PROCEEDINGS OF SCIENCE

Multiscale Normalizing Flows for Gauge Theories

Ryan Abbott,[a,b,*] **Michael S. Albergo,**[c] **Denis Boyda,**[a,b] **Daniel C. Hackett,**[f,a,b] **Gurtej Kanwar,**[a,b,d] **Fernando Romero-López,**[a,b] **Phiala E. Shanahan**[a,b] **and Julian M. Urban**[a,b,e]

[a] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b] *The NSF AI Institute for Artificial Intelligence and Fundamental Interactions*
[c] *Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[d] *Albert Einstein Center, Institute for Theoretical Physics, University of Bern, 3012 Bern, Switzerland*
[e] *Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany*
[f] *Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.*

E-mail: rabbott@mit.edu

Scale separation is an important physical principle that has previously enabled algorithmic advances such as multigrid solvers. Previous work on normalizing flows has been able to utilize scale separation in the context of scalar field theories, but the principle has been largely unexploited in the context of gauge theories. This work gives an overview of a new method for generating gauge fields using hierarchical normalizing flow models. This method builds gauge fields from the outside in, allowing different parts of the model to focus on different scales of the problem. Numerical results are presented for $U(1)$ and $SU(3)$ gauge theories in 2, 3, and 4 spacetime dimensions.

The 40th International Symposium on Lattice Field Theory (Lattice 2023)
July 31st - August 4th, 2023
Fermi National Accelerator Laboratory

*Speaker

PoS(LATTICE2023)035

1. Introduction

Normalizing flows [1–3] are a novel machine-learning based tool for sampling which has shown promise in the context of lattice field theory for alleviating or eliminating problems such as critical slowing down and topological freezing [4–6]. Applications of normalizing flows to lattice field theories have seen great progress in recent years [7, 8], with demonstrations involving Abelian and non-Abelian gauge theories in 2, 3, and 4 dimensions, with and without fermions, including preliminary work on QCD [9]. In addition recent efforts in scaling from demonstrations in toy lattice volumes towards physically relevant theories have seen some success [10, 11].

One physical principle which has yet to be fully integrated into normalizing flow architectures for field theories is that of *scale separation*. Scale separation is the principle that physical processes and systems can often be decomposed into separate processes occurring at differing energy and length scales. Properly utilizing scale separation is key element of calculations in many physical domains and in the context of lattice field theory has lead to the development of many useful algorithms such as multigrid methods [12, 13]. Previous machine-learning based work has explored the use of scale separation in scalar field theories [14–17], 2D $U(1)$ gauge theory [18], and in the context of linear preconditioners [19, 20], but not in the context of non-Abelian gauge theories.

This work provides a construction of a new class of hierarchically-constructed models referred to as *multiscale models*. These models sample gauge fields by starting with coarse degrees of freedom, and proceeding to add successively finer and finer degrees of freedom. This structure allows these models to operate separately on the UV and IR degrees of freedom, enabling more expressive models that are able to take advantage of the physical structure of the theory.

2. Normalizing Flows

Normalizing flows are a method from machine learning for constructing an expressive, learned change of variables [1, 2, 21]. This change of variables takes the form of a parametrized bijective map f_θ which can be used to transform a density $r(z)$ (typically referred to as the "prior" density) into a new "model" density $q_\theta(U)$ via

$$q_\theta(U) = r(f_\theta^{-1}(U)) \left| \det \frac{\partial f_\theta^{-1}}{\partial U} \right| . \tag{1}$$

Given a desired target density $p(U) = \frac{1}{Z} e^{-S(U)}$, the model density can be trained to replicate the target density by minimizing the reverse Kullback-Leibler (KL) divergence,[1]

$$\begin{aligned} \mathrm{KL}(q||p) &= \mathbb{E}_{U\sim q(U)}[\log q(U) - \log p(U)] &(2)\\ &= \mathbb{E}_{U\sim q(U)}[\log q(U) + S(U)] + \log Z \, . &(3)\end{aligned}$$

Once a model has been trained, the target density $p(U)$ can be sampled via methods such as independence Metropolis [21–25], or direct reweighting from the model density. The efficiency of

[1] Note that the $\log Z$ term in (Eq. (3)) is a constant independent of U, and hence determining this (typically intractable) term is not required for training.

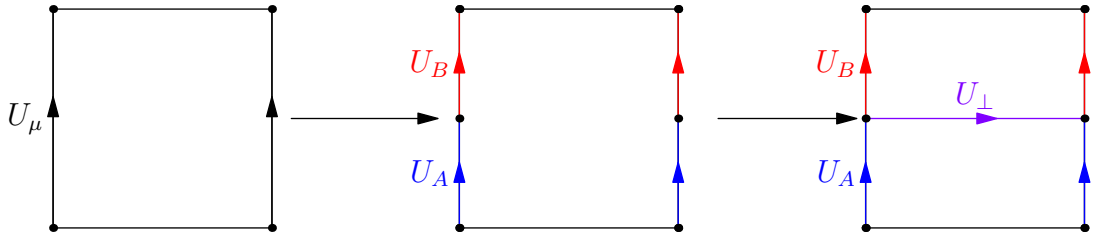

Figure 1: Schematic of the procedure for doubling layers, which serve to double the number of lattice sites in a particular spacetime direction

such methods can be estimated using the effective sample size (ESS) statistic, defined by [26, 27]

$$\text{ESS} = \frac{\mathbb{E}_{U\sim q(U)}[w(U)]^2}{\mathbb{E}_{U\sim q(U)}[w(U)^2]}, \tag{4}$$

where $w(U) = \frac{e^{-S(U)}}{q(U)}$ is the (unnormalized) reweighting factor associated to U. The ESS gives an approximation of how many independent target samples are obtained per model sample, and the ESS is always bounded between 0 and 1.

3. Multiscale Models in Two Dimensions

Similar to a typical normalizing flow model, multiscale models begin by sampling from a prior density, referred to as the *coarse prior*. Unlike most normalizing flow-based models, however, samples from the coarse prior do *not* live in the same space as target samples; instead the samples from the coarse prior are gauge fields living on a coarser lattice, possibly as coarse as a single lattice site. The coarse prior can be any tractable density that can be sampled; for instance, the coarse prior might be taken as a gauge theory with a coarser lattice spacing than the target density, or the coarse gauge links could be sampled from the uniform (Haar) distribution.

Once the coarse degrees of freedom have been sampled, new degrees of freedom are added via successive *doubling layers*. A doubling layer takes as input a set of "coarse" gauge links U_ν^{coarse} living on a lattice Λ along with a *doubling direction* μ, and outputs a new set of "fine" gauge links U_ν^{fine} arranged on a lattice Λ' with twice as many sites appearing along the $\hat{\mu}$ direction. Conceptually, the lattice Λ' is considered to be a refinement of Λ, meaning that Λ and Λ' are both considered subsets of the same underlying space with $\Lambda \subset \Lambda'$. This means that applying a doubling layer halves the lattice spacing in the $\hat{\mu}$-direction, leading to an anisotropic lattice spacing a_ν. By applying successive doubling layers with different doubling directions, the underlying lattice spacing can be made arbitrarily small in every direction.

Doubling layers can be constructed in arbitrary dimensions; however, it is simpler to first construct 2-dimensional doubling layers, and then later generalize the layers to accommodate arbitrary dimensions (see Sec. 4). A 2-dimensional doubling layer with doubling direction μ can be separated into two steps, as illustrated in Fig. 1. First each gauge link oriented along the $\hat{\mu}$-direction is split into two gauge links U_A and U_B, subject to the condition

$$U_\mu(x) = U_A(x)U_B(x + a_\mu\hat{\mu}/2)\ . \tag{5}$$

This can be achieved by sampling U_B from the uniform (Haar) distribution on the given gauge group, and then defining U_A from Eq. (5). Note that at this point no new physical information has been added, since the added gauge links U_A and U_B do not create any new loops. All physical information is added in the second part of the doubling layer, wherein a new set of perpendicular links $U_\perp$ is sampled from a chosen distribution. For instance, one possible distribution for sampling $U_\perp$ would be the heatbath-like distribution

$$p_{HB}(U_\perp) \propto \exp\left\{\tilde{\beta}\sum_x \mathrm{Re}\,\mathrm{Tr}[U_\perp(x+a_\mu\hat{\mu}/2)(S_A+S_B)(x+a_\mu\hat{\mu}/2)]\right\} \tag{6}$$

where $\tilde{\beta}\in\mathbb{R}$ parameterizes the probability distribution, and S_A and S_B are the staples defined by

$$S_A(x+a_\mu\hat{\mu}/2) = U_A^\dagger(x+a_\nu\hat{\nu})U_\nu^\dagger(x)U_A(x) \tag{7}$$

$$S_B(x+a_\mu\hat{\mu}/2) = U_B(x+a_\mu\hat{\mu}/2+a_\nu\hat{\nu})U_\nu^\dagger(x+a_\mu\hat{\mu})U_B^\dagger(x+a_\mu\hat{\mu}/2) \tag{8}$$

and ν is the direction orthogonal to μ. Alternatively $U_\perp$ could be a sampled from a normalizing flow model conditioned on S_A and S_B, as will be discussed in Sec. 3.1.

Once $U_\perp$ has been generated, the final fine lattice U_ν^{fine} can be assembled via

$$U_\nu^{\text{fine}}(x) = \begin{cases} U_\nu^{\text{coarse}}(x) & x_\mu/a'_\mu \text{ even and } \nu\neq\mu \\ U_\perp(x) & x_\mu/a'_\mu \text{ odd and } \nu\neq\mu \\ U_A(x) & x_\mu/a'_\mu \text{ even and } \nu=\mu \\ U_B(x) & x_\mu/a'_\mu \text{ odd and } \nu=\mu \end{cases} \tag{9}$$

where $a'_\mu = a_\mu/2$ indicates the lattice spacing of the fine lattice Λ' in the direction μ. This completes the construction of the doubling layers in 2 dimensions; all that remains is to compute the density of the resulting field U_ν^{fine}, which can be accomplished using the factorization

$$q(U_\nu^{\text{fine}}) = q(U_\perp \mid U_A, U_B, U_\nu^{\text{coarse}})q(U_B \mid U_\nu^{\text{coarse}})q(U_\nu^{\text{coarse}}). \tag{10}$$

Here $q(U_\perp \mid U_A, U_B, U_\nu^{\text{coarse}})$ is the model density of a staple-conditional model, which will be discussed in Sec. 3.1, and U_B is sampled from the Haar distribution, which has density $q(U_B \mid U_\nu^{\text{coarse}}) = 1$. Note here that U_A is determined by U_B and U_ν^{coarse}, and hence does not require an additional term in Eq. (10).

3.1 Staple-Conditional Models

In order to sample $U_\perp$ (see Fig. 1), it is desirable to have a class of expressive, conditional models that can incorporate as much gauge-equivariant information from the local neighborhood of $U_\perp$ as possible. One natural piece of gauge-equivariant conditional information are the staples S_A and S_B defined in Eqs. (7) and (8). More generally, any composition of gauge links passing from the endpoint to the starting point of $U_\perp$ has the same gauge transformation properties as S_A and S_B, and hence can be considered as an abstract staple. Attempting to integrate this information into the sampling of $U_\perp$ then leads to the general notion of staple-conditional models, which are models capable of sampling a single gauge matrix U, conditioned on a particular number of

"staples" $S_1, \ldots, S_n$. Note that this architecture requires that each gauge link within $U_\perp$ is sampled independently conditioned on the staples. Staple-conditional models can be based on any tractable distribution; in this work we will primarily consider staple-conditional models based on normalizing flows.

There are two components to a normalizing flow-based staple-conditional model: a prior (or base) distribution $r(z \mid S_1, \ldots, S_n)$, $z \in G$, and a conditional normalizing flow $f(z \mid S_1, \ldots, S_n)$. These two pieces can then be combined to define a model density $q(U \mid S_1, \ldots, S_n)$:

$$q(U \mid S_1, \ldots, S_n) = r(f^{-1}(U) \mid S_1, \ldots, S_n) \left| \det \frac{\partial f^{-1}(U)}{\partial U} \right| . \tag{11}$$

The prior distribution can be any tractable distribution, such as the uniform (Haar) distribution, or the heatbath-type distribution defined by Eq. (6) in the case of a $U(1)$ gauge theory.[2] For the conditional normalizing flow, the tools and methods developed for constructing gauge-equivariant flow models can be adapted to the problem with slight modifications. For instance, in the $\mathrm{SU}(N)$ case both spectral and residual flows [28] provide building blocks that can be composed to create expressive conditional transformations. For the $\mathrm{SU}(N)$ staple-conditional flows involved in this work, a spectral flow is used, acting on an effective active loop P defined by

$$P = \mathrm{Proj}_{\mathrm{SU}(N)} \left\{ U \sum_{i=1}^{n} \alpha_i S_i \right\} \tag{12}$$

where $\{\alpha_i\}_{i=1}^n$ are learned coefficients, and $\mathrm{Proj}_{\mathrm{SU}(N)}$ is a projection onto $\mathrm{SU}(N)$, accomplished via polar decomposition:

$$\mathrm{Proj}_{\mathrm{SU}(N)}(M) = \frac{M(M^\dagger M)^{-1/2}}{\det \left[M(M^\dagger M)^{-1/2} \right]^{1/N}}. \tag{13}$$

The spectral flow acting on P produces a new, transformed value $P' \in \mathrm{SU}(N)$. This transformation can then be pushed back to the gauge matrix U to produce a new matrix U' via

$$U' = P' P^\dagger U. \tag{14}$$

This procedure can then be iterated with different values of α_i and different spectral flows in order to construct an expressive, learnable transformation. Alternatively, architectures based on residual flows [28] may also be used as coupling layers, either in place of or alongside spectral flows. Although the numerical demonstrations in Secs. 3.2 and 4.1 utilize only spectral flow based staple conditional models, simiar models based on residual flows achieve similar performance.

For $U(1)$ gauge fields, staple-conditional models can be constructed in a similar manner as the $\mathrm{SU}(N)$ staple-conditional models. The numeric demonstrations in Sec. 3.2 are constructed using the heatbath-like prior defined in Eq. (6) combined with a sequence of coupling layers based on circular rational quadratic splines. [29] The coupling layers in the $U(1)$ case have the same structure

[2] Note that here "tractable" includes the requirement of computing the normalized density, which is not possible in the case of a SU(3) heatbath due to the lack of an analytic formula for the normalizing constant for the distribution Eq. (6). Computing the normalizing constant as well as its gradients numerically is theoretically possible, but the computational cost of doing so would be impractically large for the models considered in Secs. 3.2 and 4.1.

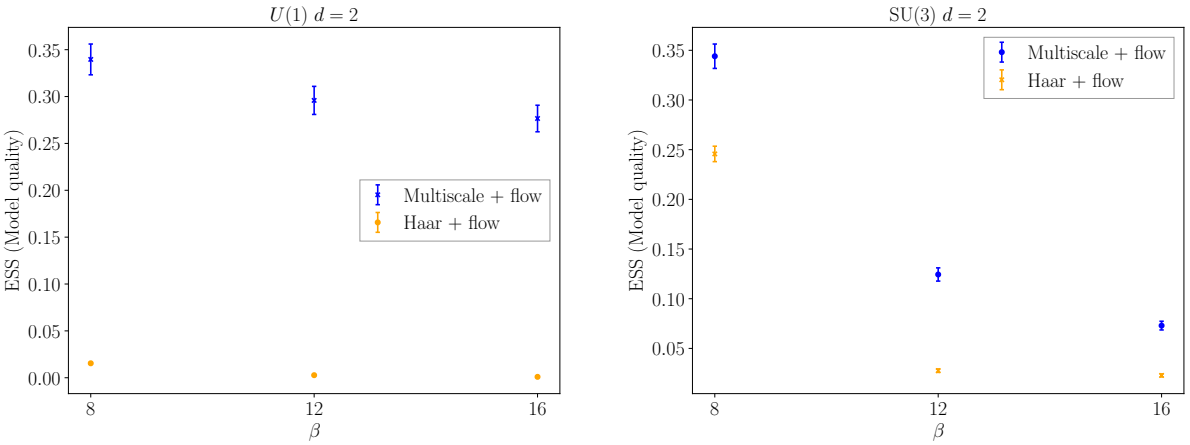

Figure 2: Effective sample size (ESS) for 2-dimensional multiscale models (higher is better). Blue data points indicate the performance of the multiscale model combined with a small fine-lattice flow, while the orange data points represent the fine-lattice flow alone.

as the SU(N) staple-conditional flows, except that the projection in Eq. (12) is replaced with a normalization step

$$\text{Proj}_{U(1)}(z) = \frac{z}{|z|}. \tag{15}$$

3.2 Numeric Results

Numerical demonstrations of multiscale models are investigated for both $U(1)$ and SU(3) gauge theories in 2 dimensions; results are shown in Fig. 2. All models start from a Haar uniform coarse prior on a 2×2 lattice, and target an 8×8 fine lattice. In all cases the multiscale model has been integrated into the overall model as a prior, with an additional flow applied at the finest scale after all of the degrees of freedom have been generated. This is necessary due to the fact that the multiscale models as constructed only attempt to capture a subset of possible correlations in the output gauge configurations, and hence a full fine-lattice flow is still required for maximum expressivity. For the $U(1)$ models the fine lattice flow consists of 48 gauge-equivariant coupling layers utilizing rational quadratic splines [29], while the find lattice flows for the SU(3) models consist of a single iteration of direction and location updates, for a total of $8 + 4 = 12$ layers in 2 dimensions [28].

In both the $U(1)$ and SU(3) cases the multiscale models show significant improvement over equivalent models without the multiscale component used as a prior. In the $U(1)$ case the quality of the multiscale models is effectively independent of β, while the non-multiscale models have an effective sample size consistent with 0. Meanwhile in the SU(3) case the model quality does decline with β, but more slowly for the multiscale models, which maintain $\sim 10\%$ ESS even for $\beta = 16$.

4. Multiscale models in higher dimensions

In greater than two dimensions, models can be constructed in the same manner as in two dimensions, with a few additional complications. As in two dimensions, higher-dimensional multiscale models are also constructed as a sequence of doubling layers, each doubling the lattice

extent along a particular direction. The primary complication in adapting the 2-dimensional models to higher dimensions occurs during the generation of the new gauge links $U_\perp$ that are orthogonal to the doubling direction. In d-dimensions, the role that $U_\perp$ plays in 2-dimensional models is replaced by a $(d-1)$-dimensional slice of the lattice, as illustrated in Fig. 3. This slice contains both UV and IR degrees of freedom, and hence more care is required handling this slice than in the 2-dimensional case.

One viable solution for sampling the $(d-1)$-dimensional slice needed in a d-dimensional doubling layer is to utilize a $(d-1)$-dimensional multiscale model. This gives the models a recursive structure, with each multiscale model depending recursively on several lower-dimensional models, terminating at the base case of a 2-dimensional model, for which the previously described 2-dimensional multiscale models are sufficient. This gives the broad structure of the model, which is subject to a handful of complications.

The first complication arises in the conditioning of the models. In order to ensure that the lower dimensional models can generate correlations in the perpendicular field $U_\perp$, models need to have access to sufficient gauge-equivariant information, including information not present within the slice that the lower-dimensional model operates within. In order to pass such information, the multsicale models as described above can be modified by adding additional conditioning in the form of staples constructed out of gauge links present in the higher-dimensional models. This means that, for instance, a 2-dimensional doubling layer contained within a 3-dimensional model will also receive as input staples which reach outside of the 2-dimensional plane. These staples can be passed to the staple-conditional models present within the doubling layers, allowing the staple-conditional models to properly account for out-of-plane information.

The second complication of the higher dimensional doubling layers arises during the first part of the doubling layers, wherein the gauge links oriented along the doubling direction are split in half (see Eq. (6)). In 2-dimensional doubling layers this step does not add any new physical information, but this is not necessarily true if the 2-dimensional doubling layer is a subcomponent of a higher-dimensional model. In particular, if there already exist paths of links connecting the two endpoints of U_B, then U_B could be sampled from a staple-conditional model conditioned on these paths, adding new physical information to the sampling step. Furthermore, it is also possible to sample U_B at some sites before others, allowing the later values of U_B to be conditioned on previous values. In practice this requires careful bookkeeping to route the appropriate staples into a new staple-conditional model for sampling U_B.

4.1 Numeric Results

Numeric results for 4-dimensional models for both $U(1)$ and $\mathrm{SU}(3)$ gauge theories are shown in Fig. 4, in terms of the KL divergence as defined in Eq. (3). Both models start from a Haar uniform coarse prior of size $L/a_{\text{coarse}} = 2$, and target a periodic fine lattice of size $L/a_{\text{fine}} = 4$. The constant $\log Z$ in Eq. (3) is estimated from the best model available at the given β, and the same value of $\log Z$ is used for every model at the same value of β. As with the 2-dimensional models, the multiscale models here are integrated into a larger model as a prior, with a small additional fine-lattice flow. For the $U(1)$ models the fine lattice flow consists of 48 gauge-equivariant coupling layers utilizing rational quadratic splines [29], while the fine lattice flows for the $\mathrm{SU}(3)$ models

Algorithm 1 Generic d-dimensional doubling layers

Input: Coarse gauge field U_ν^{coarse}, doubling direction μ, higher-dimensional staples S
Sample U_B from staple-conditional model
Compute $U_A = U_\mu U_B^\dagger$
Add S_A, S_B from Eqs. (7) and (8) to S
if $d = 2$ **then**
    Sample $U_\perp$ from staple-conditional model
else
    Sample $(d-1)$-dimensional slices $U_\perp$ from $(d-1)$-dimensional multiscale model
end if
Combine $(U_\nu^{\text{coarse}}, U_A, U_B, U_\perp)$ via Eq. (9)
Output: fine gauge field U_ν^{fine}

Algorithm 2 d-dimensional multiscale model

Input: Higher-dimensional staples S
Sample gauge field U_ν at coarsest scale
while U_ν has not reached finest scale **do**
    Choose doubling direction μ
    $U_\nu \leftarrow \text{DoublingLayer}(U_\nu, \mu, S)$
end while

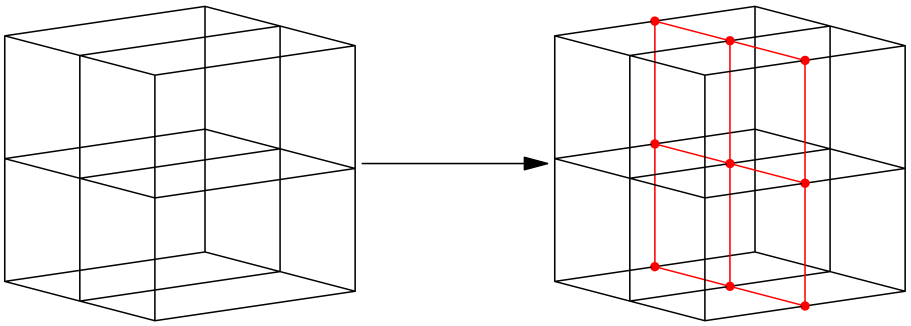

Figure 3: Illustration of a 3-dimensional doubling layer; red points indicate the new lattice sites in the doubled lattice, and red lines indicate the added gauge links.

consist of a single iteration of direction and location updates, for a total of 48 + 16 = 64 layers in 4 dimensions. [28]

In all cases the KL divergence of the multiscale models is smaller than the KL divergence of the fine-lattice flow alone. The degree of improvement is slight at smaller values of β, but increases for larger values of β, indicting that the multiscale models have increased expressivity at higher β relative to the fine-lattice flows utilized in this work. One plausible explanation for this difference is that the higher β distributions tend to have stronger correlations at longer distances. Multiscale models can build such correlations directly when acting at the coarser scales, whereas the full fine-lattice flow needs to build in the correlation structure starting from the finest scale upwards,

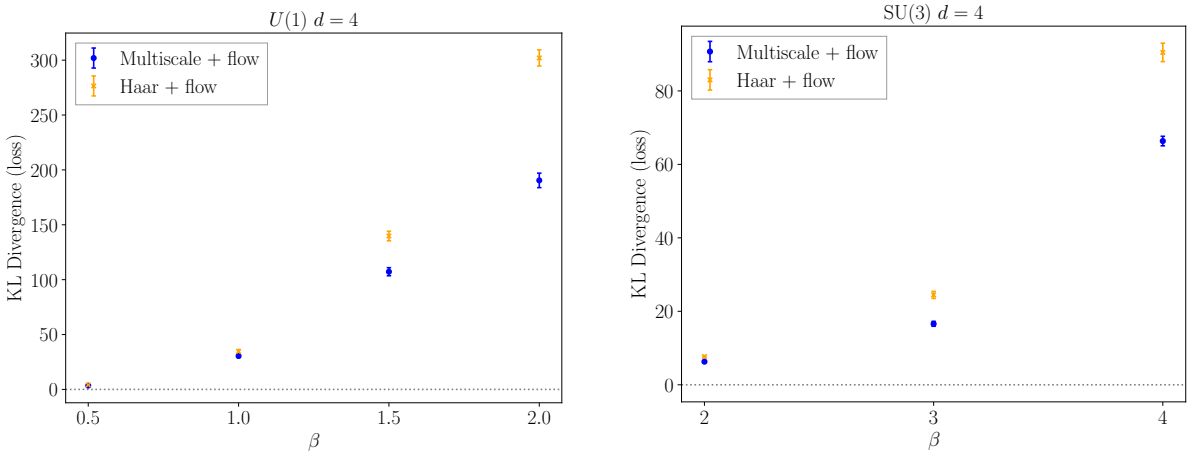

Figure 4: KL divergence (see Eq. (3)) for 4-dimensional multiscale models (lower is better). Blue data points indicate the performance of the multiscale model combined with a small fine-lattice flow, while the orange data points represent the fine-lattice flow alone.

which is particularly difficult for the relatively small fine-lattice flows utilized in this work.

5. Future work

There are several possible avenues for future improvements for the models presented here.

- The fine-lattice flows utilized in the numeric results here are intentionally small in order to isolate the effects of the multiscale models. Future studies could combine the multiscale models with larger, more expressive fine-lattice flows in order to obtain a maximally expressive combination.
- For the sake of simplicity, the multiscale models presented here assume independence of links sampled at a given scale after conditioning on the coarser scales. Future models could relax this assumption, which would also have the benefit of making the multiscale models universal density approximators, meaning that the models could (in principle) approximate any density to an arbitrary degree of precision, given a sufficient number of parameters.
- The present work has focused on the method of direct sampling as a benchmarking task for these new models; however, this is not necessarily the most efficient method for utilizing multiscale models. Instead a hybrid approach that combines more traditional MCMC methods such as HMC with flow-based methods could provide a better avenue for near-term physics applications (see, for instance, the recent work in Refs. [10, 11]).

6. Conclusion

Multiscale models show great potential for improving near- and far-term normalizing flow capabilities in the context of lattice field theory. By operation on the UV and IR degrees of freedom separately, these models are able to exploit scale separation in order to more effectively replicate

the desired gauge field distribution. Though more work is needed to extend these models and integrate them more fully with other methods, this represents a promising step in understanding and implementing normalizing flows for gauge generation, and more broadly improving the efficiency of gauge generation as a whole.

Acknowledgements

We thank Aleksandar Botev, Kyle Cranmer, Alexander G. D. G. Matthews, Sébastien Racanière, Ali Razavi, and Danilo J. Rezende for useful discussions and valuable contributions to the early stages of this work. RA, DCH, FRL, PES, and JMU are supported in part by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under grant Contract Number DE-SC0011090. PES is additionally supported by the U.S. DOE Early Career Award DE-SC0021006, by a NEC research award, and by the Carl G and Shirley Sontheimer Research Fund. FRL acknowledges support by the Mauricio and Carlota Botton Fellowship. GK was supported by the Swiss National Science Foundation (SNSF) under grant 200020_200424. This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This work is supported by the U.S. National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/) and is associated with an ALCF Aurora Early Science Program project, and used resources of the Argonne Leadership Computing Facility which is a DOE Office of Science User Facility supported under Contract DEAC02-06CH11357. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center [30] for providing HPC resources that have contributed to the research results reported within this paper. Numerical experiments and data analysis used PyTorch [31], JAX [32], Haiku [33], Horovod [34], NumPy [35], and SciPy [36]. Figures were produced using matplotlib [37].

References

[1] D.J. Rezende and S. Mohamed, *Variational inference with normalizing flows*, 1505.05770.

[2] L. Dinh, J. Sohl-Dickstein and S. Bengio, *Density estimation using real nvp*, 1605.08803.

[3] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, *Journal of Machine Learning Research* **22** (2021) 1.

[4] B. Alles, G. Boyd, M. D'Elia, A. Di Giacomo and E. Vicari, *Hybrid Monte Carlo and topological modes of full QCD*, *Phys. Lett. B* **389** (1996) 107 [hep-lat/9607049].

[5] L. Del Debbio, G.M. Manca and E. Vicari, *Critical slowing down of topological modes*, *Phys. Lett. B* **594** (2004) 315 [hep-lat/0403001].

[6] ALPHA collaboration, *Critical slowing down and error analysis in lattice QCD simulations*, *Nucl. Phys. B* **845** (2011) 93 [1009.5228].

[7] D. Boyda et al., *Applications of Machine Learning to Lattice Quantum Field Theory*, in *2022 Snowmass Summer Study*, 2, 2022 [2202.05838].

[8] K. Cranmer, G. Kanwar, S. Racanière, D.J. Rezende and P.E. Shanahan, *Advances in machine-learning-based sampling motivated by lattice quantum chromodynamics*, *Nature Rev. Phys.* **5** (2023) 526 [2309.01156].

[9] R. Abbott et al., *Sampling QCD field configurations with gauge-equivariant flow models*, in *39th International Symposium on Lattice Field Theory*, 8, 2022 [2208.03832].

[10] R. Abbott, A. Botev, D. Boyda, D.C. Hackett, G. Kanwar, S. Racanière et al., *Applications of flow models to the generation of correlated lattice QCD ensembles*, .

[11] R. Abbott, A. Botev, D. Boyda, D.C. Hackett, G. Kanwar, S. Racanière et al., *Practical applications of machine-learned flows on gauge fields*, .

[12] J. Brannick, R.C. Brower, M.A. Clark, J.C. Osborn and C. Rebbi, *Adaptive Multigrid Algorithm for Lattice QCD*, *Phys. Rev. Lett.* **100** (2008) 041601 [0707.4018].

[13] R. Babich, J. Brannick, R.C. Brower, M.A. Clark, T.A. Manteuffel, S.F. McCormick et al., *Adaptive multigrid algorithm for the lattice Wilson-Dirac operator*, *Phys. Rev. Lett.* **105** (2010) 201602 [1005.3043].

[14] T. Marchand, M. Ozawa, G. Biroli and S. Mallat, *Wavelet Conditional Renormalization Group*, 2207.04941.

[15] H.-Y. Hu, D. Wu, Y.-Z. You, B. Olshausen and Y. Chen, *RG-Flow: a hierarchical and explainable flow model based on renormalization group and sparse prior*, *Mach. Learn. Sci. Tech.* **3** (2022) 035009 [2010.00029].

[16] J.J. Yu, K.G. Derpanis and M.A. Brubaker, *Wavelet flow: Fast training of high resolution normalizing flows*, *Advances in Neural Information Processing Systems* **33** (2020) 6184.

[17] F. Guth, S. Coste, V. De Bortoli and S. Mallat, *Wavelet score-based generative modeling*, *Advances in Neural Information Processing Systems* **35** (2022) 478.

[18] N. Matsumoto, R.C. Brower and T. Izubuchi, *Decimation map in 2D for accelerating HMC*, in *40th International Symposium on Lattice Field Theory*, 12, 2023 [2312.04800].

[19] C. Lehner and T. Wettig, *Gauge-equivariant neural networks as preconditioners in lattice QCD*, 2302.05419.

[20] C. Lehner and T. Wettig, *Gauge-equivariant pooling layers for preconditioners in lattice QCD*, 2304.10438.

[21] 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** (2019) 034515 [1904.12072].

[22] F. Noé, S. Olsson, J. Köhler and H. Wu, *Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning*, *Science* **365** (2019) eaaw1147 [1812.01729].

[23] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, *Equation of state calculations by fast computing machines*, *J. Chem. Phys.* **21** (1953) 1087.

[24] W.K. Hastings, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, *Biometrika* **57** (1970) 97.

[25] L. Tierney, *Markov chains for exploring posterior distributions*, *the Annals of Statistics* (1994) 1701.

[26] A. Doucet, N. De Freitas, N.J. Gordon et al., *Sequential Monte Carlo methods in practice*, vol. 1, Springer (2001).

[27] J.S. Liu and J.S. Liu, *Monte Carlo strategies in scientific computing*, vol. 10, Springer (2001).

[28] R. Abbott et al., *Normalizing flows for lattice gauge theory in arbitrary space-time dimension*, 2305.02402.

[29] G. Kanwar, M.S. Albergo, D. Boyda, K. Cranmer, D.C. Hackett, S. Racanière et al., *Equivariant flow-based sampling for lattice gauge theory*, *Phys. Rev. Lett.* **125** (2020) 121601 [2003.06413].

[30] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor et al., *Interactive supercomputing on 40,000 cores for machine learning and data analysis*, *2018 IEEE High Performance extreme Computing Conference (HPEC)* (2018) 1 [1807.07814].

[31] A. Paszke et al., *Pytorch: An imperative style, high-performance deep learning library*, in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds., pp. 8024–8035, Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[32] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin et al., *JAX: composable transformations of Python+NumPy programs*, 2018.

[33] T. Hennigan, T. Cai, T. Norman and I. Babuschkin, *Haiku: Sonnet for JAX*, 2020.

[34] A. Sergeev and M. Del Balso, *Horovod: fast and easy distributed deep learning in TensorFlow*, 1802.05799.

[35] C.R. Harris, K.J. Millman, S.J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau et al., *Array programming with numpy*, *Nature* **585** (2020) 357.

[36] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau et al., *Scipy 1.0: fundamental algorithms for scientific computing in python*, *Nature methods* **17** (2020) 261.

[37] J.D. Hunter, *Matplotlib: A 2d graphics environment*, *Computing in Science & Engineering* **9** (2007) 90.