Neural Field Transformations for Hybrid Monte Carlo: Architectural Design and Scaling

Jinchen He
Maryland Center for Fundamental Physics
University of Maryland, College Park
College Park, MD 20742
jinchen@umd.edu

Xiao-Yong Jin
Computational Science Division
Argonne National Laboratory
Lemont, IL 60439
xjin@anl.gov

James C. Osborn
Computational Science Division
Argonne National Laboratory
Lemont, IL 60439
osborn@anl.gov

Yong Zhao
Physics Division
Argonne National Laboratory
Lemont, IL 60439
yong.zhao@anl.gov

Abstract

Critical slowing down, where autocorrelation grows rapidly near the continuum limit due to Hybrid Monte Carlo (HMC) moving through configuration space inefficiently, still challenges lattice gauge theory simulations. Combining neural field transformations with HMC (NTHMC) can reshape the energy landscape and accelerate sampling, but the choice of neural architectures has yet to be studied systematically. We evaluate NTHMC on a two-dimensional U(1) gauge theory, analyzing how it scales and transfers to larger volumes and smaller lattice spacing. Controlled comparisons let us isolate architectural contributions to sampling efficiency. Good designs can reduce autocorrelation and boost topological tunneling while maintaining favorable scaling. More broadly, our study highlights emerging design guides, such as wider receptive fields and channel-dependent activations, paving the way for systematic extensions to four-dimensional SU(3) gauge theory.

1 Introduction

Lattice gauge theory (LGT) simulates gauge fields non-perturbatively by discretizing spacetime on a periodic hypercubic grid. Its flagship application, lattice quantum chromodynamics (QCD) has become the leading non-perturbative tool for studying the strong nuclear interaction. State-of-the-art calculations use gauge configurations sampled from the Euclidean partition function with Hybrid Monte Carlo (HMC) [1]. Approaching the continuum limit, as the lattice volume grows and lattice spacing shrinks, HMC suffers from *critical slowing down* [2–4]—the autocorrelation between gauge configurations in the generated Markov chain increases sharply.

Normalizing flows and related generative models [5, 6] can generate decorrelated samples but struggle with the high dimensionality of realistic lattices [7]. Lüscher's trivializing map [8] proposes a construction to transform the physical gauge field to a trivialized field, performing a change of variables on the path integral. Practical implementations use invertible smearing steps such as Wilson flow or stout smearing [9]. A framework known as neural network transformed HMC (NTHMC) [10, 11] constructs gauge-covariant field transformations using machine learning.

This work presents a systematic study of NTHMC, focusing on how neural architectural designs such as residual connections, attention mechanisms, channel-dependent activation, and batch size

affect performance and scaling. Testing on a two-dimensional U(1) gauge theory, we evaluate both small-lattice performance and the transferability to larger volumes and smaller lattice spacing. Our results suggest that wider receptive fields and channel-dependent activation improve efficiency while maintaining favorable scaling, thereby informing extensions to four-dimensional SU(3) gauge fields and large-scale lattice QCD.

The NTHMC framework uses an invertible, gauge-covariant field transformation $\mathcal{F} : \tilde{U} \mapsto U$ parameterized by a neural network to perform a change of variables in the integration over the Boltzmann weight $\exp(-S(U))$, leading to an effective action,

$$S_{\mathrm{FT}}(\tilde{U}) = S\big(\mathcal{F}(\tilde{U})\big) - \log\big|\det \mathcal{F}_*(\tilde{U})\big|, \tag{1}$$

where $\mathcal{F}_*(\tilde{U})$ is the Jacobian matrix of the transformation. HMC is then performed in the space of auxiliary variables $\tilde{U}$ with the Boltzmann weight $\exp(-S_{\mathrm{FT}}(\tilde{U}))$.

Intuitively, the field transformation acts like a learned preconditioner: it reshapes the energy landscape so that HMC trajectories encounter smoother gradients. We construct the field transformation from gauge-covariant, locally equivariant convolutional neural networks (CNNs). For each link variable $\tilde{U}_{x,\mu}$ (a gauge field element going from site x to $x + \hat{\mu}$), the update takes the form

$$\mathcal{F} : \tilde{U}_{x,\mu} \mapsto U_{x,\mu} = e^{\Pi_{x,\mu}} \tilde{U}_{x,\mu}, \quad \Pi_{x,\mu} = \sum_l \epsilon_{x,\mu,l}\, W_{x,\mu,l}\,. \tag{2}$$

The coefficients are given by $\epsilon_{x,\mu,l} = \phi[N_l(X, Y, \dots)]$, with N_l a CNN mapping local gauge-invariant inputs $(X, Y, \dots)$ to the coefficients for the loop functions $W_{x,\mu,l}$, and ϕ a bounded activation function (e.g. $\mathtt{tanh}$ or $\mathtt{arctan}$) that enforces the invertibility of the transformation. $W_{x,\mu,l}$ is the Lie algebra projection of a specific (labeled by l) Wilson loop (path ordered products of gauge fields along a closed loop) that starts and ends at site x and first moves in the μ direction (the first on the right of the product). For the $U(1)$ gauge theory, the projection just takes the imaginary part.
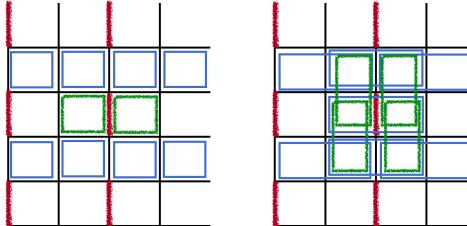

Figure 1: Cropped section of a lattice configuration illustrating the field transformation. Red lines mark one of the eight link subsets updated sequentially, green lines indicate Wilson loops $W_{x,\mu,l}$, and blue lines show the gauge-invariant features $(X, Y, \dots)$ used as CNN inputs.

As suggested in Ref. [8], the lattice links are partitioned into eight disjoint subsets and updated sequentially. This ensures that the Jacobian determinant of each update step is reduced to the product of diagonal elements. Moreover, by choosing CNN inputs $(X, Y, \dots)$ to be independent of the links in the active subset, these diagonal elements can be written in a closed form, making the Jacobian determinant explicitly tractable. In this work, we take the l index to run over 1×1 plaquettes and 1×2 rectangles, as illustrated in Fig.1, where one link subset (red) is updated, while the corresponding loops (green) and local gauge-invariant CNN inputs (blue) are highlighted.

For U(1) gauge theory, we use $U = e^{i\theta}$, and the Wilson plaquette action [12] as $S(U)$. The transformation reduces to exponentials of sine terms of plaquette and rectangle loops,

$$\mathcal{F} : \tilde{U}_{x,\mu} \mapsto U_{x,\mu} = e^{\sum_l \epsilon_{x,\mu,l} W_{x,\mu,l}} \tilde{U}_{x,\mu} = e^{\sum_p i\epsilon_{x,\mu,p}\sin(\theta_p) + \sum_r i\epsilon_{x,\mu,r}\sin(\theta_r)} \tilde{U}_{x,\mu}, \tag{3}$$

with p and r denoting individual plaquette and rectangle loops, respectively, and θ_p, θ_r the phase angle of the corresponding Wilson loop. The Jacobian has the matrix elements

$$\mathcal{F}_*(\tilde{U})_{x,\mu,x',\mu'} = \delta_{x,x'}\delta_{\mu,\mu'} \left[1 + \sum_p \epsilon_{x,\mu,p}\cos(\theta_p) + \sum_r \epsilon_{x,\mu,r}\cos(\theta_r) \right]. \tag{4}$$

2

In order to have an invertible transformation, the Jacobian determinant may not change sign. The invertibility condition is then

$$\sum_p |\epsilon_{x,\mu,p}| + \sum_r |\epsilon_{x,\mu,r}| < 1 \,. \tag{5}$$

The field transformation is trained on configurations generated from regular HMC by minimizing the transformed action force:

$$\mathcal{L}(\tilde{U}) = \sum_{p \in \{2,4,6,8\}} \frac{c_p}{V^{1/p}} \left\| \frac{\partial S_{\mathrm{FT}}(\tilde{U})}{\partial \tilde{U}} \right\|_p, \tag{6}$$

where V is the lattice volume and c_p are hyperparameters that weight different force norms, thereby balancing the emphasis between average and peak forces. In our experiments, we set $c_2 = c_4 = c_6 = c_8 = 1$. Different from Ref. [10], which matched the transformed force to that at a larger lattice spacing, we directly minimize the transformed force itself, effectively training the transformation to smooth the action landscape, reduce the stiff directions and improve the sampling efficiency.

2   Experiments

Inside the HMC update we employ the 2nd order minimum norm symplectic integrator of Omelyan et al. [13] with 10 MD steps per trajectory, tuning the step size to achieve $\sim 80\%$ acceptance. All HMC simulations were verified by checking the plaquette values of the generated configurations, which were found to agree with the theoretical expectations within statistical uncertainties. We benchmark five architectures: a baseline (`Base`) model similar to the design in Ref. [10], and four variants (`Tanh`, `Resn`, `Attn`, `Comb`) introducing different modifications.

Base The baseline model is a two-layer CNN that processes six input channels (two plaquette, four rectangle). A first 3×3 convolution maps them to 12 hidden channels with GELU activation, followed by a second 3×3 convolution producing coefficients, which are scaled by $\arctan$ to $(-1/6, 1/6)$, satisfying the invertibility condition in Eq. 5. This design yields an effective 5×5 receptive field with $\sim 2{,}000$ parameters.

Tanh Same as `Base`, but replaces the $\arctan$ scaling with a channel-dependent $\tanh$ activation, ensuring $|\epsilon_{x,\mu,p}| < 0.4$ and $|\epsilon_{x,\mu,r}| < 0.05$, with $\sim 2{,}000$ parameters.

Resn Adds two residual blocks (each has two 3×3 convolutions with skip connections) before the output, and replaces the second 3×3 convolution of `Base` with a 1×1 projection. This enlarges the receptive field to 11×11 and increases the parameters to $\sim 6{,}000$.

Attn Augments `Base` with a channel-attention module between the two convolutions, reweighting channels without changing the 5×5 receptive field. The parameter count is $\sim 2{,}100$.

Comb Combines residual blocks, channel attention, and channel-dependent $\tanh$ activation ($|\epsilon_{x,\mu,p}| < 0.4$, $|\epsilon_{x,\mu,r}| < 0.05$). The resulting architecture has an 11×11 receptive field and $\sim 6{,}100$ parameters.

All models are trained with the p-norm force loss (Eq.6) on 2^{12} independent configurations at $\beta = 3$, $V = 32^2$ using an 80/20 train–test split. The input gauge coupling, β, enters in the action $S(U)$, and in this theory, the lattice spacing is proportional to $\beta^{-1/2}$. The default batch size is 64, except that `Base` and `Comb` are also trained with 32 (denoted `Base32`, `Comb32`) to study batch-size effects. Each model is trained for 32 epochs in PyTorch [14], with one single run taking about 12–20 minutes on a single NVIDIA A100 GPU for batch size 64, and roughly twice as long for batch size 32.

To assess the sampling efficiency of regular HMC and NTHMC, we consider two complementary metrics. The first is $\gamma(\delta)$, which measures the relative independence of the configuration [10]:

$$\gamma(\delta) = \frac{1}{1 - \Gamma_t(\delta)} \,, \quad \Gamma_t(\delta) \equiv \frac{\langle Q_\tau Q_{\tau+\delta} \rangle}{\langle Q^2 \rangle} = 1 - \frac{\langle (Q_\tau - Q_{\tau+\delta})^2 \rangle}{2 V \chi_t^\infty(\beta)} \,. \tag{7}$$

Here $\Gamma_t(\delta)$ is the autocorrelation of the topological charge Q at separation δ in HMC steps, and $\chi_t^\infty(\beta)$ is the topological susceptibility [10]. The larger $\gamma(\delta)$ indicates stronger autocorrelations and therefore less efficient sampling.

The second metric is the average topological change ΔQ, the mean absolute shift in topological charge per HMC step, which directly probes the tunneling between sectors; larger values of ΔQ indicate more frequent topological transitions and thus more efficient sampling. All training and evaluation experiments are repeated with 8 random seeds. Statistical uncertainties are estimated and consistently propagated through derived quantities using Jackknife resampling [15] across the seed variability.

3   Results
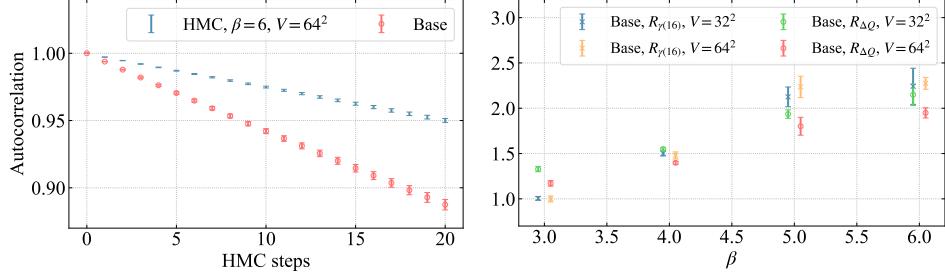
Figure 2: Comparison of regular HMC and NTHMC with the `Base` model. Left: autocorrelation decay at $\beta = 6.0$, $V = 64^2$. Right: scaling of two efficiency ratios across β and volume, with $R_{\gamma(16)} \equiv \gamma(16)_{\mathrm{HMC}}/\gamma(16)_{\mathrm{Base}}$ and $R_{\Delta Q} \equiv \Delta Q_{\mathrm{Base}}/\Delta Q_{\mathrm{HMC}}$.

As shown in the left panel of Fig. 2, NTHMC with the `Base` model achieves a markedly faster decay of autocorrelations than regular HMC. The right panel reports the efficiency ratios $R_{\gamma(16)} = \gamma(16)_{\mathrm{HMC}}/\gamma(16)_{\mathrm{Base}}$ and $R_{\Delta Q} = \Delta Q_{\mathrm{Base}}/\Delta Q_{\mathrm{HMC}}$. Despite being trained only at $\beta = 3$, $V = 32^2$, NTHMC yields even larger relative gains at higher β (smaller lattice spacing) and larger volumes, demonstrating favorable scaling and generalization. At $\beta = 3$, where regular HMC is already efficient, both ratios remain close to unity.
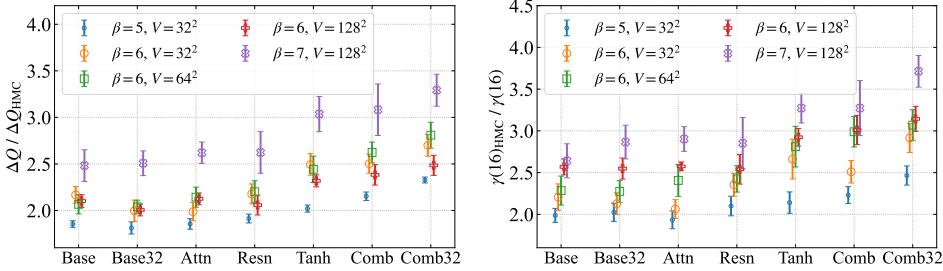
Figure 3: Comparison of NTHMC with various network architectures across different lattice couplings and volumes. The plots show ΔQ (left panel) and $\gamma(16)$ (right panel), both normalized to standard HMC results.

Fig. 3 summarizes the performance of different CNN architectures within NTHMC across several lattice couplings and volumes. The comparison includes results at $\beta = 5$, 6, and 7 with lattice sizes $V = 32^2$, 64^2, and 128^2, allowing us to examine both coupling and volume dependence relative to standard HMC. Residual blocks and channel-dependent activation lead to improvements over the `Base` model, and the combined design (`Comb`) yields the strongest effect, achieving nearly a 40% reduction in $\gamma(16)$ with a corresponding increase in the average topological change ΔQ. Moreover, as the coupling β increases, all NTHMC models exhibit a higher sampling efficiency compared to standard HMC, and the efficiency does not show statistically significant changes with increasing lattice volume, indicating that our method can be effectively extended to larger volumes and finer lattices.

Interestingly, the `Tanh` model achieves competitive gains despite having the same parameter count as `Base`, which we attribute to the faster saturation of $\tanh$ compared to $\arctan$ and, more importantly,

4

to the larger magnitude of plaquette coefficients $\epsilon_{x,\mu,p}$ enabled by channel-dependent activation. Likewise, residual connections expand the receptive field, which might be expected to hinder transferability, yet we find that an appropriate enlargement does not compromise scaling or generalization and may even be beneficial. A possible explanation is that, as the continuum limit is approached in LGT, infrared (long distance) structures are preserved while ultraviolet (short distance) details become increasingly refined, so the transferable features are precisely the long-range correlations that benefit from a relatively wider receptive field.

Finally, batch size interacts mildly with architecture. Although the baseline shows little sensitivity, the richer `Comb` model benefits from smaller batches, consistent with gradient noise acting as an implicit regularizer that helps larger capacity models generalize more effectively.

4   Prospects

Our preliminary results demonstrate that neural field transformations can substantially reduce autocorrelations, yet several open directions remain. Future work includes exploring integrator choices such as varying the number of MD steps at fixed trajectory length, refining training objectives beyond minimizing the force norms, and systematically analyzing training regimes (e.g. number of epochs, learning rate, training set size and data from different β and volumes). Finally, extending from U(1) to SU(3) will be essential to connect this approach with large-scale lattice QCD applications.

Data Availability

The code used in this work are available at https://github.com/Greyyy-HJC/Scaling-FT-HMC.

References

[1] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[2] Ulli Wolff. Critical slowing down. *Nucl. Phys. B-Proc. Sup*, 17:93–102, 1989.

[3] B Alles, G Boyd, Massimo D'Elia, Adriano Di Giacomo, and Ettore Vicari. Hybrid monte carlo and topological modes of full qcd. *Physics Letters B*, 389(1):107–111, 1996.

[4] Luigi Del Debbio, Gian Mario Manca, and Ettore Vicari. Critical slowing down of topological modes. *Physics Letters B*, 594(3-4):315–323, 2004.

[5] Michael S Albergo, Denis Boyda, Daniel C Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racaniere, Danilo Jimenez Rezende, and Phiala E Shanahan. Introduction to normalizing flows for lattice field theory. *arXiv preprint arXiv:2101.08176*, 2021.

[6] Ryan Abbott, Michael S Albergo, Aleksandar Botev, Denis Boyda, Kyle Cranmer, Daniel C Hackett, Gurtej Kanwar, Alexander GDG Matthews, Sébastien Racanière, Ali Razavi, et al. Normalizing flows for lattice gauge theory in arbitrary space-time dimension. *arXiv preprint arXiv:2305.02402*, 2023.

[7] Andrea Bulgarelli, Elia Cellini, and Alessandro Nada. Scaling of stochastic normalizing flows in su (3) lattice gauge theory. *Physical Review D*, 111(7):074517, 2025.

[8] Martin Lüscher. Trivializing maps, the wilson flow and the hmc algorithm. *Communications in mathematical physics*, 293(3):899–919, 2010.

[9] Shuhei Yamamoto, Peter Boyle, Taku Izubuchi, Luchang Jin, Christoph Lehner, and Nobuyuki Matsumoto. Improvement in autocorrelation times measured by the master-field technique using field transformation hmc in 2+ 1 domain wall fermion simulations. *arXiv preprint arXiv:2502.05452*, 2025.

[10] Xiao-yong Jin. Neural network field transformation and its application in hmc. *arXiv preprint arXiv:2201.01862*, 2022.

[11] Xiao-Yong Jin. Neural network gauge field transformation for 4d su (3) gauge fields. *arXiv preprint arXiv:2405.19692*, 2024.

[12] Kenneth G Wilson. Confinement of quarks. *Physical review D*, 10(8):2445, 1974.

[13] IP Omelyan, IM Mryglod, and R Folk. Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations. *Computer Physics Communications*, 151(3):272–314, 2003.

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[15] Rupert G Miller. The jackknife-a review. *Biometrika*, 61(1):1–15, 1974.