Leapfrog Layers

A Trainable Framework for Effective Topological Sampling

Sam Foreman*, Xiao-Yong Jin, James C. Osborn
July, 2021

*foremans@anl.gov

arXiv:2105.03418

bit.ly/l2hmc-lattice21

bit.ly/l2hmc-surprise

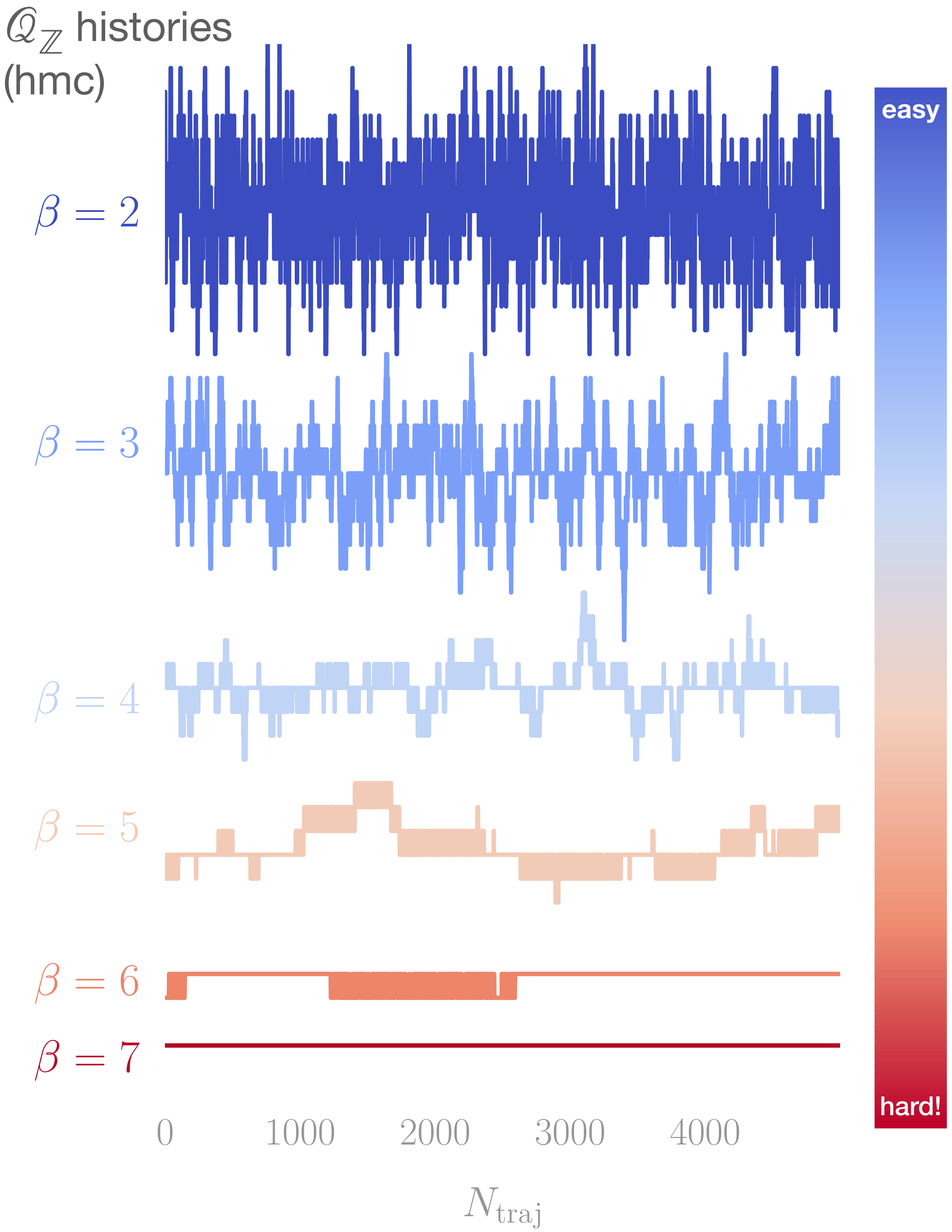
github.com/saforem2/l2hmc-qcd

Argonne NATIONAL LABORATORY

LATTICE21
JULY 26-30 2021, ZOOM/GATHER@MIT

Critical Slowing Down

- Goal: Draw *independent samples* from target distribution $p(x)$.
  - Generating independent gauge configurations is a *major* bottleneck for LatticeQCD.
- **Topological Freezing**
  - As we approach the continuum limit $\beta \to \infty$, the MCMC updates get stuck in sectors of fixed gauge topology.
    - Number of trajectories needed to adequately sample different topological sectors **increases exponentially**



Hamiltonian Monte Carlo (HMC)

- Introduce $v \sim \mathcal{N}(0,\mathbb{1})$, then the target becomes:

$$p(x, v) = p(x) \cdot p(v) = e^{-Sx} \cdot e^{-v^T v/2}$$

- Evolve the joint $\xi \equiv (x, v)$ system using Hamilton's equations along $H = \text{const}$:

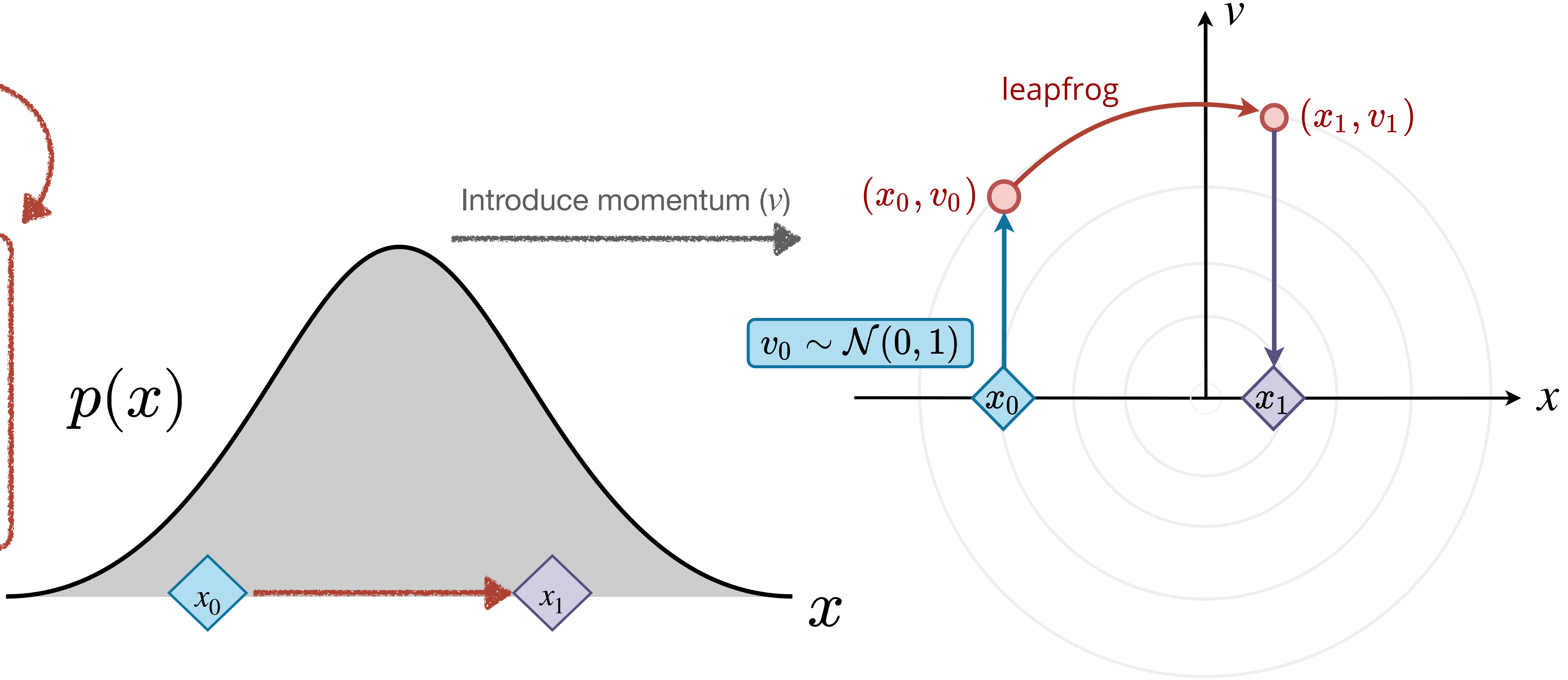
$$\dot{x} = \frac{\partial H}{\partial v}, \quad \dot{v} = \frac{\partial H}{\partial v}$$

- **Leapfrog Integrator:**

1. $\tilde{v} \leftarrow v - \varepsilon \cdot \partial_x S(x)/2$
2. $x' \leftarrow x + \varepsilon \tilde{v}$
3. $v' \leftarrow \tilde{v} - \varepsilon \partial_x S(x')/2$

- Accept / reject proposal x' using MH:

$$x_{i+1} \leftarrow \begin{cases} x' & \text{w/prob.} \quad A(\xi'|\xi) = \min\left\{1, \frac{p(\xi')}{p(\xi)} \left|\frac{\partial \xi'}{\partial \xi}\right|\right\}, \\ x_i & \text{w/prob.} \quad 1 - A(\xi'|\xi) \end{cases}$$



Leapfrog Layer

- Introduce persistent direction $d \sim \mathcal{U}(+,-)$ *(forward, backward).*
- **Target distribution:** $p(\xi) = p(x) \cdot p(v) \cdot p(d)$
- **k[th]-Leapfrog Layer:** $\xi_k \equiv (x_k, v_k, \pm) \longrightarrow (x''_k, v''_k, \pm) \equiv \xi_{k+1}$

(input) $\xi_0 \to \xi_1 \to \cdots \to \xi_k \to \xi_{k+1} \to \cdots \to \xi_{N_{\rm LF}} \equiv \xi''$ (proposal)

- Construct a *trajectory* by passing ξ_k through $k \in \{1, 2, \ldots, N_{\rm LF}\}$ leapfrog layers.

Invertible NN

momentum scaling

force scaling

translation

v update: $(d = +)$

$$v''_k = \Gamma^+(v_k; \zeta_{v_k}) \equiv v_k \odot \exp\left(\frac{\varepsilon_v^k}{2} s_v^k(\zeta_{v_k})\right) - \frac{\varepsilon_v^k}{2}\left[\partial_x S(x_k) \odot \exp\left(\varepsilon_v^k q_v^k(\zeta_{v_k})\right) + t_v^k(\zeta_{v_k})\right]$$

$$\zeta_{v_k} = [x_k, \partial_x S(x_k)]$$

x update: $(d = +)$

$$x''_k = \Lambda^+(x_k; \zeta_{v_k}) \equiv x_k \odot \exp\left(\varepsilon_x^k s_x^k(\zeta_{x_k})\right) + \varepsilon_x^k\left[v'_k \odot \exp\left(\varepsilon_x^k q_x^k(\zeta_{x_k})\right) + t_x^k(\zeta_{x_k})\right]$$

$$\zeta_{x_k} = [\bar{m}^k \odot x_k, v_k]$$

l2hmc: Generalized Leapfrog

- **Leapfrog Step:** $\xi_k \to \xi''_k$

Invertible NN

1. Half-step v update:

$$v'_k = \Gamma^\pm(v_k; \zeta_{v_k})$$

2. Full-step, ***half***-x update:

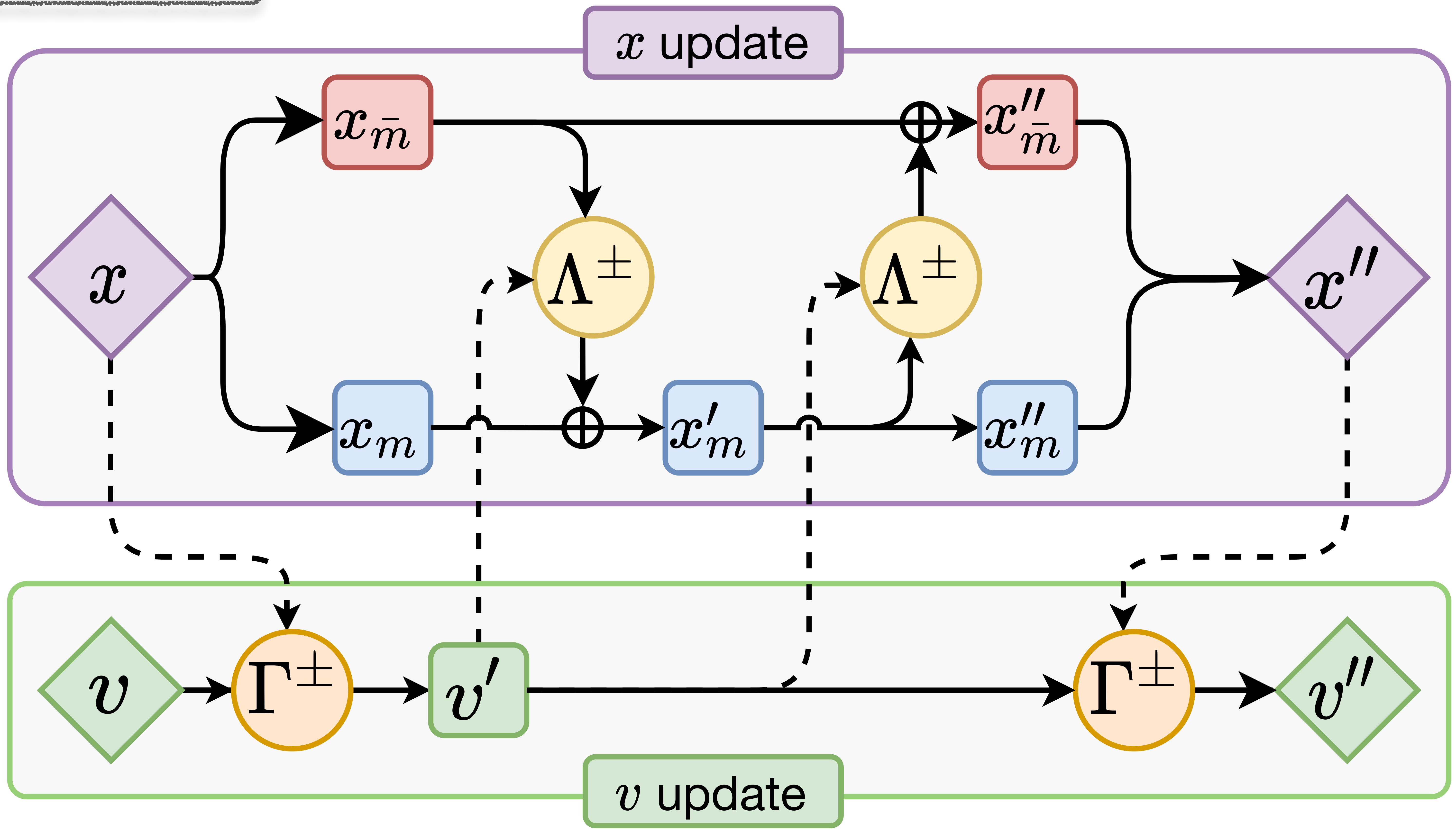
$$x'_k = m^k \odot x_k + \bar{m}^k \odot \Lambda^\pm(x_k; \zeta_{x_k})$$

3. Full-step, ***half***-x update:

$$x''_k = \bar{m}^k \odot x'_k + m^k \odot \Lambda^\pm(x'_k; \zeta_{x'_k})$$

4. Half-step v update:

$$v''_k = \Gamma^\pm(v'_k; \zeta_{v_k})$$



l2hmc: Generalized Leapfrog

- **Leapfrog Step:** $\xi_k \rightarrow \xi''_k$

1. Half-step v update:

$$v'_k = \Gamma^{\pm}(v_k; \zeta_{v_k})$$

2. Full-step, ***half***-x update:

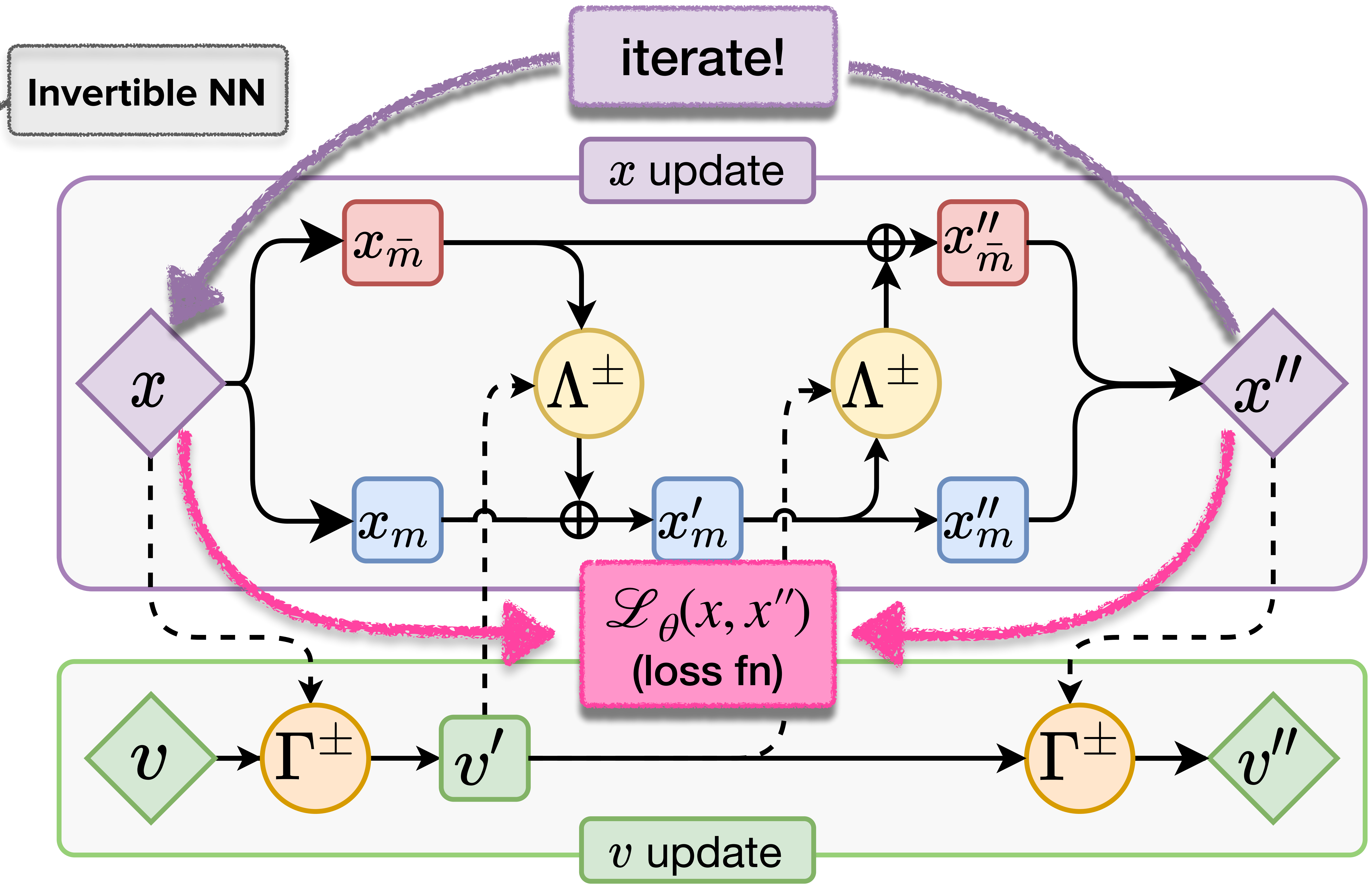
$$x'_k = m^k \odot x_k + \bar{m}^k \odot \Lambda^{\pm}(x_k; \zeta_{x_k})$$

3. Full-step, ***half***-x update:

$$x''_k = \bar{m}^k \odot x'_k + m^k \odot \Lambda^{\pm}(x'_k; \zeta_{x'_k})$$

4. Half-step v update:

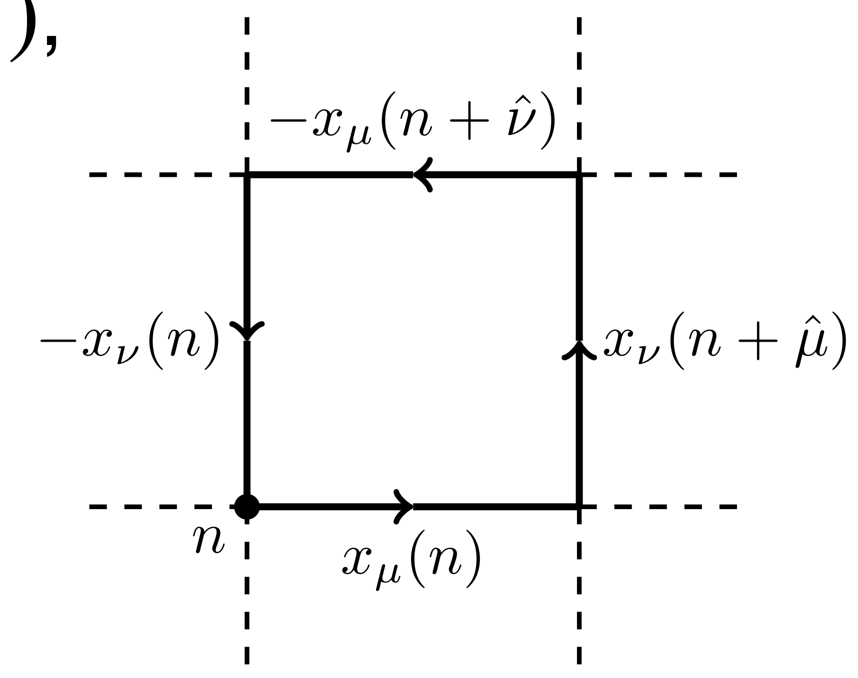
$$v''_k = \Gamma^{\pm}(v'_k; \zeta_{v_k})$$



2D $U(1)$ Lattice Gauge Theory

- Link variables $U_\mu(n) = e^{ix_\mu(n)} \in U(1)$,

  with $x_\mu(n) \in [-\pi, \pi]$.

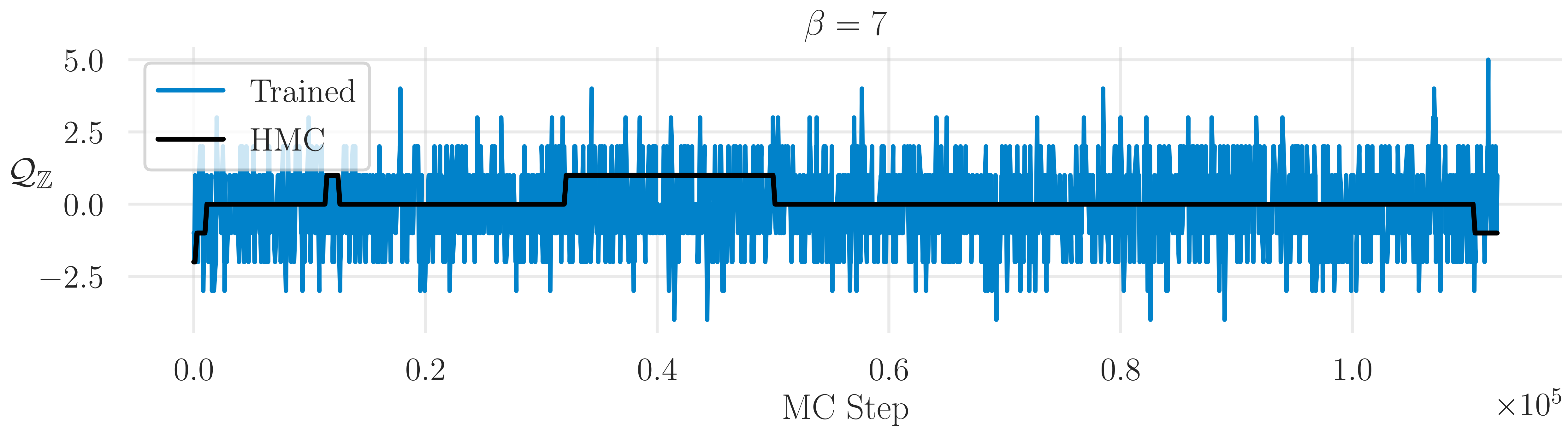


- **Wilson action:**
  - $S_\beta(x) = \beta \sum_P 1 - \cos x_P,$
  - $x_P = x_\mu(n) + x_\nu(n+\hat{\mu}) - x_\mu(n+\hat{\nu}) - x_\nu(n)$
- **Topological charge:**
  - $\mathcal{Q}_\mathbb{R} = \frac{1}{2\pi} \sum_P \sin x_P \in \mathbb{R}$ — continuous, differentiable
  - $\mathcal{Q}_\mathbb{Z} = \frac{1}{2\pi} \sum_P \lfloor x_P \rfloor \in \mathbb{Z}$ — discrete, hard to work with

    $\lfloor x_P \rfloor = x_P - 2\pi \left\lfloor \frac{x_P + \pi}{2\pi} \right\rfloor$

Loss function, $\mathcal{L}(\theta)$

- We maximize the *expected squared charge difference:*
  - $\mathcal{L}(\theta) = \mathbb{E}_{p(\xi)}\left[-\delta\mathcal{Q}^2_\mathbb{R}(\xi', \xi) \cdot A(\xi'|\xi)\right]$
  - $\delta\mathcal{Q}^2_\mathbb{R}(\xi', \xi) = \left(\mathcal{Q}_\mathbb{R}(x') - \mathcal{Q}_\mathbb{R}(x)\right)^2$ *(squared charge diff.)*
  - $A(\xi'|\xi) = \min\left\{1, \frac{p(\xi')}{p(\xi)}\left|\frac{\partial \xi'}{\partial \xi^T}\right|\right\}$ *(acceptance prob.)*



Simulated Annealing

- Introduce an **annealing schedule** during the training phase:
  - $\{\gamma_t\}_{t=0}^{N} = \{\gamma_0, \gamma_1, \ldots, \gamma_{N-1}, \gamma_N\}$,
  - $\gamma_0 < \gamma_1 < \cdots < \gamma_N \equiv 1$,
  - $\delta_\gamma \equiv \|\gamma_{t+1} - \gamma_t\| \ll 1$

ex: $\{0.1, 0.2, 0.3, \ \ldots, 0.9, 1.0\}$

increasing

varied *slowly*

- For $\|\gamma_t\| < 1$, this helps to rescale *(shrink)* the energy barriers between isolated modes
  - Allows sampler to explore previously inaccessible regions of the target distribution.
- Target distribution becomes:
  - $p_t(x) \propto e^{-\gamma_t S_\beta(x)}$, for $t = 0, 1, \ldots, N$

Training Algorithm

input:

1. Loss function, $\mathcal{L}_\theta(\xi', \xi, A(\xi'|\xi))$
2. Batch of initial states, x
3. Learning rate schedule, $\{\alpha_t\}_{t=0}^{N_{\text{train}}}$
4. Annealing schedule, $\{\gamma_t\}_{t=0}^{N_{\text{train}}}$
5. Target distribution, $p_t(x) \propto e^{-\gamma_t S_\beta(x)}$

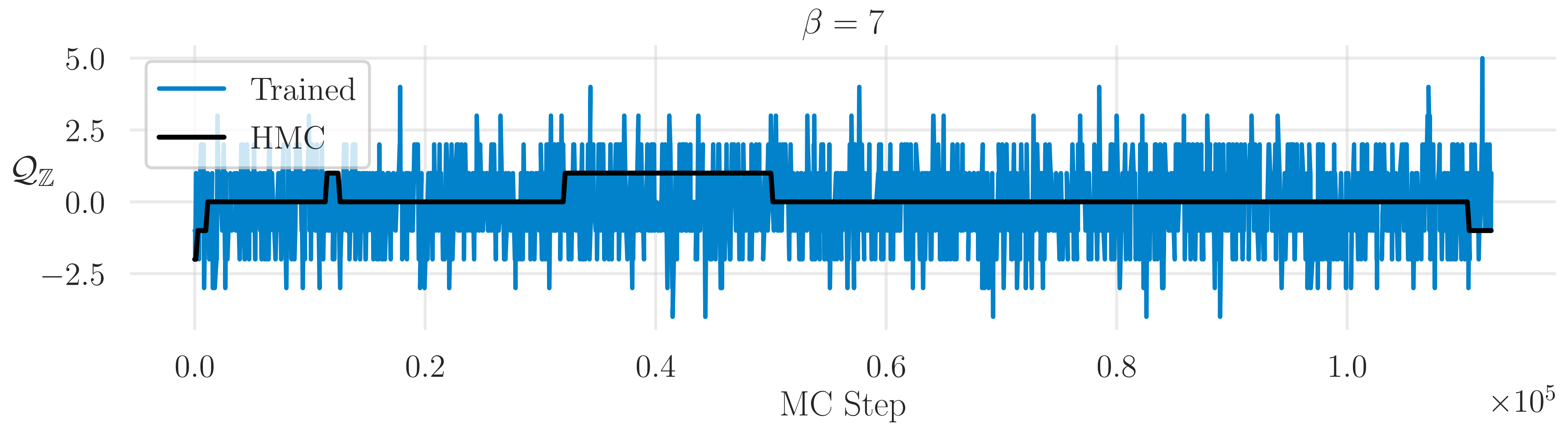
Initialize weights θ

for $0 \leq t < N_{\text{train}}$:

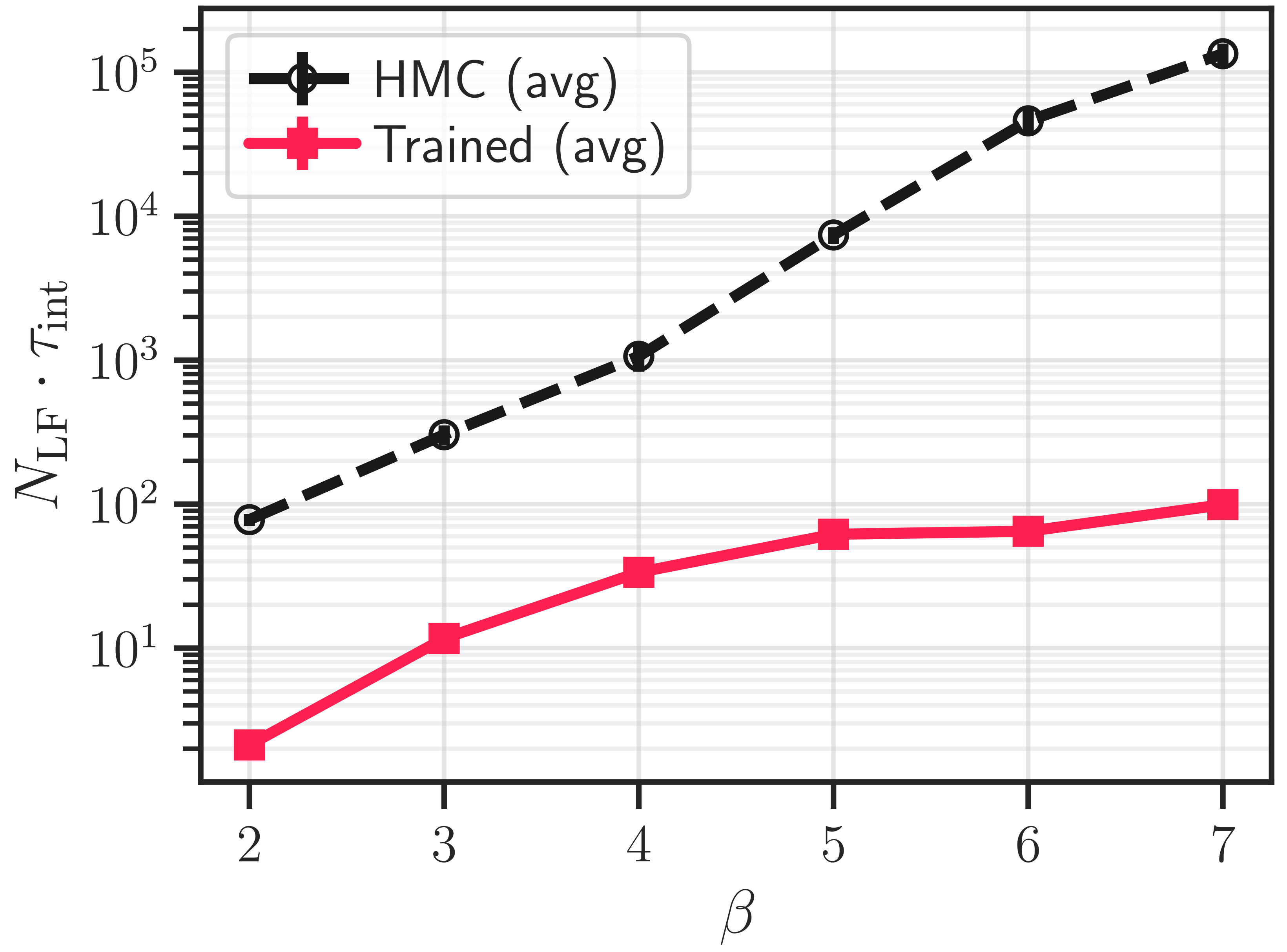
- resample $v \sim \mathcal{N}(0, \mathbb{1})$ — re-sample momentum
- resample $d \sim \mathcal{U}(+, -)$ — + direction
- construct $\xi_0 \equiv (x_0, v_0, d_0)$
- **for** $0 \leq k < N_{\text{LF}}$: — construct trajectory
  - propose (leapfrog layer) $\xi'_k \leftarrow \xi_k$
- compute $A(\xi'|\xi) = \min\left\{1, \frac{p(\xi')}{p(\xi)}\left|\frac{\partial \xi'}{\partial \xi^T}\right|\right\}$ — Compute loss
- update $\mathcal{L} \leftarrow \mathcal{L}_\theta(\xi', \xi, A(\xi'|\xi))$
- backprop $\theta \leftarrow \theta - \alpha_t \nabla_\theta \mathcal{L}$ — + backprop
- assign $x_{t+1} \leftarrow \begin{cases} x' & \text{with probability } A(\xi'|\xi) \\ x & \text{with probability } (1 - A(\xi'|\xi)). \end{cases}$ — Metropolis-Hastings accept/reject

Results

- Want to calculate $\langle \mathcal{O} \rangle \propto \int [\mathcal{D}x]\ \mathcal{O}(x) e^{-S(x)}$
- If we had *independent* configurations, we could approximate by
  - $\langle \mathcal{O} \rangle \simeq \frac{1}{N} \sum_{n=1}^{N} \mathcal{O}(x_n) \to \sigma^2 = \frac{1}{N} \mathrm{Var}\left[\mathcal{O}(x)\right]$
- Accounting for *autocorrelation*: $\sigma^2 = \frac{\tau_{\mathrm{int}}^{\mathcal{O}}}{N} \mathrm{Var}\left[\mathcal{O}(x)\right]$
- We measure the performance of our model by looking at the *integrated autocorrelation time*, τ_{int} of the topological charge $\mathcal{Q}_{\mathbb{Z}}$.
- For generic HMC, it is known that τ_{int} grows exponentially as $\beta \to \infty$ **(critical slowing down)**



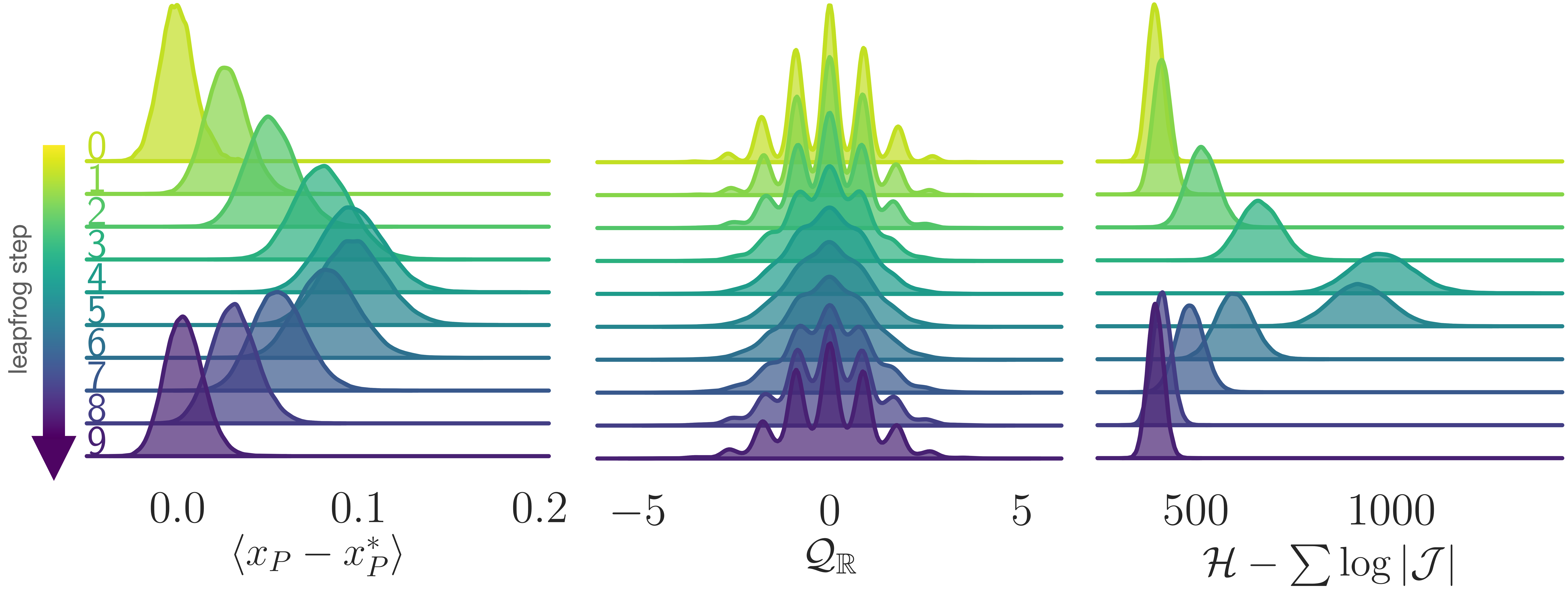
(d.) Plot of the topological charge history $\mathcal{Q}_{\mathbb{Z}}$ vs MC Step



(c.) Estimate of the integrated autocorrelation time τ_{int} vs β for both the trained model and generic HMC.

Interpretation

- Look at how different quantities evolve over the course of a trajectory (N_{LF} leapfrog layers)
  - See that the sampler artificially *increases the energy* during the first half of the trajectory

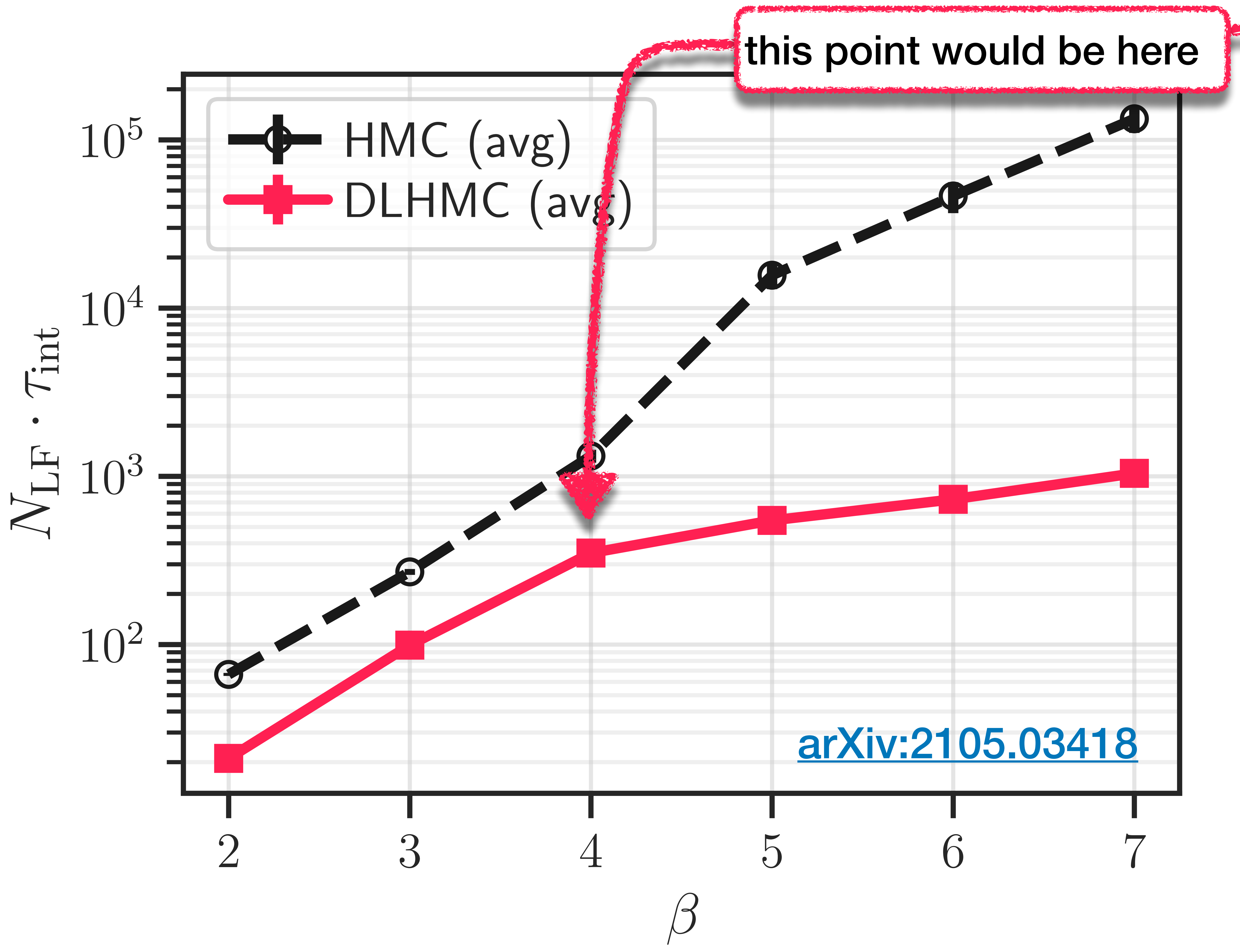


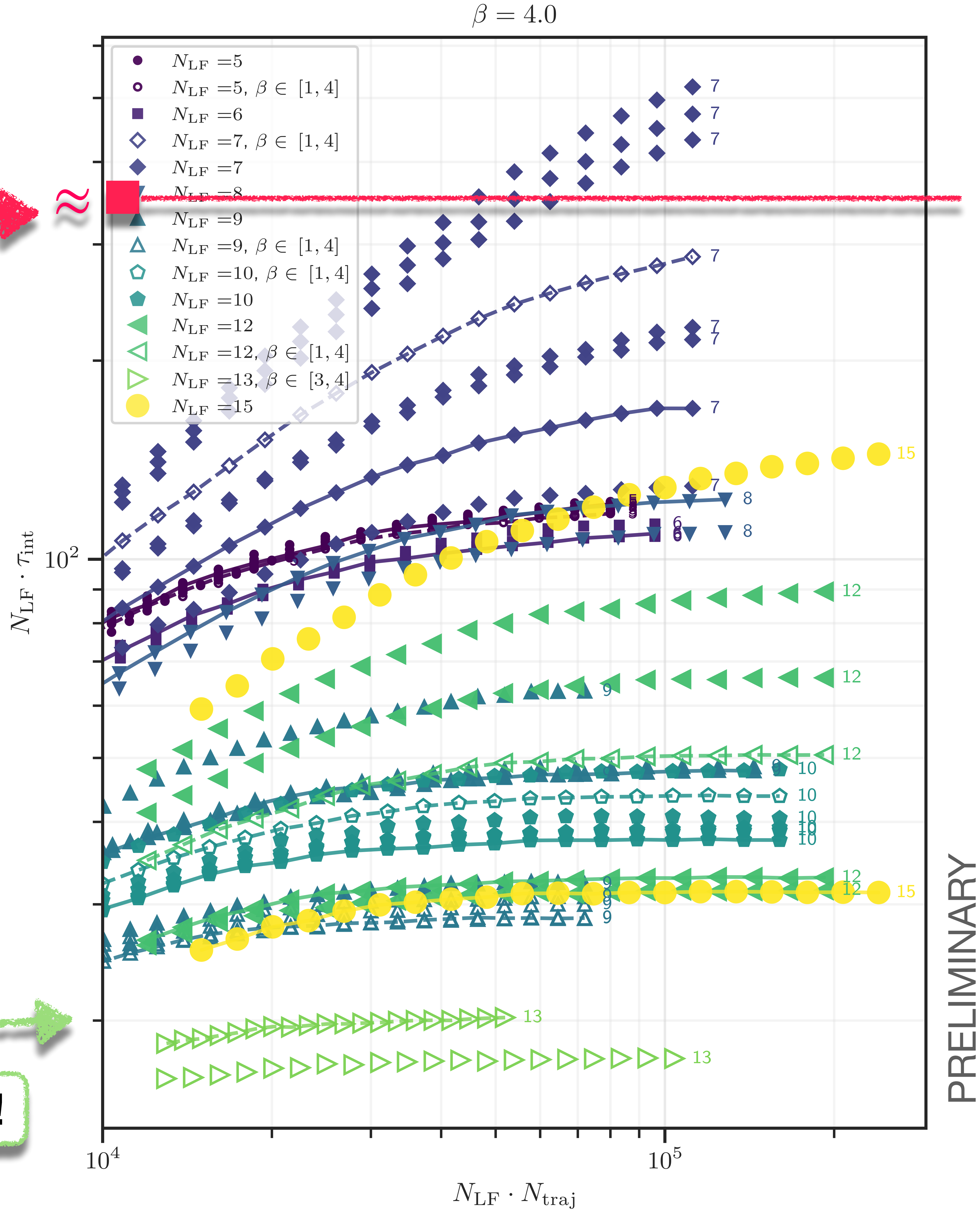
(a.) Deviation in the average plaquette, x_P

(b.) Evolution of the continuous charge $\mathcal{Q}_{\mathbb{R}}$

(c.) Evolution of the effective energy

New Results



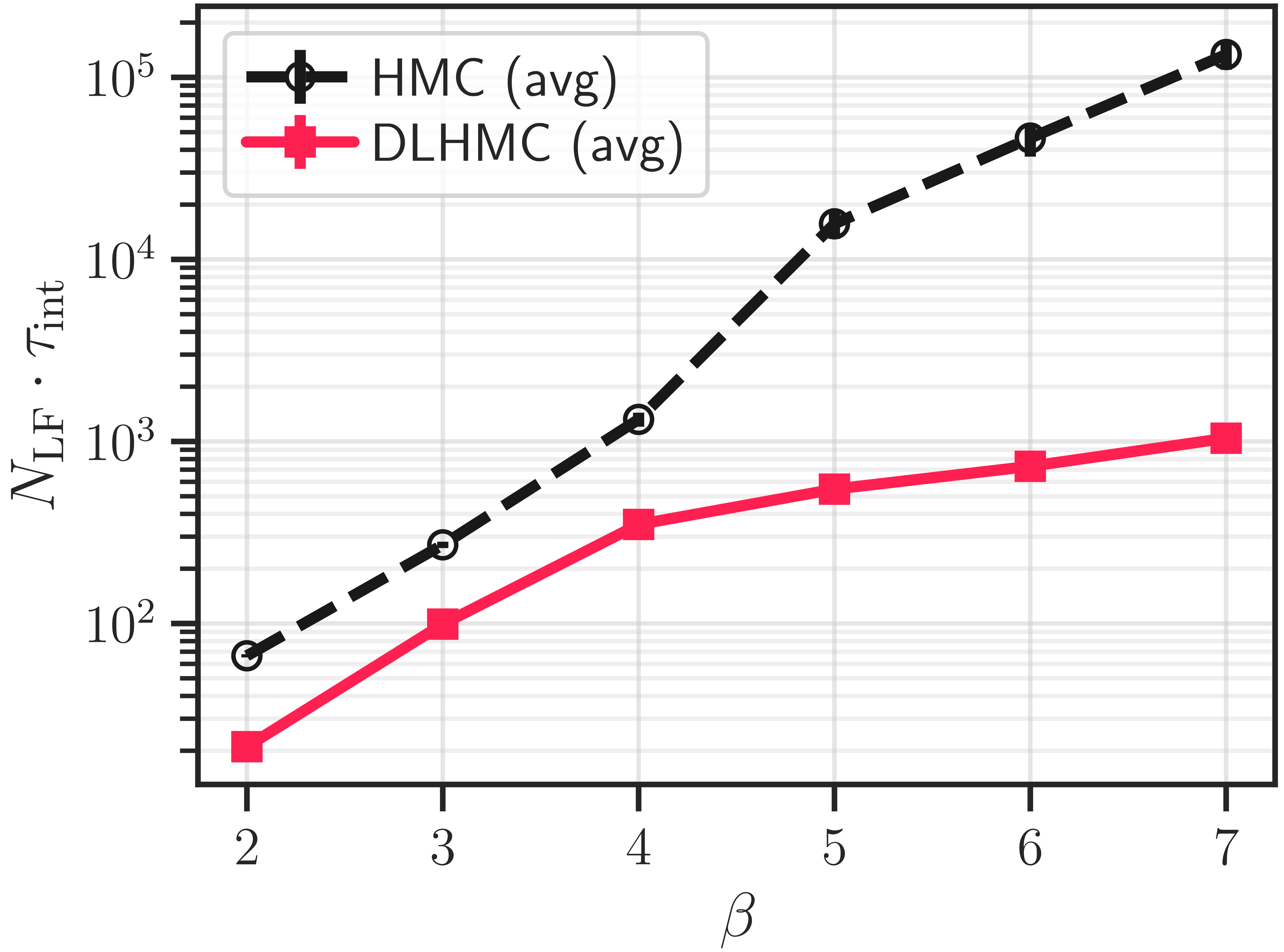


- **Better** performance: ≈**10** × previous results

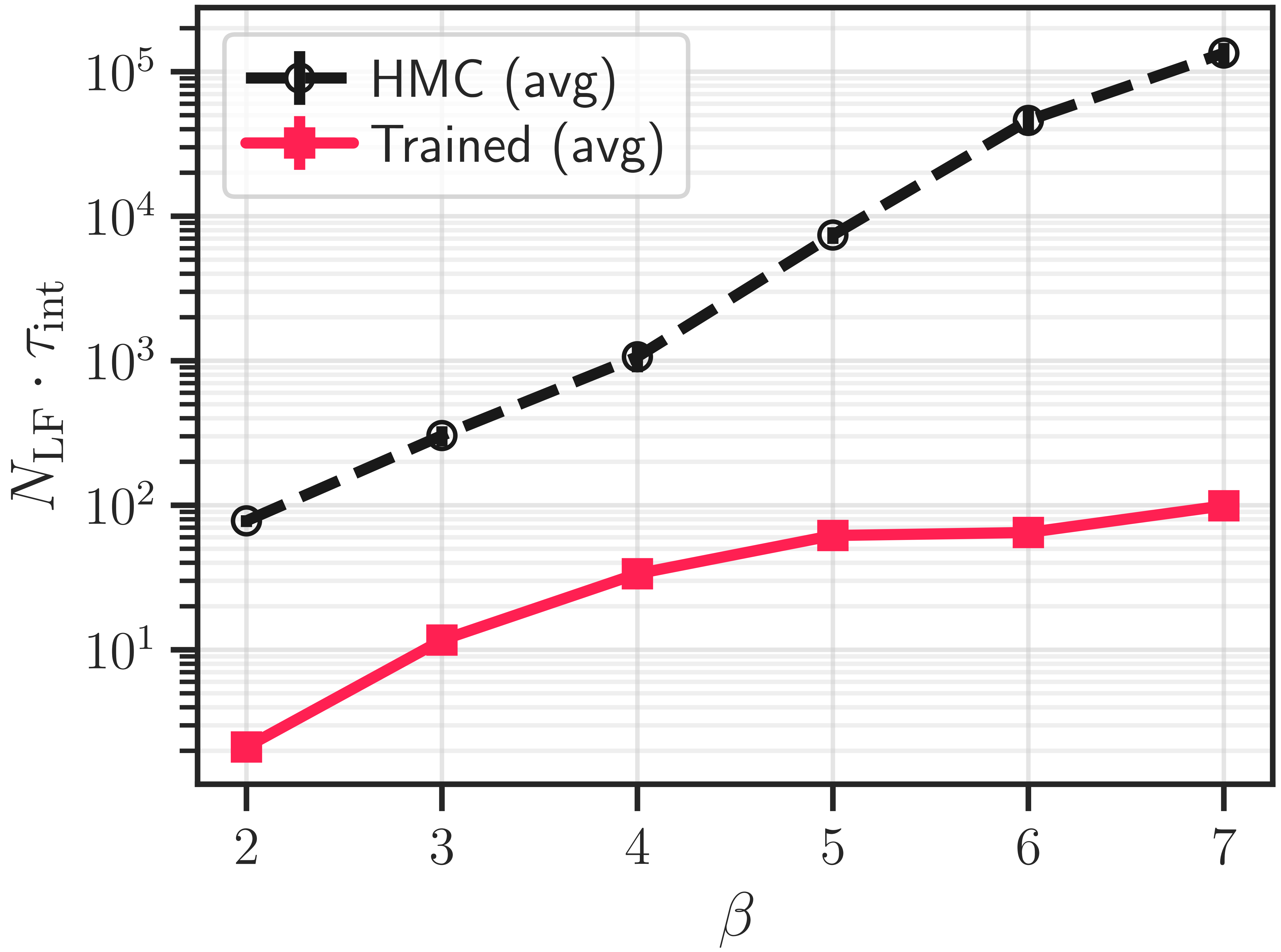
New networks!

Comparison

previous (from arXiv:2105.03418)



new (preliminary)



Acknowledgements

- **Collaborators:**
  - Xiao-Yong Jin,
  - James C. Osborn
- Huge thank you to:
  - Yannick Meurice
  - Norman Christ
  - Akio Tomiya
  - Luchang Jin
  - Chulwoo Jung
  - Peter Boyle
  - Taku Izubuchi
  - Critical Slowing Down group (ECP)
  - ALCF Staff + Datascience group

U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

ECP EXASCALE COMPUTING PROJECT

LATTICE 21

l2hmc-qcd

Sam Foreman*, Xiao-Yong Jin, & James C. Osborn

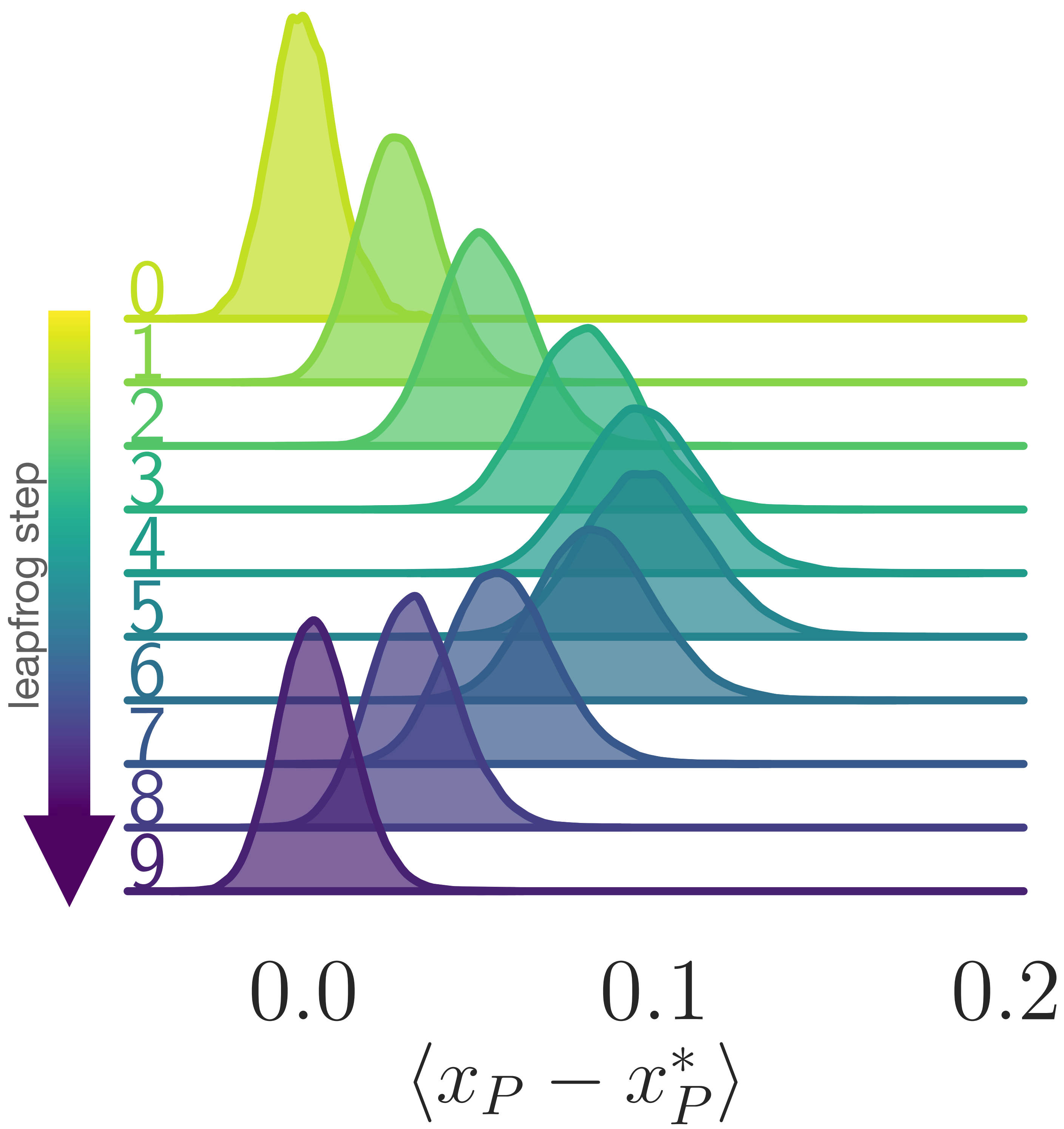
*foremans@anl.gov

A trainable framework for accelerating HMC on lattice gauge models.

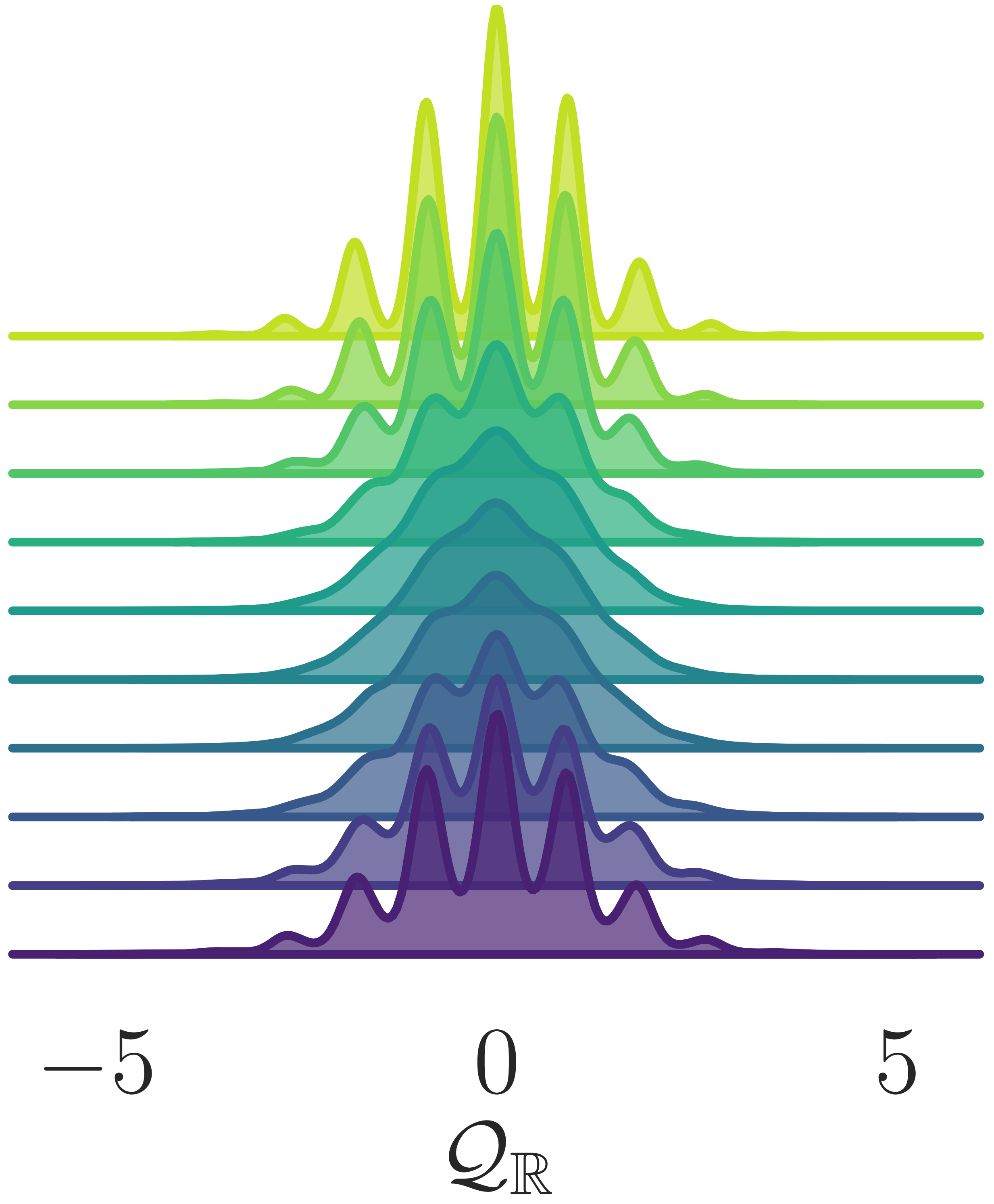
arXiv:2105:03418 bit.ly/l2hmc-lattice21 github.com/saforem2/l2hmc-qcd

LATTICE 21 JULY 26-30 2021, ZOOM/GATHER@MIT

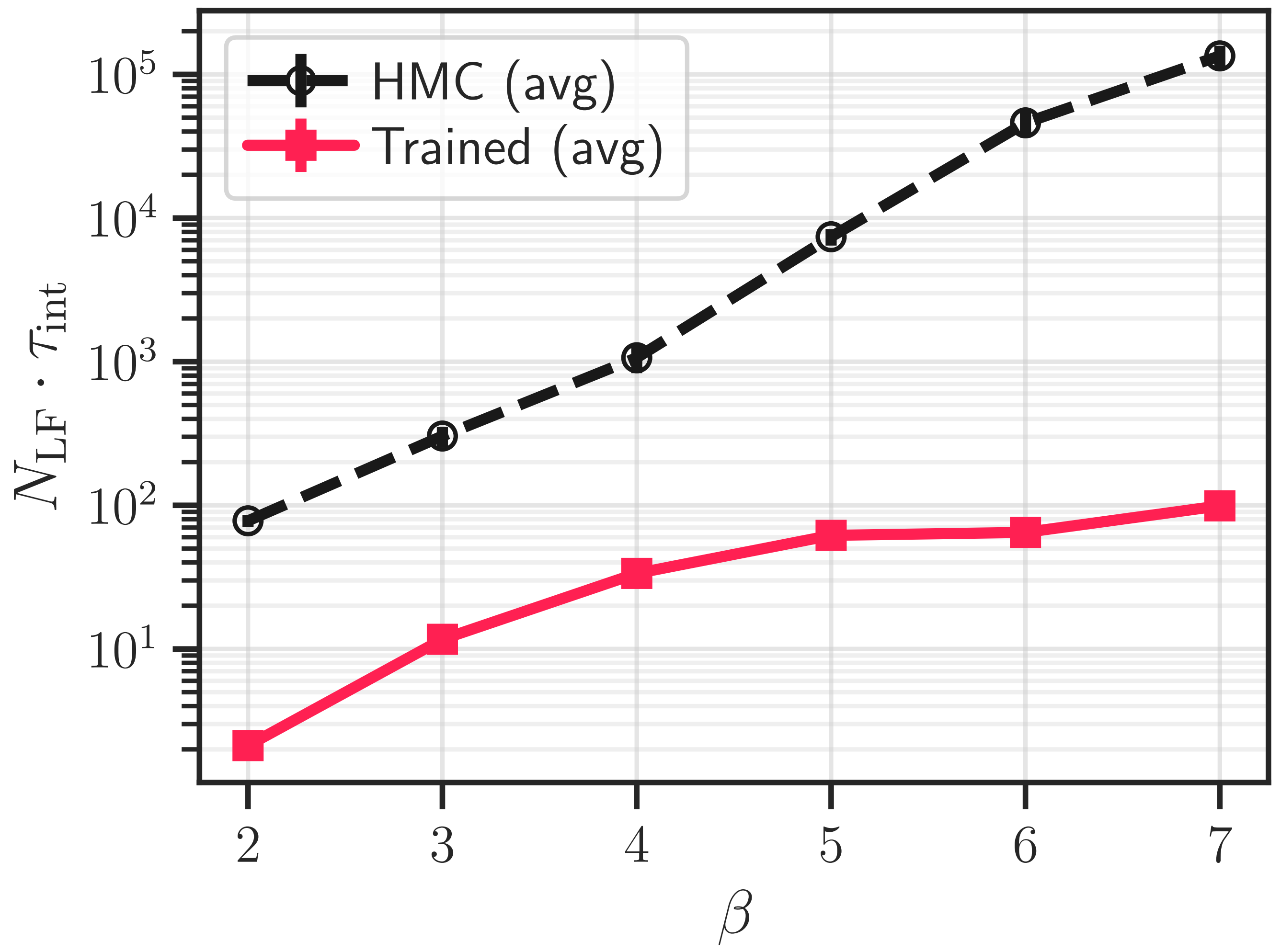
Argonne NATIONAL LABORATORY



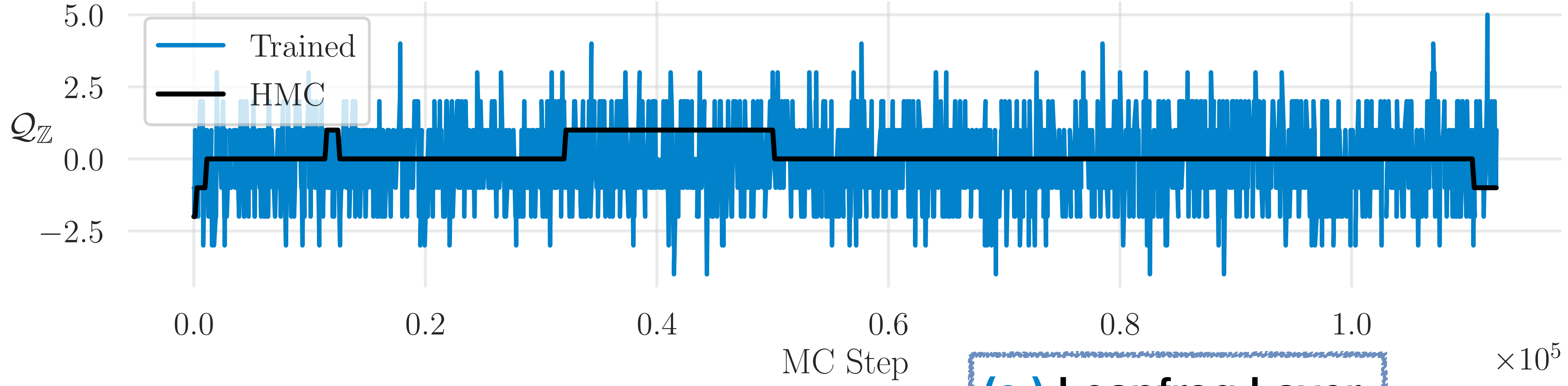
(a.) Deviation in the average plaquette, x_P over a single trajectory.



(b.) Evolution of $\mathcal{Q}_{\mathbb{R}} = \frac{1}{2\pi}\sum_P \sin x_P \in \mathbb{R}$



(c.) Estimate of the integrated autocorrelation time τ_{int} of $\mathcal{Q}_{\mathbb{Z}} \in \mathbb{Z}$



(d.) Topological charge history

(e.) Leapfrog Layer