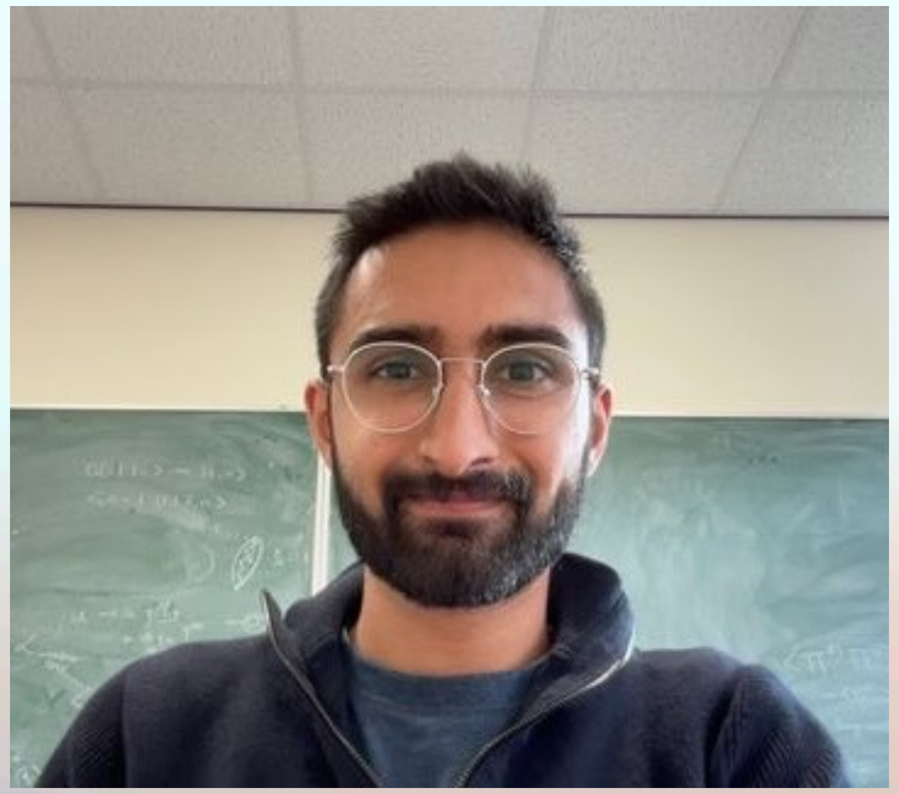
NEURAL ENHANCED OUT-OF-EQUILIBRIUM SAMPLING FOR LATTICE FIELD THEORY

ELIA CELLINI

30/07/2026
Lattice 2026
University of Maryland

In collaboration with:



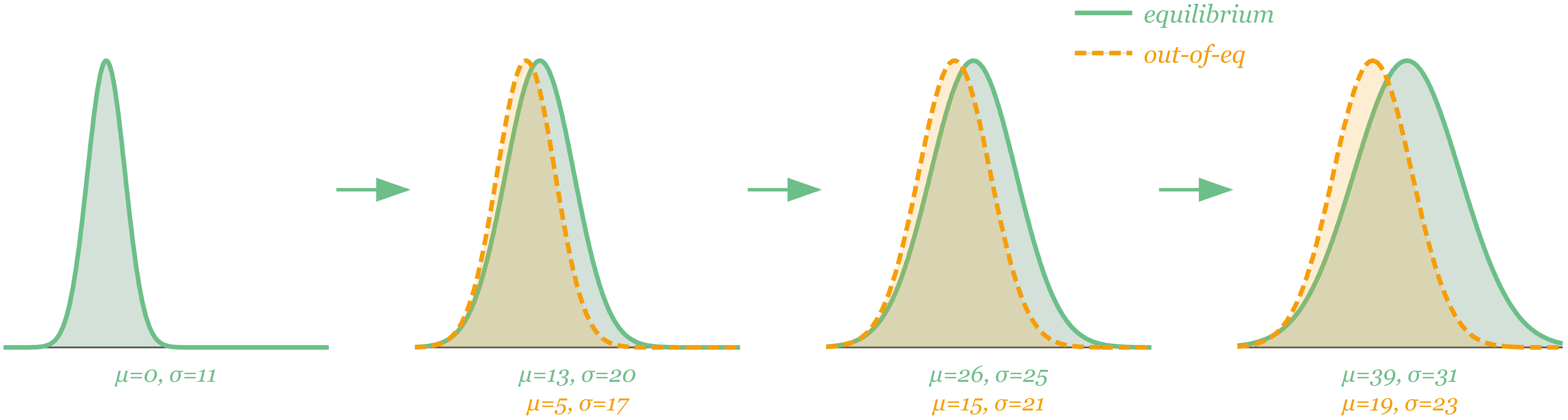
Gurtej Kanwar

Satria Widyanto

THE UNIVERSITY *of* EDINBURGH

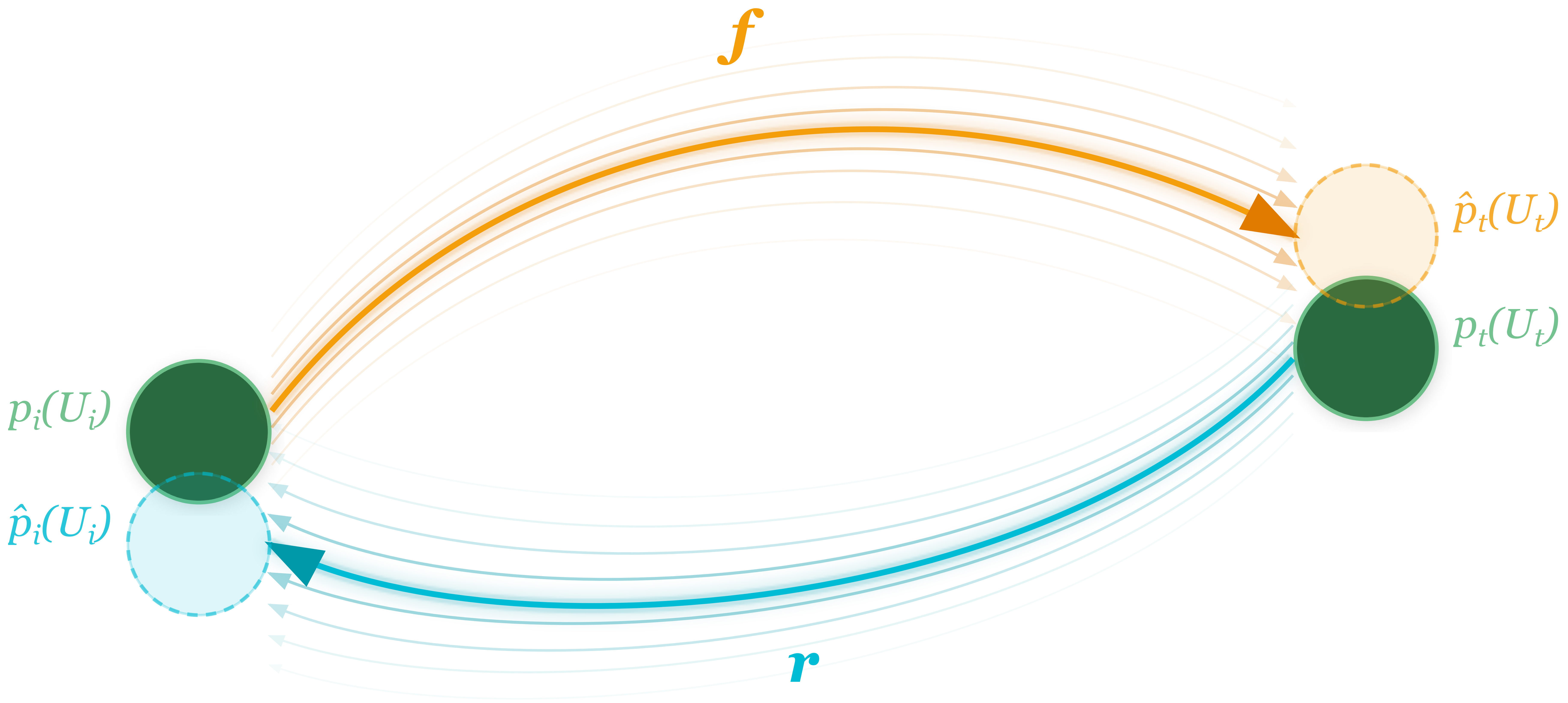
OUT-OF-EQUILIBRIUM TRANSFORMATIONS

A system is driven out of equilibrium when it is **perturbed faster than its relaxation timescale**, so it never settles into the stationary distribution of the current conditions.



TRANSIENT ENSEMBLE

Given an initial and final distributions p_i and p_t, driving the system out of equilibrium generates a **transient ensemble of stochastic trajectories**, a biased transport between the two distributions.



For each trajectory, we can compute the **thermodynamic work** done on the system: $W(U) = \int_0^T dt \, \frac{\partial S(U(t), \lambda(t))}{\partial \lambda} \dot{\lambda}(t)$

THE UNIVERSITY of EDINBURGH

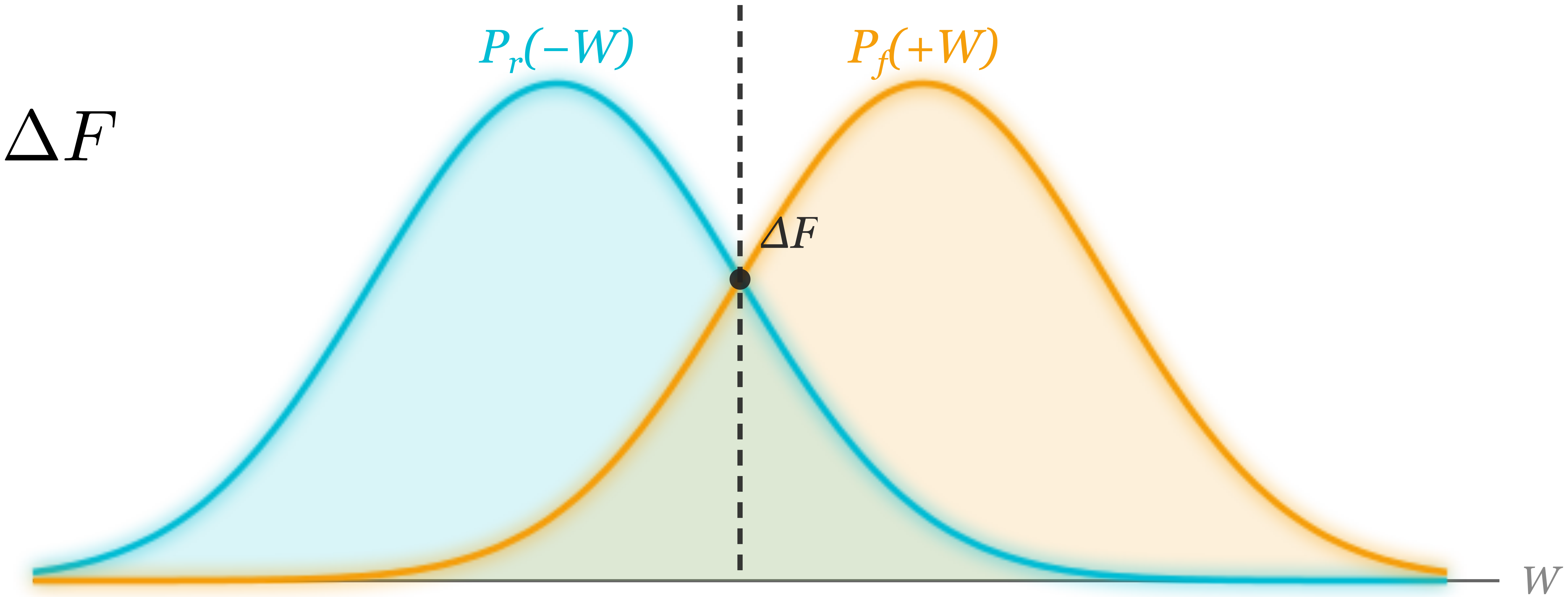
CROOKS' FLUCTUATION THEOREM

The ratio of **forward and reverse work distributions** gives direct access to the **dissipated work** W_d along each trajectory.

[Crooks; cond-mat/9901352]

$$W_d = W - \Delta F$$

$$\frac{P_f(+W)}{P_r(-W)} = e^{W-\Delta F}$$



JARZYNSKI'S EQUALITY

Jarzynski's Equality (JE) provides a powerful reweighting to **compute the free energy** and **correct the bias** of the non-equilibrium transport

[Jarzynski; cond-mat/9610209]

$$1 = \int dW\, P_f(W) \left(\frac{P_r(-W)}{P_f(W)} \right) = \int dW\, P_f(W)\, e^{-(W-\Delta F)} = e^{\Delta F} \langle e^{-W} \rangle_f$$

$$\langle e^{-W} \rangle_f = e^{-\Delta F} \qquad\qquad \langle \mathcal{O}\, e^{-W_d} \rangle_f = \langle \mathcal{O} \rangle_{p_t}$$

General identity, widely used for **computing free energies and sampling**, also in LFT.

[Neal; physics/9803008],[Caselle et al.; 1604.05544]

DRAWBACKS

The identities derived before are exact, however, the **exponential average:**

$$\langle e^{-W_d} \rangle_f$$

can be highly inefficient when W_d **is large** and **statistics are limited**.

In order to fight this problem, we want W_d to be “small”

Solution 1) **quasi-static transformations** → “small” W_d systematic but expensive scaling

Solution 2) use Machine Learning to minimize W_d → **Neural-Enhanced Out-of Equilibrium (NEO)** sampling

THE UNIVERSITY of EDINBURGH

STOCHASTIC NORMALIZING FLOWS

Stochastic Normalizing Flows (SNFs) combine **Non-Equilibrium Monte Carlo updates** and **Normalizing Flows blocks**:

$$x_0 \longrightarrow g_\theta^1(x_0) \xrightarrow{P_1} x_1 \longrightarrow g_\theta^2(x_1) \xrightarrow{P_2} \ldots \xrightarrow{P_N} x_N \equiv x$$

Where g_θ^i are NF layers and P_i are MCMC updates interpolating between prior and target

$$W_d^\theta = S_N(x_N) - S_0(x_0) - Q_\theta - \Delta F$$

$$Q_\theta = \sum_{n=0}^{N-1} \left(S_{n+1}(x_{n+1}) - S_{n+1}(x_n) + \ln|\det J_{g_\theta^n}| \right)$$

- **Power law scaling** with N
- **Linear scaling** with the **volume**
- **Improvement of the prefactor** with g_θ

[Wu et al; 2002.06707],[Caselle, EC, Nada, Panero; 2201.08862],[Bulgarelli, EC, Nada.; 2412.00200], [Bonanno, Bulgarelli, EC et al.: 2510.25704]

See **D. Hackett Tu09:45**

LOSS FUNCTION

NEO samplers can be optimized by minimizing:

$$\mathscr{L}(\theta) = \langle W_d^{\theta} \rangle_f = D_{KL}(P_f || P_r) \geq 0$$

$$\langle W \rangle_f \geq \Delta F$$

Second Law!

More reversible trajectories → smaller work → better error!

NON-EQUILIBRIUM SAMPLERS FOR LFT

JE-based samplers have been **widely investigated in lattice field theory:**

- **Free energy calculations** e.g. pressure, entropies, etc..

  [Caselle et al.; 1604.05544],[Caselle et al.; 1801.03110], [Francesconi et al.; 2003.13734], [Bulgarelli and Panero; 2304.03311, 2404.01987] [Bulgarelli et al.; 2505.20403],[Bulgarelli, EC et al.; 2410.14466]

- **Transport between distributions** e.g. Open-Periodic boundary conditions, couplings, etc…

  [Bonanno et al.; 2402.06561, 2411.00620],[Caselle, EC, Nada, Panero; 2201.08862],[Caselle, EC, Nada; 2409.15937], [Bulgarelli, EC, Nada.; 2412.00200], [Bonanno, Bulgarelli, EC et al.: 2510.25704]

  See **O. Vega Th15:00**

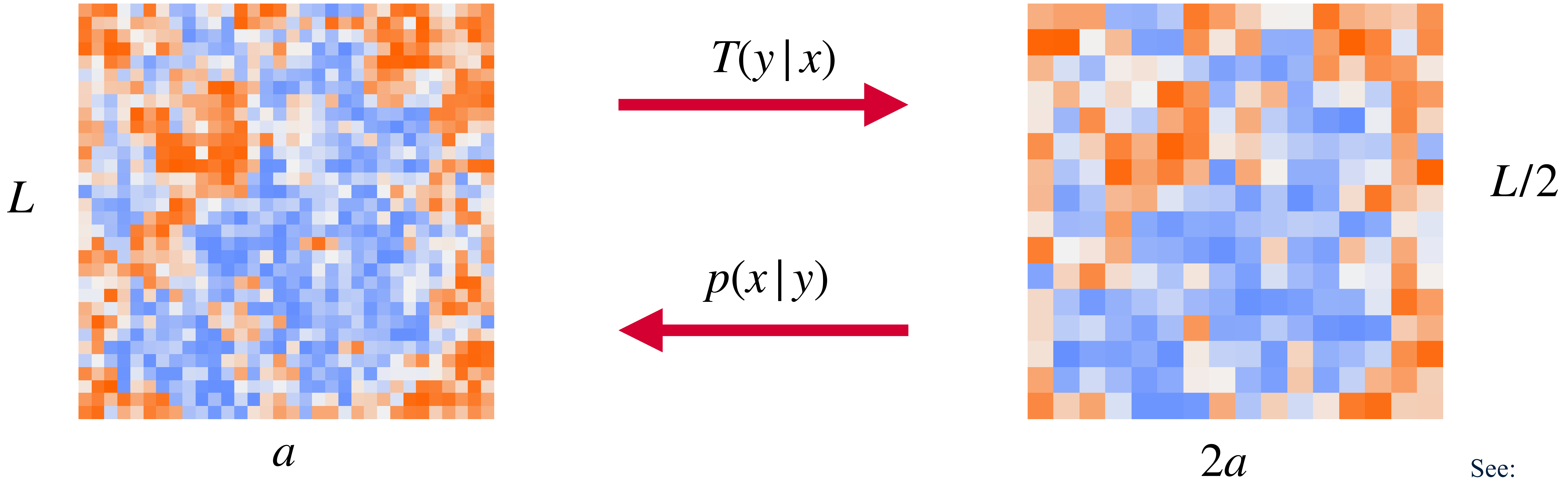
Are there **new contexts** in which Jarzynski’s equality and Crooks’ theorem can be used?

RENORMALIZATION GROUP

Consider a **target distribution** $p(x)$ and a **stochastic blocking** $T(y|x)$

Thanks to **Bayes' theorem:**

$$p(x|y) = \frac{p(x)T(y|x)}{\int dx' p(x')T(y|x')} = \frac{p(x,y)}{p(y)}$$



Growing interest in ML for LFT: learn samplers for $p(x|y)$

[Bachtis et al; 2107.00466][Bauer et al.; 2412.12842],[Singha, EC et al.; 2503.08918],[Singha, Kauffmann, EC et al.; 2604.10209]

See:

- **A. Singha M16:10**
- **L. Parato Th15:20**
- **A. Hasenfratz Poster**

BRIDGING RG AND STOCHASTIC THERMODYNAMICS

Building **a non-equilibrium transport** on the joint distribution, **at fixed y**:

$$p(x,y)^t = p(x)^t T(y\,|\,x)^t; \quad t : 0 \to 1$$

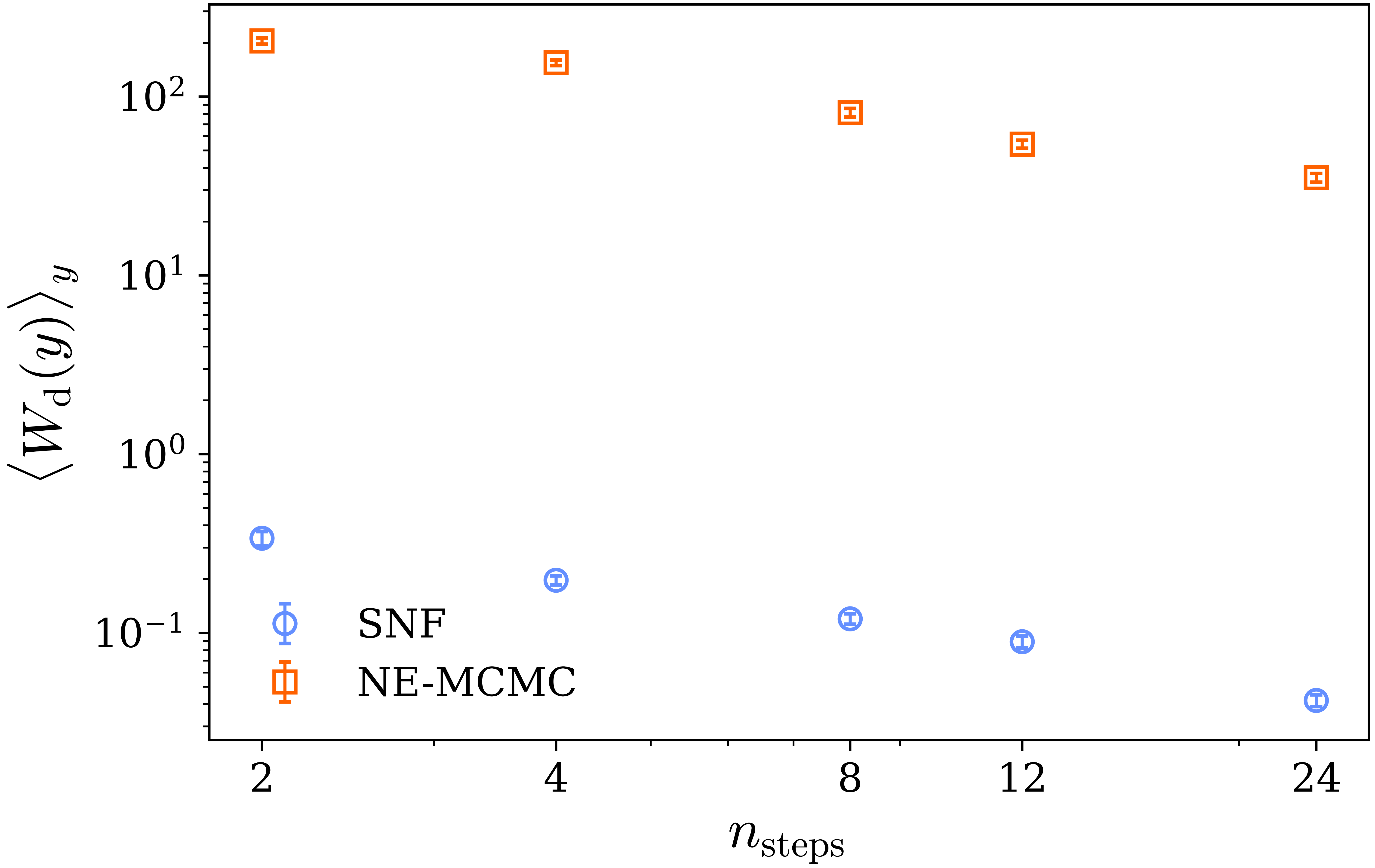
Allows to **compute the exact normalization of** $p(x\,|\,y)$ **i.e** $p(y)$ **using the Jarzynksi's equality:**

$$\langle e^{-W_{0\to 1}(x,y)} \rangle_{f_x}(y) = \frac{p(y)}{C}$$

$p(y)$ **can then be used to build a hierarchical multiscale sampler**

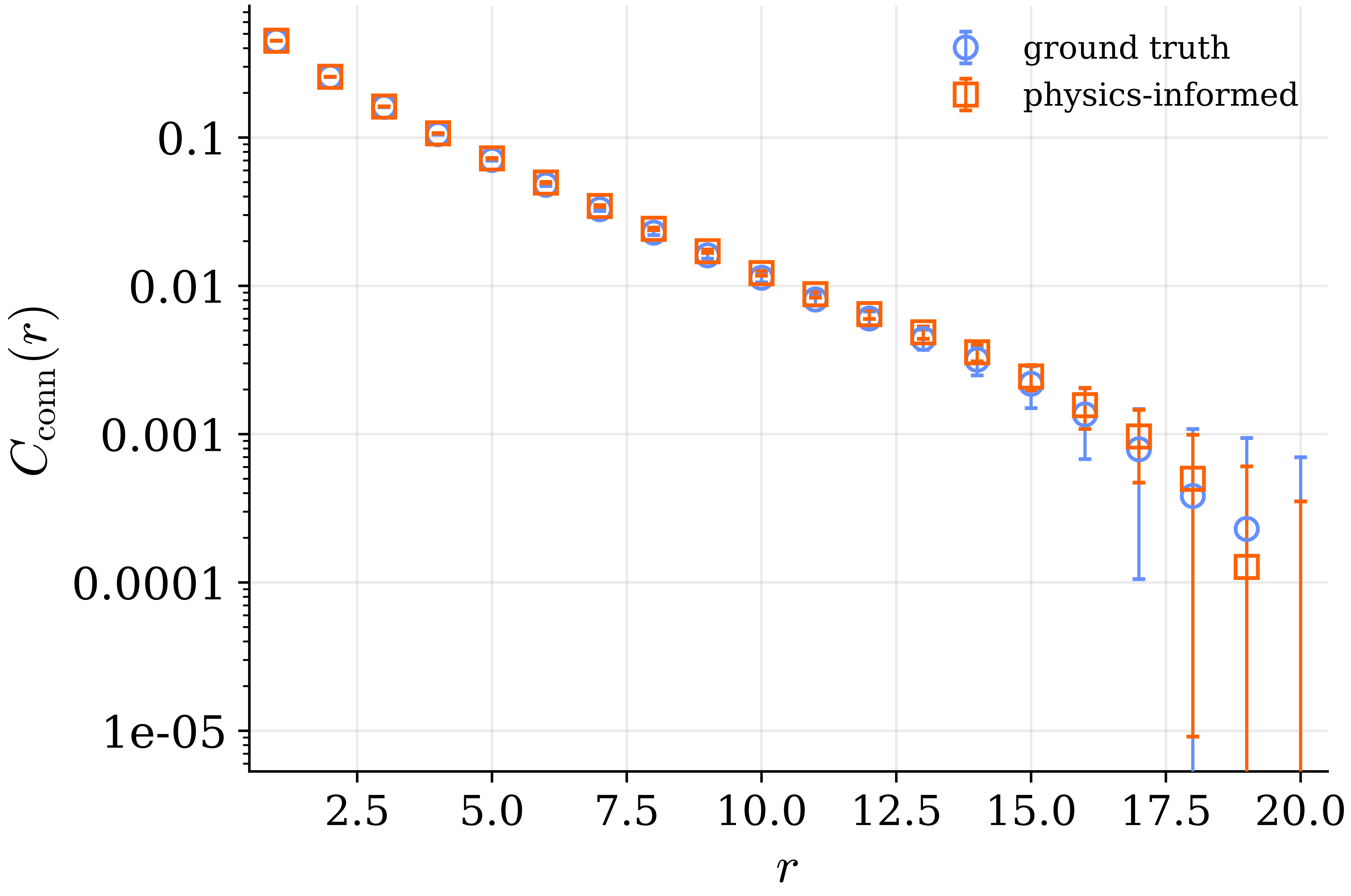
LEARNING SNF FOR $p(x,y)^t$

- Results for **O(3) non linear sigma model,** trained on 32×32 lattices, $\beta = 1.4$:
- We used an **SNF** interleaving **HMC** and physics-informed **continuous NF**.
- We then trained the models on a **synthetic** y **distribution → no target samples for training**
- Training is done **on thermalised configurations** obtained with **Parallel Tempering**



LEARNING $p(y)$

- **Physics-informed NN** for $-\log q_\theta(y)$
- **Score matching** training:
  $\mathcal{L}(\theta) = \left(\nabla_y \log \hat{p}(y) - \nabla_y \log q_\theta(y)\right)^2$
- Training done on **synthetic coarse 16×16 lattices**
- **Evaluated on coarse 64×64 lattices**



[Favoni et al.; 2012.12901][Backfried; Master Thesis]

SAMPLING $p(x)$

We found two main ways to sample $p(x\,|\,y)$:

Biased but really cheap:

Sample a set of $y \sim q_\theta(y) \simeq p(y)$ and then run HMC on:

$$p(x\,|\,y) = \frac{p(x, y)}{p(y)}$$

Unbiased:

Sample a set of $y \sim q_\theta(y) \simeq p(y)$, then run the SNFs on:

$$p(x, y)^t = p(x)^t T(y\,|\,x)^t; \quad t : 0 \to 1$$

Then reweigh with:

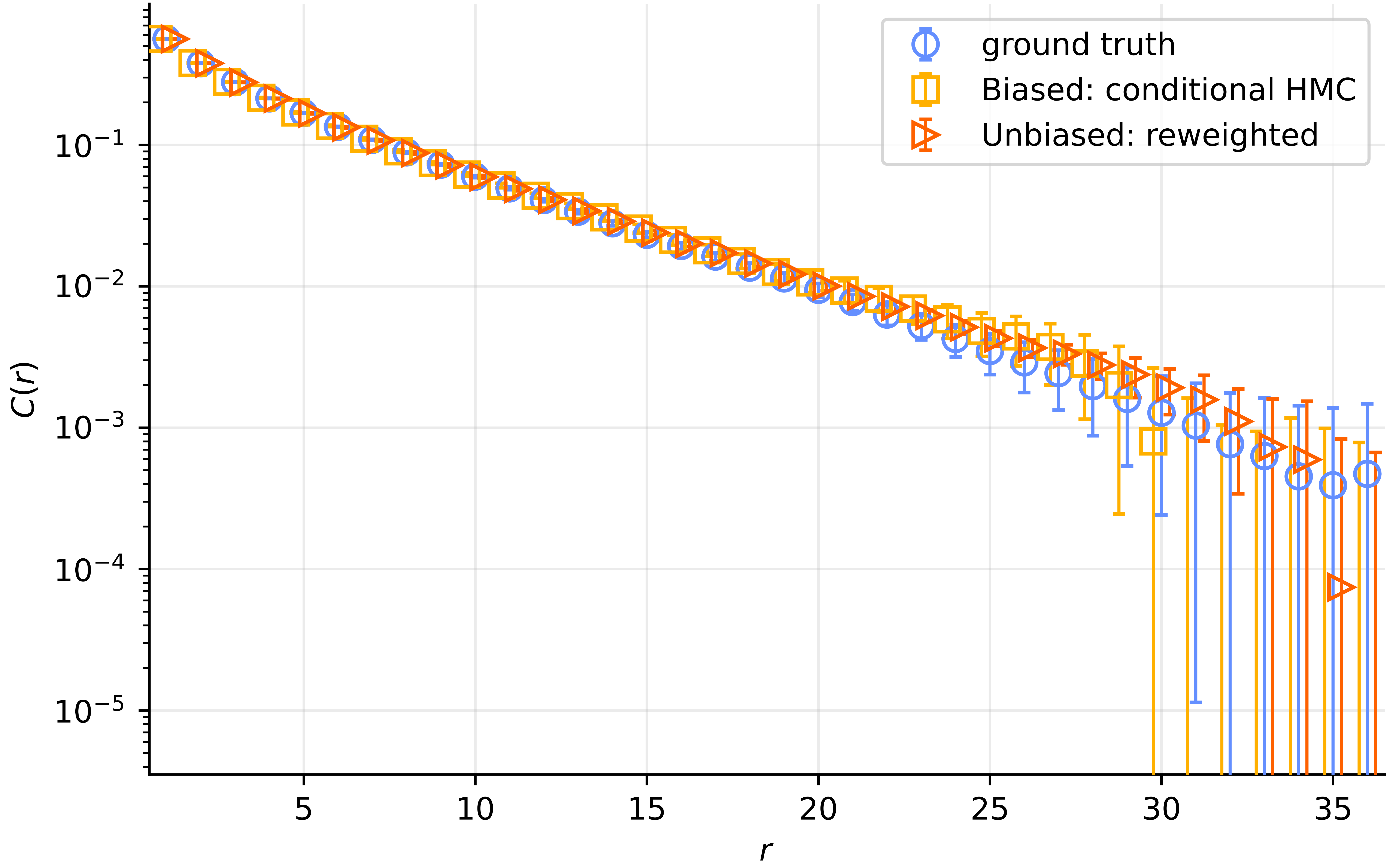
$$w = \frac{e^{-W(x,y)}}{q_\theta(y)}$$

Thus:

$$\langle \mathcal{O} \rangle_{p(x)} = \frac{\langle \mathcal{O} w \rangle_{f_x, y}}{\langle w \rangle_{f_x, y}}$$

SAMPLING $p(x)$

- Sample 128×128 fine lattices from coarser machine-learned 64×64
- **Remark:** Models have been **trained on synthetic fine 32×32 lattices**
- Upscaling works because $q_\theta(y)$ **is volume independent**, and **SNFs work only on UV fluctuations**



CONCLUSIONS AND OUTLOOKS

- We introduced **Crooks' theorem and Jarzynki's equality** and outlined **SNF**, a class of NEO samplers that display **great scalability with the capacity and the volume**
- We introduced a new way of using Jarzynski's, **bridging Renormalization Group with Stochastic Thermodynamics through Bayes's theorem**
- We introduced a **scalable** single scale **"inverse" RG sampler**

- Complete the **single-scale inverse RG** samplers for $\mathbf{CP}^{n-1}$
- **Multi-hit approaches** leveraging different scales
- **Multi-scale** samplers for $\mathbf{CP}^{n-1}$
- **Single-scale** sampler for **SU**(3)

III MACHINE LEARNING APPROACHES IN LATTICE QCD - AN INTERDISCIPLINARY EXCHANGE

May 2027

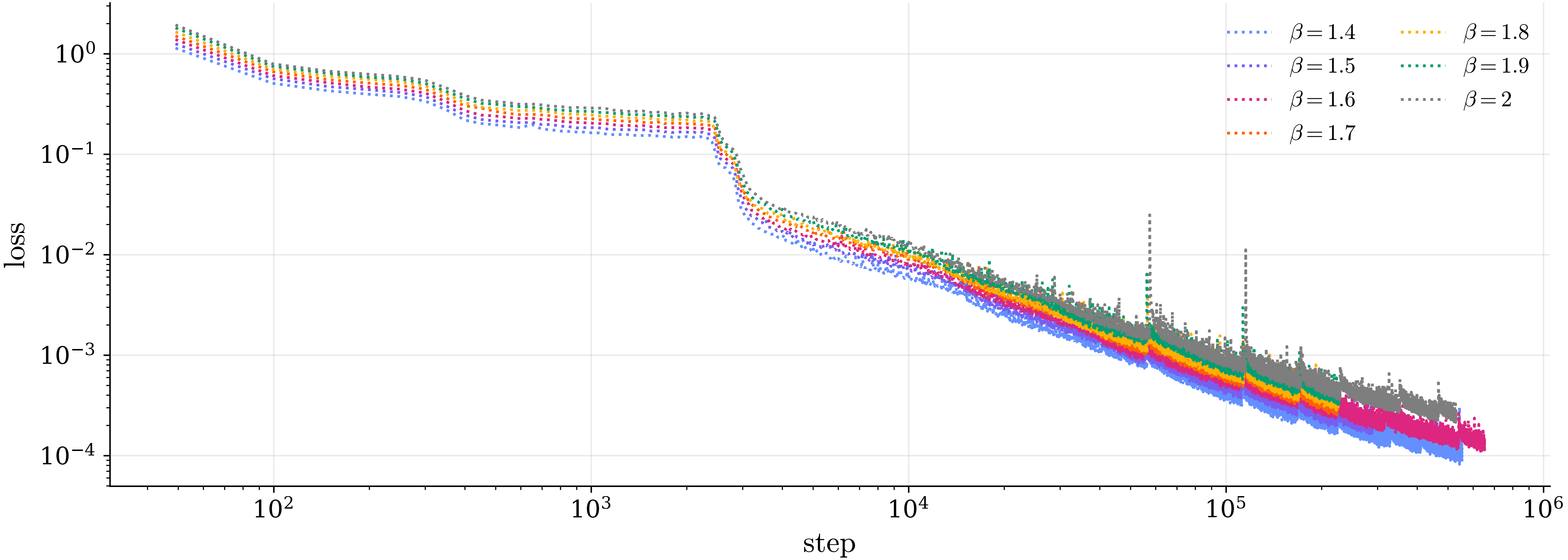
Higgs Centre for Theoretical Physics

University of Edinburgh

THANK YOU

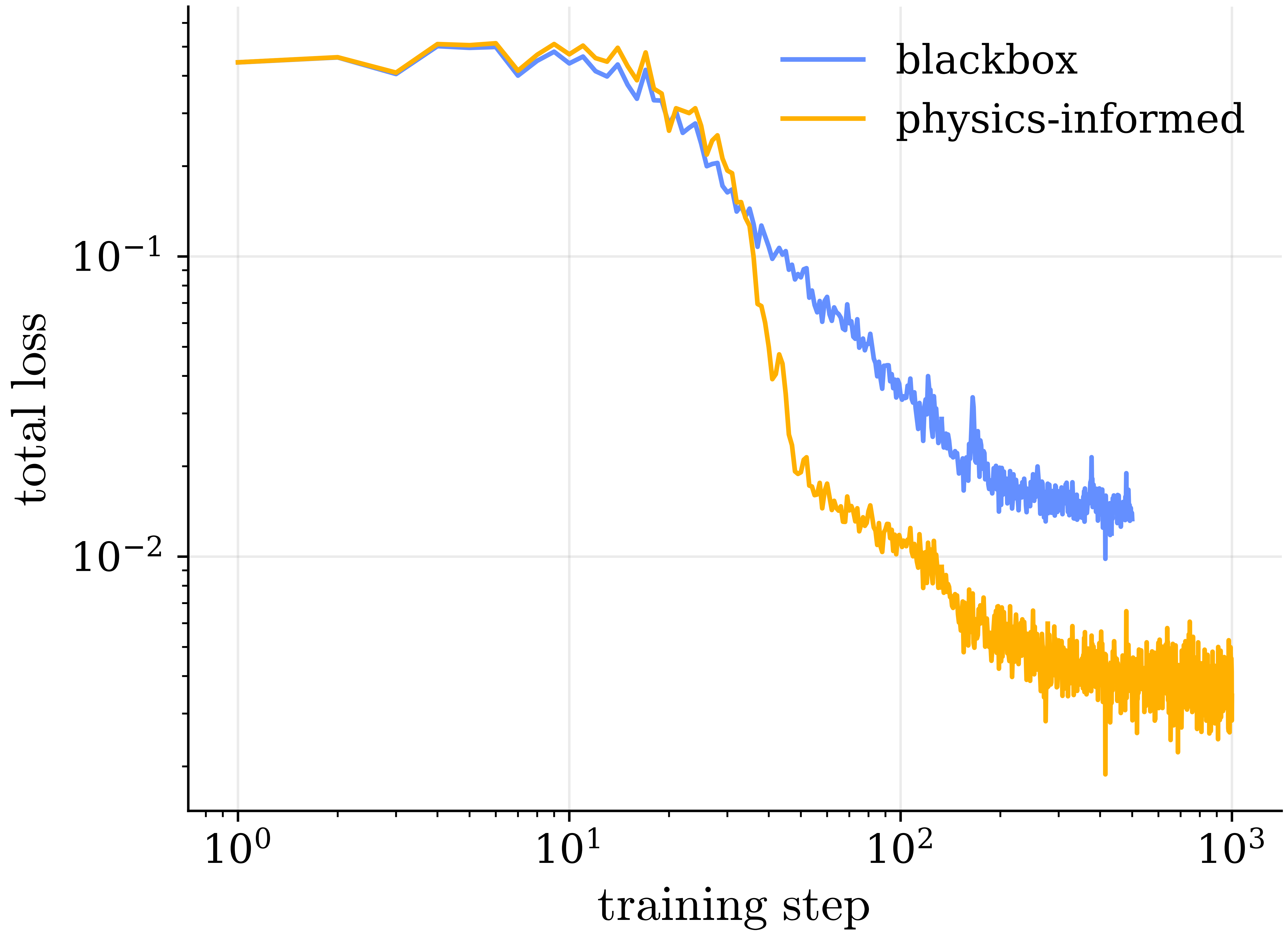
BACKUP

LEARNING SNF FOR $p(x, y)^t$: DIFFERENT β



THE UNIVERSITY *of* EDINBURGH

LEARNING $p(y)$

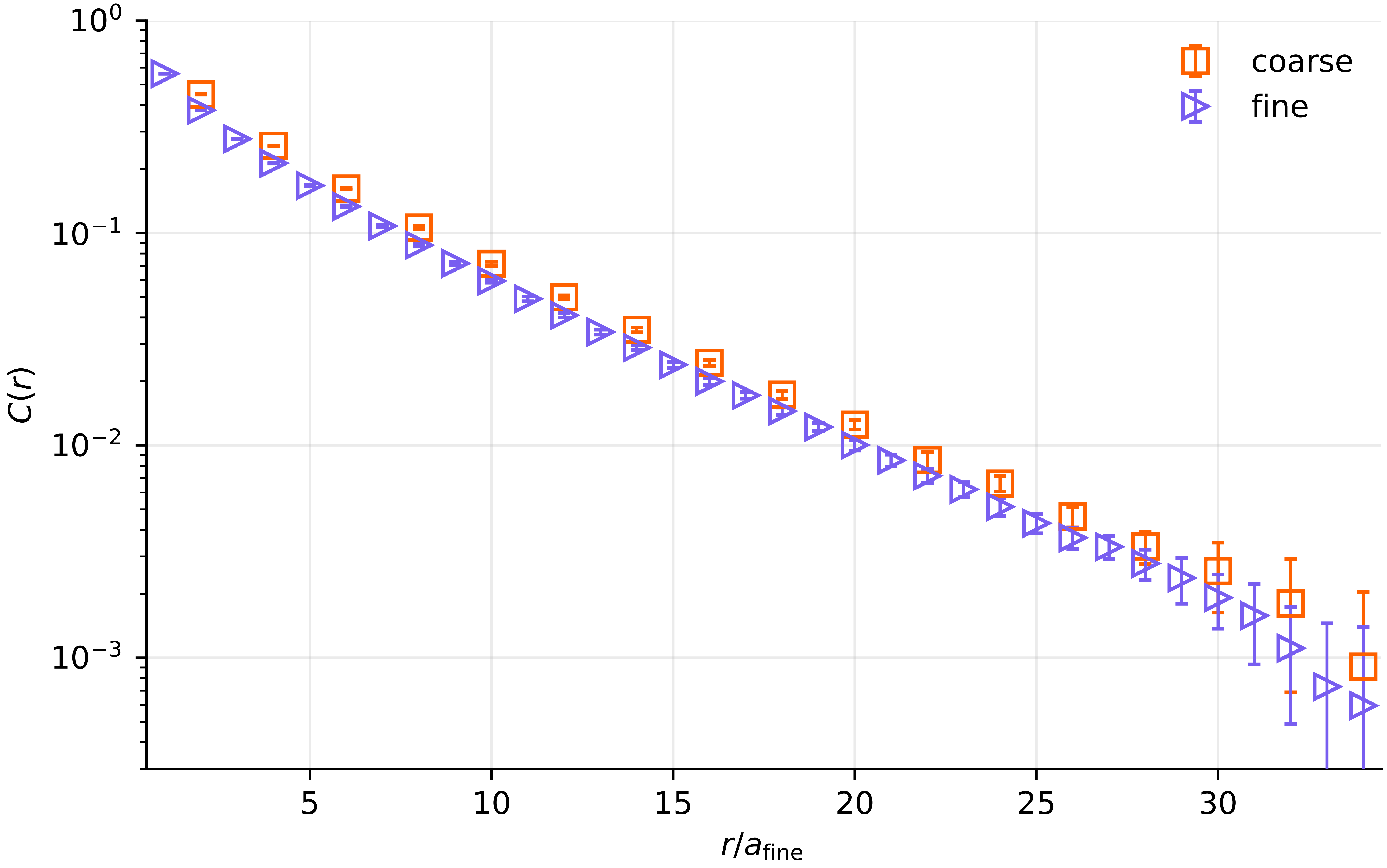


THE UNIVERSITY *of* EDINBURGH

[Favoni et al.; 2012.12901][Backfried; Master Thesis]

FINE VS COARSE CORRELATION LENGTHS

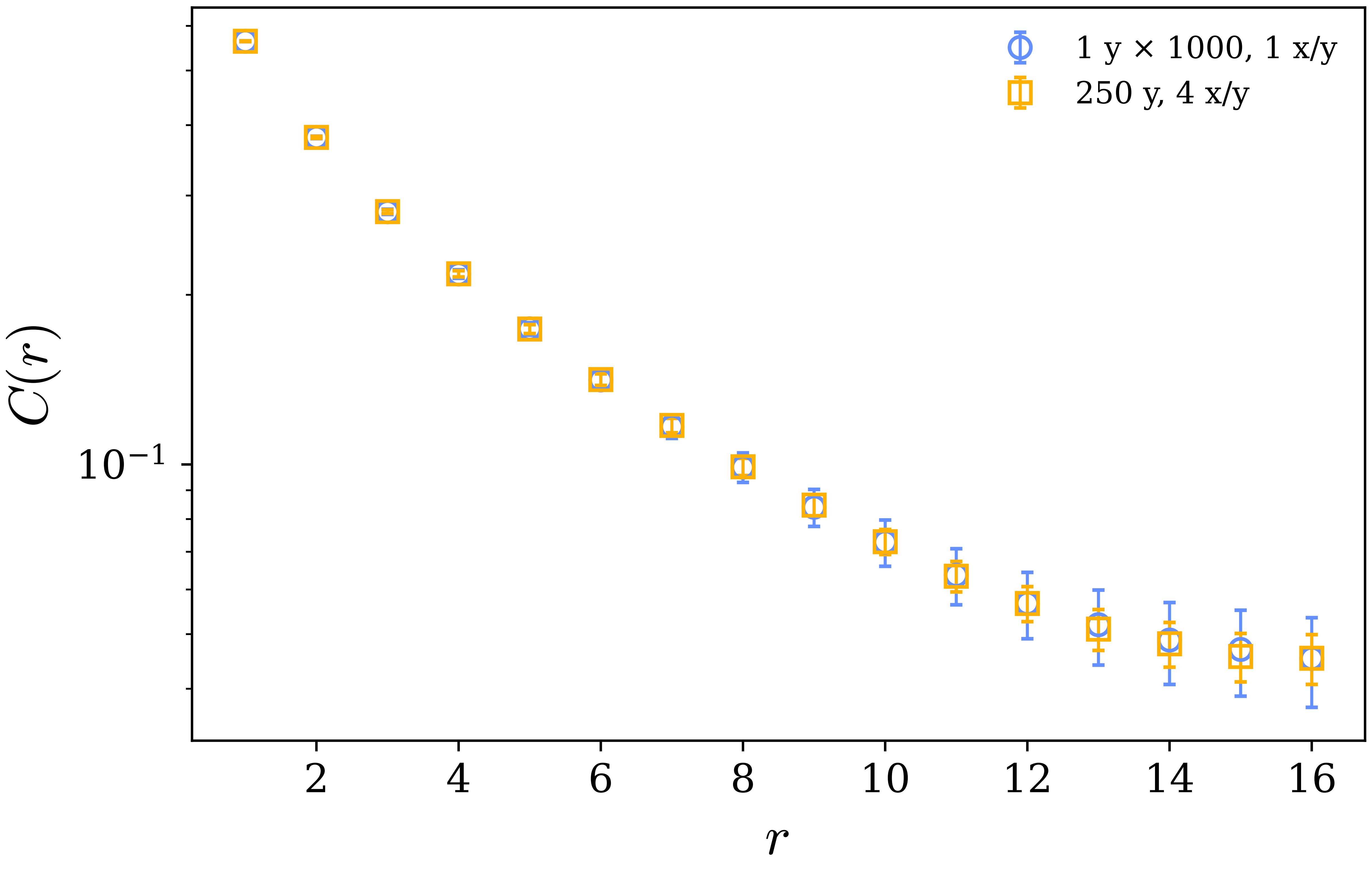
- ML generated coarse and fine ensembles have **compatible physical correlation lengths.**

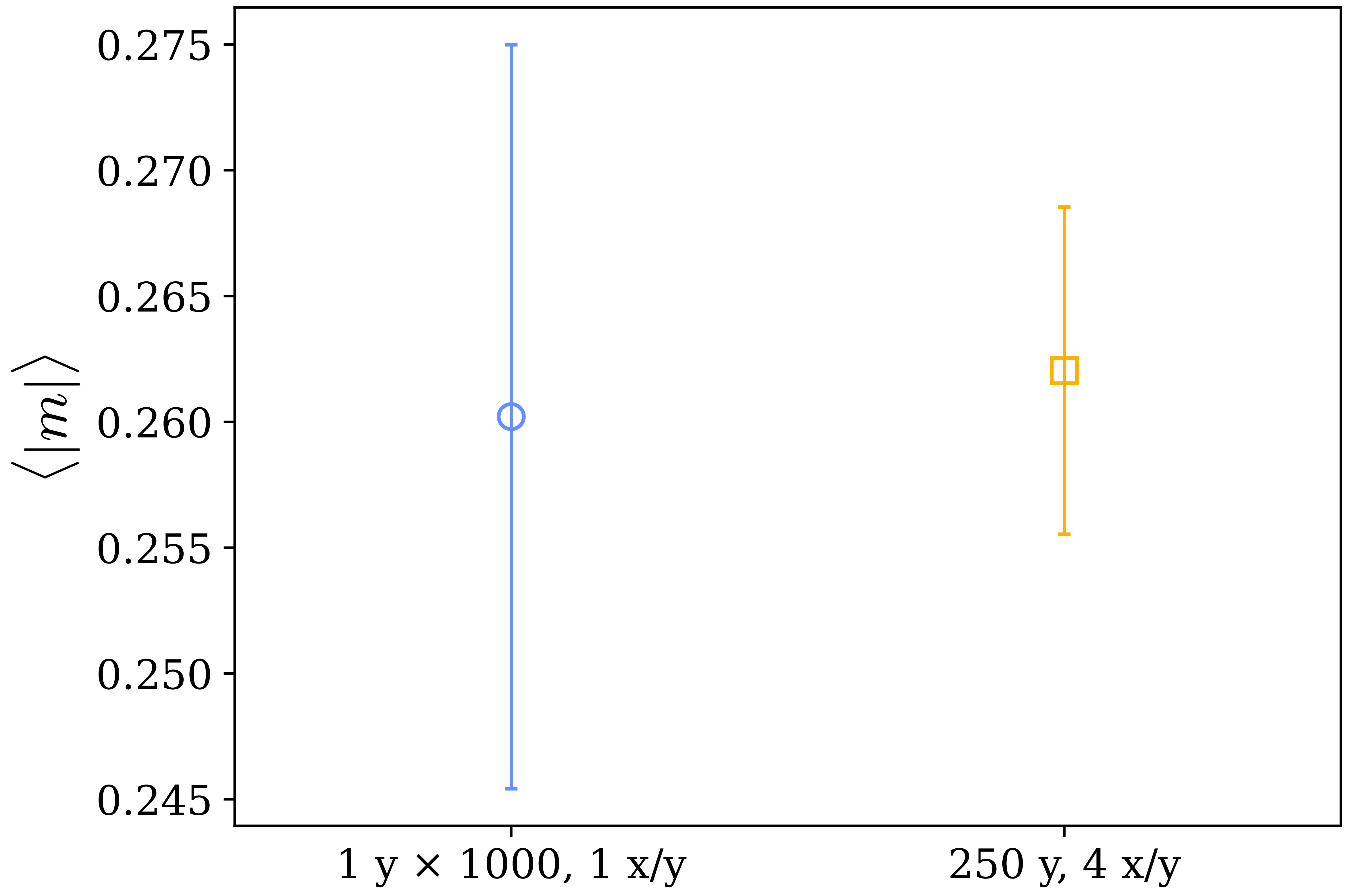


CONTROL VARIATES

Sampling **many fine configurations for a fixed coarse sample** can be used as **multi-hit**

[Parisi, Petronzio, Rapuano; Phys. Lett. B 128 (1983)]





VANILLA TRAINING

$p(x, y)^0$ ▼ → ▼ → ▼ → ▼ →

▼ : Compute the full SNF forward → gradient optimized locally for NF layers

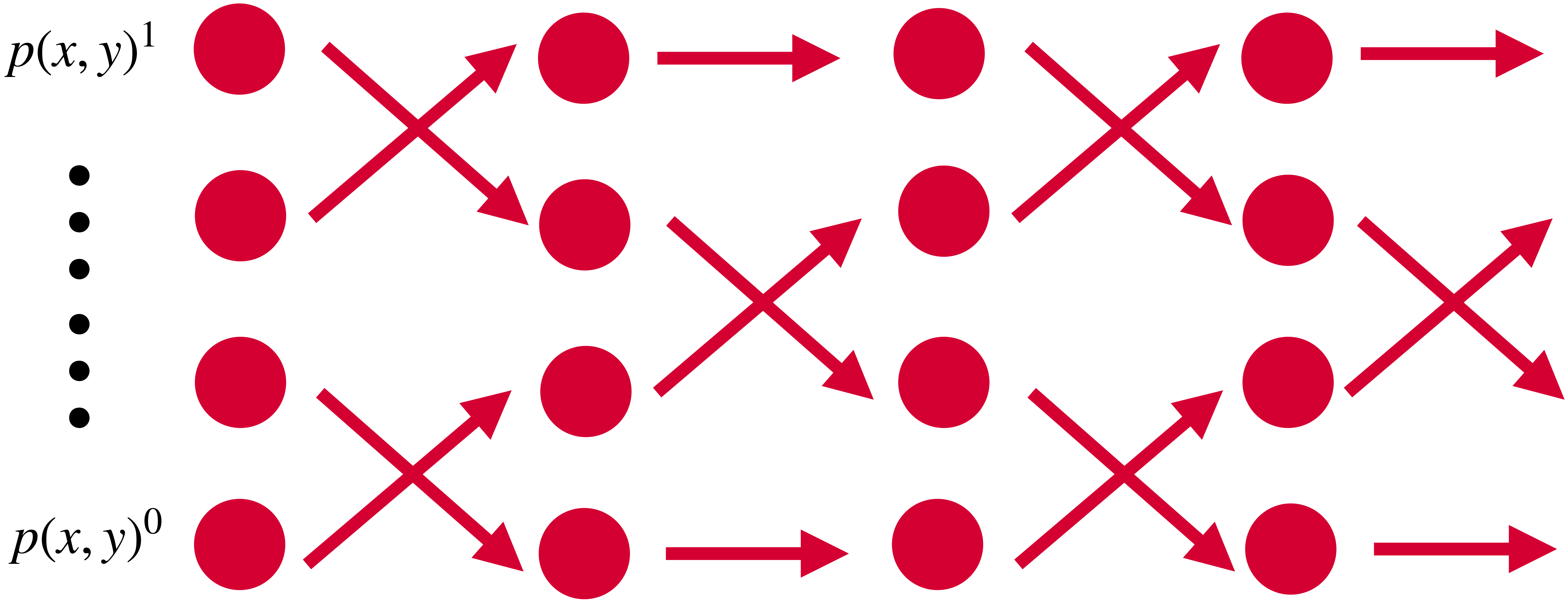
Usually batched/parallelized over N independent replicas:

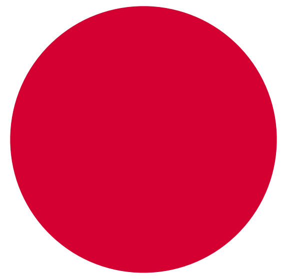
$p(x, y)^0_0$ ▼ → ▼ → ▼ → ▼ →

⋮

$p(x, y)^0_N$ ▼ → ▼ → ▼ → ▼ →

PARALLEL TEMPERING TRAINING



 : Compute the local loss and update the gradients → no explicit forward pass

Can be batched!