# Conditional normalizing flow for Markov chain Monte Carlo sampling in the critical region of lattice field theory

Ankur Singha,[1] Dipankar Chakrabarti,[1] and Vipul Arora[2]

[1]*Department of Physics, Indian Institute of Technology Kanpur, Kanpur-208016, India*
[2]*Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur-208016, India*

(Received 8 August 2022; accepted 4 January 2023; published 25 January 2023)

The traditional Markov chain Monte Carlo (MCMC) suffers from the problem of critical slowing down. Generative machine learning methods, such as normalizing flows, offer a promising method to speed up MCMC simulations, especially in critical regions of lattice field theory. However, training these models for different parameter values in the critical region is inefficient. In this paper, we address this issue by interpolating or extrapolating the flow model to any parameter value in the critical region. We demonstrate the effectiveness of the proposed method for MCMC sampling in critical regions.

DOI: 10.1103/PhysRevD.107.014512

## I. INTRODUCTION

Lattice field theory is a popular nonperturbative method for solving quantum field theory which describes particle physics and many condensed matter systems. Markov Chain Monte Carlo (MCMC) methods are used in lattice field theory for sampling a probability distribution specified by the action or Hamiltonian of the lattice system. The cost of MCMC simulation depends on the parameter value of the theory at which the lattice samples are generated. This cost increases swiftly for the parameter values in the critical region, where the autocorrelation between the generated samples shoots up. This problem is known as critical slowing down [1,2]. Several MCMC algorithms have been developed for specific theories [3–10] to reduce the effect of critical slowing down. However, critical slowing down still poses a significant hurdle in estimating the physical observables in many lattice field theories.

Generative ML has recently has shown a great potential to efficiently address this issue [11–24]. Flow-based generative models have been found to be effective in avoiding the problem of critical slowing down for specific theories [25–33]. However, simulations at multiple parameter values near the critical region are necessary to study the critical properties of a statistical lattice system and take the continuum limit of a lattice field theory. In the traditional MCMC methods, the Markov chain starts from a random state for each parameter value. In the existing flow-based methods, the Markov chain utilizes a flow model that is trained for a particular parameter value. This is particularly inefficient if estimates at many parameter values are needed. For instance, estimation of free energy [34] requires lattice ensembles at discrete steps in the parameter space of the theory. Hence, training different flow models for each parameter value increases the simulation cost. As a side note, there is one approach [33] which does not require ensembles at discrete steps for calculating the free energy. There are several flow-based methods [26,27] which learn the flow models in an online fashion, i.e., while generating the samples, instead of pretraining with samples from training data. But online training could be unstable and may lead to mode collapse.

In ML literature, transfer learning techniques have been developed to efficiently transfer the knowledge across domains [35]. This transfer enables initializing the model in an efficient way for the new domain. Likewise, there are conditional flow models [36] which can regress across continuous parameters, thereby transferring knowledge across parameters. The question we address in this work is whether the knowledge from the noncritical region can be used to generate samples in the critical region of lattice theory. We develop a provably correct conditional normalizing flow-based (C-NF) sampling method for scalar $\phi^4$ theory.

Our goal is to generate lattice configurations from the distribution:

$$p_\Phi(\phi|\lambda) = \frac{1}{Z} e^{-S(\phi|\lambda)},$$
$$\text{where, } Z = \int D\phi e^{-S(\phi|\lambda)}. \tag{1}$$

Here, $\phi$ denotes the lattice field, Z is the partition function and $S$ is the lattice action of $\phi^4$ theory. The action can be defined in 2D lattice as:

$$S(\phi,\lambda,m)=\sum_x\Bigg[\sum_{\mu=1,2}[2\phi(x)^2-\phi(x)\phi(x+\hat{\mu}) -\phi(x)\phi(x-\hat{\mu})]+m^2\phi(x)^2+\lambda\phi(x)^4\Bigg], \quad (2)$$

where $\lambda$ and $m$ are the two parameters of the theory. In our numerical experiments we fix $m^2=-4$ and hence we omit this parameter in the expression of condition distribution $p_\Phi(\phi|\lambda)$. Here, $x$ is a $2d$ discrete vector and $\hat{\mu}$ represents two possible directions on the lattice.

We train a C-NF model using HMC samples from $p_\Phi(\phi|\lambda)$ for $\lambda\in\Lambda_{NC}$, where $\Lambda_{NC}$ denotes the noncritical region[1] of the theory. We use the trained C-NF model to generate proposals for constructing a Markov chain via an independent MH algorithm [25]. We are particularly interested in lattice configurations for $\lambda\in\Lambda_C$, where $\Lambda_C$ denotes the critical region of the theory.

There are two kinds of critical regions we study, one where noncritical regions exist on both sides of the critical point and the other where noncritical region exists only on one side. We apply C-NF to generate lattices for both these cases. For the former case, we interpolate the C-NF model, and for the latter, we extrapolate the model to the critical region. Interpolation is useful for studying phase transition in a statistical system, while extrapolation is useful for obtaining the continuum limit in lattice gauge theory.

## II. CONDITIONAL NORMALIZING FLOW

Normalizing flows [37] are generative models for constructing complex distributions by transforming a simple known distribution via a series of invertible and smooth mapping $f\colon\mathbb{R}^d\to\mathbb{R}^d$ with inverse $f^{-1}=g$. If $p_Z(z)$ is the prior distribution and $p_\Phi(\phi)$ is the complex target distribution, then the model distribution $q_\Phi(\phi;\theta)$ which is parametrized by variational weights $\theta$ can be written as

$$q_\Phi(\phi;\theta)=p_Z(z)\left|\det\frac{\partial f_\theta^{-1}(\phi)}{\partial\phi}\right|, \quad \text{where, } \phi=f_\theta(z). \quad (3)$$

Fitting a flow-based model $q_\Phi(\phi;\theta)$ to a target distribution $p_\Phi(\phi)$ can be accomplished by minimizing their KL divergence. The quantity minimized can be called as loss function and is defined as:

$$\mathcal{L}=D_{KL}[p_\Phi(\phi;c)||q_\Phi(\phi;c,\theta)] \quad (4)$$

We have estimated the network parameters $\theta_{ML}$ by optimizing the $\mathcal{L}(\theta)$ with respect to $\theta$ using gradient descent based method.

The most crucial step is to construct an invertible neural network representing the function $f_\theta(z)$, which enables an efficient evaluation of $|\det\frac{\partial f_\theta^{-1}(\phi)}{\partial\phi}|$. One such method is to use affine coupling blocks, which divides the input $z$ into two halves $z_1$ and $z_2$ and applies an affine transformation to produce upper or lower triangular Jacobians. The transformation rules for such a coupling block are as follows [38]:

$$\phi_1=z_1\odot\exp(s_1(z_2,c))+t_1(z_2,c),$$
$$\phi_2=z_2\odot\exp(s_2(\phi_1,c))+t_2(\phi_1,c), \quad (5)$$

where, $\odot$ represent element-wise product of two vectors. For conditional learning, we concatenate the conditional parameter $c$ along with the input $z$ and feed into the convolutional networks $s$ and $t$ [39] as shown in Fig. 1. We combine many such coupling blocks to construct the C-NF model. We have not conditioned each coupling layer of C-NF model with $\lambda$. In experiments, we found that if we condition each layer of the model with lambda, the model overfits the $\lambda$ values used for training and does not generalize to interpolated/extrapolated $\lambda$ values. For the results shown in this paper, we condition only the layers with index $n\times k$, where $n=0,1,\ldots$ and $k=1,5,9$ for lattices of size $(6\times6\ \&\ 8\times8)$, $12\times12$ and $16\times16$, respectively. The model descriptions can be found in the Appendix.

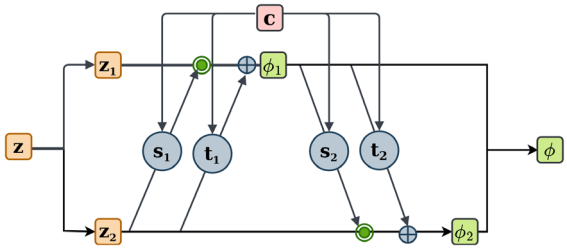



FIG. 1. One affine block of conditional normalizing flow, where s and t are convolutional neural networks.

Let us designate the C-NF model as $f(\phi;c,\theta)$ and its inverse as $g(z;c,\theta)$. The invertibility for any fixed condition $c$ is given by

$$f^{-1}(.;c,\theta)=g(.;,c,\theta) \quad (6)$$

To estimate the probability density function of generated samples, we can use

$$q_\Phi(\phi;c,\theta)=p_Z(f(\phi;c,\theta))\left|\det\frac{\partial f_\theta^{-1}(\phi)}{\partial\phi}\right|. \quad (7)$$

After training, for a fixed $c$, we can execute conditional generation of $\phi$ by sampling $z$ from a simple prior distribution $p_Z(z)$ and employing the inverted network $g(z;c,\theta_{ML})$.

## III. APPLICATION TO LATTICE SCALAR $\phi^4$ THEORY

The lattice action as in Eq. (2) possesses $\phi(x)\to-\phi(x)$ symmetry, however it is spontaneously broken at a specific parameter region. For the numerical investigation we consider the phase transition with respect to the parameter $\lambda$ only. For a given $\lambda$ the probability distribution of lattice configuration is given by $p_\Phi(\phi|\lambda)$ defined in Eq. (1).

[1]We define critical and noncritical sets based on integrated autocorrelation time.

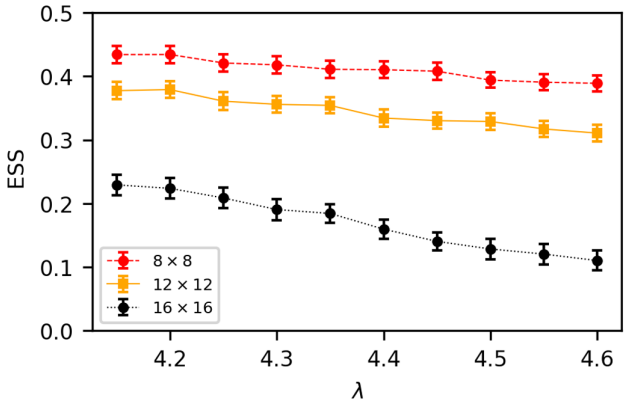



FIG. 2. ESS for extrapolated $\lambda$ calculated from models trained on three different lattice volumes. For each lattice, we use 65 k iterations for training.

More information on lattice $\phi^4$ theory can be found in [40]. Some of the observables which we will be calculating on the lattices are

(1) $\langle\tilde{\phi}^2\rangle$: $\tilde{\phi} = \frac{1}{V}\sum_x \phi(x)$.
(2) Zero momentum correlation function:
$C(t) = \sum_{x_1} G_c(x_1, t)$,
where, $x = (x_1, t)$ and
$G_c(x) = \frac{1}{V}\sum_y[\langle\phi(y)\phi(x+y)\rangle - \langle\phi(y)\rangle\langle\phi(x+y)\rangle]$.
(3) Two point susceptibility: $\chi = \sum_x G(x)$.

In the C-NF model, $\phi$ in Eq. (5) corresponds to the lattice field configuration of scalar $\phi^4$ theory and the action parameter $\lambda$ represent the conditional parameter c. $\lambda$ is fed to the network as a separate channel appended to $z$ with values $\lambda\mathbb{1}$, where $\mathbb{1}$ is a matrix with all entries as identity and size same as that of $z$. We generate training samples $\phi$ from the distribution defined in Eq. (1):

$$z = g(\phi; \lambda, \theta); \qquad \phi \sim p_\Phi(\phi|\lambda). \tag{8}$$

The loss function for optimizing the C-NF model is the forward KL divergence between the model distribution $q_\Phi(\phi; \lambda, \theta)$ and target distribution $p_\Phi(\phi; \lambda)$:

$$\begin{aligned}\mathcal{L}(\theta) &= \int d\phi\, p_\Phi(\phi;\lambda)(\log(p_\Phi(\phi;\lambda)) - \log(q_\Phi(\phi;\lambda,\theta))) \\ &= E_{\phi\sim p_\Phi}[\log(p_\Phi(\phi;\lambda)) - \log(q_\Phi(\phi;\lambda,\theta))].\end{aligned} \tag{9}$$

## IV. TRAINING AND SAMPLING PROCEDURE

### A. Training

We train two models for studying two different cases, namely, interpolation and extrapolation to the critical region. We use HMC generated samples for $\lambda \in \Lambda_{NC}$ to train these models. In HMC simulation we use molecular dynamics (MD) step size = 0.1 and MD trajectory length = 1. For each $\lambda \in \Lambda_{NC}$ we generate 10,000 lattice configurations for the interpolation case and 15,000 configurations for the extrapolation case. For both the cases we use a fixed number of training configurations which are rotated through in the training process.

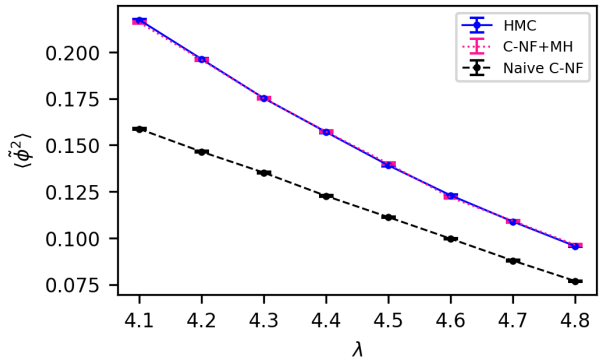
For estimation of observables from HMC we generate $10^5$ lattice configurations. We use bootstrap resampling method for estimating the uncertainty in the observables with a bin-size of 100.

To define the sets $\Lambda_C$ and $\Lambda_{NC}$, we choose two threshold values of $\lambda$ on both phases based on integrated autocorrelation time ($\tau$) of $\chi$. We select the threshold value of $\lambda$ such that maximum $\tau$ for $\Lambda_{NC}$ set is 1.5.

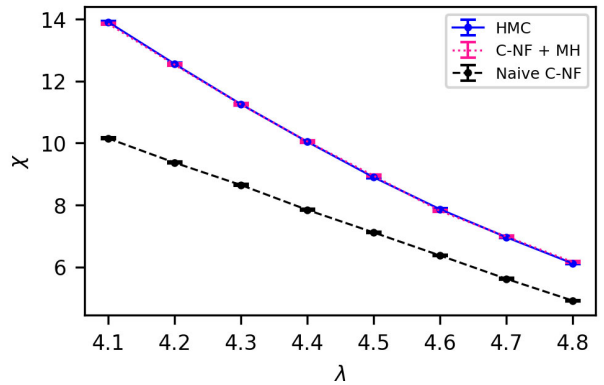
During training, we use a batch size of 1024 with each sample consisting of $(\phi, \lambda)$, chosen randomly from the ensembles of different $\lambda$ values. We employ an Adam optimizer with an initial learning rate of 0.0003.

### B. Sampling

For sampling, we draw $z$ from a prior distribution which we set as standard normal, $z \sim \mathcal{N}(0, 1)$. We feed z into the C-NF model to generate lattice samples $\phi = f(z)$, which are used as proposal samples for constructing a Markov chain. We choose 40%–45% acceptance rate as the




(a) $\langle\tilde{\phi}^2\rangle$




(b) $\chi$

FIG. 3. Interpolation to the critical region:-$\langle\tilde{\phi}^2\rangle$ and $\chi$ are calculated on samples generated from (i) HMC, (ii) C-NF followed by MH, and (iii) Naive C-NF. The error bars indicate standard deviation calculated using bootstrap resampling with bin-size 100.

stopping criteria for training. For the C-NF model we get the target acceptance rate around 65 k training iterations. Once training is over, we interpolate or extrapolate the model to the critical region. For generation in the critical region we give critical $\lambda$ values as conditional parameters. We use the acceptance rate, effective sample size (ESS) and three observables $\langle\tilde{\phi}^2\rangle$, $\chi$, and $C(t)$ as metric to evaluate the performance of the C-NF model. We calculate the observables for four different lattice sizes $6\times 6$, $8\times 8, 10\times 10$, and $12\times 12$. The results are almost same, so we presents the results only for one lattice namely $8\times 8$.

### C. C-NF model quality: Effective sample size (ESS)

The ESS is defined as [28]

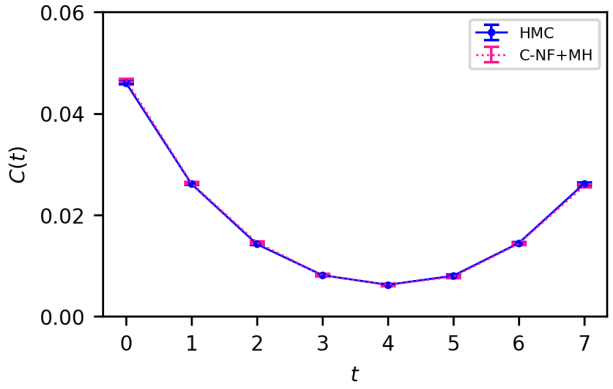
$$ESS = \frac{1}{N}\frac{(\sum_{i=1}^{N} p_\Phi(\phi_i;\lambda)/q_\Phi(\phi_i;\lambda,\theta))^2}{\sum_{i=1}^{N}(p_\Phi(\phi_i;\lambda)/q_\Phi(\phi_i;\lambda,\theta))^2}. \quad (10)$$

We estimate ESS for three different C-NF models corresponds to three lattice sizes namely, $8\times 8, 12\times 12, 16\times 16$. For estimation of ESS we use a batch-size of 5000 lattice configurations from each model trained to 65 k iterations. We took 100 independent estimation of ESS from each model to calculate the uncertainty in it. In Fig. 2 we plot the ESS for 10 different $\lambda$ values for each model. The ESS is almost constant for the extrapolated $\lambda$ values for 8 and $12\times 12$ lattices. For $16\times 16$ the ESS gradually decreases to 10% at $\lambda$ values very close to the critical point.

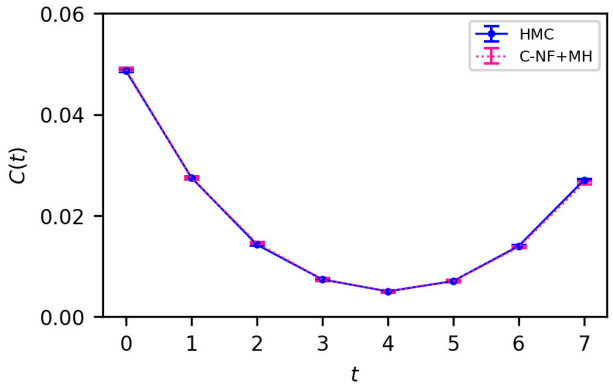
## V. NUMERICAL EXPERIMENTS AND RESULTS

### A. Interpolation to the critical region

For interpolation case, we choose training $\Lambda_{NC}$ set as: $\{3, 3.2, 3.5, 3.6, 3.7, 3.8, 5.8, 6, 6.5, 7, 8, 9\}$, with threshold $\lambda = 3.8$ and 5.8. We interpolate the trained model for multiple $\lambda$ values, $\Lambda_C$: $\{4.1, 4.2, 4.25, 4.3, 4.35, 4.4, 4.45, 4.5, 4.55, 4.6, 4.65, 4.7, 4.8, 5.0\}$. For each $\lambda$ values we generates a Markov chain of $10^5$ configurations from the proposed method. Then observables are calculated on each ensemble for both the phases around the critical point. We compare the observables from our proposed method with those from HMC simulation in Figs. 3–7. Observables from the naive C-NF without MH algorithm are also shown to illustrate the C-NF model's proximity to the true distribution.

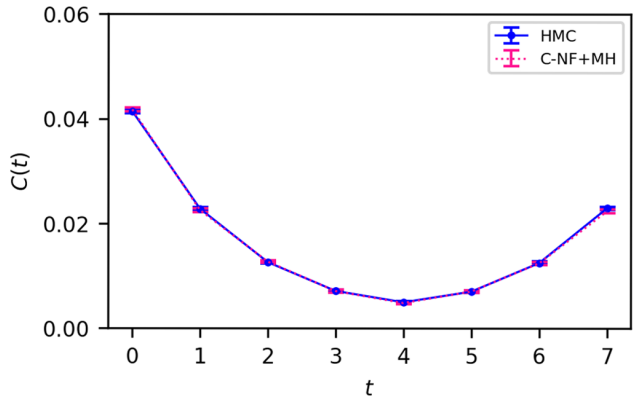



(a) $\lambda = 4.5$

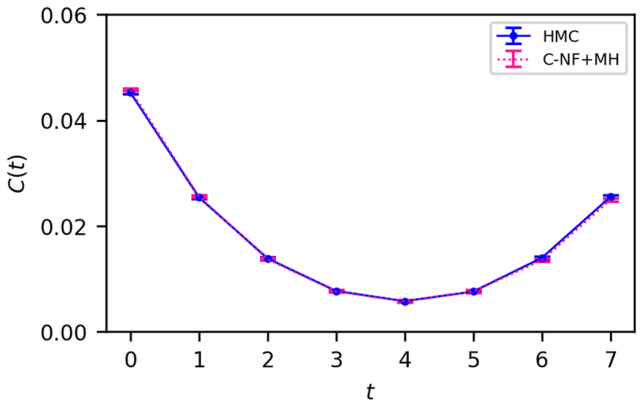



(b) $\lambda = 4.70$

FIG. 4. Interpolation: Zero momentum correlation function $C(t)$ calculated on samples generated from (i) HMC and (ii) C-NF followed by MH. The error bars indicate standard deviation calculated using bootstrap resampling with bin-size 100.

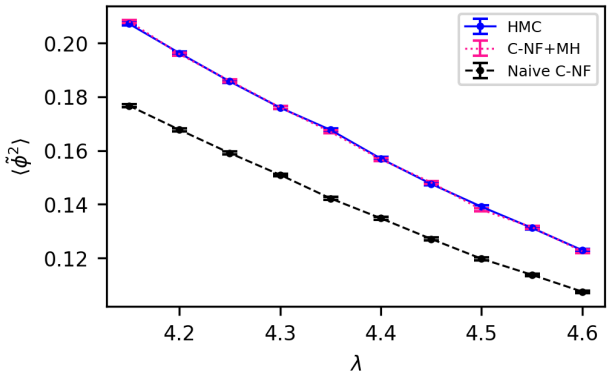



(a)

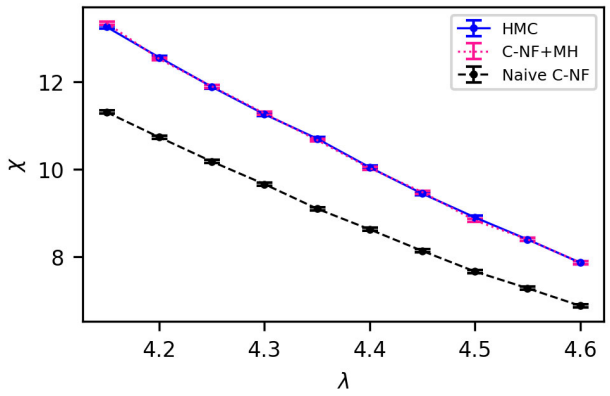



(b)

FIG. 5. Extrapolation:-Zero momentum Correlation function calculated on samples generated from (i) HMC and (ii) C-NF followed by MH. The error bars indicate standard deviation calculated using bootstrap resampling with bin-size 100.

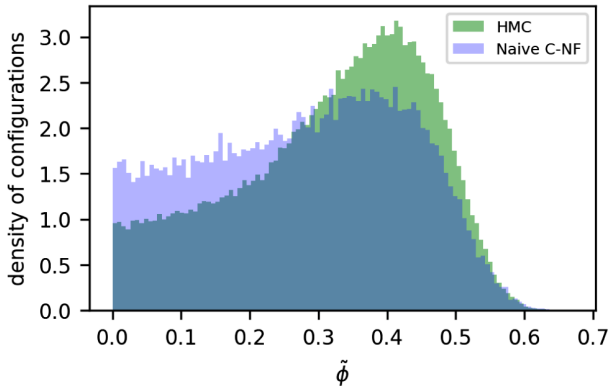



(a) $\langle\tilde{\phi}^2\rangle$

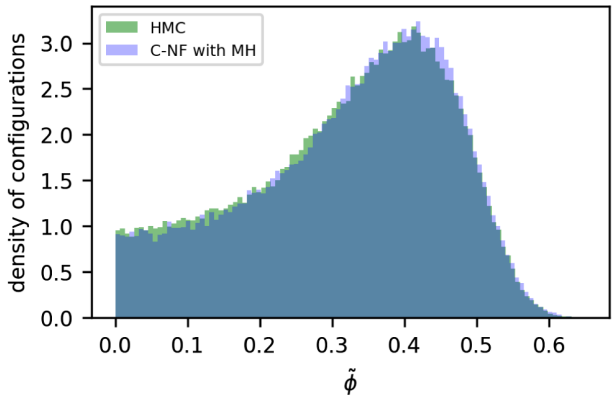



(b) $\chi$

FIG. 6. Extrapolation to the critical region: $\langle\tilde{\phi}^2\rangle$ and $\chi$ are calculated on samples generated from (i) HMC, (ii) C-NF followed by MH, and (iii) Naive C-NF. The error bars indicate standard deviation calculated using bootstrap resampling with bin-size 100.




(a)




(b)

FIG. 7. Histogram of $\tilde{\phi}$ for $\lambda = 4.6$ from the (a) naive C-NF model and (b) C-NF with MH is compared against HMC results with a bin-size = 100.

In Fig. 3, we plot the observables $\langle\tilde{\phi}^2\rangle$ and $\chi$ for $\lambda \in \Lambda_C$. In Fig. 4, the two point zero momentum correlation function $C(t)$ is shown for two $\lambda$ values. For these $\lambda$ values, we observe larger correlation compared to non-critical $\lambda$ values.

In Fig. 7, we also show the histogram of $\tilde{\phi}$ for a particular critical $\lambda = 4.6$. It visually demonstrates the elimination of artifacts by the MH algorithm, which the C-NF model produced.

### B. Extrapolation to the critical region

The $\Lambda_{NC}$ set used to train the C-NF model for the extrapolation case in the broken phase is :$\{3, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9\}$. We find that in this case the size of training dataset has to be increased to achieve ESS comparable to the interpolation case. After training, we extrapolate it in the critical region around $\lambda = 4.6$, which is taken as the critical region's midpoint, i.e., $\lambda \in \{4.2, 4.3, 4.4, 4.5, 4.6\}$. For each $\lambda$ value we generate one ensemble of $10^5$ configurations from our method.

We plot the observables $\langle\tilde{\phi}^2\rangle$ and $\chi$ in Fig. 6 and the zero momentum correlation function is plotted in Fig. 5 for two different $\lambda \in \Lambda_C$.

The naive C-NF model produces inherently uncorrelated lattice configurations. However, using such configurations directly from the naive C-NF model will lead to biases in the observables. We use the independent MH algorithm to avoid the model's biases, which guarantees the asymptotically exact Markov chain.

### C. Comparison to a nonconditional NF model

A nonconditional NF model trained on a single $\lambda$ value can be used to generate proposals for MH algorithm to construct ensembles for different $\lambda$ values. However, this model may not generate good proposals for the target distribution. The efficiency for constructing such ensembles for other $\lambda$ values will depend on the acceptance rate in MH algorithm. In Fig. 8 we compare the acceptance rate for three different models. We train a nonconditional NF with samples for a single $\lambda = \lambda_0 = 3.8$ and use the model to generate proposals for MH algorithm at other $\lambda$ values. We observe a faster decline in the acceptance rate as we move away from $\lambda_0$ as compared to the acceptance rate obtained with the interpolated and extrapolated C-NF.

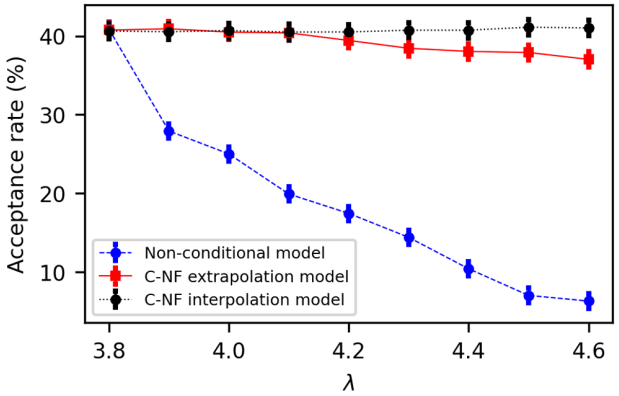



FIG. 8. MH acceptance rate from three different models are shown: (1) model trained on a single $\lambda = 3.8$ (blue dashed); (2) C-NF extrapolated model (red solid) and (3) C-NF interpolated model (black dotted).

## VI. COST ANALYSIS

The sampling algorithm for the baseline approach (HMC) and the suggested method is vastly different; thus, a direct cost comparison is opaque. Nonetheless, we separate the simulation cost for the proposed technique into two components: training time and sample generation time. On a Colab Tesla P100 GPU, the training time for the C-NF model is roughly 5-6 hours. However, sample generation is very fast for the C-NF model once training is over. Generating a Markov chain of $10^5$ configuration from C-NF + MH takes only 5–7 minutes with ~40% acceptance rate. Due to the short generation time, it can be utilized for generation of large ensembles of lattice configurations. The proposed method allows us to extend the same model to generate lattice samples for any value of action parameters $\lambda$ in the range [3.0–10]. Therefore, the proposed method could be efficient when lattice simulation at multiple $\lambda$ values, especially in the critical region, is required.

## VII. CONCLUSION

The critical slowing down problem prevents generating a large ensemble in the critical region of lattice theory. To resolve this issue, we employ a conditional normalizing flow model (C-NF) to efficiently generates proposals for constructing a Markov chain using the independent MH algorithm. We train the C-NF model to learn a conditional distribution over action parameters which enables us to generate proposals at any action parameter of the theory. We apply this method to the scalar lattice $\phi^4$ theory. We train the C-NF model on HMC samples in the noncritical region where simulation cost is low. The trained model can generates proposal samples for any parameter $\lambda$. Thus the proposed method can generate Markov chain for any $\lambda$ in the critical and noncritical regions of the theory.

We experimented with interpolating and extrapolating our model to the critical region. We generate lattice configurations for multiple $\lambda$ values in the critical region for both the cases around 40% acceptance rate. We also compare various observables from our method with that from HMC simulation and find a good agreement. The method developed in this paper, can be extended for lattice simulation of gauge theory.

## ACKNOWLEDGMENTS

We acknowledge partial financial support from SPARC under Project code P331. We also acknowledge the partial financial support from Science and Engineering Research Board under the Grant No. CRG/2021/003466.

## APPENDIX: C-NF ARCHITECTURE

We construct a C-NF model by alternate combination of affine layers as shown in Fig. 1. One such affine layer is shown here. Parameters of the C-NF architecture for different lattice sizes are also listed. Note that we have not conditioned $\lambda$ to each coupling layer as the model overfit to the trained $\lambda$ values. We use a gap of 0,0,4, and 8 in $6 \times 6$, $8 \times 8$, $12 \times 12$, and $16 \times 16$ lattices respectively.

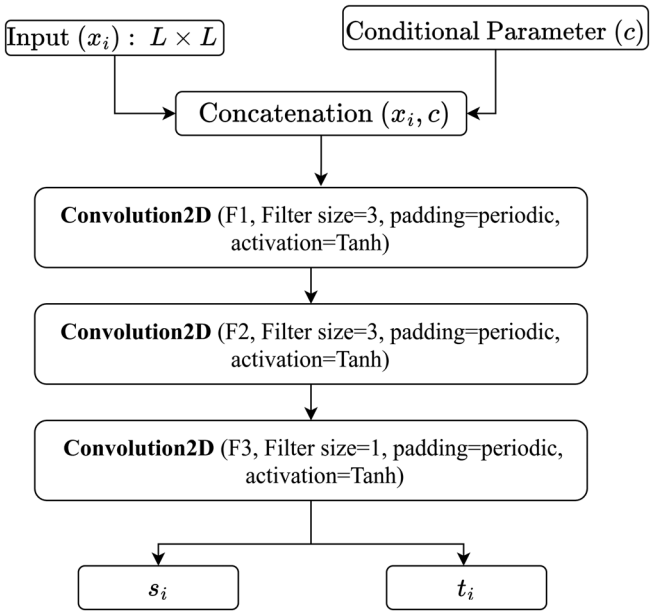



FIG. 9. Architecture of the neural networks **s** and **t** of the $i$th affine coupling layer.

TABLE I. Parameters of the C-NF architecture for different lattice sizes. $F1, F2$, and $F3$ are as defined in Fig. 9.

| Lattice sizes | No. of affine layers | $F1, F2, F3$ |
|---|---|---|
| $6 \times 6$ | 8 | 16, 16, 2 |
| $8 \times 8$ | 8 | 32, 32, 2 |
| $12 \times 12$ | 16 | 32, 32, 2 |
| $16 \times 16$ | 24 | 16, 16, 2 |

[1] Ulli Wolff, Critical slowing down, Nucl. Phys. B, Proc. Suppl. **17**, 93 (1990).

[2] Stefan Schaefer, Rainer Sommer, and Francesco Virotta, Critical slowing down and error analysis in lattice QCD simulations, Nucl. Phys. **B845**, 93 (2011).

[3] Robert H. Swendsen and Jian-Sheng Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, Phys. Rev. Lett. **58**, 86 (1987).

[4] J. Hoshen and R. Kopelman, Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm, Phys. Rev. B **14**, 3438 (1976).

[5] Ulli Wolff, Monte Carlo simulation of a lattice field theory as correlated percolation, Nucl. Phys. **B300**, 501 (1988).

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

[7] W. Bietenholz, A. Pochinsky, and U. J. Wiese, Meron-Cluster Simulation of the $\theta$ Vacuum in the 2d o(3) Model, Phys. Rev. Lett. **75**, 4524 (1995).

[8] H. G. Evertz, The loop algorithm, Adv. Phys. **52**, 1 (2003).

[9] Nikolay Prokof'ev and Boris Svistunov, Worm Algorithms for Classical Statistical Models, Phys. Rev. Lett. **87**, 160601 (2001).

[10] Herbert Neuberger, Adler's Overrelaxation Algorithm for Goldstone Bosons, Phys. Rev. Lett. **59**, 1877 (1987).

[11] Kai Zhou, Gergely Endrődi, Long-Gang Pang, and Horst Stöcker, Regressive and generative neural networks for scalar field theory, Phys. Rev. D **100**, 011501 (2019).

[12] Dian Wu, Lei Wang, and Pan Zhang, Solving Statistical Mechanics Using Variational Autoregressive Networks, Phys. Rev. Lett. **122**, 080602 (2019).

[13] Jan M. Pawlowski and Julian M. Urban, Reducing autocorrelation times in lattice simulations with generative adversarial networks, Mach. Learn. **1**, 045011 (2020).

[14] 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**, 023304 (2020).

[15] Juan Carrasquilla, Machine learning for quantum matter, Adv. Phys. **5**, 1797528 (2020).

[16] Junwei Liu, Huitao Shen, Yang Qi, Zi Yang Meng, and Liang Fu, Self-learning Monte Carlo method and cumulative update in fermion systems, Phys. Rev. B **95**, 241104 (2017).

[17] Johanna Vielhaben and Nils Strodthoff, Generative neural samplers for the quantum Heisenberg chain, Phys. Rev. E **103**, 063304 (2021).

[18] Chuang Chen, Xiao Yan Xu, Junwei Liu, George Batrouni, Richard Scalettar, and Zi Yang Meng, Symmetry-enforced self-learning Monte Carlo method applied to the Holstein model, Phys. Rev. B **98**, 041102 (2018).

[19] Giacomo Torlai and Roger G. Melko, Learning thermodynamics with Boltzmann machines, Phys. Rev. B **94**, 165134 (2016).

[20] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[21] Lei Wang, Discovering phase transitions with unsupervised learning, Phys. Rev. B **94**, 195105 (2016).

[22] Pengfei Zhang, Huitao Shen, and Hui Zhai, Machine Learning Topological Invariants with Neural Networks, Phys. Rev. Lett. **120**, 066401 (2018).

[23] Japneet Singh, Mathias Scheurer, and Vipul Arora, Conditional generative models for sampling and phase transition indication in spin systems, SciPost Phys. **11**, 043 (2021).

[24] Ankur Singha, Dipankar Chakrabarti, and Vipul Arora, Generative learning for the problem of critical slowing down in lattice Gross Neveu model, SciPost Phys. Core **5**, 052 (2022).

[25] 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).

[26] 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, Phys. Rev. D **104**, 114507 (2021).

[27] 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 (2020).

[28] 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, arXiv:2101.08176.

[29] Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Fernando Romero-López, Phiala E. Shanahan, and Julian M. Urban, Flow-based sampling in the lattice Schwinger model at criticality, Phys. Rev. D **106**, 014514 (2022).

[30] 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, arXiv:2107.00734.

[31] Phiala E. Shanahan, Daniel Trewartha, and William Detmold, Machine learning action parameters in lattice quantum chromodynamics, Phys. Rev. D **97**, 094506 (2018).

[32] Kim A. Nicoli, Christopher Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Pan Kessel, Shinichi Nakajima, and Paolo Stornati, Machine learning of thermodynamic observables in the presence of mode collapse, Proc. Sci. **LATTICE2021**, 338 (2022).

[33] 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**, 032001 (2021).

[34] Owe Philipsen, The QCD equation of state from the lattice, Prog. Part. Nucl. Phys. **70**, 55 (2013).

[35] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He, A comprehensive survey on transfer learning, Proc. IEEE **109**, 43 (2021), arXiv:1911.02685.

[36] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari, C-flow: Conditional generative flow models for images and 3d point clouds, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), arXiv:1912.07009.

[37] Danilo Jimenez Rezende and Shakir Mohamed, Variational inference with normalizing flows, in *International Conference on Machine Learning* (2015), arXiv:1505.05770.

[38] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe, Guided image generation with conditional invertible neural networks, 2019, arXiv:1907.02392.

[39] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe, Guided image generation with conditional invertible neural networks, arXiv:1907.02392.

[40] Ingmar Vierhaus, Simulation of phi 4 theory in the strong coupling expansion beyond the Ising limit, Master's thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2010.