

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](https://arxiv.org/abs/2105.03418)

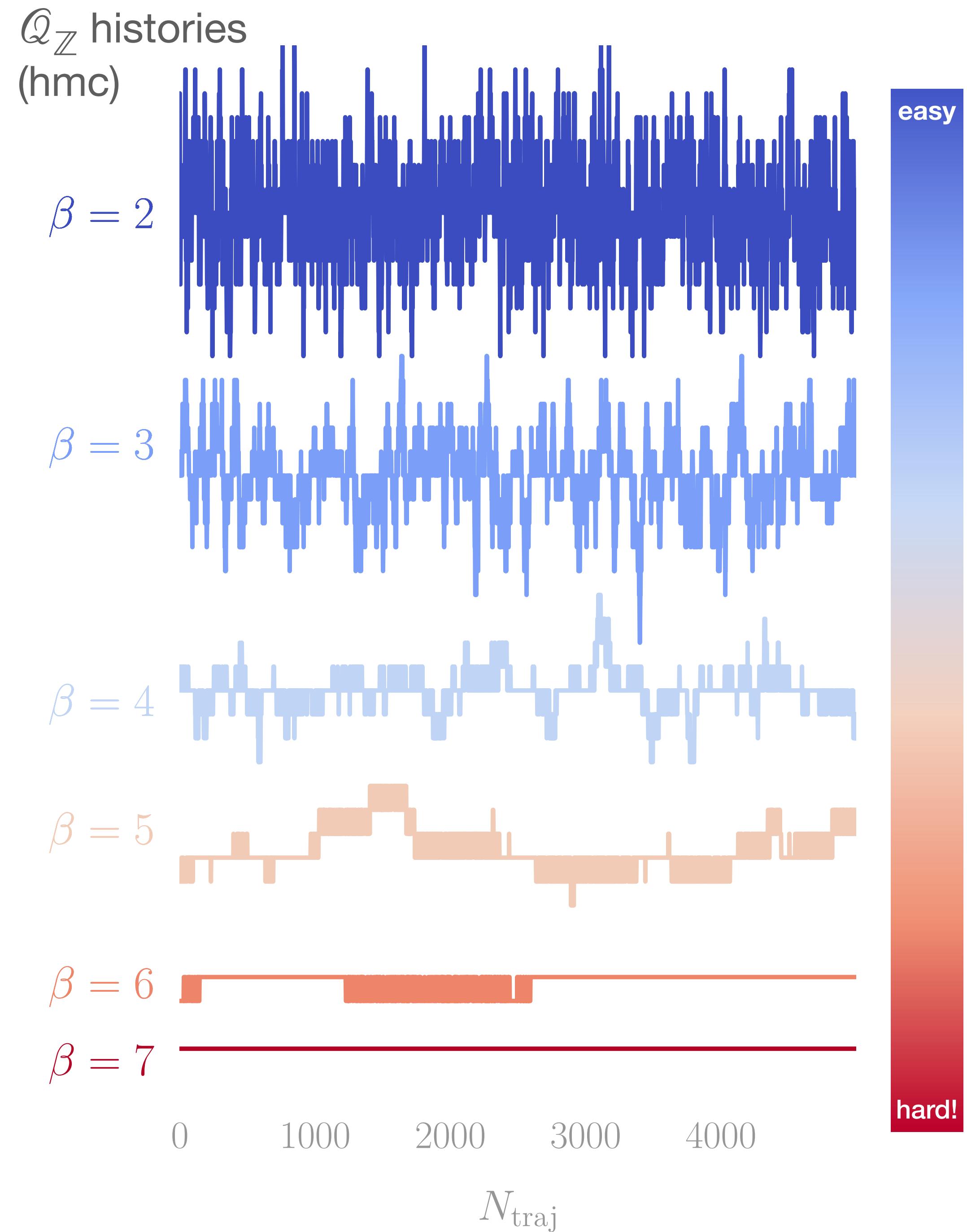
bit.ly/12hmc-lattice21

bit.ly/12hmc-surprise

github.com/saforem2/12hmc-qcd

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 $\beta \rightarrow \infty$, 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 $v \sim \mathcal{N}(0, \mathbb{I})$, 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 $H = \text{const}$:

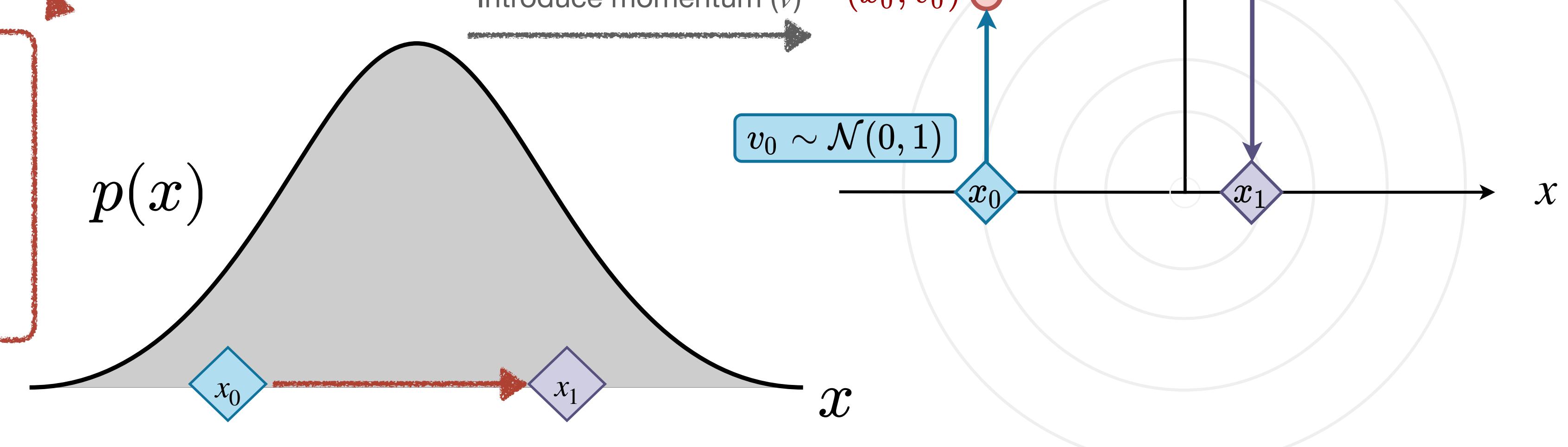
$$\dot{x} = \frac{\partial H}{\partial v}, \quad \dot{v} = \frac{\partial H}{\partial x}$$

- Leapfrog Integrator:**

```

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

```



- Accept / reject proposal x' using MH:

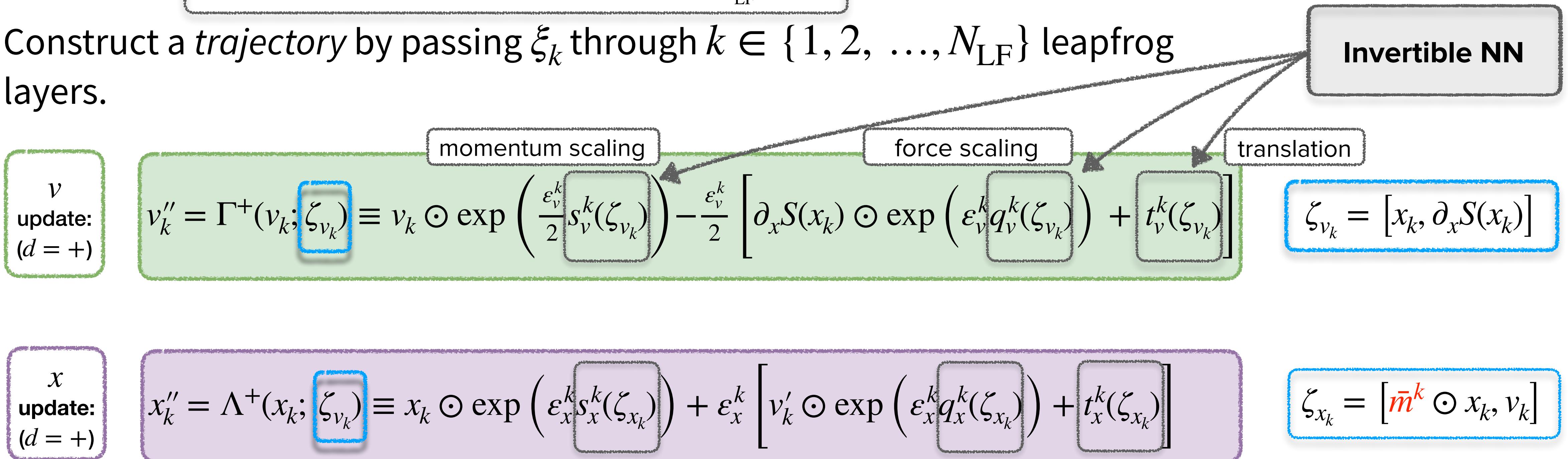
$$x_{i+1} \leftarrow \begin{cases} x' & \text{w/prob. } A(\xi' | \xi) = \min \left\{ 1, \frac{p(\xi')}{p(\xi)} \left| \frac{\partial \xi'}{\partial \xi} \right| \right\}, \\ x_i & \text{w/prob. } 1 - A(\xi' | \xi) \end{cases}$$

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) $\xi_0 \rightarrow \xi_1 \rightarrow \dots \rightarrow \xi_k \rightarrow \xi_{k+1} \rightarrow \dots \rightarrow \xi_{N_{\text{LF}}} \equiv \xi''$ (proposal)

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



I2hmc: 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:

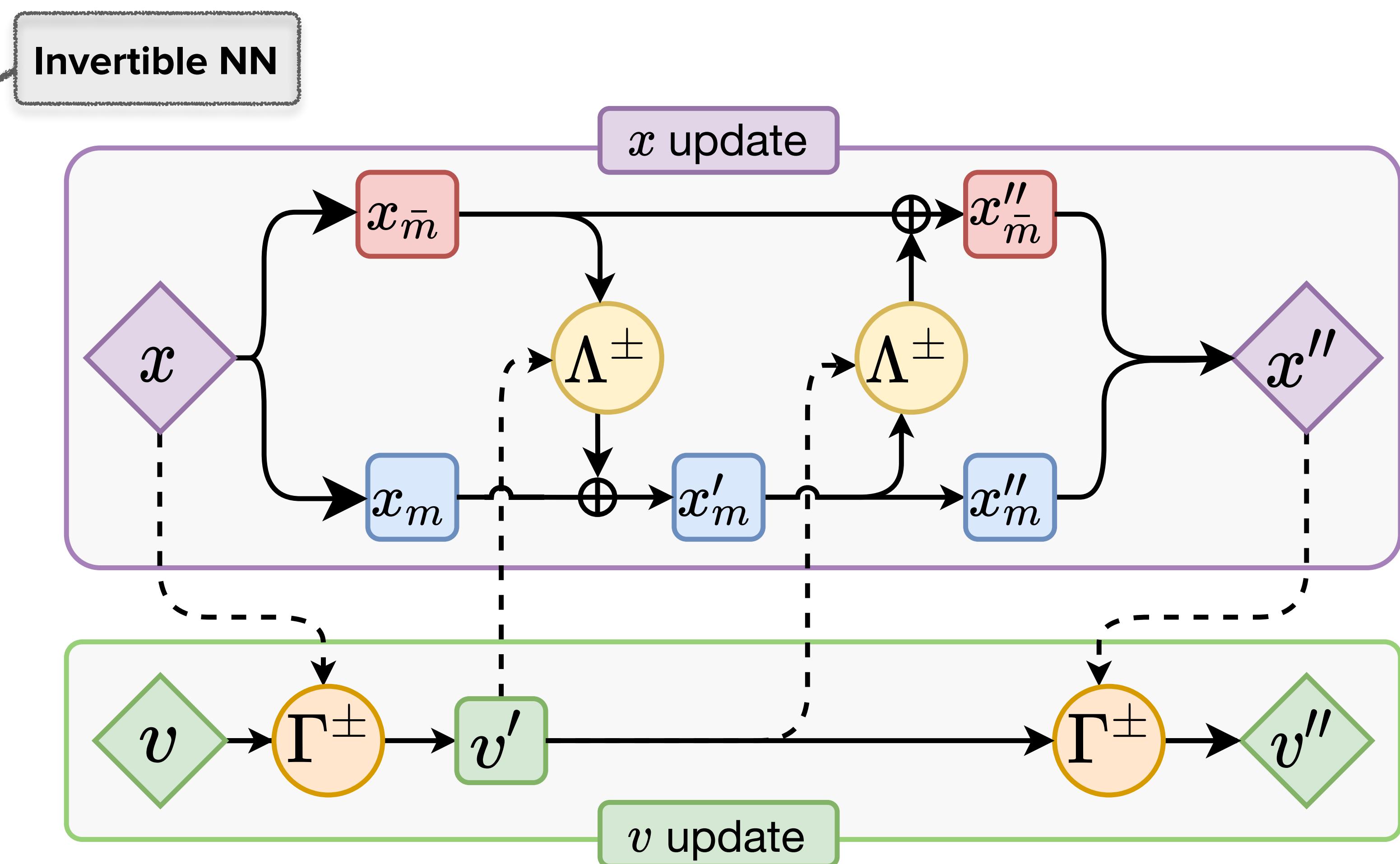
$$x'_k = m^k \odot x_k + \bar{m}^k \odot \Lambda^\pm(x_k; \zeta_{x_k})$$

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:

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



I2hmc: 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:

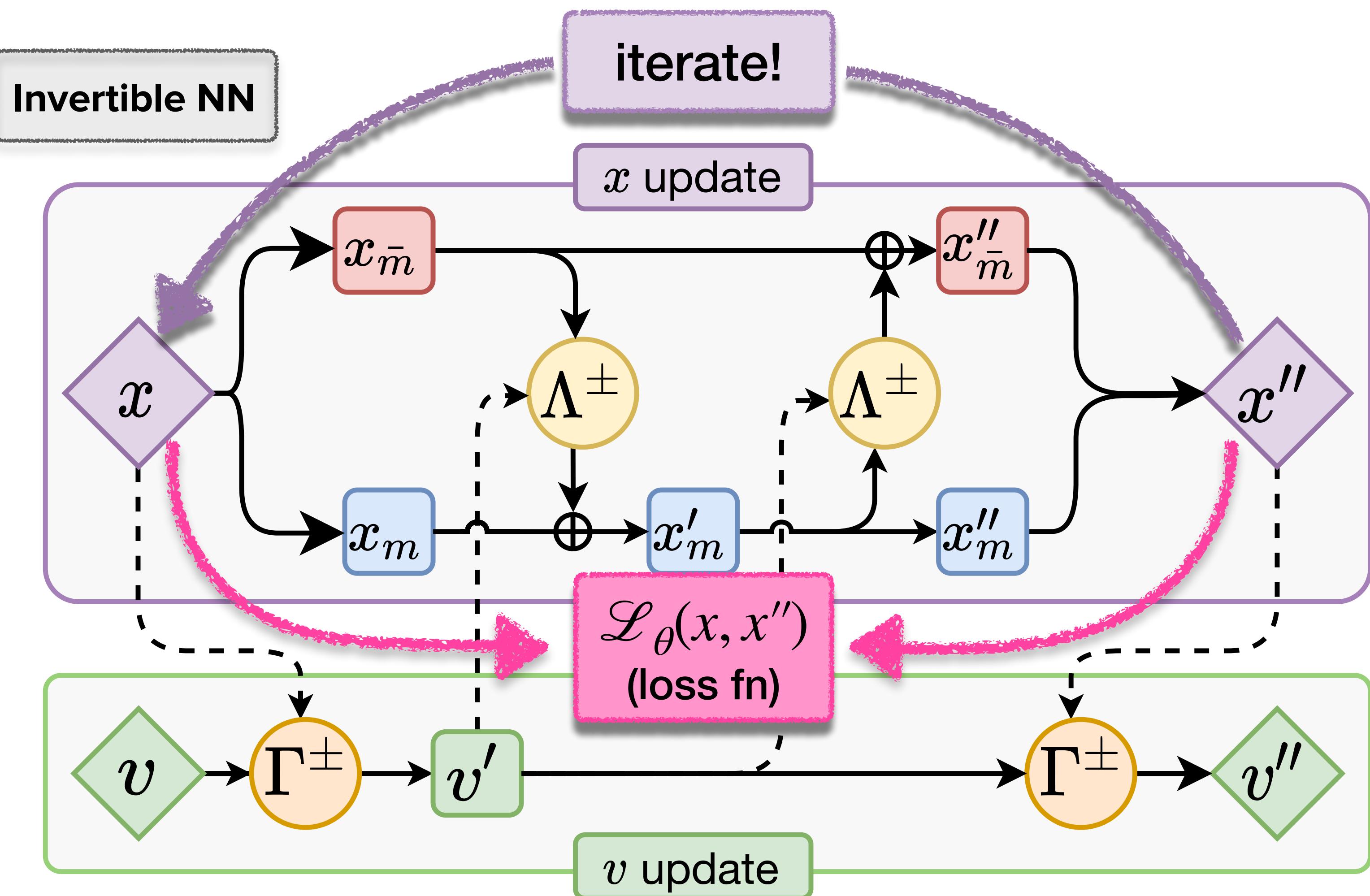
$$x'_k = m^k \odot x_k + \bar{m}^k \odot \Lambda^\pm(x_k; \zeta_{x_k})$$

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:

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



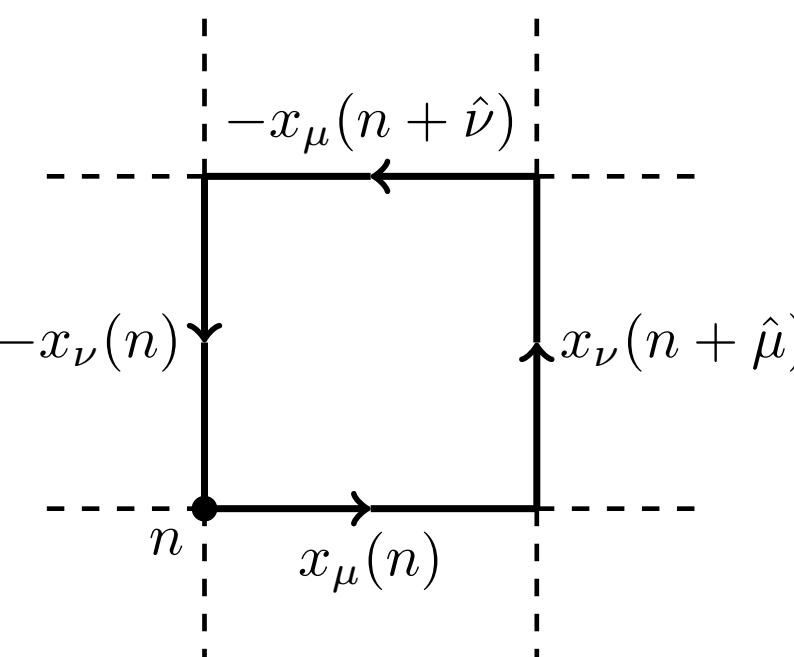
2D $U(1)$ Lattice Gauge Theory

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

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

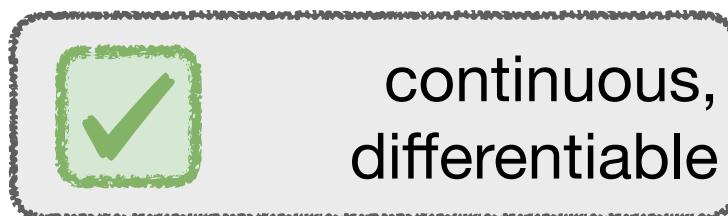
- Wilson action:**

- $S_\beta(x) = \beta \sum_P 1 - \cos x_P$,
- $x_P = x_\mu(n) + x_\nu(n + \hat{\mu}) - x_\mu(n + \hat{\nu}) - x_\nu(n)$

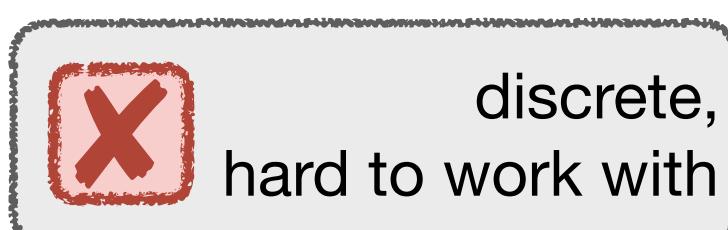


- Topological charge:**

- $\mathcal{Q}_{\mathbb{R}} = \frac{1}{2\pi} \sum_P \sin x_P \in \mathbb{R}$



- $\mathcal{Q}_{\mathbb{Z}} = \frac{1}{2\pi} \sum_P \lfloor x_P \rfloor \in \mathbb{Z}$



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

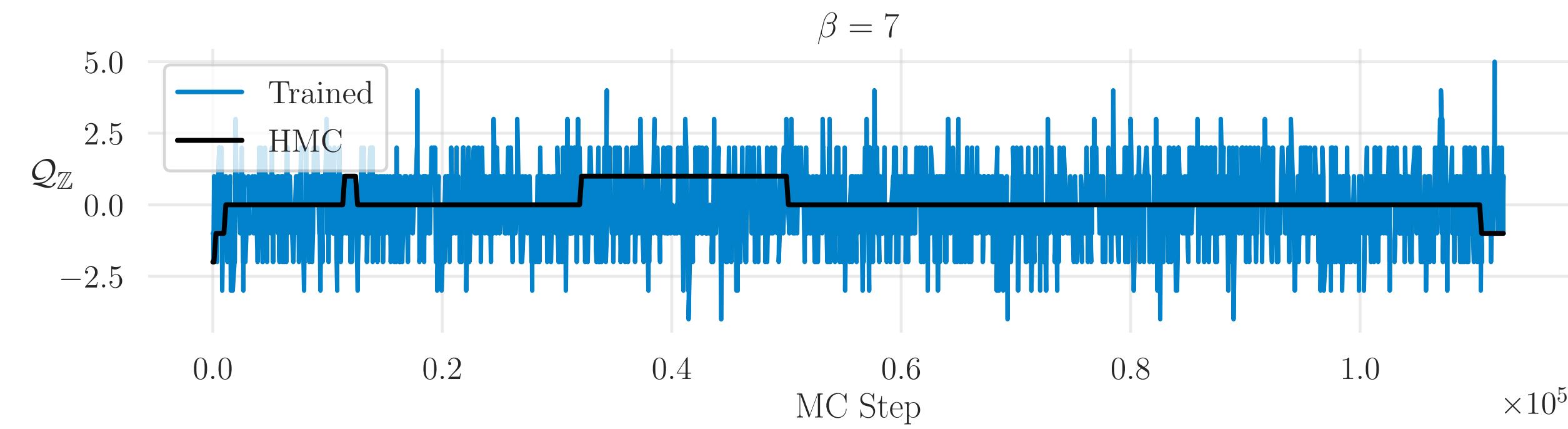
Loss function, $\mathcal{L}(\theta)$

- We maximize the *expected squared charge difference*:

- $$\mathcal{L}(\theta) = \mathbb{E}_{p(\xi)} [-\delta\mathcal{Q}_{\mathbb{R}}^2(\xi', \xi) \cdot A(\xi' | \xi)]$$

- $$\delta\mathcal{Q}_{\mathbb{R}}^2(\xi', \xi) = (\mathcal{Q}_{\mathbb{R}}(\xi') - \mathcal{Q}_{\mathbb{R}}(\xi))^2$$
 (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:

- $\{\gamma_t\}_{t=0}^N = \{\gamma_0, \gamma_1, \dots, \gamma_{N-1}, \gamma_N\}$,

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

- $\gamma_0 < \gamma_1 < \dots < \gamma_N \equiv 1$,

- increasing

- $\delta_\gamma \equiv \|\gamma_{t+1} - \gamma_t\| \ll 1$

- 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 $t = 0, 1, \dots, N$

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 $v \sim \mathcal{N}(0, \mathbb{I})$
 resample $d \sim \mathcal{U}(+, -)$
 construct $\xi_0 \equiv (x_0, v_0, d_0)$

for $0 \leq k < N_{\text{LF}}$:

 | propose (leapfrog layer) $\xi'_k \leftarrow \xi_k$

 compute $A(\xi'|\xi) = \min \left\{ 1, \frac{p(\xi')}{p(\xi)} \left| \frac{\partial \xi'}{\partial \xi^T} \right| \right\}$

 update $\mathcal{L} \leftarrow \mathcal{L}_\theta(\xi', \xi, A(\xi'|\xi))$

 backprop $\theta \leftarrow \theta - \alpha_t \nabla_\theta \mathcal{L}$

 assign $x_{t+1} \leftarrow \begin{cases} x' & \text{with probability } A(\xi'|\xi) \\ x & \text{with probability } (1 - A(\xi'|\xi)). \end{cases}$

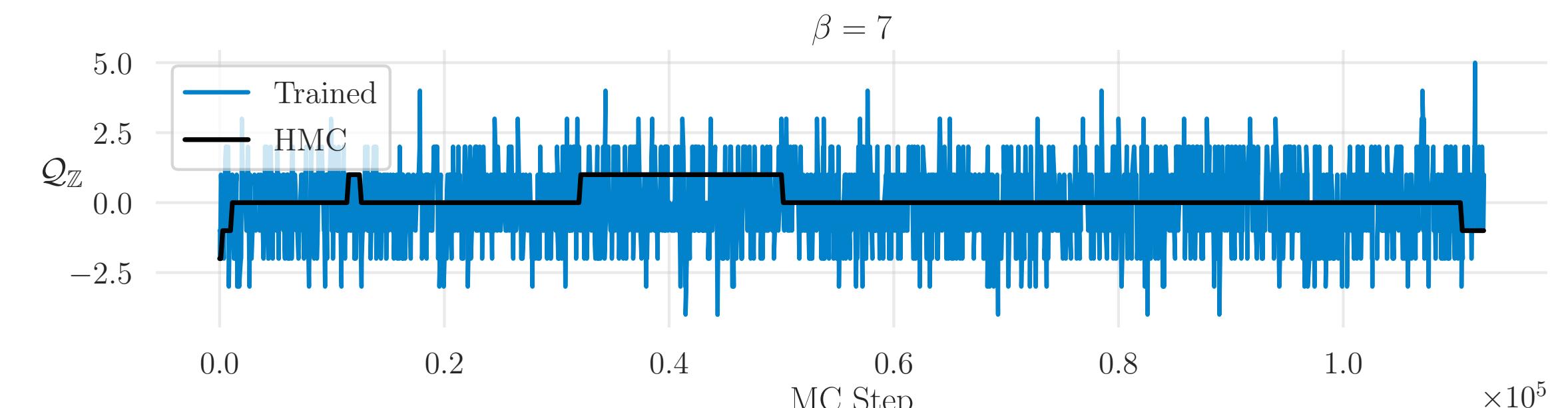
 re-sample
 momentum
 + direction
 construct
 trajectory

 Compute loss
 + backprop

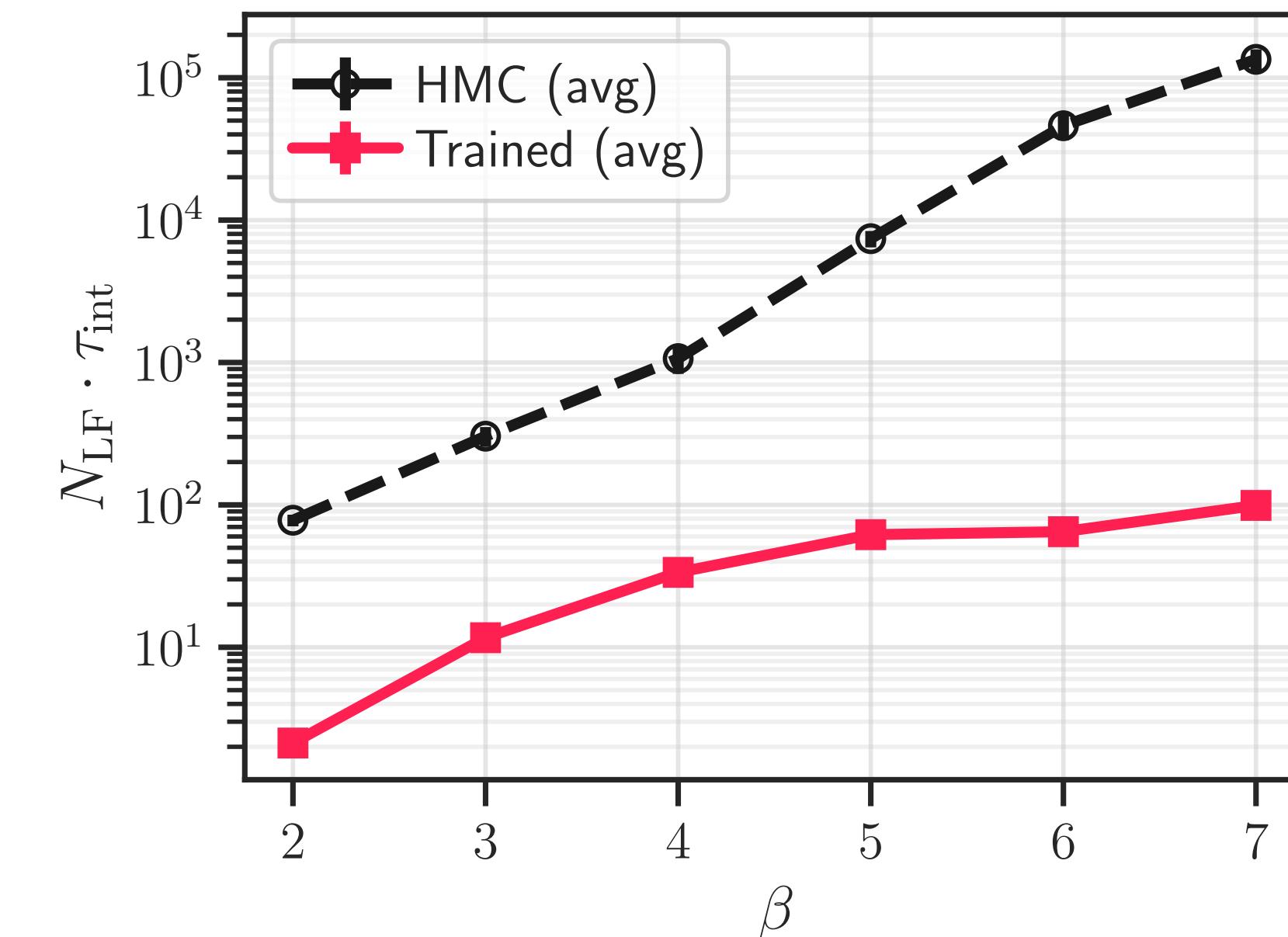
 Metropolis-Hastings
 accept/reject

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
 - $\langle \mathcal{O} \rangle \simeq \frac{1}{N} \sum_{n=1}^N \mathcal{O}(x_n) \rightarrow \sigma^2 = \frac{1}{N} \text{Var} [\mathcal{O}(x)]$
- Accounting for *autocorrelation*: $\sigma^2 = \frac{\tau_{\text{int}}^{\mathcal{O}}}{N} \text{Var} [\mathcal{O}(x)]$
- 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 $\beta \rightarrow \infty$ (**critical slowing down**)



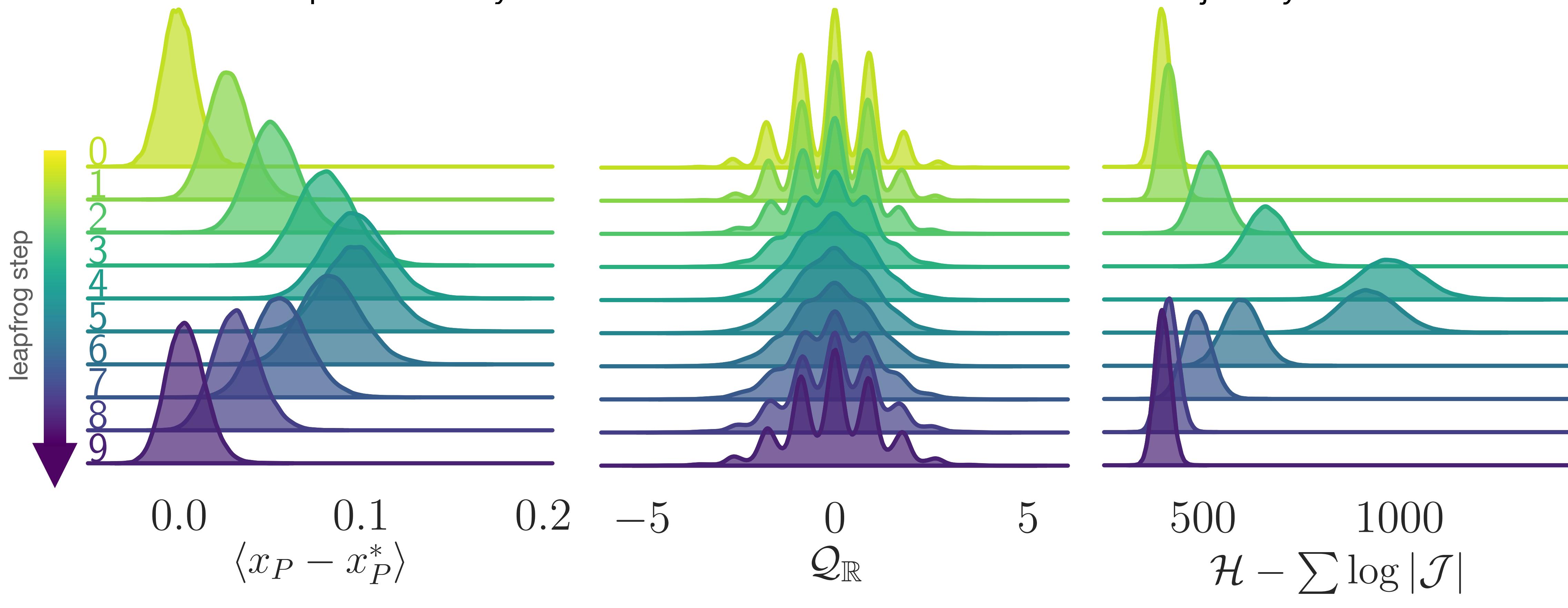
(d.) Plot of the topological charge history $\mathcal{Q}_{\mathbb{Z}}$ vs MC Step



(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

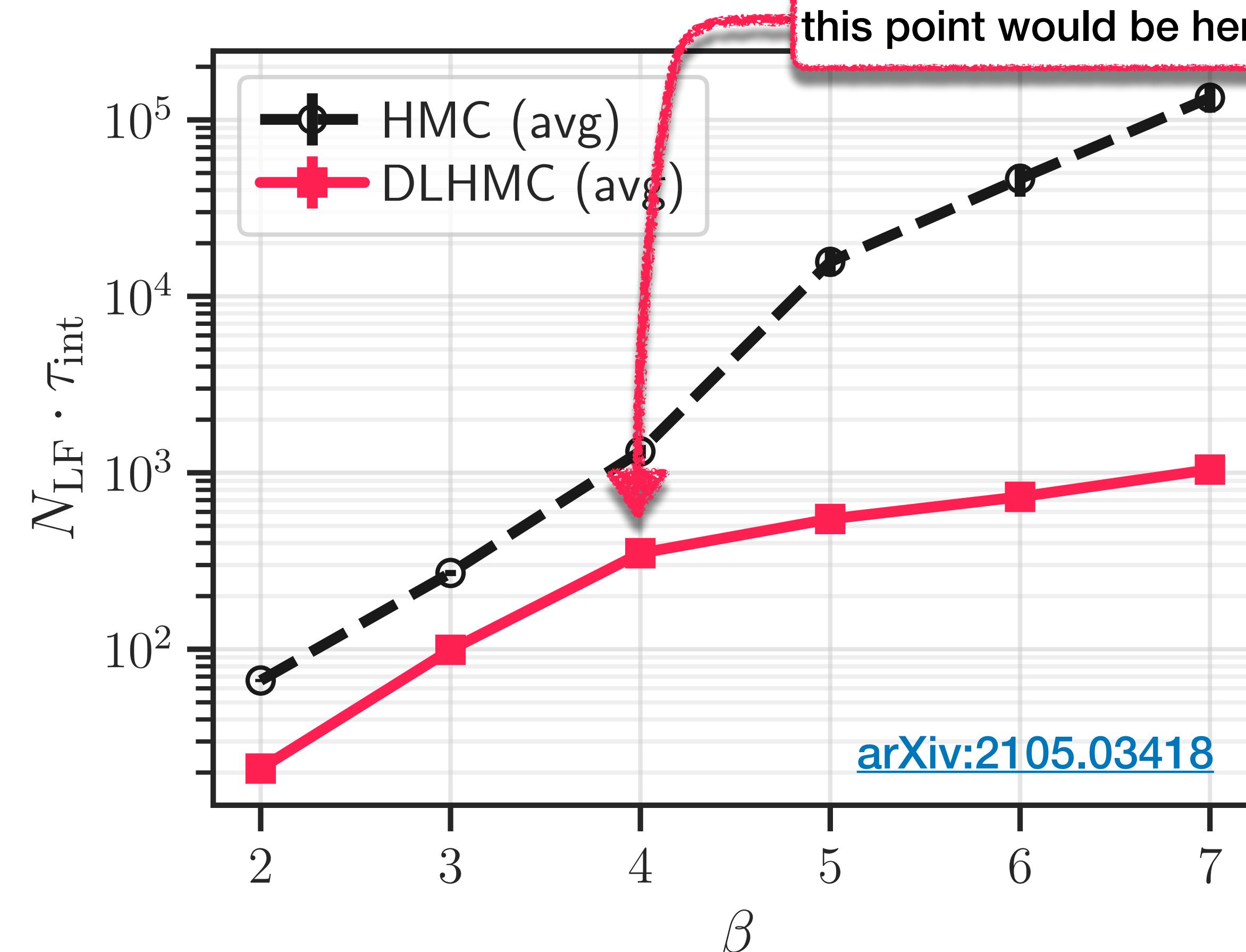


(a.) Deviation in the average plaquette, x_P

(b.) Evolution of the continuous charge Q_R

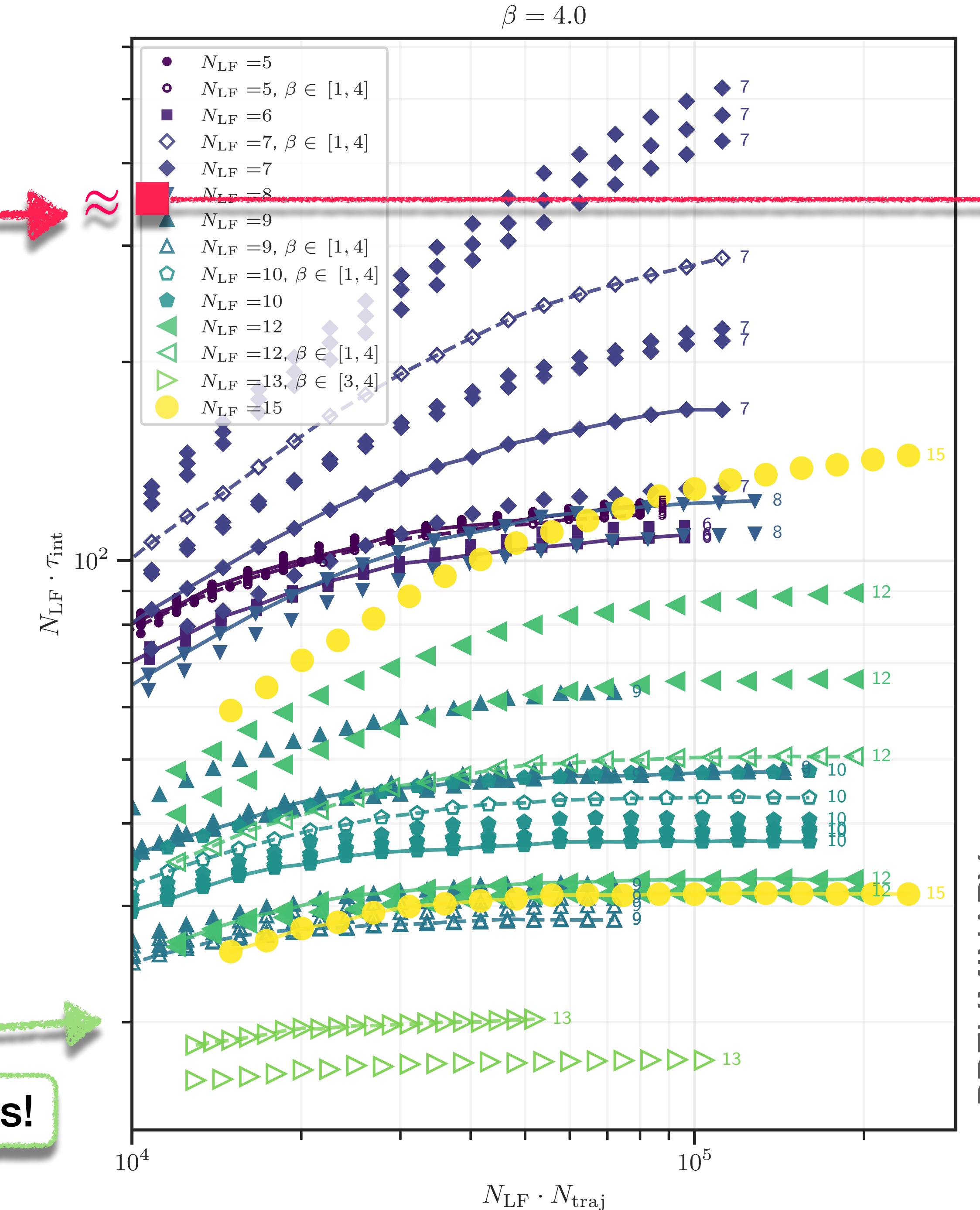
(c.) Evolution of the effective energy

New Results



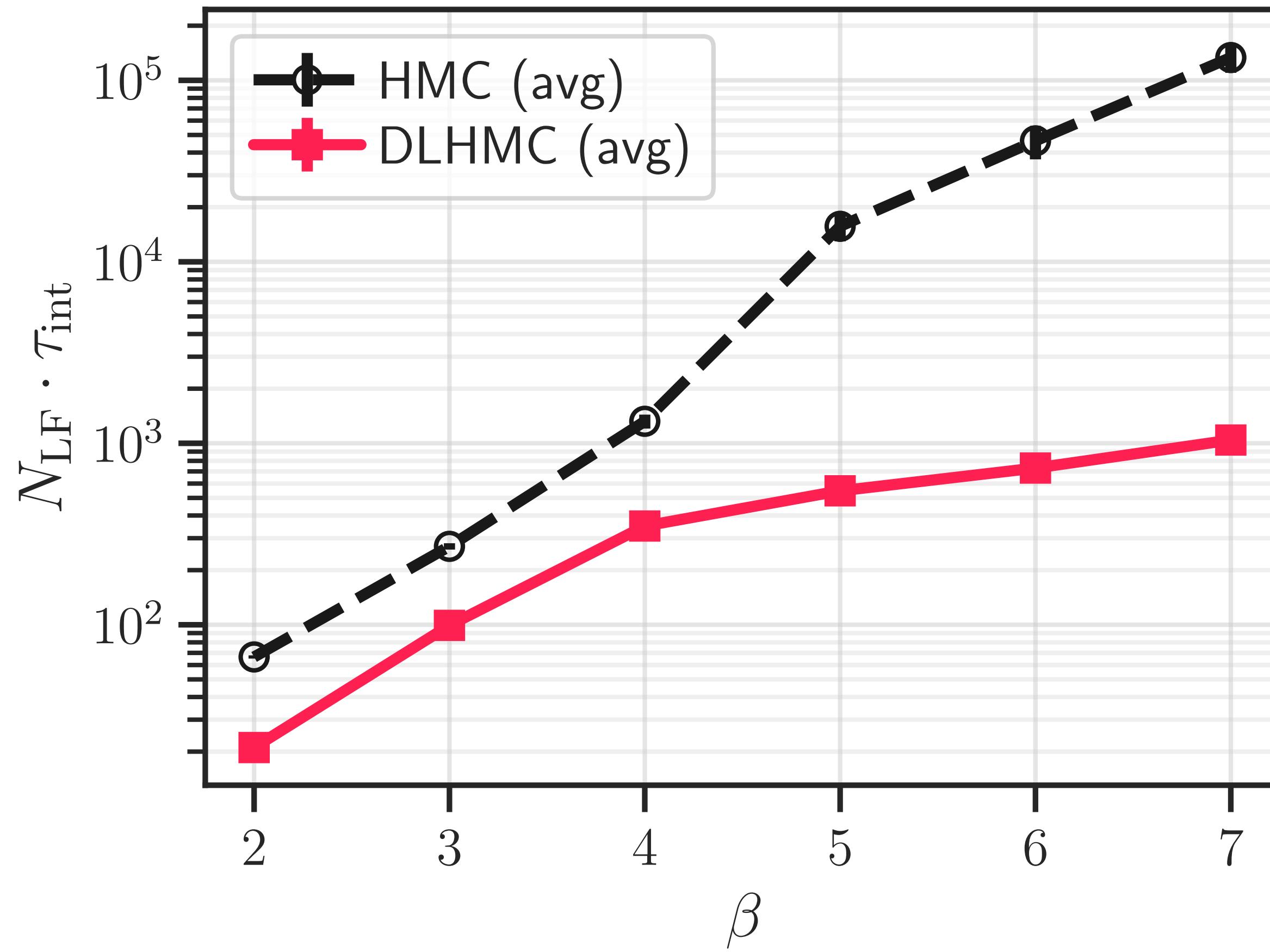
- Better performance: $\approx 10 \times$ previous results

New networks!

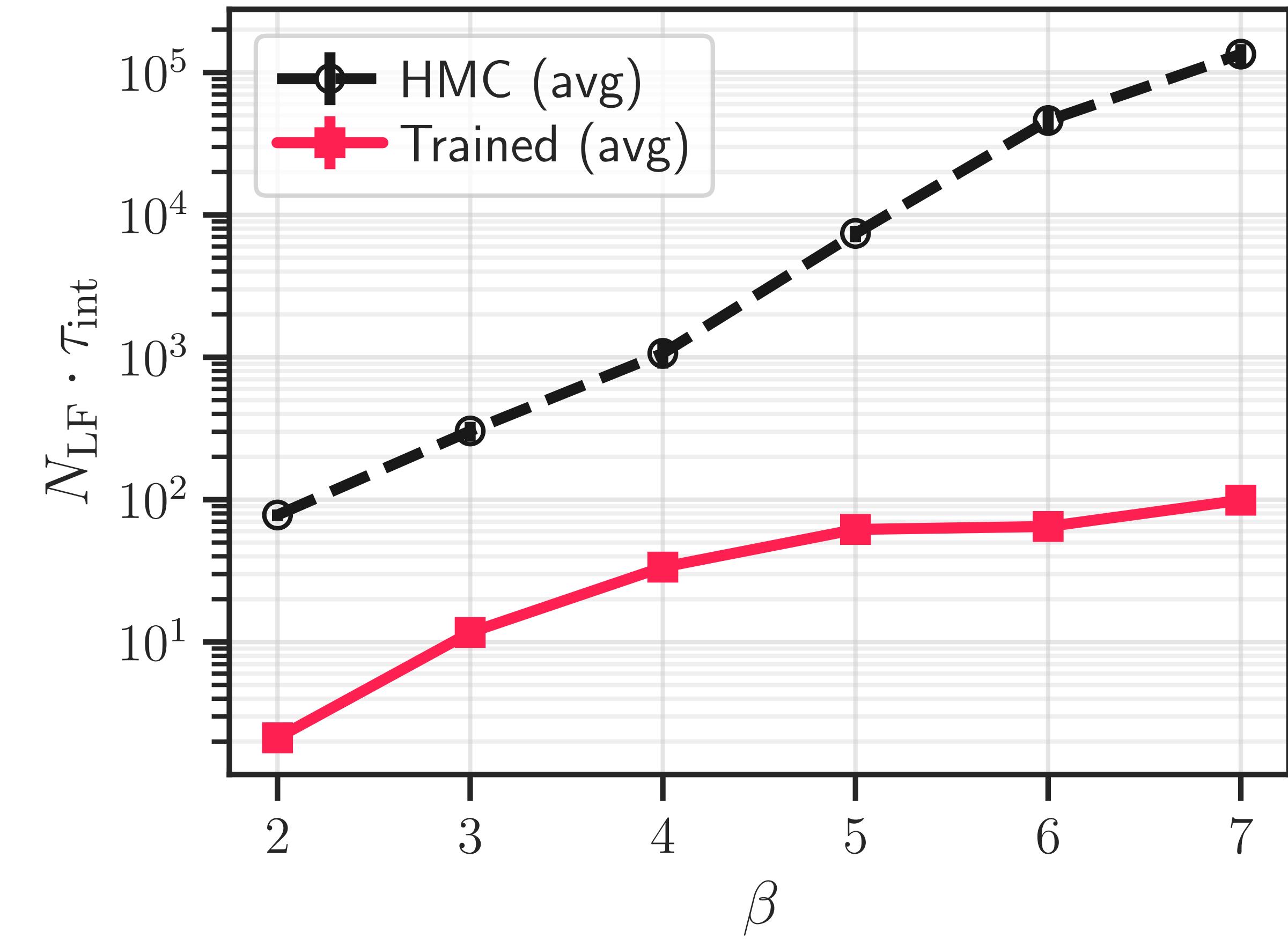


Comparison

previous (from [arXiv:2105.03418](https://arxiv.org/abs/2105.03418))



new (preliminary)



Acknowledgements

- **Collaborators:**

- ▶ Xiao-Yong Jin, ▶ James C. Osborn

- Huge thank you to:

- ▶ Yannick Meurice
- ▶ Peter Boyle
- ▶ Norman Christ
- ▶ Taku Izubuchi
- ▶ Akio Tomiya
- ▶ Critical Slowing Down group (ECP)
- ▶ Luchang Jin
- ▶ ALCF Staff + Datascience group
- ▶ Chulwoo Jung



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.



l2hmc-qcd



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

A trainable framework for accelerating HMC on lattice gauge models.

[arXiv:2105.03418](https://arxiv.org/abs/2105.03418)

bit.ly/l2hmc-lattice21

github.com/saforem2/l2hmc-qcd

