# Neural Triangular Transport Maps for Sampling in Lattice QCD

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Lattice field theories are fundamental testbeds for computational physics, yet sampling their Boltzmann distributions remains challenging due to multimodality and long-range correlations. While normalizing flows offer a promising alternative, their scalability to large lattices remains a challenge. We propose sparse triangular transport maps that explicitly encode the conditional independence structure of the lattice graph under periodic boundary conditions using monotone rectified neural networks (MRNN). We introduce a comprehensive framework for triangular transport maps that navigates the fundamental trade-off between *exact sparsity* (respecting marginal conditional independence in the target distribution) and *approximate sparsity* (computational tractability without fill-ins). Unlike dense normalizing flows that suffer from $\mathcal{O}(N^2)$ dependencies, our approach leverages locality to reduce complexity to $\mathcal{O}(N)$ while maintaining expressivity. Using $\phi^4$ in two dimensions as a controlled setting, we analyze how node labelings (orderings) affect sparsity and performance of triangular maps. We compare against Hybrid Monte Carlo (HMC) and established flow approaches (RealNVP). Our results suggest that structure-exploiting triangular transports deliver better scaling and competitive decorrelation compared to dense or coupling-based flows, while preserving physical symmetries via localized stencils.

## 1 Introduction

Lattice field theories provide a non-perturbative framework for fundamental physics, but their study is often constrained by the computational cost of sampling from the high-dimensional Boltzmann distribution, $P[\phi] \propto e^{-S[\phi]}$. While standard MCMC methods like Hybrid Monte Carlo (HMC) are asymptotically exact, they are notoriously hampered by critical slowing down near phase transitions, where autocorrelation times grow exponentially [17].

To address this bottleneck, normalizing flows (NFs) have recently been introduced to lattice field theory, demonstrating the potential for orders-of-magnitude improvements in sampling efficiency [2, 3, 6]. However, existing architectures struggle to scale to large lattice volumes because their inherent design limits parallelizability, requiring that the entire state is processed sequentially through the network's layers, which is inefficient for large configurations [1]. In this work, we propose a solution by developing sparse triangular transport maps, which leverage the use of Knothe-Rosenblatt rearrangements.

Our method achieves linear scaling in lattice size $N$. We accomplish this by constraining each output of the map to depend only on a local neighborhood of preceding variables, determined by a specific node ordering. This approach navigates the crucial trade-off between exact sparsity, which requires modeling computationally expensive "fill-in" effects from marginalization, and approximate sparsity, which enforces strict locality for tractability. Any error introduced by this approximation is corrected

Submitted to the AI for Science workshop (NeurIPS 2025).

by a final Metropolis-Hastings step, guaranteeing exactness. Each map component is parameterized as a monotone rectified neural network (MRNN), an integral of a strictly positive neural network, which ensures invertibility and a tractable Jacobian.

**Contributions:** We present several innovations for sampling in lattice field theory:

1. We use Monotone Rectified Neural Networks (MRNNs) to flexibly define the map's invertible components. This avoids the rigidity of polynomial-based methods, which require presetting the polynomial degree.
2. We introduce a **conditionally sparse triangular map** for lattice models using MRNN components, reducing the computational and memory complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.
3. We provide a **systematic analysis of labeling strategies**, demonstrating how the choice of site ordering directly controls the map's sparsity pattern and performance.
4. We empirically demonstrate that on the 2D $\phi^4$ theory the method shows competitive sampling efficiency, establishing a robust framework for future extensions to gauge theories.

## 2 Related Work

Our work is positioned at the intersection of machine learning for physics, generative modeling, and classical methods for sparse systems.

Pioneering work by [2] introduced normalizing flows to lattice physics using the RealNVP architecture [8], later refining the model by replacing the initial dense MLP coupling layers with convolutional neural networks to enforce parameter sharing and improve scalability [3]. These models are computationally efficient but must be stacked deeply to capture long-range physical correlations. A parallel line of research has focused on enforcing physical symmetries, leading to gauge-equivariant flows [11]. While powerful, these approaches can be computationally expensive, often relying on continuous normalizing flows that require costly ODE solvers [5]. Despite this progress, developing flow architectures that are both expressive and can scale to large, physically relevant lattices remains a central challenge for the field [16].

An alternative to coupling-based architectures are triangular transport maps [13, 4, 14], which correspond to autoregressive models. These models are highly expressive and are known to be universal approximators [10]. However, their power comes with a critical drawback for sampling: generating a single sample is an inherently sequential process with $\mathcal{O}(N^2)$ complexity, making dense autoregressive models slow for large systems. Our work directly targets this sampling bottleneck. Recent work has explored monotone parameterizations using polynomials and structure exploitation [4, 7].In our case we want to construct maps that respect the conditional independence structure of the target distribution, using the relation between graphical models and conditional independence [18]. The primary obstacle is the phenomenon of "fill-in," where marginalizing variables introduces dense dependencies not present in the original model. This is a classic challenge, known as the minimum fill-in problem in the context of sparse Cholesky factorization in numerical linear algebra [9] and as a core problem for exact inference in probabilistic graphical models [12].

## 3 The $\phi^4$ Lattice Field Theory

Consider a $D$-dimensional hypercubic lattice $\Lambda = (\mathbb{Z}/L\mathbb{Z})^D$ with $N_{sites} = L^D$ sites, where $L$ is the extent in each dimension. A scalar field $\phi_x \in \mathbb{R}$ is defined at each site $x \in \Lambda$. The Euclidean action for the $\phi^4$ theory is given by:

$$S[\phi] = \sum_{x\in\Lambda} \left[ \frac{1}{2} \sum_{\mu=1}^{D} (\phi_{x+\hat{\mu}} - \phi_x)^2 + \frac{m_0^2}{2}\phi_x^2 + \frac{\lambda_0}{4!}\phi_x^4 \right] \tag{1}$$

where $\phi_{x+\hat{\mu}}$ is the field at the site adjacent to $x$ in the positive $\mu$-direction, $m_0^2$ is the bare mass-squared parameter, and $\lambda_0$ is the bare coupling constant. We assume periodic boundary conditions $\phi_{x+L\hat{\mu}} = \phi_x$. The induced Gibbs distribution is a positive Markov random field (MRF) with cliques given by on-site and nearest-neighbor interactions:

$$P[\phi] = \frac{1}{Z} e^{-S[\phi]} \tag{2}$$

where $Z = \int \mathcal{D}\phi\, e^{-S[\phi]}$ is the partition function, and $\mathcal{D}\phi = \prod_{x\in\Lambda} d\phi_x$. This Markov property, $\phi_x \perp \phi_{\Lambda\setminus(\{x\}\cup\mathcal{N}(x))}|\phi_{\mathcal{N}(x)}$, will be the key to our sparse construction.

# 4 Triangular Maps

Transport maps learn a diffeomorphism $T : \mathcal{Z} \to \mathcal{X}$ between a simple base distribution $p_Z(z)$ (e.g., a standard $D \cdot N_{sites}$-dimensional Gaussian) over $z \in \mathcal{Z}$ and a complex target distribution $p_\Phi(\phi)$ (approximating $P[\phi]$) over $\phi \in \mathcal{X}$. The KR-type maps are uniquely defined once an ordering of coordinates is chosen. In general, orderings impact both the expressivity and the computational structure, especially if we look closely into approximate sparsity. If $\phi = T(z)$, the change of variables formula gives:

$$p_\Phi(\phi) = p_Z(T^{-1}(\phi))\,|\det J_{T^{-1}}(\phi)| \tag{3}$$

or, equivalently, for $z = T^{-1}(\phi)$, $p_\Phi(T(z)) = p_Z(z)\,|\det J_T(z)|^{-1}$, where $J_T(z)$ is the Jacobian matrix of the transformation $T$ at $z$. We impose an ordering on the $N_{sites}$ components of $z = (z_0, \ldots, z_{N_{sites}-1})$ and $\phi = (\phi_0, \ldots, \phi_{N_{sites}-1})$. The triangular map has the form where each output component $\phi_j$ depends on the corresponding input $z_j$ and all preceding input components $z_{<j}$. The triangular map $T : \mathcal{Z} \to \mathcal{X}$ is defined component-wise as $\phi = T(z)$, such that each component $\phi_j$ is generated as:

$$\begin{aligned}
\phi_0 &= T_0(z_0) \\
\phi_1 &= T_1(z_1; z_0) \\
&\vdots \\
\phi_j &= T_j(z_j; z_0, z_1, \ldots, z_{j-1}) \quad \text{or simply } T_j(z_j; z_{<j})
\end{aligned} \tag{4}$$

In this structure, the $j$-th component function $T_j$ takes the $j$-th base variable $z_j$ as its primary input (the variable with respect to which it is made monotonic and invertible for the transformation) and all preceding base variables $z_{<j} = (z_0, \ldots, z_{j-1})$ as conditioning context or parameters. This autoregressive definition ensures that the Jacobian matrix of the transformation, $J_T(z)$ with elements $(J_T)_{ij} = \frac{\partial \phi_i}{\partial z_j}$, is lower triangular. The determinant is then simply the product of the diagonal entries: $\det J_T(z) = \prod_{j=0}^{d-1} \frac{\partial \phi_j}{\partial z_j}$. A specific parameterization for each component $T_j$ that ensures invertibility with respect to $z_j$ and a positive partial derivative $\frac{\partial \phi_j}{\partial z_j}$ is the Monotone Rectified component:

$$\phi_j = T_j(z_j; z_{<j}) = f_j(z_{<j}) + \int_0^{z_j} r(g_j(s, z_{<j}))ds \tag{5}$$

Crucially, we parameterize the shift function $f_j$ and the scale integrand $g_j$ using neural networks. This approach, which we term MRNN, contrasts with prior work that typically uses expansions of orthogonal polynomials [4]. The use of neural networks offers a greater flexibility, and the universal approximation property of neural networks ensures that the existence guarantees provided by the polynomial formulation still hold.

**Why Neural Networks over Polynomials?** Neural networks provide a non-parametric, adaptive parameterization for the high-dimensional and a priori unknown target distributions of lattice theories. In contrast to fixed-degree polynomial expansions, they learn the required functional basis and complex conditional dependencies directly from data. Universal approximation theorems formally guarantee their expressive capacity is at least equivalent to that of polynomial maps.

$r : \mathbb{R} \to \mathbb{R}^+$ is a strictly positive rectification function (e.g., $r(s) = \exp(s)$ or Softplus). This ensures $\frac{\partial \phi_j}{\partial z_j} > 0$. The partial derivative required for the Jacobian determinant is $\frac{\partial \phi_j}{\partial z_j} = r(g_j(z_j, z_{<j}))$. Thus, the log-determinant of the full map $T$ is $\log|\det J_T(z)| = \sum_{j=0}^{N_{sites}-1} \log r(g_j(z_j, z_{<j}))$. The integral in Eq. (5) is one-dimensional and can be approximated using a change of variables and numerical quadrature:

$$\int_0^{z_j} r(g_j(s, z_{<j}))ds = z_j \int_0^1 r(g_j(tz_j, z_{<j}))dt \approx z_j \sum_{q=1}^{Q} w^{(q)} r(g_j(t^{(q)} z_j, z_{<j})) \tag{6}$$

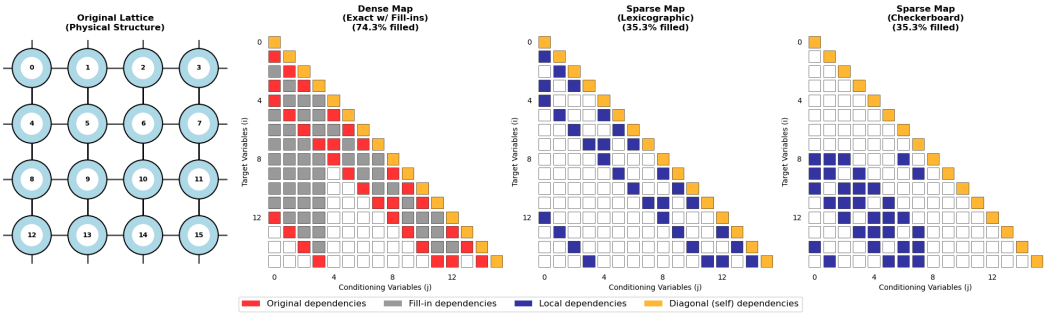



Figure 1: Dependency structures for a triangular map on a $4 \times 4$ lattice. It contrasts the dense, exact map for a lexicographic ordering (including "fill-in") with the enforced sparse maps for lexicographic and checkerboard ordering.

where $(w^{(q)}, t^{(q)})$ are the weights and nodes of a chosen quadrature rule (e.g., Gauss-Legendre) on $[0, 1]$, and $Q$ is the number of quadrature points. This formulation is highly efficient for GPU computation. For a batch of size $B$, the evaluation of $g_j$ can be parallelized across the $B \times Q$ inputs corresponding to the quadrature points. The component evaluation reduces largely to tensor contractions:

$$T_j(z_j; z_{<j}) \approx f_j(z_{<j}) + z_j \mathbf{w}^\top r(g_j(\mathbf{t} z_j, z_{<j})) \tag{7}$$

This allows the integral for every site in a large batch to be estimated with a single, highly parallelized forward pass through the networks $f_j$ and $g_j$.

# 5 From Conditional Independence to Sparse Triangular Maps

The objective is to construct an efficient autoregressive model, or triangular map, for a probability distribution $P[\phi]$ defined by a local action. The joint distribution is factorized as:

$$P[\phi] = \prod_{k=1}^{N_{\text{sites}}} P(\phi_k | \phi_{<k})$$

where $\phi_{<k} = (\phi_0, \ldots, \phi_{k-1})$ for a chosen ordering of the lattice sites. A fundamental challenge arises from this factorization. While the original distribution exhibits the local dependencies of a Markov Random Field, the exact conditionals $P(\phi_k|\phi_{<k})$ are generally dense. This is because marginalizing out "future" variables ($\phi_l$ for $l > k$) induces long-range correlations among the remaining variables. This phenomenon, known as fill-in, is analogous to the new non-zero entries that appear during the Cholesky factorization of a sparse matrix [15]. Finding an ordering that minimizes this fill-in is an NP-complete problem, rendering the construction of an exactly sparse map computationally intractable. To ensure scalability, we instead enforce approximate sparsity. This is achieved by restricting the dependencies of the $j$-th component of our triangular map, $\phi_j = T_j(z_j; \{\phi_i\}_{i \in \mathcal{C}(j)})$, to a local conditioning set $\mathcal{C}(j)$. Motivated by the physical locality of the action, we define this set as the "past neighbors" of site $j$:

$$\mathcal{C}(j) = N_p(j) \equiv N(j) \cap \{0, \ldots, j-1\}$$

where $N(j)$ is the set of immediate neighbors of site $j$ on the lattice (for a concrete structure of the distribution see A.4). This formulation presents a clear trade-off. By defining the conditioning context via $N_p(j)$, the size of the set is bounded by the lattice coordination number (e.g., $2D$), guaranteeing that the map evaluation scales linearly with the system volume, $\mathcal{O}(N_{\text{sites}})$. However, by ignoring the fill-in from marginalization, the map only approximates the true conditional structure. Consequently, to sample exactly from the target distribution, this approximate map must be used as a proposal within a Metropolis-Hastings correction framework. The quality of this approximation, and thus the overall sampling efficiency, critically depends on the chosen variable ordering.

## 5.1 Orderings and Sparsity

The ordering determines which neighbors are "past" (and thus available as conditioning variables) and which are "future" (and must be marginalized over implicitly). We want an ordering that

maximizes the information captured by the past neighbors. Our key insight is that while exact sparsity requires considering marginalization graphs with inevitable fill-ins during triangular decomposition, approximate sparsity based on conditional independence patterns leads computationally efficient maps at the cost of reduced expressivity. This mirrors classical fill-in behavior in sparse Cholesky factorizations [15]. We investigate three strategies:

**Lexicographic Ordering** Sites are ordered row-by-row: $\pi(i) = (i \mod L, \lfloor i/L \rfloor)$ for 2D lattices. The advantages are the simple implementation, predictable structure, however it creates asymmetric dependencies, poor for periodic boundaries.

**Checkerboard Ordering** Alternates between "black" and "white" sites as in checkerboard pattern. It is natural for bipartite lattices, symmetric dependencies, but it requires two-stage generation. The lattice is divided into even and odd sites. All even sites are ordered first, followed by all odd sites. When modeling the odd sites, all their neighbors (which are even) are in the preceding context. This maximizes the size of $N_p(j)$ to $2D$ for the second half of the variables. This structure provides the most complete local information for the sparse map.

**Max-Min Distance Ordering** It greedily selects the next site to maximize minimum distance from already-ordered sites:

$$\pi(j) = \arg \max_{x \in \Lambda \setminus \{\pi(1),\ldots,\pi(j-1)\}} \min_{i<j} d(x, \pi(i)) \tag{8}$$

This creates an optimal neighborhood preservation and balanced dependencies. However we have to account for higher reprocessing cost.

Periodic Boundary Conditions (PBCs) introduce topological complexity. Sites that are physically adjacent may be far apart in the ordering due to boundary wrap-around (e.g., site 0 and site $L-1$ in 1D). We rely on the flexibility of the MRNN parameterization and the subsequent MCMC correction to ensure exactness. The effect of the different orderings on the structure of the triangular map for a lattice problem can be seen in Figure 2. Here we represented a $L = 4$ lattice, with periodic boundary conditions, and show the difference between the exact conditional independence and enforced sparse representation. A more detailed analysis of the scaling behavior of fill-ins and an overall overview on the effect of the different orderings can be seen in A.5.

# 6 Experiments

We evaluate our proposed sparse triangular maps on the 2D $\phi^4$ theory ($D = 2$). The map parameters are optimized by minimizing the variational free energy, $\mathcal{L}(\theta) = \mathbb{E}_{z\sim p_Z(z)}[S[T_\theta(z)] - \log|\det J_{T_\theta}(z)|]$, via stochastic gradient descent (see A.1). This is equivalent to minimizing the reverse Kullback-Leibler divergence between the model distribution $p_\Phi$ and the target Gibbs distribution $P[\phi]$. To ensure exact sampling from the target distribution, we use the trained map as a proposal within an Independent Metropolis-Hastings (IMH) algorithm [2] (see A.2).

**Setup.** We consider an $L = 8$ lattice ($N = 64$ sites) with periodic boundary conditions. The $\phi^4$ theory parameters are fixed at $m_0^2 = -4.0$ and $\lambda_0 = 8.0$, placing the system in the challenging broken-symmetry phase near the critical line (moderate correlation lengths). The primary performance metric is the effective sample size (ESS) (see A.3).

**Model and Training.** Unless specified otherwise, maps are constructed from Monotone Rectified Neural Network (MRNN) components. The neural network for each component $T_j$ consists of 3 hidden layers (64 units each, GELU activation). Monotonicity is enforced via a Softplus activation on the final layer of the integrand network, and the required integral is approximated using a 15-point Gauss-Legendre quadrature. All models were trained for 3000 epochs using the AdamW optimizer (initial LR $10^{-3}$, weight decay $10^{-5}$) with a batch size of 256. A cosine annealing schedule decayed the learning rate to a minimum of $10^{-6}$.

## 6.1 Impact of Variable Ordering and Neighborhood Size

We investigate the fundamental trade-off between the sparsity of the triangular map and its expressivity by analyzing how different variable ordering strategies and conditioning neighborhood sizes affect performance.

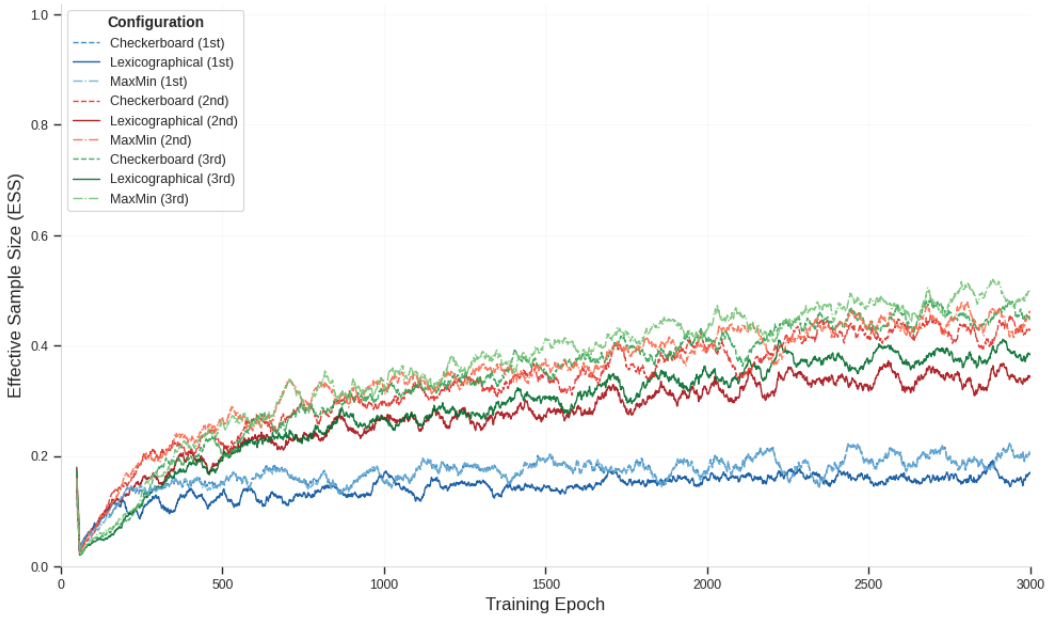



Figure 2: Effective Sample Size (ESS) as a function of training epochs for the nine model configurations. Each panel corresponds to a different variable ordering strategy (Lexicographical, Checkerboard, MaxMin). Within each panel, lines represent models trained with cumulatively increasing neighborhood orders (1st, 2nd, and 3rd). Expanding the conditioning set consistently improves sampling efficiency, with the MaxMin ordering achieving the highest overall performance.

We evaluated three distinct ordering strategies: a standard **Lexicographical** ordering, a physics-motivated **Checkerboard** (CB) ordering, and a **MaxMin** ordering designed to maximize spatial separation between causally dependent variables. For each ordering, we systematically increased the map's complexity by cumulatively expanding the conditioning neighborhood: **1st-order** (nearest-neighbors), **2nd-order** (including diagonals), and **3rd-order** (including "knight-moves").

We analyzed these nine configurations based on the realized sparsity (Avg. $|C(j)|$) and sampling efficiency (ESS). As visualized in Figure 2, increasing the neighborhood order consistently improves the final ESS across all orderings, demonstrating that a richer local context allows the model to better capture the underlying physics. The MaxMin ordering achieves the highest overall performance (however similiar in performance to the Checkerboard ordering), confirming that its structure naturally better preserver long-range interactions providing a more effective conditioning context compared to the other strategies.

### 6.2 Architecture Comparison and Scalability

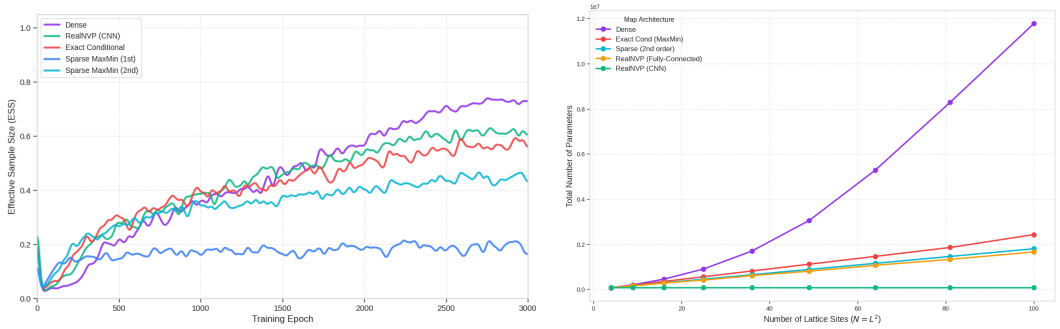
We perform a direct benchmark of various flow-based architectures to determine the most effective and computationally efficient design for lattice field theory sampling. We compare the expressive power of our MRNN models against a convolutional RealNVP baseline.

Five architectures are compared: (1) A **Dense** triangular map (lexicographical ordering), serving as an upper-bound for expressivity but with prohibitive parameter count and quadratic scaling. (2) A **RealNVP (CNN)** model, built from 8 coupling layers, representative of standard flow models for structured data. (3-5) **Sparse Triangular Maps** use MaxMin ordering, with dependencies restricted to 1st-order neighbors, 2nd-order neighbors, and the **Exact Conditional** dependencies derived from graph elimination.

The results are summarized in Figure 3. The CNN-based RealNVP performs competitively, achieving an ESS comparable to the Exact Conditional map. The approximate sparse maps (2nd order) perform slightly worse, highlighting the challenge of hand-picking an optimal, fixed-size conditioning set.

However, a crucial distinction emerges regarding scalability. The RealNVP parameter count is nearly constant $\mathcal{O}(1)$ with respect to the lattice size $N$. Yet, its architecture is inherently sequential; the entire configuration must pass through the deep stack of coupling layers, limiting parallelization primarily to the batch dimension.

In contrast, while the sparse MRNN's parameter count grows linearly $\mathcal{O}(N)$, its architecture is massively parallel. Each of the $N$ map components is an independent neural network that requires only the base sample $z_j$ and its small, local conditioning set $\{z_k\}_{k \in C(j)}$. This structure allows for spatial parallelism, where all $N$ components can be computed simultaneously. Therefore, the MRNN's superior parallelizability makes it a far more scalable and computationally efficient architecture for large lattices.




(a) ESS evolution during training.

(b) Final ESS comparison.

Figure 3: Performance comparison of various flow-based architectures. Subfigure (a) plots the ESS during training. Subfigure (b) summarizes the final ESS, demonstrating that the sparse triangular maps achieve superior sampling efficiency compared to the RealNVP baseline.

### 6.3 Statistical Error and Physical Observables

We analyze the statistical error of physical observables as a function of the number of generated samples, comparing our optimized triangular map against a standard Hybrid Monte Carlo (HMC) sampler.

The HMC sampler uses a standard leapfrog integrator with 10 steps. The integrator step size, $\epsilon$, was dynamically tuned to achieve a target acceptance rate of $\approx 70\%$. The implementation correctly handles the lattice's periodic boundary conditions using circular shifts in the force term computation. The triangular map, used within the IMH framework, was tuned for a 50% acceptance rate (following the structure of [2]).

For each sampler, we generated a chain of 20,000 configurations, discarding the first 2000 samples as burn-in. We measured the energy $\langle E \rangle$ and the susceptibility $\chi_2$ (for exact definition see B). The statistical error was estimated using the bootstrap method (500 resamples, 68% confidence interval) for varying sub-sample sizes ($N = 200$ up to $N = 20,000$). The results, presented in Figure 4, confirm that the triangular map closely follows the ideal $1/\sqrt{N}$ scaling behavior, achieving a slightly lower statistical error for a given number of samples than the HMC method for the susceptibility.

## 7 Conclusion and Future Outlook

**Limitations** Triangular transport maps do not perform dimensionality reduction, meaning the latent space must have the same dimension as the configuration space. Furthermore, approximation quality can sometimes degrade for configurations far from the expected support of the function class, and performance remains dependent on the predefined ordering.

This work introduces a highly scalable framework for sampling in lattice field theories using triangular transport maps. By leveraging the inherent locality of the physical action, we construct maps with $\mathcal{O}(N)$ complexity, a significant improvement over the $\mathcal{O}(N^2)$ scaling of dense autoregressive models. Our central contribution is a principled navigation of the trade-off between exactness and efficiency.

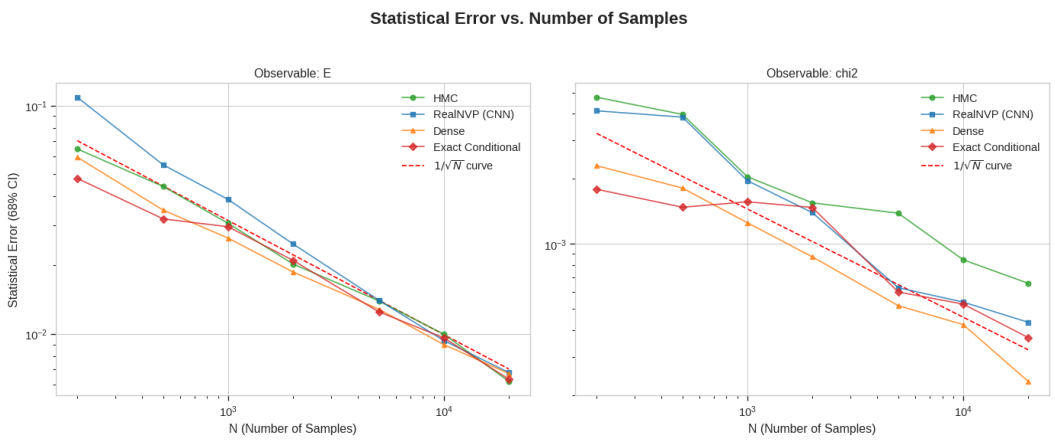



Figure 4: The statistical error of the measured energy and susceptibility as a function of the number of samples, $N$. The plot compares HMC to the transport maps. The solid lines show the statistical error, while the red dashed line represents the theoretical $1/\sqrt{N}$ behavior of an ideal sampler.

We demonstrate that while an exact triangular map requires modeling dense, non-local dependencies ("fill-in") created by marginalization, an approximate sparse map—conditioning only on local, preceding physical neighbors—provides a powerful and scalable alternative.

We show that parameterizing the map components with Monotone Rectified Neural Networks (MRNNs) offers superior flexibility and expressivity compared to traditional polynomial-based methods. Any error introduced by the sparsity approximation is corrected by a final Metropolis-Hastings step, guaranteeing that the resulting samples are drawn from the exact Boltzmann distribution. Our experiments on the 2D $\phi^4$ theory confirm the effectiveness of this approach. We systematically show that physics-informed orderings, like the checkerboard or MaxMin pattern, outperform simpler ones. The method produces physical observables with lower statistical error for a given number of samples, closely tracking the ideal $1/\sqrt{N}$ scaling and confirming its practical advantage, especially the parallelizability, for physics simulations.

Our framework establishes an alternative foundation for parallelizable samplers in lattice QCD, with several exciting avenues for future work.

The most significant frontier is the application to non-Abelian gauge theories. In these theories, variables are elements of a Lie group (e.g., SU(N)) associated with the links of the lattice, and the action is constructed from gauge-invariant objects. For example, the Wilson gauge action is built from plaquette variables $U_\square$:

$$S[U] = \beta \sum_{\square} \mathrm{Re}\,\mathrm{Tr}(1 - U_\square) \tag{9}$$

Extending our framework requires developing gauge-equivariant maps on the SU(N) group manifold that rigorously preserve local gauge symmetry. A promising path involves designing gauge-equivariant MRNNs that condition on local, gauge-covariant stencils (e.g., small Wilson loops). This would involve parameterizing transformations in the Lie algebra via exponential coordinates and incorporating corrections for the Haar measure to ensure the map is properly defined on the group.

While we have shown the power of sparsity, there is room for further optimization. Instead of relying on fixed, predefined orderings, it may be possible to learn an optimal ordering as part of the training process or develop adaptive orderings that change during sampling to improve decorrelation.

# References

[1] Ryan Abbott, Michael S. Albergo, Aleksandar Botev, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Alexander G. D. G. Matthews, Sébastien Racanière, Ali Razavi, Danilo J. Rezende, Fernando Romero-López, Phiala E. Shanahan, and Julian M. Urban. Aspects of scaling and scalability for flow-based sampling of lattice qcd, 2022.

[2] M S Albergo, Gurtej Kanwar, and Phiala Shanahan. A flow-based sampling algorithm for lattice field theories. *Physical Review D*, 100(3):034515, 2019.

[3] Michele S Albergo, Denis Boyda, Daniel C Hackett, Gurtej Kanwar, Sébastien Racanière, Kyle Cranmer, and Phiala Shanahan. Introduction to normalizing flows for lattice field theory. *arXiv preprint arXiv:2101.08176*, 2021.

[4] Ricardo Baptista, Youssef Marzouk, and Olivier Zahm. On the representation and learning of monotone triangular transport maps. *Foundations of Computational Mathematics*, 24(6):2063–2108, November 2023.

[5] Marc Bauer, Renzo Kapust, Jan M. Pawlowski, and Finn L. Temmen. Super-resolving normalising flows for lattice field theories, 2024.

[6] Denis Boyda, Daniel C Hackett, Gurtej Kanwar, Sebastien Racaniere, Phiala E Shanahan, and M S Albergo. Sampling SU(N) gauge configurations using normalizing flows. *Physical Review D*, 103(7):074504, 2021.

[7] Felicity Brennan, Daniele Bigoni, and Youssef Marzouk. Greedy inference with structure-exploiting lazy-map triangular transport. In *Uncertainty in Artificial Intelligence (UAI)*, pages 1204–1213. PMLR, 2020.

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations (ICLR)*, 2017.

[9] Alan George and Joseph WH Liu. *Computer solution of large sparse positive definite systems*. Prentice-Hall, 1981.

[10] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning (ICML)*, pages 2078–2087. PMLR, 2018.

[11] Gurtej Kanwar, Michael S Albergo, Denis Boyda, Kyle Cranmer, Daniel C Hackett, Sébastien Racanière, Phiala E Shanahan, and Julian M Toth. Equivariant flow-based sampling for lattice gauge theory. *Physical Review Letters*, 125(12):121601, 2020.

[12] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[13] Youssef M Marzouk, Tarek A Moselhy, Matthew D Parno, and Alessio Spantini. An introduction to sampling for bayesian inference. *The first 25 years of the Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, pages 1–44, 2016.

[14] Maximilian Ramgraber, Daniel Sharp, Mathieu Le Provost, and Youssef Marzouk. A friendly introduction to triangular transport, 2025.

[15] Florian Schäfer, Matthias Katzfuss, and Houman Owhadi. Sparse cholesky factorization by kullback-leibler minimization, 2021.

[16] Phiala Shanahan, Michael L Wagman, Denis Boyda, Kyle Cranmer, Daniel C Hackett, and Gurtej Kanwar. Machine learning for lattice gauge theory. *Reviews of Modern Physics*, 95(2):025002, 2023.

[17] Alan D Sokal. Monte carlo methods in statistical mechanics: foundations and new algorithms. *Functional integration*, pages 131–192, 1997.

[18] Alessio Spantini, Daniele Bigoni, and Youssef Marzouk. Inference via low-dimensional couplings, 2018.

# A Supplementary Material

## A.1 Training: KL Divergence Minimization

The parameters of the neural networks modeling $f_j$ and $g_j$ (collectively denoted $\theta$) are trained by minimizing the Kullback-Leibler (KL) divergence between the model distribution $p_\Phi(\phi)$ (induced by $T_\theta$) and the target distribution $P[\phi]$ (Eq. (2)). We minimize $KL(p_\Phi(\phi)||P[\phi])$:

$$KL(p_\Phi||P) = \int p_\Phi(\phi) \log \frac{p_\Phi(\phi)}{P[\phi]} d\phi \tag{10}$$

$$= \mathbb{E}_{\phi\sim p_\Phi}[\log p_\Phi(\phi) - \log P[\phi]] \tag{11}$$

This is the appropriate loss (reverse KL or variational free energy minimization) when the target density is known (via the action $S[\phi]$) but samples are unavailable. Using the change of variables $\phi = T_\theta(z)$ where $z \sim p_Z(z)$ (the base Gaussian distribution), and $p_\Phi(T_\theta(z)) = p_Z(z)|\det J_{T_\theta}(z)|^{-1}$:

$$KL(p_\Phi||P) = \mathbb{E}_{z\sim p_Z(z)}[\log(p_Z(z)|\det J_{T_\theta}(z)|^{-1}) - \log(Z^{-1}e^{-S[T_\theta(z)]})] \tag{12}$$

$$= \mathbb{E}_{z\sim p_Z(z)}[\log p_Z(z) - \log|\det J_{T_\theta}(z)| + S[T_\theta(z)] + \log Z] \tag{13}$$

To minimize this KL divergence, we can drop terms constant with respect to model parameters $\theta$ (namely $\mathbb{E}_{z\sim p_Z(z)}[\log p_Z(z)]$ and $\log Z$). The loss function to minimize is thus:

$$\mathcal{L}(\theta) = \mathbb{E}_{z\sim p_Z(z)}[S[T_\theta(z)] - \log|\det J_{T_\theta}(z)|] \tag{14}$$

which in this setup becomes:

$$\mathcal{L}(\theta) = \mathbb{E}_{z\sim p_Z(z)}\left[S[T_\theta(z)] - \sum_{j=0}^{N_{sites}-1} \log r(g_j(z_j, z_{<j}^{(j)}; \theta))\right] \tag{15}$$

where $z_{<j}^{(j)}$ denotes the appropriate conditioning set for the $j^{th}$ component (all $z_{<j}$ for dense, or $z_i$ for $i \in N_p(j)$ for sparse). The expectation is approximated by Monte Carlo sampling from $p_Z(z)$ and using mini-batch stochastic gradient descent.

## A.2 Metropolis-Hastings Correction Step

While the trained normalizing flow $p_\Phi(\phi)$ approximates $P[\phi]$, it may not be exact. To obtain samples from the exact target distribution $P[\phi]$, an MCMC correction step is applied. Similiar as in [2]. We use the Independent Metropolis-Hastings (IMH) algorithm, where the proposal distribution is the learned map itself, $Q(\phi') = p_\Phi(\phi')$. Given a current sample $\phi_c$, a new sample $\phi_p$ is proposed by drawing $z_p \sim p_Z(z)$ and setting $\phi_p = T_\theta(z_p)$. The acceptance probability is:

$$\alpha(\phi_p|\phi_c) = \min\left(1, \frac{P[\phi_p]Q(\phi_c)}{P[\phi_c]Q(\phi_p)}\right) = \min\left(1, \frac{P[\phi_p]p_\Phi(\phi_c)}{P[\phi_c]p_\Phi(\phi_p)}\right) \tag{16}$$

This can be rewritten using importance weights $w(\phi) = P[\phi]/p_\Phi(\phi)$: $\alpha(\phi_p|\phi_c) = \min\left(1, \frac{w(\phi_p)}{w(\phi_c)}\right)$. The log-importance weight for a sample $\phi = T_\theta(z)$ is (ignoring the constant $\log Z$):

$$\log w'(\phi) = \log(e^{-S[\phi]}) - \log p_\Phi(\phi) \tag{17}$$

$$= -S[T_\theta(z)] - (\log p_Z(z) - \log|\det J_{T_\theta}(z)|) \tag{18}$$

The ratio $w(\phi_p)/w(\phi_c)$ becomes $w'(\phi_p)/w'(\phi_c)$, and the acceptance probability calculation proceeds using these relative weights. The MCMC step ensures that the resulting chain of accepted samples converges to the exact target distribution $P[\phi]$.

## A.3 Effective Sample Size (ESS) Definition

The quality of the approximation $p_\Phi$ can be measured by the Effective Sample Size (ESS) of the samples generated directly from the flow, using importance weights $w(\phi) = P[\phi]/p_\Phi(\phi)$. For $M$ samples $\{\phi_i\}_{i=1}^M$:

$$\text{ESS} = \frac{(\sum_{i=1}^M w(\phi_i))^2}{\sum_{i=1}^M w(\phi_i)^2}/M \tag{19}$$

An ESS close to 1 indicates that $p_\Phi \approx P$.

### A.4 Markov Property of $\phi^4$ theory

The Markov property emerges from the locality of the action. For the $\phi^4$ theory, the full conditional is:

$$P\left(\phi_x \mid \phi_{\Lambda\setminus\{x\}}\right) = \frac{P[\phi]}{P\left[\phi_{\Lambda\setminus\{x\}}\right]} \propto \exp\left(-\sum_{\mu}\frac{1}{2}\left(\phi_{x+\hat{\mu}}-\phi_x\right)^2 - \frac{m^2}{2}\phi_x^2 - \frac{\lambda}{4!}\phi_x^4\right)$$

This depends only on $\phi_{x\pm\hat{\mu}}$, confirming $\phi_x \perp \phi_{\Lambda\setminus(\{x\}\cup\mathcal{N}(x))} \mid \phi_{\mathcal{N}(x)}$.

### A.5 Triangular Map Structures and Fill-in Scaling

The ordering of variables is a critical choice in constructing triangular maps, as it directly dictates the structure of both the exact and approximate dependency graphs. The exact dependency structure, required for a perfect transformation, accounts for all correlations induced by marginalizing "future" variables. This process, known as fill-in, typically results in a dense graph. In contrast, an approximate map achieves computational efficiency by enforcing a sparse structure based only on local, "past" physical neighbors. In Figure 5 and Figure 6 , we provide a detailed visualization of these effects and analyze the scaling behavior of the fill-in phenomenon for different orderings.

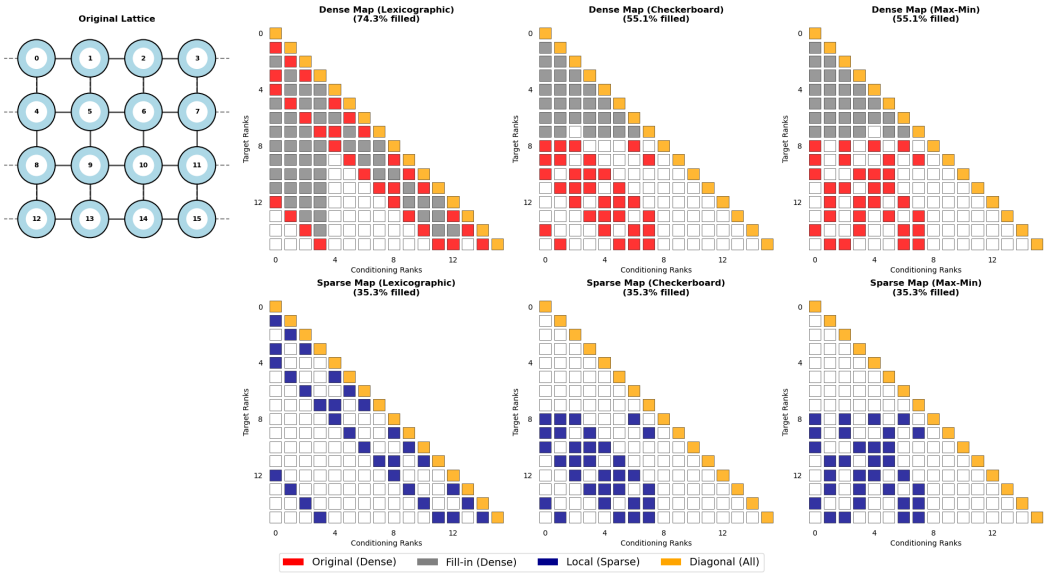



Figure 5: Comparison of exact (top row) and enforced sparse (bottom row) dependency structures for a triangular map on a $4\times4$ lattice under different orderings. The exact maps reveal the non-local fill-in patterns unique to each ordering, with lexicographic ordering creating a distinctly different dense structure from the more symmetric checkerboard ordering. The sparse maps are, by construction, limited to preceding physical neighbors ($N_p(j)$), highlighting the significant reduction in complexity at the cost of approximation.

## B Validation of Physical Observables

To validate the exactness of the sampling procedure (flow + IMH), we will calculate key physical observables and compare them against HMC results.

- **Average Magnetization:** $\langle M\rangle = \langle|\frac{1}{N_{\text{sites}}}\sum_x \phi_x|\rangle$.

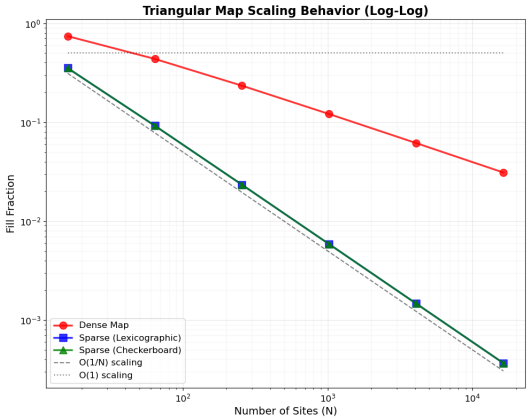



Figure 6: Scaling of the fill-in fraction as a function of lattice size $L$ for a 2D system. The plot shows the ratio of non-zero entries in the exact dependency map (excluding the main diagonal and physical neighbors) to the total possible entries. This analysis quantifies how rapidly the map densifies, demonstrating the computational challenge of using exact maps for larger systems and motivating the use of sparse approximations.

- **Magnetic Susceptibility:** $\chi_2 = N_{\text{sites}}(\langle M^2 \rangle - \langle M \rangle^2)$.