PROCEEDINGS OF SCIENCE

MLMC: Machine Learning Monte Carlo for Lattice Gauge Theory

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

[a] *Leadership Computing Facility, Argonne National Laboratory, 9700 S. Cass Ave, Lemont IL, USA*

[b] *Computational Science Division, Argonne National Laboratory, 9700 S. Cass Ave, Lemont IL, USA*

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

We present a trainable framework for efficiently generating gauge configurations, and discss ongoing work in this direction. In particular, we consider the problem of sampling configurations from a 4D $SU(3)$ lattice gauge theory, and consider a generalized leapfrog integrator in the molecular dynamics update that can be trained to improve sampling efficiency.

The 40th International Symposium on Lattice Field Theory (Lattice 2023)
July 31st - August 4th, 2023
Fermi National Accelerator Laboratory

*Speaker

1. Introduction

2. Background

We would like to calculate observables O:

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

If these were independent, we could approximate the integral as $\langle O \rangle \simeq \frac{1}{N}\sum_{n=1}^{N} O(x_n)$ with variance

$$\sigma_O^2 = \frac{1}{N}\,\mathrm{Var}\left[O(x)\right] \Longrightarrow \sigma_O \propto \frac{1}{\sqrt{N}}. \tag{2}$$

Instead, nearby configurations are correlated, causing us to incur a factor of τ_{int}^{O} in the variance expression

$$\sigma_O^2 = \frac{\tau_{\text{int}}^{O}}{N}\mathrm{Var}\left[O(x)\right] \tag{3}$$

2.1 Hamiltonian Monte Carlo (HMC)

The typical approach [**?**] is to use Hamiltonian Monte Carlo (HMC) algorithm for generating configurations distributed according to our target distribution distributed according to our target distribution. This typically to help reduce these auto-correlations. Specifically, we want to (sequentially) construct a chain of states:

$$x_0 \to x_1 \to x_i \to \cdots \to x_N \tag{4}$$

such that, as $N \to \infty$:

$$\{x_i, x_{i+1}, x_{i+2}, \ldots, x_N\} \xrightarrow{N\to\infty} \pi(x) \tag{5}$$

To do this, we begin by introducing a fictitious momentum[1] $v \sim \mathcal{N}(0,1)$ normally distributed, independent of x. We can write the joint distribution $\pi(x,v)$ as

$$\pi(x,v) = \pi(x)\pi(v) \propto e^{-S(x)}e^{-\frac{1}{2}v^T v} \tag{6}$$

$$= e^{-\left[S(x)+\frac{1}{2}v^T v\right]} \tag{7}$$

$$= e^{-H(x,v)} \tag{8}$$

We can evolve the Hamiltonian dynamics of the $(\dot{x},\dot{v}) = (\partial_v H, -\partial_x H)$ system using operators $\Gamma : v \to v'$ and $\Lambda : x \to x'$. Explicitly, for a single update step of the leapfrog integrator:

$$\tilde{v} := \Gamma(x,v) = v - \frac{\varepsilon}{2}F(x) \tag{9}$$

$$x' := \Lambda(x,\tilde{v}) = x + \varepsilon\tilde{v} \tag{10}$$

$$v' := \Lambda(x',\tilde{v}) = \tilde{v} - \frac{\varepsilon}{2}F(x'), \tag{11}$$

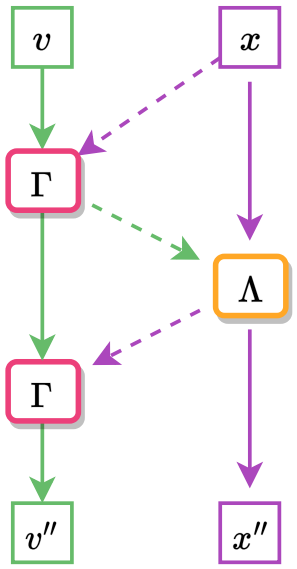

Figure 1: Illustration of the leapfrog update for HMC.

[1] Here $\sim$ means *is distributed according to*.

where we've written the force term as $F(x) = \partial_x S(x)$. Typically, we build a trajectory of N_{LF} leapfrog steps

$$(x_0, v_0) \to (x_1, v_1) \to \cdots \to (x', v'), \tag{12}$$

and propose x' as the next state in our chain. This proposal state is accepted according to the Metropolis-Hastings criteria [**?**].

$$A(x'|x) = \min\left\{1, \frac{\pi(x')}{\pi(x)}\left|\frac{\partial x'}{\partial x}\right|\right\}. \tag{13}$$

3. Method

Unfortunately, HMC is known to suffer from long auto-correlations and often struggles with multi-modal target densities. Instead, we propose building on the approach from [**? ? ?**]. We introduce two (invertible) neural networks (`xNet`, `vNet`):

$$\texttt{vNet} : (x, F) \to (s_v, t_v, q_v) \tag{14}$$
$$\texttt{xNet} : (x, v) \to (s_x, t_x, q_x) \tag{15}$$

where s, t, q are all of the same dimensionality as x and v, and are parameterized by a set of weights θ. These network outputs (s, t, q) are then used in a generalized MD update (as shown in Fig 2) via:

$$\Gamma^{\pm}_{\theta} : (x, v) \to (x, v') \tag{16}$$
$$\Lambda^{\pm}_{\theta} : (x, v) \to (x', v). \tag{17}$$

where the superscript $\pm$ on $\Gamma^{\pm}_{\theta}, \Lambda^{\pm}_{\theta}$ correspond to the direction $d \sim \mathcal{U}(-1, +1)$ of the update[2]. To ensure that our proposed update remains reversible, we split the x update into two sub-updates on complementary subsets ($x = x_A \cup x_B$):

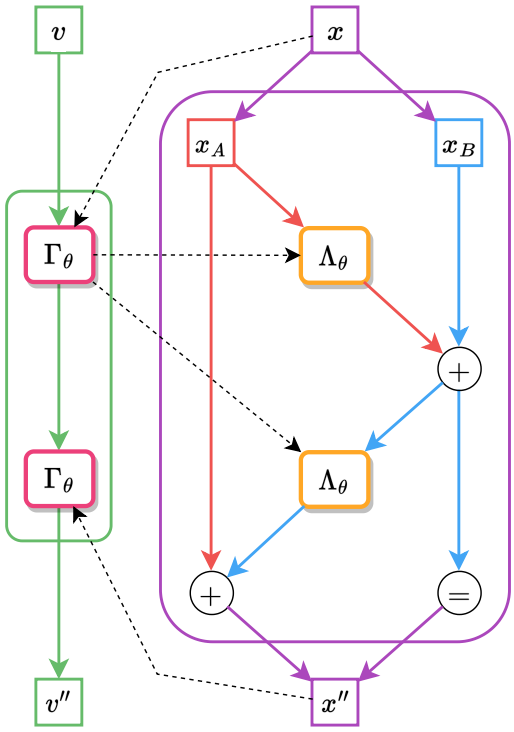

Figure 2: Illustration of the generalized MD update `leapfrog layer` : $(x, v) \to (x'', v'')$.

$$v' = \Gamma_\theta(x, v) \tag{18}$$
$$x' = x_B + \Lambda_\theta(x_A, v') \tag{19}$$
$$x'' = x'_A + \Lambda_\theta(x'_B, v') \tag{20}$$
$$v'' = \Gamma_\theta(x'', v') \tag{21}$$

[2] This can be *absorbed* by constructing trajectories of n forward (+) steps followed by n backward (−) steps. See Appendix XXX for details.

3.1 Algorithm

1. `input:` x
   - Re-sample $v \sim \mathcal{N}(0, 1)$
   - Construct initial state $\xi \coloneqq (x, v)$
2. `forward:` Generate proposal ξ' by passing initial ξ through N_{LF} leapfrog layers:

$$\xi \xrightarrow{\text{LF Layer}} \xi_1 \to \cdots \to \xi_{N_{\text{LF}}} = \xi' \coloneqq (x'', v'') \tag{22}$$

   - Metropolis-Hastings accept / reject:

$$A(\xi'|\xi) = \min\left\{1, \frac{\pi(\xi')}{\pi(\xi)}\left|\mathcal{J}\left(\xi', \xi\right)\right|\right\}, \tag{23}$$

     where $|\mathcal{J}(\xi', \xi)|$ is the determinant of the Jacobian.
3. `backward:` (if training)
   - Evaluate the loss function $\mathcal{L}(\xi', \xi)$ and back propagate
4. `return:` x_{i+1}
   - Evaluate MH criteria (Eq. 23) and return accepted config:

$$x_{i+1} \leftarrow \begin{cases} x'' & \text{w/ prob.} \quad A(\xi'|\xi) \\ x & \text{w/ prob.} \quad 1 - A(\xi'|\xi) \end{cases} \tag{24}$$

3.2 4D $SU(3)$ Model

Write link variables $U_\mu(x) \in SU(3)$:

$$U_\mu(x) = \exp\left[i\omega_\mu^k(x)\lambda^k\right] \tag{25}$$

$$= e^{iQ}, \quad Q \in \mathfrak{su}(3) \tag{26}$$

where $\omega_\mu^k(x) \in \mathbb{R}$ and λ^k are the generators of $SU(3)$. We consider the standard Wilson gauge action

$$S_G = -\frac{\beta}{6}\sum \text{Tr}\left[U_{\mu\nu}(x) + U_{\mu\nu}^\dagger(x)\right] \tag{27}$$

where $U_{\mu\nu}(x) = U_\mu(x)U_\nu(x+\hat{\mu})U_\mu^\dagger(x+\hat{\nu})U_\nu^\dagger(x)$.
As before, we introduce momenta $P_\mu(x) = P_\mu^k(x)\lambda^k$ conjugate to the real fields $\omega_\mu^k(x)$. We can write the Hamiltonian as

$$H[P, U] = \frac{1}{2}P^2 + S_G[U] \tag{28}$$

by Hamilton's equations

$$\frac{d\omega^k}{dt} = \frac{\partial H}{\partial P^k}, \quad \frac{dP^k}{dt} = -\frac{\partial H}{\partial \omega^k}. \tag{29}$$

To update the gauge field U,

$$\frac{d\omega^k}{dt}\lambda^k = P^k\lambda^k \Rightarrow \frac{dQ}{dt} = P \tag{30}$$

Discretizing with step size ε,

$$Q(\varepsilon) = Q(0) + \varepsilon P(0) \Rightarrow \tag{31}$$
$$-i \log U(\varepsilon) = -i \log U(0) + \varepsilon P(0) \tag{32}$$
$$U(\varepsilon) = e^{i\varepsilon P(0)} U(0) \Longrightarrow \tag{33}$$
$$\Lambda : U \to U' \coloneqq e^{i\varepsilon P'} U. \tag{34}$$

and similarly for the momentum update,

$$\frac{dP^k}{dt} = -\frac{\partial H}{\partial \omega^k} = -\frac{\partial H}{\partial Q} = -\frac{dS}{dQ} \Rightarrow \tag{35}$$
$$P(\varepsilon) = P(0) - \varepsilon \left.\frac{dS}{dQ}\right|_{t=0} \tag{36}$$
$$= P(0) - \varepsilon F[U] \tag{37}$$
$$\Gamma : P \to P' \coloneqq P - \frac{\varepsilon}{2} F[U], \tag{38}$$

where $F[U]$ is the force term. In this case, our `vNet` : $(U, F) = (e^{iQ}, F) \to (s_P, t_P, q_P)$. We can use this in the momentum update $\Gamma^{\pm}_{\theta}$ via[3]:

1. `forward`, (+):

$$\Gamma^{+}(U, F) = P \cdot e^{\frac{\varepsilon}{2} s_P} - \frac{\varepsilon}{2} \left[F \cdot e^{\varepsilon q_P} + t_P\right] \tag{39}$$

2. `backward`, (−):

$$\Gamma^{-}(U, F) = e^{-\frac{\varepsilon}{2} s_P} \left\{P + \frac{\varepsilon}{2} \left[F \cdot e^{\varepsilon q_P} + t_P\right]\right\} \tag{40}$$

(a) 100 train iters **(b)** 500 train iters **(c)** 1000 train iters

Figure 4: Evolution of $|\mathcal{J}|$ vs N_{LF}(`logdet`) during the first 1000 training iterations.

[3]Note that $\left(\Gamma^{+}\right)^{-1} = \Gamma^{-}$, i.e. $\Gamma^{+}\left[\Gamma^{-}(U, F)\right] = \Gamma^{-}\left[\Gamma^{+}(U, F)\right] = (U, F)$

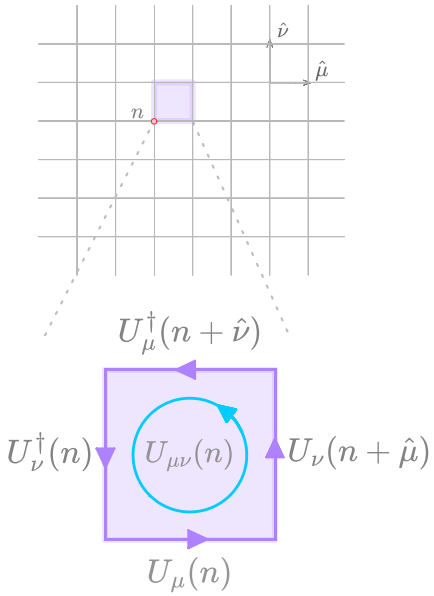

Figure 3: Illustration of the lattice