PROCEEDINGS OF SCIENCE

HMC with Normalizing Flows

Sam Foreman,[a,*] **Taku Izubuchi,**[b,c] **Luchang Jin,**[d] **Xiao-Yong Jin,**[a] **James C. Osborn**[a] **and Akio Tomiya**[b]

[a] *Argonne National Laboratory,*
Lemont, IL 60439
[b] *RIKEN,*
2-1 Hirosawa, Wako, Saitama, 351-0198, Japan
[c] *Brookhaven National Laboratory,*
Upton, NY 11973
[d] *Dept. of Physics, University of Connecticut,*
Storrs, CT 06269
E-mail: foremans@anl.gov, izubuchi@bnl.gov, luchang.jin@uconn.edu, xjin@anl.gov, osborn@alcf.anl.gov

We propose using Normalizing Flows as a trainable kernel within the molecular dynamics update of Hamiltonian Monte Carlo (HMC). By learning (invertible) transformations that simplify our dynamics, we can outperform traditional methods at generating independent configurations. We show that, using a carefully constructed network architecture, our approach can be easily scaled to large lattice volumes with minimal retraining effort.

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

*Speaker

1. Introduction

For a random variable z with a given distribution $z \sim r(z)$, and an invertible function $x = f(z)$ with $z = f^{-1}(x)$, we can use the change of variables formula to write

$$p(x) = r(z)\left|\det \frac{\partial z}{\partial x}\right| = r(f^{-1}(x))\left|\det \frac{\partial f^{-1}}{\partial x}\right| \tag{1}$$

where $r(z)$ is the (simple) prior density, and our goal is to generate independent samples from the (difficult) target distribution $p(x)$. This can be done using *normalizing flows* [6] to construct a model density $q(x)$ that approximates the target distribution, i.e. $q(\cdot) \simeq p(\cdot)$ for a suitably-chosen flow f.

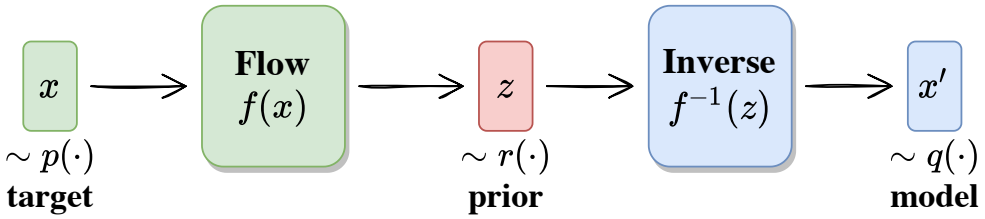


Figure 1: Using a flow to generate data x'. Image adapted from [7]

We can construct a normalizing flow by composing multiple invertible functions f_i so that $x \equiv [f_1 \odot f_2 \odot \cdots \odot f_K](z)$. In practice, the functions f_i are usually implemented as *coupling layers*, which update an "active" subset of the variables, conditioned on the complimentary "frozen" variables [1, 4].

1.1 Affine Coupling Layers

A particularly useful template function for constructing our normalizing flows is the affine coupling layer [3, 6],

$$f(x_1, x_2) = \left(e^{s(x_2)}x_1 + t(x_2),\, x_2\right), \quad \text{with} \quad \log J(x) = \sum_k [s(x_2)]_k$$
$$f^{-1}(x_1', x_2') = \left((x_1' - t(x_2'))e^{-s(x_2')},\, x_2'\right), \quad \text{with} \quad \log J(x') = \sum_k -\left[s(x_2')\right]_k$$

where $s(x_2)$ and $t(x_2)$ are of the same dimensionality as x_1 and the functions act element-wise on the inputs.

In order to effectively draw samples from the correct target distribution $p(\cdot)$, our goal is to minimize the error introduced by approximating $q(\cdot) \simeq p(\cdot)$. To do so, we use the (reverse) Kullback-Leibler (KL) divergence from Eq. 2, which is minimized when $p = q$.

$$D_{\mathrm{KL}}(q\|p) \equiv \int dy q(y)\left[\log q(y) - \log p(y)\right] \tag{2}$$
$$\simeq \frac{1}{N}\sum_{i=1}^{N}\left[\log q(y_i) - \log p(y_i)\right], \text{ where } y_i \sim q \tag{3}$$

2. Trivializing Map

Our goal is to evaluate expectation values of the form

$$\langle O \rangle = \tfrac{1}{\mathcal{Z}} \int dx\, O(x) e^{-S(x)}. \tag{4}$$

Using a normalizing flow, we can perform a change of variables $x = f(z)$, so Eq. 4 becomes

$$\langle O \rangle = \frac{1}{\mathcal{Z}} \int dz\, |\det\left[J(z)\right]|\, O(f(z)) e^{-S(f(z))}, \text{ where } J(z) = \frac{\partial f(z)}{\partial z} \tag{5}$$

$$= \frac{1}{\mathcal{Z}} \int dz O(f(z)) e^{-S(f(z)) + \log|\det[J(z)]|}. \tag{6}$$

We require the Jacobian matrix, $J(z)$, to be:

1. Injective (1-to-1) between domains of integration
2. Continuously differentiable (*or*, differentiable with continuous inverse)

The function f is a *trivializing map* [5] when $S(f(z)) - \log|\det J(z)| = \text{const.}$, and our expectation value simplifies to

$$\langle O \rangle = \frac{1}{\mathcal{Z}^*} \int dz\, O(f(z)), \text{ where } \frac{1}{\mathcal{Z}^*} = \frac{1}{\mathcal{Z}} \exp(-\text{const.}). \tag{7}$$

3. Field Transformation HMC: `fthmc`

We can implement the trivializing map defined above using a normalizing flow model. For conjugate momenta π, we can write the Hamiltonian as

$$H(z, \pi) = \frac{1}{2}\pi^2 + S(f(z)) - \log\left|\det J(f(z))\right|, \tag{8}$$

and the associated equations of motion as

$$\dot{z} = \frac{\partial H}{\partial \pi} = \pi \tag{9}$$

$$\dot{\pi} = -J(z) S'(f(z)) + \operatorname{tr}\left[J^{-1} \frac{d}{dz} J\right]. \tag{10}$$

If we introduce a change of variables, $\pi = J(z)\rho = J(f^{-1}(x))\rho$ and $z = f^{-1}(x)$, the determinant of the Jacobian matrix reduces to 1, and we obtain the modified Hamiltonian

$$\tilde{H}(x, \rho) = \frac{1}{2}\rho^\dagger \rho + S(x) - \log|\det J|. \tag{11}$$

As shown in Fig. 2, we can use $f^{-1} : z \to x$ to perform HMC updates on the transformed variables x, and $f : x \to z$ to recover the physical target distribution.

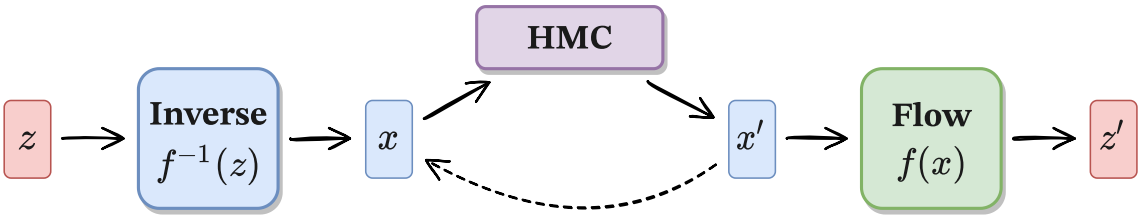


Figure 2: Normalizing flow with inner HMC block.

3.1 Hamiltonian Monte Carlo (HMC)

We describe the general procedure of the Hamiltonian Monte Carlo algorithm [2].

1. Introduce $v \sim \mathcal{N}(0, \mathbb{I}_n) \in \mathbb{R}^n$ and write the joint distribution as

$$p(x, v) = p(x)p(v) \propto e^{-S(x)} e^{-\frac{1}{2}v^T v} \tag{12}$$

2. Evolve the joint system $(\dot{x}, \dot{v})$ according to Hamilton's equations along $H = \text{const.}$ using the leapfrog integrator:

$$\textbf{(a.)}\ \tilde{v} \leftarrow v - \frac{\varepsilon}{2}\partial_x S(x) \quad \textbf{(b.)}\ x' \leftarrow x + \varepsilon\tilde{v} \quad \textbf{(c.)}\ v' \leftarrow \tilde{v} - \frac{\varepsilon}{2}\partial_x S(x') \tag{13}$$

3. Accept or reject the proposal configuration using the Metropolis-Hastings test,

$$x_{i+1} = \begin{cases} x' \text{ with probability } A(\xi'|\xi) \equiv \min\left\{1, \frac{p(\xi')}{p(\xi)}\left|\frac{\partial \xi'}{\partial \xi^T}\right|\right\} \\ x \text{ with probability } 1 - A(\xi'|\xi) \end{cases} \tag{14}$$

4. 2D $U(1)$ Gauge Theory

Let $U_\mu(n) = e^{ix_\mu(n)} \in U(1)$, with $x_\mu(n) \in [-\pi, \pi]$ denote the *link variables*, where $x_\mu(n)$ is a link at the site n oriented in the direction $\hat{\mu}$. We can write our target distribution $p(x)$ in terms of the Wilson action $S(x)$ as

$$p(x) \propto e^{-S(x)}, \quad S(x) \equiv \sum_P 1 - \cos x_P \tag{15}$$

and $x_P = x_\mu(n) + x_\nu(n+\hat{\mu}) - x_\mu(n+\hat{\nu}) - x_\nu(n)$, as shown in Fig. 3. For a given lattice configuration, we can define the topological charge $Q \in \mathbb{Z}$ as

$$Q = \frac{1}{2\pi}\sum_P \arg(x_P), \tag{16}$$

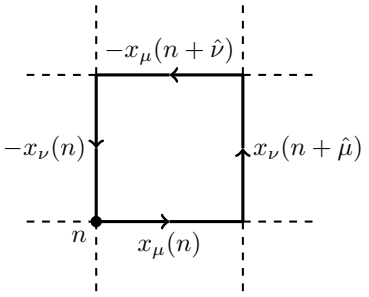


Figure 3: Elementary plaquette x_P on the lattice.

where $\arg(x_P) \in [-\pi, \pi]$. We are interested in how this quantity evolves over a finite length Markov chain, and in particular, we can define the tunneling rate, δQ as

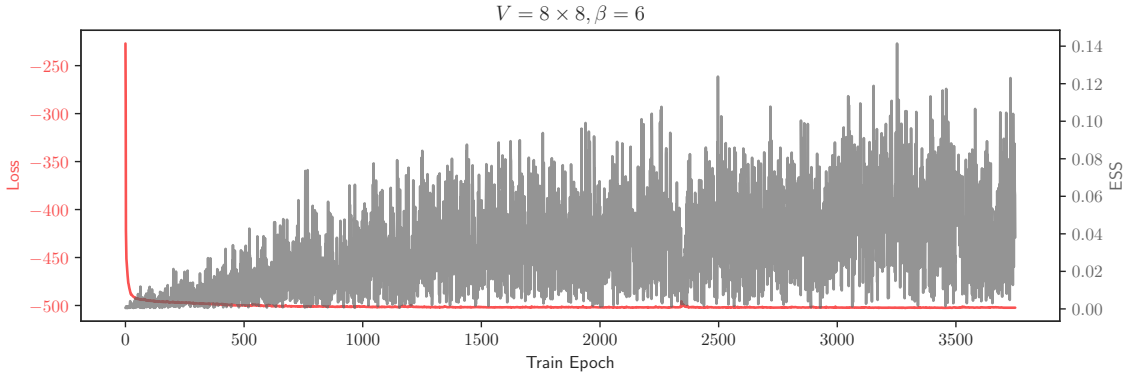
Remove references to δQ?

$$\delta Q = \sqrt{(Q_{i+1} - Q_i)^2} \tag{17}$$

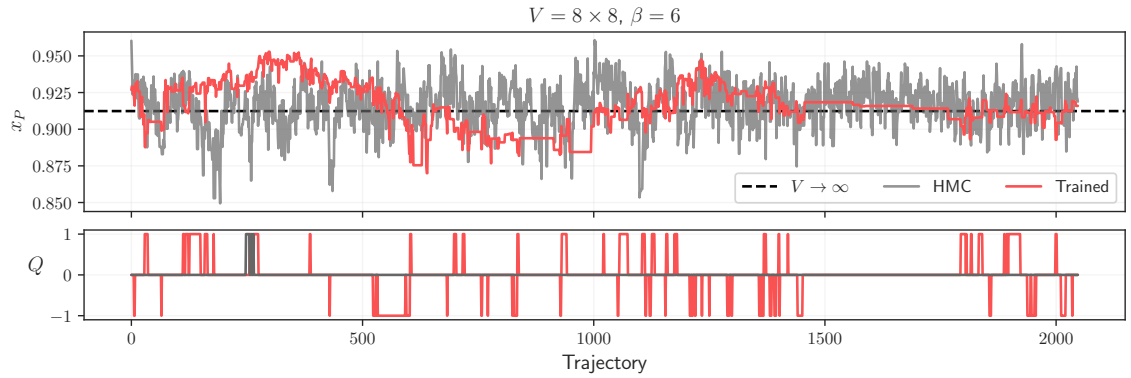
where the difference is between subsequent states in the chain. This quantity is analogous to the lag-one autocorrelation in the topological charge.

5. Results

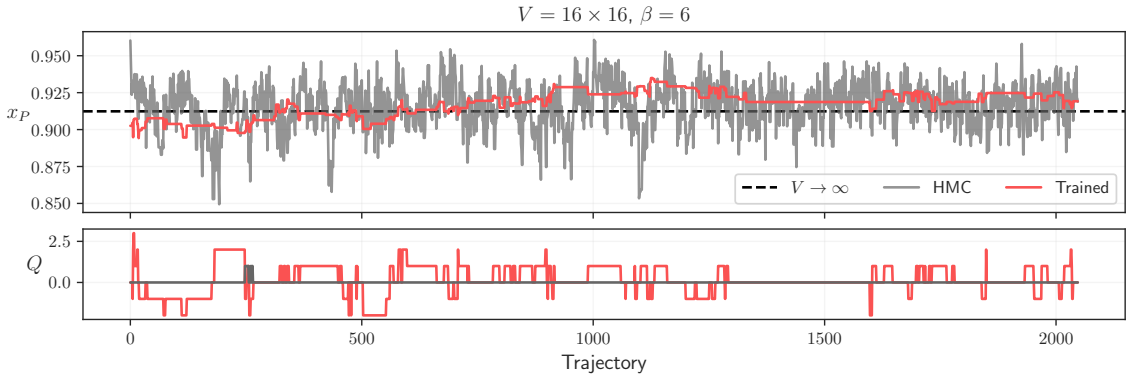
For traditional HMC, we see in Fig 4b,4c that $Q \simeq 0$ for across all trajectories for both 8×8 and 16×16 lattice volumes. Conversely, we see in Fig 4b,4c that multiple tunneling events (characterized by $\delta Q > 0$) occur for both 8×8 and 16×16 volumes.



(a) Loss and ESS vs train epoch at $\beta = 6$ on $V = 8 \times 8$ lattice.



(b) x_P and Q histories for both HMC and the trained model, at $\beta = 6$ with $V = 8 \times 8$ lattice.



(c) The same model from Fig 4b used to generate configurations on $V = 16 \times 16$ lattice.

Figure 4

NOTE: The HMC data in Fig 4c is incorrectly reusing the same data from the 8×8 run (the Trained data is correct, however). I will update this soon.

5.1 Volume Scaling

We use gauge equivariant coupling layers that act on plaquettes as the base layer for our network architecture. As in [1], these layers are composed of inner coupling layers which are implemented as stacks of convolutional layers. One advantage of using convolutional layers is that we can re-use the trained weights when scaling up to larger lattice volumes. Explicitly, when scaling up the lattice volume we can initialize the weights of our new network with the previously trained values. This approach has the advantage of requiring minimal retraining effort while being able to efficiently generate models on large lattice volumes.

References

[1] Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Introduction to Normalizing Flows for Lattice Field Theory. *arXiv e-prints*, 1 2021.

[2] Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv e-prints*, page arXiv:1701.02434, January 2017.

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016.

[4] Gurtej Kanwar, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Equivariant flow-based sampling for lattice gauge theory. *Phys. Rev. Lett.*, 125(12):121601, 2020.

[5] Martin Lüscher. Trivializing Maps, the Wilson Flow and the HMC Algorithm. *Communications in Mathematical Physics*, 293(3):899919, Nov 2009.

[6] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[7] Lilian Weng. Flow-based deep generative models. *lilianweng.github.io/lil-log*, 2018.