Scalable generative sampling and multilevel estimation for lattice field theories near criticality

Ankur Singha,[1,2,*] Jacob Kauffmann,[1,2] Elia Cellini,[3] Karl Jansen,[4,5] and Shinichi Nakajima[1,2,6,†]

[1]*Berlin Institute for the Foundations of Learning and Data (BIFOLD), Berlin, Germany*
[2]*Technische Universität Berlin, Berlin, Germany*
[3]*Higgs Centre for Theoretical Physics, School of Physics and Astronomy, The University of Edinburgh, Edinburgh EH9 3FD, United Kingdom*
[4]*Deutsches Elektronen-Synchrotron (DESY), Platanenallee 6, 15738 Zeuthen, Germany*
[5]*Computation-Based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[6]*RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan*

(Received 20 April 2026; accepted 5 October 2026; published 26 October 2026)

Sampling lattice field theories near criticality is severely hindered by critical slowing down, which makes standard Markov chain methods increasingly inefficient at large lattice volumes. We introduce a multiscale generative sampler, inspired by renormalization-group ideas, that models the Boltzmann distribution through a coarse-to-fine hierarchy across length scales. At each level, a conditional Gaussian mixture model captures the main local dependence of newly introduced variables on the already-sampled coarse field, while a masked continuous normalizing flow refines the remaining conditional structure. Coarse levels encode the dominant long-wavelength modes, and finer levels progressively add short-distance fluctuations. In addition, because the architecture preserves coarse fields exactly during refinement, it provides exact restriction maps at no additional computational cost and directly enables asymptotically unbiased multilevel Monte Carlo variance reduction. For the two-dimensional scalar ϕ^4 theory near criticality, the method achieves integrated autocorrelation times orders of magnitude smaller than hybrid Monte Carlo (HMC) on large volumes, maintains high importance-sampling efficiency relative to other generative baselines, and reproduces asymptotically unbiased physical observables in statistical agreement with long HMC simulations.

DOI: 10.1103/jjss-lwzv

I. INTRODUCTION

Markov chain Monte Carlo (MCMC) simulations are the standard framework for sampling lattice field theories and computing observables from first principles [1–3]. Their impact is particularly evident in lattice QCD, where large-scale simulations have enabled precision determinations of hadronic observables [4], as well as first-principles calculations of hadronic contributions to the muon anomalous magnetic moment [5,6]. At the same time, the reliability and scope of these calculations depend directly on the availability of algorithms capable of generating representative field configurations efficiently across a wide range of couplings, lattice spacings, and volumes.

A major limitation of MCMC methods is the rapid increase of computational cost near criticality, and more generally as the continuum limit is approached. As the correlation length grows in lattice units, slow long-wavelength modes come to dominate the dynamics and conventional Markov-chain algorithms suffer from severe critical slowing down [7–9]. Quantitatively, the integrated autocorrelation time typically scales as a power of the correlation length, $\tau_{\mathrm{int}} \sim \xi^z$, so the cost of producing statistically independent configurations rises rapidly with increasing system size and decreasing lattice spacing. In some theories, this problem is further compounded by the slow evolution of global or topological modes. Overcoming these dynamical bottlenecks is therefore essential for extending precision lattice calculations to larger volumes and finer lattices.

Motivated by these limitations, and by the rapid progress of deep learning methods in high-dimensional generative modeling, alternative sampling strategies based on neural networks have attracted increasing attention in lattice field theory. The central idea is to learn a parametric approximation of the target Boltzmann measure, thereby enabling

*Contact author: a.singha@tu-berlin.de
†Contact author: nakajima@tu-berlin.de

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

the generation of proposals with low autocorrelation. Among the corresponding deep generative models, normalizing flows (NFs) [10–13] have emerged as a particularly prominent framework. These algorithms have been explored in several forms, including coupling-layer flows [14–28], continuous normalizing flows [29–32], and stochastic normalizing flows [33–37]. Other generative approaches have also been studied, notably variational autoregressive networks [38–42], diffusion models [43–49], and generative adversarial networks [50–52].

However, their practical usefulness for large-scale lattice simulations is ultimately limited by scalability [53,54]. This limitation is particularly severe near criticality for single-scale architectures, where the model must learn the joint distribution of the entire lattice at a single resolution. Capturing the long-range correlations near criticality typically requires large receptive fields or very deep networks, leading to a rapid increase in training cost, memory usage, and sampling inefficiency near criticality as the system size grows.

Several strategies have been proposed to mitigate these issues. Among them, stochastic normalizing flows can improve scalability by introducing auxiliary non-equilibrium MCMC updates and effectively constructing non-equilibrium transport paths between simple and target distributions [36,37]. While this stochasticity can significantly enhance sampling efficiency at large volumes, it comes at the cost of requiring a predefined annealing schedule and a more involved generation procedure.

Normalizing flows can nevertheless be highly effective in specific applications. A notable example is the calculations of derivatives of physical observables, where flow-based changes of variables in the path integral can be optimized to reduce the variance of the corresponding estimators [21,23,55]. However, this advantage is tied to the optimization of specific observables and does not directly translate into scalable, fully generative samplers of the full lattice distribution. The tension between expressivity, scalability, and generation cost therefore remains a central challenge for their application to large-scale lattice field theory.

A natural route toward improving scalability is to exploit the multiscale structure of critical lattice systems. Under coarse graining, the slow infrared modes of the fine lattice are transferred to coarser levels, where they appear as shorter-wavelength modes in lattice units and can therefore be treated more efficiently. This idea underlies a long history of multiscale methods in lattice field theory and Monte Carlo simulations, including early multigrid Monte Carlo approaches [56–58] and simulation techniques inspired by renormalization groups [59–62]. In multigrid Monte Carlo, long-distance modes of the fine lattice are represented and updated on coarser grids, after which the resulting coarse-grid corrections are interpolated back to the fine lattice as collective updates. This strategy can significantly reduce critical slowing down in controlled settings, such as free or weakly interacting theories. However, it remains an acceleration method for fine-lattice MCMC rather than a direct coarse-to-fine generative scheme, and its performance can deteriorate when efficient coarse-grid updates become difficult to construct for the underlying theory [63] and also in the presence of a nontrivial global or topological structure [57].

Renormalization-group-inspired approaches [59–62], by contrast, proceed more directly in a coarse-to-fine direction, aiming to reconstruct fine configurations by sequentially introducing short-distance fluctuations from coarse variables. However, this requires renormalized effective densities at each scale, which are generally intractable beyond simple models. Deep generative models offer a way around this limitation by learning such effective distributions implicitly, without requiring their explicit construction. This idea has been successfully leveraged in Ref. [39], where variational autoregressive networks were used to model the coarse scale of the two-dimensional Ising model, leading to an efficient deep generative sampler with reduced training and sampling costs relative to single-scale models. Similar machine learning approaches have also been investigated in Refs. [40,42,64–67].

In this work, we generalize the multilevel generative strategy of Ref. [39], originally developed for discrete systems, to continuous lattice field theories. We introduce a multiscale generative sampling framework in which the Boltzmann measure is represented through a hierarchy of coarse-to-fine conditional densities such that long-range correlations are captured at coarse levels and short-distance fluctuations are introduced progressively during refinement. At each refinement step, a conditional Gaussian mixture model provides a tractable local approximation to the conditional distribution of the newly introduced fine degrees of freedom given the coarse field, and a continuous normalizing flow then refines this approximation toward the target conditional density. The resulting hierarchical construction enables scalable coarse-to-fine sampling at large lattice volumes, particularly in the critical regime where both conventional sampling methods and single-scale generative models become increasingly inefficient.

We demonstrate the method for two-dimensional scalar ϕ^4 lattice field theory near criticality. The resulting sampler reproduces physical observables in agreement, within statistical uncertainties, with hybrid Monte Carlo (HMC) reference runs, exhibits more favorable scaling with lattice size than existing generative baselines, and achieves orders-of-magnitude smaller autocorrelations than HMC on the largest lattices considered. Additionally, in our proposed method the coarse-to-fine hierarchy induces a built-in coupling between neighboring resolutions, providing exactly the structure needed to construct multilevel Monte Carlo (MLMC) estimators [62,68] and enabling further variance reduction at fixed computational cost.

The remainder of this paper is organized as follows. In Sec. II we review the sampling problem in continuous lattice systems near criticality. In Sec. III we introduce the multiscale generative framework and the associated multilevel estimator. In Sec. IV we present numerical results for the two-dimensional scalar ϕ^4 theory. Finally, Sec. V concludes with a broader discussion of possible future directions, including further developments of the multiscale generative framework and its application to more challenging lattice field theories.

II. BACKGROUND: SAMPLING LFT NEAR CRITICALITY

Consider lattice field theories defined on a hypercubic lattice $\Lambda_L = \{1, \ldots, L\}^D$ with continuous degrees of freedom $\phi_x \in \mathbb{R}$ at each site $x \in \Lambda_L$. A field configuration $\phi = \{\phi_x\}_{x\in\Lambda_L}$ is distributed according to the Boltzmann measure

$$p(\phi) = \frac{1}{Z}\exp[-S(\phi)], \tag{1}$$

where $S(\phi)$ is a local action functional and $Z = \int \mathcal{D}\phi e^{-S(\phi)}$ is the partition function. Physical observables are expectation values $\langle\mathcal{O}\rangle = Z^{-1}\int \mathcal{D}\phi\mathcal{O}(\phi)e^{-S(\phi)}$, which in practice are estimated numerically from ensembles of configurations generated by MCMC.

A standard workhorse algorithm for continuous-variable lattice systems is HMC [69], which combines molecular-dynamics trajectories with a Metropolis acceptance step to generate global proposals for the field configuration.

A. Critical slowing down

A central measure of sampling efficiency is the integrated autocorrelation time of an observable $\mathcal{O}$,

$$\tau_{\mathrm{int}}[\mathcal{O}] = \frac{1}{2} + \sum_{t=1}^{\infty}\frac{C_{\mathcal{O}}(t)}{C_{\mathcal{O}}(0)}, \tag{2}$$

where $C_{\mathcal{O}}(t) = \langle\mathcal{O}(\phi^{(0)})\mathcal{O}(\phi^{(t)})\rangle - \langle\mathcal{O}\rangle^2$ is the autocorrelation function of the Markov chain [8,70]. If M correlated configurations are available, the statistical error on $\langle\mathcal{O}\rangle$ scales as

$$\delta\langle\mathcal{O}\rangle \sim \sqrt{\frac{2\tau_{\mathrm{int}}[\mathcal{O}]}{M}}\sigma_{\mathcal{O}}, \tag{3}$$

so that the effective number of independent samples is $M_{\mathrm{eff}} = M/(2\tau_{\mathrm{int}})$. Achieving a fixed target precision therefore requires $M \propto \tau_{\mathrm{int}}$ configurations, making large autocorrelation times directly expensive.

Near a second-order phase transition or near a critical point, the correlation length $\xi \sim |g - g_c|^{-\nu}$ diverges as the coupling g approaches the critical value g_c. At finite volume, ξ is bounded by L, but the integrated autocorrelation time typically grows with system size as a power law,

$$\tau_{\mathrm{int}} \sim \xi^{z_{\mathrm{dyn}}} \sim L^{z_{\mathrm{dyn}}}, \tag{4}$$

where z_{dyn} is the dynamical critical exponent of the algorithm [7,9]. For HMC applied to scalar field theories, $z_{\mathrm{dyn}} \approx 2$ [69], so that the cost of obtaining M_{eff} independent samples scales as

$$\mathcal{C}_{\mathrm{HMC}} \sim M_{\mathrm{eff}}\tau_{\mathrm{int}}L^D \sim M_{\mathrm{eff}}L^{D+z_{\mathrm{dyn}}} = M_{\mathrm{eff}}L^{D+2}. \tag{5}$$

This power-law growth becomes prohibitive near criticality at large lattice volumes, where the long-distance physics of interest resides. It therefore constitutes a central bottleneck for scalable lattice simulations and motivates sampling strategies that explicitly exploit scale separation in configuration space.

III. MULTISCALE GENERATIVE SAMPLING

A. Renormalization of the action

Renormalization-group (RG) transformations provide a natural framework for describing the scale dependence of lattice field theories by systematically integrating out short-distance degrees of freedom. Consider a lattice field configuration $\phi \in \mathbb{R}^V$ distributed according to the Boltzmann measure

$$p(\phi) = \frac{1}{Z}\exp\left[-S(\phi;\{K_i^{(0)}\})\right], \tag{6}$$

where $S(\phi;\{K_i^{(0)}\})$ denotes the microscopic action parametrized by a set of couplings $\{K_i^{(0)}\}$. The effective coarse-grained action $S^{(1)}$ is defined implicitly by integrating out fine degrees of freedom,

$$\exp\left[-S^{(1)}(\phi^{(1)})\right] \propto \int \mathcal{D}\phi\,\delta(\phi^{(1)} - B(\phi)) \times \exp\left[-S(\phi;\{K_i^{(0)}\})\right], \tag{7}$$

where $\phi^{(1)} = B(\phi)$ is a blocking map from the fine lattice to a coarser one. In general, the renormalized action takes the form

$$S^{(1)}(\phi^{(1)}) = \sum_i K_i^{(1)}\mathcal{O}_i(\phi^{(1)}). \tag{8}$$

Iterating this construction yields a sequence of effective actions $\{S^{(\ell)}\}$ on lattices of progressively lower resolution, together with corresponding couplings $\{K_i^{(\ell)}\}$. Under RG flow, the operator basis $\{\mathcal{O}_i\}$ grows rapidly, so that the effective action generally acquires higher-order and increasingly nonlocal interactions even when the

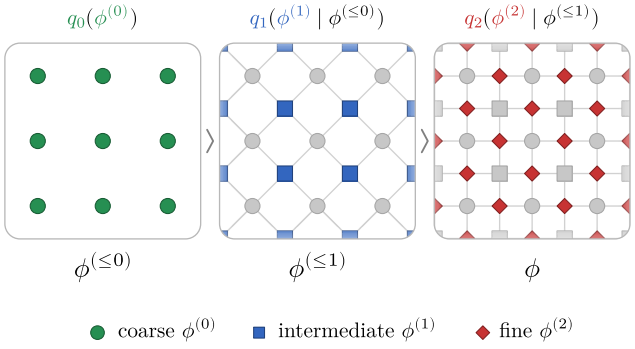


FIG. 1. Kadanoff-inspired multilevel partition of a periodic lattice into coarse [$\phi^{(0)}$, circles], intermediate [$\phi^{(1)}$, squares], and fine [$\phi^{(2)}$, diamonds] sites, illustrating the coarse-to-fine factorization (11). Each panel samples new variables (colored markers) conditioned on previously sampled sites (gray markers); connections show the nearest-neighbor dependencies used by the conditional models q_ℓ.

microscopic action is local. Although the exact coarse-grained action is therefore intractable in practice, the RG picture motivates a hierarchical construction in which different resolutions are modeled separately.

B. Multiscale generative decomposition

The RG picture motivates a reverse, coarse-to-fine construction in which configurations are generated from low to high resolution. Concretely, we introduce a multilevel partition of the lattice sites inspired by Kadanoff blocking [71] (see Fig. 1),

$$\phi \equiv \phi^{(\leq \ell_{\max})} \to \{\phi^{(0)}, \phi^{(1)}, \ldots, \phi^{(\ell_{\max})}\}, \tag{9}$$

where $\phi^{(0)}$ denotes the coarsest degrees of freedom and $\phi^{(\leq \ell_{\max})}$ corresponds to the full field on the finest lattice. The exact distribution then admits the coarse-to-fine factorization

$$p(\phi) = p(\phi^{(0)}) \prod_{\ell=1}^{\ell_{\max}} p\left(\phi^{(\ell)} | \phi^{(\leq \ell-1)}\right). \tag{10}$$

This decomposition would require access to the effective action at every level. However, as discussed above, these actions are generically nonlocal and not known in closed form. Working with an approximate effective action and correcting at the finest level by a Metropolis–Hastings step typically results in a very low acceptance rate [59], which has limited the practical use of such constructions in the traditional MCMC setting [62].

We therefore replace the unknown conditionals $p(\phi^{(\ell)}|\phi^{(\leq \ell-1)})$ by the learned ones and define a hierarchical generative model

$$q(\phi) = q_0(\phi^{(0)}) \prod_{\ell=1}^{\ell_{\max}} q_\ell\left(\phi^{(\ell)} | \phi^{(\leq \ell-1)}\right), \tag{11}$$

where $\phi^{(\leq \ell)} = \{\phi^{(0)}, \phi^{(1)}, \ldots, \phi^{(\ell)}\}$ collects all variables up to level ℓ. Sampling proceeds coarse-to-fine where one first draws $\phi^{(0)}$ from q_0, and then successively draws the newly introduced variables at each finer scale from the conditional models q_ℓ. Each q_ℓ is parametrized as a flow conditioned on $\phi^{(\leq \ell-1)}$, so that $q(\phi)$ admits a tractable density and the entire model can be trained by reverse Kullback–Leibler (KL) divergence minimization against the target $p(\phi)$.

A practical advantage of our decomposition is the locality. When variables at level ℓ are sampled, their nearest neighbors in the fine lattice have already been drawn at coarser levels and are contained in $\phi^{(\leq \ell-1)}$. The conditional models q_ℓ therefore only need to capture short-range correlations around the newly introduced sites, and can be implemented with small receptive fields. This should be contrasted with a single-scale flow acting on the full fine field, which must learn correlations across all length scales simultaneously for which the cost typically grows unfavorably near criticality.

Hierarchical generative constructions of related form have recently appeared in the literature. Reference [39] developed a learned multilevel sampler for discrete spin systems, where coarse-to-fine conditionals are categorical and can be parametrized directly; further variants in the discrete setting are explored in Ref. [41]. In the continuous case, Refs. [65,66] introduced multiscale flows with a different partition of the degrees of freedom and a different training objective. Our construction targets LFT in the regime where long-range correlations dominate (i.e. near criticality), and differs from these works in both the coarse-to-fine partition and in the form of the conditional models q_ℓ, which we describe in the following Subsections.

C. Approximate local conditional sampler with Gaussian mixture models

At intermediate and finest levels (Fig. 1), we approximate the exact coarse-to-fine conditional distributions using local conditional models, which later be used as prior distribution for the refinement process. Concretely, we assume that the newly introduced variables at level ℓ are conditionally independent given the already-sampled neighboring variables in $\phi^{(\leq \ell-1)}$, and model this approximation with a conditional Gaussian mixture model (GMM),

$$q_\ell^{\mathrm{GMM}}\left(z^{(\ell)} | \phi^{(\leq \ell-1)}\right) = \prod_{x \in \Lambda^{(\ell)}} q_{\ell,x}^{\mathrm{GMM}}\left(z_x^{(\ell)} | \phi^{(\leq \ell-1)}_{\mathcal{N}(x)}\right), \tag{12}$$

where $\Lambda^{(\ell)}$ are the lattice sites to be sampled at level l and

$$q_{\ell,x}^{\mathrm{GMM}}\left(z_x^{(\ell)}|\phi_{\mathrm{NN}(x)}^{(\leq \ell-1)}\right)=\sum_{k=1}^{K}\pi_{k,x}^{(\ell)}\mathcal{N}\left(z_x^{(\ell)}|\mu_{k,x}^{(\ell)},\sigma_{k,x}^{2(\ell)}\right). \quad (13)$$

Here, the mixture weights $\pi_{k,x}^{(\ell)}$, means $\mu_{k,x}^{(\ell)}$, and variances $\sigma_{k,x}^{2(\ell)}$ are functions of the neighboring coarse variables $\phi_{\mathrm{NN}(x)}^{(\leq \ell-1)}$ and are parametrized by fully convolutional neural networks.

The GMM provides an explicit, tractable density that can be sampled and evaluated efficiently, making it well suited as a local conditional prior within the hierarchical framework. It also allows parallel sampling across lattice sites.

D. Conditional sampling with continuous normalizing flows

Although the GMM captures the leading local conditional structure at each refinement level, it does not represent the full target conditional distribution. To model the remaining mismatch, we refine GMM samples $z^{(\ell)}$ with a conditional continuous normalizing flow (CNF) [13]. Specifically, we define an invertible transformation

$$\phi^{(\ell)}=f_\ell(z^{(\ell)};\phi^{(\leq \ell-1)}). \quad (14)$$

The map f_ℓ is generated by integrating a conditional neural ordinary differential equation (ODE),

$$\frac{d\phi(t)}{dt}=v_\ell\left(\phi(t),t;\phi^{(\leq \ell-1)}\right), \qquad t\in[0,1], \quad (15)$$

with $\phi(0)=z^{(\ell)}$ and $\phi(1)=\phi^{(\ell)}$. The corresponding conditional density is

$$\log q_\ell(\phi^{(\ell)}|\phi^{(\leq \ell-1)})=\log q_\ell^{\mathrm{GMM}}(z^{(\ell)}|\phi^{(\leq \ell-1)}) - \int_0^1 dt\nabla\cdot v_\ell\left(\phi(t),t;\phi^{(\leq \ell-1)}\right). \quad (16)$$

Together, the GMM and CNF provide a tractable approximation to each conditional factor in Eq. (10).

E. Continuous multilevel sampler

The full multilevel model is obtained by combining the ingredients introduced in the previous sections across all resolutions according to Eq. (11). At the coarsest level, the unconditional model q_0 is a CNF with a standard Gaussian prior. At each finer level, the conditional model q_ℓ is a continuous normalizing flow whose prior is the corresponding conditional GMM. What distinguishes our construction from existing multilevel samplers is the combination of three ingredients: (i) our coarse-to-fine decomposition of the lattice sites across scales, (ii) a fast, independently samplable conditional GMM that captures the dominant local structure of the newly introduced sites at each upsampling step, and (iii) a lightweight CNF that learns the residual inter-site correlations on top of the GMM prior. The full architecture, along with a comparison to related approaches such as the Super Resolving NF (SR-NF) [66], is described in Appendix B.

The model parameters of the multilevel sampler are optimized by minimizing the reverse Kullback-Leibler divergence between the hierarchical model and the target distribution,

$$\mathcal{L}=\mathrm{KL}(q(\phi)\|p(\phi)). \quad (17)$$

Once trained, the model generates samples by ancestral coarse-to-fine sampling, thereby avoiding Markov-chain autocorrelations and enabling efficient exploration of configuration space near criticality.

F. Multilevel control variates for variance reduction

CNF samplers are typically used as single-level importance samplers where one generates N configurations from the learned distribution $q^{(L)}$ on an $L\times L$ lattice and computes the self-normalized importance sampling (IS) estimator

$$\hat{\mathcal{O}}_{\mathrm{IS}}=\sum_{n=1}^{N}\bar{w}_n\mathcal{O}(\phi_n), \quad (18)$$

where

$$w(\phi)=\frac{e^{-S(\phi)}}{q(\phi)}, \qquad \bar{w}_n=\frac{w(\phi_n)}{\sum_{m=1}^{N}w(\phi_m)}. \quad (19)$$

Its variance scales as $\mathrm{Var}_{\mathrm{IS}}\propto 1/N$. The difficulty is that sampling at the finest level is expensive and each sample requires integrating all upsampling CNFs over the full L^2-dimensional space, accordingly the per-sample cost grows roughly as $c_L\sim L^2$. Achieving a fixed target precision, therefore, becomes increasingly costly as the lattice size grows.

Multilevel Monte Carlo [68] addresses this by exploiting the fact that expensive fine-level evaluations are only needed to correct cheaper coarse-level estimates. Since coarser levels have costs $c_\ell\ll c_L$, many more samples can be generated for the same computational budget.

Finest-level weighted estimator. In practice, the target marginals $p^{(\ell)}$ are generally unavailable at intermediate levels. We therefore define importance weights only at the finest level,

$$w^{(L)}(\phi)=\frac{e^{-S(\phi)}}{q(\phi)}, \qquad \bar{w}_n^{(L)}=\frac{w^{(L)}(\phi_n)}{\sum_{m=1}^{N_L}w^{(L)}(\phi_m)}. \quad (20)$$

Given samples $\{\phi_n^{(\leq \ell)}\}_{n=1}^{N_\ell}$ from the hierarchical model $q^{(\ell)}(\phi^{(\leq \ell)})$, and level-dependent observables $\mathcal{O}^{(\ell)}$ with $\mathcal{O}^{(L)}=\mathcal{O}$ and $\mathcal{O}^{(-1)}\equiv 0$ we define the estimator

$$\hat{\mathcal{O}}_{\mathrm{MLMC}} = \sum_{n=1}^{N_L}\left[\tilde{w}_n^{(L)}\mathcal{O}^{(L)}(\phi_n^{(\leq L)}) - \frac{1}{N_L}\mathcal{O}^{(L-1)}(\phi_n^{(\leq L-1)})\right] + \sum_{\ell=0}^{L-1}\frac{1}{N_\ell}\sum_{n=1}^{N_\ell}\left[\mathcal{O}^{(\ell)}(\phi_n^{(\leq \ell)}) - \mathcal{O}^{(\ell-1)}(\phi_n^{(\leq \ell-1)})\right]. \tag{21}$$

Its efficiency depends on the interlevel correlations of $\mathcal{O}^{(\ell)}$ and $\mathcal{O}^{(\ell-1)}$ reflecting the variance of each term in the form of $\mathcal{O}^{(\ell)} - \mathcal{O}^{(\ell-1)}$. Near criticality, the coupling between successive levels is strong, so the finer level terms have lower variances and can be accurately estimated with fewer samples. Consequently, the overall computational budget is reduced, which makes the proposed MLMC estimator especially well suited to hierarchical sampling in the critical regime.

IV. NUMERICAL EXPERIMENTS

A. Model and simulation setup

We study the two-dimensional scalar ϕ^4 lattice field theory near criticality as a representative continuous lattice system with local interactions. The theory is defined on a square lattice of linear size L with periodic boundary conditions where a real-valued field variable $\phi_x \in \mathbb{R}$ is assigned at each lattice site x.

The Euclidean lattice action is

$$S(\phi) = \sum_x [(1-2\lambda)\phi_x^2 + \lambda\phi_x^4] - 2\kappa\sum_{x,\mu}\phi_x\phi_{x+\hat{\mu}}, \tag{22}$$

where κ denotes the nearest-neighbor coupling, λ controls the quartic self-interaction, and the second sum runs over nearest neighbors in the two lattice directions $\mu = 1, 2$. The action is invariant under the global $\mathbb{Z}_2$ symmetry $\phi_x \to -\phi_x$, which is spontaneously broken in the ordered phase. The model undergoes a second-order phase transition between symmetric and broken phases and therefore provides a standard benchmark for critical sampling in continuous lattice field theories.

We determine the critical coupling κ_c from the Binder cumulant [72],

$$U_4 = 1 - \frac{\langle m^4\rangle}{3\langle m^2\rangle^2}, \qquad m = L^{-2}\sum_x \phi_x. \tag{23}$$

As a challenging benchmark point, we choose $\kappa_c = 0.2705 \pm 0.0007$, estimated from the crossing of the Binder cumulant for $L = 32$ and $L = 64$. This estimate is used only to define a representative near-critical simulation point; our aim is not a precision determination of the critical coupling, but to study the method in a regime where critical slowing down is particularly pronounced. Figure 2 shows $U_4(\kappa)$ obtained from 5×10^5 HMC configurations

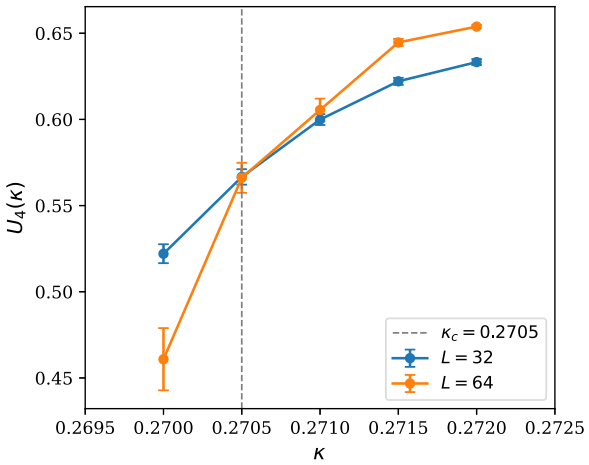


FIG. 2. Binder cumulant $U_4(\kappa)$ for $L = 32$ and $L = 64$ from HMC (5×10^5 configurations each). The curves cross at $\kappa_c = 0.2705$ (dashed vertical line). Error bars are blocked jackknife estimates.

for each lattice size. The curves cross at $\kappa_c = 0.2705$, where $U_4(L = 32) = 0.567$ and $U_4(L = 64) = 0.566$ differ by only 0.001 ($\ll 1\sigma$). This is also the regime where HMC critical slowing down is severe, with $\tau_{\mathrm{int}}(|m|) \approx 4416$ at $L = 128$ (Appendix A).

We evaluate the proposed sampler near the critical point on lattices of size $L \in \{16, 32, 64, 128, 256\}$. As a benchmark observable, we consider the absolute magnetization

$$|m| = \left|L^{-2}\sum_x \phi_x\right|, \tag{24}$$

and the connected susceptibility

$$\chi_{\mathrm{conn}} = L^2(\langle m^2\rangle - \langle m\rangle^2). \tag{25}$$

We use HMC as the reference baseline for three purposes. First, the HMC runs provide reference estimates of physical observables, such as χ_{conn} and $\langle|m|\rangle$, against which the generative samplers are tested for statistical bias. Second, they determine the autocorrelation time $\tau_{\mathrm{int}}(|m|)$, which quantifies the severity of the critical slowdown at each lattice size. Third, they define the reference computational target i.e. we measure how much wall-clock time each generative method requires to match the effective number of independent samples obtained by HMC. For $L \leq 128$ we run HMC chains of 2×10^6 steps; for $L = 256$ we run a chain of 3×10^6 steps. Autocorrelation times are estimated with the Wolff method [8,70]. Further details of the HMC simulations are given in Appendix A.

Generative methods. We compare four flow-based samplers, all trained at $\kappa_c = 0.2705$ with the reverse-KL objective:

(i) *Dense CNF* [30]: A single CNF on the target distribution with a Gaussian prior, evaluated at $L \in \{16, 32, 64\}$; extending training beyond $L = 64$ becomes computationally impractical (see Appendix B 3).

(ii) *Hutch CNF*: A CNF with multiple CNN layers that uses the Hutchinson stochastic trace estimator [73] to estimate the divergence stochastically, evaluated at $L \in \{16, 32\}$; scaling to larger lattices becomes increasingly difficult (see Appendix B 4).

(iii) *Super resolving CNF (SR-NF)* [66]: A hierarchical sampler based on a *rectangular-flow* construction [66] with coarsest level HMC sampling replaced by a plain CNF sampler. At each super-resolution stage, a coarse $L_\ell \times L_\ell$ field is first deterministically block replicated to a fine $2L_\ell \times 2L_\ell$ lattice, then augmented with zero-sum Gaussian noise that injects the extra degrees of freedom, and finally evolved through a full-lattice CNF that transports all sites jointly. We evaluated SR-NF at $L \in \{16, 32, 64, 128, 256\}$ (see Appendix B 2 for more details).

(iv) *Our multilevel method*: The multilevel coarse-to-fine sampler introduced in Sec. III. We evaluated our method at $L \in \{16, 32, 64, 128, 256\}$ (see Appendix B 1 for more details).

For each model, physical observables are estimated from $N = 10^5$ samples using self-normalized importance sampling (IS) Eq. (19). Autocorrelation times for the generative samplers are measured from independent Metropolis-Hastings (IMH) chains of 5×10^4 steps after a burn in of 2×10^3 steps, using the Sokal windowing estimator [8]. Timing measurements for all generative methods are reported as wall-clock time on the same NVIDIA A100 graphics processing unit (GPU), while the HMC timings are measured on the CPU hardware used for the reference simulations.

B. Mixing and autocorrelation time

Table I reports the integrated autocorrelation time $\tau_{\text{int}}(|m|)$ for all methods across lattice sizes. For HMC, the dynamical exponent $z = 1.81 \pm 0.14$ is obtained from a power-law fit to all six measured points $L \in \{8, 16, 32, 64, 128, 256\}$ (see Appendix A).

Three regimes can be distinguished. At small volumes ($L \leq 32$), our method, Dense CNF, and SR-NF all achieve $\tau_{\text{int}} < 3$, indicating very low autocorrelation compared with the HMC values $\tau_{\text{int}} = 64.6 \pm 1.3$ and 228 ± 9. Hutch CNF is already substantially worse ($\tau_{\text{int}} = 25.7 \pm 1.2$ at $L = 16$), reflecting the additional variance introduced during training by stochastic log-determinant estimation.

At $L = 64$, the methods begin to separate. Our method maintains $\tau_{\text{int}} = 1.19 \pm 0.01$, Dense CNF remains competitive at 0.98 ± 0.01, whereas SR-NF rises to 5.32 ± 0.11, signaling the onset of degraded mixing as the full-lattice CNF struggles to represent the growing configuration space. The contrast becomes much stronger at larger volumes where at $L = 128$ and $L = 256$, our method yields $\tau_{\text{int}} = 3.69 \pm 0.06$ and 72.2 ± 5.5, respectively, while SR-NF yields 186 ± 23 at $L = 128$ and 3615 ± 1944 at $L = 256$ (the latter has large uncertainty as shown in Table I, note the SR-NF's IMH acceptance rate is $\sim 0.05\%$ at $L = 256$, yielding only ≈ 7 effective samples).

Even at $L = 256$, our method remains roughly $116\times$ below the measured HMC value of 8410 ± 1991 demonstrating that the hierarchical construction substantially mitigates critical slowing down at lattice sizes where conventional HMC becomes prohibitively expensive. Low autocorrelation is meaningful only when accompanied by sufficient importance-sampling efficiency; the ESS$/N$ values reported in Fig. 5 confirm that our method maintains ESS$/N \geq 0.20$ up to $L = 128$, where SR-NF drops to ESS$/N \approx 4 \times 10^{-3}$ at the same scale, rendering its low τ_{int} practically unusable at large volumes.

C. Physical observable consistency

A generative sampler is only useful if its importance-weighted estimates remain statistically consistent with the target distribution. We test this by comparing importance-sampling estimates of the connected susceptibility χ_{conn}, obtained from $N = 10^5$ samples, with the long HMC reference runs. Figure 3 shows the relative deviation $(\chi_{\text{ML}} - \chi_{\text{HMC}})/\chi_{\text{HMC}}$, with gray bands indicating the uncertainty $1/2/3\sigma$ of the HMC reference.

TABLE I. Integrated autocorrelation time $\tau_{\text{int}}(|m|)$ at $\kappa_c = 0.2705$, $\lambda = 0.022$. HMC: Wolff method on 2×10^6-step chains ($L \leq 128$) and 3×10^6 steps ($L = 256$). Generative methods: IMH chains of 5×10^4 steps, 2×10^3 burn in. The corresponding importance-sampling efficiency ESS$/N$ for all methods is reported in Fig. 5 (Sec. IV D).

Method	$L = 16$	$L = 32$	$L = 64$	$L = 128$	$L = 256$
HMC	64.6 ± 1.3	228 ± 9	1370 ± 120	4416 ± 928	8410 ± 1991
Our method	0.76 ± 0.01	0.85 ± 0.01	1.19 ± 0.01	3.69 ± 0.06	72.2 ± 5.5
SR-NF	0.89 ± 0.01	1.50 ± 0.02	5.32 ± 0.11	186 ± 23	3615 ± 1944
Dense CNF	0.70 ± 0.01	0.76 ± 0.01	0.98 ± 0.01	⋯	⋯
Hutch CNF	25.7 ± 1.2	55.4 ± 3.7	⋯	⋯	⋯

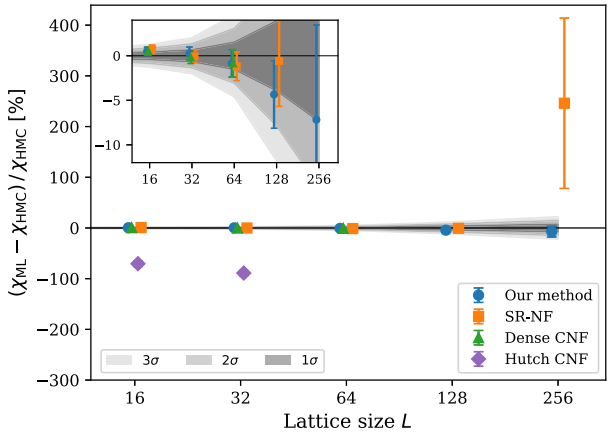


FIG. 3. Relative deviation of the connected susceptibility from HMC, $(\chi_{ML} - \chi_{HMC})/\chi_{HMC}$, versus lattice size L at $\kappa_c = 0.2705$. Gray bands show the 1σ, 2σ, and 3σ HMC uncertainty intervals for visual guidance. For clearer visualization of the error bars, we also included an enlarged view.

The Dense CNF [30] and our method reproduce the HMC reference values for χ_{conn} within $1-2\sigma$ across all tested lattice sizes ($L = 16$ to 256), indicating no statistically significant bias at our level of precision. By contrast, SR-NF estimates become increasingly unreliable in large volumes: at $L = 256$, where IS efficiency drops below 0.004%, the reweighted χ_{conn} acquires uncertainties that exceed the central value, and the mean deviates well beyond the HMC error bands. The Hutchinson-trace CNF exhibits substantial mode-dropping already at both $L = 16, 32$, rendering its IS estimates systematically biased. Our method, in contrast, maintains a subpercent relative deviation from HMC up to $L = 256$, providing stable and accurate observable estimates at scales where the baseline methods become unreliable or nontrainable.

We also compare the histograms of the signed magnetization m, the absolute magnetization $|m|$, the action density S/V, and the connected two-point function of the time-slice field $C(t) = \langle s(0)s(t)\rangle - \langle s\rangle^2$, $s(x_1) = L^{-1}\sum_{x_2}\phi_{x_1,x_2}$, with HMC simulations.

For computing these observable, we do independent Metropolis-Hastings with our model as a proposal (acceptance ≈0.76). Figure 4 shows that the distributions of our method overlap the HMC ones for all four observables (e.g. $\langle S/V\rangle = 0.3931 \pm 0.0001$ vs 0.3931 ± 0.0004 and $\langle|m|\rangle = 0.7156 \pm 0.0009$ vs 0.7173 ± 0.0066), and that the two-point functions agree within errors across the accessible range of t.

D. IS efficiency comparison: ESS

Effective sample size (ESS) per sample is computed as

$$\mathrm{ESS}/N = \frac{1}{N\sum_{i=1}^{N}\bar{w}_i^2},$$

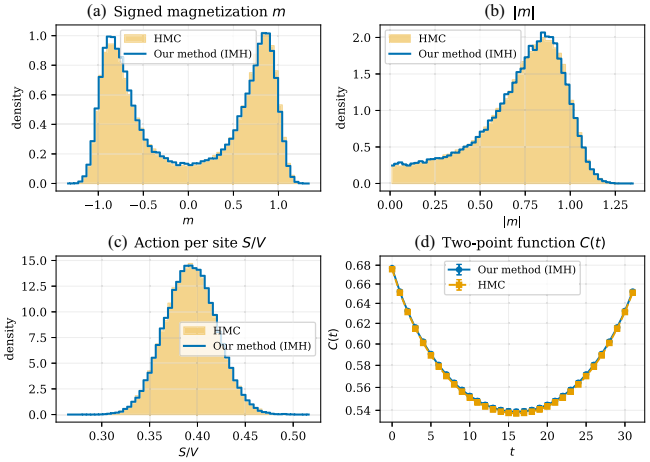


FIG. 4. Observables at the finest lattice ($L = 32$, κ_c) by our method (blue) and HMC (orange): (a) signed magnetization m, (b) absolute magnetization $|m|$, (c) action density S/V and (d) the connected two-point function $C(t)$ of the time-slice field computed from configurations drawn from the model by independent Metropolis-Hastings sampling (acceptance ≈0.76).

where $\bar{w}_i$ is defined in Eq. (19). Figure 5 shows the IS efficiency ESS/N at $N = 10^5$ samples for all four methods. Our method maintains $\mathrm{ESS}/N \geq 66\%$ up to $L = 64$ and 24% and 0.8% at $L = 128, 256$, respectively, while SR-NF drops to 0.4% at $L = 128$ and collapses to 0.04% at $L = 256$. Dense CNF is competitive up to $L = 64$ ($\mathrm{ESS}/N = 0.71$) but cannot be trained at larger volumes. Hutch CNF achieves near-zero efficiency even at small L (0.5% at $L = 16$) reflecting noisier training signal from the stochastic log determinant.

E. Time to match HMC precision

As practical performance of a sampler depends not only on sample quality but also on how quickly samples can be

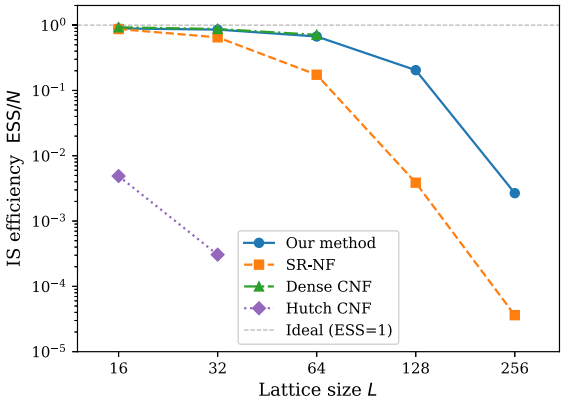


FIG. 5. IS efficiency ESS/N vs L at $\kappa_c = 0.2705$. Evaluations use $N = 10^5$ IS samples per method. Our method (blue circles) maintains high efficiency across all evaluated sizes; SR-NF (orange squares) collapses at large L; Dense CNF (green triangles) is limited to $L \leq 64$; Hutch CNF (purple diamonds) suffers from stochastic-estimator noise.

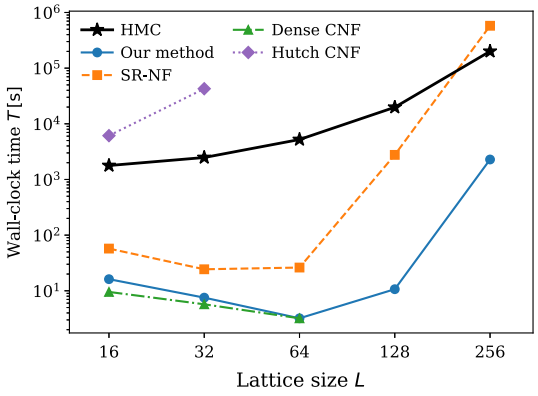


FIG. 6. Wall-clock time required to match the effective number of independent samples obtained in the HMC reference run, as a function of lattice size L [Eq. (27)]. The black line shows the actual HMC wall-clock time.

generated, both ESS and the sample-generation rate must be taken into account when assessing a generative sampler. To capture this trade-off, we measure the wall-clock time required by each method to produce the same effective number of independent samples as the HMC reference run.

At each lattice size L, the HMC reference uses N_{HMC} steps (2×10^6 for $L \leq 128$, 3×10^6 for $L = 256$), corresponding to

$$N_{\mathrm{eff}}^{\mathrm{HMC}} = \frac{N_{\mathrm{HMC}}}{2\tau_{\mathrm{int}}} \tag{26}$$

effectively independent samples. For a generative method with IS efficiency ESS/N and sample-generation rate Γ, the time required to accumulate the same number of effective samples is

$$T = \frac{N_{\mathrm{eff}}^{\mathrm{HMC}}}{(\mathrm{ESS}/N)\Gamma}. \tag{27}$$

The HMC wall-clock time

$$T_{\mathrm{HMC}} = N_{\mathrm{HMC}} t_{\mathrm{step}} \tag{28}$$

is measured on the same CPU hardware used for the reference chains, with t_{step} benchmarked from 500 HMC steps at each lattice size.

Figure 6 compares the wall clock time of generating samples with the HMC reference. Across these lattice sizes, the generative methods stay below the HMC baseline, suggesting a practical computational advantage. Our method achieves speedups of 106× at $L = 16$, 1653× at $L = 64$, and 211× at $L = 256$ relative to HMC. SR-NF is competitive at small L but rises close to the HMC line at $L = 128$ and 256, where its ESS drops to 0.4% and 0.04% respectively.

The inference-time comparison above excludes the one-time offline training cost. GPU training times for all methods and lattice sizes are reported in Table VI (Appendix B 7). At the largest lattice considered, our method requires 11.5 GPU hours at $L = 256$. The evolution of the training ESS/N as a function of GPU time for our method and SR-NF is shown in Fig. 11 (Appendix E), demonstrating that our masked architecture reaches higher ESS in less training time at both $L = 64$ and $L = 128$. Dense CNF remains faster than HMC for $L \leq 64$ but cannot be trained beyond that scale; the memory scaling underlying this limitation is analyzed in Appendix F.

F. Performance across a wider range of action parameters

We perform a broader scan in κ and compare the importance-sampling efficiency of our multilevel sampler with the SR-NF baseline at $L = 32$ across 11 values of κ, spanning the symmetric phase, the critical point $\kappa_c = 0.2705$, and the broken phase (Fig. 7). Seven values are chosen around the critical region, where the correlation length is largest and sampling is most challenging, while four additional values probe the deep symmetric phase ($\kappa = 0.24,\ 0.2505$) and the deep broken phase ($\kappa = 0.2905,\ 0.30$).

Our multilevel sampler achieves a higher ESS/N than SR-NF near criticality, and across the entire near-critical region. The improvement is most pronounced where long-range correlations are strongest; for example, ESS/N = 0.84 for our multilevel compared with 0.67 for SR-NF at κ_c, and 0.95 compared with 0.81 at $\kappa = 0.2505$.

At the four points farthest from criticality, both models perform well, as expected, since the correlation length is of the order of the lattice spacing. In the deep symmetric phase, at $\kappa = 0.24$ and $\kappa = 0.2505$, our method outperforms SR-NF, whereas in the deep broken phase, at $\kappa = 0.2905$ and $\kappa = 0.30$, SR-NF gives a higher ESS/N. Nevertheless, both methods remain effective in these regimes. The main motivation for the multilevel construction, however, is to capture long-range correlations through hierarchical coarse-to-fine generation and thereby alleviate

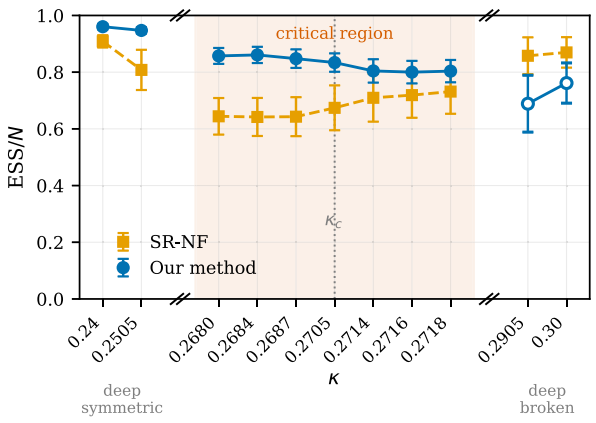


FIG. 7. Importance-sampling efficiency ESS/N at $L = 32$ across different values of κ, comparing our method to the SR-NF baseline.

critical slowing down. This is precisely where our method shows its main advantage: in the near-critical region, where the correlation length becomes large, it consistently outperforms the SR-NF baseline. We do not include comparisons with other baseline models, as these approaches are not designed to scale systematically to larger lattice sizes.

G. A qualitative picture of the performance gain

Reproducing long-range correlations is one of the central challenges that our sampler is designed to address near criticality. Although SR-NF also follows a multilevel coarse-to-fine strategy, at each refinement level it replicates the coarse configuration onto the finer lattice, adds noise, and applies a flow that updates the entire field without further forcing the mean zero constraint. The inherited coarse values are therefore not preserved, and the long-wavelength structure may need to be transported again at every level, including on the finest lattice. In our construction, the coarse sites are instead embedded into the finer lattice and kept fixed, while masked flows generate only the newly introduced sites conditioned on their frozen coarse neighborhood. Long-range correlations established on the coarse lattice are thus inherited directly, leaving each refinement flow to model primarily the local conditional structure or UV fluctuations.

In our sampler, each refinement level is further divided into two stages, center sites are generated first, followed by edge sites conditioned on both the frozen coarse field and the newly generated center sites as shown in Fig. 1. In contrast, the single-step upsampling used in SR-NF requires a single flow to reconstruct the full fine-lattice structure at once, which we find difficult to stabilise with a small number of ODE steps. The ablation studies in Appendix C show that the conditional GMM prior plays an important role in stabilising this training. By placing the newly introduced sites close to their conditional target distributions before the flow acts, the GMM prior enables stable and efficient sampling with a relatively small ODE integration budget.

Overall, these results show that our hierarchical coarse-to-fine construction provides a substantial computational advantage near criticality when both sampling efficiency and generation cost are taken into account.

H. Multilevel Monte Carlo variance reduction

A key advantage of our hierarchical construction is that it naturally enables MLMC estimation. Now we compare the MLMC estimator defined in (21) against plain importance sampling (IS) at equal wall-clock budget,

$$T = N_{\mathrm{IS}} c_L,$$

with $N_{\mathrm{IS}} = 10^5$, where c_L is the measured per-sample cost at the finest level. Per-level costs c_ℓ and correction variances σ_ℓ^2 are estimated from a probe run, and the allocation of the sample number is then chosen according to the standard MLMC rule

$$N_\ell \propto \sqrt{\sigma_\ell^2 / c_\ell}. \tag{29}$$

As an observable, we use absolute magnetization $|m|$. We compute our estimator for $L = 32$ and $L = 64$, with the coarsest level fixed at $L_0 = 2$. Table II summarizes the results of our estimator. The MLMC estimator achieves lower variance than plain IS at the same computational budget at both lattice sizes. At $L = 32$, the variance reduction corresponds to a 38% savings in wall-clock time for a fixed precision target. At $L = 64$, the speedup is 20% and the lower gain at $L = 64$ can be traced to the reduction in the finest-level of ESS, from $\mathrm{ESS}/N = 85\%$ at $L = 32$ to 66% at $L = 64$. The achievable gain is currently limited by the quality of the finest-level model, indicating that further improvements in fine-level modeling could directly enhance the MLMC speedup. So, maintaining a sufficiently high ESS at larger volumes is therefore necessary to obtain a scalable MLMC improvement. Note that in SR-NF the observables at adjacent levels do not couple as strongly as in our decomposition, and hence the MLMC estimator does not provide any computational gain over the plain IS estimator. The possibility of applying MLMC to SR-NF is discussed in Appendix G 2.

V. CONCLUSION

We have introduced a multiscale generative sampling framework for continuous lattice field theories near criticality. The method combines a hierarchical coarse-to-fine factorization of the target distribution with learned local conditional priors and flow-based refinement across resolutions. The important long-range correlation is modeled at coarse levels, whereas finer resolutions are generated through conditional GMM priors and continuous normalizing flows that refine the newly introduced variables while preserving the embedded coarse sites.

Applied to the two-dimensional scalar theory ϕ^4 near criticality, the proposed method shows clear advantages over all flow-based baselines considered in this work. It achieves substantially lower integrated autocorrelation times at the large lattice sizes, while maintaining higher ESS than

TABLE II. Comparison of plain IS and MLMC estimators for $\langle |m| \rangle$ using our method at $\kappa_c = 0.2705$, evaluated at equal computational budget T.

L	T (s)	IS estimate	MLMC estimate	Variance reduction
32	614	0.7143 ± 0.00085	0.7130 ± 0.00067	38%
64	1266	0.5982 ± 0.00102	0.5996 ± 0.00091	20%

baseline generative models. This advantage is also reflected in wall-clock efficiency, where among all generative baselines considered, our method requires the least time to match the effective precision of the HMC reference, with the benefit becoming most pronounced at large lattice sizes. Physical observables are reproduced in statistical agreement with long HMC reference simulations across all tested lattice sizes. In addition, the hierarchical construction enables the use of an MLMC estimator, which provides variance reduction near criticality beyond plain importance sampling.

Several directions remain open. The decrease in ESS at the largest volumes suggests that further gains may be possible through larger models or improved training at the finest levels, which would likely also enhance the attainable MLMC speedup. More broadly, the present results establish hierarchical generative sampling as a promising strategy for continuous lattice systems in the critical regime. An important next step is the extension of this framework to continuous group-valued theories such as lattice gauge theories. A natural extension would be to formulate the coarse-to-fine hierarchy directly in terms of plaquette, or field-strength variables rather than link variables. This would preserve the desirable feature that each lattice site carries only a small number of local degrees of freedom, allowing the same masked, level-wise refinement strategy to be applied. The main conceptual challenge is to adapt the construction to group-valued variables while consistently enforcing the associated gauge constraints; we leave this development to future work.

ACKNOWLEDGMENTS

This work was supported by the German Ministry for Education and Research (BMBF) under the Grant No. BIFOLD25B, and the European Union's HORIZON MSCA Doctoral Networks program project AQTIVATE (Program No. 101072344). We thank Stefan Kühn for helpful discussions and valuable comments on the manuscript. This work was also supported by funds from the Ministry of Science, Research, and Culture of the State of Brandenburg within the Centre for Quantum Technologies and Applications (CQTA).

DATA AVAILABILITY

The code used to generate the results presented in this article, together with the pretrained models and evaluation data, is available at [74].

APPENDIX A: HMC SIMULATION: CRITICAL SLOWING DOWN AND SCALING

1. Simulation details

For $L \in \{8, 16, 32, 64, 128, 256\}$ we run HMC chains with molecular-dynamics trajectory length $\tau_{\mathrm{MD}} = 1$. For $L \leq 64$ we use $n_{\mathrm{leap}} = 10$ leapfrog steps ($\epsilon = 0.1$); at the largest volumes the integrator is retuned to maintain a $\gtrsim$80% acceptance, using $n_{\mathrm{leap}} = 14$ ($\epsilon \simeq 0.071$) at $L = 128$ and $n_{\mathrm{leap}} = 20$ ($\epsilon = 0.05$) at $L = 256$, consistent with the expected $\epsilon \propto V^{-1/4}$ scaling. All simulations are performed at $\kappa_c = 0.2705$, $\lambda = 0.022$ (acceptance rates in Table III).

TABLE III. HMC acceptance rate and wall-clock time for the reference chains at $\kappa_c = 0.2705$, $\lambda = 0.022$, $\tau_{\mathrm{MD}} = 1$ (n_{leap} retuned per volume for $L \geq 128$; see text).

L	Chain length	Acceptance	Wall-clock time
8	2×10^6	98%	0.5 h
16	2×10^6	96%	0.5 h
32	2×10^6	91%	0.7 h
64	2×10^6	85%	1.5 h
128	2×10^6	85%	5.5 h
256	3×10^6	82%	55.8 h

The integrated autocorrelation time $\tau_{\mathrm{int}}(|m|)$, shown in Table IV, is estimated from each chain using the Wolff method [8,70], the Sokal automatic windowing procedure.

2. Power-law fit

Table IV collects the measured $\tau_{\mathrm{int}}(|m|)$ values. The dynamical critical exponent z_{dyn} is extracted by fitting the power-law ansatz $\tau_{\mathrm{int}} = AL^{z_{\mathrm{dyn}}}$ in the log-log space using a maximum likelihood fit (with the Gaussian noise model) to six measured points at $L \in \{8, 16, 32, 64, 128, 256\}$ as shown in Fig. 8,

$$z_{\mathrm{dyn}} = 1.81 \pm 0.10, \qquad A = 0.14, \tag{A1}$$

where the uncertainty is a bootstrap standard deviation (50,000 resamples). This is consistent with the theoretically expected value $z_{\mathrm{dyn}} \approx 2$ for HMC applied to scalar field theories [69], and with other measurements [7,62]. The $L = 256$ measurement ($\tau_{\mathrm{int}} = 8,410 \pm 1,991$) has a relative uncertainty of 24%, reflecting the ESS $N_{\mathrm{eff}} = 3 \times 10^6/(2 \times 8,410) \approx 178$.

TABLE IV. HMC integrated autocorrelation time $\tau_{\mathrm{int}}(|m|)$ at $\kappa_c = 0.2705$, $\lambda = 0.022$, measured by the Wolff method [8].

L	Chain length	$\tau_{\mathrm{int}}(\lvert m\rvert)$	Note
8	2×10^6	22.9 ± 0.3	
16	2×10^6	64.6 ± 1.3	
32	2×10^6	228 ± 9	$\tau_{\mathrm{int}} \propto L^{1.81 \pm 0.14}$
64	2×10^6	1370 ± 120	
128	2×10^6	4416 ± 928	
256	3×10^6	$8,410 \pm 1991$	$N_{\mathrm{eff}} \approx 178$

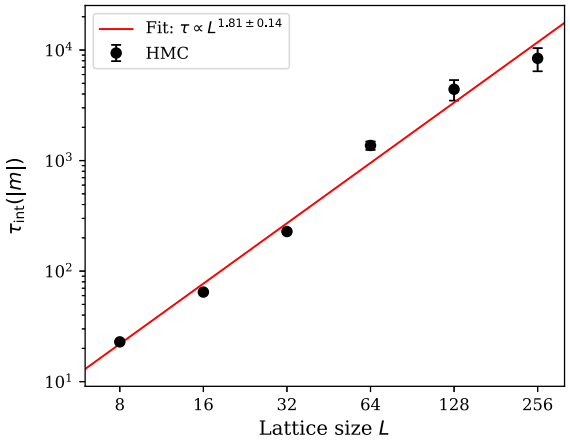


FIG. 8. HMC integrated autocorrelation time $\tau_{\rm int}(|m|)$ vs L on a log-log scale at $\kappa_c = 0.2705$. Filled circles: measured values (Table IV). Red line: power-law fit $\tau_{\rm int} = 0.37L^{1.81}$ ($z_{\rm dyn} = 1.81 \pm 0.14$), fitted to six points.

APPENDIX B: MODEL ARCHITECTURE DETAILS AND HYPERPARAMETERS

1. Our multilevel method

Overall structure. Our method is a hierarchical coarse-to-fine sampler. It starts from a coarse 8×8 lattice and applies $K = \log_2(L/8)$ successive upsampling steps, each doubling the linear lattice size. The full model therefore consists of a standalone coarse-level CNF followed by K upsampling modules, where each upsampling module is composed of two conditional GMM and two masked convolutional CNFs.

Coarse-level CNF. The coarse 8×8 field is generated by a plain Dense CNF, whose architecture follows [30] with different hyperparameters. It uses an FFT-based vector field, integrated with the RK4 solver for $n_{\rm steps}^{\rm coarse} = 40$ function evaluations (when considered as baseline, in the Dense CNF we have used 50 steps). The vector field is conditioned on a Fourier basis with $n_{\rm sin} = 15$ sine harmonics and $n_{\rm time} = 8$ time-kernel features; bond (nearest-neighbor) interactions are encoded with $n_{\rm freq}^{\rm bond} = 18$ and $n_{\rm time}^{\rm bond} = 18$. Although this coarse CNF acts on only 64 sites, it is expressive enough to capture the long-wavelength modes, the dominant source of critical slowing down as the lattice at this scale is small enough for a single dense flow to learn global correlations directly.

Upsampling step. When refining an $L_\ell \times L_\ell$ lattice to $2L_\ell \times 2L_\ell$, the fine lattice is partitioned into even-even coarse sites (which carry the already-sampled coarse field), odd-odd body-center sites, and edge sites $(2i+1, 2j)$ and $(2i, 2j+1)$. The even-even sites remain fixed throughout the upsampling step, while the odd-odd and edge sites are generated sequentially (see Fig. 1):

(1) *Body-center generation.* A conditional GMM produces a local proposal for each body-centre site, conditioned on its nearest coarse neighbors. A masked convolutional CNF then refines the full fine lattice, with its vector field masked to zero on the even-even and edge sites so that only body-center values are modified by the ODE flow.

(2) *Edge generation.* The same procedure is applied to the edge sites, conditioned on the now-complete coarse and body-center variables. Again, the CNF vector field is masked to zero everywhere except on edge sites.

Because the CNF velocity field vanishes on the previously sampled coordinates, those values are preserved exactly under the ODE integration, and the restriction $\phi^{(\ell+1)} \to \phi^{(\ell)}$ obtained by reading off the even-even positions recovers the coarse sample without error. This exact restriction is what enables the MLMC estimator in Sec. IV H.

Role of the GMM prior. At each upsampling level, the GMM provides a conditional prior for the newly introduced variables. It is fully factorized across the new sites, i.e. the sites $\phi^{(\ell)}$ are *conditionally independent* given the already-sampled coarse field $\phi^{(\le \ell-1)}$. Each site distribution is a K-component mixture of Gaussians whose parameters $(\pi_{k,x}^{(\ell)}, \mu_{k,x}^{(\ell)}, \sigma_{k,x}^{2(\ell)})$ are output by a small convolutional network that sees only a local patch of the coarse field around x. This design has two benefits. First, it concentrates the bulk of the short-distance variability into a cheap, tractable, closed-form density that can be sampled and evaluated in parallel across all sites. Second, it supplies the subsequent CNF with a sample already close to the correct local distribution, so the CNF only needs to learn the *residual* intersite correlations.

Gradient estimation for the conditional GMM prior. At each refinement level the newly introduced sites z are drawn from a conditional Gaussian mixture prior

$$q_\theta(z|c) = \sum_{k=1}^{K} \pi_k(c;\theta)\mathcal{N}(z;\mu_k(c;\theta),\sigma_k^2(c;\theta)), \qquad \text{(B1)}$$

whose weights π_k, means μ_k, and standard deviations σ_k are produced by a shared network conditioned on the coarse field c.

Sampling proceeds in two steps. A component index is first drawn from the categorical weights, $k \sim \mathrm{Cat}(\pi_1, \ldots, \pi_K)$, via a hard inverse-CDF draw. Conditional on k, the value is generated by reparameterization,

$$z = \mu_k(c;\theta) + \sigma_k(c;\theta)\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \mathbf{I}), \qquad \text{(B2)}$$

so that pathwise gradients reach μ_k and σ_k.

The model is trained by minimizing the reverse Kullback-Leibler objective $\mathcal{L}(\theta) = \mathbb{E}_{\phi\sim q_\theta}[\ell_\theta(\phi)]$ with $\ell_\theta(\phi) = \log q_\theta(\phi) + S(\phi)$ (up to an additive constant). Because the sampling distribution depends on θ through

both the continuous parameters and the discrete component index, its gradient contains two contributions,

$$\nabla_\theta \mathcal{L} = \underbrace{\mathbb{E}[\nabla_\theta \ell_\theta(\phi)]}_{\text{pathwise}} + \underbrace{\mathbb{E}[(\overline{\ell}_\theta(\phi) - b) \sum_{\text{levels}} \nabla_\theta \log \pi_k(c;\theta)]}_{\text{score function (weights)}}, \tag{B3}$$

where $\overline{\ell}_\theta = \mathrm{sg}[\log q_\theta(\phi) + S(\phi)]$ is the per-sample objective used as a detached reward, $\mathrm{sg}[\cdot]$ denotes the stop gradient, and $b = \langle \overline{\ell}_\theta \rangle$ is a batch-mean baseline. We use unit weight on the score term, so that Eq. (B3) is precisely the gradient $\nabla_\theta \mathcal{L}$ of the reverse-KL objective; the baseline b is a zero-mean control variate ($\mathbb{E}[b \nabla_\theta \log \pi_k] = 0$) that reduces its variance without biasing the estimator.

To limit the variance introduced by the score-function term, we apply it only to the mixture-weight readout head: the logits are computed from a detached copy of the shared trunk features, $\eta = \mathrm{head}(\mathrm{sg}[h])$ with $\pi = \mathrm{softmax}(\eta)$, so the REINFORCE gradient updates the weight head without perturbing the shared representation used by the means, standard deviations, and the downstream flow.

Why small convolutional CNFs still scale. The upsampling CNFs use relatively shallow convolutional vector fields with small kernels (Table V) and only $n_{\text{steps}} = 20$ ODE steps. Individually, such networks have limited receptive field and cannot directly represent long-range correlations on the fine lattice. Scalability is recovered by the hierarchical decomposition where long-wavelength structure is already encoded in the coarse field by the coarse-level CNF, and each subsequent upsampling level only needs to introduce short-distance fluctuations on top of it.

GMM and CNF hyperparameters. The GMM network at each upsampling level uses two convolutional layers, 32 hidden channels, and kernel size 5×5, with $H = 6$ mixture components at intermediate levels and $H = 12$ at the finest level. The body-center and edge CNFs use convolutional vector fields with periodic boundary conditions, the same Fourier-basis conditioning as the coarse CNF, and RK4 integration with $n_{\text{steps}} = 20$. Kernel sizes follow the schedule in Table V.

2. SR-NF

SR-NF follows the coarse-to-fine upsampling architecture of [66], with the HMC-based coarsest sampler of the original proposal replaced by the same Dense CNF used in our method (RK4, 40 steps). At each upsampling step, the coarse field is naively block-replicated on a 2×2 grid, Gaussian block noise is injected, and a full-lattice (unmasked) CNF refines all $L_{\ell+1}^2$ sites jointly.

The hyperparameter choices for SR-NF are dictated by a sequence of coupled constraints. First, with the coarse lattice set at $L_0 = 8$ (to match our method), we observed that SR-NF training became unstable: the ESS initially rose, then collapsed as the loss diverged irreversibly (see Appendix F for details). Training stability was recovered only by pushing the coarse lattice down to $L_0 = 2$, which replaces each large-stride upsampling step with a deeper chain of smaller 2×2 refinements [$K = \log_2(L/2)$ levels, compared to $\log_2(L/8)$ for our method]. Second, this deeper hierarchy increases peak training memory substantially; to fit on an A100 80 GB GPU at $L = 128, 256$ we had to reduce the convolutional kernel size (see Table V). Because the receptive field compounds across the extra levels, the *effective* receptive field on the finest lattice remains comparable to that of our method despite the smaller per-level kernels. Finally, the number of ODE steps is $n_{\text{steps}} = 70$ at all sizes except $L = 256$, where $n_{\text{steps}} = 60$ was needed for memory constraint.

3. Dense CNF

Dense CNF [30] is a single-scale baseline: one CNF acts directly on the full $L \times L$ lattice, starting from an isotropic Gaussian prior. The vector field is a *dense* single-layer FFT-based parametrization in which each spatial mode interacts with all others, with $\mathcal{O}(L^4)$ parameters. It is integrated with RK4 for $n_{\text{steps}} = 50$ evaluations. A richer Fourier basis is used: $n_{\text{sin}} = 49$, $n_{\text{time}} = 11$, $n_{\text{freq}}^{\text{bond}} = 20$, $n_{\text{time}}^{\text{bond}} = 20$. The $\mathcal{O}(L^4)$ memory cost of the exact log determinant makes training infeasible beyond $L = 64$. Dense CNF was run at

TABLE V. Per-size hyperparameters at $\kappa_c = 0.2705$. Our method uses $L_0 = 8$; SR-NF uses $L_0 = 2$. K denotes the number of upsampling levels. n_{steps} is the fine-level ODE budget per upsampling CNF; the coarse Dense CNF always uses 40 steps.

	Our method				SR-NF			
L	K	Kernel sizes	n_{steps}	Batch	K	Kernel sizes	n_{steps}	Batch
16	1	[7]	20	256	3	[9, 9, 7]	70	200
32	2	[9, 7]	20	128	4	[9, 9, 9, 7]	70	128
64	3	[9, 9, 7]	20	100	5	[9, 9, 9, 9, 7]	70	100
128	4	[9, 9, 7, 7]	20	80	6	[7, 7, 7, 7, 7, 7]	70	32
256	5	[9, 9, 7, 7, 7]	20	22	7	[5, 5, 5, 5, 5, 5, 5]	60	12

$L \in \{16, 32, 64\}$ with batch size 256,128,64 and 3,000,5,000,10,000 epochs.

4. Hutch CNF

Hutch CNF is also a single-scale baseline acting directly on the full lattice. Its vector field is a three-layer convolutional network with 32 hidden channels, 7×7 kernels, and GELU activations. The divergence is estimated with the Hutchinson stochastic trace. The ODE is solved with a fixed-step RK4 integrator at $n_{\text{steps}} = 40$, and gradient checkpointing is used to reduce memory. The same Fourier basis as our method is used ($n_{\text{sin}} = 15$, $n_{\text{time}} = 8$). Hutch CNF was run at $L \in \{16, 32\}$ with batch size 256,128 and 3,000,5,000 epochs.

5. Training

All methods are trained end-to-end with the reverse-KL objective using the Adam optimizer [75] ($\beta_1 = 0.8$, $\beta_2 = 0.9$). Our method and SR-NF use initial learning rate 10^{-2}; Dense CNF uses 5×10^{-3} as in Ref. [30]. All methods apply an exponential learning-rate decay with $\gamma = 0.9994$ and minimum 10^{-4}. Gradient norms are clipped at 0.5 for our method, SR-NF, and Hutch CNF, and at 1.0 for Dense CNF. All runs use a single NVIDIA A100 GPU.

6. Hyperparameters

Table V lists hyperparameters that vary with target lattice size L for the two hierarchical methods.

7. Training and sampling time

Table VI reports the total GPU time consumed during training for each method and lattice size, all on a single NVIDIA A100 80 GB GPU. Our method and SR-NF scale moderately with L, reaching 11.5 and 23.3 GPU hours at $L = 256$, respectively. Hutch CNF is disproportionately expensive at $L = 32$ (4.1 GPU hours) relative to its model capacity, reflecting the high memory and compute overhead of the graph-retention divergence estimator discussed in Appendix F. Dense CNF is cheap at small L but cannot be trained beyond $L = 64$.

Table VII reports the raw sampling throughput (generated samples per second) for each method at $\kappa_c = 0.2705$, measured during the importance-sampling evaluation on an NVIDIA A100 80 GB GPU.

TABLE VI. Training cost in GPU-hours (NVIDIA A100 80 GB) at $\kappa_c = 0.2705$, $\lambda = 0.022$. Dashes indicate the lattice sizes for which training is infeasible.

L	Our method	SR-NF	Dense CNF	Hutch CNF
16	0.3	0.8	0.3	0.6
32	0.8	2.2	0.5	4.1
64	2.3	5.3	1.5	···
128	7.8	18.3	···	···
256	11.5	23.3	···	···

TABLE VII. Raw sampling throughput [samples/s] at $\kappa_c = 0.2705$, $\lambda = 0.022$. Measured during IS evaluation with $N = 100000$ samples (batch size 256 for $L \leq 64$, 128 for $L = 128$, 64 for $L = 256$). Dashes indicate the lattice sizes for which training is infeasible.

L	Our method	SR-NF	Dense CNF	Hutch CNF
16	995	310	1750	516
32	677	277	877	336
64	337	160	321	···
128	103	21	···	···
256	30	9	···	···

APPENDIX C: ABLATION STUDY

We evaluate the main architectural and numerical components of the sampler at the critical point $\kappa_c = 0.2705$ on an $L = 32$ lattice. Unless stated otherwise, all hyperparameters are kept fixed. The resulting importance-sampling efficiencies, ESS/N, are reported in Fig. 9 and Table VIII. We examine five components.

Full model. We evaluate the ESS of the full model using different numbers of ODE integration steps in the CNF. The ESS remains sufficiently higher 56% even with as few as ten ODE steps, far above the value obtained without the conditional prior 0.09 indicating that the combined model is numerically stable and robust to a relatively coarse ODE discretization.

Flow refinement: GMM only. Removing the masked flow and sampling the newly introduced sites solely from the conditional Gaussian mixture reduces the ESS from 0.83 to 0.09. This demonstrates that the flow refinement is essential for capturing the residual correlations among the newly generated sites that are not described by the Gaussian-mixture prior.

CNF only, without the GMM prior. With a sufficiently large number of ODE steps, the CNF can learn most of the required transformation and provides the dominant contribution to the model accuracy. However, the ESS decreases sharply as the number of ODE steps is reduced. In contrast, the full model maintains an ESS approximately 86% down to 15 steps and 56% at ten steps of ode. This shows that the conditional GMM provides an effective prior for the CNF and stabilizes the model when a coarse ODE discretization is used.

The benefit of GMM becomes considerably larger as the hierarchy deepens, at $L = 64$, the efficiency decreases from 0.71 with the conditional GMM to 0.50 without it as shown in Table VIII. The difference also grows as the ODE integration budget is reduced. At $L = 32$, for example, decreasing the number of steps from 20 to 10 retains an efficiency of 0.56 with the GMM, whereas the efficiency

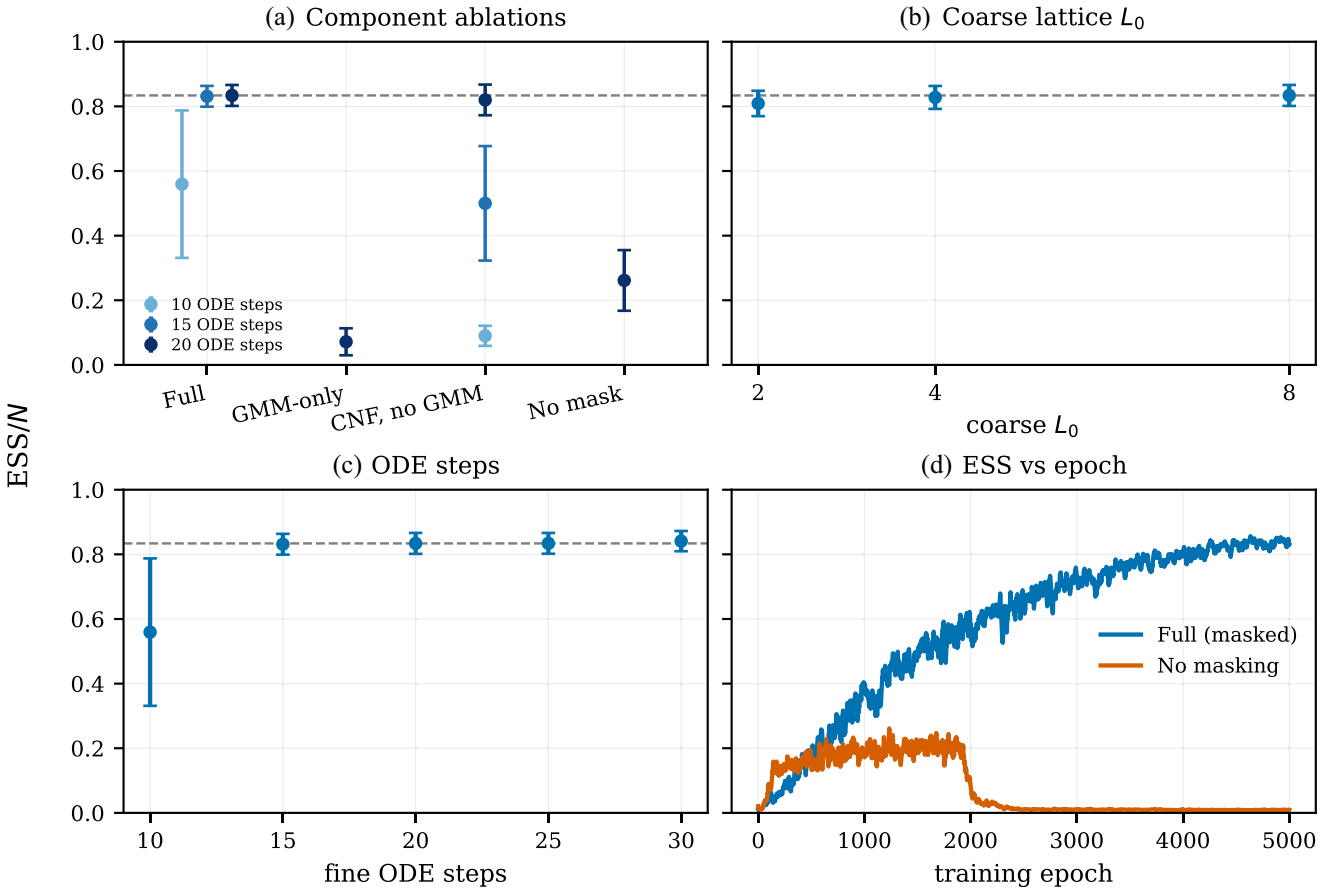


FIG. 9. Ablation studies at $\kappa_c = 0.2705$ on an $L = 32$ lattice. The reported efficiency is ESS/N, and the error bars show the per-epoch variation of the ESS over the converged training window. (a) Removing individual components shows that both the masked CNF refinement and the masking structure are essential and GMM plays an important role in cost reduction by achieving high ESS at lowered numbers of ode steps. (b) The dependence on the initial coarse lattice L_0 is weak. (c) The efficiency saturates at approximately 15 fine ODE steps per flow. (d) The unmasked full-lattice model is unstable, its ESS initially rises and then collapses, whereas the masked model converges stably.

without it falls to 0.09. The conditional prior therefore enables efficient training at a substantially lower integration cost.

Conceptually, the unconditional Gaussian is farther from the target distribution of the newly introduced sites conditioned on the coarse field. Each flow must consequently perform a larger and more difficult transport, increasing the possibility of deviating from the desired conditional structure and destabilizing training. By initializing the new sites closer to their target conditional distribution, the conditional GMM reduces the transport required at each refinement level. Its advantage is therefore most pronounced for larger lattices and smaller ODE budgets, where it improves both sampling efficiency and training stability.

TABLE VIII. Importance-sampling efficiency ESS/N at $\kappa_c = 0.2705$ with and without the conditional GMM prior. The lattice-size comparison is performed using 20 ODE steps per flow, while the ODE-step comparison is performed at $L = 32$.

Configuration	With GMM	Without GMM
(a) Lattice size at 20 ODE steps per flow		
$L = 32$	0.86	0.83
$L = 64$	0.71	0.50
(b) ODE steps per flow at $L = 32$		
5	0.02	0.02
10	0.56	0.09
15	0.86	0.50
20	0.86	0.82

Masking. Replacing the two parity-restricted flows by a single flow acting on the entire fine lattice destabilizes training. The efficiency initially increases but subsequently collapses irreversibly [Fig. 9(d)], with a peak below ∼0.2. Preserving the inherited coarse sites exactly is therefore crucial for maintaining long-range correlations in the hierarchy and ensuring stable training.

Coarse lattice L_0. Varying the initial coarse resolution over $L_0 \in 2, 4, 8$ changes ESS/N only from approximately 0.81 to 0.83 [Fig. 9(b)]. Thus, within this range,

the long-wavelength modes are captured comparably well and the final efficiency depends only weakly on the choice of L_0.

Fine ODE steps. The efficiency saturates at approximately 15 integration steps per flow [Fig. 9(c)]. Increasing the budget beyond this value produces little additional improvement, whereas reducing it to 10 steps lowers the efficiency and can destabilize training, as indicated by the superscript *.

Overall, the dominant performance gain comes from combining the coarse-to-fine hierarchy with masked flow-based refinement. The precise coarse resolution is comparatively unimportant, while the conditional prior plays a more subtle but essential role in reducing the integration cost and stabilizing training.

APPENDIX D: RELATION TO RENORMALIZATION-GROUP BLOCKING

Our construction follows a Kadanoff blocking coarse-to-fine hierarchy, but the continuous normalizing flows used at each level are not themselves RG transformations. A Wilsonian RG transformation integrates out short-distance degrees of freedom and produces an effective action with scale-dependent couplings. In contrast, our CNFs transport samples while the original fine-lattice action remains fixed. Rather than constructing the generally intractable conditional densities $p(\phi^{(\ell)}|\phi^{(\leq \ell-1)})$, we approximate them with learned conditional models q_ℓ, consisting of a conditional GMM prior followed by a masked CNF.

The masking structure nevertheless provides a natural RG-inspired diagnostic. Because the inherited coarse sites are preserved exactly during refinement, the restriction

$$R(\phi^{(\leq \ell)}) = \phi^{(\leq \ell-1)} \tag{D1}$$

is deterministic and exact. The marginal distribution produced by the model at each coarse level can therefore be compared directly with an HMC ensemble obtained by applying the same restriction to fine-lattice configurations. This comparison requires neither an explicit coarse action nor a parametrization of effective couplings.

We perform this test at κ_c using the model with $L_0 = 2$, whose hierarchy contains $L \in \{2, 4, 8, 16, 32\}$. Starting from an HMC ensemble at $L = 32$, we repeatedly retain the even–even sublattice, $32 \to 16 \to 8 \to 4 \to 2$, matching exactly the decimation used by the model. At every level, we compare the raw, unreweighted model samples with the corresponding decimated HMC ensemble using the absolute magnetization $\langle|m|\rangle$ and the connected susceptibility

$$\chi_{\text{conn}} = L^2(\langle m^2\rangle - \langle|m|\rangle^2). \tag{D2}$$

Uncertainties are estimated by bootstrap resampling for our method and by blocked jackknife for HMC.

As shown in Fig. 10, the model reproduces both observables across all five resolutions. The values of $\langle|m|\rangle$ agree within their combined uncertainties, while χ_{conn} agrees to within a few percent. At $L = 32$, for example, we obtain $\langle|m|\rangle = 0.7112 \pm 0.0008$ compared

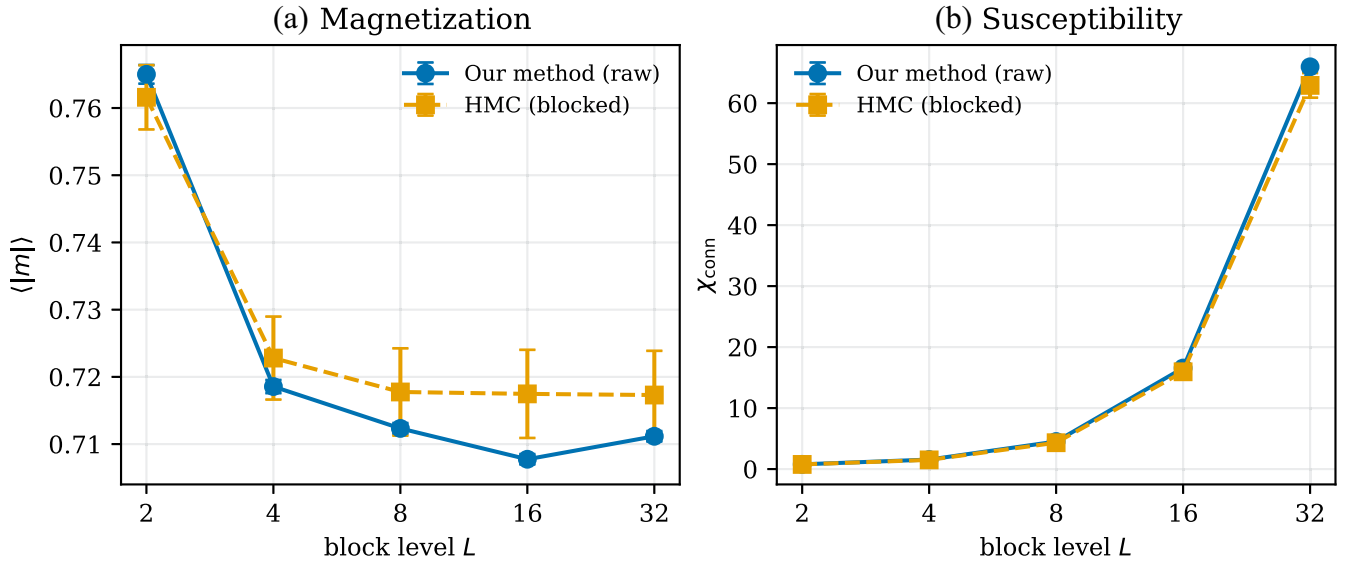


FIG. 10. Comparison of the learned coarse marginals with even–even decimated HMC ensembles at κ_c for $L_0 = 2$. Raw model samples and blocked HMC results are shown at $L \in 2, 4, 8, 16, 32$ for (a) $\langle|m|\rangle$ and (b) χ_{conn}. Agreement is observed at all levels without importance reweighting.

with 0.7173 ± 0.0066 from HMC, and $\chi_{\mathrm{conn}} = 66.0 \pm 0.3$ compared with 62.9 ± 2.0. Similar agreement is observed at every coarser level.

The small systematic offset in the raw model results—slightly lower $\langle|m|\rangle$ and slightly higher χ_{conn}—reflects the remaining mismatch between the learned distribution and the target, consistent with $\mathrm{ESS}/N \simeq 0.81$. We deliberately use unreweighted samples here so that the agreement is not enforced by importance reweighting. The comparison therefore shows directly that the learned hierarchy produces coarse marginals consistent with genuine decimations of the fine-lattice target ensemble.

Determining the effective action associated with these coarse marginals would require a separate inverse-RG analysis. The exact blocked action is generally nonlocal and generates an increasingly large operator basis, as indicated in Eq. (7). Reconstructing its couplings and following their evolution across scales is beyond the scope of this work, whose objective is an efficient and asymptotically unbiased sampler.

APPENDIX E: TRAINING EFFICIENCY: ESS VS GPU TIME

Figure 11 shows the evolution of the training ESS/N (20-epoch moving average) as a function of wall-clock GPU time, clipped to a 4-hour budget, for our method and SR-NF at $L = 64$ and $L = 128$.

For $L = 64$, our method completes its full training schedule (10,000 epochs) in ≈2.3 GPU-hours, reaching $\mathrm{ESS}/N \approx 0.67$. In contrast, SR-NF requires ≈5.3 GPU-hours for the same number of epochs and achieves only $\mathrm{ESS}/N \approx 0.29$ at the 4-h mark.

For $L = 128$, the training time gap widens: our method processes each epoch roughly 2.3× faster than SR-NF. Within 4 GPU-hours, our method achieves $\mathrm{ESS}/N \approx 0.30$, exceeding SR-NF's value of ≈0.13, with a steeper improvement trajectory. The full run of our method (20,000 epochs, 7.8 GPU-hours) reaches $\mathrm{ESS}/N \approx 0.34$, whereas SR-NF at its full run (20,000 epochs, 18.3 GPU-hours) plateaus at ≈0.15.

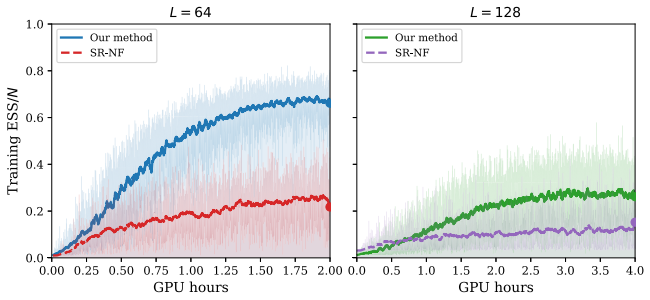


FIG. 11. Training ESS/N (Moving Average-20) vs GPU hours, clipped to 4 h, for our method (solid blue) and SR-NF (dashed orange) at $L = 64$ (left) and $L = 128$ (right). Filled circles mark the ESS at the end of the clipped window.

APPENDIX F: COMPUTATIONAL SCALING ANALYSIS

Let $N = L^2$ denote the number of lattice sites and B the batch size. We analyze the peak training memory and explain why the baseline methods fail to scale to large lattices.

In common implementations, all models are trained by backpropagating through the full ODE integration trajectory, rather than using the reverse integration method. As a result, the complete computation graph over all evaluations of the ODE functions must be retained in memory during training. For an RK4 integrator with n_{steps} steps, this corresponds to storing activations from evaluations of functions $4n_{\mathrm{steps}}$.

1. Dense CNF

The Dense CNF vector field is a dense operator that is translationally and D_4-equivariant acting on the entire lattice. With the FFT implementation [30], the number of independent parameters scales as $\mathcal{O}(N/8)$ due to orbit reduction, while the forward cost per ODE evaluation scales as

$$\mathcal{C}^{\mathrm{fwd}}_{\mathrm{Dense}} \sim \mathcal{O}(F_{\mathrm{basis}} N \log N), \tag{F1}$$

where

$$F_{\mathrm{basis}} = n_{\mathrm{sin}} \times n_{\mathrm{time}} = 49 \times 11 = 539. \tag{F2}$$

The divergence is available in closed form and is therefore relatively cheap; the dominant bottleneck is memory. At each ODE evaluation the model materializes a basis-expansion tensor of shape

$$[B, N, F_{\mathrm{basis}}], \tag{F3}$$

so the peak training memory scales as

$$M_{\mathrm{Dense}} \sim 4n_{\mathrm{steps}} \times B \times N \times F_{\mathrm{basis}} \times \mathrm{size\,of\,(float32)}, \tag{F4}$$

where the leading factor of four accounts for the four stages of the RK4 integrator, each of which must retain its activations for backpropagation. For $L = 64$ ($N = 4096$, $B = 64$, $n_{\mathrm{steps}} = 50$), this gives

$$4 \times 50 \times 64 \times 4096 \times 539 \times 4\mathrm{B} \approx 113\ \mathrm{GB}, \tag{F5}$$

which already exceeds the 80 GB hardware limit and explains why, at $L = 64$, Dense CNF had to be trained with batch size $B = 64$, whereas the hierarchical methods use $B = 100$ at the same lattice size Table V. The memory cost of exact divergence evaluation prevents Dense CNF from reaching the same batch size. At $L = 128$ ($N = 16384$), the lattice volume quadruples, driving the memory requirement

to approximately 450 GB, far beyond available hardware. Thus the effective scaling limitation of Dense CNF is

$$M_{\text{Dense}} \sim \mathcal{O}(F_{\text{basis}} n_{\text{steps}} BN), \tag{F6}$$

with a very large prefactor that makes the method memory infeasible beyond $L = 64$ even at reduced batch sizes.

2. Hutch CNF

Hutch CNF replaces the exact divergence by a stochastic estimator. The vector field is a three-layer convolutional network with parameter count independent of L, and the forward cost per ODE evaluation scales as

$$C_{\text{Hutch}}^{\text{fwd}} \sim \mathcal{O}(DHK^2N), \tag{F7}$$

which is formally well behaved in lattice volume.

However, the stochastic divergence estimator requires differentiating through an additional Jacobian-based quantity at every ODE evaluation. This introduces a substantial memory overhead beyond the forward pass. The peak training memory therefore scales approximately as

$$M_{\text{Hutch}} \sim 4 n_{\text{steps}} \times B \times N \times (F_{\text{basis}} + D \times H) \times c_{\text{div}}, \tag{F8}$$

where $c_{\text{div}} > 1$ represents the extra graph-storage cost of the stochastic divergence evaluation. In practice, despite the smaller parameter count, this makes the peak training memory comparable to Dense CNF: $L = 16$ remains feasible on a 40 GB GPU, but already $L = 32$ requires 80 GB-class hardware.

The second limitation is statistical rather than memory related. The stochastic divergence estimate has variance that grows with system size,

$$\text{Var}[\widehat{\text{div}}(v)] \propto N^2 \propto L^4, \tag{F9}$$

so the gradient signal-to-noise ratio deteriorates rapidly with L. The combined effect of increased memory use and increasingly noisy training limits Hutch CNF to small lattices. Both SR-NF and our method are hierarchical models based on convolutional CNF components and have the same asymptotic O(L 2) memory scaling

3. Empirical observation and discussion on SR-NF vs our method

Both SR-NF and our method are hierarchical models based on convolutional CNF components and the same asymptotic memory scaling $\mathcal{O}(L^2)$. The difference is therefore not the formal scaling law but the amount of work performed per level and the resulting statistical efficiency.

At level ℓ, our method transports only the newly introduced sites, i.e. $\frac{3}{4}N_\ell$, whereas SR-NF transports all N_ℓ sites. In addition, SR-NF need 70–100 ODE steps per level for stable training, while for our method 20 ODE steps are sufficient. As a result, SR-NF is several times more expensive per refinement level, although both methods have the same asymptotic $\mathcal{O}(L^2)$ scaling.

The more important difference is statistical. In our method, the masked CNF leaves the coarse sites unchanged, so the restriction $R(\phi^{(\leq \ell)}) = \phi^{(\leq \ell-1)}$ is exact. Each refinement level therefore only has to learn the conditional distribution of the newly introduced sites given the coarse field. The associated per-level log-weight variance σ_ℓ^2 remains controlled as L grows.

In SR-NF, by contrast, the full-lattice CNF updates all sites at every level, including the coarse ones. Each level must therefore both correct the newly introduced variables and re-correct the coarse field. This makes the task progressively harder as the lattice becomes larger, so that σ_ℓ^2 increases with L_ℓ. Since the overall IS efficiency behaves as

$$\frac{\text{ESS}}{N} \propto \exp\left(-\frac{1}{2}\sum_\ell \sigma_\ell^2\right), \tag{F10}$$

the accumulated variance across levels leads to an exponential deterioration of ESS with depth. Furthermore, unlike our method with the GMM prior, local correlation must be learned independently for each site, which can be achieved by CNF but with less statistical efficiency. This factor leads to the observed growth of the induced autocorrelation time from $\tau_{\text{int}} = 186 (L = 128)$ to $\tau_{\text{int}} = 3615 (L = 256)$.

Also as in SR-NF single-step upsampler is used, we find that it is difficult to stabilize the training with a small number of ODE steps. The ablation studies in Appendix C show that the conditional GMM prior plays an important role in stabilizing this training. By placing the newly introduced sites close to their conditional target distributions before the flow acts, the GMM prior enables stable and efficient sampling with a relatively small ODE integration budget. In summary, Dense CNF fails because of the large memory footprint of its dense exact-divergence architecture, Hutch CNF fails because of both memory overhead and rapidly growing gradient noise, and SR-NF fails because its unmasked hierarchical transport and lacks of systemical structure capturing local conditions across lattice sites causes strong deterioration in ESS. Our method is the only one that avoids all three failure modes simultaneously in the tested regime.

APPENDIX G: MLMC ESTIMATOR

1. Per-level budget allocation

Table IX gives the per-level costs and optimal sample counts for MLMC applied to our method at $L = 32$ and $L = 64$, derived from a probe run of 10^4 samples per level.

TABLE IX. Per sample cost c_ℓ (μs/sample at level l) and optimal sample count N_ℓ [Eq. (29)] for MLMC at equal wall-clock budget $T = N_{\mathrm{IS}} \times c_L$ with $N_{\mathrm{IS}} = 10^5$.

L_ℓ	$L = 32$ run		$L = 64$ run	
	c_ℓ (μs)	N_ℓ	c_ℓ (μs)	N_ℓ
2	13.9	4.76×10^6	6.5	6.85×10^6
4	34.4	2.46×10^6	17.8	3.51×10^6
8	135.3	6.32×10^5	73.2	9.08×10^5
16	406.6	1.53×10^5	238.2	2.23×10^5
32	6286	5.24×10^4	1003.6	5.13×10^4
64	…	…	12812	7.83×10^4

The coarser levels are sampled with orders-of-magnitude more configurations than the finest level, concentrating the fine-level budget on the coarser levels which have high variance. Figure 12 compares the variances of the MLMC and plain IS estimators at equal computational budget for our method at $L = 32$ and $L = 64$.

2. Possibility to apply MLMC to SR-NF

MLMC requires an *exact restriction map*, i.e., given a fine sample $\phi^{(\leq \ell)}$, the restriction $R(\phi^{(\leq \ell)}) = \phi^{(\leq \ell-1)}$ recovers the coarse sample actually used to generate it. For our decomposition, R is trivial, as we preserve coarse sites at the each level. For SR-NF this property does not hold and the upsampling CNF acts on the full fine lattice and displaces every site, so a naïve projection of the fine output onto the coarse sublattice yields a configuration that was never drawn from $q^{(\leq \ell-1)}$, and the level-ℓ MLMC correction

$$\Delta_n^{(\ell)} = \mathcal{O}^{(\ell)}(\phi_n^{(\leq \ell)}) - \mathcal{O}^{(\ell-1)}(\phi_n^{(\leq \ell-1)}),$$
$$\phi_n^{(\leq \ell-1)} = R(\phi_n^{(\leq \ell)}), \tag{G1}$$

is biased. The exact coarse ancestor in SR-NF can be recovered either by integrating the level-ℓ CNF backward for each fine sample or by caching $\phi_n^{(\leq \ell-1)}$ during forward generation; the former costs one extra full-lattice ODE per level and eliminates any MLMC advantage on its own, while the latter trades memory for computational cost and is the variant we adopt below.

However, unbiasedness alone is not enough. MLMC delivers a speedup only when $\mathrm{Var}[\mathcal{O}^{(\ell)} - \mathcal{O}^{(\ell-1)}] \ll \mathrm{Var}[\mathcal{O}^{(\ell)}]$, which requires strong cross-level correlation. Figure 13 compares plain IS against MLMC for SR-NF and for our method at $L = 32$, $\kappa = 0.2705$, at equal wall-clock budget T_{IS}: both estimators are asymptotically unbiased within one standard deviation, but SR-NF shows no computational benefit over plain IS, whereas our method reduces the required compute by roughly 38% for the same statistical precision. A plausible explanation is the nature of the interlevel step where in SR-NF, the transition from $\ell - 1$ to ℓ injects fresh Gaussian noise on the new sites and the observables are weakly coupled between adjacent levels. Our method avoids this by construction, as coarse sites are preserved across levels and the interlevel coupling is strong.

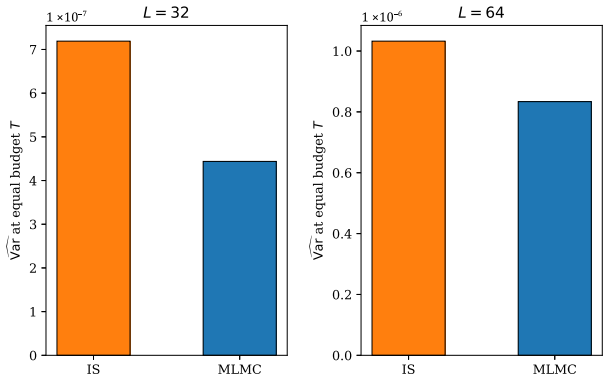


FIG. 12. MLMC vs plain IS variance at equal budget $T = 10^5 c_L$ for our method at $L = 32$ and $L = 64$. MLMC (blue) achieves lower variance than IS (red) at both sizes. (Table II).

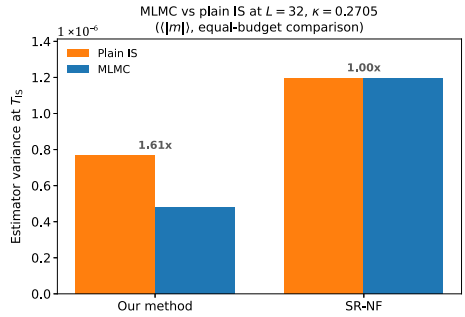


FIG. 13. MLMC versus plain IS at $L = 32$, $\kappa = 0.2705$, for $\langle|m|\rangle$, at equal wall-clock budget T_{IS}. Both estimators are asymptotically unbiased. SR-NF offers no computational benefit over plain IS, while MLMC applied to our method reduces the computational cost required for equivalent precision by roughly 38%.

[1] K. G. Wilson and J. Kogut, Phys. Rep. **12**, 75 (1974).
[2] M. Creutz, *Quarks, Gluons and Lattices* (Cambridge University Press, Cambridge, England, 1983).
[3] M. Creutz, L. Jacobs, and C. Rebbi, Phys. Rep. **95**, 201 (1983).
[4] Y. Aoki *et al.* [Flavour Lattice Averaging Group (FLAG)], Phys. Rev. D **113**, 014508 (2026).
[5] T. Aoyama *et al.*, Phys. Rep. **887**, 1 (2020).
[6] R. Aliberti *et al.*, Phys. Rep. **1143**, 1 (2025).
[7] U. Wolff, Nucl. Phys. **B17**, 93 (1990).
[8] U. Wolff, Comput. Phys. Commun. **156**, 143 (2004).
[9] S. Schaefer, R. Sommer, and F. Virotta (ALPHA Collaboration), Nucl. Phys. **B845**, 93 (2011).
[10] E. G. Tabak and E. Vanden-Eijnden, Commun. Math. Sci. **8**, 217 (2010).
[11] E. G. Tabak and C. V. Turner, Commun. Pure Appl. Math. **66**, 145 (2013).
[12] D. Rezende and S. Mohamed, in *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 (PMLR, Lille, France, 2015), pp. 1530–1538, https://proceedings.mlr.press/v37/rezende15.html.
[13] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Adv. Neural Inf. Process. Syst. **31** (2018), https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.
[14] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Phys. Rev. D **100**, 034515 (2019).
[15] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Phys. Rev. Lett. **125**, 121601 (2020).
[16] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Phys. Rev. D **104**, 114507 (2021).
[17] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel, Phys. Rev. E **101**, 023304 (2020).
[18] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Phys. Rev. Lett. **126**, 032001 (2021).
[19] A. Singha, D. Chakrabarti, and V. Arora, Phys. Rev. D **107**, 014512 (2023).
[20] A. Singha, D. Chakrabarti, and V. Arora, Phys. Rev. D **108**, 074518 (2023).
[21] R. Abbott, A. Botev, D. Boyda, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Phys. Rev. D **109**, 094514 (2024).
[22] P. Tuo, Z. Zeng, J. Chen, and B. Cheng, J. Chem. Theory Comput. **21**, 11427 (2025).
[23] R. Abbott, D. Boyda, Y. Fu, D. C. Hackett, G. Kanwar, F. Romero-López, P. E. Shanahan, and J. M. Urban, Phys. Rev. D **114**, 014513 (2026).
[24] A. Faraz, A. Singha, D. Chakrabarti, S. Nakajima, and V. Arora, Proc. Sci. LATTICE2024 (**2025**) 036 [arXiv:2308.08615].
[25] J. Finkenrath, Proc. Sci. LATTICE2023 (**2024**) 022.
[26] J. Kreit, A. Bulgarelli, L. Funcke, T. Luu, D. Schuh, S. Singh, and L. Verzichelli, Proc. Sci. LATTICE2025 (**2026**) 043 [arXiv:2601.18273].
[27] D. Schuh, L. Funcke, J. Kreit, T. Luu, and S. Singh, arXiv:2603.18205.
[28] V. Kanaujia, M. S. Scheurer, and V. Arora, SciPost Phys. **16**, 132 (2024).
[29] S. Bacchio, P. Kessel, S. Schaefer, and L. Vaitl, Phys. Rev. D **107**, L051504 (2023).
[30] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, SciPost Phys. **15**, 238 (2023).
[31] M. Caselle, E. Cellini, and A. Nada, J. High Energy Phys. 02 (2024) 048.
[32] M. Gerdes, P. de Haan, R. Bondesan, and M. C. N. Cheng, Phys. Rev. D **112**, 094516 (2025).
[33] M. Caselle, E. Cellini, A. Nada, and M. Panero, J. High Energy Phys. 07 (2022) 015.
[34] M. Caselle, E. Cellini, and A. Nada, J. High Energy Phys. 02 (2025) 090.
[35] A. Bulgarelli, E. Cellini, K. Jansen, S. Kühn, A. Nada, S. Nakajima, K. A. Nicoli, and M. Panero, Phys. Rev. Lett. **134**, 151601 (2025).
[36] A. Bulgarelli, E. Cellini, and A. Nada, Phys. Rev. D **111**, 074517 (2025).
[37] C. Bonanno, A. Bulgarelli, E. Cellini, A. Nada, D. Panfalone, D. Vadacchino, and L. Verzichelli, J. High Energy Phys. 04 (2026) 051.
[38] D. Wu, L. Wang, and P. Zhang, Phys. Rev. Lett. **122**, 080602 (2019).
[39] A. Singha, E. Cellini, K. A. Nicoli, K. Jansen, S. Kühn, and S. Nakajima, in *The Thirteenth International Conference on Learning Representations (ICLR 2025)* (2025).
[40] P. Białas, P. Korcyl, and T. Stebel, Comput. Phys. Commun. **281**, 108502 (2022).
[41] P. Białas, V. Chahar, P. Korcyl, T. Stebel, M. Winiarski, and D. Zapolski, Comput. Phys. Commun. **318**, 109892 (2026).
[42] X. Du, J. Nam, S. Liu, and R. Gómez-Bombarelli, arXiv:2603.14695.
[43] L. Wang, G. Aarts, and K. Zhou, J. High Energy Phys. 05 (2024) 060.
[44] Q. Zhu, G. Aarts, W. Wang, K. Zhou, and L. Wang, J. High Energy Phys. 03 (2026) 111.
[45] G. Aarts, D. E. Habibi, A. Ipp, D. I. Müller, T. R. Ranner, L. Wang, W. Wang, and Q. Zhu, arXiv:2601.19552.
[46] G. Kanwar and O. Vega, Proc. Sci. LATTICE2025 (**2026**) 040 [arXiv:2512.19877].
[47] O. Vega, J. Komijani, A. El-Khadra, and M. Marinkovic, J. High Energy Phys. 07 (2026) 213.
[48] V. Kanaujia and V. Arora, in *Proceedings of the 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2026), pp. 4336–4340.
[49] H. Alharazin, J. Y. Panteleeva, and B. D. Sun, Phys. Rev. D **114**, 014522 (2026).
[50] J. M. Pawlowski and J. M. Urban, Mach. Learn. Sci. Technol. **1**, 045011 (2020).
[51] J. Singh, M. Scheurer, and V. Arora, SciPost Phys. **11**, 043 (2021).
[52] A. Singha, D. Chakrabarti, and V. Arora, SciPost Phys. Core **5**, 052 (2022).
[53] L. Del Debbio, J. M. Rossney, and M. Wilson, Phys. Rev. D **104**, 094507 (2021).
[54] R. Abbott *et al.*, Eur. Phys. J. A **59**, 257 (2023).
[55] G. Catumba and A. Ramos, Eur. Phys. J. C **85**, 1037 (2025).

[56] J. Goodman and A. D. Sokal, Phys. Rev. Lett. **56**, 1015 (1986).
[57] J. Goodman and A. D. Sokal, Phys. Rev. D **40**, 2035 (1989).
[58] W. Janke and T. Sauer, Phys. Rev. E **49**, 3475 (1994).
[59] K. E. Schmidt, Phys. Rev. Lett. **51**, 2175 (1983).
[60] M. Faas and H. Hilhorst, Physica A (Amsterdam) **135**, 571 (1986).
[61] M. G. Endres, R. C. Brower, W. Detmold, K. Orginos, and A. V. Pochinsky, Phys. Rev. D **92**, 114516 (2015).
[62] K. Jansen, E. H. Müller, and R. Scheichl, Phys. Rev. D **102**, 114512 (2020).
[63] M. Grabenstein and B. Mikeska, Phys. Rev. D **47**, R3103 (1993).
[64] P. Białas, P. Czarnota, P. Korcyl, and T. Stebel, Phys. Rev. E **107**, 054127 (2023).
[65] R. Abbott, M. S. Albergo, D. Boyda, D. C. Hackett *et al.*, Proc. Sci. LATTICE2023 (**2024**) 035 [arXiv:2404.10819].
[66] M. Bauer, R. Kapust, J. M. Pawlowski, and F. L. Temmen, SciPost Phys. **19**, 077 (2025).
[67] F. Ihssen, R. Kapust, and J. M. Pawlowski, Proc. Sci. LATTICE2025 (**2026**) 055 [arXiv:2510.26678].
[68] M. B. Giles, Acta Numer. **24**, 259 (2015).
[69] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[70] N. Madras and A. D. Sokal, J. Stat. Phys. **50**, 109 (1988).
[71] L. P. Kadanoff, Phys. Phys. Fiz. **2**, 263 (1966).
[72] K. Binder, Z. Phys. B **43**, 119 (1981).
[73] M. F. Hutchinson, Commun. Stat. Simul. **19**, 433 (1990).
[74] https://git.tu-berlin.de/rigcs/phi4/multilevel_phi4.
[75] D. P. Kingma and J. Ba, arXiv:1412.6980.