Machine learning for sampling in lattice quantum field theory
Snowmass Computational Frontier Letter of Interest

Michael S. Albergo,[1] Denis Boyda,[2] Kyle Cranmer,[1] Daniel C. Hackett,[2,*]
Gurtej Kanwar,[2] Phiala E. Shanahan,[2,†] and Julian M. Urban[3]

[1] *Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[2] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*
[3] *Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany*

(Dated: August 31, 2020)

Topical Groups:

■ (CompF3) Machine Learning
■ (CompF2) Theoretical Calculations and Simulation
■ (TF05) Lattice Gauge Theory
■ (CompF4) Storage and processing resource access (Facility and Infrastructure R&D)

Advances in artificial intelligence over the past few years and across virtually all fields of computational science have demonstrated that algorithms based on machine learning (ML) can be more efficient than and/or enable qualitatively new sorts of computations over human-designed algorithms. Recent efforts have begun the work of adapting and deploying these methods for use in theoretical physics where, unlike in many typical artificial intelligence applications, it is often critical to guarantee exactness. In this Letter of Interest, we discuss two properties that can be built into ML-based algorithms that we believe will be critical features of ML for theory in the coming decade: provable exactness and explicitly encoded symmetries. Early results in these directions suggest that ML will be a promising avenue to improve and accelerate calculations in the computationally demanding context of numerical lattice quantum field theory (LQFT), and in particular its application to quantum chromodynamics (LQCD). We highlight the example of ML-based samplers for LQFTs, where ML methods may be fruitfully applied without compromising systematic control of uncertainties and which are enabled by symmetries built into the models.

Configuration generation: In numerical LQFT, we evaluate the lattice-regulated path integral numerically by phrasing the problem as sampling a probability distribution $p = \exp[-S_E]/Z$ defined by the Euclidean-time lattice action S_E. Importance sampling algorithms, typically Markov-chain Monte Carlo (MCMC) methods, allow us to sample these distributions with asymptotic correctness (in the limit of large sample sizes), yielding provably unbiased results with controlled uncertainties. The modern LQCD program has very successfully employed the Hybrid Monte Carlo (HMC) algorithm for this task but, in limits of physical interest, HMC suffers from poor scaling due to critical slowing down and (exponentially) slow tunneling between vacua (e.g. topological sectors, center sectors in deconfined phases of pure gauge theories). It may be possible to replace or augment HMC with ML-based algorithms to avoid or reduce these scaling problems.

ML-accelerated updating: One option is to replace some or all of the HMC updates used to construct Markov chains with some computationally cheaper and/or faster-mixing ML-based updater (constructed with an accept/reject step to maintain correctness). Forward evaluation of ML models can be made inexpensive and parallelizable, so obtaining a practical performance advantage is potentially easier than beating HMC mixing times on a per-step basis. Hybrid algorithms may also be efficient (i.e., replacing only some HMC steps with ML-based updates, in rough analogy to the use of overrelaxation to augment Heat Bath in sampling pure gauge theories). While early demonstrations of ML-based update algorithms [1–6] suggest that ML-based methods may be able to outperform HMC, applications which circumvent HMC's scaling issues have not yet been conclusively demonstrated. However, update-based algorithms may be more conservative than necessary, as ML methods may make a radically different approach possible: direct sampling of the probability distribution.

Direct samplers: Recent works have demonstrated that it is possible to construct direct samplers for lattice field theories using ML methods [7–12]. In this approach, one specifies and optimizes a variational ansatz to obtain an approximate direct sampler that generates independent samples from a model distribution similar to the target one $\tilde{p} \approx p \propto \exp[-S_E]$. If it is tractable to compute the probability $\tilde{p}(U)$ of drawing each sample U, one can sample the target distribution p by either reweighting or constructing a Markov chain via the Independence Metropolis algorithm [7]. So long as $\tilde{p}$ is a sufficiently good approximation of p that reweighting factors are close to one or the Metropolis accept rate is not too low, this approach can be used to efficiently evaluate path integrals with asymptotically correct statistics (given that ergodicity is guaranteed, i.e. $\tilde{p}$ has support everywhere p does).

Normalizing flows, a particular class of generative ML model, are particularly well-suited to this task and have been used to construct direct samplers for two-dimensional field theories, including real scalar field the-

[*] dhackett@mit.edu
[†] phiala@mit.edu

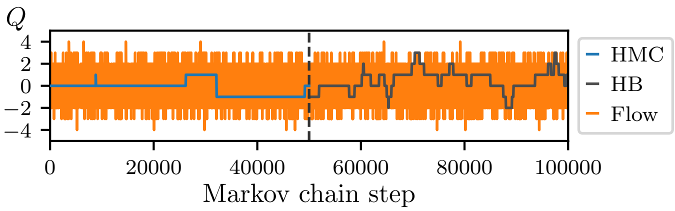

FIG. 1. Standard approaches (HMC and Heat Bath) to MCMC sampling for U(1) gauge theory explore the distribution of topological charge Q very slowly compared with the flow-based direct sampler. Results are shown for coupling $\beta = 7$ on a 16×16 lattice. The first (second) half of the Markov chain history is displayed for HMC (HB). Figure reproduced from [8].

ory [7, 12] and U(1) [8] and SU(N) gauge theories [9]. A normalizing flow takes samples drawn from some easily sampled prior distribution r (such as independent Gaussians on each lattice site for scalar field theory, or the uniform distribution over the Haar measure for each gauge link for gauge theories) and then performs a sequence of invertible changes-of-variables which "flow" the prior distribution to a more complicated target, $\tilde{p}$. These changes-of-variables are engineered to be invertible and have simple Jacobians so that computation of $\tilde{p}(U)$ is tractable. By parametrizing these transformations with neural networks, one obtains a trainable, expressive ansatz for an approximate direct sampler. Normalizing flows can be "self-trained" simply by drawing samples from the current model and then using them to estimate some optimizable metric of how much p and $\tilde{p}$ differ (e.g. the Kullback-Leibler divergence); this avoids the need for an expensive HMC-generated training dataset or the use of adversarial methods.

Direct samplers offer significant advantages over update-based samplers like HMC. Formally, they may be able to circumvent scaling problems experienced by updaters: in [7] it was demonstrated that flow-based direct sampling of real scalar field theory does not suffer from critical slowing down, while [8] demonstrated that flow-based direct samplers for U(1) have an asymptotic scaling advantage over HMC and Heat Bath in regimes where those methods suffer from topological freezing (see Fig. 1). Direct samplers may also offer novel ways to access physics, like thermodynamic equation-of-state observables in LQFTs, that are difficult to probe using update-based samplers [11, 12]. Practically, unlike update-based samplers, generating an ensemble of configurations using direct samplers is embarrassingly parallel: independent instances of the sampler may generate configurations without communicating, after which the configurations can be gathered and composed into a single Markov chain as a postprocessing step.

The benefits of direct sampling come at the cost of up-front training of the model, which may be significant. However, early results [9] suggest that retraining a model trained to sample one set of physical parameters can rapidly produce efficient samplers for nearby sets of parameters, allowing the cost of training a single model from scratch to be amortized.

Encoding symmetries in ML models:

When ML models are applied to physics problems, the underlying symmetries of the physical system are reflected in a constrained relationship between the input and output of the model. Any ML model which is not constructed to explicitly respect the physical symmetries of the problem will necessarily have to learn a (typically highly non-trivial) set of constraints on the model parameters that enforce invariance or equivariance with respect to those symmetries. This is computationally wasteful in the best case but, more problematically, if these constraints are too complicated or impossible to satisfy in the model architecture, it can be difficult or impossible to train the model to an acceptable solution. For models applied to highly symmetric systems like lattice gauge theories, training without explicitly encoding the symmetries of the problem is practically infeasible.

Some symmetries can be encoded using standard architectures developed for other applications: when applied to a spacetime lattice, convolutional neural nets (CNNs) naturally encode translational equivariance, and in some cases a few additional by-hand parameter restrictions can make these models equivariant under the hypercubic symmetry group of the lattice. Beyond massively reducing the number of parameters versus using a fully-connected architecture, encoding translational invariance using CNNs results in models that can be applied to different volumes. This yields an important practical advantage in that models can be trained at one (small) volume and then cheaply applied to another (up to whatever retraining of the model may be necessary to account for the change in finite-volume effects).

Other physical symmetries, like gauge invariance, require non-standard architectures. A framework to construct *gauge-equivariant flows* has been worked out for U(1) and SU(N) gauge symmetries and applied to construct flow-based direct samplers for both [8, 9]. Flows are versatile and straightforwardly applicable outside the original application of direct sampling; for example, they could be used in ML-based updaters. However, this framework will need to be extended to different model architectures to address problems for which other ML approaches are more natural; for example, flows map from configuration to configuration, and so are ill-suited for information-lossy tasks like regression on observables.

Infrastructure: ML methods generically require a large up-front training cost, but this cost does not have to be paid for every study: the heavily symmetry-constrained models likely to be successful in LQCD applications have relatively few independent parameters to store, and so are easy to distribute. As ML applications to LQCD mature, trained models and the software required to use them should be considered a common good to be distributed publicly.

It will be important for the LQCD community to encourage the development of software and hardware optimized for the specific ML tasks important to us (especially given that the relevant hardware optimizations may translate to better performance for classical LQCD computations). For example, in contrast to typical high-throughput ML tasks, training (and possibly evaluating) lattice-relevant ML models is likely to require tightly-networked computing resources. Additionally, fast high-dimensional (4D, at least) convolutions are particularly important, as well as sparse convolutions and convolutions with parameter restrictions. Further investigation is warranted into what resources AI for theory will require in the coming decade.

[1] L. Wang, Exploring cluster Monte Carlo updates with Boltzmann machines, Phys. Rev. E **96**, 051301 (2017).

[2] L. Huang and L. Wang, Accelerated Monte Carlo simulations with restricted Boltzmann machines, Physical Review B **95**, (2017).

[3] J. Song, S. Zhao, and S. Ermon, A-NICE-MC: Adversarial training for MCMC, in *Advances in Neural Information Processing Systems* (2017) pp. 5140–5150.

[4] D. Levy, M. D. Hoffman, and J. Sohl-Dickstein, Generalizing Hamiltonian Monte Carlo with neural networks, (2017), arXiv:1711.09268.

[5] J. M. Pawlowski and J. M.-Y. Urban, Reducing autocorrelation times in lattice simulations with generative adversarial networks, Machine Learning: Science and Technology 10.1088/2632-2153/abae73 (2020).

[6] S.-H. Li and L. Wang, Neural Network Renormalization Group, Phys. Rev. Lett. **121**, 260601 (2018).

[7] M. Albergo, G. Kanwar, and P. Shanahan, Flow-based generative models for Markov chain Monte Carlo in lattice field theory, Phys. Rev. D **100**, 034515 (2019), arXiv:1904.12072 [hep-lat].

[8] 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, (2020), arXiv:2003.06413 [hep-lat].

[9] 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, (2020), arXiv:2008.05456 [hep-lat].

[10] D. Wu, L. Wang, and P. Zhang, Solving Statistical Mechanics Using Variational Autoregressive Networks, Phys. Rev. Lett. **122**, 080602 (2019).

[11] 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), arXiv:1910.13496 [cond-mat.stat-mech].

[12] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, On Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models, (2020), arXiv:2007.07115 [hep-lat].