Particle Monte Carlo methods for Lattice Field Theory

David Yallup
Kavli Institute for Cosmology Cambridge,
University of Cambridge
dy297@cam.ac.uk

Abstract

High-dimensional multimodal sampling problems from lattice field theory (LFT) have become important benchmarks for machine learning assisted sampling methods. We show that GPU-accelerated particle methods, Sequential Monte Carlo (SMC) and nested sampling, provide a strong classical baseline that matches or outperforms state-of-the-art neural samplers in sample quality and wall-clock time on standard scalar field theory benchmarks, while also estimating the partition function. Using only a single data-driven covariance for tuning, these methods achieve competitive performance without problem-specific structure, raising the bar for when learned proposals justify their training cost.

1 Introduction

Computing observables in strongly coupled quantum field theories (QFTs) remains a central challenge in fundamental physics. When perturbation theory breaks down, lattice field theory (LFT) offers a non-perturbative formulation via discretized Euclidean path integrals and has long underpinned precision studies in particle physics [1, 2]. In parallel, LFT-inspired statistical problems have recently become established benchmarks for machine-learning–based sampling methods [3–8], and ML approaches are now increasingly deployed within the LFT community itself [9–11]. These models provide attractive testbeds because they are high dimensional, induce strong parameter correlations, exhibit phase transitions, and are frequently multimodal—features known to stress both classical and modern samplers [12, 13].

Neural network constructed *trivializing maps* [14–16] represent one successful line of attack within LFT, and connect directly to broader developments in *neural samplers* inspired by advances in normalizing flows, diffusion models, and neural transport [17–28]. These methods aim to amortize sampling by learning transformations of an unnormalized target distribution. While promising results have been obtained for a range of LFT benchmarks, open questions remain concerning their scalability and robustness. Frontier QCD calculations require sampling in $\sim 10^8$ dimensions, far beyond current neural deployments. Auxiliary learning paradigms may help bridge this gap [29, 30]; however, robustly addressing topological challenges at increasingly fine spacing remains elusive. Moreover, many neural samplers rely on carefully engineered equivariance or problem-specific symmetry handling. How these approaches transfer to domains where the relevant symmetries are unknown *a priori*, or indeed how well they are then actually learning a trivializing map, remains poorly understood.

These considerations motivate a re-examination of classical baselines under modern computational constraints. Although traditional Monte Carlo algorithms were designed for CPUs and often rely on control-flow–heavy updates that map poorly to accelerators [31–33], recent evidence shows that careful GPU implementations of classical techniques, such as non-reversible parallel-tempering schemes [34], can outperform state-of-the-art neural samplers even on synthetic multimodal problems [12]. Motivated by this, we consider a broad family of accelerator-friendly *particle Monte*

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

Carlo methods [35–37] and evaluate them on a canonical multimodal LFT benchmark: the broken-phase scalar ϕ^4 theory [4]. Our findings show that simple particle-based methods—implemented in a batched, GPU-native framework and tuned only through a single data-driven covariance estimate—provide a surprisingly strong baseline. Particularly if we treat the problem as truly black-box. We introduce the ϕ^4 benchmark and its sampling challenges in Section 2, present our empirical evaluation in Section 3, and discuss implications for future work in Section 4.

2 Sampling Challenges in Lattice Field Theory

This section introduces the benchmark, outlines its characteristic sampling pathologies, and summarizes the particle-based Monte Carlo baselines we evaluate. Our primary experiments use a 10×10 lattice (state dimension $V = L^2$) and we later consider larger L to assess scaling behaviour.

Scalar ϕ^4 theory in 1+1D. Following Hackett et al. [4], we study the $\mathbb{Z}_2$–symmetric real scalar ϕ^4 model in 1+1 dimensions. After Wick rotation, we work on a two-dimensional Euclidean lattice $\Lambda = \{0, \ldots, L-1\}^D$ with $D = 2$, lattice spacing $a = 1$, and periodic boundary conditions $\phi_{x+L\hat{\mu}} = \phi_x$ for unit vectors $\hat{\mu}$. The Euclidean action is

$$S_E(\phi) = \sum_{x\in\Lambda} \left[\tfrac{1}{2} \sum_{\mu=1}^{D} (\phi_{x+\hat{\mu}} - \phi_x)^2 \;+\; \tfrac{1}{2} m_0^2\, \phi_x^2 \;+\; \lambda\, \phi_x^4 \right], \tag{1}$$

and defines the Boltzmann target

$$p(\phi) = \frac{1}{Z}\, e^{-S_E(\phi)}, \qquad Z = \int_{\mathbb{R}^V} e^{-S_E(\phi)}\, d\phi.$$

We typically take $m_0^2 = -4$ and $\lambda = 1$, giving a double-well potential and a bimodal magnetization distribution. As an order parameter we use the lattice-averaged magnetization

$$\langle \phi \rangle = \tfrac{1}{V} \sum_{x\in\Lambda} \phi_x,$$

whose marginal is bimodal due to the exact $\mathbb{Z}_2$ symmetry $\phi \mapsto -\phi$. Example field configurations at $\beta = 0$ and $\beta = 1$ appear in Figure 2.

To incorporate annealing or tempering, we follow the standard construction introducing an inverse temperature $\beta \in [0, 1]$ and a Gaussian reference

$$U_0(\phi) = \tfrac{1}{2} \sum_x \phi_x^2, \qquad p_\beta(\phi) \propto \exp\big(- [U_0(\phi) + \beta\, S_E(\phi)] \big).$$

Thus $\beta = 0$ yields the Gaussian base distribution and $\beta = 1$ recovers the physical ensemble. In physics applications, β is often used to interpolate between the free and interacting theories while retaining physical meaning—e.g. in scans toward criticality (see Section G). The auxiliary model retains the essential sampling challenges, and the severity of bimodality can be tuned through (m_0^2, λ).

Sampling challenges This benchmark concentrates several difficulties typical of lattice field theory: (i) multimodality and rare transitions between symmetry-related minima, which hinder global mixing; (ii) strong nearest-neighbor couplings that induce long autocorrelation times for local updates and lead to critical slowing down [38]; and (iii) barriers that become more pronounced as L increases. These properties stress both neural and classical samplers and motivate hardware-accelerated, particle-based baselines evaluated in this work. Related phenomena in gauge theories include topological freezing—suppressed tunneling between topological sectors at fine lattice spacing—which further complicates sampling [39]. As a natural extension, one can consider the Schwinger model (lattice QED in 1+1D) [7], which introduces gauge constraints and fermionic degrees of freedom and further amplifies these challenges.

Particle-based Monte Carlo methods. Hamiltonian (Hybrid) Monte Carlo (HMC) remains the standard tool for large-scale lattice simulations [40], yet near criticality it also suffers from long autocorrelation times and—in more realistic gauge theories—topological freezing. Motivated by the

need for scalable, accelerator-compatible alternatives, we consider simple particle-based methods with minimal tuning. These methods combine sequential importance tempering with lightweight MCMC rejuvenation steps and parallelize naturally on GPUs.

We evaluate three sequential Monte Carlo (SMC) samplers [41] and one nested sampling (NS) baseline, implemented via `blackjax` [42]: (i) SMC–RW, an adaptive tempered SMC scheme [43] using Gaussian random-walk proposals scaled by the particle covariance; (ii) SMC–IRMH, using an independent Metropolis–Hastings proposal tuned to the particle covariance, similar to a basic annealed importance sampling scheme [44]; (iii) SMC–HMC, replacing random-walk rejuvenation with short HMC trajectories preconditioned via window adaptation [45]; and (iv) NS, nested sampling using slice-sampling rejuvenation [46] with direction covariance tuned to the particle cloud [47]. Additional details appear in Section A.

Black-box samplers. A central theme of this work is the comparison between methods that exploit explicit problem structure and those that do not. All particle MC methods considered here are intentionally configured as *black-box samplers*: they incorporate no symmetry-aware moves or physics-specific modifications. To obtain a high-fidelity reference, we include an Adaptive HMC (AHMC) control that *does* exploit the exact $\mathbb{Z}_2$ symmetry via occasional global flips $\phi \mapsto -\phi$, which greatly improves mixing. Such symmetry moves could be added to particle methods as well, but are intentionally omitted to assess their intrinsic performance. For completeness, we compare against the neural sampler of [30] (see Section B) and additionally include a purely black-box neural sampler without encoded symmetries, clarifying the gap between physics-informed and genuinely symmetry-agnostic learned proposals.

3 Experiments

We first evaluate the sampling efficiency and quality of four particle-based methods alongside two variants of a neural sampler on the bimodal ϕ^4 scalar field theory at lattice size $L \times L$ with $L = 10$, mass parameter $m_0^2 = -4.0$, and quartic coupling $\lambda = 1.0$. Our goals are to assess both sample quality from the physical ensemble at $\beta = 1$ and computational efficiency. As a high-fidelity reference, we run a long Adaptive HMC (AHMC) chain. All experiments are run on an NVIDIA L4 GPU. The summary table of results is given in Table 1. Further detail is given in Section C.

Table 1: Sampling Quality and Performance Comparison. We report the mean and standard deviation of quality metrics computed against 10 reference sample sets from a long AHMC chain. MMD and W_2 measure discrepancy from ground truth (lower is better). The AHMC row establishes a baseline for inherent variance between different sets of true samples. All runtimes are in seconds on an NVIDIA L4 GPU (*AHMC takes around 200s to run on this GPU, the time listed is on CPU).

Method	MMD$\times$1000	$W_2 \times 100$	$\log Z$	Runtime (s)
AHMC (control)	2.96 ± 0.29	418.45 ± 1.09	—	7.4*
CNF [30]	6.04 ± 0.44	418.13 ± 1.07	—	1028.5
Black-box methods				
NS	3.70 ± 0.28	412.90 ± 0.71	-65.20	36.85
SMC-RW	7.45 ± 0.60	421.76 ± 0.61	-65.26	**7.74**
SMC-HMC	**3.43 ± 0.42**	416.80 ± 0.53	-65.64	8.15
SMC-IRMH	9.55 ± 0.85	425.77 ± 0.38	-66.17	34.95
CNF MLP	9.64 ± 0.31	**411.77 ± 1.43**	—	2450.5

We assess sample quality by comparing a number of standard quality metrics (see Section F). We report means (and standard errors) over multiple batches to ensure stability. Summary statistics appear in Table 1, and magnetization density profiles for particle are shown in Figure 1.

Additional experiments We conduct two supplementary studies. First, in Section D we scale to larger lattices with $L \in \{15, 18\}$, i.e., state dimension $V = L^2 \in \{225, 324\}$, finding that only SMC-IRMH starts to struggle, with all other algorithms scaling well and converging rapidly. Second, in Section E we consider how the hyperparameters of the particle methods can be tuned to achieve

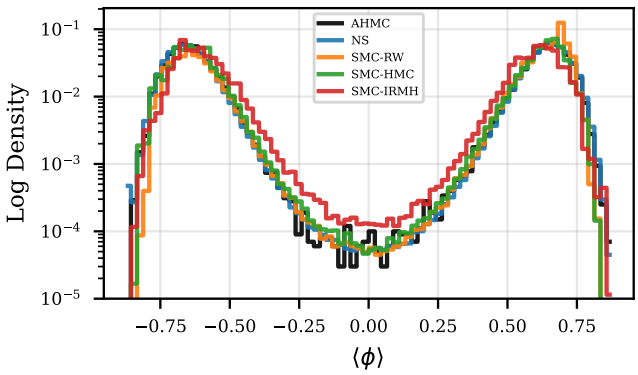


Figure 1: Posterior magnetization density histograms for AHMC (reference), NS, SMC-RW, SMC-HMC, and SMC-IRMH.

asymptotic convergence, demonstrating that even gradient-free SMC-RW can be tuned to achieve high accuracy using simple settings.

4 Discussion

Our experiments show that simple, minimally tuned particle-based methods perform strongly on the bimodal ϕ^4 benchmarks. As illustrated in Figures 1, 5 and 6, all methods except SMC–IRMH at the largest scales produce high-quality samples in the broken phase, with close agreement to the AHMC reference. Table 1 indicates that the three local walk methods (SMC–RW, NS, SMC–HMC) offer complementary strengths and achieve accuracy consistent with the reference across our metrics, while Table 3 shows that modest tuning of a few hyperparameters yields further gains with favorable runtime trade-offs, particularly when compared to the multiple GPU hours neural samplers take to train on similar problems [16]. Scaling results in Table 2 demonstrate competitive performance at larger lattice sizes. It is also possible to simply encode the $\mathbb{Z}_2$ symmetry into particle methods to improve mixing further.

The result of including a neural sampling baseline show two patterns. Firstly, that the cost of training these proposals vastly exceeds the cost of even un-tuned stochastic methods. Secondly the neural baselines fail to match the quality metrics of stochastic sampling despite the increased cost. The best W_2 result in Table 1 is achieved by a purely black-box neural sampler, but given the poor MMD and sampling efficiency this is likely due to over-concentration rather than a faithful representation of the target. After training, the neural proposal can be used in an MH independence sampler, giving rise to efficiencies of 85% and less than 1% for the CNF and CNF-MLP models respectively. While neural proposals can learn efficient representations of the target, and if symmetries are apriori known these can be encoded into the architecture to great effect, our results suggest two caveats. Firstly, apriori symmetries can be encoded into existing single particle HMC samplers just as easily in this case, solving the numerical sampling problem far more efficiently and scalably. Secondly, if these symmetries are not known, and we have to use black-box sampling (as is the case on many inverse problems), then simple particle methods can outperform neural proposals both in quality and wall-clock time without any problem-specific tuning.

5 Conclusion

Simple, minimally tuned particle-based samplers—using a data-driven particle covariance for scaling and preconditioning—achieve high accuracy and competitive or superior wall-clock performance on benchmark multimodal lattice problems. Across a number of metrics we demonstrate that local MCMC walks within SMC/NS can give state-of-the-art performance. These results challenge the view that particle methods are hard to tune or inherently hard to scale, and they raise the bar for when learned proposals justify their training cost on these tasks. Looking forward, we propose that initializing neural samplers with particle-based methods could be a promising direction.

Acknowledgements

We thank the authors of [30] for making their neural sampler code publicly available. The authors were supported by the research environment and infrastructure of the Handley Lab at the University of Cambridge.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

[1] Gross, F. et al. 50 Years of Quantum Chromodynamics. 12 2022.

[2] Kronfeld, A. S. Twenty-first Century Lattice Gauge Theory: Results from the QCD Lagrangian. *Ann. Rev. Nucl. Part. Sci.*, 62:265–284, 2012. doi: 10.1146/annurev-nucl-102711-094942.

[3] Albergo, M. S., Kanwar, G., and Shanahan, P. E. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Phys. Rev. D*, 100(3):034515, 2019. doi: 10.1103/PhysRevD.100.034515.

[4] Hackett, D. C., Hsieh, C.-C., Albergo, M. S., Boyda, D., Chen, J.-W., Chen, K.-F., Cranmer, K., Kanwar, G., and Shanahan, P. E. Flow-based sampling for multimodal distributions in lattice field theory. 7 2021.

[5] Abbott, R. et al. Sampling QCD field configurations with gauge-equivariant flow models. *PoS*, LATTICE2022:036, 2023. doi: 10.22323/1.430.0036.

[6] Abbott, R. et al. Aspects of scaling and scalability for flow-based sampling of lattice QCD. *Eur. Phys. J. A*, 59(11):257, 2023. doi: 10.1140/epja/s10050-023-01154-w.

[7] Albergo, M. S., Boyda, D., Cranmer, K., Hackett, D. C., Kanwar, G., Racanière, S., Rezende, D. J., Romero-López, F., Shanahan, P. E., and Urban, J. M. Flow-based sampling in the lattice Schwinger model at criticality. *Phys. Rev. D*, 106(1):014514, 2022. doi: 10.1103/PhysRevD.106.014514.

[8] Holderrieth, P., Albergo, M. S., and Jaakkola, T. Leaps: A discrete neural sampler via locally equivariant networks. *arXiv preprint arXiv:2502.10843*, 2025.

[9] Boyda, D. et al. Applications of Machine Learning to Lattice Quantum Field Theory. In *Snowmass 2021*, 2 2022.

[10] Kanwar, G. Flow-based sampling for lattice field theories. In *40th International Symposium on Lattice Field Theory*, 1 2024.

[11] Bialas, P., Korcyl, P., Stebel, T., and Zapolski, D. NeuMC – a package for neural sampling for lattice field theories. 3 2025.

[12] He, J., Du, Y., Vargas, F., Zhang, D., Padhy, S., OuYang, R., Gomes, C., and Hernández-Lobato, J. M. No trick, no treat: Pursuits and challenges towards simulation-free training of neural samplers, 2025. URL `https://arxiv.org/abs/2502.06685`.

[13] Grenioux, L., Noble, M., and Gabrie, M. Improving the evaluation of samplers on multi-modal targets. In *ICLR workshop on Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025.

[14] Luscher, M. Trivializing maps, the Wilson flow and the HMC algorithm. *Commun. Math. Phys.*, 293:899–919, 2010. doi: 10.1007/s00220-009-0953-7.

[15] Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport, 2019.

[16] Del Debbio, L., Marsh Rossney, J., and Wilson, M. Efficient modeling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability. *Phys. Rev. D*, 104:094507, Nov 2021. doi: 10.1103/PhysRevD.104.094507. URL `https://link.aps.org/doi/10.1103/PhysRevD.104.094507`.

[17] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. URL https://arxiv.org/abs/2011.13456.

[18] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[19] Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[20] Midgley, L. I., Stimper, V., Simm, G. N. C., Schölkopf, B., and Hernández-Lobato, J. M. Flow annealed importance sampling bootstrap, 2023. URL https://arxiv.org/abs/2208.01893.

[21] Matthews, A. G. D. G., Arbel, M., Rezende, D. J., and Doucet, A. Continual repeated annealed flow transport monte carlo, 2023. URL https://arxiv.org/abs/2201.13117.

[22] Gabrié, M., Rotskoff, G. M., and Vanden-Eijnden, E. Adaptive Monte Carlo augmented with normalizing flows. *Proc. Nat. Acad. Sci.*, 119(10):e2109420119, 2022. doi: 10.1073/pnas.2109420119.

[23] Blessing, D., Berner, J., Richter, L., and Neumann, G. Underdamped diffusion bridges with applications to sampling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Q1QTxFm0Is.

[24] Richter, L. and Berner, J. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=h4pNROsO06.

[25] Vargas, F., Padhy, S., Blessing, D., and Nüsken, N. Transport meets variational inference: Controlled monte carlo diffusions, 2025. URL https://arxiv.org/abs/2307.01050.

[26] Albergo, M. S. and Vanden-Eijnden, E. Nets: A non-equilibrium transport sampler, 2025. URL https://arxiv.org/abs/2410.02711.

[27] Grenioux, L., Noble, M., Gabrié, M., and Durmus, A. O. Stochastic localization via iterative posterior sampling, 2024. URL https://arxiv.org/abs/2402.10758.

[28] Zhang, D., Chen, R. T. Q., Liu, C.-H., Courville, A., and Bengio, Y. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=OIsahq1UYC.

[29] Albandea, D., Del Debbio, L., Hernández, P., Kenway, R., Rossney, J. M., and Ramos, A. Learning Trivializing Flows in a ϕ^4 theory from coarser lattices. *PoS*, LATTICE2023:013, 2024. doi: 10.22323/1.453.0013.

[30] Gerdes, M., de Haan, P., Rainone, C., Bondesan, R., and Cheng, M. C. N. Learning lattice quantum field theories with equivariant continuous flows. *SciPost Phys.*, 15(6):238, 2023. doi: 10.21468/SciPostPhys.15.6.238.

[31] Hoffman, M. D. and Sountsov, P. Tuning-free generalized hamiltonian monte carlo. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I. (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 7799–7813. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/hoffman22a.html.

[32] Hoffman, M., Radul, A., and Sountsov, P. An adaptive-mcmc scheme for setting trajectory lengths in hamiltonian monte carlo. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3907–3915. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/hoffman21a.html.

[33] Riou-Durand, L., Sountsov, P., Vogrinc, J., Margossian, C. C., and Power, S. Adaptive tuning for metropolis adjusted langevin trajectories, 2023. URL https://arxiv.org/abs/2210.12200.

[34] Syed, S., Bouchard-Côté, A., Deligiannidis, G., and Doucet, A. Non-reversible parallel tempering: A scalable highly parallel mcmc scheme. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2):321–350, December 2021. ISSN 1467-9868. doi: 10.1111/rssb.12464. URL http://dx.doi.org/10.1111/rssb.12464.

[35] Doucet, A., De Freitas, N., and Gordon, N. *An introduction to sequential Monte Carlo methods*. Springer Science & Business Media, 2001. URL https://www.stats.ox.ac.uk/~doucet/doucet_defreitas_gordon_smcbookintro.pdf. Accessed: 2024-08-21.

[36] Del Moral, P., Doucet, A., and Jasra, A. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[37] Skilling, J. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833 – 859, 2006. doi: 10.1214/06-BA127. URL https://doi.org/10.1214/06-BA127.

[38] Wolff, U. Critical slowing down. *Nuclear Physics B - Proceedings Supplements*, 17:93–102, 1990. ISSN 0920-5632. doi: https://doi.org/10.1016/0920-5632(90)90224-I. URL https://www.sciencedirect.com/science/article/pii/092056329090224I.

[39] Del Debbio, L., Manca, G. M., and Vicari, E. Critical slowing down of topological modes. *Phys. Lett. B*, 594:315–323, 2004. doi: 10.1016/j.physletb.2004.05.038.

[40] Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987. ISSN 0370-2693. doi: https://doi.org/10.1016/0370-2693(87)91197-X. URL https://www.sciencedirect.com/science/article/pii/037026938791197X.

[41] Chopin, N. and Papaspiliopoulos, O. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics. Springer International Publishing, 2020. ISBN 9783030478452. URL https://books.google.co.uk/books?id=ZZEAEAAAQBAJ.

[42] Cabezas, A., Corenflos, A., Lao, J., and Louf, R. Blackjax: Composable Bayesian inference in JAX, 2024.

[43] Fearnhead, P. and Taylor, B. M. An adaptive sequential monte carlo sampler, 2010. URL https://arxiv.org/abs/1005.1193.

[44] Neal, R. M. Annealed importance sampling, 1998. URL https://arxiv.org/abs/physics/9803008.

[45] Stan Development Team. Hmc algorithm parameters. https://mc-stan.org/docs/2_20/reference-manual/hmc-algorithm-parameters.html, 2023. Accessed: 2024-08-21.

[46] Neal, R. M. Slice sampling. *The Annals of Statistics*, 31(3):705 – 767, 2003. doi: 10.1214/aos/1056562461. URL https://doi.org/10.1214/aos/1056562461.

[47] Yallup, D., Kroupa, N., and Handley, W. Nested slice sampling. In *Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. URL https://openreview.net/forum?id=ekbkMSuPo4.

[48] Geyer, C. J. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473 – 483, 1992.

[49] Buchholz, A., Chopin, N., and Jacob, P. E. Adaptive tuning of hamiltonian monte carlo within sequential monte carlo, 2020. URL https://arxiv.org/abs/1808.07730.

[50] Hoffman, M. D. and Gelman, A. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. URL http://jmlr.org/papers/v15/hoffman14a.html.

[51] Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv e-prints*, art. arXiv:1701.02434, January 2017. doi: 10.48550/arXiv.1701.02434.

[52] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

[53] Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

A Algorithm Details

The outer kernels of all four main algorithms are largely similar. All feature the following free parameters: (i) population size, (ii) number of MCMC steps per tempering transition, and (iii) target effective sample size (ESS) for adaptive tempering (NS differs in using a fraction of particles to delete). These parameters trade off computational cost against accuracy: increasing the population size increases particle diversity but increases computational cost linearly; increasing the number of MCMC steps improves mixing with a linear cost increase; increasing the target ESS induces more tempering steps, improving accuracy at additional cost (for NS, decreasing the delete fraction toward zero has a similar effect).

We consider fairly standard settings for these parameters across our experiments.

- Population size: $N = 5000$ particles.
- MCMC steps per tempering transition: $n_{\text{MCMC}} = 3 \times V$.
- Target ESS: $\text{ESS}_{\text{target}} = 0.9$.
- NS delete fraction: $n_{\text{delete}} = 0.5$.

In Section B we decrease N to 1000 to match the batch size of the forward ODE solver. In Section E we systematically increase the target ESS to demonstrate asymptotic convergence and also set a higher baseline of $n_{\text{MCMC}} = 5 \times V$. The choice of inner kernel offers substantial opportunity for tuning; we forgo extensive tuning by basing all inner kernels on simple Gaussian covariances estimated from the particle population at each outer step, with the RW proposal scaled by $2.38^2/V$ to target optimal acceptance rates for RW MH algorithms [48]. The HMC proposal uses a diagonal mass matrix tuned in warm-up and then frozen throughout the run. The HMC is run with a fixed trajectory of 20 leapfrog steps, and the number of MCMC steps taken is reduced to a factor proportional to V, reflecting the higher per-step cost and faster mixing of each HMC step. Further gains could be made by online tuning of the mass matrix [49].

NS is distinctive, in that the NSS construction of [47] requires a successful MCMC step to satisfy the current likelihood constraint. Although the slice-sampling kernel (also tuned using the particle covariance) is configured to deliver efficiency comparable to the RW kernel, the additional work associated with enforcing the constraint (e.g., bracketing and shrinkage) typically leads to higher computational cost in practice.

The AHMC reference is comprised of a single long chain, with 5000 samples drawn in warm-up that is used to tune the diagonal of the mass matrix and step size, targeting an acceptance rate of 0.7 in the symplectic integrator. In particular we use the No-U-turn Sampling (NUTS) variant of [50, 51] HMC, leveraging the `blackjax` implementation of the window adaptation algorithm from Stan for tuning the HMC kernel parameters [45]. The NUTS chain is run for 10^6 iterations, thinning every 100 steps to reduce autocorrelation. To ensure good mixing, we interleave standard NUTS trajectories with global $\mathbb{Z}_2$ flips $\phi \mapsto -\phi$ every 20 steps. This is a simple problem-specific augmentation that is not available to the other methods, but it ensures that the AHMC reference is of high-fidelity. The resulting AHMC samples are then split into 10 separate tranches of 10^3 samples, and used to compute error bars on quality metrics as in Section F. All methods are run on an NVIDIA L4 GPU, except an M1 Studio Max that is used to compute CPU runtimes where noted. All experiments are run in `float32` precision.

B Neural Network training and Sampling Methods

To test the performance of Neural Sampling methods, we employ the methods as outlined in [30]. This trails a continuous normalizing flow (CNF) trained via minimization of the reverse Kullback-Leibler (KL) divergence. We perform the training on an NVIDIA L4 GPU, using a batch size of 1,024 samples and training for 1,000 steps. The flow is simulated using a simple Euler solver with a fixed step size of 0.02. We run on the same 10×10 ϕ^4 lattice field theory with parameters $M^2 = -4.0, \lambda = 1.0$ as in the main text, and report the training time of the algorithm as by far the dominant factor. Such a flow can be used as a proposal to generate samples via IRMH, where the efficiency of the proposal is determined by the quality of the trained flow. Some of the sampling cost can be considered *amortized* over multiple draws from the trained flow. To create a realistic *black-box* version, we alter the neural

network used to parameterize the flow to be a simple fully connected Multi-Layer Perceptron (MLP) with three hidden layers of 128 units each, using SiLU activations. We also use as input for this the raw field values as opposed to a physics-informed feature set as in [30]. Both methods are found to have reasonably converged after this training time, through tracking the KL divergence and ESS of samples drawn from the trained flow.

C Additional Experimental Details

This section compiles additional quantitative and qualitative results referenced in the main text. Table 1 compares sampling quality and runtime. Figure 1 shows posterior magnetization densities. Figure 3 summarizes magnetization across inverse temperature, and Figure 2 visualizes representative field configurations.

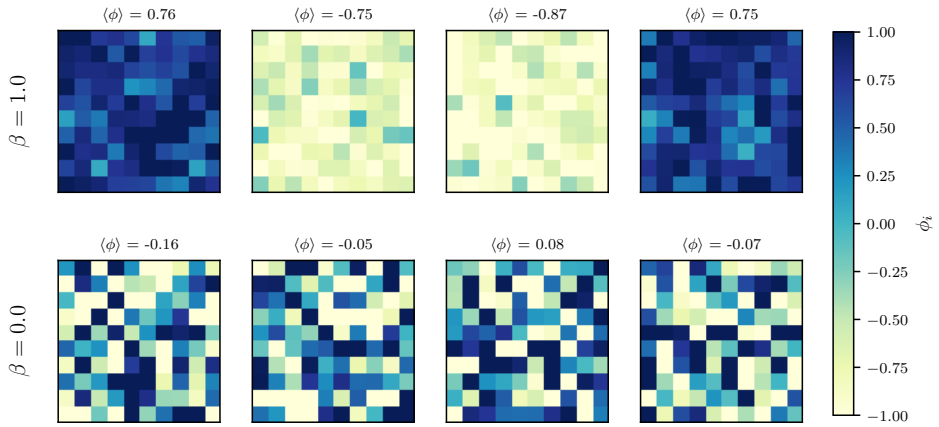
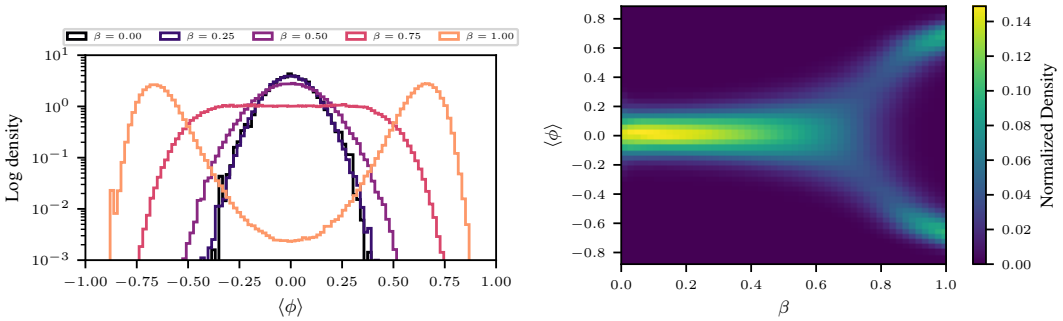


Figure 2: Eight 10×10 field configurations sampled by NS at $\beta = 1.0$ (top) and $\beta = 0.0$ (bottom). Colors indicate site-wise fields ϕ_i with a shared colorbar; each panel is annotated with its mean magnetization $\langle\phi\rangle$.



(a) (b)

Figure 3: Magnetization across inverse temperature β: (a) tempered histograms of $\langle\phi\rangle$ at five selected β values; (b) density over $(\beta, \langle\phi\rangle)$ across the full temperature range.

Aside from features highlighted in the main text, we also point to some other interesting features of the results. Firstly, in introducing an auxiliary variable, as shown in Figure 3, all methods are able to build a bridge that can connect the two modes of the distribution at $\beta = 1.0$. This is a known feature of tempering methods, where we note that NS is not strictly a tempering method, but it does introduce an auxiliary variable in the form of the likelihood constraint, and is a key reason for their success on this problem. Secondly, we note that all methods also give much higher statistics in the low density region at $\beta = 1.0$ than even the AHMC reference, this is again a feature of auxiliary variable methods that bridge from reference to target. Lastly, there are some statistical artifacts (bins with anomalously high weights) throughout the results. We consider that this is likely a fixable numerical artifact.

Beyond the main text, two patterns are noteworthy. Methods that employ auxiliary variables—temperature schedules in the SMC variants and likelihood constraints in NS—induce sequences of intermediate distributions that effectively bridge the two magnetization modes at $\beta = 1.0$ (see Figure 3). These methods also show greater coverage of the low-probability valley between modes at $\beta = 1.0$ than the short AHMC reference, consistent with improved inter-mode mixing along the bridge. Finally, there are some visible "statistical artifacts" (bins with anomalously high weights) throughout the results. We consider that this is likely a fixable numerical artifact from `float32` numerical precision of the weight calculations.

We also include a diagnostic plot of the MCMC kernel acceptance rate during SMC annealing in Figure 4. This plot shows the average acceptance rate of the internal MCMC kernel at each step of the annealing process for SMC-RW, SMC-IRMH, and SMC-HMC. This metric reveals the efficiency of the proposal mechanism as the algorithm transitions from the prior to the posterior. We see that SMC-IRMH starts with a high acceptance rate but quickly drops off as the annealing progresses, indicating that the independent proposal becomes less effective in exploring the target distribution. In contrast, SMC-RW and SMC-HMC maintain more stable acceptance rates throughout the annealing process, suggesting that their local proposal mechanisms are better suited to adapt to the changing landscape of the target distribution.

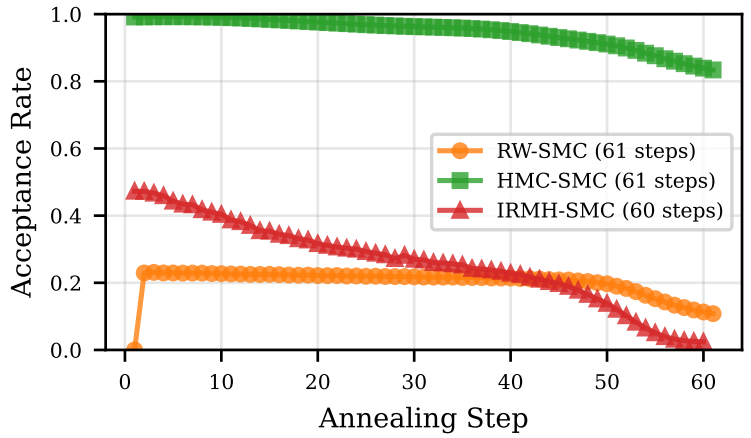


Figure 4: MCMC kernel acceptance rate during SMC annealing. This diagnostic plot shows the average acceptance rate of the internal MCMC kernel at each step of the annealing process for SMC-RW, SMC-IRMH, and SMC-HMC. This metric reveals the efficiency of the proposal mechanism as the algorithm transitions from the prior to the posterior.

D Scaling with Lattice Size

We increase the lattice size from $L = 10$ to $L \in \{15, 18\}$, corresponding to state dimensions $V = L^2 = 100, 225$, and 324. We fix the physical parameters at $m_0^2 = -4.0$ and $\lambda = 1.0$; as L grows, the posterior magnetization distribution becomes more sharply bimodal, increasing sampling difficulty. All algorithmic parameters are held fixed (population size $N = 5000$, MCMC steps per tempering transition $n_{\text{MCMC}} = 3 \times V$, target ESS $\text{ESS}_{\text{target}} = 0.9$). Results are summarized in Figure 5 and Table 2. SMC-IRMH degrades at the larger sizes, whereas NS, SMC-RW, and SMC-HMC maintain good performance and converge rapidly (as reflected by the mode-coverage metrics).

For these runs we report mode-coverage metrics in place of full-distribution discrepancies because storing all field configurations becomes costly at larger L; comprehensive evaluation with full state storage is left to future work. A higher-fidelity scaling study is needed to characterize computational scaling precisely, but runtimes remain modest across the tested sizes (see Table 2). At $L = 18$, the AHMC (reference) chain with 10^6 draws does not populate the low-density valley between modes (Figure 5c), indicating how sharply separated this problem has become. In theory all these methods can be scaled further, but more extensive tuning may be required to maintain high accuracy.

Table 2: Scaling performance and quality across increasing lattice dimensions. We report runtime, two metrics for evaluating the exploration of the bimodal posterior (Mode Balance and Mode Ratio), and the estimated log evidence ($\log Z$). A mode balance of 1.0 indicates an equal number of samples were drawn from each of the two posterior modes. AHMC serves as the reference.

Lattice Size	Method	Runtime (s)	Mode Balance	Mode Ratio	$\log Z$
10×10	AHMC (control)	216.7	1.001	1.000	—
	SMC-RW	6.7	0.986	1.096	-65.2
	SMC-HMC	6.7	0.999	1.764	-65.7
	NS	34.3	0.785	0.959	-65.4
	SMC-IRMH	10.7	0.998	1.220	-66.0
15×15	AHMC (control)	237.6	1.000	1.000	—
	SMC-RW	21.5	1.001	1.452	-146.4
	SMC-HMC	25.5	0.998	1.823	-148.5
	NS	175.6	0.848	1.052	-147.9
	SMC-IRMH	47.5	0.999	0.957	-151.7
18×18	AHMC (control)	237.4	1.000	1.000	—
	SMC-RW	46.0	1.007	2.063	-209.9
	SMC-HMC	73.6	0.999	1.083	-214.1
	NS	414.2	1.189	1.016	-212.9
	SMC-IRMH	114.8	1.011	0.604	-218.3

E Asymptotic Convergence of Particle Monte Carlo Methods

To study asymptotic behavior, we revisit the weakest-performing method (in terms of quality, but fastest in runtime) from the main experiments and show that increasing the number of tempering steps (via a higher target ESS) yields near-convergent results while retaining substantially lower runtime than neural samplers reported in the main text. We benchmark Sequential Monte Carlo with random-walk kernels (SMC-RW) against the same AHMC reference, running on the base $L = 10$ lattice with the same setting for parameters $\lambda = 1$, $m_0^2 = -4$.

We evaluate three SMC-RW configurations with $N = 5{,}000$ particles and an adaptive random-walk MCMC kernel using 500 inner transitions per tempering step, distinguished only by their target ESS for adaptive tempering:

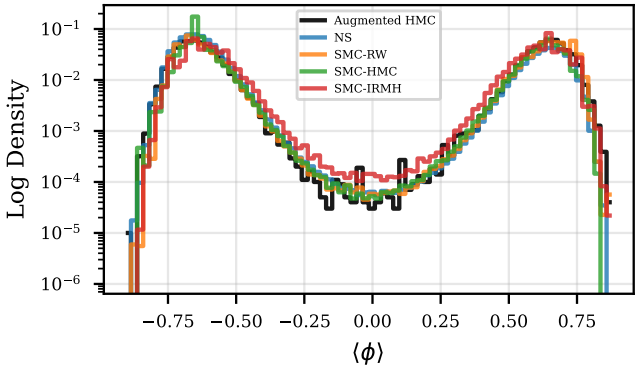
1. Standard: $\text{ESS}_{\text{target}} = 0.90$
2. High: $\text{ESS}_{\text{target}} = 0.99$
3. Max: $\text{ESS}_{\text{target}} = 0.999$

We note that there is a maximum target ESS of $1 - 1/N = 0.9998$ that can be achieved with N particles, so the Max configuration is close to this limit.

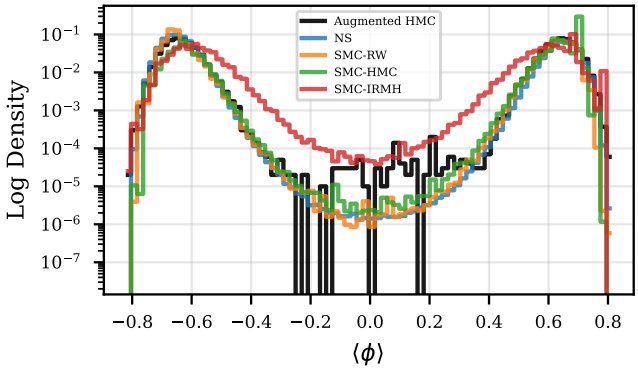
The results are summarized in Table 3 and Figure 6. As $\text{ESS}_{\text{target}}$ increases, the number of action (log-density) evaluations and hence runtime grow, but sample quality improves and approaches statistical compatibility with the AHMC control according to the reported metrics.

F Sample Quality Metrics

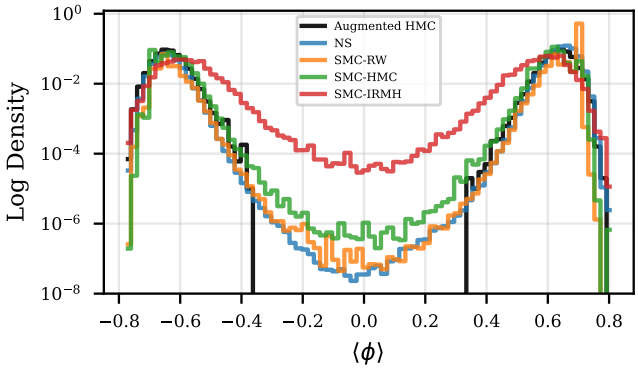
To provide a principled comparison of posterior sample quality, we employ two integral probability metrics to measure the statistical distance between the samples generated by a given method, P, and our reference distribution from AHMC, Q. For our analysis, we draw $n = m = 1000$ samples for each comparison.



(a) 10×10 Lattice



(b) 15×15 Lattice



(c) 18×18 Lattice

Figure 5: Posterior magnetization density as a function of increasing lattice size. Each panel compares the ability of the sampling methods to capture the bimodal posterior distribution. As the lattice dimension grows, the two modes become more sharply peaked and separated, increasing the difficulty of the sampling problem. Quantitative details relating to this scaling are listed in Table 2.

F.1 Maximum Mean Discrepancy (MMD)

We use the kernel two-sample discrepancy MMD [52], which measures the distance between the mean embeddings of two distributions in a reproducing kernel Hilbert space (RKHS). For a characteristic kernel k, the squared MMD is

$$\mathrm{MMD}^2(P, Q) = \mathbb{E}_{x,x'\sim P}\big[k(x, x')\big] - 2\,\mathbb{E}_{x\sim P,\, y\sim Q}\big[k(x, y)\big] + \mathbb{E}_{y,y'\sim Q}\big[k(y, y')\big].$$

With a characteristic kernel, $\mathrm{MMD}(P, Q) = 0$ if and only if $P = Q$. We use a Gaussian (RBF) kernel

$$k(x, y) = \exp\big(-\|x - y\|^2/(2\sigma^2)\big),$$

Table 3: Table showing the trade-off between sample quality and computational cost for the SMC-RW sampler when varying the target Effective Sample Size (ESS). A higher ESS value forces more frequent resampling, which improves quality at the cost of longer runtimes.

Method	MMD×1000	Mode Balance	$\log Z$	Runtime (s)
Augmented HMC (Control)	2.76 ± 0.28	1.000	–	113.2
SMC-RW (ESS=0.900)	7.18 ± 0.79	0.925	-65.50	8.8
SMC-RW (ESS=0.990)	4.86 ± 0.23	0.957	-65.48	18.4
SMC-RW (ESS=0.999)	3.28 ± 0.38	1.044	-65.43	49.9

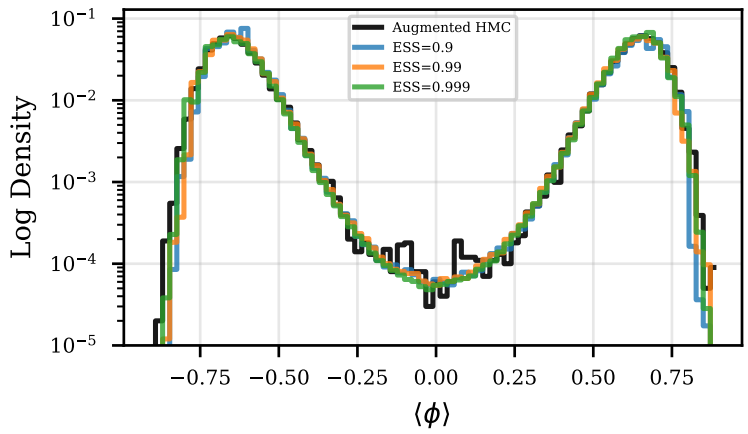


Figure 6: Effect of target ESS on posterior magnetization density for SMC-RW. Increasing $\text{ESS}_{\text{target}}$ from 0.90 to 0.999 induces a closer match to the bimodal distribution given by the AHMC (reference).

with bandwidth σ set by the median heuristic. MMD is particularly effective at detecting differences in the shape and moments of distributions.

F.2 2-Wasserstein Distance (W_2)

We compute a Sinkhorn-regularized 2-Wasserstein cost. Given samples $X = \{x_i\}_{i=1}^n$ and $Y = \{y_j\}_{j=1}^m$ with uniform weights $a = \frac{1}{n}\mathbf{1}_n$ and $b = \frac{1}{m}\mathbf{1}_m$, and squared Euclidean ground cost $C_{ij} = \|x_i - y_j\|^2$, the entropic-regularized optimal transport problem is

$$\mathrm{OT}_\varepsilon(X,Y) = \min_{T\geq 0} \sum_{i,j} T_{ij}\, C_{ij} + \varepsilon \sum_{i,j} T_{ij}\big(\log T_{ij} - 1\big) \quad \text{s.t.} \quad T\,\mathbf{1}_m = a, \;\; T^\top \mathbf{1}_n = b.$$

We solve this with the Sinkhorn algorithm [53], as implemented in `ott-jax`, using $\varepsilon = 10^{-3}$. We report

$$D_\varepsilon(X,Y) \;=\; \sqrt{\mathrm{OT}_\varepsilon(X,Y)},$$

i.e., the square root of the regularized objective value ("Sinkhorn cost"). This is not the exact W_2 distance for finite ε, but $\mathrm{OT}_\varepsilon(X,Y) \to W_2(X,Y)^2$ as $\varepsilon \downarrow 0$. This effectively measures the distance between the empirical distributions of X and Y, capturing differences in their support and overall geometry, with lower values indicating closer distributions.

F.3 Mode balance and mode-height ratio

Mode balance. Given a set of magnetization samples, we define the mode balance as the ratio of the number of positive samples to the number of negative samples. Samples exactly equal to zero are ignored. If there are no negative samples the score is $+\infty$ (all mass on the positive side); if there are no positive samples the score is 0 (all mass on the negative side). Values near 1 indicate a balanced occupancy of the two modes.

Histogram mode-height ratio. To compare peak sharpness against a reference, we construct histograms for the method and the reference over a common support and then compare their maximal heights on the positive and negative sides. Concretely, we form $B = 80$ equally spaced bins over the combined range of the method and reference samples, compute the method histogram either with provided sample weights (accumulated in the log-domain via log-sum-exp for numerical stability and then normalized) or as unweighted counts (normalized), and compute the reference histogram as unweighted counts (normalized). We split the bins at zero magnetization into a positive side and a negative side; if a side has no bins, we use a small floor (10^{-10}) for its maximum. For each side, we take the maximum (mode height) of the method histogram and divide by the corresponding maximum of the reference histogram, and we report the geometric mean of these two ratios. A value of 1 indicates matched peak sharpness in both modes relative to the reference; values greater (smaller) than 1 indicate sharper (flatter) modes on average.

G Alternative parameterizations

To avoid confusion with the annealing parameter β used in the main text, in this subsection we denote by κ a *physical* nearest-neighbor coupling that scales only the kinetic term.

A common, physically meaningful parameterization of lattice ϕ^4 writes the action as

$$S_E(\phi;\kappa,\lambda) = \frac{\kappa}{2}\sum_{x,\mu}\big(\phi_{x+\hat{\mu}} - \phi_x\big)^2 \;+\; \sum_x \Big[\phi_x^2 + \lambda\,(\phi_x^2 - 1)^2\Big], \tag{2}$$

see, e.g., [16]. Expanding the nearest-neighbor term and collecting coefficients gives

$$\frac{\kappa}{2}\sum_{x,\mu}\big(\phi_{x+\hat{\mu}} - \phi_x\big)^2 = -\,\kappa\sum_{x,\mu}\phi_x\,\phi_{x+\hat{\mu}} \;+\; \kappa D\sum_x \phi_x^2, \tag{3}$$

$$\phi_x^2 + \lambda(\phi_x^2 - 1)^2 = \lambda\phi_x^4 \;+\; (1-2\lambda)\phi_x^2 \;+\; \lambda, \tag{4}$$

so that, up to an additive constant $|\Lambda|\,\lambda$ that only shifts $\log Z$,

$$S_E(\phi;\kappa,\lambda) = -\,\kappa\sum_{x,\mu}\phi_x\,\phi_{x+\hat{\mu}} \;+\; (\kappa D + 1 - 2\lambda)\sum_x \phi_x^2 \;+\; \lambda\sum_x \phi_x^4 \;+\; \text{const.} \tag{5}$$

This makes explicit that κ strengthens nearest-neighbor correlations and shifts the effective quadratic term by κD. We do not use this parameterization in our experiments. Our experiments target the Gaussian-regularized surrogate $\tilde{p}(\phi) \propto \exp(-[U_0(\phi) + S_E(\phi)])$ and use the main-text β purely as an annealing parameter to bridge between distributions. We include (5) to illustrate a physics-meaningful κ that could be exploited in future work to scan for criticality with particle methods.