(V)$, then applying f to produce $U = f(V)$. In general, the invertible function f stretches and concentrates the density of points over the domain; thus, the output samples are distributed according to a new effective distribution $q(U)$. The output density can be explicitly computed in terms of the log-det-Jacobian of f, LDJ$_f$,

$$q(U) = \frac{r(V)}{e^{\mathrm{LDJ}_f(V)}}, \qquad e^{\mathrm{LDJ}_f(V)} \coloneqq \left|\det_{ij}\frac{\partial[f(V)]_i}{\partial V_j}\right|. \quad (2)$$

Here, the indices i and j run over directions in the Lie algebra of $\mathcal{G}$ translated to $f(V)$ and V, respectively [26].

When f is parametrized[3] by a collection of model parameters ξ, the model output distribution $q(U)$ can be considered a variational ansatz for the target distribution $p(U)$. Its free parameters can be optimized to produce an approximation to the target distribution, $q(U) \approx p(U)$, by applying stochastic gradient descent to a loss function defined to be a measure of the divergence between $q(U)$ and $p(U)$. For this optimization to be viable without a large body of training data from existing samplers, we must be able to approximate the divergence and its gradients using only samples from the model and the functional form of the action. This may be achieved by employing the Kullback-Leibler (KL) divergence between the two distributions as a loss function,

$$D_{\mathrm{KL}}(q||p) \coloneqq \int \mathcal{D}U\,q(U)[\log q(U) - \log p(U)] \geq 0. \quad (3)$$

For lattice theories, it is convenient to shift the KL divergence to remove the (unknown) constant $\log Z$, defining a modified KL divergence [21],[4]

$$D'_{\mathrm{KL}}(q||p) \coloneqq \int \mathcal{D}U q(U)[\log q(U) + S(U)] \geq -\log Z. \quad (4)$$

The gradients and location of the minimum are unaffected by this constant shift. The KL divergence can then be

[1] Sampling for lattice field theories based on generative adversarial networks has also been investigated in related work [19,20].

[2] We specify the distribution using a density function $r(V)$. Here and in the following, this is implicitly a density with respect to the path integral measure $\mathcal{D}V$ (or $\mathcal{D}U$).

[3] The prior $r(V)$ may also be parametrized, though parameters controlling deterministic transformations of stochastic variables, as in f, have been shown to be easier to optimize [27–29].

[4] This can be considered a special case of the variational lower bound [30].

stochastically estimated by drawing samples U from the model and computing the sample mean of $\log q(U) + S(U)$, from which stochastic gradients with respect to the model parameters ξ can be computed via backpropagation.

It is illuminating to consider the variational ansatz as defining a family of *effective actions*, any of which we can directly sample, i.e., the model density can be interpreted as arising from the effective action $S_{\mathrm{eff}}(U) \coloneqq -\log(q(U))$. The ability to both compute the effective action and sample from it enables producing unbiased estimates of observables under the true distribution. For example, a reweighting approach can be used [23], in which the vacuum expectation value of an operator $\mathcal{O}$ can be computed as

$$\langle \mathcal{O} \rangle = \frac{\int \mathcal{D}U q(U)[\mathcal{O}(U)w(U)]}{\int \mathcal{D}U q(U)[w(U)]} = \frac{\langle \mathcal{O}(U)w(U)\rangle_{S_{\mathrm{eff}}}}{\langle w(U)\rangle_{S_{\mathrm{eff}}}},$$

where $w(U) = \exp(-S(U) + S_{\mathrm{eff}}(U))$. (5)

Since S_{eff} is an approximation of the true action, the reweighting factors $w(U)$ will vary with U. A measure of the quality of the reweighted ensemble is the *effective sample size* (ESS),

$$\mathrm{ESS} \coloneqq \frac{(\frac{1}{n}\sum_i w(U_i))^2}{\frac{1}{n}\sum_i w(U_i)^2}, \qquad U_i \sim q(U), \tag{6}$$

which is normalized relative to the total number of samples n such that $\mathrm{ESS} = 1$ for a perfect model. This reweighting approach is computationally efficient when computing observables is inexpensive relative to drawing samples from the model, because the extra cost of computing observables on samples which will be severely downweighted is small.

When computing observables is instead expensive relative to drawing samples from the model, producing unbiased estimates of observables by resampling techniques can be more efficient than reweighting. A *flow-based Markov chain* is one such approach [11,16].[5] In a flow-based Markov chain, samples from the model are used as proposals for each step of the chain, with a Metropolis accept/reject step to guarantee asymptotic exactness. Each proposal is independent of the previous configuration in the chain, and therefore the appropriate acceptance probability is

$$p_{\mathrm{acc}}(U \to U') = \min\left(1, \frac{p(U')}{q(U')}\frac{q(U)}{p(U)}\right). \tag{7}$$

When the model closely approximates the target, $q(U) \approx p(U)$, the acceptance rate will be close to 1. Rejections duplicate the previous state of the chain, and observables only need to be computed once on each sequence of duplicated samples in the chain. Essentially, the Markov chain approach acts as an integer rounding of the reweighting factors, and thus resources are efficiently allocated toward computing observables only on sufficiently likely configurations. In the flow-based Markov chain, the analog of the effective sample size is determined by correlations between sequential configurations; these correlations are introduced entirely through rejections, since proposals are independently drawn from the model.

The efficiency of the flow-based sampling approach hinges on implementing a general and well-parametrized function f, which must be invertible and for which LDJ_f must be tractable. A powerful approach to constructing such functions is through composition of simpler functions g_i,

$$f(V) \coloneqq g_n(g_{n-1}(\ldots g_1(V)\ldots)). \tag{8}$$

When each g_i is invertible and has a tractable log-det-Jacobian, f satisfies these properties as well. In the following sections, we choose the g_i to be *coupling layers*: functions that act elementwise on a subset of the components of the input, conditioned on the complimentary ("frozen") subset. This structure guarantees a triangular Jacobian matrix, allowing LDJ_f to be efficiently computed from the diagonal elements of the matrix. Coupling layers generally guarantee invertibility by defining the transformation as an explicitly invertible operation on the input. For example, a coupling layer could transform a link in a gauge configuration by left multiplication with a group element that only depends on nearby frozen links and model parameters, ξ,

$$U_\mu(x) \xrightarrow{\text{e.g.}} U'_\mu(x) = W_\xi(\text{frozen neighbors})U_\mu(x), \tag{9}$$

where $W_\xi(\text{frozen neighbors}) \in G$. Regardless of the function W_ξ, this transformation is invertible: to undo it, we compute $[W_\xi(\text{frozen neighbors})]^{-1}$ and left multiply. In our models, the g_i each depends on an independent subset of the model parameters, though sharing parameters is an interesting possibility for future exploration.

In general, coupling layers are written in terms of functions of the frozen links and model parameters (analogous to W_ξ in the example above), which we call *context functions*. The outputs of these context functions are used to transform the input in a manifestly invertible way, but the functions themselves may be arbitrary, up to producing output in the correct domain (in our example, returning values in G). These functions are therefore typically implemented in terms of feed-forward neural networks, with the model parameters ξ specifying the neural network weights.

[5] In some situations, either bootstrap resampling with weights (also known as sampling importance resampling) [31] or rejection sampling may be useful. In the former approach, the ensemble size cannot easily be expanded, while in the latter, a multiplicative factor M must be chosen such that $Mq(U) \geq p(U)$ while avoiding excessive rejection; these challenges motivate the use of flow-based Markov chain Monte Carlo (MCMC) in this work.

B. Symmetries in flow models

Symmetries in a lattice gauge theory manifest as transformations of field configurations that leave the action invariant for all field configurations. We write the transformation t acting on a field configuration U as $t \cdot U$; a group of transformations H is then a symmetry group when $S(U) = S(t \cdot U)$ for all $t \in H$ and all U. Lattice actions $S(U)$ are commonly constructed to preserve discrete geometric symmetries of the Euclidean spacetime as well as internal symmetries. In particular, actions are typically invariant under the following:

(1) Discrete translational symmetry group, $T = \{T_{\delta x, \delta y}\}$, where $\delta x, \delta y$ enumerate all possible lattice offsets

(2) Hypercubic symmetry group $R = \{R_i\}$, where i enumerates all $2^D(D!)$ unique combinations of rotations and reflections of the D-dimensional hypercube[6]

(3) Gauge symmetry group, where each element Ω can be defined as a group-valued field over lattice sites, $\Omega(x) \in G$, that transforms links of a field configuration as follows:

$$(\Omega \cdot U)_\mu(x) = \Omega(x) U_\mu(x) \Omega^\dagger(x + \hat{\mu}). \tag{10}$$

Any expressive flow-based model should approximately reproduce the symmetries of the original action after optimization, even if these symmetries are not imposed in the model. Exact symmetries are recovered on average in the sampled distribution after reweighting or composing samples into a Markov chain. Nevertheless, any breaking of the symmetries in the model reflects differences between the model and target distribution, and is thus associated with sampling inefficiencies in the form of increased variance or correlations in the Markov chain. Imposing symmetries explicitly in the form of the model effectively reduces the variational parameter space to include only symmetry-respecting maps, i.e., those that factorize the distribution. An example of such factorization is illustrated for gauge symmetry in Fig. 1. In many machine learning contexts, it has been found that explicitly preserving the symmetries of interest in models improves both the optimization costs and ultimate model quality [22,32–37]. For example, gauge symmetry is a large symmetry group with dimension proportional to the number of lattice sites; in our study of U(1) gauge theory in Ref. [11], it was shown that imposing this symmetry exactly was necessary to construct flow-based samplers of comparable or better efficiency than traditional sampling approaches.

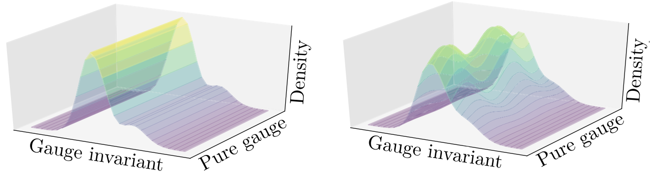

FIG. 1. Left: distributions that exactly respect gauge symmetry factor over the degrees of freedom, such that they have uniform density in the pure-gauge degrees of freedom and a nontrivial density only in the gauge invariant degrees of freedom. Right: arbitrary distributions on the space of gauge configurations do not factor, and uniformity in the pure-gauge direction must be approximately learned by the model.

Interactions between symmetry groups are also an important consideration. For example, a simple way to achieve the factorization of the model distribution depicted in Fig. 1 would be to employ a gauge fixing procedure that reduces configurations to gauge invariant degrees of freedom only and sample only in the remaining lower-dimensional space. This could be achieved with a maximal tree gauge fixing [38,39]. However, gauge fixing procedures like the maximal tree procedure that explicitly factorizes the degrees of freedom are not translationally invariant. On the other hand, gauge fixing procedures based on implicit differential equation constraints instead of an explicit factorization are known to preserve translational invariance in the path integral formulation [40], but it is unclear how to restrict flow-based models to act on configurations satisfying these constraints. Recent work in the Hamiltonian formulation has suggested ways to factor out pure-gauge degrees of freedom for U(1) gauge theory, but it is not clear whether this can be extended to $\mathrm{SU}(N)$ gauge theory or the path integral formulation [41]. Here we develop an approach to simultaneously impose gauge and translational symmetries on models acting on all of the degrees of freedom of an $\mathrm{SU}(N)$ gauge field, without any preemptive factorization along the lines of gauge fixing.

To preserve a symmetry in a flow-based sampling model, it is sufficient to sample from a prior distribution that is exactly invariant under the symmetry and transform the samples using an invertible transformation that is *equivariant* under the symmetry [42–44], meaning that symmetry transformations t commute with application of the function,

$$f(t \cdot U) = t \cdot f(U). \tag{11}$$

For lattice gauge theories, a uniform prior distribution (with respect to the product Haar measure) is easily sampled and is symmetric under translations, hypercubic symmetries, and gauge symmetry. Equivariance of the map f can be guaranteed by ensuring that the individual coupling layers in the decomposition of f are each equivariant,

$$\begin{aligned} & g_i(t \cdot U) = t \cdot g_i(U) \\ \Rightarrow\ & f(t \cdot U) = g_n(g_{n-1}(\ldots g_1(t \cdot U)\ldots)) = t \cdot f(U). \end{aligned} \tag{12}$$

[6] These operations represent the symmetry about a distinguished point on the lattice. In general, the whole geometric symmetry group is given by the combination of this group with the translational symmetry group.

In our approach [11], coupling layers decompose the components of a field configuration by spacetime location, and therefore making coupling layers equivariant to spacetime symmetries (translational and hypercubic symmetries) and making coupling layers equivariant to internal symmetries (such as gauge symmetry) must be handled in different ways, but can be simultaneously achieved.

It has been noted that convolutional neural networks are equivariant to discrete translations, and a similar approach can extend equivariance to rotations and reflections [9,32]. For lattice gauge theory, using these equivariant networks acting on the frozen links inside each coupling layer *and choosing symmetric decompositions* into frozen and updated links ensures, each coupling layer is equivariant under (a large subgroup of) translations. For example, in Sec. IV, we construct models for two-dimensional gauge theory using convolutional neural networks with a decomposition pattern that repeats after offsets by four sites in both directions on the lattice, resulting in equivariance under the translational symmetry group modulo $\mathbb{Z}_4 \times \mathbb{Z}_4$. Though the full translational symmetry group is not preserved exactly, the residual group that must be learned has a fixed size independent of the lattice volume.

Internal symmetries, on the other hand, do not mix links at different spacetime locations. The symmetry transformations acting on the frozen links already commute through the coupling layer. The updated links, however, must be transformed specifically to guarantee equivariance. Generally, this can be achieved by making the context function [i.e., the analog of W_ξ acting on frozen links in Eq. (9)] *invariant* to symmetry transformations, and defining how the function is applied to the remaining links such that the operation commutes with symmetry transformations. This must be done based on the form of the symmetry group; we review how this can be achieved for the case of gauge symmetries in the following section.

C. Gauge equivariance

In Ref. [11], we presented a framework for the construction of coupling layers that are equivariant under gauge symmetries. At a high level, each coupling layer is constructed to the following:

(1) Change variables to open (untraced) loops of links that start and end at a common point

(2) Act on these loops in a way that is equivariant under matrix conjugation; we call the function acting in this way a kernel

(3) Change variables back to links to compute the resulting action on the gauge configuration

Under a gauge transformation, each open loop transforms by matrix conjugation. The kernel acting on open loops is equivariant under matrix conjugation; thus, the whole coupling layer is gauge equivariant. Matrix conjugation leaves the set of eigenvalues, i.e., the *spectrum*, of the open loop invariant. Arranging the coupling layer in terms of the spectra of open loops thus allows the flow to directly manipulate these physical, gauge invariant, marginal distributions independently of the pure-gauge degrees of freedom.

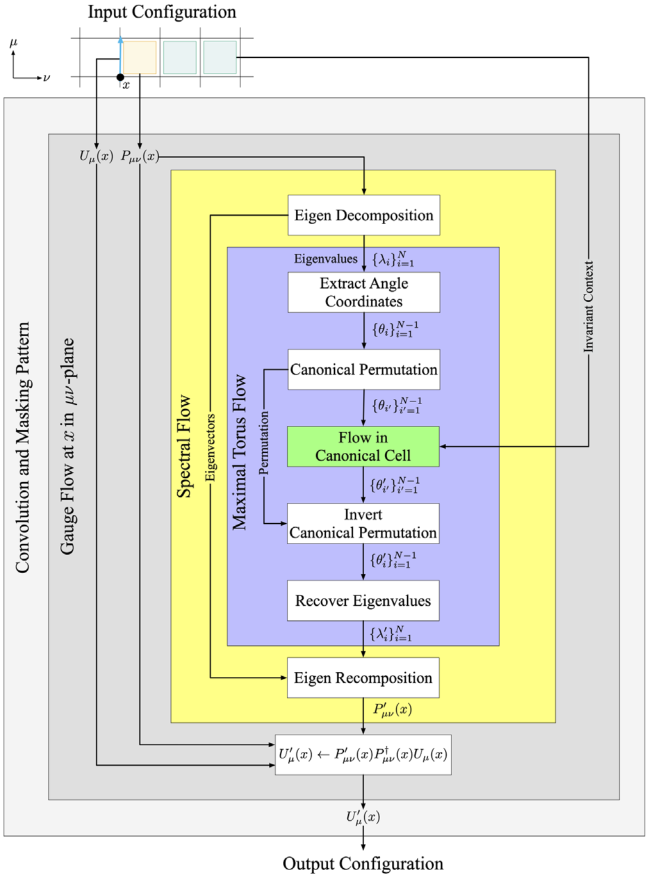

FIG. 2. Decomposition of a single gauge equivariant coupling layer. Outer gray sections depict the general formulation of gauge equivariant flows detailed in Ref. [11]. Inner colored sections detail the kernel we construct in Sec. III for a single SU(N) variable.

In our implementation, we use 1 × 1 loops, or *plaquettes*, as the open loops transformed by the kernel. The plaquette oriented in the $\mu\nu$ plane and located at site x is defined in terms of the links by[7]

$$P_{\mu\nu}(x) \coloneqq U_\mu(x)U_\nu(x+\hat{\mu})U^\dagger_\mu(x+\hat{\nu})U^\dagger_\nu(x). \qquad (13)$$

A subset of plaquettes is transformed by the kernel, while the traces of unmodified plaquettes are used as gauge invariant input to the context functions in the transformation.[8] After the kernel acts on untraced plaquettes, $P_{\mu\nu}(x) \to P'_{\mu\nu}(x)$, we

[7] Note that there is no trace and $P_{\mu\nu}(x)$ is matrix valued.

[8] The use of plaquettes as the open loops and gauge invariant inputs is one of the many possible choices. For either the open loops or gauge invariant inputs, plaquettes could be replaced or augmented by other choices of loops.

change variables back to links and implement the update on the gauge configuration as

$$U'_\mu(x) = P'_{\mu\nu}(x) P^\dagger_{\mu\nu}(x) U_\mu(x), \tag{14}$$

so that the plaquette is updated as desired,

$$U'_\mu(x) U_\nu(x+\hat{\mu}) U^\dagger_\mu(x+\hat{\nu}) U^\dagger_\nu(x) = P'_{\mu\nu}(x). \tag{15}$$

Equivariance under matrix conjugation ensures that output plaquettes transform appropriately under the gauge symmetry, $(\Omega \cdot P')_{\mu\nu}(x) = \Omega(x) P'_{\mu\nu}(x) \Omega^\dagger(x)$, and therefore the output configuration does as well,

$$\begin{aligned}(\Omega \cdot U')_\mu(x) &= [\Omega(x) P'_{\mu\nu}(x) \Omega^\dagger(x)][\Omega(x) P^\dagger_{\mu\nu}(x) \Omega^\dagger(x)] \\ &\quad \times [\Omega(x) U_\mu(x) \Omega^\dagger(x+\hat{\mu})] \\ &= \Omega(x) U'_\mu(x) \Omega^\dagger(x+\hat{\mu}).\end{aligned} \tag{16}$$

This general construction is schematically depicted in the outer, gray sections of Fig. 2.

Finally, to ensure invertibility, we require that the term $P^\dagger_{\mu\nu}(x) U_\mu(x) = U_\nu(x) U_\mu(x+\hat{\nu}) U^\dagger_\nu(x+\hat{\mu})$ in Eq. (14) does not contain any links that are updated as a result of other plaquettes being transformed. In our construction, we must choose the subsets of loops to transform, and the corresponding links to update, in such a way that any loop that is *actively* transformed is not also modified *passively* as a byproduct of another loop being transformed. There are many possible ways to choose subsets satisfying these constraints; to ensure that all links are updated, we should also choose different subsets of loops to update in each coupling layer. For example, in our application to two-dimensional gauge theory, we choose to update rows or columns of plaquettes that are spaced four sites apart, with a repeating cycle of offsets and rotations in each successive coupling layer, as depicted in Fig. 3. Note that in the figure the subsets of plaquettes that are actively and passively updated are disjoint in all coupling layers. This updating scheme is also applicable to higher spacetime dimensions, as the actively and passively updated plaquettes will similarly be disjoint in each coupling layer.

In Ref. [11], we applied this general gauge equivariant construction to U(1) gauge theory. Our contribution in the present work is the development of transformations that are equivariant under matrix conjugation in SU(N) [with a straightforward adaptation to U(N)] which can be used as kernels for gauge equivariant coupling layers in SU(N) or U(N) lattice gauge theory. This novel contribution is depicted in the inner, colored sections of Fig. 2. We detail these transformations in the next section.

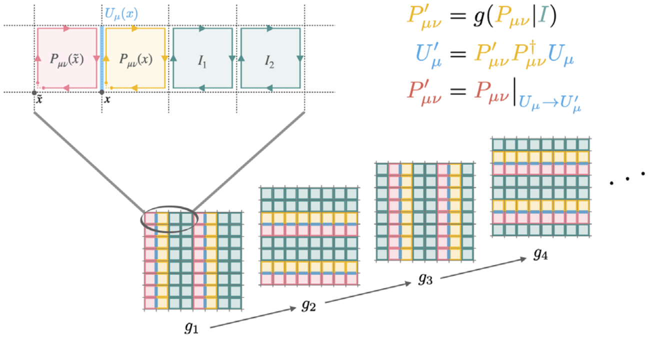
$U_\mu(x)$
$P_{\mu\nu}(\tilde{x})$
$P_{\mu\nu}(x)$
I_1
I_2
$\tilde{x}$
x
$P'_{\mu\nu} = g(P_{\mu\nu}|I)$
$U'_\mu = P'_{\mu\nu} P^\dagger_{\mu\nu} U_\mu$
$P'_{\mu\nu} = P_{\mu\nu}|_{U_\mu \to U'_\mu}$
g_1
g_2
g_3
g_4
. . .

FIG. 3. Our choice of plaquettes to update [$P_{\mu\nu}(x)$, yellow], gauge invariant context for that transformation [I_1 and I_2, green], the corresponding updated link [$U_\mu(x)$, blue], and the plaquettes passively modified as a result of the link update [$P_{\mu\nu}(\tilde{x})$, red] for two-dimensional gauge theory. A repeating cycle of rotations and translations are applied to the pattern for successive coupling layers; composition of eight coupling layers is sufficient to update every link once for this pattern.

III. FLOW MODELS FOR SINGLE SU(N) VARIABLES

The key component of a gauge equivariant flow-based model is a kernel: an invertible map that acts on a single group-valued variable and is equivariant under matrix conjugation. Specifically, an invertible map $h: G \to G$ is a kernel if $h(XUX^{-1}) = Xh(U)X^{-1}$ for all $U, X \in G$. In constructing a gauge equivariant flow-based model, the kernel is used to transform untraced loops of links starting and ending at a common point (whose spectrum has physical, gauge invariant meaning). Here, we specify a general method to construct such kernels and investigate application of these kernels to sampling probability densities on single SU(N) or U(N) variables (representing marginal distributions on open loops in the full gauge theory).

In the language of groups, a kernel should move density *between* conjugacy classes while preserving structure *within* those classes. Each conjugacy class is defined by a set $\{XUX^{-1}: X \in G\}$, for some U. It is useful, however, to think of each conjugacy class in SU(N) or U(N) as a set of all matrices with some particular spectrum; e.g., all matrices with eigenvalues $\{e^{i3\pi/12}, e^{i5\pi/12}, e^{-i8\pi/12}\}$ form a conjugacy class in SU(3). Intuitively, a kernel should therefore move density between possible N-tuples of eigenvalues while preserving the eigenvector structure. In Appendix A, we prove that this intuition is exact: a kernel can generally be defined as an invertible map that acts on the list of eigenvalues of the input matrix, is equivariant under permutations of the eigenvalues, and leaves the eigenvectors unchanged. In our applications, we therefore structure the kernel to accept a matrix-valued input, diagonalize it to produce a (arbitrarily ordered) list of eigenvalues and eigenvectors, transform the eigenvalues in a permutation equivariant fashion, then reconstruct the

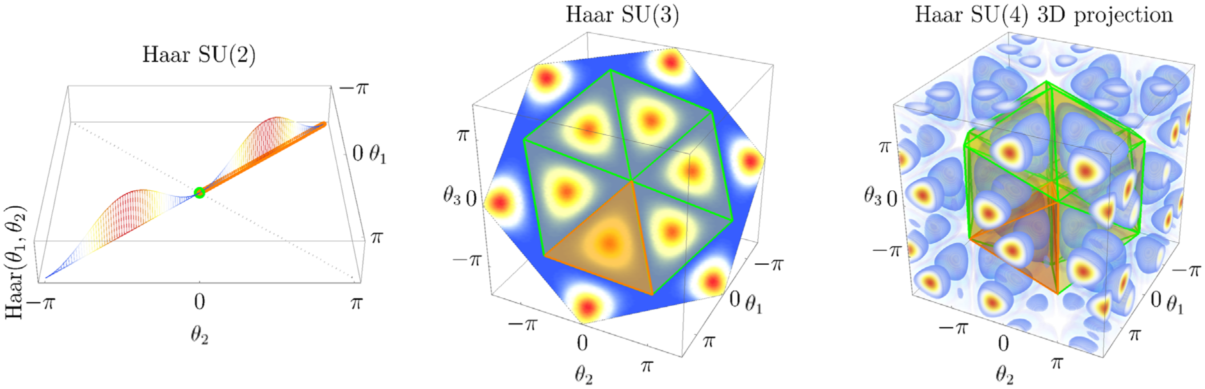

FIG. 4. Illustration of the eigenvalue spaces and respective Haar measures in the angular coordinate system $\theta_k = \arg(\lambda_k)$ for SU(2) [left], SU(3) [middle], and SU(4) [right]. Equation (19) describes how the Haar measure is included in these plots over the space of eigenvalues. The constraint $\det U = 1$ restricts the space of eigenvalues to the surface of codimension 1 defined by $\sum_k \theta_k = 0 \pmod{2\pi}$ depicted in each space. On each surface, permutation of the axes corresponds to permutation among the $N!$ cells delineated by green boundaries. A canonical cell used to construct permutation equivariant coupling layers is highlighted in orange for each surface. For SU(4), we show the surface of eigenvalues projected to an orthonormal basis in the constraint surface. For clarity in the SU(3) and SU(4) figures, we extend the range of the axes rather than showing the parts of the eigenvalue surface that would wrap around the periodic boundaries.

matrix using the new eigenvalues. Figure 2 depicts how this *spectral flow* is applied in the context of a gauge equivariant coupling layer.

Permutation equivariance is required to ensure that the kernel acts only based on the spectrum, not the particular order of eigenvalues produced during diagonalization. Normalizing flows that are permutation equivariant have previously been investigated in the machine-learning community to learn densities over sets (such as point clouds, objects in a 3D scene, particles in molecular dynamics, and particle tracks in collider events) [42–52]. Such approaches are directly applicable to kernels for U(N) variables (see Appendix E), because the eigenvalues can be transformed independently. For an SU(N) variable, however, the constraint $\det U = 1$ must additionally be satisfied, which prevents these methods from being straightforwardly applied. As an example, Fig. 4 depicts the space of eigenvalues of SU(2), SU(3), and SU(4) variables and illustrates the constrained surface of possible eigenvalues as well as the cells on this surface that are related by permutations in each case. To be equivariant, a spectral flow for SU(N) must transform values within each cell identically.

In this section, we describe special-case constructions of permutation equivariant transformations on the eigenvalues of an SU(2) or SU(3) variable, then generalize the approach to SU(N). In each case, we demonstrate the expressivity of these transformations by constructing flow-based models in terms of these transformations and training the models to learn several target families of densities that are invariant under matrix conjugation.

A. Target densities

As target distributions to test this approach, we define densities on SU(N) matrices that are invariant under matrix conjugation. For an SU(N) variable in the fundamental matrix representation, such a class of probability densities can be defined in terms of traces of powers of the variable,

$$p^{(i)}_{\mathrm{toy}}(U) \coloneqq e^{-S_i(U)}/Z_i, \qquad Z_i = \int dU\, e^{-S_i(U)}, \tag{17}$$

where

$$S_i(U) \coloneqq -\frac{\beta}{N}\mathrm{Re}\,\mathrm{tr}\left[\sum_n c^{(i)}_n U^n\right] \tag{18}$$

and $\int dU$ is integration with respect to the Haar measure of the group. Any distribution in this family is manifestly invariant under matrix conjugation and is therefore a function of the spectrum only. The coefficients $c^{(i)}$ determine the shape of the density on the group manifold, while β determines the scale of the density.

The coefficients $c^{(i)}$ defining the target densities for this study are reported in Table I. The first set of coefficients, $c^{(0)}$, was chosen to exactly match the marginal distribution on each open plaquette in the case of two-dimensional lattice gauge theory. To further investigate densities with similar structure, two additional sets of coefficients were chosen by randomly drawing values for $c^{(i)}_1$, $c^{(i)}_2$, and $c^{(i)}_3$

TABLE I. Sets of coefficients $c^{(i)}_n$ used to investigate the SU(2) and SU(3) matrix conjugation equivariant flow.

Set i	$c^{(i)}_1$	$c^{(i)}_2$	$c^{(i)}_3$
0	1	0	0
1	0.17	−0.65	1.22
2	0.98	−0.63	−0.21

and restricting to coefficients that produce a single peak in the density across all values of β. Performance on this set of coefficients is therefore representative of the ability of these flows to learn the local densities relevant to sampling for two-dimensional lattice gauge theory.

To investigate the expressivity of the permutation equivariant transformations that we define, we construct flow-based models that combine a uniform prior density with one kernel defined using the equivariant transformations under study. This combination of an invariant prior distribution with application of an equivariant kernel imposes matrix conjugation symmetry on each flow-based model exactly. As a metric for the expressivity of the permutation equivariant transformations used in each kernel, we checked the ability of the flow-based models to reproduce the target densities. Measurements of the ESS and plots of the densities are used to investigate model quality.

When plotting densities in the space of eigenvalues, as in Fig. 4 above and the density plots below, we always plot with respect to the Lebesgue measure on the eigenvalues. This is a natural choice, as densities with respect to this measure are what one expects to reproduce using histograms in the space of eigenvalues. However, the full model on SU(N) reports densities with respect to the Haar measure. When restricting to the space of eigenvalues, the resulting measure is absolutely continuous with respect to the Lebesgue measure with density given by the volume in SU(N) of conjugacy classes. This volume is given by [53]

$$\mathrm{Haar}(\lambda_1, \ldots, \lambda_N) = \prod_{i<j} |\lambda_i - \lambda_j|^2. \tag{19}$$

See also the Weyl integration formula and the case of SU(3) in [54].

B. Flows on SU(2)

The eigenvalues of an SU(2) matrix can generically be written in terms of a single angular coordinate as $\lambda_1 = e^{i\theta}$ and $\lambda_2 = e^{-i\theta}$. The permutation group S_2 on these eigenvalues is generated by the exchange $\lambda_1 \leftrightarrow \lambda_2$, which corresponds to $\theta \to -\theta$. We can therefore define a flow on θ which is equivariant under this transformation by separately handling the case of $\theta \in [-\pi, 0]$ and $\theta \in [0, \pi]$.

(1) If θ is in the first interval, negate it (otherwise, do nothing).

(2) Take the result and apply any invertible flow suitable for a variable in the finite interval $[0, \pi]$; e.g., a spline flow with fixed endpoints could be applied [55].

(3) If θ was negated in the first step, negate the result (otherwise, do nothing).

In effect, this extends the action of a flow on one *canonical cell*, $\theta \in [0, \pi]$, to the entire domain in a permutation equivariant fashion. The canonical cell for SU(2) is schematically depicted in the left panel of Fig. 4. This intuition is useful to extend the method to SU(3) and generic SU(N) variables in the following subsections.

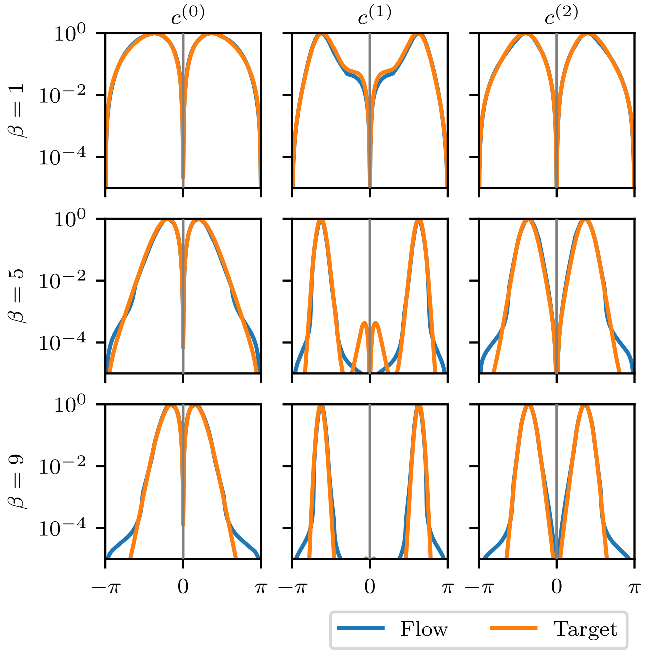

FIG. 5. Densities on the angular coordinate θ describing the eigenvalues of an SU(2) variable. The mirror symmetry across $\theta = 0$ corresponds to invariance of the distribution with respect to permutation of the eigenvalues; this symmetry is exactly enforced in the flow-based distribution using a permutation equivariant coupling layer.

To investigate the efficacy of this permutation equivariant spectral flow, we constructed SU(2) flow-based models to sample from each of the families of distributions defined by Eq. (17), with coefficients listed in Table I, for each $\beta \in \{1, 5, 9\}$. All models were constructed with a uniform prior distribution [with respect to the Haar measure of SU(2)] and a single matrix conjugation equivariant coupling layer, defined using the permutation equivariant spectral flow above. The transformation on the canonical cell $[0, \pi]$ was performed with a spline flow defined using four knots. Each model was trained using the Adam optimizer [56] with gradients of the loss function in Eq. (4) stochastically evaluated on batches of 1024 samples per step. Appendix C describes how gradients can be backpropagated through matrix diagonalization during optimization.

The densities learned by the flow-based model are compared against the target densities in Fig. 5. The peaks of the distribution are very precisely reproduced by the flow-based model, and the exact symmetry between the two cells (left and right halves of each plot) is apparent for both the model and target densities. Minor deviations between the model and target densities appear in the tails of the distribution, below roughly a density of 10^{-4}. These are rarely sampled regions; thus, these deviations only have

TABLE II. Final values of the ESS for each model trained for distributions on an SU(2) variable.

	$c^{(0)}$			$c^{(1)}$			$c^{(2)}$		
β	1	5	9	1	5	9	1	5	9
ESS (%)	100	100	100	98	98	97	100	99	100

a minor impact on model quality: all models reached an ESS above 97% for all sets of coefficients, as shown in Table II.

C. Flows on SU(3)

The eigenvalues of an SU(3) matrix can generically be written in terms of two angular variables as $\lambda_1 = e^{i\theta_1}$, $\lambda_2 = e^{i\theta_2}$, and $\lambda_3 = e^{-i\theta_1 - i\theta_2}$. There are six cells related by the permutation group S_3 on these three eigenvalues, as depicted in the middle panel of Fig. 4. We can define a permutation equivariant flow on these angular variables by extending a flow on a canonical cell to the whole space, as was done for SU(2) in the previous section.

(1) Enumerate all possible permutations of $[\theta_1, \theta_2, \theta_3]$, where $\theta_3 \coloneqq \mathrm{wrap}(-\theta_1 - \theta_2)$ is the phase of λ_3 in the interval $[-\pi, \pi]$.

(2) Choose the order $[\theta_{1'}, \theta_{2'}, \theta_{3'}]$ satisfying the canonical condition, $\mathrm{iscanon}(\theta_{1'}, \theta_{2'}, \theta_{3'})$. This makes $(\theta_{1'}, \theta_{2'})$ fall in the shaded region in Fig. 6. Record the permutation required to move from the original order to the canonical order.

(3) Since the shaded domain in Fig. 6 is split in two, replace $\theta_{1'}$ with $(\theta_{1'} - 2\pi)$ if $\theta_{1'} > 0$ to maintain a connected domain. Apply any invertible flow suitable for the canonical triangular domain of $\theta_{1'}$ and $\theta_{2'}$; our implementation is discussed below.

(4) Reconstruct the final angular variable $\theta'_{3'} = \mathrm{wrap}(-\theta'_{1'} - \theta'_{2'})$, then apply the inverse of the permutation in step 2 to produce the final eigenvalue phases $[\theta'_1, \theta'_2, \theta'_3]$.

For SU(3), we can define the canonical condition on eigenvalue phases in an *ad hoc* fashion,

$$\mathrm{iscanon}(\theta_1, \theta_2, \theta_3) = \begin{cases} \theta_3 \geq \theta_2 \geq \theta_1 & \sum_i \theta_i = 0 \\ \theta_1 \geq \theta_3 \geq \theta_2 & \sum_i \theta_i = 2\pi \\ \theta_2 \geq \theta_1 \geq \theta_3 & \sum_i \theta_i = -2\pi \end{cases}. \tag{20}$$

Intuitively, this function defines a canonical ordering of the eigenvalues while smoothly accounting for the fact that they are circular variables. This intuition is made more precise in the generalization of this approach to SU(N) variables in the following subsection. The *ad hoc* shift used to move the cell to a contiguous region is also addressed when generalizing.

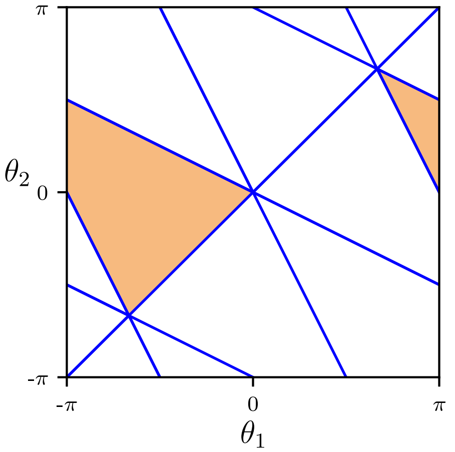

FIG. 6. The cell decomposition of the maximal torus of SU(3) viewed in the (θ_1, θ_2) coordinate system. The orange shaded cell is our choice of canonical cell.

Mapping to and from canonical cells is one particular construction for permutation equivariant flows. Appendix D details an alternate method based on averaging over all permutations for SU(3). In that approach, equivariance is also guaranteed, but the cost scales as $N!$ making it unsuitable for large N.

We investigated the efficacy of this permutation equivariant spectral flow by constructing SU(3) flow-based models to sample from the families of distributions defined by Eq. (17), with coefficients listed in Table I, for each $\beta \in \{1, 5, 9\}$. All models were constructed with a uniform prior distribution [with respect to the Haar measure of SU(3)] and a single matrix conjugation equivariant coupling layer, defined using the spectral flow above. The transformation on the triangular canonical cell was performed using two spline flows with four knots each, independently acting on the height and width coordinates. Each model was trained using the Adam optimizer with gradients of the loss function in Eq. (4) stochastically evaluated on batches of 1024 samples per step.

Figure 7 compares the distributions learned by the flow-based models to the target distributions when $\beta = 9$. The structure of the peaks of the distribution is reproduced accurately, and the exact sixfold symmetry between the cells is apparent in both the model and target densities. Minor deviations between the model and target densities appear in the tails of the distribution, below roughly a density of 10^{-3}. As with the SU(2) models, these deviations are in rarely sampled regions and therefore only have a minor impact on model quality. Quantitatively, our flow-based models achieved ESSs greater than 73% on all distributions, with the full set of final ESS values reported in Table III. The performance on this $SU(3)$ gauge group is observed to be marginally worse than $SU(2)$, likely due to the additional complexity of modeling density in the higher-dimensional space of eigenvalues. In practice, this

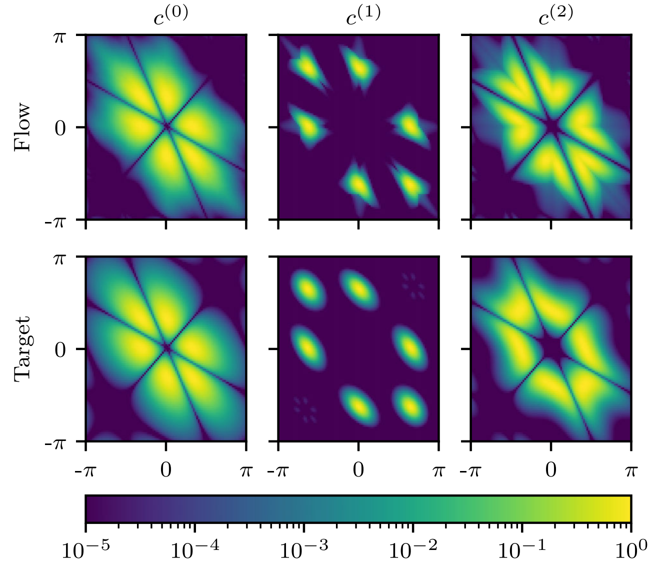

FIG. 7. Densities on the angular coordinates θ_1 and θ_2 defining the eigenvalues of an SU(3) variable. The densities learned by the flow-based models are compared to the target densities for three distributions, each with $\beta = 9$. The sixfold symmetry in each density is due to permutation invariance; this symmetry is exactly enforced in the flow-based distributions by using permutation equivariant coupling layers.

performance is sufficient to observe significant ESS when applying the approach to $SU(3)$ gauge theory. The leftmost distribution is the marginal distribution on plaquettes for two-dimensional SU(3) gauge theory; the high value of the ESS for this distribution indicates that this spectral flow is well suited to learn such distributions in the lattice gauge theory.

D. Flows on SU(N)

To apply the method to SU(N) variables for any N, we develop a general version of three of the steps used above which are as follows:

(1) Computing the vertices bounding a canonical cell
(2) Mapping eigenvalues into that canonical cell
(3) Applying spline transformations within that cell

We define cells in SU(N) as subsets of the maximal torus T, the subgroup of diagonal matrices of SU(N), as follows. An element of T is called regular if it has N distinct eigenvalues [54]. The set of regular matrices in T is an open set with N! connected components; the closure of each component is a *cell*.

TABLE III. Final values of the ESS for each model trained for distributions on an SU(3) variable.

	$c^{(0)}$			$c^{(1)}$			$c^{(2)}$		
β	1	5	9	1	5	9	1	5	9
ESS (%)	99	98	99	97	80	82	99	91	73

To construct a general spectral flow for SU(N), we first choose a particular cell, which we call the *canonical cell*. It is helpful to define the canonical cell in the Lie algebra $\mathfrak{t}$ of the maximal torus, rather than on the maximal torus directly. The Lie algebra $\mathfrak{t}$ is the $(N-1)$-hyperplane $\sum_{k=1}^{N}\theta_k = 0$ in $\mathbb{R}^N$ and is related to the maximal torus by the exponential map $\exp(\theta_1, \ldots, \theta_N) = \mathrm{Diag}(e^{2\pi i\theta_1}, \ldots, e^{2\pi i\theta_N})$. In this space, cells are $(N-1)$-simplexes enclosed by $(N-2)$-hyperplanes, each defined by a pair of eigenvalues becoming degenerate, i.e., $\theta_j = \theta_k \pmod{2\pi}$ for some j and k.

The N vertices of any of these simplexes are mapped by the exponential map to the N elements of the center of SU(N) (which are also elements of T). We define one such simplex Ψ by defining the bounding vertices $y_1, \ldots, y_N$ inside $\mathfrak{t}$,

$$[y_k]_j \coloneqq 2\pi\left(\frac{k}{N} - \delta_{k\geq j}\right), \tag{21}$$

where $\delta_{k\geq j} = 1$ when $k \geq j$, and is 0 otherwise. A proof that $\exp(\Psi)$ is a cell and a derivation of this formula is given in Appendix B. Thus, we choose $\exp(\Psi)$ as our canonical cell.

There are N! ways of reordering the eigenvalues of a regular point $x = \mathrm{Diag}(\lambda_1, \ldots, \lambda_N)$ in T, and exactly one of those falls in the canonical cell. It is intractable for large N to find the element that falls in the canonical cell by checking all permutations, as we did for SU(3). Instead, we explain in Algorithm 1 an approach to find the preimage in Ψ of this canonical element based on sorting.

Algorithm 1. Map into simplex Ψ

```
canon($\lambda_1, \ldots, \lambda_N$)
  1. Extract angles in range $[0, 2\pi)$, $\theta_k = \arg(\lambda_k) \bmod 2\pi$.
  2. Set $S = \frac{1}{2\pi}\sum_k \theta_k$; it is an integer because $\det U = 1$.
  3. Sort the angles in ascending order $\theta^{\mathrm{sort}} = \mathrm{sort}(\theta)$.
  4. Snap the angles to the hyperplane $\mathfrak{t}$ by
     $\theta^{\mathrm{snap}} = (\theta_1^{\mathrm{sort}}, \ldots, \theta_{N-S+1}^{\mathrm{sort}} - 2\pi, \ldots, \theta_N^{\mathrm{sort}} - 2\pi)$.
  5. Set $\theta^{\mathrm{canon}} = \mathrm{sort}(\theta^{\mathrm{snap}})$.
  6. Return $\theta^{\mathrm{canon}}$ and the combined permutation that was used to
     sort in steps 3 and 5.
```

The output of Algorithm 1 is a point in Ψ (see Sec. B 3) and a permutation. To invert the map into the canonical cell after we apply a flow, we permute the flowed values θ'_k using the inverse of the returned permutation, then map them to the torus using the exponential map. Appendix B 3 proves that this algorithm maps into the correct simplex. We can then show that applying the algorithm to any point in some cell returns the same output permutation by checking that the permutation does not change along any connected path within the cell. No two eigenvalues become degenerate along such a path; therefore, the order of the eigenvalue phases only changes when some θ_k crosses the boundary between 0 and 2π. For example, when

θ_k crosses from 0 to 2 π, it will become the largest angle (instead of the smallest) and S increments by 1; thus, the value of θ_k after snapping changes smoothly due to the additional 2π subtracted in step 4, all other angles are unaffected, and the final permutation is unchanged. A similar argument can be made when angles cross from 2π to 0.

Finally, we describe the implementation of a flow on Ψ, which concretely defines the "flow in canonical cell" step of Fig. 2. To be invertible, the flow must preserve the boundaries of Ψ. We implement such a flow by first using a coordinate transformation to map Ψ to an open box $\Omega = (0,1)^{N-1}$. On this box, an arbitrary boundary-preserving flow $\chi: \Omega \to \Omega$ can easily be applied (e.g., by using transformations suitable for a finite interval along each axis). Finally, the coordinate transformation can be undone to map back to Ψ. It is helpful to further introduce an intermediate $(N-1)$-simplex Δ, which is a right-angled simplex with equal leg lengths. Its vertices are $\{\kappa_1, \ldots, \kappa_N\}$, where κ_1 is the origin and $[\kappa_k]_j = \delta_{(k-1)j}\ \forall\ k \in \{2, \ldots, N\}$. The map $\phi: \Omega \to \Delta$ maps the box Ω to the simplex Δ by collapsing one end of the box in each direction,

$$\phi_i(\alpha) = \begin{cases} \alpha_1 & i = 1 \\ \alpha_i \prod_{j=1}^{j<i} (1 - \alpha_j) & i > 1, \end{cases} \tag{22}$$

where $\alpha \in \Omega$. The map $\zeta: \Delta \to \Psi$ then sends the intermediate right-angled simplex to the canonical simplex by

$$\zeta(\rho) = y_1 + \rho M, \tag{23}$$

where $\rho \in \Delta$ and M is the $(N-1) \times N$ matrix defined by $M_{ij} = [y_{i+1}]_j - [y_1]_j$. Both maps are invertible. The inverse map $\phi^{-1}: \Delta \to \Omega$ is given by

$$\phi_i^{-1}(\rho) = \frac{\rho}{1 - \sum_{j=1}^{i-1} \rho_j}, \tag{24}$$

for $\rho \in \Delta$, while $\zeta^{-1}: \Psi \to \Delta$ is given by

$$\zeta^{-1}(x) = (x - y_1) M^T (M M^T)^{-1}. \tag{25}$$

The entire chain of coordinate transformations, flow, and inverse coordinate transformations is depicted in Fig. 8.

The Jacobian of the entire flow can be computed by composing the Jacobian factors from each transformation in the chain. While the Jacobian factors acquired from the coordinate transformations are fixed, the flow acting on Ω is parametrized by, and the resulting density depends on, the action of this inner flow. For example, the inner flow could be a spline flow [55] constructed to transform each coordinate of Ω as a function of the model parameters and possibly the other coordinates of Ω. It is this inner flow that must be trained in each coupling layer to reproduce the final density on SU(N). A complete listing of the algorithm to apply the matrix conjugation equivariant kernel defined by the above spectral flow is given in Appendix B.

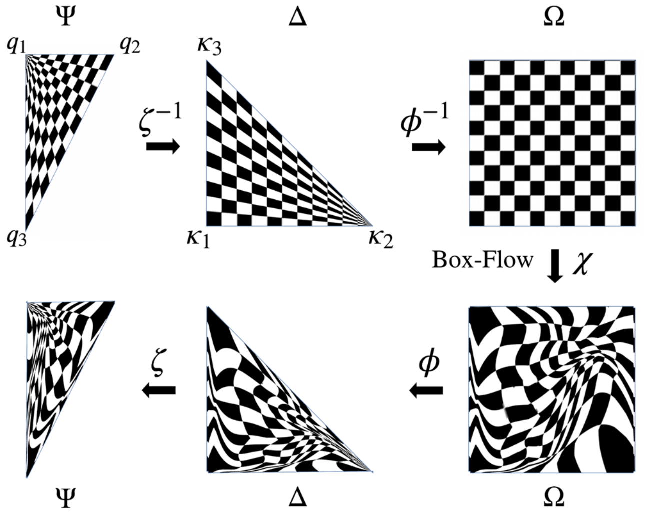

FIG. 8. Illustration of the steps we use to apply a flow to an $(N-1)$-simplex, shown for $N = 3$ as an example. Starting from an initial density on the simplex Ψ, we map it to an axis-aligned simplex Δ then to an open box Ω. We apply a parametric boundary preserving flow χ to the box and finally invert the chain back to the original coordinate system.

We implemented this general approach to matrix conjugation equivariant flows on SU(N) variables for a range of N. For $N \leq 9$, we trained these flows to reproduce target densities defined by Eq. (17), with coefficients listed in Table I, and $\beta = 9$. An ESS of greater than 5% was achieved on all target densities, with $c^{(0)}$ performing significantly better with greater than 90% ESS across all densities. Figure 9 compares the flow-based densities to the target densities for $N = 9$. Worse performance on $c^{(1)}$ and $c^{(2)}$ is reflective of their multimodal nature for some values of β/N. To investigate performance at large N, we trained flows to reproduce the $c^{(0)}$ density for $10 \leq N \leq 100$ and found ESSs greater than 90% for all models. All model distributions were confirmed to have exact permutation invariance.

IV. APPLICATION TO SU(2) AND SU(3) LATTICE GAUGE THEORY IN 2D

With an invertible kernel that is equivariant under matrix conjugation, the methods presented in Ref. [11] immediately allow construction of gauge equivariant coupling layers for SU(N) lattice gauge theory. To study the efficacy of such coupling layers for this application, we trained flow-based models to sample from distributions relevant for $1+1$-dimensional gauge theory. Specifically, we considered the distribution defined by the imaginary-time path integral in Eq. (1) with the action given by the Wilson discretization of the continuum gauge action,

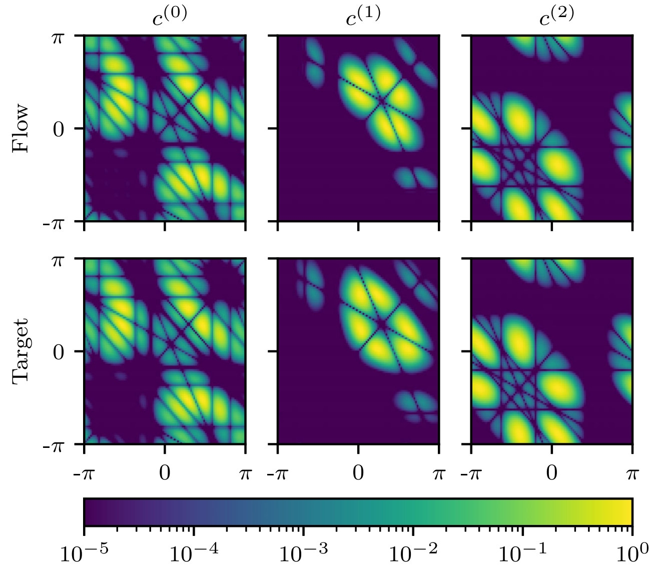

FIG. 9. Densities on a two-dimensional slice through the space of SU(9) eigenvalues defined by varying θ_1 and θ_2, keeping $\theta_3, \ldots, \theta_8$ fixed to random values, and assigning $\theta_9 = \mathrm{wrap}(-\sum_{k=1}^{8} \theta_k)$. The densities learned by the flow-based models are compared to the target densities for three distributions, each with $\beta = 9$. Horizontal, vertical, and diagonal lines of zero density correspond to locations where the chosen slice crosses through walls of the cells (on which the Haar measure forces the density to zero). Due to exact permutation invariance of the flow-based distribution, these lines are exactly reproduced.

$$S(U) \coloneqq -\frac{\beta}{N} \sum_x \mathrm{Re}\,\mathrm{tr}[P_{01}(x)]. \tag{26}$$

We investigated SU(3) gauge theory at a range of values of inverse coupling β and SU(2) gauge theory matched to approximately equivalent ’t Hooft couplings $\lambda = 2N^2/\beta$ on 16×16 periodic lattices, as listed in Table IV.

In the following subsections, we describe the architecture and training of our flow-based models, confirm the exactness of results using our sampler, and demonstrate that all symmetries are either exactly built into the model or are approximately learned by the model.

A. Model architecture and training

In all cases, we constructed flow-based models using a prior distribution $r(V)$ that is uniform with respect to the Haar measure of the link variables, for which configurations in the matrix representation are easily sampled.[9] The invertible function f acting on samples from the prior was composed of 48 coupling layers $g_1, \ldots, g_{48}$. We constructed each coupling layer using the gauge equivariant architecture presented in Sec. II C. Coupling layers specifically acted on plaquettes as the choice of open loops, transforming rows or columns of plaquettes spaced four sites apart on the lattice in each coupling layer, as denoted by $P_{\mu\nu}(x)$ (yellow) in Fig. 3; plaquettes that were unaffected by the transformation were used as the gauge invariant inputs to the inner spectral flow, as denoted by I_1 and I_2 (green) in Fig. 3. Coupling layers used an alternating sequence of rotations and a sweep over all possible translations of the transformation pattern to ensure that every link was updated after every eight layers.

TABLE IV. Choices of parameters on which we investigated the performance of our flow-based sampler. We selected three values of β for both SU(2) and SU(3) gauge theory, corresponding to approximately equivalent ’t Hooft couplings λ. $n_{\mathrm{d.o.f}} = DL^2(N^2 - 1)$ indicates the dimensionality of the gauge configuration manifold in each case.

SU(N)	L	β	$\lambda = 2N^2/\beta$	n_{dof}
SU(2)	16	$\{1.8, 2.2, 2.7\}$	$\{4.4, 3.6, 3.0\}$	1536
SU(3)	16	$\{4.0, 5.0, 6.0\}$	$\{4.5, 3.6, 3.0\}$	4096

The updating pattern that we define here is just one of the many possible choices. One could vary the open loops that are updated, change how the links are updated as a function of the open loops, choose a different pattern of translations and rotations between coupling layers, or alter which closed loops are passed as gauge invariant inputs to context functions. The choices made here were sufficient to learn distributions in two-dimensional gauge theory, but in generalizing beyond this proof-of-principle study, in particular to higher spacetime dimensions, these choices must be studied more carefully.

For SU(2) gauge theory, we implemented the spectral flow itself in a permutation equivariant fashion as described in Sec. III B. The flow acting on the interval $\theta \in [0, \pi]$ was a spline flow consisting of four knots, with the positions of the knots in $[0, \pi]$ computed as a function of the gauge invariant neighboring plaquettes $I_1, I_2, \ldots$ using convolutional neural networks with 32 channels in each of the two hidden layers. Throughout we used circular padding to support periodic boundaries, never used pooling operations, and used leaky ReLU activation functions between each convolution [58]. The model parameters defining the variational ansatz distribution consisted of the weights in these convolutional neural networks across all coupling layers.

For SU(3) gauge theory, we implemented the spectral flow as described in Sec. III C. The inner flow acted on eigenvalues in the canonical triangular cell by changing coordinates to an open box and applying a spline flow in that space, as discussed in Sec. III D. The spline flow acted on the open box in two steps, transforming the horizontal coordinate first, then the vertical coordinate conditioned on the new horizontal coordinate. The 16 knots of the splines were computed as a function of the gauge invariant neighboring plaquettes, and in the second step as a function

[9] To sample the prior distribution, the method presented in Ref. [57] can be used for U(N) and can also be modified to fix the determinant to 1 for SU(N).

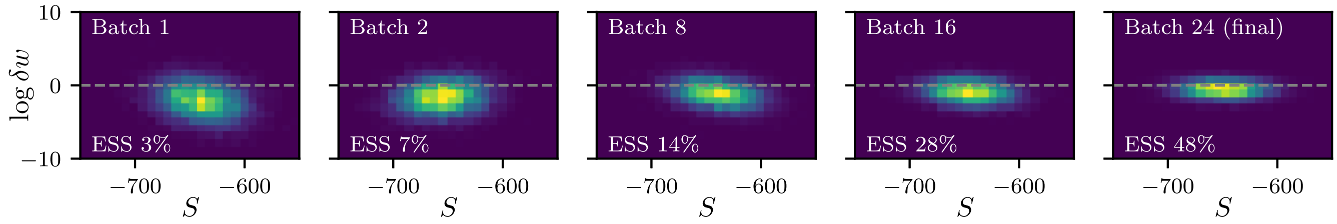

FIG. 10. Normalized reweighting factors $\log \delta w = -S(U) + S_{\mathrm{eff}}(U) - \log Z$ vs action S per configuration across 10 000 model proposals for $SU(3)$ gauge theory with $\beta = 6.0$. Reweighting factors are plotted at various points throughout training, reported in terms of the number of batches of size 3072 that have been used at that point in training.

of the horizontal coordinate as well. These two functions in each coupling layer were implemented using convolutional neural networks with 32 channels in each of the two hidden layers, with similar use of circular padding, no pooling operations, and leaky ReLU activation functions. The model parameters defining the variational ansatz distribution thus consisted of the weights in this pair of convolutional neural networks across all coupling layers.

In both cases, the model parameters were optimized using the Adam optimizer with the default hyperparameters used in the Pytorch library [59]. The learn rate was set to 10^{-3} early in training and reduced to 10^{-4} partway through training. Each optimization step consisted of sampling a batch of size 3072, estimating the modified KL divergence in Eq. (4), then using the optimizer to update the parameters. Between 14 000 and 29 000 total batches were used for training each model, divided into two stages detailed in the discussion of volume transfer below. During training, we monitored the ESS on each batch to assess model quality. Figure 10 shows how ESS and the spread of reweighting factors evolve over the course of training on a representative model. The final values of ESS for each model are reported in Table V. For the fixed architecture used, we observe a decrease in ESS as β is increased, approaching the continuum limit. It is natural to instead scale the model size to counteract this effect if attempting to approach the continuum along a line of constant physics. We comment further in Sec. V.

For this proof-of-principle study, we did not perform an extensive search over training hyperparameters. When scaling the method, we expect careful tuning of these hyperparameters and the model architecture can improve the model quality and allow more efficient training. Automatic tuning of hyperparameters, in particular, have been shown to significantly reduce model training costs in other machine learning applications [60–62].

TABLE V. Final values of the ESS for each model trained for SU(2) and SU(3) gauge theory.

	SU(2)			SU(3)		
β	1.8	2.2	2.7	4.0	5.0	6.0
ESS (%)	91	80	56	88	75	48

In general, models defined in terms of convolutional neural networks acting on invariant quantities in a localized region capture the local correlation structure defining the distribution. This local structure is independent of volume as long as finite volume effects are not too large. Thus, models can largely be trained on much smaller volumes than the target volume, requiring few training steps at the final volume to correct for any finite volume effects learned by the model. In this study, fewer training steps were used at the final volume than in the earlier small-volume training, resulting in computational gains over training immediately at the large volume.

The two-dimensional gauge theories used to investigate this model consist entirely of ultralocal dynamics, with any finite volume corrections exponentially small in the number of lattice sites [63]. In our study, we were thus able to train nearly optimal models on much smaller volumes, which enabled significantly more efficient training. For example, Fig. 11 shows that transferring a model that has already

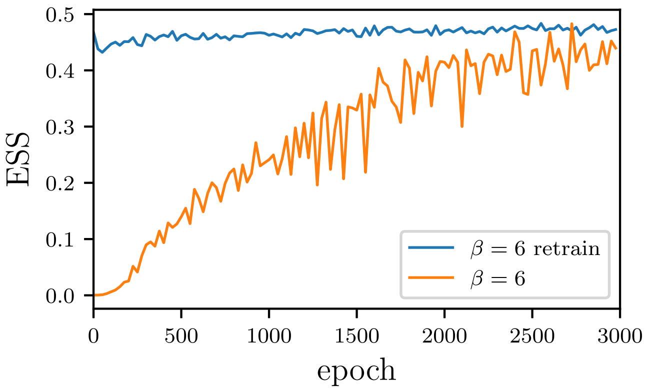

FIG. 11. Comparison of training dynamics for a model for SU(3) gauge theory on a 16 × 16 lattice, when initialized with weights from a model trained on an 8 × 8 geometry, versus the dynamics for an identical model trained from a random initialization. Results are shown for the $\beta = 6$ target in SU(3) gauge theory. The model transferred from the 8 × 8 geometry almost immediately converges to a plateau in model quality, while the model trained from a random initialization requires many training steps to converge to similar quality, despite adjusting the optimization hyperparameters to improve the rate of optimization from a random initialization.

learned to capture the local structure of the $\beta = 6$, SU(3) gauge theory on an 8×8 lattice almost immediately results in an optimized model for the target 16×16 lattice geometry, whereas it takes many training steps at the large volume to reach similar model quality when beginning training from a randomly initialized model. In any theory with a mass gap M, we expect that finite volume effects will be exponentially small in ML when the side length L of the lattice is large enough. Initially training at the smallest value of L in this regime thus provides an efficient approach to training models with larger L since the corrections that must be learned are exponentially small. These gains will be even more significant in higher spacetime dimensions, where the number of lattice sites scales with a larger power of the lattice side length L.

B. Observables

For each model, we constructed a flow-based Markov chain using independent proposals from the model with a Metropolis accept/reject step, as described in Sec. II A. Composing proposals into a Markov chain in this way ensures exactness in the limit of infinite statistics.

At finite statistics, it is possible that large correlations between samples at widely separated points in the Markov chain could result in apparent bias due to underestimated errors or insufficient thermalization time. We confirmed that this is not the case by comparing against a variety of analytically known observables. Specifically, we measure the expectation values of the following:

(1) Wilson loops W_{ab}, i.e., traced loops of links of shape $a \times b$

(2) Polyakov loops $\ell(x) = \mathrm{tr}\{\prod_t U_0(t,x)\}$, winding around the periodic boundary of the lattice

(3) Two-point functions of Polyakov loops, $\ell^*(x)\ell(y)$

These observables can also be computed analytically by a simple extension of analytical results found for $U(N)$ lattice gauge theory in two dimensions in Refs. [64,65] to the $SU(N)$ gauge group. The expectation value of any Polyakov loop is zero due to an exact center symmetry; this result was reproduced by the flow-based samples (as we discuss below, center symmetry is also exact in our models; therefore, this quantity is exact based on model proposals even before composition into a Markov chain). Due to confinement, Wilson loops have an expectation value exponentially small in the area of the loop; thus, we considered loops of simple shapes up to area 4 and the Polyakov loop two-point function with zero separation, $|\ell(x)|^2$. The flow-based estimates of these quantities for SU(2) and SU(3) gauge theory are shown graphically in Fig. 12. The results are statistically consistent with the analytical result.

We further checked that as statistics are increased, estimated errors fall as $1/\sqrt{n}$. This must be true asymptotically, but could be modified if there are correlations longer than the finite Markov chain length. We find that errors are indeed consistent with $1/\sqrt{n}$ scaling, as shown, e.g., in Fig. 13 for estimates of ReW_{11} for SU(3) gauge theory with $\beta = 6$.

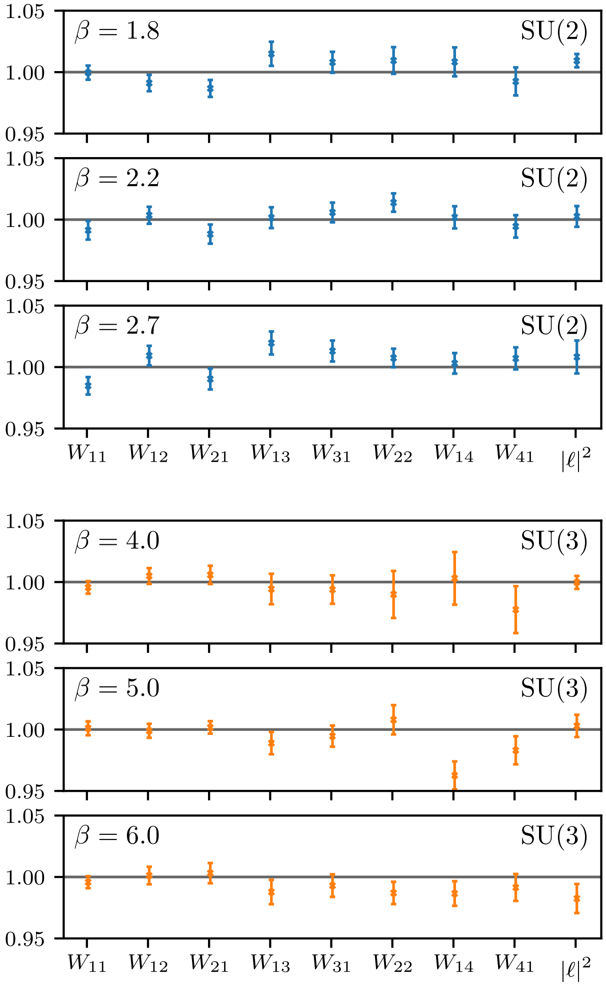

FIG. 12. Selection of observables, relative to the true values, computed using the flow-based SU(2) (top) and SU(3) (bottom) gauge theory ensembles. Observables were measured on configurations separated by a number of steps equal to the Markov chain autocorrelation time, as determined by the self-consistent estimator presented in Ref. [66]. The autocorrelation time ranged from 1 to 4 for all observables. Per observable, a total number of samples ranging from 20 to 15 000 was chosen to give percent-level errors.

C. Symmetries

After composition into a Markov chain, flow-based samples are guaranteed to asymptotically reproduce the exact distribution, including all symmetries. However, to reduce Markov chain correlations and improve training efficiency, we constructed our flow-based models to exactly reproduce some symmetries even *when generating proposals*. In terms of the factorization schematically shown in Fig. 1, exactly imposing symmetries in the model can

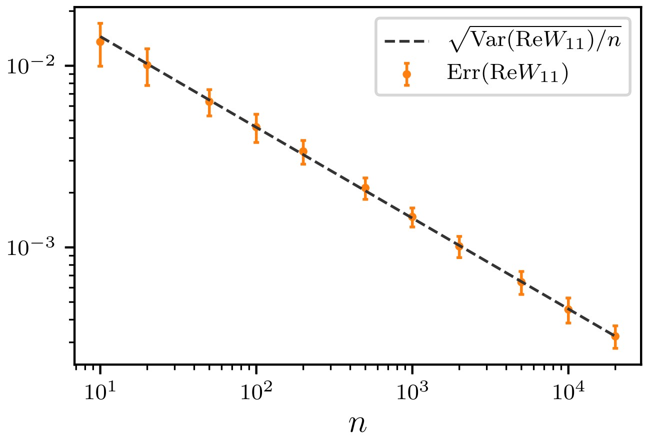

FIG. 13. Statistical errors on estimates of $\mathrm{Re}W_{11}$ in SU(3) gauge theory with $\beta = 6$ scale as expected as the number of independent samples n is varied. Errors (orange points) are estimated by a bootstrap procedure after thinning the data based on the measured autocorrelation time; the uncertainties on these estimates are measured using an outer bootstrap resampling step. The normalization $\mathrm{Var}(\mathrm{Re}W_{11})$ for the theoretical scaling (gray dashed line) is computed using the rightmost measured point.

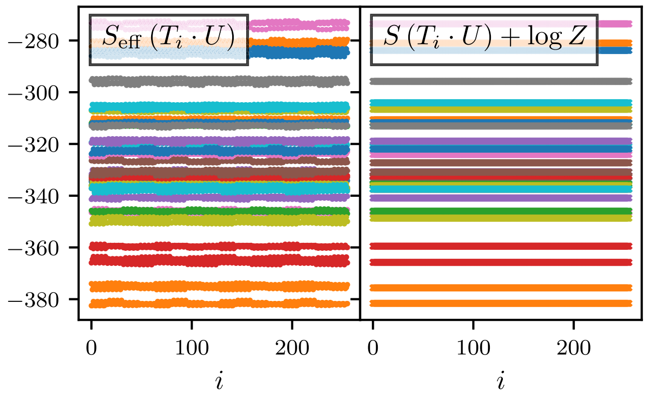

FIG. 15. Effective action S_{eff} vs normalized true action $S + \log Z$ on a sequence of translations of 32 gauge configuration samples for SU(3) gauge theory with $\beta = 6$. All 16×16 translations are plotted in a sequential pattern with index given by $i = \delta y + 16\delta x$.

reduce variance in reweighting factors along the pure-symmetry directions of the distribution.

As detailed in Sec. II B, we used coupling layers exactly equivariant to gauge symmetry and translational symmetry modulo a $\mathbb{Z}_4 \times \mathbb{Z}_4$ breaking arising from the size of the tiled pattern. To confirm the exact invariance of the flow-based distribution under gauge transformations, we measured the flow-based effective action over a range of gauge transformations on 32 random configurations along a randomly selected pure-gauge direction. Figure 14 depicts the invariance of both the effective and true actions under this random direction of gauge transformation. The data shown in different colors, corresponding to different random configurations, are approximately aligned in the left and right panels of Fig. 14, indicating that the true action is approximately matched by the effective action in the gauge invariant directions. We performed a similar investigation of translational invariance by scanning over all 16×16 possible translations of 32 random configurations. Figure 15 shows that there are fluctuations in the flow-based effective action, which arise from symmetry breaking within each 4×4 tile, but a large subgroup of the translational group is preserved as can be seen by the lines of constant effective action across various translations of each configuration. The spatial structure of the residual fluctuations in the effective action is shown in Fig. 16.

We also expect the hypercubic symmetry group to be an exact symmetry in most studies of lattice gauge theories. In the two-dimensional gauge theories under study, this group consists of the eight possible combinations of rotations and reflections of the lattice. While this symmetry could be imposed in every convolutional neural network used in all

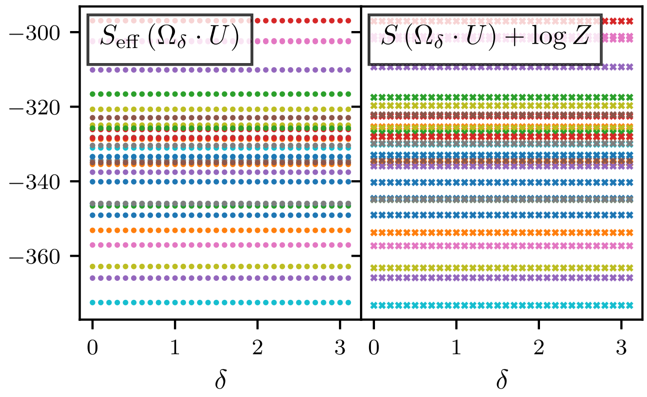

FIG. 14. Effective action S_{eff} vs normalized true action $S + \log Z$ on a sequence of gauge transformations of 32 gauge configuration samples for SU(3) gauge theory with $\beta = 6$. The gauge transformation applied is smoothly varied as δ is increased. Both the flow-based action and true action are exactly invariant to gauge transformations.

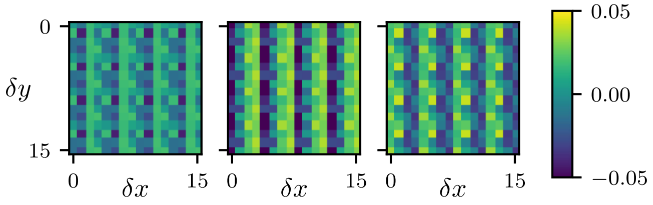

FIG. 16. Fluctuations in the flow-based effective action across all possible translations of three random gauge configurations. Configurations are drawn from the model for SU(3) gauge theory with $\beta = 6$. Fluctuations are reported relative to the mean effective action across all possible translations in each configuration and are normalized with respect to the standard deviation of the action in the path integral, $\sqrt{\langle (S - \langle S \rangle)^2 \rangle}$. The action is invariant for shifts $\delta x = 0 \pmod 4$, $\delta y = 0 \pmod 4$ demonstrating the exact translational symmetry modulo $\mathbb{Z}_4 \times \mathbb{Z}_4$.

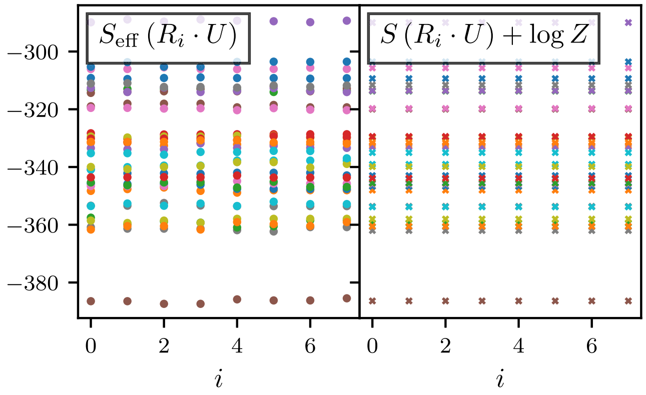

FIG. 17. Measured effective action S_{eff} vs normalized true action $S + \log Z$ on all eight possible hypercubic transformations of 32 gauge configurations sampled for SU(3) gauge theory with $\beta = 6$.

coupling layers, the pattern of open loops and the relation of these loops to updated links is difficult to make invariant; choosing a link to "absorb" the update of each open loop fundamentally breaks the hypercubic symmetry. On the other hand, this discrete symmetry group has few elements, consisting of only eight elements in two dimensions, 48 elements in three dimensions, and 384 elements in four dimensions. As such, we instead allowed our flow-based models to learn this small symmetry group over the course of training. Figure 17 depicts the approximate invariance of the flow-based effective action on 32 random gauge configurations under all eight elements of the hypercubic group in 2D.

For the pure-gauge theories under consideration, *center symmetry* and *complex conjugation symmetry* are additionally exact symmetries of the action; we included both symmetries explicitly in all of our models. Center symmetry is defined by the transformations

$$U_0(x) \to U_0(x) e^{i2\pi n/N}, \qquad n \in \{0, \ldots, N-1\}, \tag{27}$$

for all links on a fixed time slice, $x_0 = t$, with other links unaffected. Our coupling layers are already equivariant under this symmetry, which can be seen by considering the updated value of any modified link, $U'_\mu(x) = P'_{\mu\nu}(x) P^\dagger_{\mu\nu}(x) U_\mu(x)$: plaquettes do not transform under center symmetry, and by definition center transformations on the link $U_\mu(x)$ are free to commute all the way to the left. If open Polyakov loops were transformed in the coupling layers, or if traced Polyakov loops were used as part of the gauge invariant inputs to any transformation, this property would no longer hold; including terms like these will be necessary for theories in which center symmetry is explicitly broken. We also explicitly constructed our spectral flows to be equivariant under complex conjugation. For SU(2) matrices, this is equivalent to permutation of the eigenvalues and is therefore immediate. For SU(3) matrices, it corresponds to a nontrivial mirror symmetry within a single canonical cell. We implemented this mirror symmetry by extending a spline flow from one half of the canonical cell to the entire space using an approach similar to that applied for SU(2) permutation equivariance. Both center symmetry and complex conjugation symmetry were reproduced to within numerical precision.

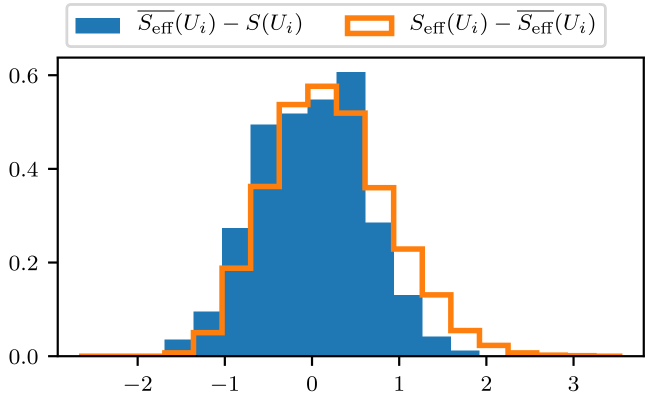

FIG. 18. Reweighting factors after post hoc symmetrization (filled blue) vs the log difference of the symmetrized effective action from the original effective action (outlined orange). The width of the latter distribution indicates the gains made by averaging over these fluctuations in Eq. (28). The width of the former distribution indicates the remaining errors in the model.

Finally, we considered explicitly symmetrizing model proposals under a discrete symmetry group H. Such an approach could be used, e.g., to impose the residual $\mathbb{Z}_4 \times \mathbb{Z}_4$ translational symmetry or hypercubic symmetry on the flow-based model *post hoc*. To do so, a random symmetry transformation is applied after drawing a model proposal and the averaged model weight,

$$\overline{S_{\mathrm{eff}}}(U) \coloneqq -\log\left(\frac{1}{|H|}\sum_{h\in H} e^{-S_{\mathrm{eff}}(h\cdot U)}\right), \tag{28}$$

is reported. This averaging over all possible symmetry transformations is required to faithfully report the probability density of the output sample for reweighting or composition into a flow-based Markov chain. It is also very costly if the symmetry group is large (and is intractable for continuous symmetry groups).

We studied the possibility of employing such averaging for the residual $\mathbb{Z}_4 \times \mathbb{Z}_4$ translational symmetry breaking. Figure 18 compares the reweighting factors required for the translationally symmetrized model vs the fluctuations that have been averaged over by the sum in Eq. (28). The comparable width of these histograms indicates that the improvement in the spread of reweighting factors (which controls the variance of estimators) is $O(1)$; evaluating the ESS directly, we found in this example that the ESS was increased by roughly a factor of 2. Thus, the additional factor of 16 in cost required to generate the symmetrized proposals outweighed the variance reduction benefits. We conclude that it is beneficial to impose symmetries when possible in the flow-based model itself, as we did with gauge symmetry, center symmetry, conjugation symmetry, and a large subgroup of translational symmetry, but in our application we found it counterproductive to impose a residual symmetry by brute force averaging of proposals.

V. OUTLOOK

It has recently been shown in proof-of-principle work that the challenging computational task of sampling field configurations for lattice gauge theory may be accelerated by orders of magnitude compared with more traditional sampling approaches through the use of flow-based models [11,16]. In other lattice field theories, it has been demonstrated that these models can also be used to estimate observables, such as the absolute value of the free energy, that are difficult to estimate with existing MCMC methods [17].

Here, we present a definitive step toward more efficient sampling for lattice gauge theories by developing flow-based models that are equivariant under SU(N) gauge symmetries, thus enabling the construction of model architectures that respect the symmetries of the Standard Model of particle and nuclear physics and other physical theories. We demonstrate the application of this approach to sampling both single SU(N) variables and SU(2) and SU(3) lattice gauge theory configurations, showing that observables computed using samples from flow-based models are correct within uncertainties and have the predicted statistical scaling with an increasing number of samples.

In the proof-of-principle implementation presented here, we have not attempted to optimize the model architecture and training approaches for expressivity or efficiency. State-of-the-art calculations will likely require further development in these directions. For one, alternative patterns of loops to update in each coupling layer could increase expressivity of the model, and we expect that exploring these choices will have significant impact in higher dimensions, where the degree of connectivity between links and loops is higher. Second, studies of whether the kernels and coupling layers that we constructed can generalize to multimodal distributions will help to understand the ability of these models to capture distributions in broken symmetry phases of lattice gauge theories. Third, investigation of hyperparameter tuning and further ways to exploit existing models for training and model initialization could allow more efficient training and improve model quality. Finally, studying the scaling of model complexity required to take the continuum limit will determine the viability of this approach on the fine-grained lattices employed in state-of-the-art lattice field theory calculations. Due to locality, keeping the variance of reweighting factors or the flow-based Markov chain rejection rate fixed while we increase the physical volume of the lattice will require improving the model's approximation of the local correlation structure of the theory.[10] However, it is not clear how the model complexity and number of parameters (and therefore the cost of model evaluation) must scale when physical volume is held fixed and the lattice spacing is decreased. This scaling depends on the dynamics of the theory and the architecture of the flow-based model under study, and it must be determined experimentally. If these challenges can be addressed, the extension of these proof-of-principle results to state-of-the-art lattice gauge theory calculations for complex theories such as quantum chromodynamics has the potential to redefine the computational limits, and hence the impact, of lattice gauge theory in the coming exascale computing era [67].

ACKNOWLEDGMENTS

The authors thank Jiunn-Wei Chen, Will Detmold, Andreas Kronfeld, George Papamakarios, Andrew Pochinsky, and Peter Wirnsberger for helpful discussions. The work of G. K., D. B., D. C. H., and P. E. S. was supported in part by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under grant Contract No. DE-SC0011090. The work of P. E. S. was additionally supported by the National Science Foundation under CAREER Grant No. 1841699 and under EArly-concept Grants for Exploratory Research 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. The work of K. C. was supported by the National Science Foundation under the Grants No. ACI-1450310, No. OAC-1836650, and No. OAC-1841471 and by the Moore-Sloan data science environment at NYU. The work of M. S. A. was supported by the Carl Feinberg Fellowship in Theoretical Physics. This work was 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. DE-AC02-06CH11357. Some figures were produced using MATPLOTLIB [68] and *Mathematica* [69]. This work is supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, [70])

APPENDIX A: PROOF THAT EQUIVARIANCE UNDER MATRIX CONJUGATION CAN BE REPRESENTED AS EQUIVARIANCE UNDER EIGENVALUE PERMUTATION

Let G be a compact connected Lie group, such as SU(N). We are interested in characterizing the group of

[10] Instead keeping the model architecture *fixed* while increasing the physical volume results in exponential degradation of the variance of reweighting factors or the flow-based Markov chain rejection rate.

diffeomorphisms of G that are equivariant under the action by matrix conjugation. Such a diffeomorphism $f: G \to G$ satisfies $f(XWX^{-1}) = Xf(W)X^{-1}$ for any $W, X \in G$. Our aim is to show that all such diffeomorphisms are extensions to G of diffeomorphisms of a maximal torus that are equivariant under the action of the Weyl group. We note that despite the use of neural networks to generate the parameters for the diffeomorphism $f(W)$ on the maximal torus, it is a diffeomorphism on W for any fixed set of parameters and this is the only required property for the following proof to proceed.

Let T be a maximal torus in G. Recall this torus is equal to its own centralizer $Z(T) = \{X \in G | XDX^{-1} = D,\ \forall\ D \in T\}$. The Weyl group of G is a finite group equal to $N(T)/T$, where $N(T) = \{X \in G | XDX^{-1} \in T,\ \forall\ D \in T\}$ is the normalizer of T.

In the case of $G = \mathrm{SU}(N)$ or $G = \mathrm{U}(N)$, a maximal torus is given by the subgroup of diagonal matrices, and the Weyl group is isomorphic to the group of permutations on N elements acting on T by permuting the elements on the diagonal. For a permutation σ, a representative matrix in $N(T)$ is given by a permutation matrix, with potentially some elements set to -1 instead of 1 in order for the determinant to be 1 in the case of $\mathrm{SU}(N)$.

We start with the easy direction, where we restrict a diffeomorphism from G to T.

Proposition 1.—Let $f: G \to G$ be a matrix conjugation equivariant diffeomorphism. Then f restricted to T is a diffeomorphism of T that is equivariant under the action of the Weyl group.

Proof.—First, let us show that $f(T) \subset T$. Let $D \in T$. For any $X \in T$, we have $XDX^{-1} = D$ since T is commutative. By equivariance of f, we also have $f(XDX^{-1}) = Xf(D)X^{-1}$. We deduce that $Xf(D)X^{-1} = f(D)$ for all $X \in T$, which means that $f(D)$ is in the centralizer of T. Since this is equal to T for a maximal torus, we have proved $f(D) \in T$.

Since f is a diffeomorphism, its restriction to T is also a diffeomorphism on its image. This image will be both closed and open in T, and is therefore the whole of T.

Finally, the fact that f restricted to T is equivariant under the action of the Weyl group is immediate, since this action comes from the action by conjugation from $N(T)$. ■

For the opposite direction, we restrict ourselves to the cases $G = \mathrm{SU}(N)$ and $G = \mathrm{U}(N)$. We choose T to be the subgroup of diagonal matrices. The Weyl group acts by permutation on the diagonal elements in T.

In what follows, we will assume $f: T \to T$ is a diffeomorphism that is equivariant under the action of the Weyl group. We first introduce a Lemma that will be used later.

Lemma 1.—Let $D \in T$. Assume $A \in G$ commutes with D, then A also commutes with $f(D)$.

Proof.—Let i, j be distinct indices in the range $1 \ldots N$. Assume that $D_{ii} = D_{jj}$. We will first prove that $f(D)_{ii} = f(D)_{jj}$. Let $P \in \mathrm{SU}(N)$ be given by $P_{ij} = 1, P_{ji} = -1$, $P_{ii} = P_{jj} = 0$, and $P_{kk} = 1$ for all $k \neq i$, j, then $PDP^{-1} = D$. Since $P \in N(T)$, we have $Pf(D)P^{-1} = f(PDP^{-1}) = f(D)$ and P commutes with $f(D)$. This means that $f(D)_{ii} = f(D)_{jj}$.

Let $\lambda_1, \ldots, \lambda_m$ be the m distinct eigenvalues of D, with respective multiplicity $n_1, \ldots, n_m$. There exists P in $N(T)$ such that

$$PDP^{-1} = \begin{pmatrix} \lambda_1 I_{n_1} & & 0 \\ & \cdot & \\ 0 & & \lambda_m I_{n_m} \end{pmatrix}, \tag{A1}$$

where I_{n_k} is an identity matrix of size n_k. This means that $f(PDP^{-1})$ must also be of the form

$$f(PDP^{-1}) = \begin{pmatrix} \mu_1 I_{n_1} & & 0 \\ & \cdot & \\ 0 & & \mu_m I_{n_m} \end{pmatrix}. \tag{A2}$$

Since A commutes with D, we have that PAP^{-1} commutes with PDP^{-1}. Since matrices that commute must preserve each others eigenspaces, this implies that PAP^{-1} must have the form

$$PAP^{-1} = \begin{pmatrix} U_1 & & 0 \\ & \cdot & \\ 0 & & U_n \end{pmatrix}. \tag{A3}$$

Given the form of $Pf(D)P^{-1} = f(PDP^{-1})$ shown above, we conclude that PAP^{-1} commutes with $Pf(D)P^{-1}$; therefore, A commutes with $f(D)$. ■

Finally, using Lemma 1, we can prove our main result.

Proposition 2.—Assume $W \in G$ has two different decompositions $W = XDX^{-1} = YEY^{-1}$, where D and E are diagonal matrices. Then

$$Xf(D)X^{-1} = Yf(E)Y^{-1}. \tag{A4}$$

Proof.—There exists P in $N(T)$ such that $E = PDP^{-1}$, which implies $f(E) = Pf(D)P^{-1}$. This means Eq. (A4) is equivalent to

$$Xf(D)X^{-1} = Zf(D)Z^{-1}, \tag{A5}$$

where $Z = YP$. The above equation is equivalent to saying that $X^{-1}Z$ commutes with $f(D)$, and by Lemma 1 this will be the case if $X^{-1}Z$ commutes with D. This is easy to prove,

$$\begin{aligned} X^{-1}ZDZ^{-1}X &= X^{-1}YPDP^{-1}Y^{-1}X \\ &= X^{-1}YEY^{-1}X \\ &= X^{-1}WX \\ &= D. \end{aligned} \tag{A6}$$

■

Example 1.—In the case of $G = \mathrm{SU}(2)$, the maximal torus is isomorphic to U(1), the Weyl group has size 2, and its only nontrivial element transforms $\left(\begin{smallmatrix}\lambda & 0\\ 0 & \bar{\lambda}\end{smallmatrix}\right)$ to $\left(\begin{smallmatrix}\bar{\lambda} & 0\\ 0 & \lambda\end{smallmatrix}\right)$; thus, any bijection $f : \mathrm{U}(1) \to \mathrm{U}(1)$ that satisfies $f(\bar{z}) = \overline{f(z)}$ defines an equivariant bijection of SU(2).

According to Proposition 2, any matrix conjugation equivariant function on T can be extended to an equivariant function on G. If the function was invertible on T, then it is easy to see that it will also be invertible on G.

APPENDIX B: DETAILS OF PERMUTATION EQUIVARIANCE OF SU(N) SPECTRAL FLOWS

1. Proof that Eq. (21) defines a cell

We demonstrate that the vertices from Eq. (21) define an $(N-1)$-simplex Ψ corresponding to a cell $\mathcal{C}$. In practice, this means showing that any point on the boundary of Ψ maps to a point in $\mathcal{C}$ with repeated eigenvalues, while any point in the interior of Ψ maps to a regular point, i.e., one without repeated eigenvalues.

Proposition 3.—The vertices $y_1, \ldots, y_N$ from Eq. (21) define an $(N-1)$-simplex Ψ that maps to a cell $\mathcal{C} = \exp(\Psi)$ in the maximal torus.

Proof.—Let θ be a point in Ψ, the convex hull of $y_1, \ldots, y_N$, given by

$$\theta_j = 2\pi \sum_k \gamma_k \left(\frac{k}{N} - \delta_{k \geq j} \right),$$

where $\gamma_k \geq 0$ and $\sum_k \gamma_k = 1$. The boundary $\partial\Psi$ is the simplicial complex formed by all points θ such that at least one γ_k is zero.

We consider the difference between two points θ_i and θ_j, for $j > i$,

$$\theta_j - \theta_i = 2\pi \sum_k \gamma_k (\delta_{k \geq i} - \delta_{k \geq j}) \tag{B1}$$

$$= 2\pi \sum_{k=i}^{j-1} \gamma_k. \tag{B2}$$

If $\gamma_k = 0$ for some k such that $1 \leq k < N$, then $\theta_{k+1} - \theta_k = 0$ and $\exp(\theta)$ has a repeated eigenvalue. If $\gamma_N = 0$, we have that $\theta_N - \theta_1 = 2\pi \sum_{k=1}^{N-1} \gamma_k$, but since $\gamma_N = 0$, we have $\sum_{k=1}^{N-1} \gamma_k = 1$. Thus, $\theta_N - \theta_1 = 2\pi$. This shows that a point in $\partial\Psi$ is exponentiated to a point with repeated eigenvalues.

Finally, we need to show that a point in the interior of Ψ is exponentiated to a regular point. Such a point corresponds to $\gamma_k > 0, \ \ \forall \ k$. As a consequence of Eq. (B2), no pair θ_i, θ_j is equivalent modulo 2π: for an interior point, the sum $\sum_{k=1}^{j-1} \gamma_k$ is strictly positive and also strictly smaller than 1. ■

2. More on Eq. (21)

In this section, we explain where the points y_k in Eq. (21) come from. In particular, we draw some parallel between our construction of the simplex Ψ and fundamental domains in Bravais lattices. Figure 19 depicts the steps described below.

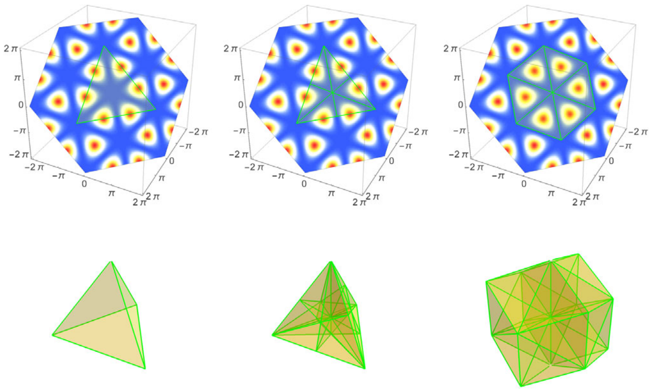

FIG. 19. Illustration of the steps to derive Eq. (21) for SU(3) (top row) and SU(4) (bottom row). Columns correspond, from left to right, to the steps: (i) starting simplex Δ on the hyperplane $\sum_j \theta_j = 0$; (ii) barycentric subdivison, and (iii) edge-length adjustment to cover the cell. For SU(3), we overlay the simplexes on top of the Haar measure for reference.

Recall that we defined a cell in Sec. III D as the closure of a connected component in T of the set of regular points. These cells are separated by regions where the eigenvalues λ_i are degenerated. That is, for every point $(\lambda_1, \ldots, \lambda_N)$ in the boundary of a cell, there exists at least one pair $i, j \in \{1, \ldots, N\}$ such that $\theta_i - \theta_j = 0 \bmod 2\pi$, where $\theta_i = \arg \lambda_i$. The inverse image of these boundaries under the exponential map are affine hyperplanes in $\mathfrak{t}$ that bound simplexes. The set of vertices of these simplexes (and all their translated copies) form a Bravais lattice Λ given by $\sum_i z_i u_i$,where $z \in \mathbb{Z}^{N-1}$ and the primitive vectors $u_i \in \mathbb{R}^N$ are defined by $[u_i]_k = 2\pi(\frac{1}{N} - \delta_{(i+1)k})$. In the theory of lattices, the canonical cell is known as the fundamental simplex and its orbit under the Weyl group is known as the root polytope [71].

Recall that we identified $\mathfrak{t}$ with the hyperplane $\sum_i \theta_i = 0$ in $\mathbb{R}^N$. Assume Δ is an $(N-1)$-simplex with N vertices $q_i \in \mathfrak{t}$, defined by the components

$$[q_i]_j = 2\pi \left(\frac{1}{N} - \delta_{ij} \right),$$

where $[q_i]_j$ indicates the jth component of the ith vertex.

Based on the observation that Weyl chambers in the Lie algebra of SU(N) are open cones determined by the barycentric subdivision of Δ [72], we then apply a barycentric transformation to the vertices q_i. Generally, a barycentric transformation is achieved by sending $\{v_k\}$ to $\{w_k\}$ by

$$w_k = \frac{1}{k} \sum_{i=1}^{k} v_i. \tag{B3}$$

Applying this to Δ gives a new simplex $\tilde{\Delta}$ with vertices x_k,

$$[x_k]_j = \frac{2\pi}{k}\sum_{i=1}^{k}\left(\frac{1}{N}-\delta_{ij}\right) = \frac{2\pi}{k}\left(\frac{k}{N}-\delta_{k\geq j}\right),$$

where $\delta_{k\geq j}$ is equal to one when $k \geq j$ and zero otherwise.

The simplex $\tilde{\Delta}$ is correctly aligned with a cell; however, it does not contain the full cell. To fix this, we adjust the length of the edges of $\tilde{\Delta}$, resulting in the final simplex $\mathcal{C}$. Let $\alpha_k \in \mathbb{R}^+$ be an arbitrary scaling of the kth edge of $\tilde{\Delta}$, so that the rescaled coordinates of the vertices are given by

$$[y_k]_j = \alpha_k \frac{2\pi}{k}\left(\frac{k}{N}-\delta_{k\geq j}\right).$$

We want the vertices $\{y_1, \ldots, y_N\}$ of $\mathcal{C}$ to correspond to neighboring points of the Bravais lattice Λ, so that the orbit of the simplex $\mathcal{C}$ with respect to the Weyl group tiles the torus without leaving holes and without overlaps. More precisely, this constraint is equivalent to imposing $[y_k]_j - [y_k]_i \in \{0, 2\pi\}, \ \forall \ j > i$. Substituting the expression for $[y_k]_j$, we obtain the constraint

$$[y_k]_j - [y_k]_i = \alpha_k \frac{2\pi}{k}(\delta_{k\geq i} - \delta_{k\geq j}) = 0 \mod 2\pi, \quad \forall \ j > i.$$

The term $\delta_{k\geq i} - \delta_{k\geq j}$ can have values 0 or 1, so that the constraint can be satisfied for all $j > i$ if $\alpha_k = k$. With this choice, the coordinates of the vertices of $\mathcal{C}$ are given by

$$[y_k]_j = 2\pi\left(\frac{k}{N}-\delta_{k\geq j}\right). \tag{B4}$$

3. Proof that Algorithm 1 projects into Ψ

In this section, we will show the output of Algorithm 1 is always a point in Ψ.

The output of Algorithm 1 is a set of angles θ^{canon}. In this section, we will write θ^c as an abbreviation for θ^{canon}. We wish to prove that θ^c is in the convex hull of the y_k defined in Eq. (21). We will do so by explicitly exhibiting the weights of the convex combination. In essence, our proof is the opposite of what lead to Eq. (B2).

Define

$$\gamma_k = \begin{cases} \frac{1}{2\pi}(\theta^c_{k+1} - \theta^c_k) & k < N, \\ 1 - \sum_{j=1}^{N-1}\gamma_j & k = N. \end{cases} \tag{B5}$$

The sum $\sum_{j=1}^{N-1}\gamma_j$ simplifies to $\frac{1}{2\pi}(\theta^c_N - \theta^c_1)$. By construction, the difference $\theta^c_N - \theta^c_1$ cannot be more than 2π. It follows that $\gamma_k \geq 0, \ \forall \ k$ and $\sum_k \gamma_k = 1$.

Let $\theta' = \sum_k \gamma_k y_k$ be in Ψ. We will now prove that $\theta' = \theta^c$, which will conclude the proof. Using the definition of y_k in Eq. (21), it follows that

$$\begin{aligned} \theta'_j &= 2\pi\sum_k \gamma_k\left(\frac{k}{N}-\delta_{k\geq j}\right) \\ &= 2\pi\left(\sum_{k=1}^{N-1}\frac{1}{2\pi}(\theta^c_{k+1}-\theta^c_k)\left(\frac{k}{N}-\delta_{k\geq j}\right)\right) + 2\pi\gamma_N[y_N]_j. \end{aligned} \tag{B6}$$

The extra term after the initial sum above is 0 because $[y_N]_j = 0$. We continue

$$\begin{aligned} \theta'_j &= \sum_{k=1}^{N-1}(\theta^c_{k+1}-\theta^c_k)\left(\frac{k}{N}-\delta_{k\geq j}\right) \\ &= \sum_{k=2}^{N}\theta^c_k\left(\frac{k-1}{N}-\delta_{k-1\geq j}\right) - \sum_{k=1}^{N-1}\theta^c_k\left(\frac{k}{N}-\delta_{k\geq j}\right) \\ &= \theta^c_N\left(\frac{N-1}{N}-\delta_{N-1\geq j}\right) + \sum_{k=2}^{N-1}\theta^c_k\left(-\frac{1}{N}+\delta_{k\geq j}-\delta_{k\geq j+1}\right) \\ &\quad -\theta^c_1\left(\frac{1}{N}-\delta_{1\geq j}\right). \end{aligned} \tag{B7}$$

In the last line above, note that we can simplify $\frac{N-1}{N} - \delta_{N\geq 1+j}$ to $\frac{-1}{N} + \delta_{j,N}$, and $\delta_{k\geq j} - \delta_{k\geq j+1}$ to $\delta_{k,j}$, and also $\delta_{1\geq j}$ to $\delta_{1,j}$. It follows that

$$\theta'_j = \theta^c_N\delta_{N,j} + \sum_{k=2}^{N}\theta^c_k\delta_{k,j} + \theta^c_1\delta_{1,j} - \frac{1}{N}\sum_{k=1}^{N}\theta^c_k = \theta^c_j. \tag{B8}$$

The last line above was obtained using that the sum of θ^c_k is 0. This concludes our proof.

4. Full algorithm

Algorithm 2. Equivariant SU(N) coupling layer.

```
Given $U \in \mathrm{SU}(N)$
  1. $\lambda, \{\vec{v}_i\} = \mathrm{eigendecomp}(U)$.
  2. Project to canonical cell $\Psi$: $I = \mathrm{canonicalize}(\arg(\lambda))$.
  3. Map to axis-aligned simplex $\Delta$: $\beta = \zeta^{-1}(I)$.
  4. Map to box $\Omega$: $\alpha = \phi^{-1}(\beta)$.
  5. Apply box flow: $\alpha' = \chi(\alpha)$.
  6. $\beta' = \phi(\alpha')$.
  7. $I' = \zeta(\beta')$.
  8. $\lambda' = \mathrm{uncanonicalize}(I')$.
  9. $U' = \mathrm{eigenrecomp}(\lambda', \{\vec{v}_i\})$.
  10. Accumulate all log-det-Jacobians,

        $\mathrm{LDJ} = \log\mathrm{Haar}(\lambda') - \log\mathrm{Haar}(\lambda)$
        $\qquad + \mathrm{LDJ}_\chi + \mathrm{LDJ}_{\phi^{-1}} - \mathrm{LDJ}_\phi$

  11. $U'$ is equivariant to SU($N$) matrix conjugations and LDJ
      is invariant to matrix conjugations.
  12. Return $U'$ and LDJ.
```

Above, there are no terms in LDJ for the map ζ because the Jacobian factor acquired from the forward and backward maps are constants that cancel. The term Haar(λ) gives the density of the Haar measure with respect to the Lebesgue measure in the space of eigenvalues, as defined in Eq. (19). The normalization of this term is unimportant as it cancels in the above algorithm.

APPENDIX C: BACKPROPAGATION THROUGH UNITARY MATRIX DIAGONALIZATION

We define the backpropagation of gradients through application of a black-box (unitary) diagonalization operation on unitary matrices, i.e., the steps required to produce a gradient of a scalar loss function L with respect to the matrices, given the gradient of L with respect to their eigenvalues and (unit-norm) eigenvectors. It is assumed that the loss function L is independent of the details of the diagonalization procedure, including the overall complex phase of each eigenvector and the permutation of eigenvalues and eigenvectors; this assumption is true for our spectral flows, for example. A gradient backpropagation algorithm suitable for a black-box diagonalization procedure allows us to implement the diagonalization using any approach that is efficient and numerically stable.

Given the $N \times N$ unitary matrix U, we define the eigenvalues and eigenvectors returned by the black-box diagonalization step to be $w = (w_1, \ldots, w_N)$ and $P = (\vec{v}_1, \ldots, \vec{v}_N)$, respectively. By definition of unitary diagonalization, they satisfy

$$U = PDP^\dagger, \qquad D \coloneqq \mathrm{diag}(w). \tag{C1}$$

Equation (C1) does not fully constrain d and P, so they may further depend on U; such dependence (e.g., how the overall phases on each $\vec{v}_i$ are chosen) is an implementation detail of the diagonalization procedure. We define the vector of gradients given as input to the backpropagation step to be

$$g \coloneqq \left(\frac{\partial L}{\partial \mathrm{Re}\, w}, \frac{\partial L}{\partial \mathrm{Im}\, w}, \frac{\partial L}{\partial \mathrm{Re}\, P}, \frac{\partial L}{\partial \mathrm{Im}\, P}\right), \tag{C2}$$

where we implicitly bundle the components of the gradients with respect to w and P into one row vector.[11]

To proceed, we use Eq. (C1) to relate the differential elements dP and dw to dU, and ultimately solve for the Jacobian elements $\partial \mathrm{Re}\, w/\partial \mathrm{Re}\, U$, $\partial \mathrm{Im}\, w/\partial \mathrm{Re}\, U$, …, $\partial \mathrm{Im}\, P/\partial \mathrm{Im}\, U$. Ambiguity due to the implementation details of the diagonalization procedure corresponds to ambiguities in components of the Jacobian that cannot affect L by our assumption above. Therefore, in what follows we simply make a valid choice.

From the unitarity of P and diagonal nature of D, we know

$$\begin{aligned} PdP^\dagger + dPP^\dagger = 0, \quad dP^\dagger P + P^\dagger dP = 0, \\ dw_i = dD_{ii}, \text{ and } dD_{ij} = 0, \quad \forall\ i \neq j. \end{aligned} \tag{C4}$$

From Eq. (C1), we can derive

$$\begin{aligned} U = PDP^\dagger &\Rightarrow P^\dagger UP = D \\ dP^\dagger UP + P^\dagger dUP + P^\dagger UdP &= dD \\ dP^\dagger PD + P^\dagger dUP + DP^\dagger dP &= dD. \end{aligned} \tag{C5}$$

Introducing

$$dH \coloneqq P^\dagger dP = -dP^\dagger P, \tag{C6}$$

which represents the differential of P translated to the identity, we simplify the relation of differential elements to

$$P^\dagger dUP - dD = -DdH + dHD = [dH, D]. \tag{C7}$$

However, since D is diagonal, we know $[dH, D]_{ii} = 0$ and therefore have an explicit expression for dw_i,

$$dw_i = dD_{ii} = [P^\dagger dUP]_{ii}. \tag{C8}$$

We can similarly compute the off-diagonal components of dH,

$$\begin{aligned} [P^\dagger dUP]_{ij} &= (w_j - w_i) dH_{ij}, \quad \forall\ i \neq j \\ dH_{ij} &= \frac{[P^\dagger dUP]_{ij}}{w_j - w_i}. \end{aligned} \tag{C9}$$

The imaginary components of the diagonal elements of dH are unconstrained (dH is anti-Hermitian so the real components are zero), reflecting the fact that the only undefined (continuous) degrees of freedom are the phases on the eigenvectors. We are free to set them to zero, giving a *valid* choice of dP,

[11] Note that machine learning libraries with support for complex numbers may provide such gradients in different formats. For example, the convention used by JAX [73] is to provide the complex-valued gradient vector

$$g^{\mathtt{jax}} = \left(\frac{\partial L}{\partial \mathrm{Re}\, w} - i\frac{\partial L}{\partial \mathrm{Im}\, w}, \frac{\partial L}{\partial \mathrm{Re}\, P} - i\frac{\partial L}{\partial \mathrm{Im}\, P}\right), \tag{C3}$$

which packs the gradient components into complex values, matching the type of w and P [74].

$$dP_{mn} = [PdH]_{mn} = P_{mi}P^{\dagger}_{ij}dU_{jk}P_{kn}V_{\mathrm{in}},$$
$$\text{where } V_{\mathrm{in}} = \begin{cases} \frac{1}{w_n - w_i} & i \neq n \\ 0 & \text{else} \end{cases}. \tag{C10}$$

Having defined a valid solution of the differentials dP and dw in terms of dU, we can solve for all the components of the Jacobian,

$$\frac{\partial \mathrm{Re} w_i}{\partial \mathrm{Re} U_{jk}} = \mathrm{Re}(P^*_{ji}P_{ki})$$
$$\frac{\partial \mathrm{Im} w_i}{\partial \mathrm{Re} U_{jk}} = \mathrm{Im}(P^*_{ji}P_{ki})$$
$$\frac{\partial \mathrm{Re} w_i}{\partial \mathrm{Im} U_{jk}} = -\mathrm{Im}(P^*_{ji}P_{ki})$$
$$\frac{\partial \mathrm{Im} w_i}{\partial \mathrm{Im} U_{jk}} = \mathrm{Re}(P^*_{ji}P_{ki})$$
$$\frac{\partial \mathrm{Re} P_{mn}}{\partial \mathrm{Re} U_{jk}} = \mathrm{Re}(P_{mi}P^*_{ji}P_{kn}V_{in})$$
$$\frac{\partial \mathrm{Im} P_{mn}}{\partial \mathrm{Re} U_{jk}} = \mathrm{Im}(P_{mi}P^*_{ji}P_{kn}V_{in})$$
$$\frac{\partial \mathrm{Re} P_{mn}}{\partial \mathrm{Im} U_{jk}} = -\mathrm{Im}(P_{mi}P^*_{ji}P_{kn}V_{in})$$
$$\frac{\partial \mathrm{Im} P_{mn}}{\partial \mathrm{Im} U_{jk}} = \mathrm{Re}(P_{mi}P^*_{ji}P_{kn}V_{in}). \tag{C11}$$

Together these components form the Jacobian matrix

$$J = \begin{pmatrix} \frac{\partial \mathrm{Re} w}{\partial \mathrm{Re} U} & \frac{\partial \mathrm{Re} w}{\partial \mathrm{Im} U} \\ \frac{\partial \mathrm{Im} w}{\partial \mathrm{Re} U} & \frac{\partial \mathrm{Im} w}{\partial \mathrm{Im} U} \\ \frac{\partial \mathrm{Re} P}{\partial \mathrm{Re} U} & \frac{\partial \mathrm{Re} P}{\partial \mathrm{Re} U} \\ \frac{\partial \mathrm{Im} P}{\partial \mathrm{Re} U} & \frac{\partial \mathrm{Im} P}{\partial \mathrm{Re} U} \end{pmatrix}, \tag{C12}$$

which allows us to backpropagate the gradients by right multiplication,

$$g' \coloneqq \left(\frac{\partial L}{\partial \mathrm{Re} U}, \frac{\partial L}{\partial \mathrm{Im} U}\right) = gJ. \tag{C13}$$

APPENDIX D: CONJUGATION EQUIVARIANT MAPS ON SU(3) VIA AVERAGING

We are interested in building diffeomorphisms of SU(3) that are equivariant under the action by conjugation of SU(3) on itself. We already know from Appendix A that it is enough to build diffeomorphisms of T that are equivariant under the action of its Weyl group. Our goal here is to tackle that problem by lifting it to the Lie algebra $\mathfrak{t}$ of T, where we will *average* diffeomorphisms of $\mathfrak{t}$ to force the equivariance. We will then identify certain sufficient properties that guarantee this still leads to diffeomorphisms of $\mathfrak{t}$ that descend to diffeomorphisms of T.

In this section, we identify the Lie algebra of T with $\mathbb{R}^2$ via the map

$$(x, y) \overset{\exp}{\mapsto} \mathrm{Diag}(e^{2\pi i x}, e^{2\pi i y}, e^{-2\pi i (x+y)}). \tag{D1}$$

Given a map $\tilde{h}: \mathfrak{t} \to \mathfrak{t}$, we say that this map descends to a map on T if there exists h such that the following diagram is commutative:

$$\begin{tikzcd} \mathfrak{t} \arrow[r, "\tilde{h}"] \arrow[d, "\exp"'] & \mathfrak{t} \arrow[d, "\exp"] \\ T \arrow[r, "h"] & T \end{tikzcd} . \tag{D2}$$

If $\tilde{h}$ is equivariant with respect to the action of the Weyl group, then so is h since exp is equivariant and surjective. Given $(x, y) \in \mathbb{R}^2$, we define $z = -x - y$.

The Weyl group $\mathcal{W}$ associated with T is the group of permutations over three elements. This group acts on both T and its Lie algebra $\mathfrak{t}$, and the exponential map is equivariant under these actions. Denote $\sigma_0, \ldots, \sigma_5$ the elements of $\mathcal{W}$.

A map $\tilde{h}: \mathfrak{t} \to \mathfrak{t}$ descends to a map $h: T \to T$ iff

$$\forall x, y \in \mathfrak{t},\ \forall a, b \in \mathbb{Z}^2, \tilde{h}(x + a, y + b) - \tilde{h}(x, y) \in \mathbb{Z}^2. \tag{D3}$$

Also, $\tilde{h}: \mathfrak{t} \to \mathfrak{t}$ if a local diffeomorphism iff its descended map $T \to T$ is a local diffeomorphism.

Proposition 4.—Let $\tilde{h}: \mathfrak{t} \to \mathfrak{t}$ be any map that satisfies Eq. (D3), then

$$G_{\tilde{h}} = \frac{1}{6}\sum_k \sigma_k^{-1}\tilde{h}\sigma_k \tag{D4}$$

also satisfies Eq. (D3) and is equivariant under the action of the Weyl group.

Proof.—The set of maps that satisfy Eq. (D3) is stable under convex combination and composition. It follows that $G_{\tilde{h}}$ satisfies Eq. (D3).

Let us check it is equivariant. Let σ_i be in $\mathcal{W}$. In particular, σ_i is an affine map, and it will preserve barycenters, so that

$$\sigma_i \circ G_{\tilde{h}} = \frac{1}{6}\sum_k \sigma_i \sigma_k^{-1}\tilde{h}\sigma_k$$
$$= \frac{1}{6}\sum_k (\sigma_k \sigma_i^{-1})^{-1}\tilde{h}\sigma_k$$
$$= \frac{1}{6}\sum_k \sigma_k^{-1}\tilde{h}\sigma_k\sigma_i$$
$$= G_{\tilde{h}} \circ \sigma_i. \tag{D5}$$

■

Note that if $\tilde{h}$ is the identity map, then $G_{\tilde{h}}$ is also the identity map. If we start with a $\tilde{h}$ that is close to the identity, we will have constructed an equivariant diffeomorphism of $\mathfrak{t}$, that descends to an equivariant diffeomorphism of T.

In order to ensure that $G_{\tilde{h}}$ is a diffeomorphism, we will now restrict $\tilde{h}$ to a particular form. Namely, assume $f: \mathbb{R} \to \mathbb{R}$ is a diffeomorphism obtained by lifting a diffeomorphism of S^1 to its Lie algebra $\mathbb{R}$, and define $\tilde{h}(x, y) = (f(x), f(y))$. This is a diffeomorphism of $\mathfrak{t}$, and $G_{\tilde{h}}$ is equivariant and descends to an equivariant map $G_h: T \to T$. We would like to find sufficient conditions for $G_{\tilde{h}}$ to descend to a diffeomorphism of T.

Let us start by computing the Jacobian of $G_{\tilde{h}}$.

Proposition 5.—The Jacobian $J(G)$ of $G_{\tilde{h}}$ is given by

$$J(G)(x, y) = \frac{1}{3}\begin{bmatrix} 2f'(x) + f'(z) & f'(z) - f'(y) \\ f'(z) - f'(x) & 2f'(y) + f'(z) \end{bmatrix}. \qquad \text{(D6)}$$

Proof.—This is a direct, albeit a bit tedious, computation using the Jacobians of the elements of the Weyl group. ■

Corollary 1.—The map $G_{\tilde{h}}$ is a diffeomorphism of $\mathfrak{t}$.

Proof.—Let us start by checking that $G_{\tilde{h}}$ is a local diffeomorphism. We only need to check that its Jacobian is always invertible. The determinant of the Jacobian in Proposition 5 simplifies nicely to $\frac{1}{3}(f'(x)f'(y) + f'(y)f'(z) + f'(z)f'(x))$. Since we assumed that f comes from a diffeomorphism of S^1, its derivative is either always strictly positive, or always strictly negative, and the determinant cannot vanish.

Since $\mathfrak{t}$ is simply connected, this means $G_{\tilde{h}}$ is indeed a diffeomorphism of $\mathfrak{t}$ that satisfies Eq. (D3). ■

Corollary 2.—If f is connected to the identity by a path of diffeomorphisms, then $G_{\tilde{h}}$ descends to a diffeomorphism G_h of T that is equivariant under the Weyl group.

Proof.—We already know that $G_{\tilde{h}}$ descends to a local diffeomorphism of T. This is necessarily a covering of T by itself. We only need to prove this covering is trivial. We cannot use the same argument with T as we did with $\mathfrak{t}$, because T is not simply connected.

We have assumed that f is homotopic to the identity of $\mathbb{R}$. This immediately gives us a homotopy from $G_{\tilde{h}}$ to the identity of $\mathfrak{t}$. This descends to a map $k: [0, 1] \times T \to T$. Using local coordinates, we can see that this map is continuous. It therefore defines a homotopy from G_h to the identity of T. In particular, we conclude that G_h must induce the identity map on the fundamental group and is necessarily a trivial covering. ■

Corollary 3.—Any circle diffeomorphism from Ref. [15], such as a mixture of non-compact projections, Möbius, or a spline, can be used to define an equivariant diffeomorphism
of SU(3).

We tested flows based on the equivariant diffeomorphisms suggested by Corollary 3 but found that networks built this way did not perform as well as those used in the main body of the paper. This is likely because using a single circle diffeomorphism in Eq. (D4) is too restrictive. An alternative would be to build a diffeomorphism of the torus from two circle diffeomorphisms by autoregressivity. In that case, Corollary 1 does not apply and one needs to be careful that averaging still leads to a diffeomorphism.

APPENDIX E: THE CASE OF U(N)

The case of U(N) is simpler than SU(N) because we do not have the constraint that the determinant must be equal to 1. We could apply the same strategy used for the SU(N) flows via a canonicalization map to map every point to a canonical cell and then build a flow in the N-simplex cell (in contrast to the $(N-1)$-simplex cell for SU(N)). An alternative and simpler direction is to directly build a permutation equivariant flow on the torus T^N. This can be achieved by first mapping T^N to $\mathbb{R}^N$ using a noncompact projection [15,75], then stacking layers alternating between those defined by Eqs. (13) and (15) or Eq. (16) in Ref. [42], before finally projecting back to T^N. We tested this flow on U(3) using the target action given in Eq. (18) with coefficients $c^{(0)}$ from Table I and $\beta = 1, 5, 9$. The flow quickly converged with ESS of more than 95% in each case.

[1] G. D. Kribs and E. T. Neil, Int. J. Mod. Phys. A **31**, 1643004 (2016).

[2] T. DeGrand, Rev. Mod. Phys. **88**, 015001 (2016).

[3] I. Ichinose and T. Matsui, Mod. Phys. Lett. B **28**, 1430012 (2014).

[4] S. Sachdev, Rep. Prog. Phys. **82**, 014001 (2019).

[5] R. Samajdar, M. S. Scheurer, S. Chatterjee, H. Guo, C. Xu, and S. Sachdev, Nat. Phys. **15**, 1290 (2019).

[6] H. Guo, R. Samajdar, M. S. Scheurer, and S. Sachdev, Phys. Rev. B **101**, 195126 (2020).

[7] Z. Bi and T. Senthil, Phys. Rev. X **9**, 021034 (2019).

[8] A. Giveon and D. Kutasov, Rev. Mod. Phys. **71**, 983 (1999).

[9] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, arXiv:1902.04615.

[10] E. J. Bekkers, arXiv:1909.12057.

[11] 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).

[12] M. Lezcano-Casado and D. Martínez-Rubio, arXiv:1901.08428.

[13] M. Lezcano-Casado, arXiv:1909.09501.

[14] L. Falorsi, P. de Haan, T. R. Davidson, and P. Forré, in *Proceedings of Machine Learning Research*, Vol. 89, edited by K. Chaudhuri and M. Sugiyama (PMLR, 2019), pp. 3244–3253.

[15] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, arXiv:1912.02762.

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

[17] 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).

[18] C. Morningstar, in *21st Annual Hampton University Graduate Studies Program (HUGS 2006), Newport News, VA* (2007).

[19] J. M. Pawlowski and J. M. Urban, Mach. Learn. Sci. Technol. **1**, 045011 (2020).

[20] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker, Phys. Rev. D **100**, 011501 (2019).

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

[22] L. Zhang, W. E, and L. Wang, arXiv:1809.10188.

[23] F. Noé, S. Olsson, J. Köhler, and H. Wu, Science **365** (2019).

[24] S.-H. Li, C.-X. Dong, L. Zhang, and L. Wang, Phys. Rev. X **10**, 021020 (2020).

[25] G. S. Hartnett and M. Mohseni, arXiv:2001.00585.

[26] R. A. Wijsman, 5. Lie groups and lie algebras, in *Invariant Measures on Groups and Their Use in Statistics*, Lecture Notes–Monograph Series, Vol. 14 (Institute of Mathematical Statistics, Hayward, CA, 1990), Chap. 5, pp. 67–97.

[27] D. Jimenez Rezende, S. Mohamed, and D. Wierstra, arXiv:1401.4082.

[28] D. P. Kingma and M. Welling, arXiv:1312.6114.

[29] M. Titsias and M. Lázaro-Gredilla, in *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32, edited by E. P. Xing and T. Jebara (PMLR, Bejing, China, 2014), pp. 1971–1979.

[30] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, J. Am. Stat. Assoc. **112**, 859 (2017).

[31] D. B. Rubin, J. Am. Stat. Assoc. **82**, 543 (1987).

[32] T. Cohen and M. Welling, Proc. Mach. Learn. Res. **48**, 2990 (2016), http://proceedings.mlr.press/v48/cohenc16.html.

[33] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, Proc. Mach. Learn. Res. **97**, 1321 (2019), http://proceedings.mlr.press/v97/cohen19d.html.

[34] J. Köhler, L. Klein, and F. Noé, arXiv:1910.00753.

[35] D. Jimenez Rezende, S. Racanière, I. Higgins, and P. Toth, arXiv:1909.13739.

[36] P. Wirnsberger, A. J. Ballard, G. Papamakarios, S. Abercrombie, S. Racanière, A. Pritzel, D. Jimenez Rezende, and C. Blundell, arXiv:2002.04913.

[37] M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson, arXiv:2002.12880.

[38] M. Creutz, Phys. Rev. D **15**, 1128 (1977).

[39] C. DeTar, J. E. King, S. P. Li, and L. McLerran, Nucl. Phys. **B249**, 621 (1985).

[40] T. DeGrand and C. E. Detar, *Lattice Methods for Quantum Chromodynamics* (World Scientific, Singapore, 2006).

[41] J. Bender and E. Zohar, Phys. Rev. D **102**, 114517 (2020).

[42] C. Bender, K. O'Connor, Y. Li, J. J. Garcia, M. Zaheer, and J. Oliva, arXiv:1902.01967.

[43] K. Rasul, I. Schuster, R. Vollgraf, and U. Bergmann, arXiv:1909.02775.

[44] J. Köhler, L. Klein, and F. Noé, arXiv:2006.02425.

[45] N. Guttenberg, N. Virgo, O. Witkowski, H. Aoki, and R. Kanai, arXiv:1612.04530.

[46] J. Rahme, S. Jelassi, J. Bruna, and S. M. Weinberg, arXiv:2003.01497.

[47] S. Ravanbakhsh, J. Schneider, and B. Poczos, arXiv:1702.08389.

[48] J. Gordon, D. Lopez-Paz, M. Baroni, and D. Bouchacourt, in *International Conference on Learning Representations, Addis Ababa, Ethiopia* (2020).

[49] H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman, arXiv:1812.09902.

[50] N. Segol and Y. Lipman, arXiv:1910.02421.

[51] A. Sannai, Y. Takai, and M. Cordonnier, arXiv:1903.01939.

[52] P. T. Komiske, E. M. Metodiev, and J. Thaler, J. High Energy Phys. 01 (2019) 121.

[53] D. Bump, The Weyl integration formula, in *Lie Groups* (Springer, New York, New York, NY, 2004), pp. 112–116.

[54] J. J. Duistermaat and J. A. Kolk, *Lie Groups* (Springer Science & Business Media, New York, 2012), Chap. 3.

[55] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, in *Advances in Neural Information Processing Systems 32, Vancouver, Canada* (2019), pp. 7511–7522.

[56] D. P. Kingma and J. Ba, arXiv:1412.6980.

[57] F. Mezzadri, Not. Am. Math. Soc. **54**, 592 (2006),

[58] A. L. Maas, A. Y. Hannun, and A. Y. Ng, in *Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia* (2013), Vol. 30, p. 3.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, in *Advances in Neural Information Processing Systems*, Vol. 32 edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., Vancouver, Canada, 2019), pp. 8024–8035.

[60] J. Snoek, H. Larochelle, and R. P. Adams, in *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 2, NIPS'12* (Curran Associates Inc., Red Hook, NY, 2012), pp. 2951–2959.

[61] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, and S. M. Wild, in *25th IEEE International Conference on High Performance Computing (HiPC), Bengaluru, India* (2018), pp. 42–51.

[62] P. Balaprakash, R. Egele, M. Salim, S. Wild, V. Vishwanath, F. Xia, T. Brettin, and R. Stevens, arXiv:1909.00311.

[63] D. Sigdel, Ph.D. thesis, FIU, 2016.

[64] D. Gross and E. Witten, Phys. Rev. D **21**, 446 (1980).

[65] S. R. Wadia, arXiv:1212.2906.

[66] U. Wolff (ALPHA Collaboration), Comput. Phys. Commun. **156**, 143 (2004); **176**, 383(E) (2007).

[67] 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).
[68] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).
[69] W. R. Inc., Mathematica, Version 12.1, Champaign, IL, 2020.
[70] http://iaifi.org/
[71] M. Koca, N. Ozdes Koca, A. Al-Siyabi, and R. Koc, Acta Crystallogr. Sect. A **74**, 499 (2018).
[72] V. Guillemin, E. Lerman, and S. Sternberg, *Symplectic Fibrations and Multiplicity Diagrams* (Cambridge University Press, Cambridge, United Kingdom, 1996), Chap. 5.
[73] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne, JAX: Composable transformations of Python +NumPy programs, 2018, http://github.com/google/jax.
[74] D. Maclaurin, Ph.D. thesis, Harvard University, 2016.
[75] M. C. Gemici, D. Rezende, and S. Mohamed, arXiv:1611.02304.