Scaling Up Machine Learning For Quantum Field Theory with Equivariant Continuous Flows

Pim de Haan[*]
Qualcomm AI Research,
Qualcomm Technologies Netherlands B.V.[†]
University of Amsterdam

Corrado Rainone[*]
Qualcomm AI Research,
Qualcomm Technologies Netherlands B.V.[†]

Miranda C. N. Cheng
Institute of Physics and
Korteweg-de Vries Institute for Mathematics,
University of Amsterdam, the Netherlands
Institute for Mathematics,
Academia Sinica, Taipei, Taiwan

Roberto Bondesan
Qualcomm AI Research,
Qualcomm Technologies Netherlands B.V.[†]

Abstract

We propose a continuous normalizing flow for sampling from the high-dimensional probability distributions of Quantum Field Theories in Physics. In contrast to the deep architectures used so far for this task, our proposal is based on a shallow design and incorporates the symmetries of the problem. We test our model on the ϕ^4 theory, showing that it systematically outperforms a realNVP baseline in sampling efficiency, with the difference between the two increasing for larger lattices. On the largest lattice we consider, of size 32×32, we improve a key metric, the effective sample size, from 1% to 66% w.r.t. the realNVP baseline.

1 Introduction and Related Work

Machine learning (ML) offers a novel tool which has the potential to outperform traditional computational methods in scientific applications, thanks to the ability of learning algorithms to improve automatically with more data and to adapt to the specific problem at hand. This has driven the fast adoption of ML in a variety of physical applications, ranging from quantum mechanics to molecular simulations to particle physics. See for instance [1, 2, 3] for reviews. In some cases, one can also show that ML methods are provably more efficient than traditional ones [4]. Central research challenges in

[*]Equal contribution
[†]Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

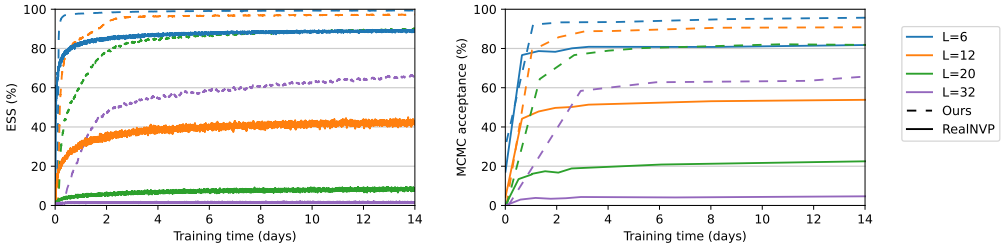

Figure 1: Effective Sample Size and MCMC acceptance rate for lattice size $L \times L$.

applying ML to physics include scaling up the ML models and interpretability. A general approach to these issues is to incorporate physical priors in the models, such as symmetries [2].

In this paper we study the application of ML to lattice field theories. Quantum field theories can be used to describe diverse physical phenomena, ranging from properties of materials to fundamental forces of nature. Some of these theories are weakly coupled and can be studied via perturbative methods (i.e. Feynman Diagrams), but others are not, with as a notable example Quantum ChromoDynamics (QCD) which is an important pillar of the standard model of particle physics. One of the major computational tools for these theories is to consider a lattice version of the theory, with the continuous space-time replaced by a discrete finite lattice. As is commonly done, we work in Euclidean space-time with no distinguished "time" direction. In this setup, the main task is to sample from a known Boltzmann distribution, which is usually done via Markov Chain Monte Carlo (MCMC) methods [5]. In all the applications, it is crucial that one should be able to extrapolate the lattice theories to the regime with a large number degrees of freedom, in order to access continuum physics or critical points. In many cases, however, the conventional MCMC methods suffer from the problem of Critical Slowing Down (CSD), namely that it takes a prohibitively long time to generate two independent distinct samples, which hampers robust extrapolations and renders the lattice field theory results unreliable.

Recently, a series of papers (see for example [6, 7, 8, 9, 10, 11]) has started to explore ML techniques, such as Normalizing Flows, applied to such sampling tasks. The idea is that if we can learn an invertible map that trivializes an interacting model to a free theory, we can easily sample the latter and push back the samples through the inverse map to obtain (proposed) samples from the original non-trivial distribution. We can then correct for the mistakes of the learning algorithm by a Metropolis-Hastings accept-reject rule, as is typically done in MCMC. In [6, 12] the authors use a real NVP normalizing flow [13] as the invertible map for the ϕ^4 theory distribution (defined below), and show that this technique can have better sampling efficiency than a pure MCMC method. Real NVP normalizing flows partition the lattice in two parts and at each layer modify only one part, which has the consequence that the symmetries of the square lattice are partially broken. Boltzmann generators [14] extended these ideas to particle systems, further studied in [15] using symmetric continuous flows. Refs. [9, 10] address the issue of incorporating gauge symmetries, and fermionic degrees of freedom, and [16] develops continuous flows for lattice QCD.

We summarize now the contributions of this paper:

- We extend [15] and develop continuous normalizing flows for lattice field theories that are fully equivariant under lattice symmetries as well as the internal $\phi \mapsto -\phi$ symmetry of the ϕ^4 model. Our flow is based on a shallow architecture which acts on handcrafted features.
- We train our model for the ϕ^4 theory and and for the 32×32 lattice we improve the effective sample size from 1% to 66% w.r.t. a real NVP baseline of similar size (Fig. 1).
- We study equivariance violations of real NVP models and contrast it with the exact equivariance of our flows.

2 Background

Samplers Based on Normalizing Flows. The Metropolis-Hastings algorithm is a popular and flexible method to sample from a target density p given a proposal distribution q [17]. It proposes a sample ϕ' from q and accepts it with probability $\rho = \min(1, q(\phi^{(i-1)})p(\phi')/p(\phi^{(i-1)})q(\phi'))$. CSD takes the form of long strings of rejections that lead to repeated samples in the chain, and therefore a reduction in sampling efficiency and quality of estimates. Conversely, when the proposal q is identical to the target p, all proposals get accepted, and efficiency reaches its theoretical maximum. The distribution q is usually painstakingly handcrafted with this goal in mind; the ML approach to this problem is conversely to *learn* q, for example using a Normalizing Flow [6]. In more detail, a normalizing flow is an invertible neural network f that maps a latent random variable z distributed according to an easy-to-sample distribution $r(z)$ to a random variable ϕ whose distribution q is the push-forward of r under f [13]. To match q with p, we can minimize the reverse KL divergence $\mathrm{KL}(q|p)$:

$$\mathrm{KL}(q|p) = \mathbb{E}_{\phi\sim q}[\log \tfrac{q(\phi)}{p(\phi)}] = \mathbb{E}_{z\sim\rho}[\log q(f^{-1}(z)) + S(f^{-1}(z))] + \log Z\,. \tag{1}$$

where we have used $-\log p(\phi) = S(\phi) + \log Z$. Since r is easy to sample, we can efficiently estimate Eq. (1) and minimize it w.r.t. the parameters of f. As a result, a flow-based sampler first

amortizes the cost of finding a good proposal distribution by training a normalizing flow f. The pushed forward distribution q under f is then used as proposal in Metropolis Hastings. See Appendix A.1 for algorithmic details.

In [6] it is shown that such a sampler achieves competitive results on small lattices for the ϕ^4 theory. Specifically, a real NVP flow, consisted of coupling layers of the form [13]

$$\phi_a = z_a\,, \quad \phi_b = z_b \odot s(z_a) + t(z_a)\,. \tag{2}$$

using neural networks s and t and $\odot$ denoting element-wise multiplication, is used in [6]. In the above, the input/output spaces are partitioned in two parts of size N_a, N_b, and this partition partially breaks the spatial symmetries and the network is not fully equivariant. See Appendix A.2 for more details on real NVP models.

In a continuous normalizing flow, we define an invertible mapping $f : \mathbb{R}^N \to \mathbb{R}^N$, $z \mapsto \phi$, as the flow under a neural ODE [18] for a fixed time T:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = g(x(t), t; \theta) \text{ with } x(t)|_{t=0} = z,\ x(t)|_{t=T} = \phi. \tag{3}$$

Here the vector field $g(x(t), t; \theta)$ is a neural network with weights θ. The log probability then follows another ODE [18]

$$\frac{\mathrm{d}\log p(x(t))}{\mathrm{d}t} = -(\nabla_x \cdot g)(x(t), t; \theta) \text{ with } p(x(t))|_{t=0} = r(z),\ p(x(t))|_{t=T} = q(\phi). \tag{4}$$

g is not constrained as in the real NVP case, which allows one to build in symmetries more easily. Following [15], if the the vector field g is equivariant, the resulting distribution on ϕ is invariant. We will show in the next section how to construct a g equivariant to the square lattice symmetries.

The ϕ^4 theory. Our goal is to improve the sampling performance for the so-called ϕ^4 theory. It is a relatively simple quantum field theory which nevertheless possesses non-trivial symmetry properties and an interesting phase transition, which make it an ideal testing ground for new computational ideas. In the case of ϕ^4 theory in two dimensions, the *field configuration* is a real function on the vertex set V_L of the square lattice with periodic boundaries and size $L \times L$: $\phi : V_L \to \mathbb{R}$. The ϕ^4 theory is described by a probability density $p(\phi) = \exp(-S(\phi))/Z$, with action

$$S(\phi) = \sum_{x,y \in V_L} \phi(x) \Delta_{x,y} \phi(y) + \sum_{x \in V_L} m^2 \phi(x)^2 + \lambda \phi(x)^4 \tag{5}$$

In the above, Δ is Laplacian matrix of the square lattice $(\mathbb{Z}/L\mathbb{Z})^{\times 2}$, m and λ are numerical parameters. In the case of this and other non-trivial field theoretical densities, direct sampling is impossible due to the statistical correlation between degrees of freedom spatially separated up to the *correlation*

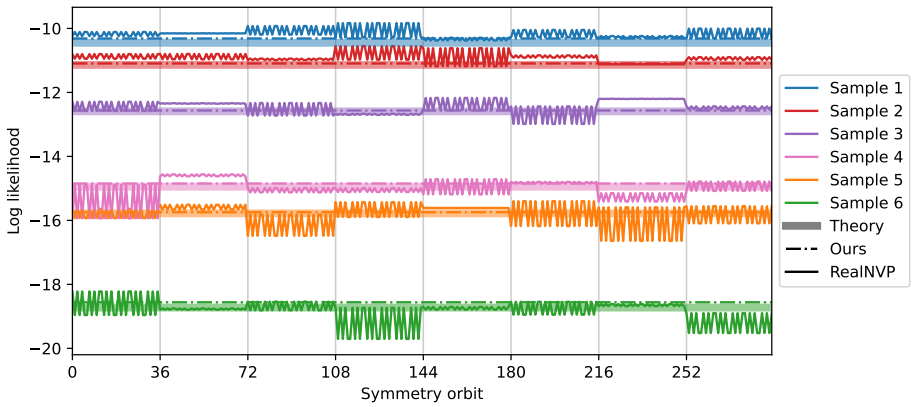

Figure 2: For 6 samples from an MCMC chain with $L = 6$, we show 1) the true log likelihood given by the action, 2) the model log likelihood of RealNVP and 3) our model. The x-axis shows the likelihoods when the sample is transformed by all $8 * 6^2$ symmetries of the lattice. Within each block of 1/8th of the x-axis, the samples are related by a translation, and by a rotation or mirror between the blocks.

length of the theory, a fundamental quantity denoted as ξ; Z is the normalisation factor that is not known analytically for $\lambda \neq 0$. Note that, besides the (space-time) symmetries of the periodic lattice, the theory possesses a discrete global symmetry $\phi \mapsto -\phi$. We shall choose the couplings in such a way that only one minimum of the action, invariant under this symmetry, exists. See [19] for relevant work in the case of a symmetry-broken case.

3 Method

Inspired by equivariant flows used for molecular modelling [15], we propose to use the following vector field for the neural ODE:

$$\frac{\mathrm{d}\phi(x,t)}{\mathrm{d}t} = \sum_{yaf} W_{xyaf} K(t)_a \sin(\omega_f \phi(y,t)) \tag{6}$$

and we have some learnable frequencies ω_f, initialised standard normal, to construct a Fourier basis expansion inspired by Fourier Features [20]. We chose only sines to enforce $\phi \mapsto -\phi$ equivariance. In Eq. (6) $x, y \in V_L$, $K(t)_a$ is a linear interpolation kernel, illustrated in the Appendix in Fig. 4 for the 5-dimensional case. W is a learnable tensor, initialised to 0, so that the initial flow in an identity transformation. The divergence of this simple flow is computed analytically.

The periodic lattice V_L has spatial symmetry group $G = C_L^2 \rtimes D_4$, the semi-direct product of two cyclic groups C_L of translations and Dihedral group D_4 of right angle rotations and mirrors. To ensure spatial equivariance of the vector field model, we should have that $\forall g \in G, x, y, a, f, W_{g(x)g(y)af} = W_{xyaf}$. Using the translation subgroup, we can map any point x to a fixed point x_0. This allows us to write $W_{xyaf} = W_{x_0 t_x(y) af}$, $t_x(y) = y - x + x_0$. Then let $H \simeq D_4$ be the subgroup of G such that $g(x_0) = x_0$ for all $g \in G$, and denote the orbit of y under H by $[y] = \{y' \mid \exists g \in H, g(y) = y'\}$. For each such orbit $[y]$ and dimension a and f, a free parameter $W_{[y]af}$ exists, so that the other parameters are generated by $W_{xyaf} = W_{[t_x(y)]af}$. As most orbits are of size 8, the number of free parameters per a and f is approximately $L^2/8$. See Fig. 5 in the Appendix for an figure of the orbits.

4 Experiments

We here discuss experimental results obtained using the flow architecture just described for the ϕ^4 theory. See Appendix C for the hyperparameters used. In order to investigate scalability, we consider a range of lattice side lengths L going from 6 to 32 with the correlation length being indirectly fixed at $L/4$ via appropriate choice of the coupling constant λ in the action, Eq. (5) [3]. The choice of m, λ for all sizes are reported in table 1. To assess model quality, we use two different but related metrics, the Effective Sample Size (ESS) and the MCMC acceptance rate, during training and test time. In the left panel of Figure 1 we report the ESS values against the real training time (in days) with the realNVP baseline for comparison, which is based on [21]. Our model attains much larger ESSs and acceptance rates in much shorter times, for all sizes L, even taking into account the fact that each of its training steps involves integration of an ODE. In the right panel we record the acceptance rates of our model, which are similarly superior across different sizes, indicating that their performance suffers less from CSD. We also cross-check the quality of the generated samples via the estimation of two physical observables: the *two point susceptibility* χ_2 and the *inverse pole mass* $m_p L = \frac{L}{\xi}$. We report the results obtained after 2 weeks of training on a single GPU in Table 1. See Appendix C for definition of these quantities that are standard in statistical physics.

L	6	12	20	32
m^2	-4	-4	-4	-4
λ	6.975	5.276	5.113	4.750
$\frac{L}{\xi}$	3.98	3.99	4.05	4.05
χ	1.06	4.12	10.57	25.34
Acceptance (%)	96	91	82	66
ESS (%)	99	97	90	66

Table 1: Observables computed with our model.

Furthermore, we verify the invariance of the model distribution $q(\phi)$, which is a result of the equivariance of our model, under the spatial symmetries. In figure 2 we report the model log-probability of some randomly chosen field configurations and their images under spatial symmetry

[3] Here and everywhere else we work with unit lattice spacing.

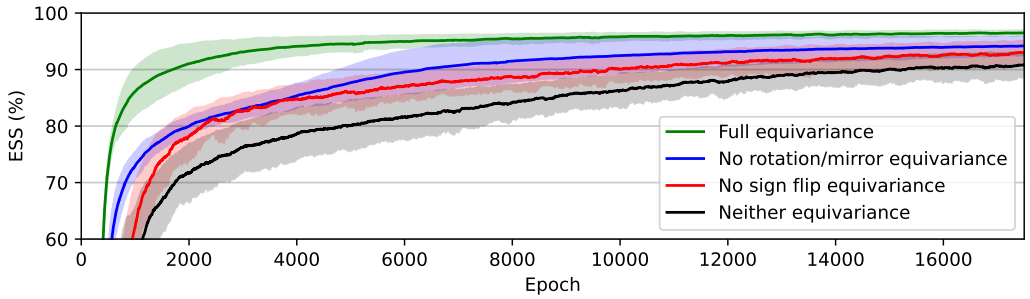

Figure 3: Ablation study on the $L = 12$ lattice, varying the equivariance properties. Shown are mean and standard deviation across three training runs per variation.

transformations. As expected, these symmetry-equivalent configurations are equally likely, while this is not the case for the realNVP. Interestingly, the symmetry violations of the latter does not seem to decrease as training progresses, indicating that these exact symmetries are not easily learned.

Finally, we conducted an ablation study to verify that equivariance is beneficial. We consider two modifications to the model. In one variation, we remove the rotational and mirror spatial symmetries, by constraining the W tensor in Eq. (6) to be equivariant only to translations, but not to rotations and mirrors. In the second variation, we remove the sign flip equivariance by replacing the sine in Eq. (6) by a combination of sine, cosine and a constant. The results in figure 3 show that removing either or both of the equivariance properties harms performance, indicating that incorporating all symmetries in the model is indeed beneficial.

5 Conclusion

In this paper, a shallow normalizing flow with features handcrafted via domain-specific knowledge, few trainable parameters, and an equivariant structure, is shown to significantly outperform a deep architecture trained end-to-end on a task of generative modelling of a simple, but paradigmatic field theory. There are various worthwhile directions for further investigations. We now discuss two of them. The first is to systematically assess the scalability of our approach, as is done in [12] for a seemingly less scalable architecture. The scalability will tell whether our method actually has the potential to provide a pragmatic solution to CSD. The second is to include equivariance under *local* and *non-abelian* symmetries to our framework, which is a necessary step towards an ML-based sampler for the more challenging and interesting case of QCD. One might believe that the simplicity and flexibility of our proposal suggests promising future results.

Acknowledgements

We would like to thank Vassilis Anagiannis for his contributions to the early stages of this project and Jonas Köhler and Max Welling for their helpful discussions.

References

[1] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[2] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual review of physical chemistry*, 71:361–390, 2020.

[3] Alexander Radovic, Mike Williams, David Rousseau, Michael Kagan, Daniele Bonacorsi, Alexander Himmel, Adam Aurisano, Kazuhiro Terao, and Taritree Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560(7716):41–48, 2018.

[4] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V. Albert, and John Preskill. Provably efficient machine learning for quantum many-body problems, 2021.

[5] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[6] MS Albergo, G Kanwar, and PE Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019.

[7] Kim A. Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Phys. Rev. E*, 101(2):023304, 2020.

[8] Jan M. Pawlowski and Julian M. Urban. Reducing Autocorrelation Times in Lattice Simulations with Generative Adversarial Networks. *Mach. Learn. Sci. Tech.*, 1:045011, 2020.

[9] Gurtej Kanwar, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Equivariant flow-based sampling for lattice gauge theory. *Phys. Rev. Lett.*, 125:121601, Sep 2020.

[10] Michael S. Albergo, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Julian M. Urban, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Flow-based sampling for fermionic lattice field theories, 2021.

[11] Kim A. Nicoli, Christopher J. Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Pan Kessel, Shinichi Nakajima, and Paolo Stornati. Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models. *Phys. Rev. Lett.*, 126(3):032001, 2021.

[12] Luigi Del Debbio, Joe Marsh Rossney, and Michael Wilson. Efficient modelling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability, 2021.

[13] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017.

[14] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), 2019.

[15] Jonas Köhler, Leon Klein, and Frank Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5361–5370. PMLR, 13–18 Jul 2020.

[16] Akio Tomiya and Yuki Nagai. Gauge covariant neural network for 4 dimensional non-abelian gauge theory, 2021.

[17] István Montvay and Gernot Münster. *Quantum fields on a lattice*. Cambridge University Press, 1997.

[18] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[19] Daniel C. Hackett, Chung-Chun Hsieh, Michael S. Albergo, Denis Boyda, Jiunn-Wei Chen, Kai-Feng Chen, Kyle Cranmer, Gurtej Kanwar, and Phiala E. Shanahan. Flow-based sampling for multimodal distributions in lattice field theory, 2021.

[20] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

[21] Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Introduction to normalizing flows for lattice field theory, 2021.

[22] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] I. Vierhaus. *Simulation of φ^4 theory in the strong coupling expansion beyond the Ising limit*. PhD thesis, Humboldt University, 2010.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss in the conclusion that we currently on have tested our model on the simple ϕ^4 model and that we haven't properly studied the scalability to even larger lattices.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We believe that this is relevant for theoretical physicists studying quantum field and that the impact to society is very limited. If anything, as our model trains faster than the realNVP baseline, there is a minimal positive environmental impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The relevant theoretical contribution is that our model in Eq. 6 is indeed sign-flip and spatially equivariant, which is argued in the text.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We hope we can release the source code soon.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Yes, see the plot of the ablation study.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We mention the resources in the appendix and show training time in the first plot.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use and cite the RealNVP code from [21]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Background Details

A.1 Algorithms

Algorithm 1 Independent Metropolis Hastings algorithm.

Require: $q(\phi)$: proposal density, $p(\phi)$: target density.
  for $i = 1 : K$ **do**
    $\phi' \sim q(\phi)$
    $u \sim \text{Uniform}(u; 0, 1)$
    $\rho = \min\left(1, \frac{q(\phi^{(i-1)})}{p(\phi^{(i-1)})} \frac{p(\phi')}{q(\phi')}\right)$
    if $u < \rho$ **then**
      $\phi^i = \phi'$ ▷ Accept
    else
      $\phi^i = \phi^{i-1}$ ▷ Reject
    end if
  end for

Algorithm 2 Reverse KL training of a normalizing flow

Require: f_θ: invertible neural network, $S(\phi)$: field theory action, $r(z)$ easy-to-sample distribution, B: batch size, γ_i: learning rates.
  for $i = 1 : K$ **do**
    $z^1, \dots, z^B \sim r(z)$
    $L = \frac{1}{B} \sum_{b=1}^{B} \log q(f_\theta^{-1}(z^b)) + S(f_\theta^{-1}(z^b))$
    $\theta = \theta - \gamma_i \nabla_\theta L$
  end for

A.2 Real NVP

A Real NVP flow uses a stack of coupling layers $g : z \in \mathbb{R}^N \mapsto \phi \in \mathbb{R}^N$ defined as follows. We partition the input/output space in two parts of size N_a, N_b, and use indices a and b for the elements in each part. Then a coupling layer is [13]:

$$\phi_a = z_a \,, \quad \phi_b = z_b \odot s(z_a) + t(z_a) \,. \tag{7}$$

Here s, t are neural networks, and this layer is invertible if s is invertible for all t. Its inverse, g^{-1}, is

$$z_a = \phi_a \,, \quad z_b = (\phi_b - t(\phi_a)) \odot s(\phi_a)^{-1} \,. \tag{8}$$

Denoted f the real NVP normalizing flow with latent distribution ρ, we can compute explicitly the push forward distribution entering the reverse KL loss:

$$\log q(f^{-1}(z)) = \log \rho(z) - \log |\det J_f| \tag{9}$$

where the log Jacobian determinant can be computed in $O(N)$ time instead of a naive $O(N^3)$, by summing each coupling layer contribution:

$$\log |\det J_f(z)| = \sum_\ell \sum_{i=1}^{N_a} \log |s_\ell(z_a)_i| \,, \tag{10}$$

where s_ℓ is the scale factor of the ℓ coupling layer. While models based on real NVP flows have achieved success in a number of ML tasks, see e.g. [22], we note that partitioning the input space prevents to construct an equivariant layer under a symmetry that maps $(z_a)_i \mapsto (z_b)_j$ for some i, j. In section 4 we show that in our experiments that the density induced by a real NVP normalizing flow trained to match the invariant density of the ϕ^4 theory does not obtain good invariance properties under the symmetries of the square lattice.

B Details model

Figures 4 and 5 illustrate the kernels used in our model.

C Experimental details

The flow architecture we propose is trained according to the following, and fairly standard experimental protocol. Starting field configurations z, composed of $D \equiv L \times L$ uncorrelated sites are sampled

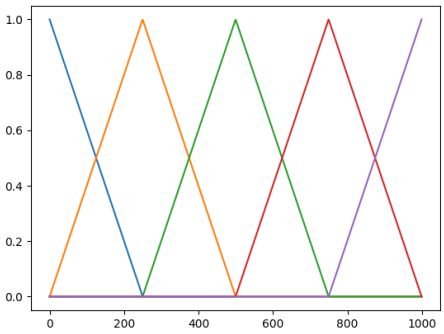

Figure 4: The 5D linear interpolation kernel.

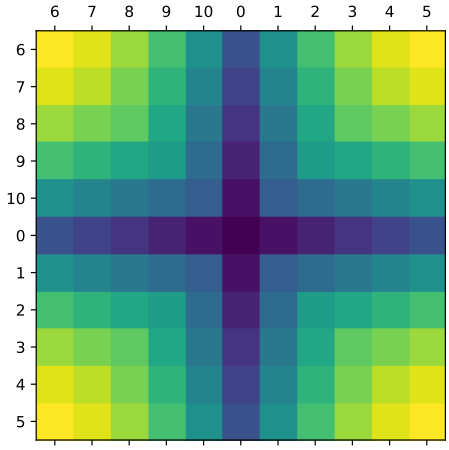

Figure 5: Each color denotes one D_4 orbit of a periodic $L = 11$ lattice, leaving $x_0 = (0, 0)$ invariant. For each color in the figure, and for each dimensions a and f, we have a free parameter $W_{[y]af}$.

from a Gaussian prior of zero mean and unit variance. They are then passed through the flow, which integrates its ODE from $x(t = 0) \equiv z$ to the transformed field configurations $x(t = 1) \equiv \phi$, as well as the ODE for the associated log Jacobian. We opted to use the 4th order Runge-Kutta solver, with a relatively generous number (50) of integration steps thanks to the simplicity of our flow. The batch size was fixed at $M = 100$ throughout. The parameters of the flow are trained via minimization of the reverse KL loss, eq. (1), which we perform with the Adam [23] optimizer with a learning rate fixed to $\eta = 0.001$ for the first 250 steps, and reduced by $\frac{1}{10}$ afterwards.

For all lattice sizes, we used in our model the same hyperparameters: 10 time dimensional time kernel K and 9 learnable frequencies w_f. We experimented also with 3 time dimensions and found worse performance, but didn't run any further hyperparameter search. During some exploratory experiments, we found indications that using larger batch sizes, like 1000 or larger, and/or other learning rate scheduling may speed up training by a significant amount.

The RealNVP model uses the code and hyperparameters from [21] (CC BY 4.0 licensed), which uses 16 coupling layers and in each coupling layer a 2 layer CNN with 8 dimensional hidden layers and kernel size 3.

All computations were run on NVidia V100 GPUs. The training for the results reported in 1 lasted two weeks.

The expected sample sizes were computed over 1000 samples. The ESS plots were smoothed over by a rolling average over 200 epochs to make trends more readable. Each epoch lasted 50 training steps, each of which had batch size 100.

C.1 Observables

A perfectly trained flow would be able to produce a set of uncorrelated field configurations $(\phi_i)_{i=1}^{\mathcal{N}}$, which could then be used to compute estimates of physical observables whose MSE would scale like $1/\sqrt{\mathcal{N}}$. This is usually not the case in practice, and the MSE actually scales like $1/\sqrt{\mathcal{N}_{\text{eff}}}$, with $\mathcal{N}_{\text{eff}} < \mathcal{N}$; the ratio $\frac{\mathcal{N}_{\text{eff}}}{\mathcal{N}}$, referred to as the Effective Sample Size (ESS, see (11)),

$$\text{ESS} \equiv \frac{\left(\frac{1}{\mathcal{N}}\sum_i p(\phi_i)/q(\phi_i)\right)^2}{\frac{1}{\mathcal{N}}\sum_i \left(p(\phi_i)/q(\phi_i)\right)^2}. \tag{11}$$

is the metric we use to monitor the progress of training, and the results are reported in the left panel of figure 1.At test time, we evaluate the trained flows by using them to generate a chain of MCMC proposals of length $\mathcal{N} = 10^6$, and use the fraction of accepted moves as our test metric. The resulting acceptance rates are reported in the right panel of figure 1.

To examine the quality of the generated samples, one of the physical quantities we compute is the two-point susceptibility

$$\chi_2 \equiv \frac{1}{D}\sum_{x,y} \mathbb{E}\left[\phi(x)\phi(x+y)\right] - \mathbb{E}\left[\phi(x)\right]\mathbb{E}\left[\phi(x+y)\right]. \tag{12}$$

Explicitly, we use the following estimator

$$\begin{aligned}\hat{G}(x) \equiv \frac{1}{D}\sum_y \frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}} \Bigg[& \phi_i(y)\phi_i(y+x) - \phi_i(x+y)\frac{1}{\mathcal{N}}\sum_{j=1}^{\mathcal{N}}\phi_j(x) \\ & -\phi_i(x)\frac{1}{\mathcal{N}}\sum_{j=1}^{\mathcal{N}}\phi_j(x+y) + \frac{1}{\mathcal{N}}\sum_{k=1}^{\mathcal{N}}\phi_k(x)\frac{1}{\mathcal{N}}\sum_{j=1}^{\mathcal{N}}\phi_j(x+y)\Bigg];\end{aligned} \tag{13}$$

the susceptibility is then simply obtained as $\hat{\chi}_2 = \sum_x \hat{G}(x)$.

As for the inverse pole mass, we use the estimator

$$\hat{m}_p \equiv \frac{1}{L-1}\sum_{x_2=1}^{L-1}\left(\frac{G_c(x_2-1)+G_c(x_2+1)}{2G_c(x_2)}\right), \tag{14}$$

where

$$G_c(x_2) \equiv \frac{1}{L}\sum_{x_1=1}^{L}\hat{G}(x_1,x_2), \tag{15}$$

and the vertices of the periodic lattice are explicitly labelled by their two coordinates (x_1, x_2). Given the choice of parameters detailed above [24], we expect $m_p L \simeq 4$ for every L, which is always the case. The resulting values of our computations are recorded in Table 1.