Letter

Learning trivializing gradient flows for lattice gauge theories

Simone Bacchio,[1] Pan Kessel,[2,3] Stefan Schaefer,[4] and Lorenz Vaitl[2]

[1]*Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[2]*Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[3]*BIFOLD Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[4]*John von Neumann-Institut für Computing NIC, Deutsches Elektronen-Synchrotron DESY, Germany*

(Received 5 January 2023; accepted 28 February 2023; published 20 March 2023)

We propose a unifying approach that starts from the perturbative construction of trivializing maps by Lüscher and then improves on it by learning. The resulting continuous normalizing flow model can be implemented using common tools of lattice field theory and requires several orders of magnitude fewer parameters than any existing machine learning approach. Specifically, our model can achieve competitive performance with as few as 14 parameters while existing deep-learning models have around 1 million parameters for $SU(3)$ Yang-Mills theory on a 16^2 lattice. This has obvious consequences for training speed and interpretability. It also provides a plausible path for scaling machine-learning approaches toward realistic theories.

DOI: 10.1103/PhysRevD.107.L051504

I. INTRODUCTION

Critical slowing down constitutes one of the fundamental challenges of modern computational sciences. A particular situation in which critical slowing down is observed is the continuum limit of lattice field theories. An important theme of research of the last few decades is to construct algorithms, such as cluster methods [1,2], that are (relatively) insensitive to critical slowing down. Despite significant effort, a cluster algorithm for lattice quantum chromodynamics has unfortunately not been found.

An interesting alternative approach to circumvent critical slowing down was proposed in a seminal paper [3] by Lüscher more than ten years ago: building on previous work in supersymmetric field theories [4], Lüscher proposed to use a field redefinition that maps pure $SU(3)$ Yang-Mills theory to its strong coupling limit and therefore trivializes it. In the redefined field variables, the theory is not (severely) affected by critical slowing down even if standard HMC algorithms are applied. Despite the considerable conceptual appeal, the main practical challenge of this approach is to construct such trivializing maps. Lüscher proposed a perturbative construction of a gradient flow that realizes the trivializing map to linear order in the flow time. Unfortunately, the expansion is around the strong coupling limit, and therefore, the approximation deteriorates as the continuum is approached. Indeed, in HMC simulations of the CP^{N-1} model, the autocorrelations were reduced, but the scaling was not substantially changed [5].

Recently, trivializing maps have gained renewed attention since they can be combined with deep learning [6–23]. In this new line of work, the trivializing map is modeled by a deep neural network. The parameters of the neural network are then trained to trivialize the redefined theory. This approach is interesting for at least three reasons: (i) it circumvents any expansion in flow time, (ii) it is (at least in principle) completely general as neural networks are universal approximators, and (iii) it establishes a connection to the rapidly progressing field of machine learning, raising the exciting prospect of using specialized deep-learning silicon, such as tensor processing units (TPUs), for lattice simulations. Despite these advantages, state-of-the-art methods for learning trivializing maps only work for low-dimensional theories. This is partly because the training relies on self-sampling from the model. In order to attain a useful gradient signal, the model has to probe relevant regions of field space of the lattice field theory. As the dimensionality of the theory increases, these regions are very unlikely to be sampled, and training will fail. This is one of the reasons for the poor volume scaling of current deep-learning-based approaches [14,15,20,24].

In this work, we outline a strategy that unifies Lüscher's perturbative approach with the recent machine-learning approach, in particular, continuous normalizing flows [18,19,25,26]. More specifically, we propose using the same gradient flow as Lüscher, but instead of fixing its coefficients by perturbation theory, we use machine-learning techniques, namely, the adjoint state method.

This has the advantage that we can use Lüscher's perturbative results to initialize the machine-learning model. The resulting model is not only manifestly equivariant under all global as well as local symmetries of the theory, but it is also constrained to a low number of required free parameters. We demonstrate that we can obtain comparable performance to the current state-of-the-art deep-learning models, which have around 1 million parameters, with as few as $\mathcal{O}(10)$ parameters, and significantly outperform them by using $\mathcal{O}(100)$ parameters. We demonstrate that the low number of parameters and the perturbative initialization are particularly beneficial in the early stage of training. This is encouraging as this phase is the main hurdle in scaling machine-learning approaches. A further advantage of our method is that it can be implemented using standard lattice tools and does not require any specialized deep-learning libraries.

The main objective of this paper is to introduce our approach and demonstrate its superior performance when compared to previous methods. For this, we restrict ourselves to two-dimensional Yang-Mills theory as this is the benchmark on which we can compare to the existing literature, leaving applications to higher-dimensional theories for future work.

II. TRIVIALIZING MAPS

In lattice gauge theory, the expectation value of a physical observable $\mathcal{O}$ is given by the Wick-rotated path integral

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

discretized on a lattice Λ. Using a (diffeomorphic) field redefinition $U = \mathcal{F}(V)$, this expectation value can be rewritten as

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[V] \mathcal{O}(\mathcal{F}(V)) \exp(-S_{\mathcal{F}}(V)), \tag{2}$$

where we have defined

$$S_{\mathcal{F}}(V) = S(\mathcal{F}(V)) - \ln \det \mathcal{F}_*(V). \tag{3}$$

The last term involving the Jacobian $\mathcal{F}_*$ is due to the change of measure $\mathrm{D}[U] = \mathrm{D}[V] \det \mathcal{F}_*(V)$.

For a trivializing map $\mathcal{F}$, this measure contribution cancels the action up to a possible constant, i.e.,

$$S_{\mathcal{F}}(V) = \mathrm{const.} \tag{4}$$

The expectation value (2) can thus be calculated using the uniform density. Such trivializing maps $\mathcal{F}$ can be constructed analytically for certain supersymmetric field theories [4]. For the case of $SU(N)$ Yang-Mills theory, a perturbative construction was put forward by Lüscher in [3].

Recently, it has been proposed to use machine learning to obtain trivializing maps nonperturbatively. In this approach, the redefinition $\mathcal{F}_\theta$ is given by a bijective machine-learning model with parameters θ [6–20]. The model is then trained by minimizing the objective function

$$\mathcal{C}(\theta) = \langle S_{\mathcal{F}_\theta}(V) \rangle, \tag{5}$$

using stochastic gradient descent. The parameters $\hat{\theta}$ are a global minimum of the objective $\mathcal{C}(\theta)$ if and only if the corresponding map $\mathcal{F}_{\hat{\theta}}$ is trivializing, i.e., fulfills the trivializing condition (4). We refer to the Supplemental Material [27] for a proof.

In practice, we cannot expect the model to be perfectly trained; i.e., $\mathcal{F}_\theta$ does not fulfill the trivializing condition (4) and thus does not completely reduce the target density $p(V) = \frac{1}{\mathcal{Z}} \exp(-S_{\mathcal{F}_\theta}(V))$ to the uniform density. One can, however, use the uniform density, $q(V) = \mathrm{const}$, as a proposal for a Markov chain to sample from $p(V)$. Specifically, one advances the Markov chain from a previous configuration V to some current configuration by accepting a candidate $V' \sim q$ with probability

$$p_A = \min\left(1, \frac{w(V')}{w(V)}\right), \tag{6}$$

with the importance weight $w(V) = \frac{p(V)}{q(V)}$. For a sufficiently trained model, the proposal V' for the update will be accepted with high probability. As a result, autocorrelation will be low as it can only arise due to repeated rejection of proposals (since the proposals are sampled independently from q).

III. LÜSCHER'S PERTURBATIVE APPROACH

Lüscher proposed a flow equation given by

$$\dot{U}_t = Z_t(U_t) U_t \tag{7}$$

which is generated by an algebra-valued link field $[Z_t(U)](x, \mu) \in \mathfrak{su}(N)$.[1] If Z_t is a smooth function, the solution U_t is unique for a given initial condition $U_0 = V$ at any $t \in \mathbb{R}$. Therefore, the flow equation implicitly defines a bijective field redefinition $\mathcal{F}(V) = U_t$, where we suppress the dependency of the map $\mathcal{F}$ on the chosen time t for notational simplicity.

It is natural to parametrize Z_t as the negative force of a certain flow action $\tilde{S}$, i.e.,

$$[Z^a(U_t)](x, \mu) = -\partial^a_{x,\mu} \tilde{S}(U_t), \tag{8}$$

where we define

[1] $[\dot{U}_t U_t^\dagger](x, \mu)$ is obviously anti-Hermitian and can easily be shown to be traceless by using Jacobi's formula. This in turn implies that Z_t is Lie-algebra valued.

$$\partial^a_{x,\mu} f(U) = \frac{d}{d\tau} f(U_\tau)\Big|_{\tau=0} \tag{9}$$

with

$$U_\tau(y,\nu) = \begin{cases} e^{\tau T^a} U(x,\mu) & \text{for } (x,\mu) = (y,\nu) \\ U(y,\nu) & \text{else.} \end{cases} \tag{10}$$

It can be shown [3] that the determinant of the Jacobian of the redefinition $\mathcal{F}(V) = U_t$ is given by

$$\ln \det \mathcal{F}_*(V) = \int_0^t \mathrm{d}s\, \mathcal{L}_0 \tilde{S}(U_s), \tag{11}$$

with the Laplacian

$$\mathcal{L}_0 = -\sum_{x,\mu} \partial^a_{x,\mu} \partial^a_{x,\mu}. \tag{12}$$

In Lüscher's approach, the flow action $\tilde{S}$ is given by a linear combination of a certain set of Wilson loops $\mathcal{W}_i$,

$$\tilde{S}(U_t, t) = \sum_i c_i(t) \mathcal{W}_i(U_t), \tag{13}$$

where c_i are coefficient functions that need to be determined. Lüscher then proposed expanding this action in flow time perturbatively,

$$\tilde{S}(U_t, t) = \sum_{k=0}^{\infty} t^k \tilde{S}^{(k)} \tag{14}$$

with $t \in [0, 1]$. Lüscher explicitly determined the leading order $\tilde{S}^{(0)}$ and next-to-leading order $\tilde{S}^{(1)}$ of the expansion.

IV. MACHINE-LEARNING APPROACH

We closely follow Lüscher's construction but circumvent the perturbative approximation of the coefficients by using machine-learning techniques. More specifically, we parametrize the coefficient functions by a simple ansatz, such as affine linear functions or cubic splines, and then learn their parameters θ by stochastic gradient descent. An advantage of this approach is that the free parameters of the coefficient functions can be initialized such that the coefficient functions match the perturbative results obtained by Lüscher. In stark contrast to standard deep-learning approaches, this provides a rigorous initialization scheme based on perturbation theory that can systematically be improved by incorporating perturbative corrections of higher order. In addition, the resulting approach only requires common tools in lattice gauge theories, such as the ability to compute a generic action $\tilde{S}$ made of Wilson loops (13) and its force (8). Furthermore, only a very limited number of free parameters are required for the coefficient functions, resulting in a drastic overall reduction of the necessary number of parameters for an expressive model. This is beneficial both for training speed and interpretability, e.g., identifying Wilson loops that are most important for the trivialization.

The main technical challenge of such an approach is to calculate the gradients of the objective $\frac{\partial \mathcal{C}}{\partial \theta}$. This is nontrivial, as one has to take the parameter dependence of the flow equation (7) into account. We overcome this challenge by using the adjoint state method; see [28] for a review. To this end, we derive a specific version of the adjoint state method for the Lie group $SU(N)$. Unlike previous work on the adjoint state method on manifolds [29–32], our method is particularly suited to the $SU(N)$ case and can be efficiently implemented using existing libraries for lattice field theory. We refer to the Supplemental Material [27] for a detailed derivation but summarize the main results in the following.

The adjoint state method starts from the observation that the optimization criterion (5) is to be minimized on the solution space of the differential equation (7). We, therefore, introduce an $\mathfrak{su}$(N)-valued Lagrange multiplier λ, which in this context is also called the adjoint state, and define the Lagrangian

$$L(\theta) = \mathcal{C}(\theta) - \left\langle \int_0^t \mathrm{d}s (\lambda_s, \dot{U}_s U_s^\dagger - Z_s) \right\rangle_q, \tag{15}$$

with the standard inner product on the $\mathfrak{su}(N)$ Lie algebra

$$(A, B) \equiv -2 \sum_{x,\mu} \mathrm{tr}(A(x,\mu) B(x,\mu)). \tag{16}$$

On the solution space, the objective $\mathcal{C}$ and the Lagrangian L agree. By differentiating the Lagrangian, it can be shown that its gradient is given by

$$\frac{\partial \mathcal{C}}{\partial \theta} = \frac{\partial L}{\partial \theta} = \left\langle \int_0^t \mathrm{d}s \{(\lambda_s, \partial_\theta Z_s) - \partial_\theta \mathcal{L}_0 \tilde{S}_s\} \right\rangle_q \tag{17}$$

when the adjoint state fulfills the terminal value problem,

$$\begin{aligned} \dot{\lambda}_s &= \partial \mathcal{L}_0 \tilde{S}_s + [Z_s, \lambda_s] - \sum_{y,\nu} \lambda_s^a(y,\nu) \partial Z_s^a(y,\nu), \\ \lambda_t &= \partial S(U_t), \end{aligned} \tag{18}$$

where we have used the notation $[\partial f(U)](x,\mu) = T^a \partial^a_{x,\mu} f(U)$, $\mathcal{L}_0$ is the Laplacian defined in (12), and t denotes the terminal flow time.

Therefore, we can calculate the gradient $\partial_\theta \mathcal{C}$ by evolving the flow equation (18) for the adjoint state λ_s backwards in flow time. This has a comparable numerical cost to solving the flow equation (7) for the gauge configuration U_t. As a result, the cost of the adjoint state method does not scale with the number of parameters, in stark contrast to finite differences.

Furthermore, it can be shown that the adjoint state λ_0 corresponds to the force of the action (3) at zero flow time,

TABLE I. For Lüscher, the coefficients of the next-to-leading order calculations of [3] are used; the ESS for Boyda *et al.* is as reported in [8]. The total number of parameters $N_{\text{params}} = N_t \times N_W$ of our approach is divided due to the number of Wilson loops N_W and parameters per coefficient function N_t.

			ESS at β		
Reference		N_{params}	4.0	5.0	6.0
Lüscher, NL [3]		8 nonzero values	42%	4%	<1%
This work	A	$14 \equiv 2_t \times 7_W$	91%	65%	26%
	B	$420 \equiv 10_t \times 42_W$	98%	88%	70%
Boyda *et al.* [8]		$\mathcal{O}(10^6)$ estimated	88%	75%	48%

$$\lambda_0^a(\mu, x) = \hat{\partial}^a_{\mu,x} S_{\mathcal{F}}(V). \tag{19}$$

We note that this force can be used for a hybrid Monte Carlo algorithm in the trivialized field V. We plan to explore this possibility as part of future work.

V. NUMERICAL RESULTS

We compare our method to the state-of-the-art deep-learning approach of [8], which we estimate uses $\mathcal{O}(10^6)$ parameters. In order to compare, we closely follow their reported experiments. To this end, we consider the two-dimensional $SU(3)$ Yang-Mills theory with the standard Wilson action

$$S_W(U) = -\frac{\beta}{6}\mathcal{W}_0(U), \tag{20}$$

where $\mathcal{W}_0$ denotes the sum of plaquettes. We train on a 16×16 lattice size with $\beta = 4.0$, 5.0, and 6.0 using two architectures: models A and B. Model A has 14 free parameters and uses affine linear coefficient functions for the seven Wilson loops of Lüscher's perturbative construction. Model B has 420 free parameters and uses cubic splines with ten knots as coefficient functions for 42 Wilson loops. Namely, all loops up to length 8, in combination with their moments and correlation functions of plaquettes, are included in its flow action (13). We refer to the Supplemental Material [27] for a detailed description.

The training of model A is initialized using Lüscher's perturbative solution. Since the flow action is a simple linear combination of Wilson loops, we initialize the training of model B from the trained model A. Specifically, we initialize the more expressive cubic spline coefficient functions such that they reproduce the affine linear coefficient functions learnt by the smaller model A and initialize the coefficients of the additional Wilson loops by small random numbers. This possibility for progressive training is a notable advantage of our approach.

For time integration, we use 20 steps of a third-order Crouch-Grossmann integrator [33], which is a suitable Runge-Kutta quadrature scheme for a Lie group. The Adam optimizer [34], with a minibatch of size 1024 and a learning rate of 0.0005, is chosen. Due to the low memory footprint of the model, each integration step can be checkpointed to reduce the error in the backward integration. The variance of the gradients is reduced by using the path-gradient VarGrad estimator [35–37]. The quality of the model is quantified using the effective sampling size (ESS),

$$\text{ESS} = \frac{1}{\langle w(V)^2 \rangle_q} \in [0, 1], \tag{21}$$

with the values reported in [8] for comparison.

The results in Table I show that our model B can significantly outperform the deep-learning-based approach by [8]. This is despite the fact that it has several orders of magnitude less parameters. This point is further illustrated by the fact that the smaller model A achieves comparable performance with only 14 parameters. Table I also shows that our models lead to a significantly larger effective sampling size than the perturbative construction by Lüscher, establishing the idea that machine learning can substantially improve upon the perturbative scheme. At the same time, our approach can benefit from this perturbative scheme as the Lüscher initialization provides a good starting point for training; see Fig. 1.

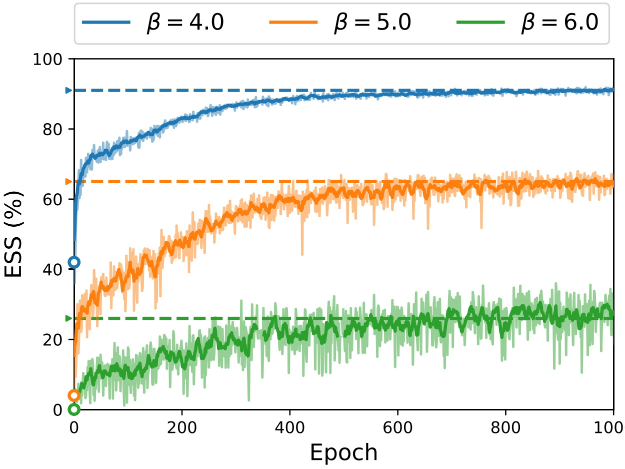

FIG. 1. ESS measured during training of model A starting from Lüscher's initialization. The faint line is the ESS over a single minibatch. The thick line is a moving average over six steps. The empty circle at zero indicates the initial ESS. The horizontal dashed line is the ESS measured at high accuracy after training.

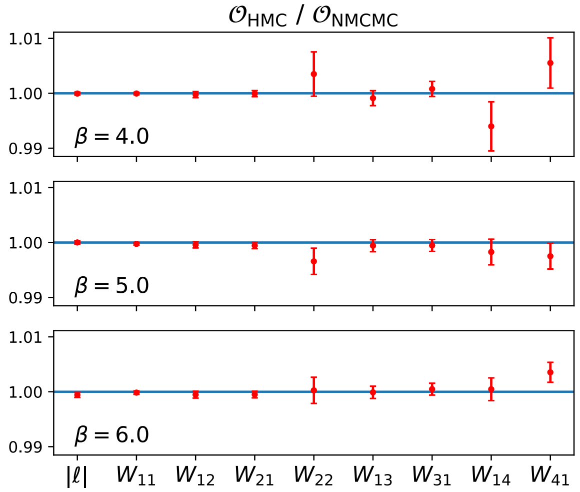

FIG. 2. Observables estimated by both HMC and flow NMCMC [38] [see Eq. (6)] using a batch size of 1024 configurations and 1000 samples, i.e., 1,024,000 samples in total.

As a final consistency check, Fig. 2 demonstrates the compatibility of estimates obtained by our method with ones from the HMC algorithm using the same observables as in [8].

VI. CONCLUSION

In this work, we have proposed a new method to learn trivializing maps, which is natural for Lie groups and has several specific strengths: (i) considerable parameter efficiency, (ii) a high level of interpretability, (iii) initialization based on perturbation theory, (iv) equivariance with respect to all symmetries of the theory, (v) progressive training, as well as (vi) the possibility of implementation and parallelization with standard lattice QCD libraries.

To the best of our knowledge, the present work is the first to apply the adjoint state method in the context of lattice gauge theory. This has the beneficial feature of providing the force of the flow action and thus allows for HMC in the trivialized field variables V, as originally proposed in [3] and recently applied to normalizing flows in [39]. This constitutes a promising route for future research, in particular, as such a HMC could also be used during training. Our approach could also be applied to correct mistuned simulation parameters as an alternative to reweighting. Furthermore, it can be naturally combined with domain decomposition algorithms; see [20,40].

As the purpose of this paper was to benchmark our approach with respect to existing methods, we leave the application to the four-dimensional case, which manifestly suffers from critical slowing down [41], for upcoming work. The beneficial properties of our method, as demonstrated by this work, make successful applications in this higher dimensional case significantly more plausible. If successful, applications of machine-learning-based trivializing maps to full QCD would be within reach.

ACKNOWLEDGMENTS

We wish to acknowledge Michael Gunther and Kevin Schafers for useful discussions on Runge-Kutta integrators for Lie groups, and Jacob Finkenrath for useful discussions during the implementation. S. B. is supported by the H2020 project PRACE 6-IP (Grant Agreement No. 82376) and the EuroCC project (Grant Agreement No. 951740). P. K. is supported by the German Ministry for Education and Research (BMBF) as BIFOLD–Berlin Institute for the Foundations of Learning and Data under Grants No. 01IS18025A and No. 01IS18037A. S. S. is supported by the German Research Foundation (DFG) research unit FOR5269 “Future methods for studying confined gluons in QCD.” L. V. thanks the Cyprus Institute for hospitality. Numerical results were obtained using the Cyclamen cluster of The Cyprus Institute equipped with P100 GPUs. The software is implemented using the Lyncs-API [42], its interface to QUDA [43], and the QUDA library [44]. All figures were obtained by the authors under a CC BY 4.0 license,[2] unless otherwise stated.

[2] https://creativecommons.org/licenses/by/4.0/.

[1] U. Wolff, Collective Monte Carlo Updating for Spin Systems, Phys. Rev. Lett. **62**, 361 (1989).

[2] R. H. Swendsen and J.-S. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, Phys. Rev. Lett. **58**, 86 (1987).

[3] M. Lüscher, Trivializing maps, the Wilson flow and the HMC algorithm, Commun. Math. Phys. **293**, 899 (2010).

[4] H. Nicolai, Supersymmetry and functional integration measures, Nucl. Phys. **B176**, 419 (1980).

[5] G. P. Engel and S. Schaefer, Testing trivializing maps in the hybrid Monte Carlo algorithm, Comput. Phys. Commun. **182**, 2107 (2011).

[6] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Flow-based generative models for Markov chain Monte Carlo in lattice field theory, Phys. Rev. D **100**, 034515 (2019).

[7] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Equivariant Flow-Based Sampling for Lattice Gauge Theory, Phys. Rev. Lett. **125**, 121601 (2020).

[8] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Sampling using $SU(N)$ gauge equivariant flows, Phys. Rev. D **103**, 074504 (2021).

[9] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Flow-based sampling for fermionic lattice field theories, Phys. Rev. D **104**, 114507 (2021).

[10] D. C. Hackett, C.-C. Hsieh, M. S. Albergo, D. Boyda, J.-W. Chen, K.-F. Chen, K. Cranmer, G. Kanwar, and P. E. Shanahan, Flow-based sampling for multimodal distributions in lattice field theory, arXiv:2107.00734.

[11] M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Flow-based sampling in the lattice Schwinger model at criticality, Phys. Rev. D **106**, 014514 (2022).

[12] R. Abbott *et al.*, Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions, Phys. Rev. D **106**, 074506 (2022).

[13] R. Abbott *et al.*, Sampling QCD field configurations with gauge-equivariant flow models, Proc. Sci. LATTICE2022 (**2023**) 036.

[14] L. Del Debbio, J. M. Rossney, and M. Wilson, Efficient modeling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability, Phys. Rev. D **104**, 094507 (2021).

[15] L. Del Debbio, J. M. Rossney, and M. Wilson, Machine learning trivializing maps: A first step towards understanding how flow-based samplers scale up, Proc. Sci. LATTICE2021 (**2022**) 059.

[16] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Machine learning of thermodynamic observables in the presence of mode collapse, Proc. Sci. LATTICE2021 (**2022**) 338.

[17] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models, Phys. Rev. Lett. **126**, 032001 (2021).

[18] P. de Haan, C. Rainone, M. C. N. Cheng, and R. Bondesan, Scaling up machine learning for quantum field theory with equivariant continuous flows, arXiv:2110.02673.

[19] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, Learning lattice quantum field theories with equivariant continuous flows, arXiv:2207.00283.

[20] J. Finkenrath, Tackling critical slowing down using global correction steps with equivariant flows: The case of the Schwinger model, arXiv:2201.02216.

[21] M. Caselle, E. Cellini, A. Nada, and M. Panero, Stochastic normalizing flows as non-equilibrium transformations, J. High Energy Phys. 07 (2022) 015.

[22] M. Favoni, A. Ipp, and D. I. Müller, Applications of lattice gauge equivariant neural networks, EPJ Web Conf. **274**, 09001 (2022).

[23] M. Favoni, A. Ipp, D. I. Müller, and D. Schuh, Lattice Gauge Equivariant Convolutional Neural Networks, Phys. Rev. Lett. **128**, 032003 (2022).

[24] R. Abbott *et al.*, Aspects of scaling and scalability for flow-based sampling of lattice QCD, arXiv:2211.07541.

[25] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, FFJORD: Free-form continuous dynamics for scalable reversible generative models, arXiv:1810.01367.

[26] J. Köhler, L. Klein, and F. Noé, Equivariant flows: Exact likelihood generative learning for symmetric densities, in *International Conference on Machine Learning* (PMLR, 2020), pp. 5361–5370.

[27] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevD.107.L051504 for more details on proofs, derivations and training models.

[28] B. Sengupta, K. J. Friston, and W. D. Penny, Efficient gradient computation for dynamical models, NeuroImage **98**, 521 (2014).

[29] L. Falorsi, Continuous normalizing flows on manifolds, arXiv:2104.14959.

[30] E. Mathieu and M. Nickel, Riemannian continuous normalizing flows, Adv. Neural Inf. Process. Syst. **33**, 2503 (2020).

[31] I. Katsman, A. Lou, D. Lim, Q. Jiang, S. N. Lim, and C. M. De Sa, Equivariant manifold flows, Adv. Neural Inf. Process. Syst. **34**, 10600 (2021).

[32] A. Lou, D. Lim, I. Katsman, L. Huang, Q. Jiang, S. N. Lim, and C. M. De Sa, Neural manifold ordinary differential equations, Adv. Neural Inf. Process. Syst. **33**, 17548 (2020).

[33] P. E. Crouch and R. Grossman, Numerical integration of ordinary differential equations on manifolds, J. Nonlinear Sci. **3**, 1 (1993).

[34] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[35] L. Richter, A. Boustati, N. Nüsken, F. Ruiz, and O. D. Akyildiz, VarGrad: A low-variance gradient estimator for variational inference, Adv. Neural Inf. Process. Syst. **33**, 13481 (2020).

[36] L. Vaitl, K. A. Nicoli, S. Nakajima, and P. Kessel, Path-gradient estimators for continuous normalizing flows, in *International Conference on Machine Learning* (PMLR, 2022), pp. 21945–21959.

[37] L. Vaitl, K. A. Nicoli, S. Nakajima, and P. Kessel, Gradients should stay on path: Better estimators of the reverse- and forward KL divergence for normalizing flows, Mach. Learn.: Sci. Technol. **3**, 045006 (2022).

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

[39] D. Albandea, L. Del Debbio, P. Hernández, R. Kenway, J. M. Rossney, and A. Ramos, Learning trivializing flows, *39th International Symposium on Lattice Field Theory* (2022), arXiv:2211.12806.

[40] M. Lüscher, Schwarz-preconditioned HMC algorithm for two-flavour lattice QCD, Comput. Phys. Commun. **165**, 199 (2005).

[41] S. Schaefer, R. Sommer, and F. Virotta (ALPHA Collaboration), Critical slowing down and error analysis in lattice QCD simulations, Nucl. Phys. **B845**, 93 (2011).

[42] S. Bacchio, J. Finkenrath, and C. Stylianou, Lyncs-API: A PYTHON API for Lattice QCD applications, Proc. Sci. LATTICE2021 (**2022**) 542.

[43] S. Yamamoto, S. Bacchio, and J. Finenrath, Running HMC simulation with PYTHON via QUDA, Proc. Sci. LATTICE2022 (**2023**) 346.

[44] M. A. Clark, R. Babich, K. Barros, R. C. Brower, and C. Rebbi, Solving lattice QCD systems of equations using mixed precision solvers on GPUs, Comput. Phys. Commun. **181**, 1517 (2010).