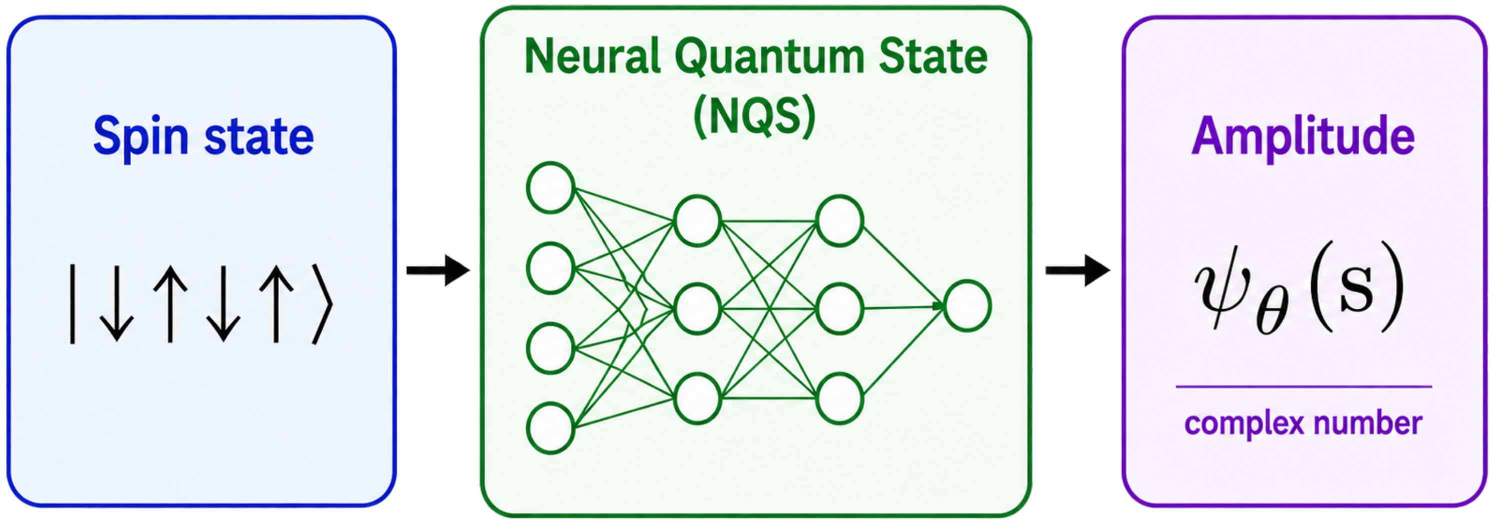
Normalising Flow-Assisted Neural Quantum States for Ground-State Estimation

Learning where to sample in an exponentially large Hilbert space

Timur Sypchenko
Institute for Particle Physics Phenomenology, Durham University
`timur.sypchenko@durham.ac.uk`

In collaboration with Michael Spannowsky and Vishal S. Ngairangbam

Ground-state estimation with neural quantum states



A neural quantum state represents the wavefunction amplitude $\psi_\theta(\mathrm{s})$ with a neural network.

$$E(\theta) = \frac{\langle\psi_\theta|H|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}$$

s: a spin state

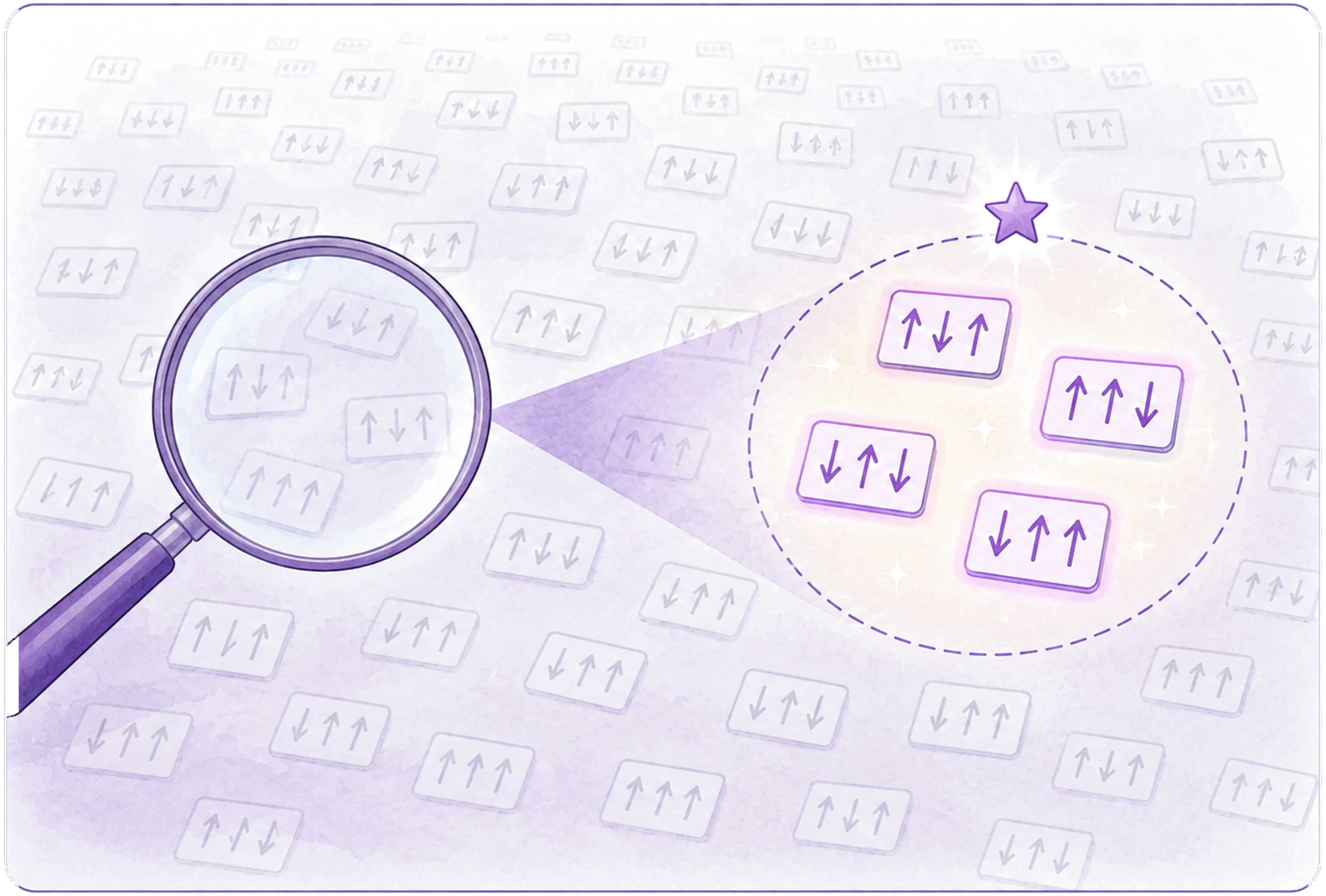
ψ_θ: the trial wavefunction with parameters θ

H: the Hamiltonian defining the physical system

E: the variational estimate of its ground-state energy

Training changes θ to lower the energy

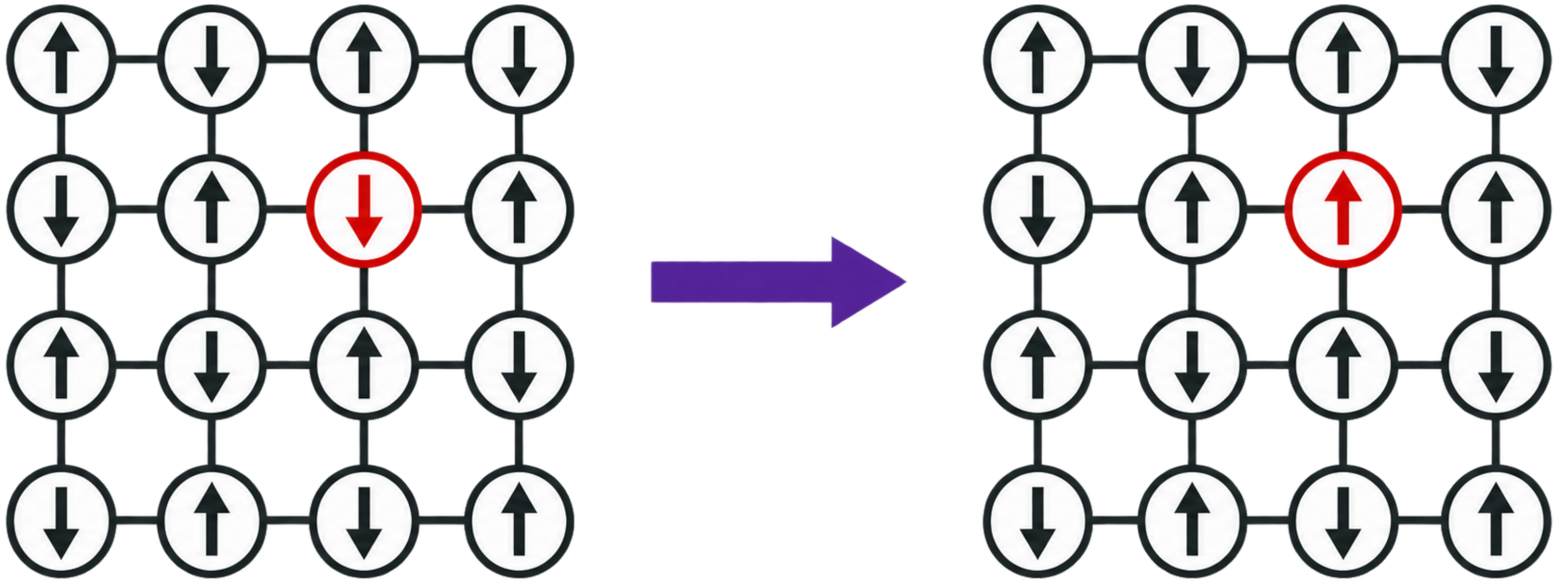
The Hilbert space grows exponentially



- An N-spin system contains up to 2^N basis states
- We cannot evaluate every state as N grows
- Training therefore relies on a sampled subset

Which states should we give the NQS?

A Metropolis step proposes a local change



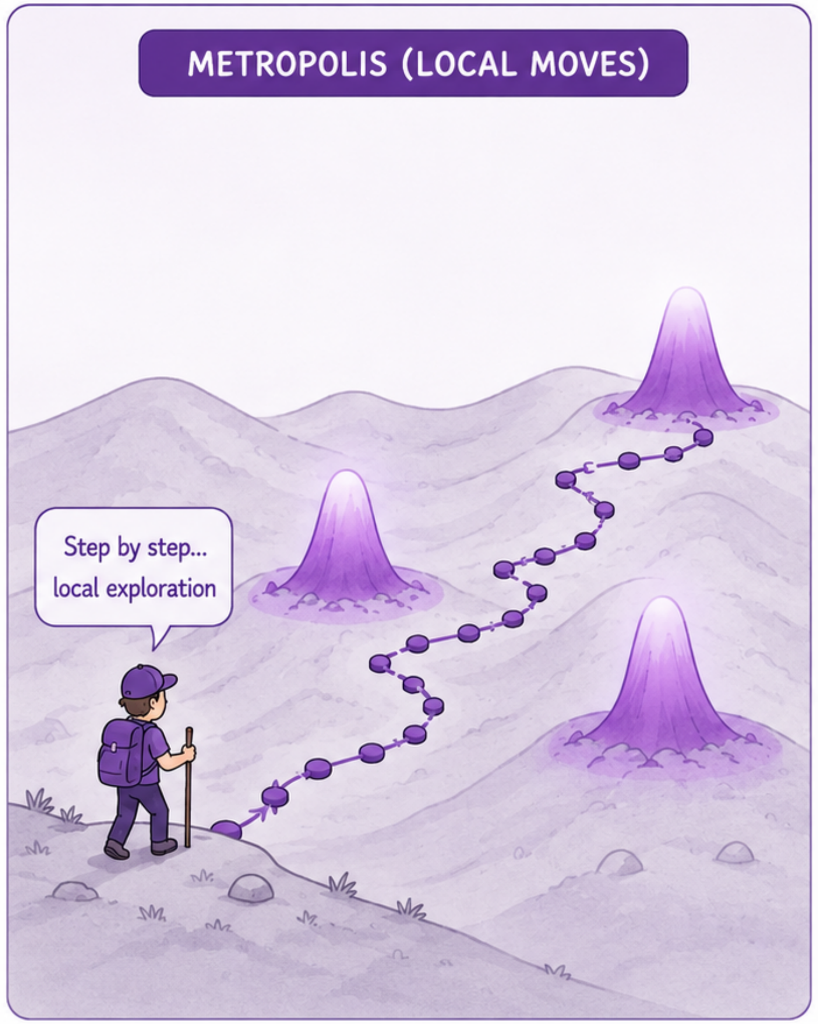
Propose a new state s' by flipping one spin in the current state s.

$$A(\mathbf{s} \to \mathbf{s}') = \min\left(1, \frac{|\psi_\theta(\mathbf{s}')|^2}{|\psi_\theta(\mathbf{s})|^2}\right)$$

A: probability of accepting the proposal
$|\psi_\theta(\mathrm{s})|^2$: target probability of spin state s

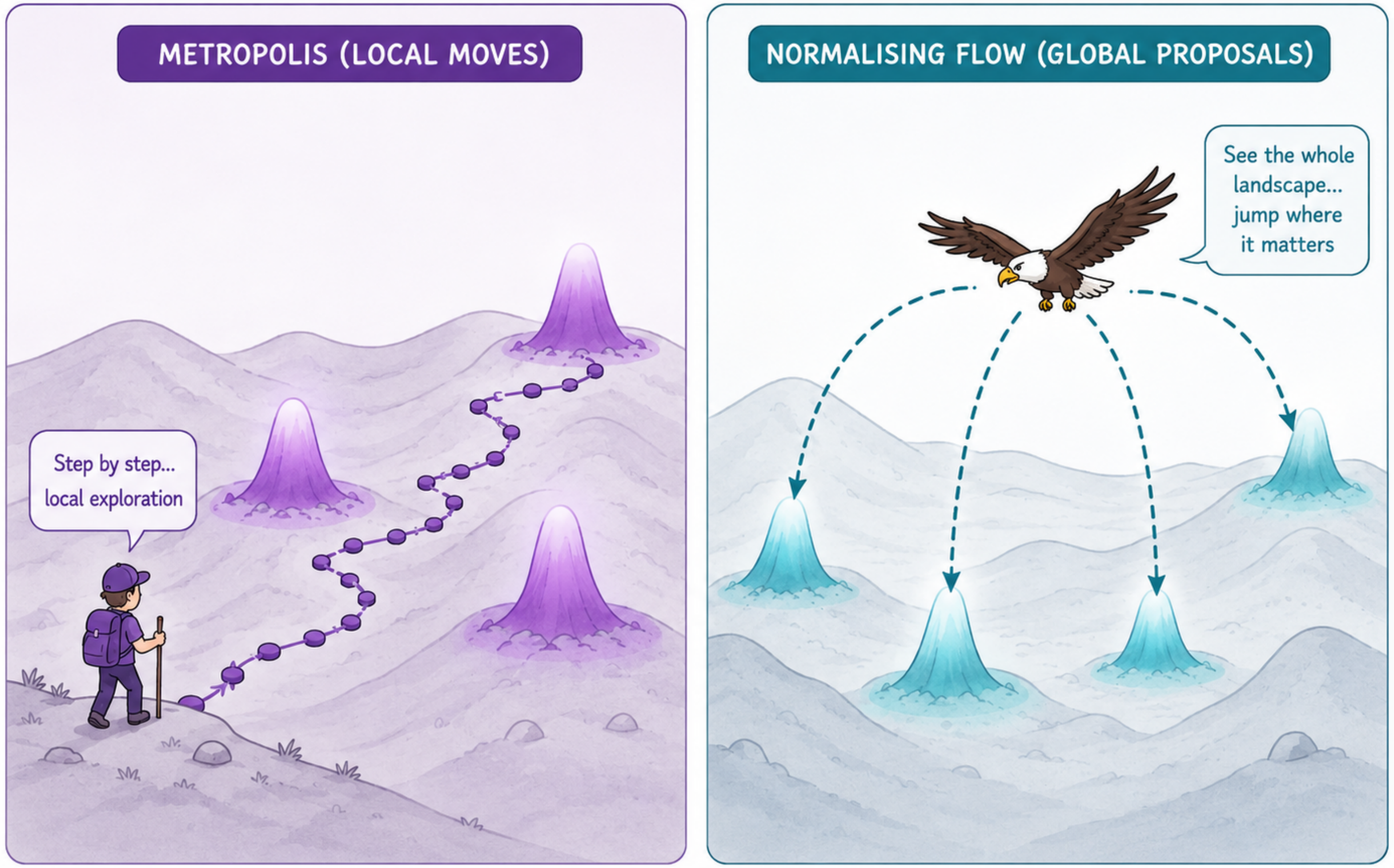
Moves to more probable states are always accepted; other moves are sometimes accepted so the chain can keep exploring

Local sampling walks through the state space



A Metropolis chain reaches distant important states through a sequence of local proposals

Local sampling has to walk between distant regions



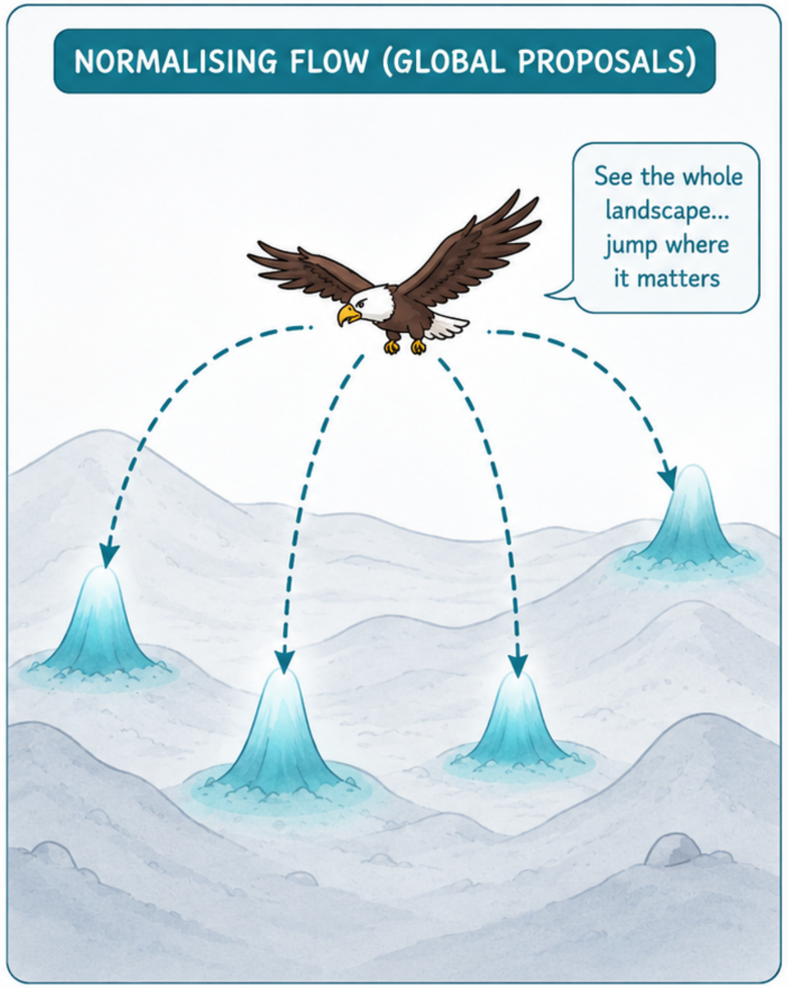
Metropolis: local moves

A chain moves step by step through nearby states

Normalising flow: global proposals

Draw states directly from a learned distribution

Local sampling has to walk between distant regions



Metropolis: local moves

A chain moves step by step through nearby states

Normalising flow: global proposals

Draw states directly from a learned distribution

Local sampling has to walk between distant regions

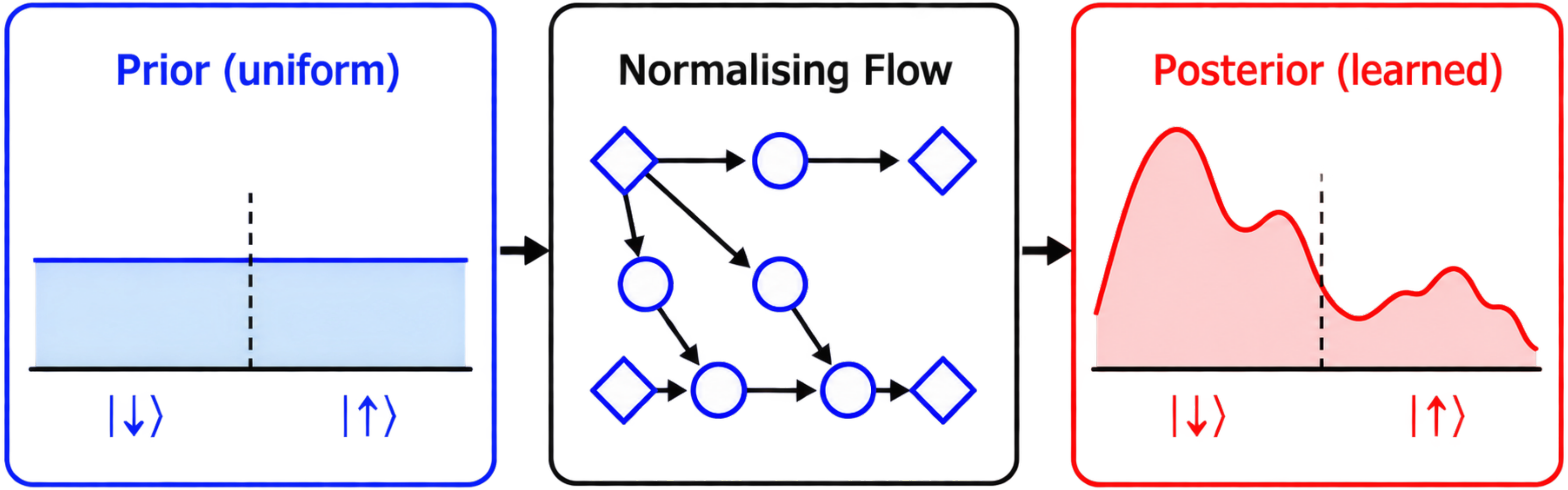
Metropolis: local moves

A chain moves step by step through nearby states

Normalising flow: global proposals

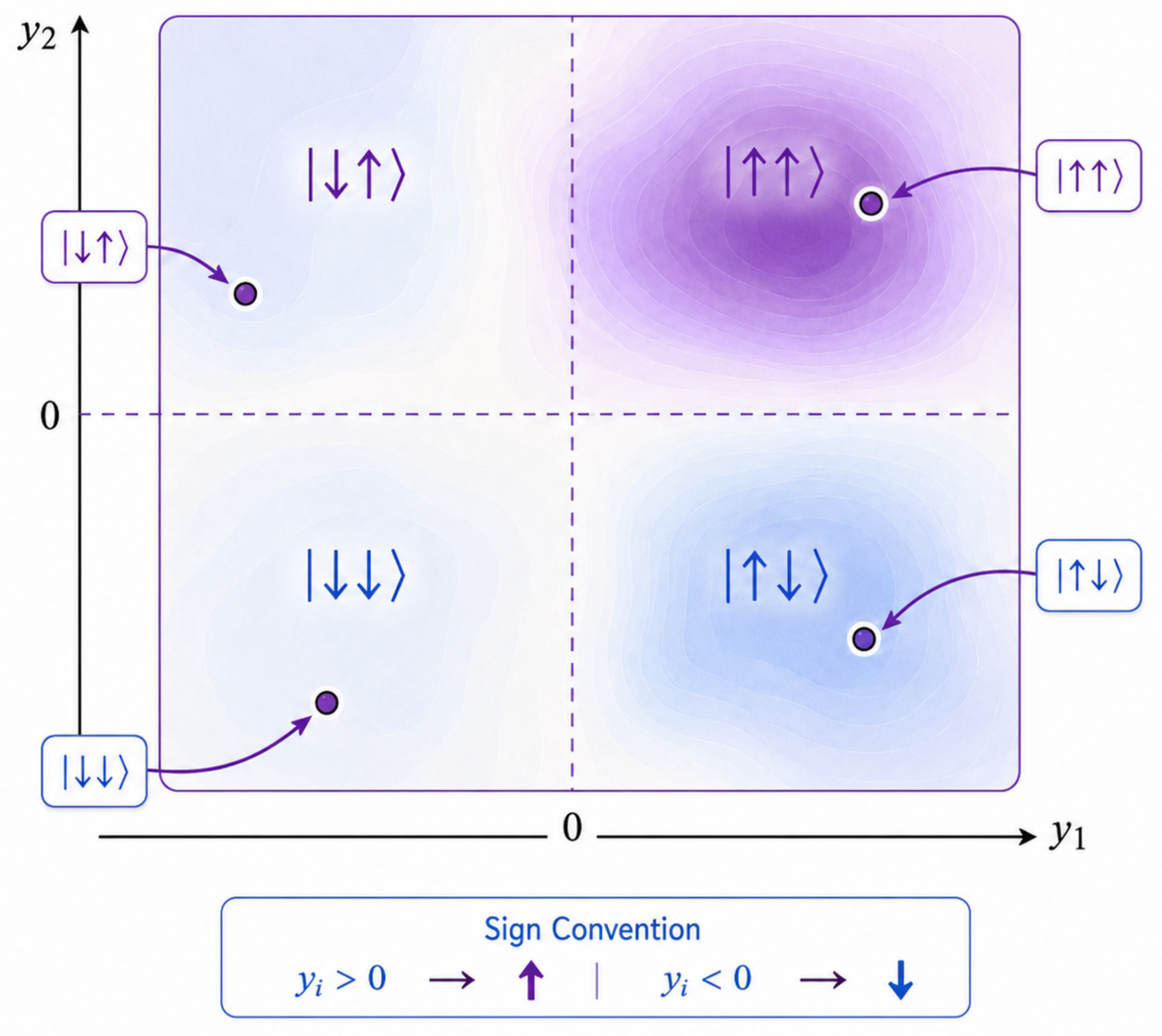
Draw states directly from a learned distribution

A normalising flow learns a global proposal



Sample a simple prior, transform each point through an invertible network, and learn a posterior concentrated around useful states

Continuous samples and discrete spin states



Divide the continuous posterior into equal-volume, non-overlapping regions R_{s}.

Each region represents one spin state s.

A continuous point becomes the state associated with the region in which it lands.

A continuous generative model can now propose discrete spin states

Probability assigned to a spin state

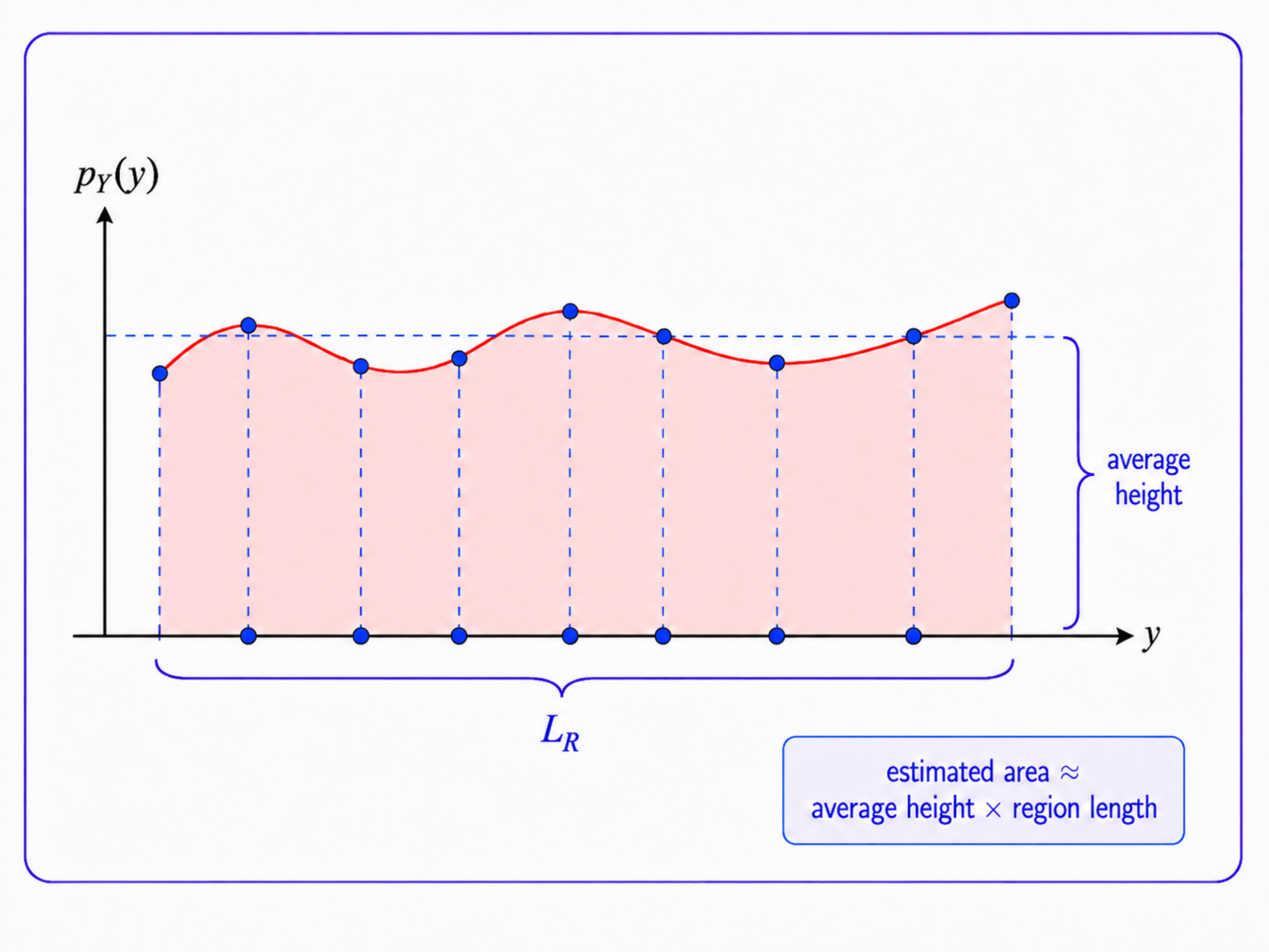
The probability of state s is the continuous probability mass inside its region $R_{\mathbf{s}}$

$$q_\phi(\mathbf{s}) \quad = \quad \int_{R_{\mathbf{s}}} p_Y(\mathbf{y})\, d\mathbf{y}$$

Estimate the integral with M points sampled uniformly inside the region

$$\widehat{q}_\phi(\mathbf{s}) \quad \approx \quad \frac{\mathrm{Vol}(R_{\mathbf{s}})}{M} \sum_{m=1}^{M} p_Y(\mathbf{y}_m)$$

$p_Y(\mathbf{y})$: flow density at continuous point $\mathbf{y}$
ϕ: flow parameters; M: number of integration points



Importance weights connect the two models

For every sampled state, compare its probability under the NQS with the probability assigned by the flow

$$\widetilde{w}_i = \frac{|\psi_\theta(\mathbf{s}_i)|^2}{\widehat{q}_\phi(\mathbf{s}_i)}, \qquad w_i = \frac{\widetilde{w}_i}{\sum_j \widetilde{w}_j}$$

Large w_i
The NQS considers the state important, but the flow samples it too rarely

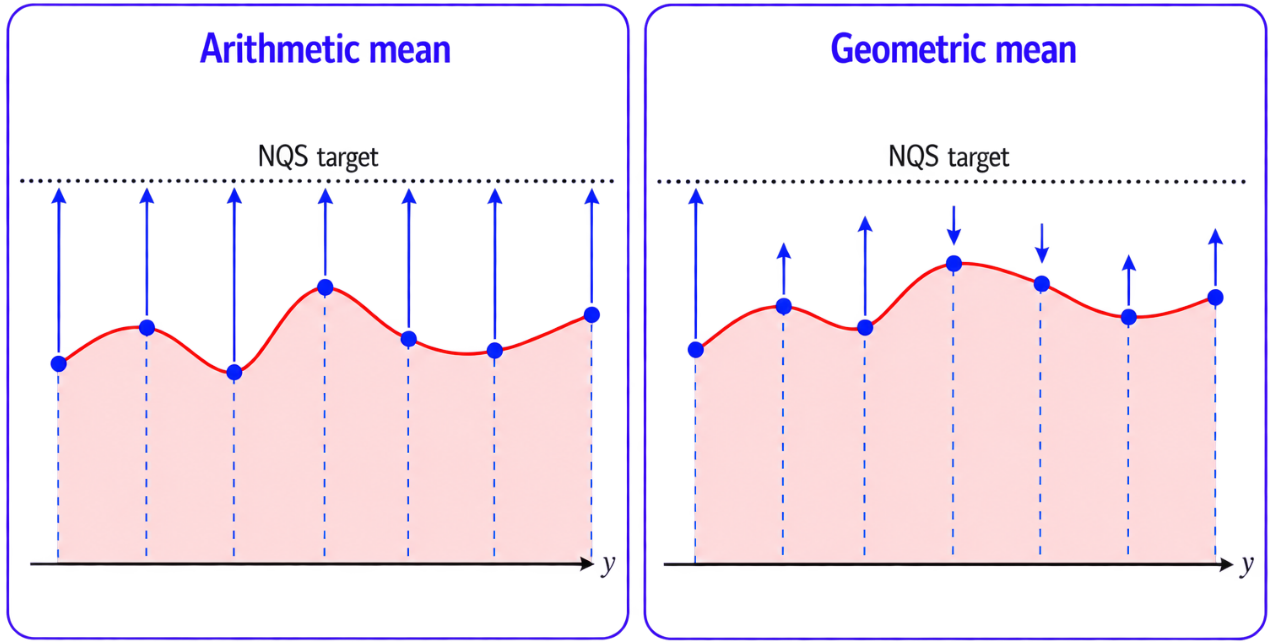
Small w_i
The flow already represents the state well

i labels sampled states and the normalization makes the weights sum to one

The normalising flow loss

The loss increases the flow probability of states that receive large importance weights

$$\mathcal{L}_{\mathrm{NF}} = -\sum_i w_i \left[(1-\lambda)\log \widehat{q}_{\mathrm{arith}}(\mathbf{s}_i) + \lambda \log \widehat{q}_{\mathrm{geom}}(\mathbf{s}_i)\right]$$



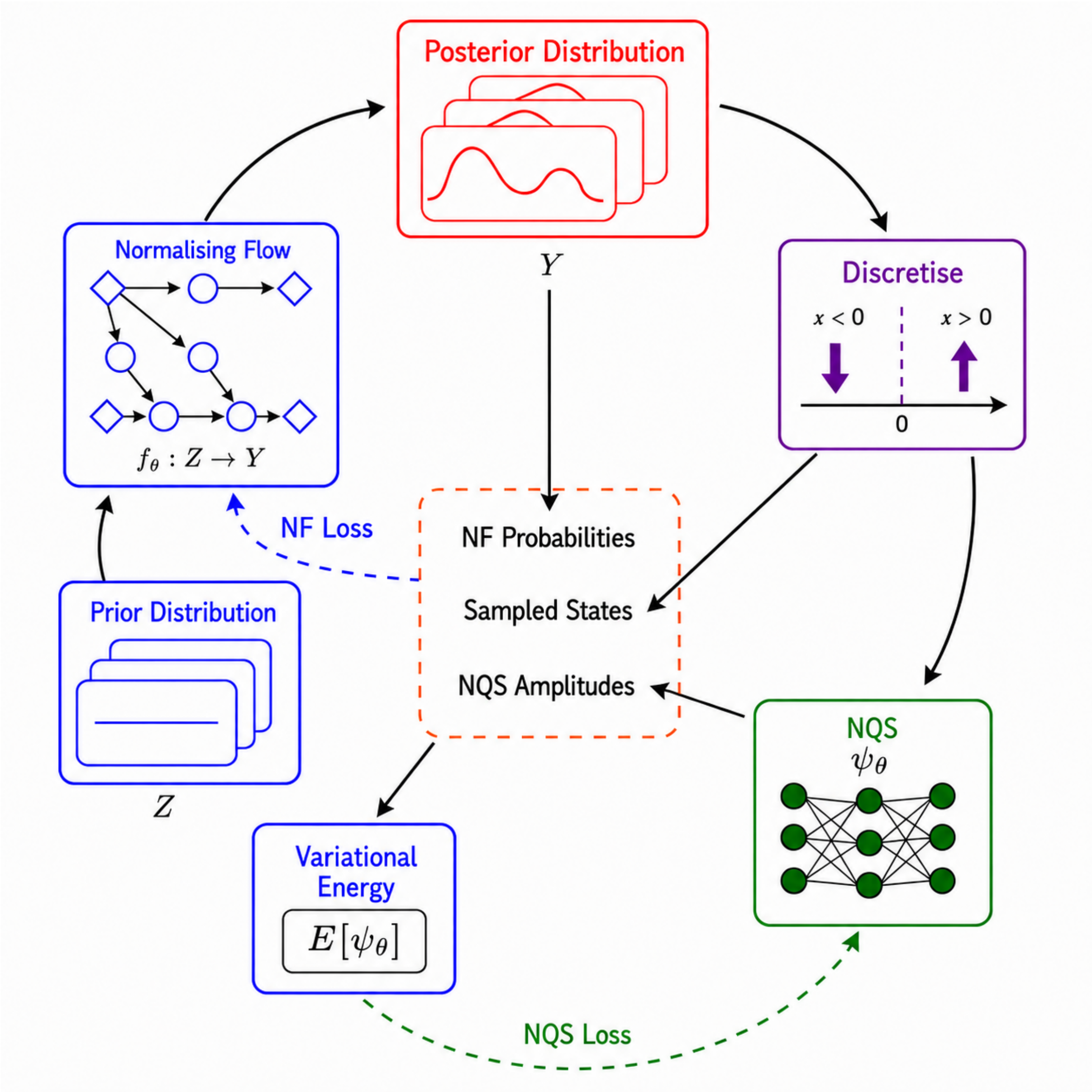
Arithmetic estimate
Matches the total probability assigned to a state region

Geometric estimate
Encourages a smoother density inside that region

λ: the balance between probability matching and smoothing

Practical message: the flow learns to place probability where the NQS needs samples

The coupled NF and NQS training loop



One training iteration

1 Sample continuous points from the flow

2 Discretise them into spin states

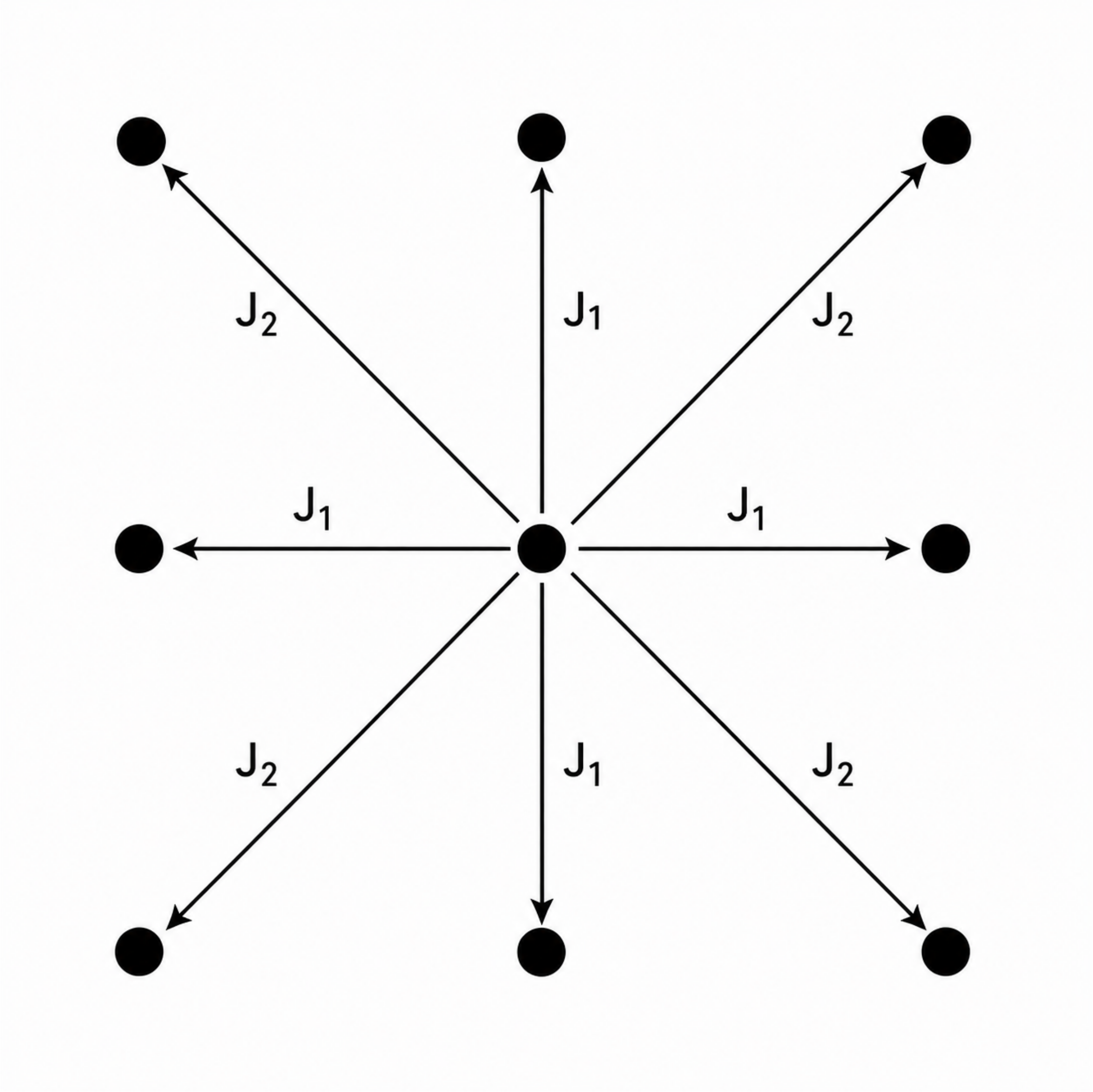
3 Estimate each state probability

4 Evaluate NQS amplitudes and local energies

5 Update both models

The NQS teaches the flow which regions matter

The frustrated Heisenberg benchmark



$$H = J_1 \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \mathbf{S}_i \cdot \mathbf{S}_j$$

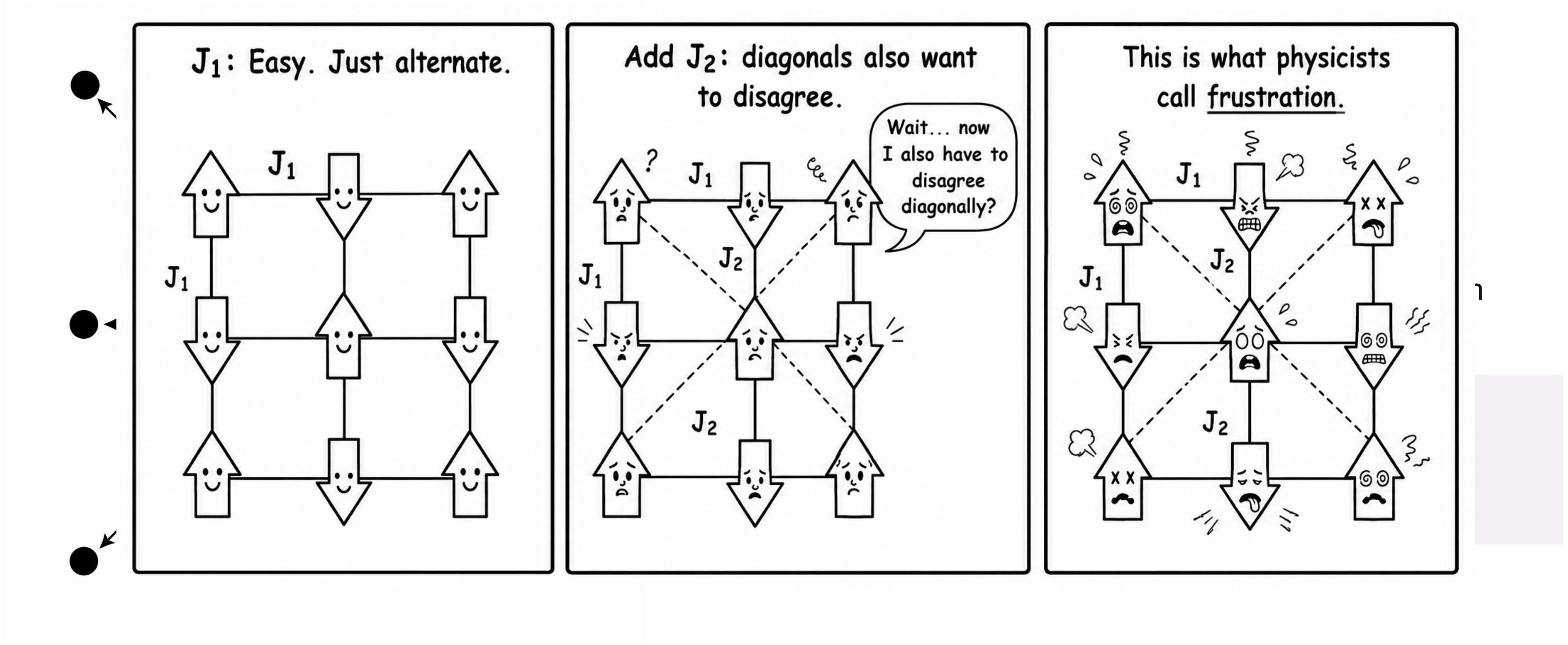
S_i: spin operator at lattice site i

J_1: nearest-neighbour interaction strength

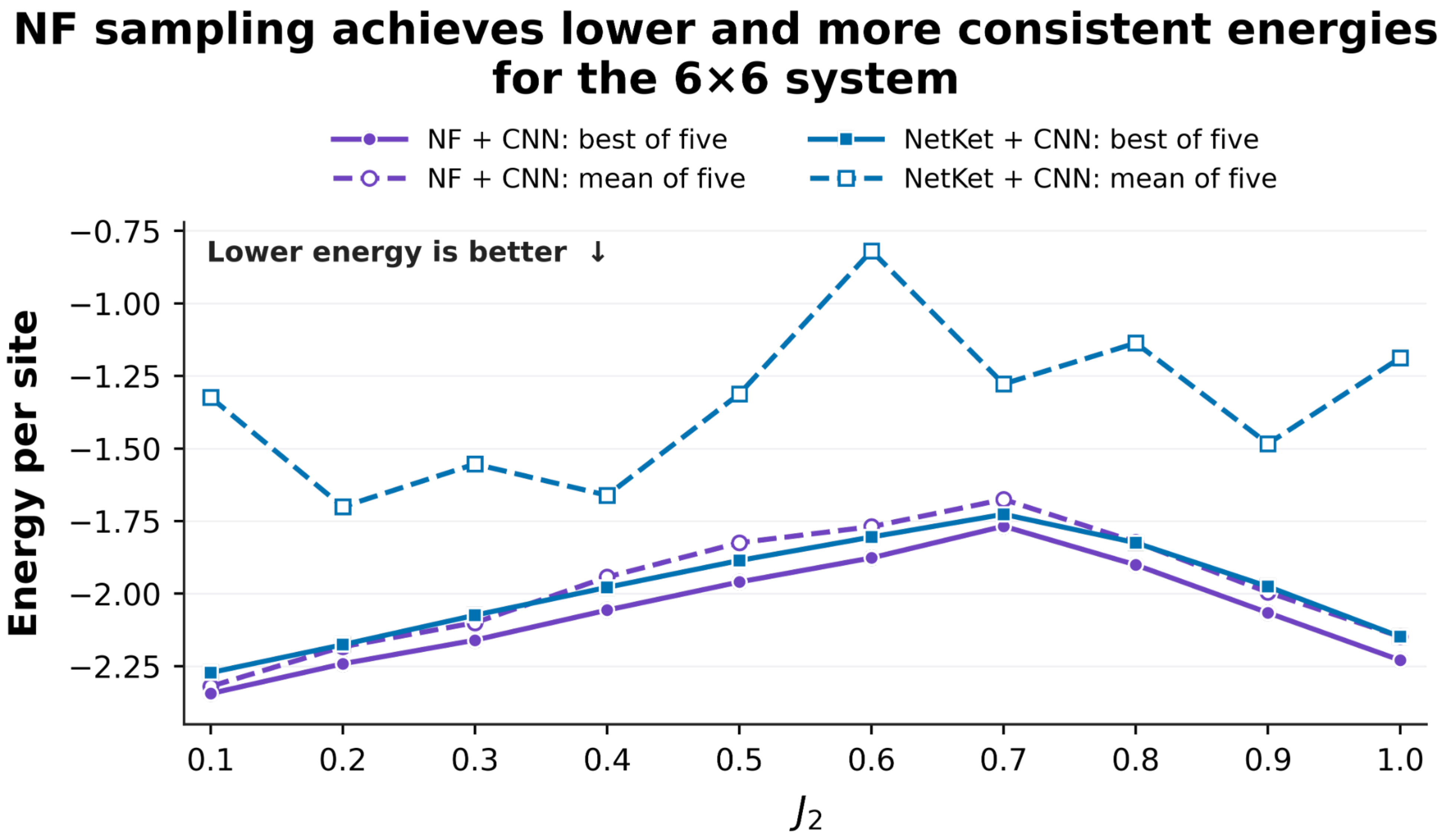
J_2: diagonal next-nearest-neighbour interaction strength

Near $J_2/J_1 \approx 0.5$, competing interactions frustrate the spin ordering and make optimisation difficult

The frustrated Heisenberg benchmark



NF proposals improve the energy results



Lower energy is better

NF reaches a better best-repeat energy

The NF mean is lower and more consistent

Local Metropolis converges in only 60% of repeats

Takeaway

Normalising flows learn global proposals for neural quantum states

- NQS optimisation depends on finding important states in an exponentially large space
- A continuous flow can propose discrete states through region-based discretisation
- The NQS trains the flow to sample useful regions more often
- On the Heisenberg benchmark, the NF sampler gives lower and more consistent energies

Timur Sypchenko `timur.sypchenko@durham.ac.uk`