PROCEEDINGS OF SCIENCE

LeapfrogLayers: A Trainable Framework for Effective Topological Sampling

Sam Foreman,[a,*] **Xiao-Yong Jin**[a] **and James C. Osborn**[a]

[a] *Argonne National Laboratory,*
Lemont, IL

E-mail: foremans@anl.gov, xjin@anl.gov, osborn@alcf.anl.gov

We introduce LeapfrogLayers, an invertible neural network architecture that can be trained to efficiently sample the topology of a 2D $U(1)$ lattice gauge theory. We show an improvement in the integrated autocorrelation time of the topological charge when compared with traditional HMC, and propose methods for scaling our model to larger lattice volumes.

The 38th International Symposium on Lattice Field Theory
26-30 July 2021
Zoom / Gather @ MIT, Cambridge MA, USA

*Speaker

1. Introduction

One of the major goals of lattice field theory calculations is to evaluate integrals of the form

$$\langle \mathcal{O} \rangle \propto \int [\mathcal{D}x]\, \mathcal{O}(x) p(x), \tag{1}$$

for some target distribution $p(x) \propto e^{-S(x)}$. We can approximate the integral using Markov Chain Monte Carlo (MCMC) sampling techniques. This is done by sequentially generating a chain of configurations $\{x_1, x_2, \dots x_N\}$, with $x_i \sim p(x)$ and averaging the value of the function $\mathcal{O}(x)$ over the chain. Accounting for correlations between states in our chain, the sampling variance of this esimator is given by

$$\sigma^2 = \frac{\tau_{\mathcal{O}}^{\text{int}}}{N} \sum_{n=1}^{N} \text{Var}\left[\mathcal{O}(x)\right] \tag{2}$$

where $\tau_{\text{int}}^{\mathcal{O}}$ is the integrated autocorrelation time. This quantity can be interpreted as the additional time required for these induced correlations to become negligible.

1.1 Charge Freezing

The ability to efficiently generate independent configurations is currently a major bottleneck for lattice simulations. In this work we consider a $U(1)$ gauge model on a 2D lattice with periodic boundary conditions. The theory is defined in terms of the link variables $U_\mu(x) = e^{i x_\mu(n)} \in U(1)$ with $x_\mu(n) \in [-\pi, \pi]$. Our target distribution is given by $p(x) \propto e^{-S_\beta(x)}$, where $S_\beta(x)$ is the Wilson action

$$\begin{aligned} S_\beta(x) &= \beta \sum_P 1 - \cos x_P, \text{ and} \\ x_P &\equiv \big[x_\mu(n) + x_\nu(n+\hat{\mu}) \\ &\qquad - x_\mu(n+\hat{\nu}) - x_\nu(n)\big], \end{aligned} \tag{3}$$

is the sum of the gauge variables around the elementary plaquette. Each lattice configuration has a distinct topological charge $Q \in \mathbb{Z}$ given by

$$\begin{aligned} Q_{\mathbb{Z}} &= \frac{1}{2\pi} \sum_P \lfloor x_P \rfloor, \text{ where} \\ \lfloor x_P \rfloor &\equiv x_P - 2\pi \left\lfloor \frac{x_P + \pi}{2\pi} \right\rfloor. \end{aligned}$$

As $\beta \to \infty$, the $Q_{\mathbb{Z}} = 0$ mode becomes dominant and we see that the value of $Q_{\mathbb{Z}}$ remains fixed for large durations of the simulation.

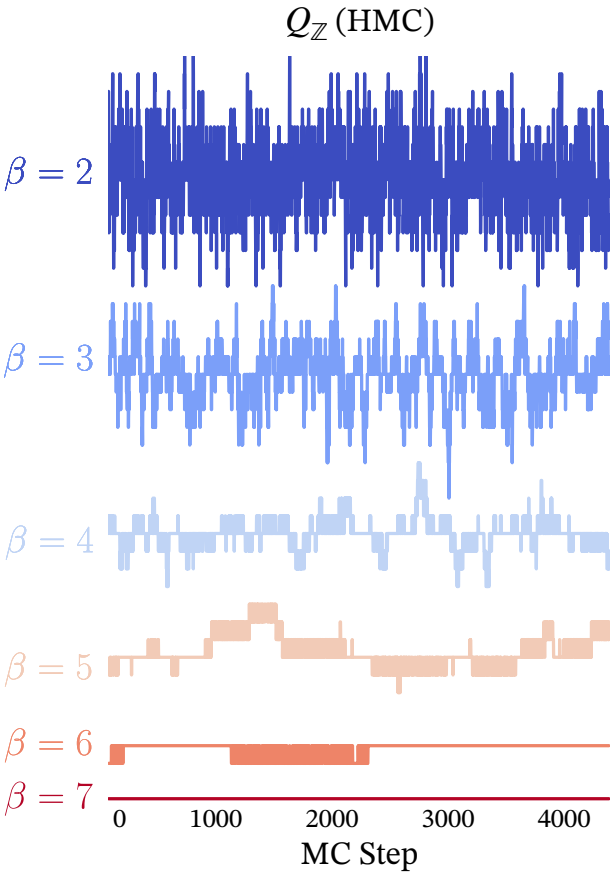

Figure 1: Illustration of the topological charge Q freezing as $\beta : 2 \to 7$ for traditional HMC.

This can be seen by introducing the *tunneling rate*[1]

$$\delta Q \equiv |Q_{i+1} - Q_i| \in \mathbb{Z}. \tag{4}$$

This quantity serves as a measure for how efficiently our chain is able to jump (tunnel) between sectors of distinct topological charge. From Fig 1, we can see that $\delta Q \to 0$ as $\beta \to \infty$.

2. Hamiltonian Monte Carlo (HMC)

We first briefly review the ideas of the Hamiltonian Monte Carlo (HMC) algorithm and highlight some of the issues faced with this approach. We begin by introducing a fictitious momentum $v \sim \mathcal{N}(0, \mathbb{1})$ distributed independently of x. This allows us to write the joint target density of the $\xi \equiv (x, v)$ system as

$$p(x, v) = p(x) \cdot p(v) = e^{-S_\beta(x)} \cdot e^{-v^T v/2} = e^{-H(x,v)} \tag{5}$$

where $H(\xi) = H(x, v) = S_\beta(x) + \frac{1}{2} v^T v$ is the Hamiltonian of the system.

2.1 Leapfrog Integrator

For a given initial state ξ_0, we can use Hamilton's equations to evolve the systems dynamics

$$\dot{x} = \frac{\partial H}{\partial v} \quad , \quad \dot{v} = -\frac{\partial H}{\partial x} \tag{6}$$

along isoprobability contours of $H = \text{const.}$ from $\xi_0 = (x_0, v_0) \to (x^*, v^*)$. We can numerically integrate this system of equations using the *leapfrog integrator*, which is composed of the following three steps:

1. Starting from x_0, resample the momentum $v_0 \sim \mathcal{N}(0, \mathbb{1})$ and construct the state $\xi_0 = (x_0, v_0)$.
2. Generate a *proposal configuration* ξ^* by integrating $\dot{\xi}$ along $H = \text{const.}$ for N leapfrog steps. i.e.
$$\xi_0 \to \xi_1 \to \ldots \to \xi_N \equiv \xi^*, \tag{7}$$
where a single leapfrog step $\xi_i \to \xi_{i+1}$ above consists of:
$$\textbf{(a.)}\ \tilde{v} \leftarrow v - \frac{\varepsilon}{2}\partial_x S(x), \qquad \textbf{(b.)}\ x' \leftarrow x + \varepsilon\tilde{v}, \qquad \textbf{(c.)}\ v' \leftarrow \tilde{v} - \frac{\varepsilon}{2}\partial_x S(x). \tag{8}$$
3. At the end of the trajectory, accept or reject the proposal configuration ξ^* using the Metropolis-Hastings (MH) test.
$$x_{i+1} \leftarrow \begin{cases} x^* \text{ with probability } A(\xi^*|\xi) \\ x_i \text{ with probability } 1 - A(\xi^*|\xi), \end{cases} \tag{9}$$
where
$$A(\xi^*|\xi) \equiv \min\left\{1, \frac{p(\xi^*)}{p(\xi)}\left|\frac{\partial \xi^*}{\partial \xi^T}\right|\right\}. \tag{10}$$

An illustration of this procedure can be seen in Fig 2.

[1] As measured between subsequent states in our chain $i, i+1$.

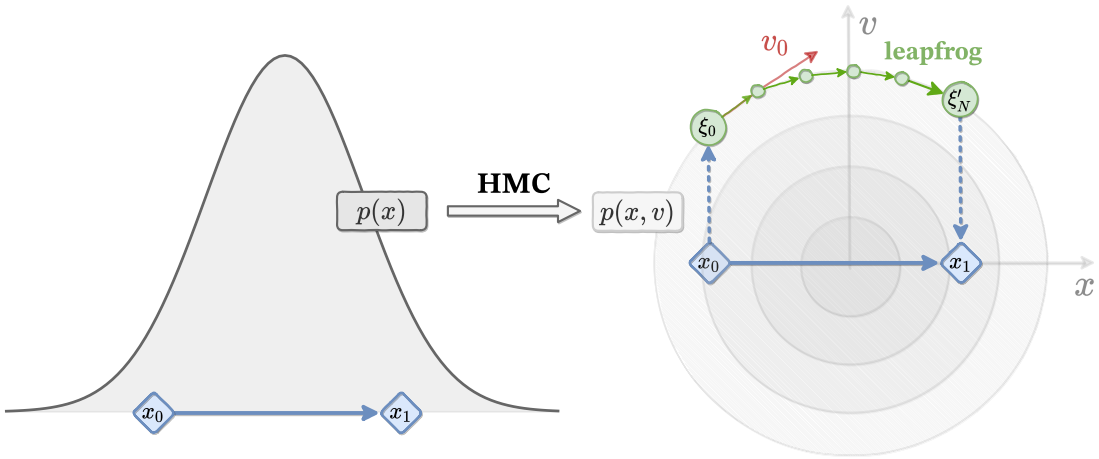

Figure 2: High-level overview of the HMC algorithm.

2.2 Issues with HMC

Re-sampling the momentum at the start of each trajectory causes the energy levels we explore to be randomly selected. This is reminiscient of the "random-walk" behavior of traditional MCMC and leads to a slow exploration of our target distribution (i.e. long autocorrelations). Additionally, the HMC sampler is known to have difficulty traversing low-density zones, resulting in poor performance for distributions which have multiple isolated modes. This is particularly relevant in the case of sampling topological quantities in lattice gauge models. This list of criteria helps to illuminate the qualities we would like our (ideal) sampler to exhibit; namely, we would like for our sampler to:

- quickly converge to the true target distribution
- mix quickly once converged (short autocorrelations)
- be able to mix across energy levels and isolated modes

3. Generalizing HMC: LeapfrogLayers

In order to preserve the asymptotic behavior of HMC, our update must

1. Be invertible / preserve reversibility, i.e. $p(a \to b) = p(b \to a)$
2. Have a tractable Jacobian determinant.

To simplify notation, we introduce two functions, Γ (Λ) to denote the v (x) updates. As in HMC, we follow the general pattern of performing alternating updates of v and x. In order to satisfy these requirements, we split the x-update into two parts, sequentially updating complementary subsets using a binary mask m and its complement $\bar{m}$. Additionally, we introduce $d \sim \mathcal{U}(+, -)$, distributed independently of both x and v, to determine the "direction" of our update. Here, we associate $+$ ($-$) with the forward (backward) direction and note that running sequential updates in opposite

directions has the effect of inverting the update. We denote the complete state by $\xi = (x, v, \pm)$, with target density given by $p(\xi) = p(x) \cdot p(v) \cdot p(\pm)$.

Explicitly, we can write this series of updates as

$$(\mathbf{1}.)\ \Gamma^{\pm} : (v; \zeta_v) \to v' \tag{11}$$

$$(\mathbf{2}.)\ \Lambda^{\pm} : (m \odot x; \zeta_x) \to x' \tag{12}$$

$$(\mathbf{3}.)\ \Lambda^{\pm} : (\bar{m} \odot x'; \zeta_{\bar{x}'}) \to x'' \tag{13}$$

$$(\mathbf{4}.)\ \Gamma^{\pm} : (v', \zeta_{v'}) \to v'' \tag{14}$$

where ζ_k is independent of k, and the subscript is used to denote the variable being updated ($k = x, v$). Each of these functions are parameterized by a set of weights θ in a neural network. Again, we proceed similar to the generic HMC leapfrog update from Sec 2.1, and include the high-level steps below:

1. Resample $v \sim \mathcal{N}(0, \mathbb{1})$, $d \sim \mathcal{U}(+, -)$, and construct initial state $\xi_0 = (x_0, v_0, \pm)$
2. Generate the proposal configuration ξ^* by passing the initial state sequentially through N *leapfrog layers*[2] to construct a trajectory.

$$\xi_0 \to \xi_1 \to \ldots \to \xi_N = \xi^* \tag{15}$$

3. Metropolis-Hastings accept/reject to determine next state in chain.

References

[1] ....

[2]We defer the details to Sec 3.