

Scaling Neural Quantum States beyond 10000 quantum spins

Rutger Berns - Radboud University, The Netherlands

Johan Mentink - Radboud University, The Netherlands

Philipp Küppers - Radboud University, The Netherlands

Alessandro Pierro - LMU Munich, Germany

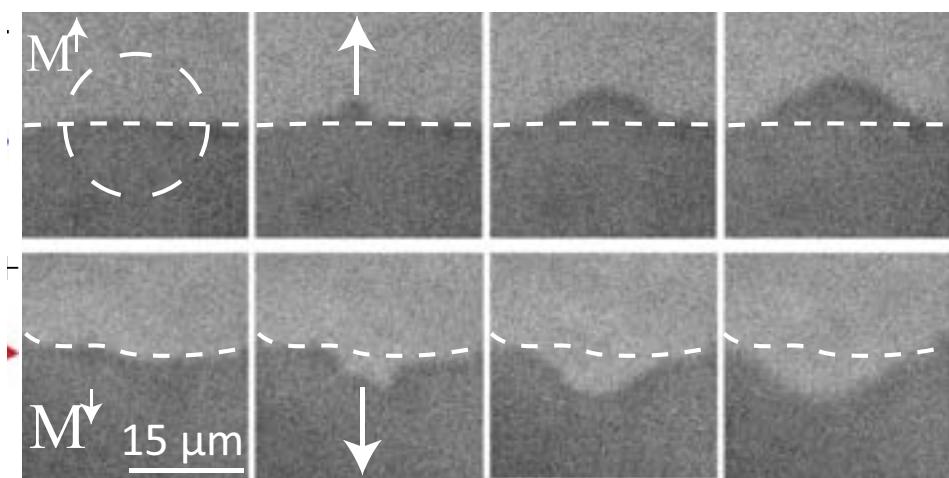
NeuralTensor – July 9th 2026

Radboud Universiteit



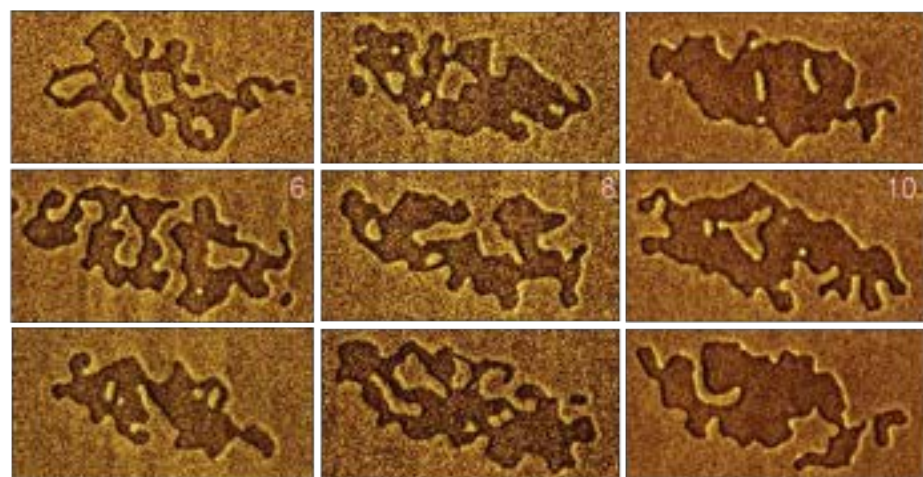
Magnetic spin textures in ultrafast nanomagnetism

Domains



Medapalli et al. (2017) *Phys. Rev. B*, 96(22), 224421.

Vortices



Khusyainov et al. (2026) *Nat. Mater.*

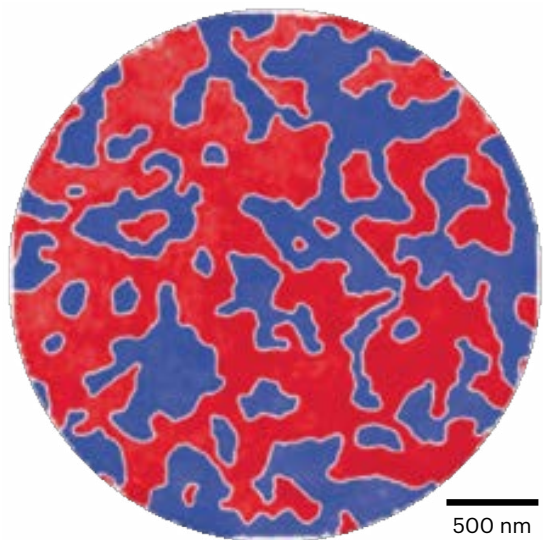
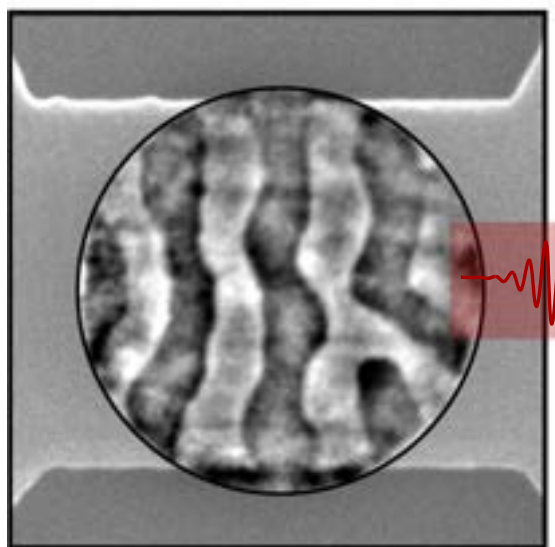


Heins et al. (2026) *Science*, 391(6781), 190–194.

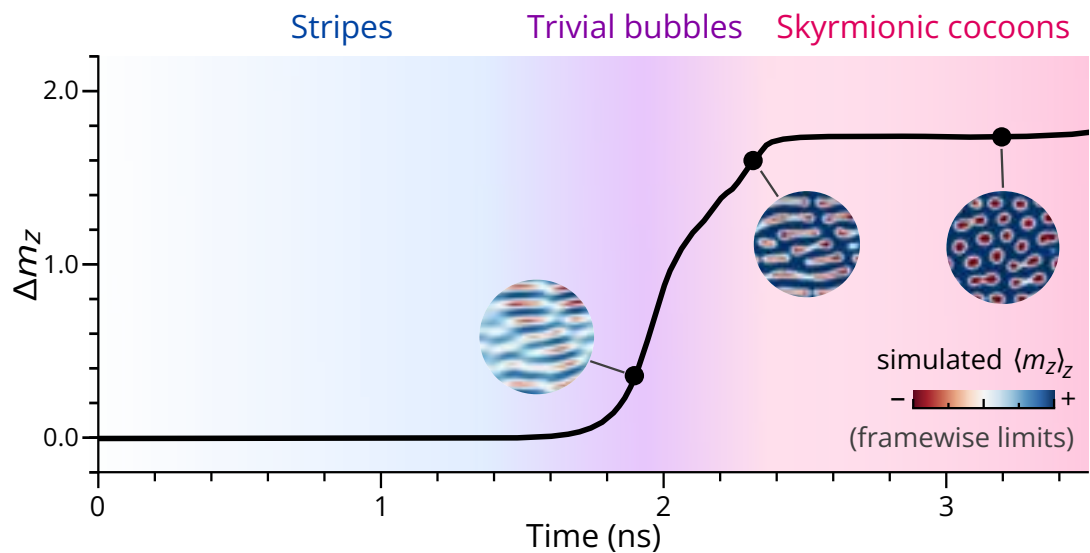
Skyrmions



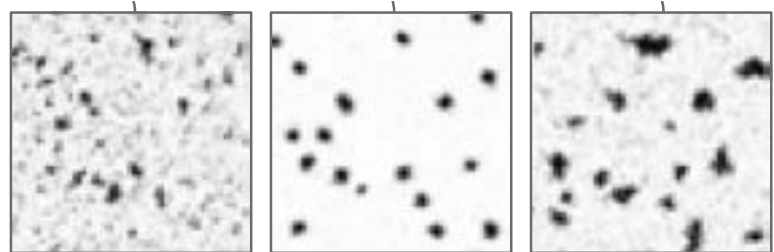
Fert et al. (2013) *Nat. Nanotech.* 8, 152–156



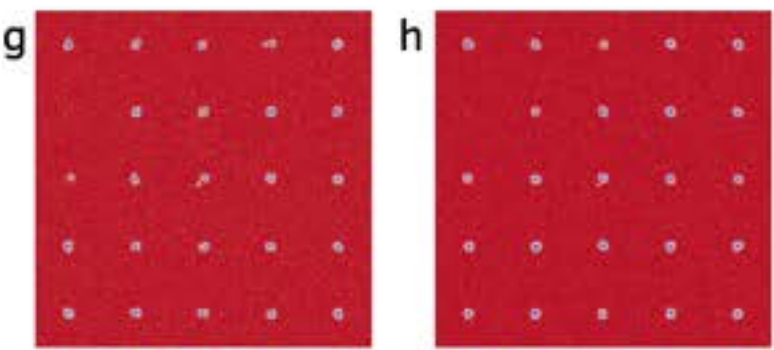
H.-T. Chang et al. (2026) *Nat. Mater.*



R. Battistelli et al. (2025) *arXiv:2512.22947*



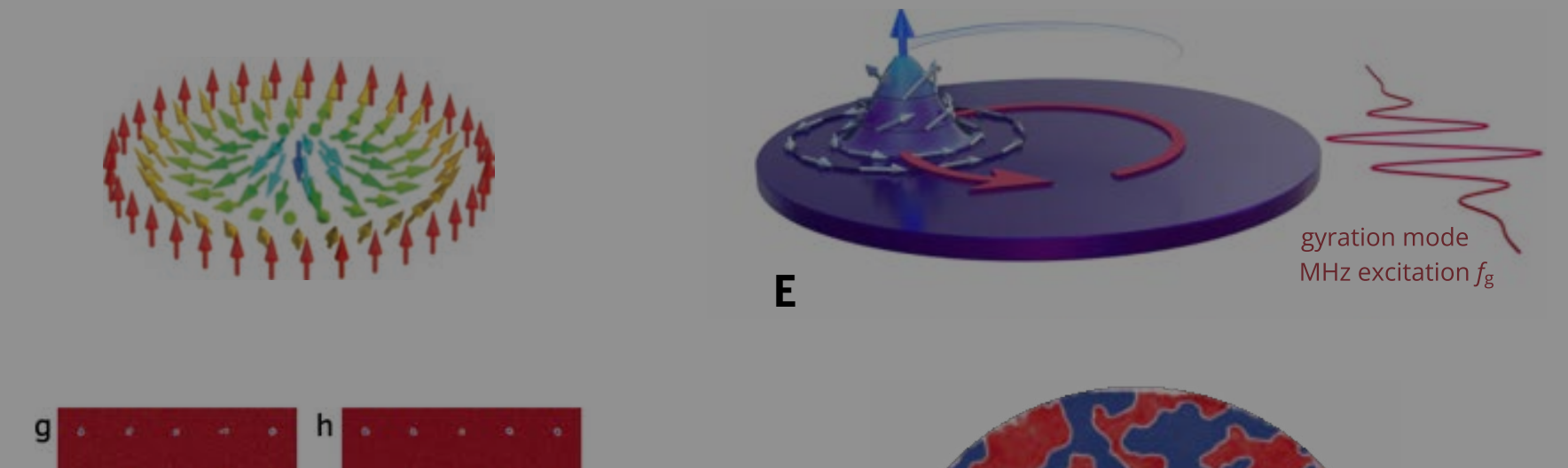
Liefferink et al. (2026) *Phys. Rev. Lett.* 136, 146705



Metternich et al. (2025) *arXiv:2512.17643*

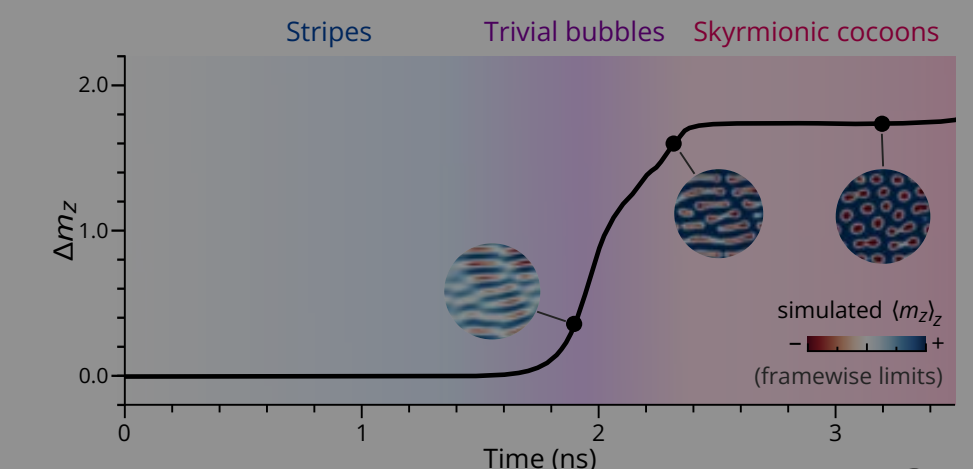
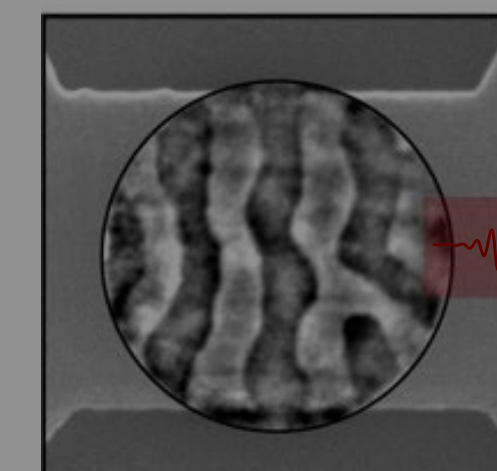
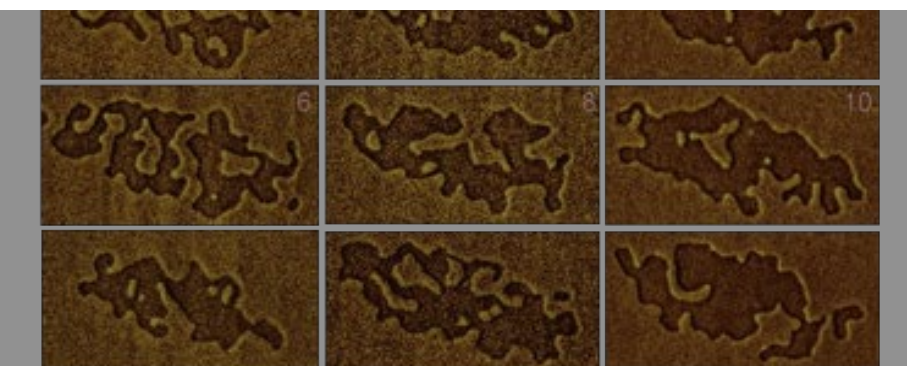
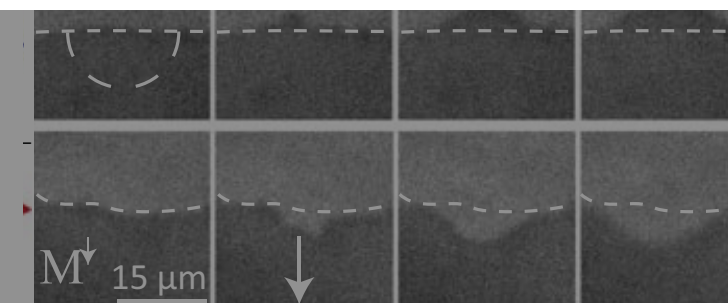
Quantum dynamics of magnetic spin textures

- Success simulating key physics with atomistic spin dynamics!



Requires scalable simulation
>10.000 quantum spins

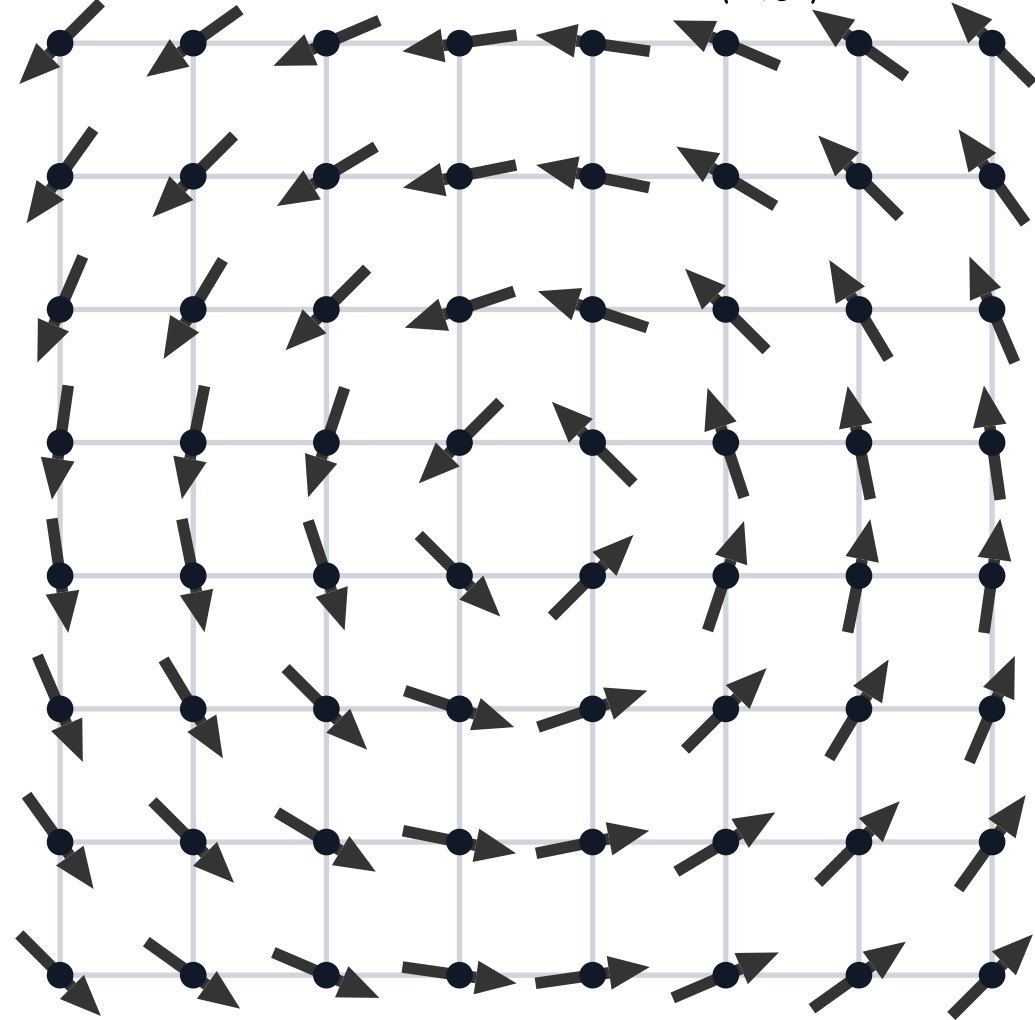
- What would happen when taking into account **quantum effects**?



Quantum dynamics of magnetic spin textures

Vortices

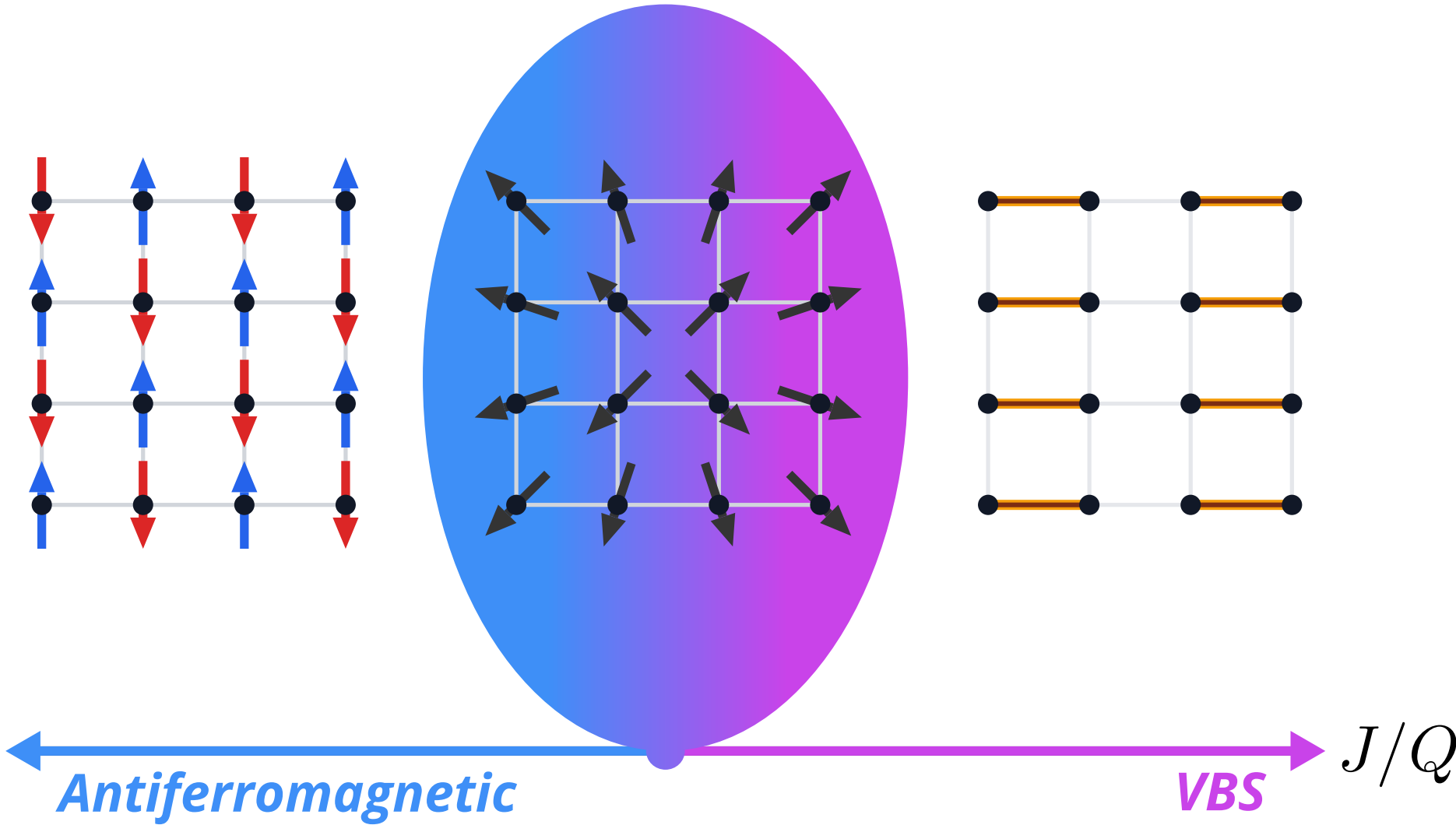
$$\hat{H} = J \sum_{\langle i,j \rangle} \left(\hat{S}_i^x \hat{S}_j^x + \hat{S}_i^y \hat{S}_j^y \right) + J^z \sum_{\langle i,j \rangle} \hat{S}_i^z \hat{S}_j^z + \sum_{b \in \mathcal{V}} \mathbf{B}_b \cdot \hat{\mathbf{S}}_b$$



Toscano et al. (2011) *J. Appl. Phys.* 109, 014301
Coenders et al. (2026)

Spin textures at quantum criticality

$$\hat{H} = J \sum_{\langle ij \rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j - Q \sum_{\langle ijkl \rangle} \left(\hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j - \frac{1}{4} \right) \left(\hat{\mathbf{S}}_k \cdot \hat{\mathbf{S}}_l - \frac{1}{4} \right)$$



Senthil et al. (2004) *Science* 303(5663), 1490–1494

Sandvik, A. W. (2007). *Phys. Rev. Let.* 98(22), 227202.

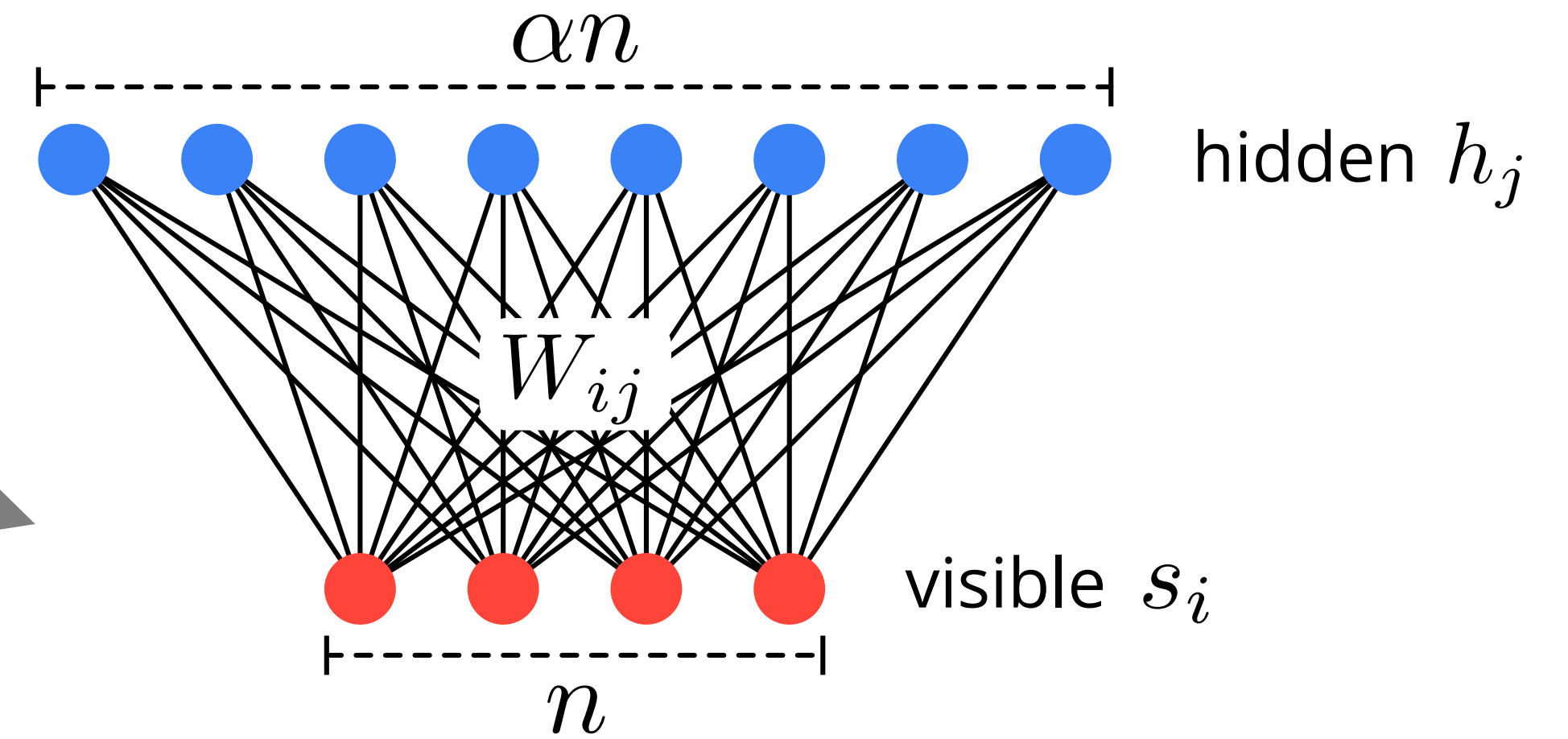
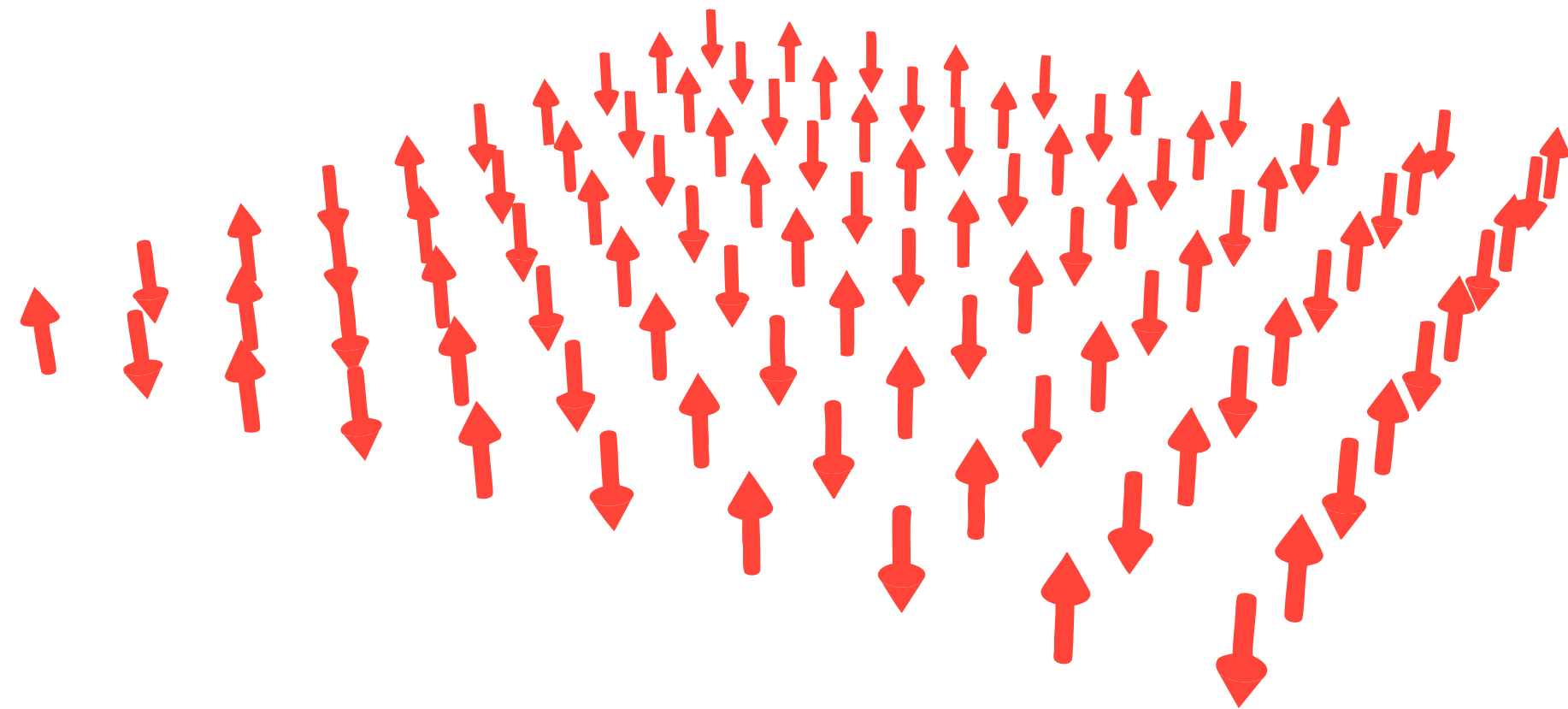
Minimal model: antiferromagnetic Heisenberg model: $\hat{H} = J \sum_{\langle ij \rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j$

Neural Quantum States

Variational technique

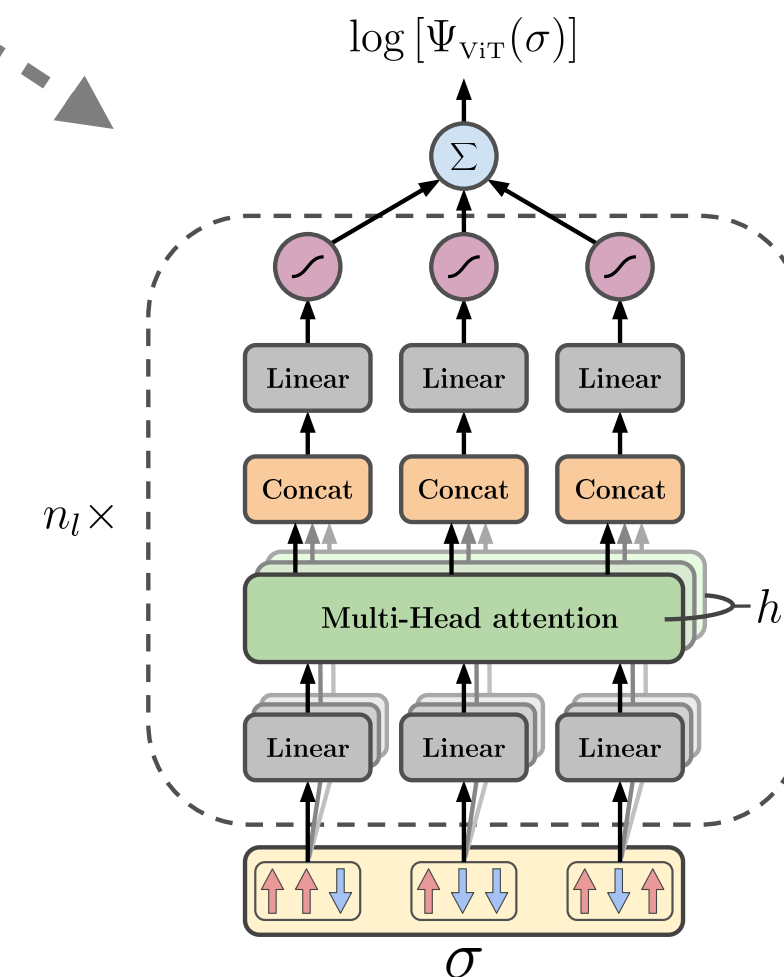
- Variational Ansatz: $|\psi\rangle = \sum_s \psi_M(s) |s\rangle$

$$E_{\text{gs}} \leq \frac{\langle \psi_M | H | \psi_M \rangle}{\langle \psi_M | \psi_M \rangle}$$

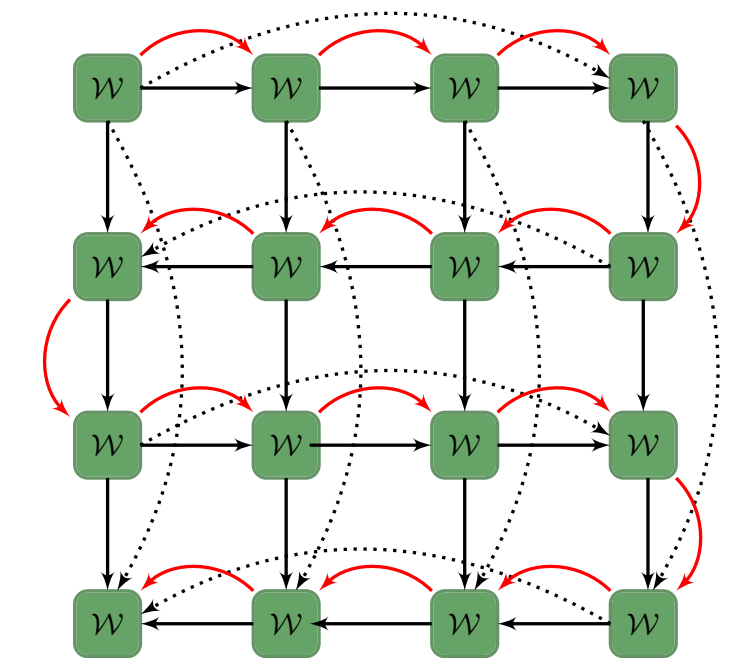


Restricted Boltzmann Machine

Carleo & Troyer (2017) *Science* 355,602-606



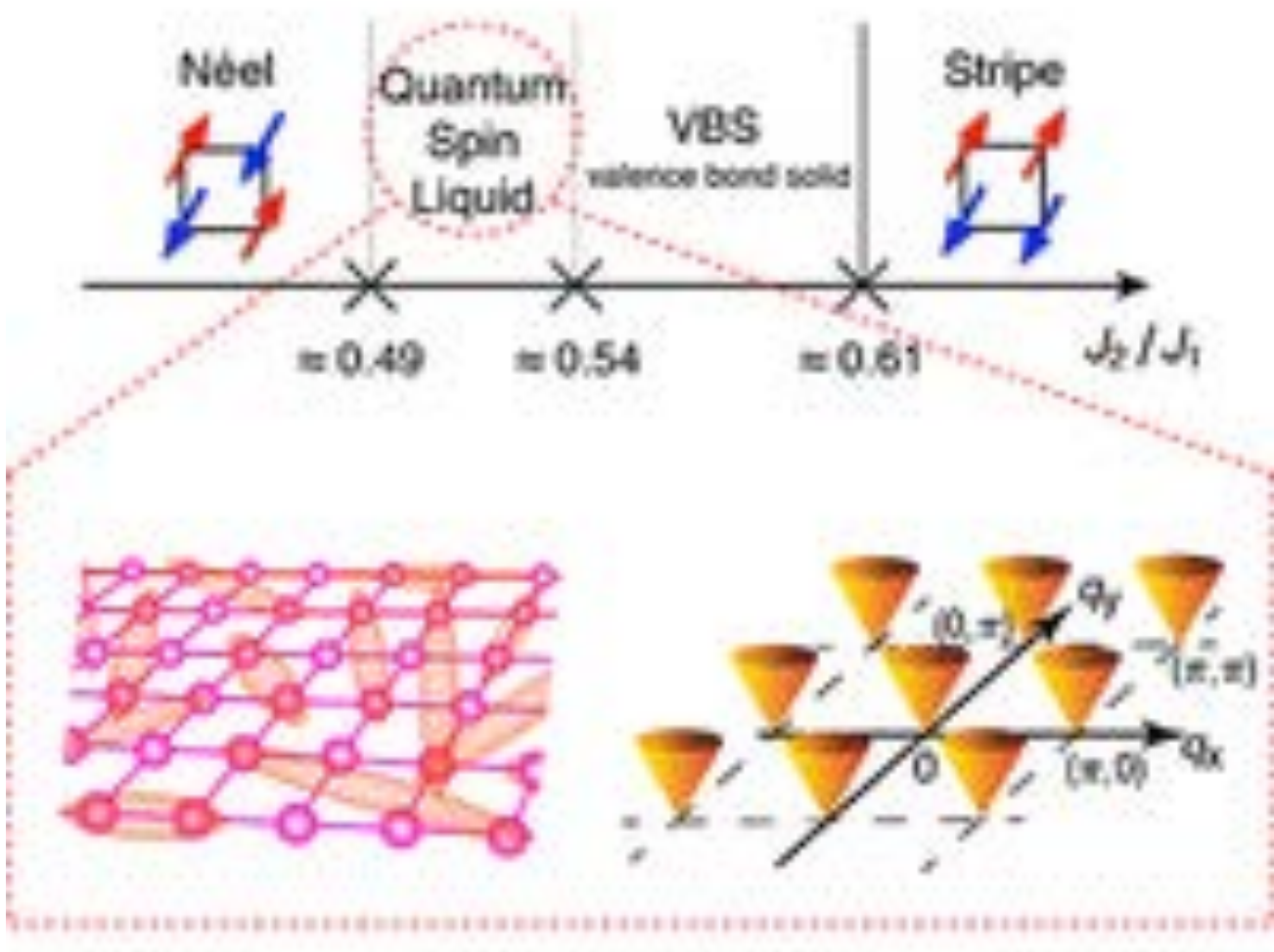
Viteritti et al. (2023)
Phys. Rev. Lett. 130, 236401



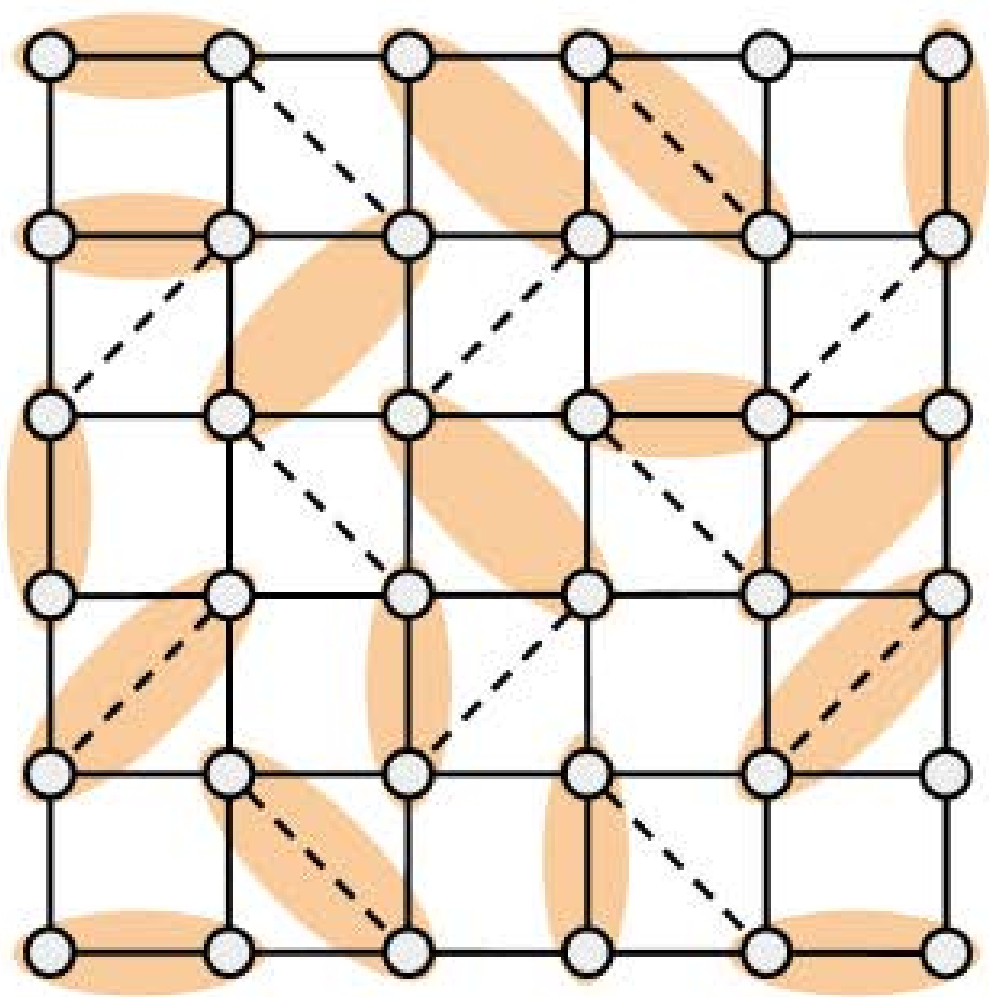
Moss et al. (2025)
PRB 112, 13445

NQS is powerful!

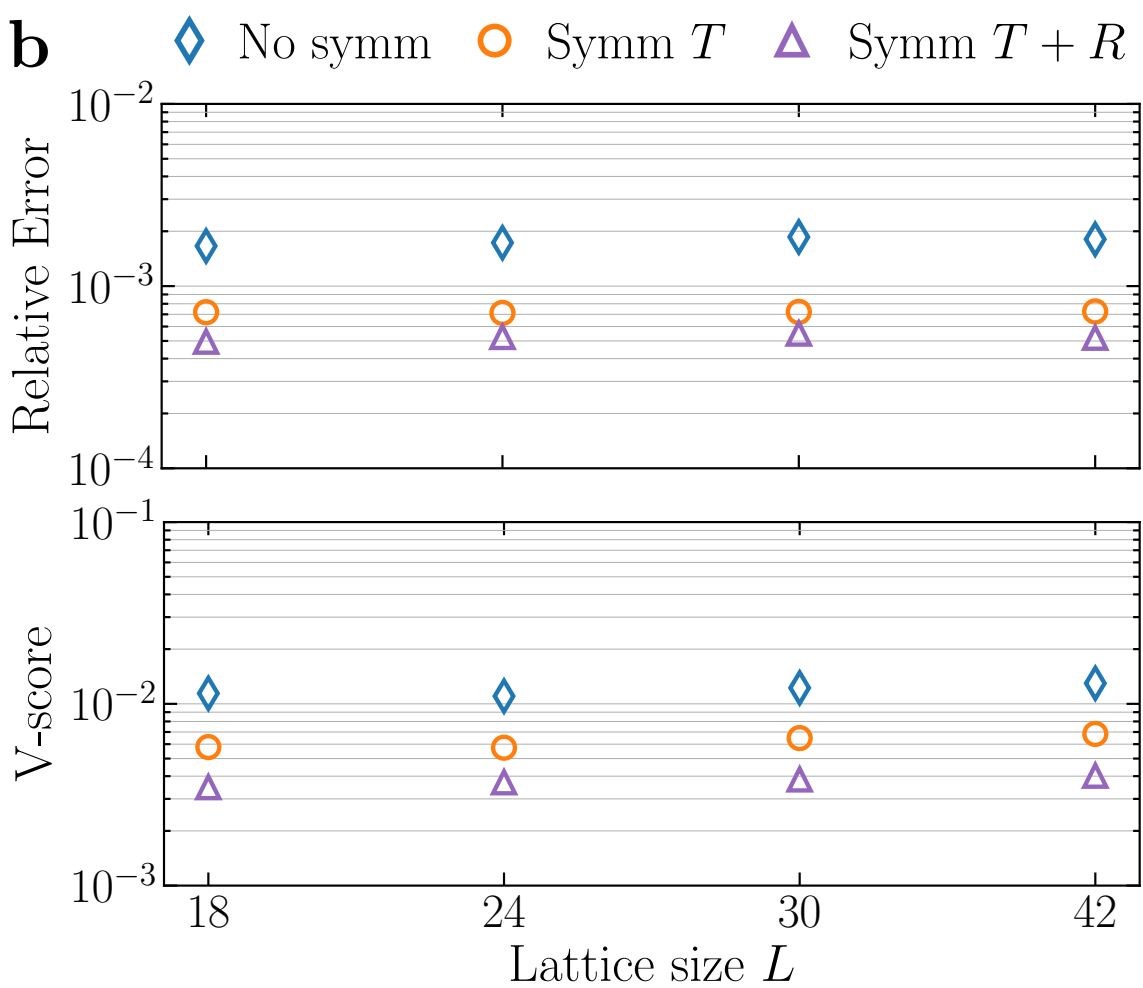
Quantum spin liquids



Shastry-Sutherland model



Frustrated models



12x12

Nomura, Imada (2021)
Phys. Rev. X 11, 031034

18x18

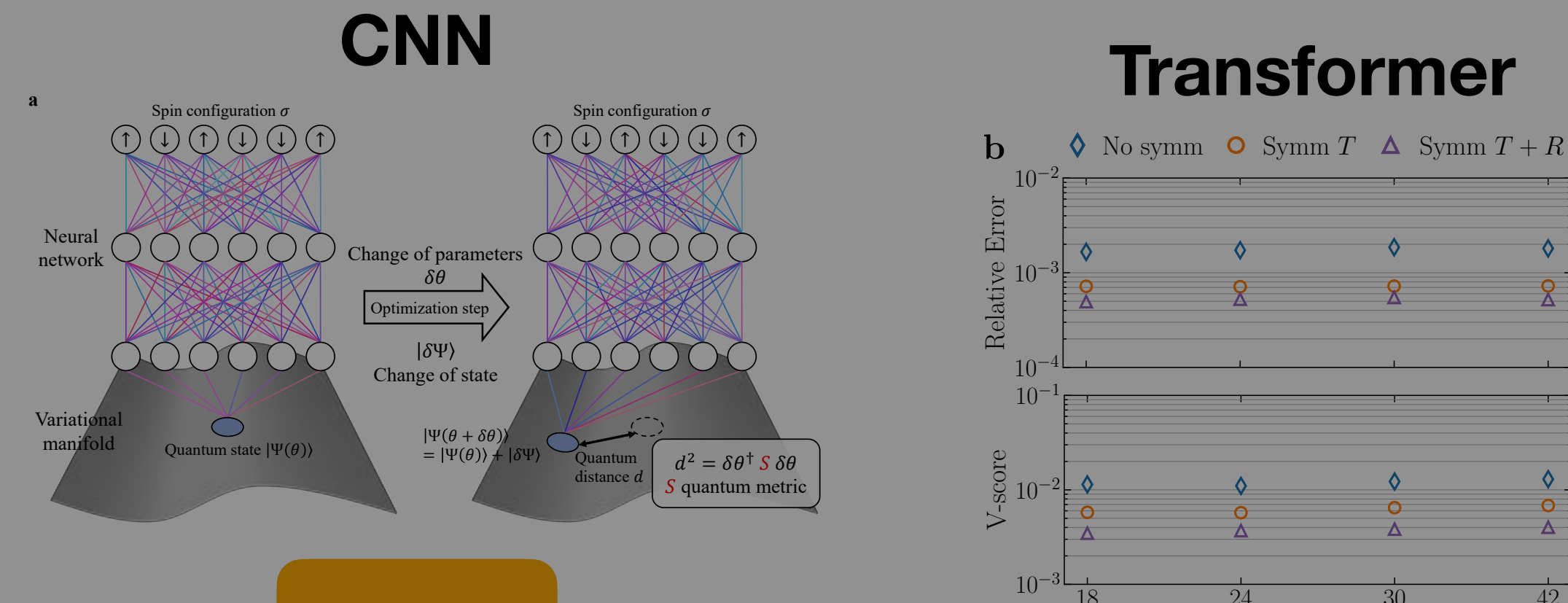
Viteritti et al. (2025)
Phys. Rev. B 111, 134411

42x42

Viteritti et al. (2026)
arXiv:2602.02665

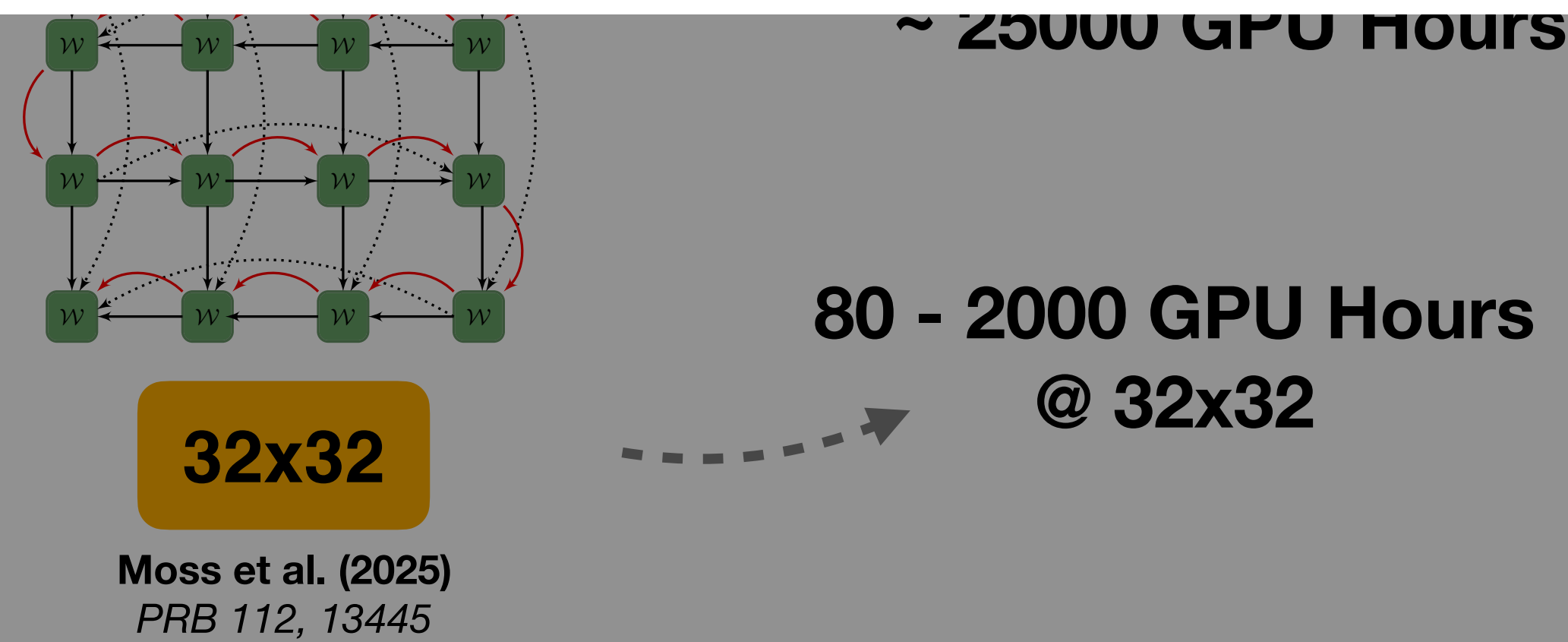
But also expensive...

- Recent attention- and CNN-based architectures:



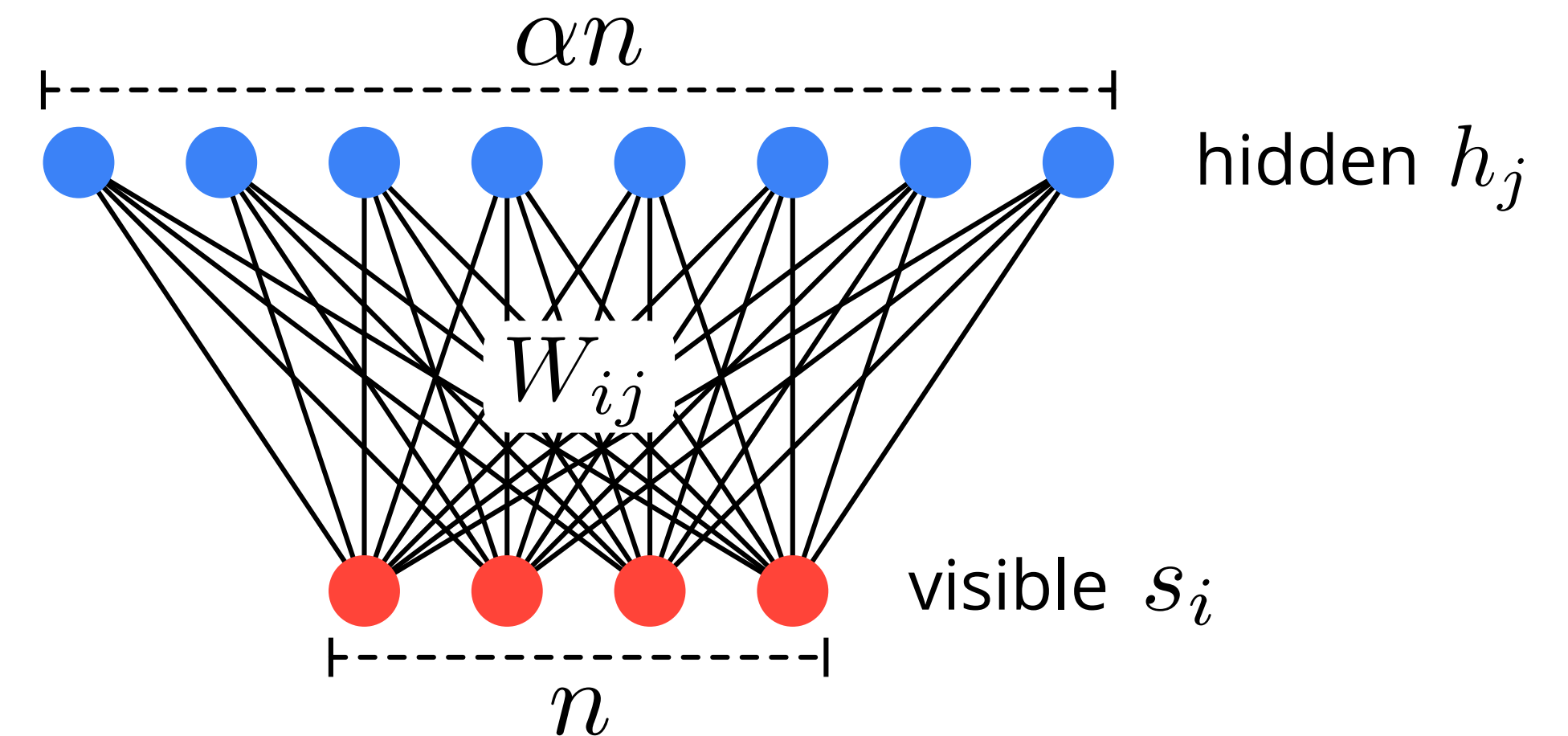
How to realize more scalable, large scale NQS?

- We need **10x** more spins, to scale to 100x100 can be **100x** more expensive!

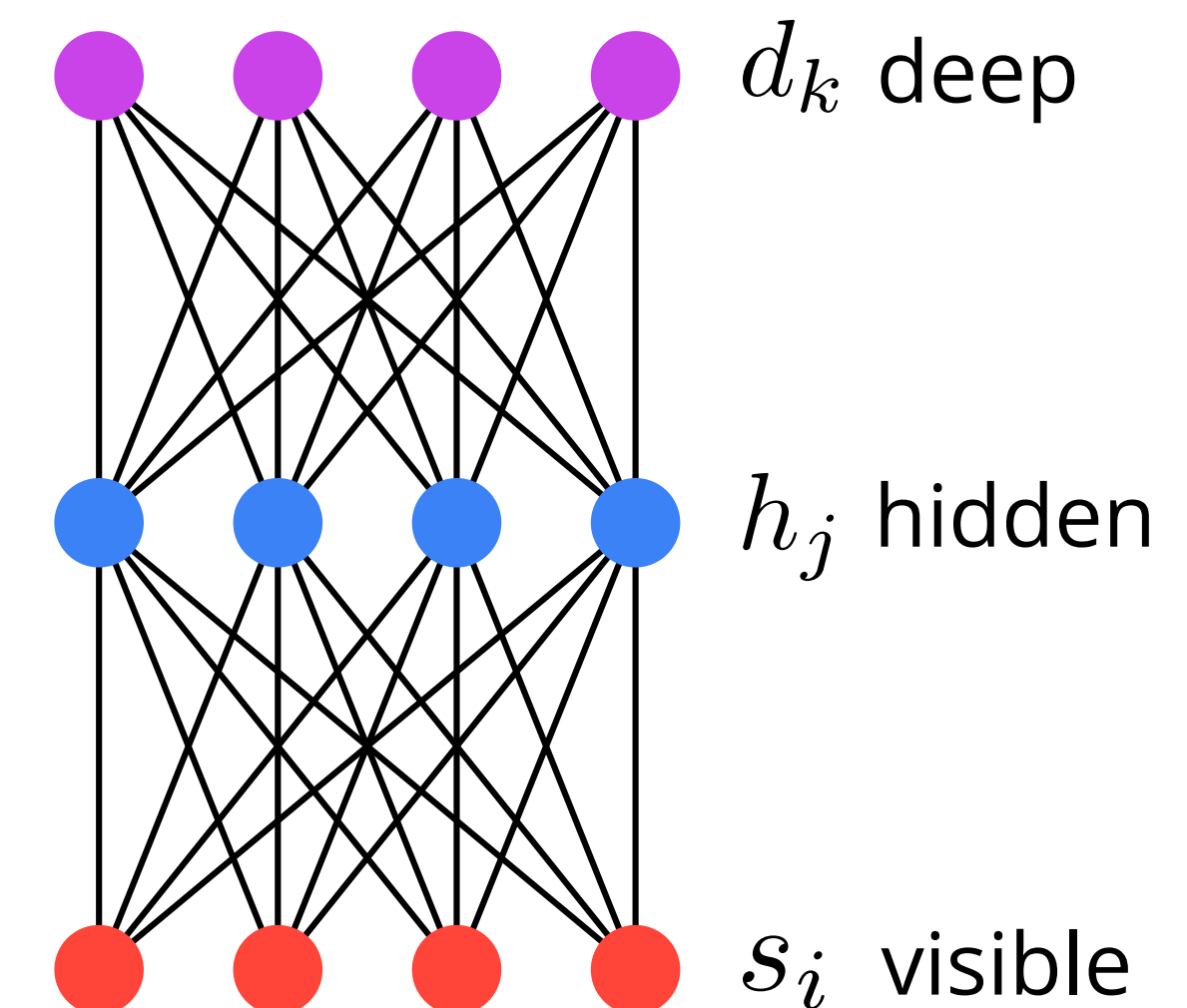


Boltzmann Machines

- Restricted Boltzmann Machine:
- Deep Boltzmann Machine:
 - More expressive & parameter-efficient
 - Nomura et al. (2021). *Phys. Rev. Let.*, 127(6), 060601.
 - Gao et al. (2017). *Nature Comm.*, 8(1), 662
 - Carleo et al. (2018). *Nature Comm.*, 9(1), 5322.
 - **Sparse Deep Boltzmann Machines**
 - Chowdhury et al. (2025). *arXiv:2512.24558 (for NQS)*
 - Niazi et al. (2024). *Nat. Elec.*, 7(7), 610–619 Montúfar (2015). *ICLR2015*
- But, hard to train!



Carleo & Troyer (2017) *Science* 355,602-606



Neural Quantum States

Variational technique

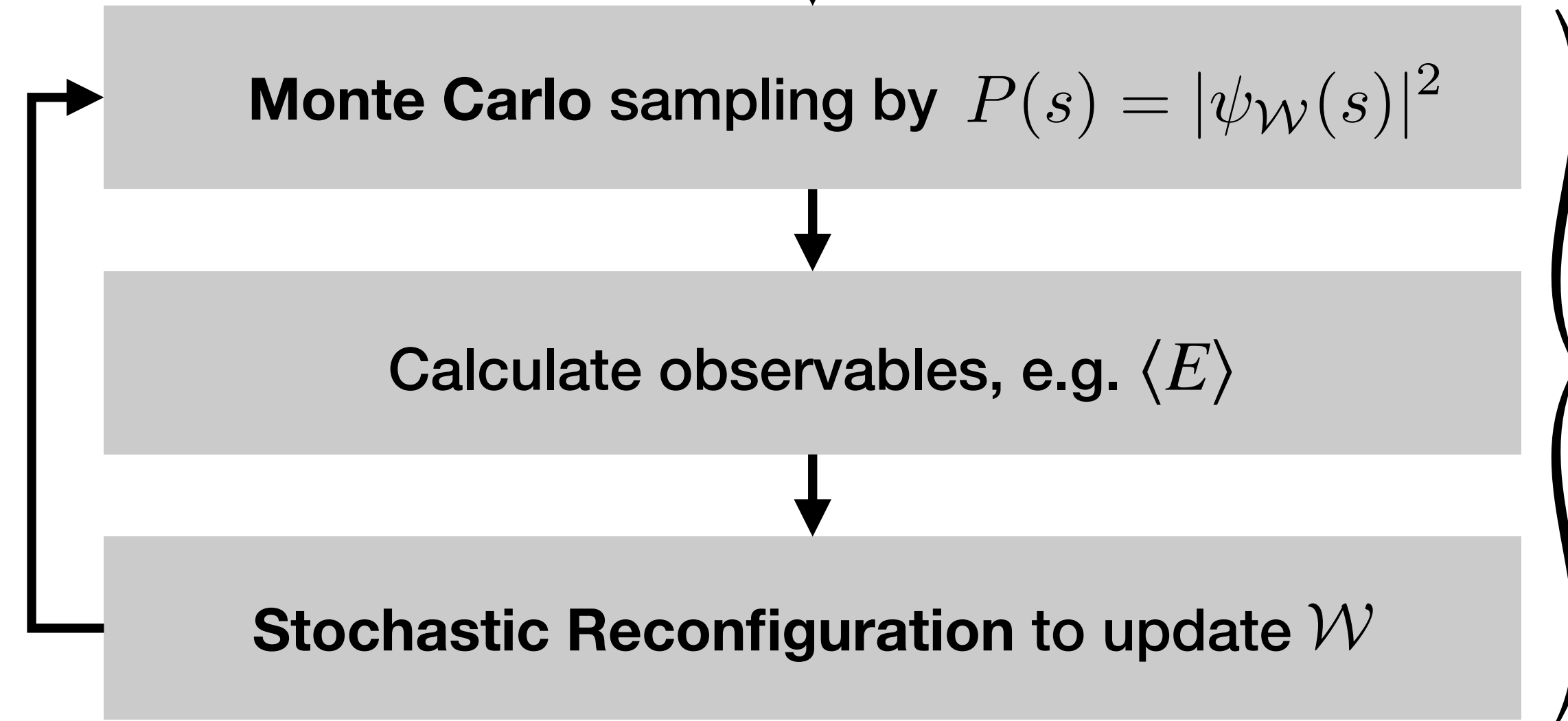
- Ansatz: $\psi_{\mathcal{W}}(s)$, solve: $\hat{H}|\psi\rangle = E|\psi\rangle$

Initialise with random weights $\mathcal{W}$

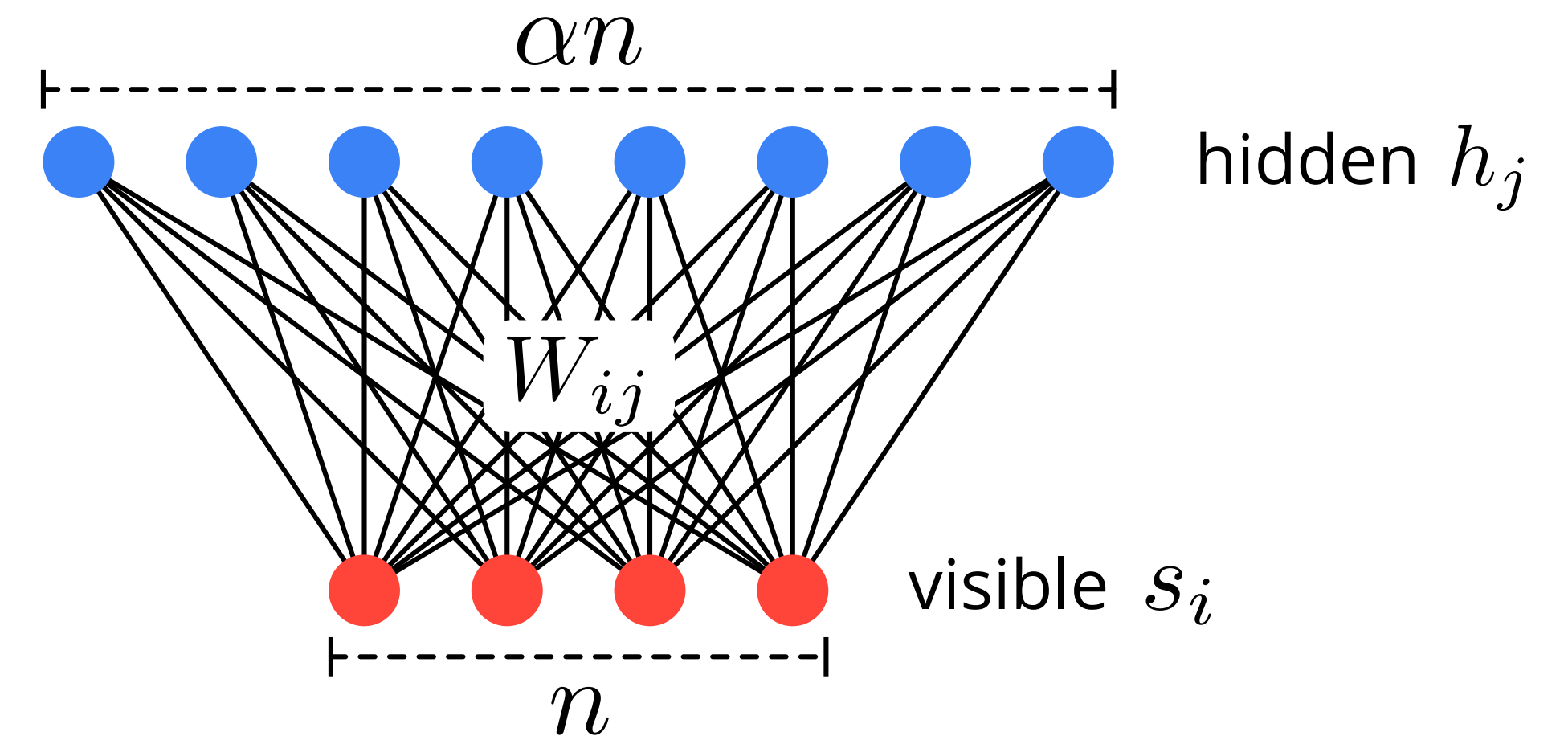
Monte Carlo sampling by $P(s) = |\psi_{\mathcal{W}}(s)|^2$

Calculate observables, e.g. $\langle E \rangle$

Stochastic Reconfiguration to update $\mathcal{W}$



1. Boltzmann Machines



Restricted Boltzmann Machine

Carleo & Troyer (2017) *Science* 355,602-606

Each step has to be **efficient to scale!**

Initialise with random weights $\mathcal{W}$

Monte Carlo sampling by $P(s) = |\psi_{\mathcal{W}}(s)|^2$

Observables:

$$\langle \mathcal{O} \rangle = \frac{\langle \psi_{\mathcal{W}} | \mathcal{O} | \psi_{\mathcal{W}} \rangle}{\langle \psi_{\mathcal{W}} | \psi_{\mathcal{W}} \rangle} \stackrel{\text{MC}}{\approx} \frac{1}{N_S} \sum_{i=1}^{N_S} \mathcal{O}_{\text{loc}}(s)$$

Metropolis-Hastings:

$$p(s \rightarrow s') = \min \left(1, \frac{|\psi(s')|^2}{|\psi(s)|^2} \right)$$

$\nearrow \mathcal{O}(n)$

Calculate observables, e.g. $\langle E \rangle$

Stochastic Reconfiguration to update $\mathcal{W}$

Initialise with random weights $\mathcal{W}$

Monte Carlo sampling by $P(s) = |\psi_{\mathcal{W}}(s)|^2$

Observables:

$$\langle \mathcal{O} \rangle = \frac{\langle \psi_{\mathcal{W}} | \mathcal{O} | \psi_{\mathcal{W}} \rangle}{\langle \psi_{\mathcal{W}} | \psi_{\mathcal{W}} \rangle} \stackrel{\text{MC}}{\approx} \frac{1}{N_S} \sum_{i=1}^{N_S} \mathcal{O}_{\text{loc}}(s)$$

Metropolis-Hastings:

$$p(s \rightarrow s') = \min \left(1, \frac{|\psi(s')|^2}{|\psi(s)|^2} \right)$$

$\nearrow \mathcal{O}(n)$

Calculate observables, e.g. $\langle E \rangle$

Gradients:

$$O_k(s) = \frac{\partial}{\partial \mathcal{W}_k} \log \psi_{\mathcal{W}}(s)$$

Local energy: $\nearrow \mathcal{O}(n)$

$$E_{\text{loc}}(s) = \sum_{s'} \langle s | \hat{H} | s' \rangle \frac{\psi_M(s')}{\psi_M(s)}$$

Stochastic Reconfiguration to update $\mathcal{W}$

GPU Acceleration

Initialise with random weights $\mathcal{W}$

Monte Carlo sampling by $P(s) = |\psi_{\mathcal{W}}(s)|^2$

Observables:

$$\langle \mathcal{O} \rangle = \frac{\langle \psi_{\mathcal{W}} | \mathcal{O} | \psi_{\mathcal{W}} \rangle}{\langle \psi_{\mathcal{W}} | \psi_{\mathcal{W}} \rangle} \stackrel{\text{MC}}{\approx} \frac{1}{N_S} \sum_{i=1}^{N_S} \mathcal{O}_{\text{loc}}(s)$$

Metropolis-Hastings:

$$p(s \rightarrow s') = \min \left(1, \frac{|\psi(s')|^2}{|\psi(s)|^2} \right)$$

$\nearrow \mathcal{O}(n)$

Calculate observables, e.g. $\langle E \rangle$

Gradients:

$$O_k(s) = \frac{\partial}{\partial \mathcal{W}_k} \log \psi_{\mathcal{W}}(s)$$

Local energy: $\nearrow \mathcal{O}(n)$

$$E_{\text{loc}}(s) = \sum_{s'} \langle s | \hat{H} | s' \rangle \frac{\psi_M(s')}{\psi_M(s)}$$

Stochastic Reconfiguration to update $\mathcal{W}$

$$S_{kk'} = \langle O_k^* O_{k'} \rangle - \langle O_k^* \rangle \langle O_{k'} \rangle$$

$$\mathcal{F}_k = \langle E_{\text{loc}} O_k^* \rangle - \langle E_{\text{loc}} \rangle \langle O_k^* \rangle$$

Update:

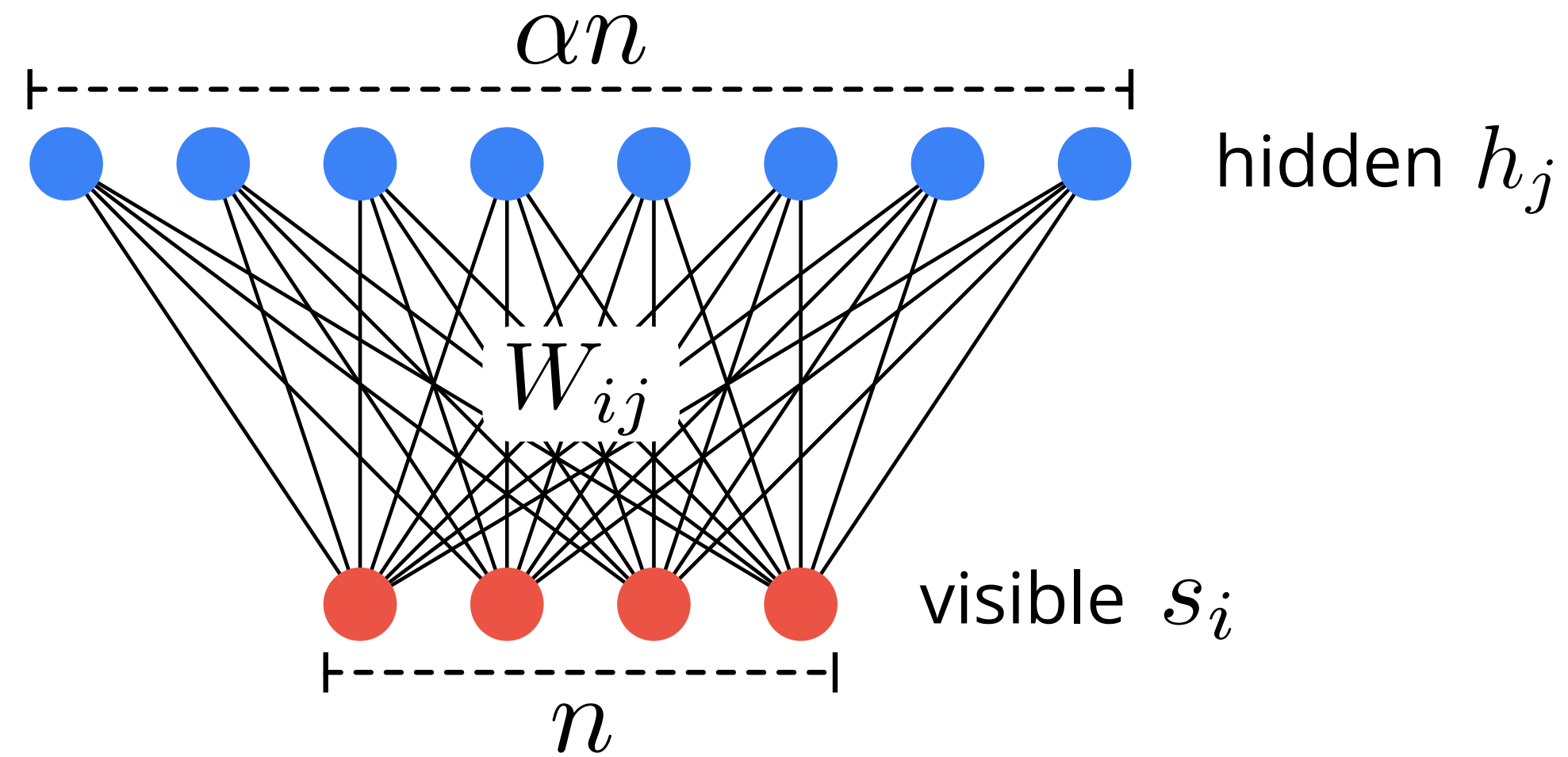
$$\mathcal{W}_k(p+1) = \mathcal{W}_k(p) - \eta(p) S_{kk'}^{-1} \mathcal{F}_k(\mathcal{W}_k(p))$$

Or minSR: $\mathcal{O}(N_S^3)$

GPU Acceleration

Outline

1. Restricted Boltzmann Machines



2. GPU Acceleration

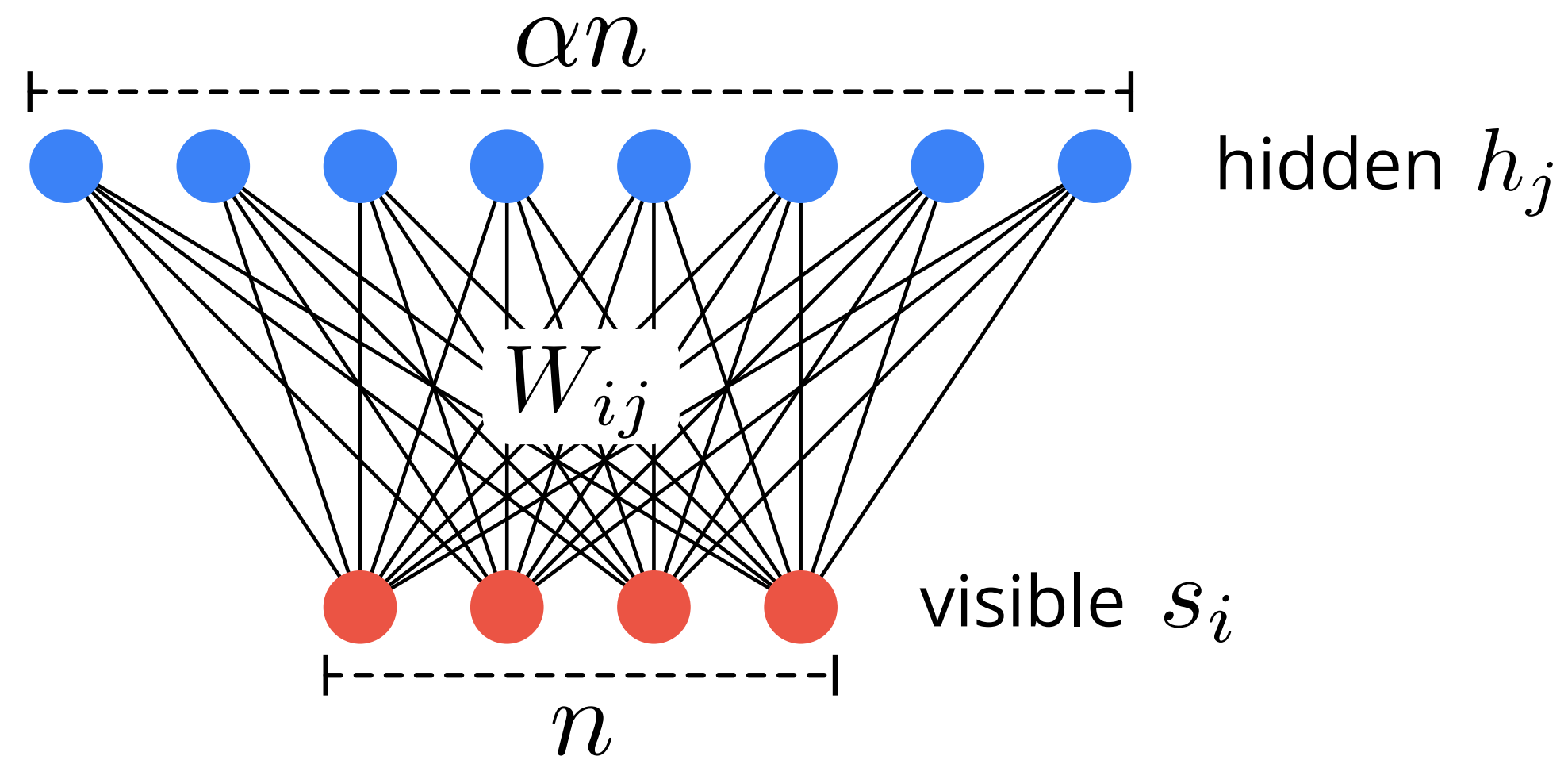
Scale up to 10.000 quantum spins

3. Neuromorphic computing

Scale to millions quantum spins

Outline

1. Restricted Boltzmann Machines



2. GPU Acceleration

Scale up to 10.000 quantum spins

3. Neuromorphic computing

Scale to millions quantum spins

Boltzmann Machines

- Variational ansatz:

$$\psi(s) = \sqrt{\sum_h e^{\sum_i^{\alpha n} b_i h_i + \sum_{i,j}^{\alpha n, n} W_{ij} h_i s_j}}$$

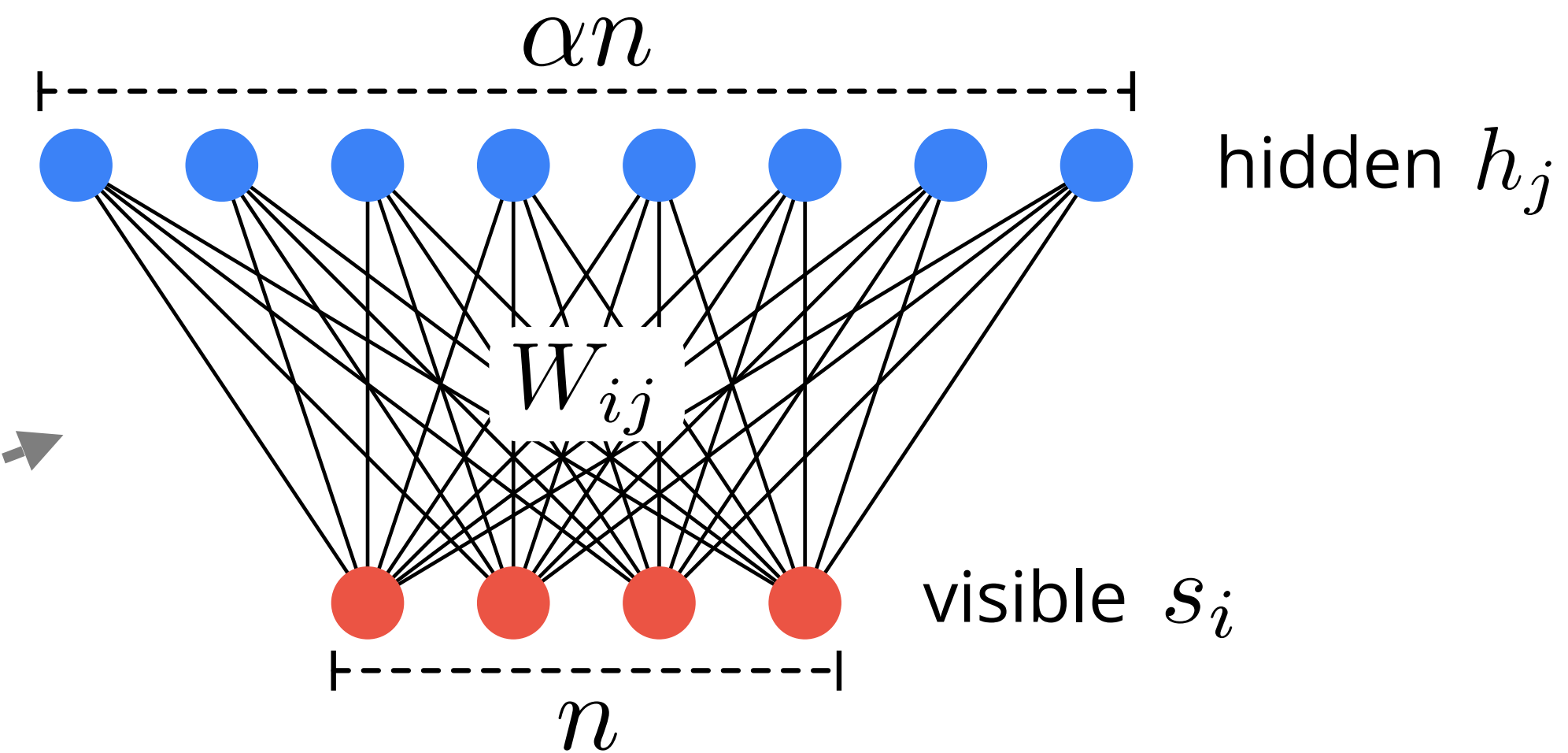
- Successful architecture:

Carleo & Troyer (2017) *Science* 355,602-606

Nomura, Imada (2021) *Phys. Rev. X* 11, 031034

Joshi et al. (2024) *Phys. Rev. B* 110, 104411

- Efficient inference/updates: $\mathcal{O}(n^3) \rightarrow \mathcal{O}(n^2)$



Carleo & Troyer (2017) *Science* 355,602-606

•

Boltzmann Machines: Efficient updates

- Efficient inference/updates:
 - Sampling & off-diagonal observables requires: $\frac{\psi(s')}{\psi(s)}$

$$\psi(s) = \sqrt{\sum_h e^{\sum_i^{\alpha n} b_i h_i + \sum_{i,j}^{\alpha n, n} W_{ij} h_i s_j}}$$

↓ analytically sum hidden units

$$\psi(s) = \sqrt{\prod_{i=1}^{\alpha n} 2 \cosh \left(\underbrace{b_i + \sum_{j=1}^n W_{ij} s_j}_{\text{update!}} \right)}$$

$$\mathcal{O}(\alpha n^2) \rightarrow \mathcal{O}(\alpha n)$$

Boltzmann Machines: Symmetries

- Translation invariance reduces trainable parameters by $\mathcal{O}(\alpha n^2 + \alpha n) \rightarrow \mathcal{O}(\alpha n + \alpha)$

$$\psi(s) = \sqrt{\sum_h e^{\sum_i^{\alpha n} b_i h_i + \sum_{i,j}^{\alpha n, n} W_{ij} h_i s_j}}$$

Carleo & Troyer (2017) *Science* 355,602-606

Fabiani et al. (2019) *SciPost Physics*, 7(1), 004

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
4	5	3	7	8	6	10	11	9	13	14	12	16	17	15	19	20	18	22	23	21	25	26	24
5	3	4	8	6	7	11	9	10	14	12	13	17	15	16	20	18	19	23	21	22	26	24	25
6	7	8	9	10	11	12	13	14	3	4	5	18	19	20	21	22	23	24	25	26	15	16	17
7	8	6	10	11	9	13	14	12	4	5	3	19	20	18	22	23	21	25	26	24	16	17	15
8	6	7	11	9	10	14	12	13	5	3	4	20	18	19	23	21	22	26	24	25	17	15	16
9	10	11	12	13	14	3	4	5	6	7	8	21	22	23	24	25	26	15	16	17	18	19	20
10	11	9	13	14	12	4	5	3	7	8	6	22	23	21	25	26	24	16	17	15	19	20	18
11	9	10	14	12	13	5	3	4	8	6	7	23	21	22	26	24	25	17	15	16	20	18	19
12	13	14	3	4	5	6	7	8	9	10	11	24	25	26	15	16	17	18	19	20	21	22	23
13	14	12	4	5	3	7	8	6	10	11	9	25	26	24	16	17	15	19	20	18	22	23	21
14	12	13	5	3	4	8	6	7	11	9	10	26	24	25	17	15	16	20	18	19	23	21	22

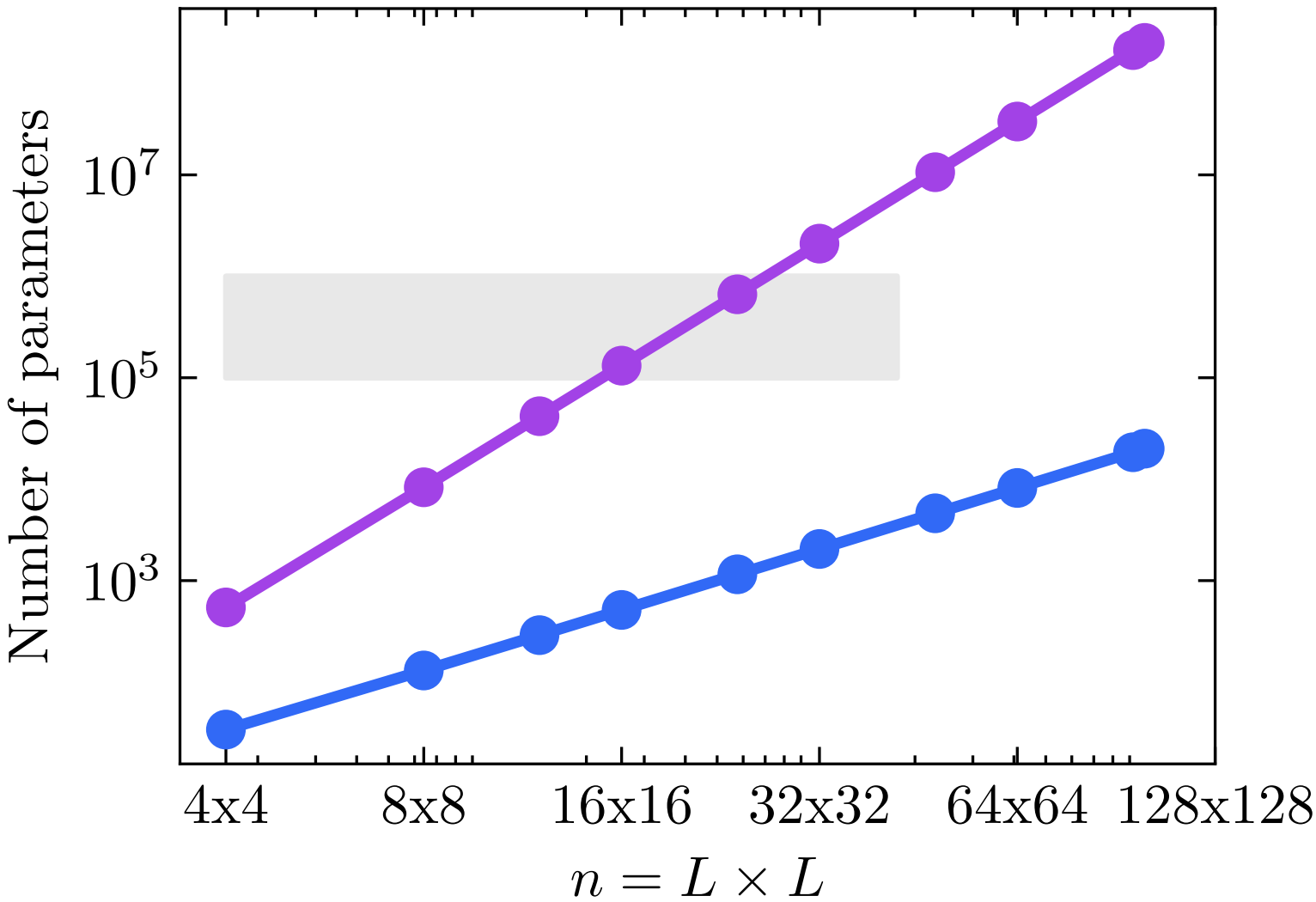
- Improves trainability

Nomura (2021) *J. Phys.: Condens. Matter* 33, 174003

- For $\alpha = 2$ only **144 KB** or **18434** parameters for 96x96 spins

➔ Almost 10.000 times less parameters

➔ **Small SMatrix!**

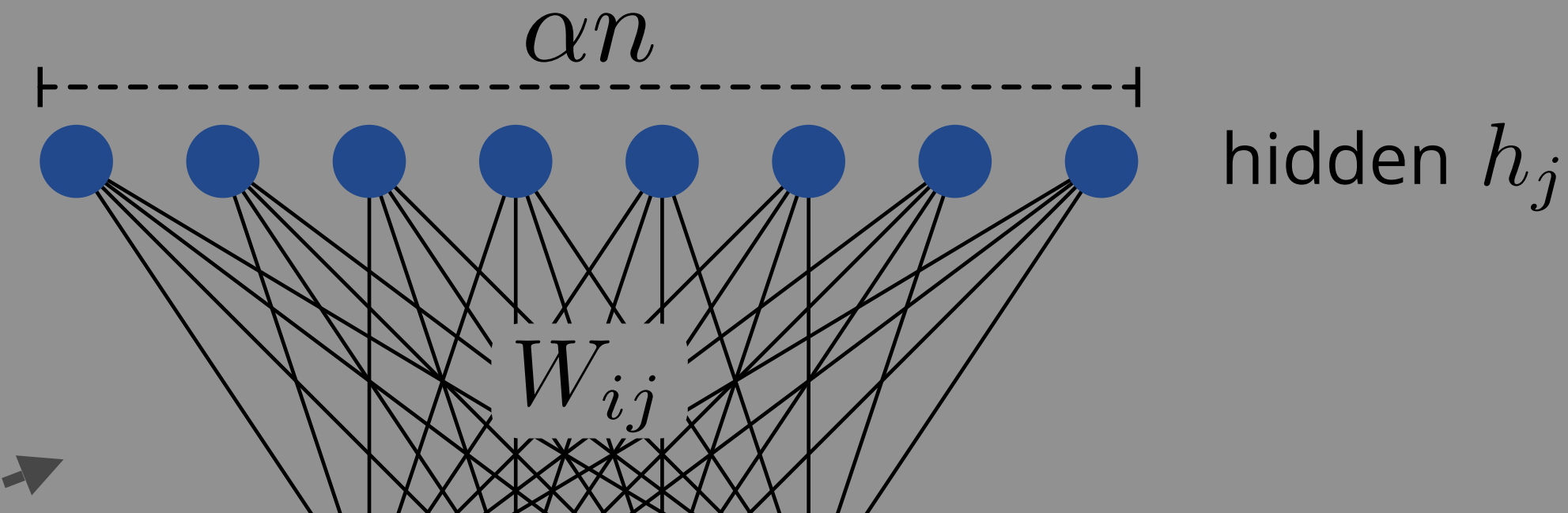


—●— Translation invariant
 —●— No translation invariance
 Typical transformer/CNN range

Boltzmann Machines

- Variational ansatz:

$$\psi(s) = \sqrt{\sum_h e^{\sum_i^{\alpha n} b_i h_i + \sum_{i,j}^{\alpha n, n} W_{ij} h_i s_j}}$$

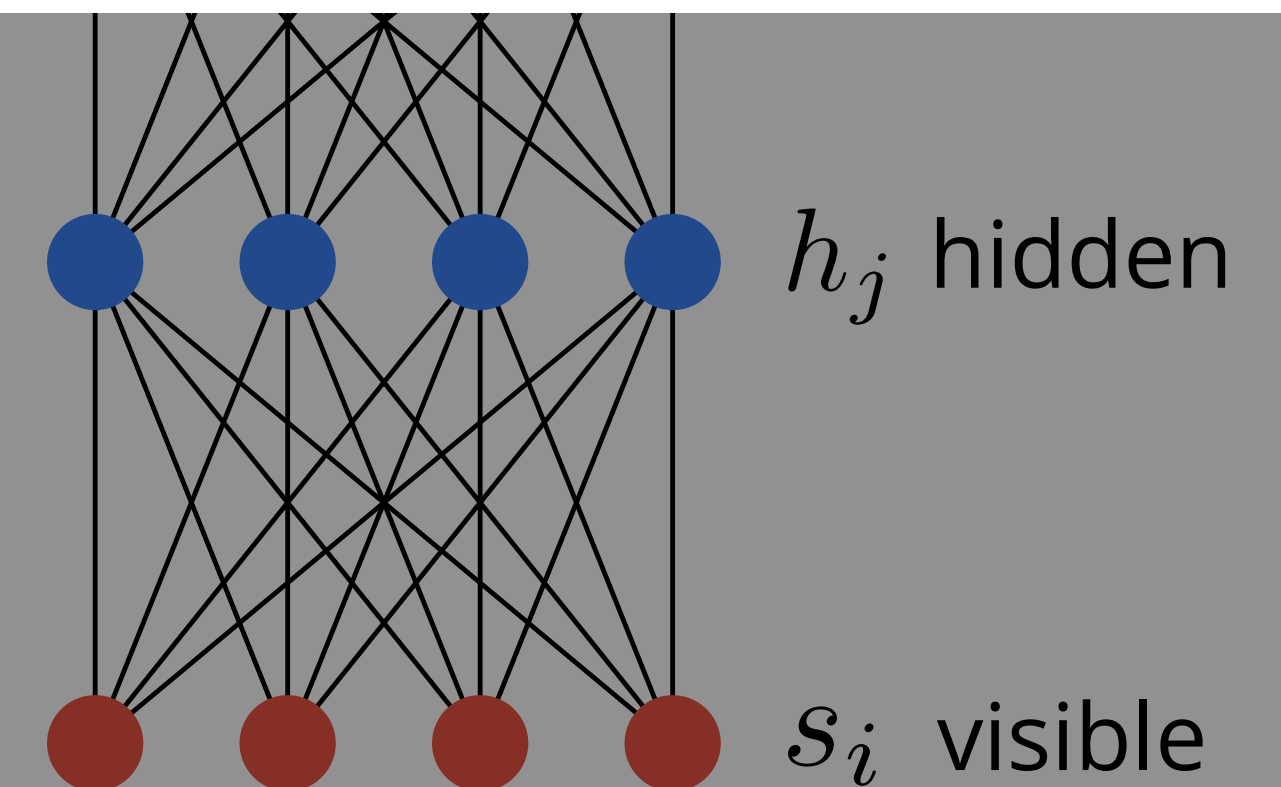


Training large RBMs is hard!

- Symmetries: translation Invariance

$$\mathcal{O}(\alpha n^2 + \alpha n) \rightarrow \mathcal{O}(\alpha n + \alpha)$$

- Outlook to Deep Boltzmann Machines

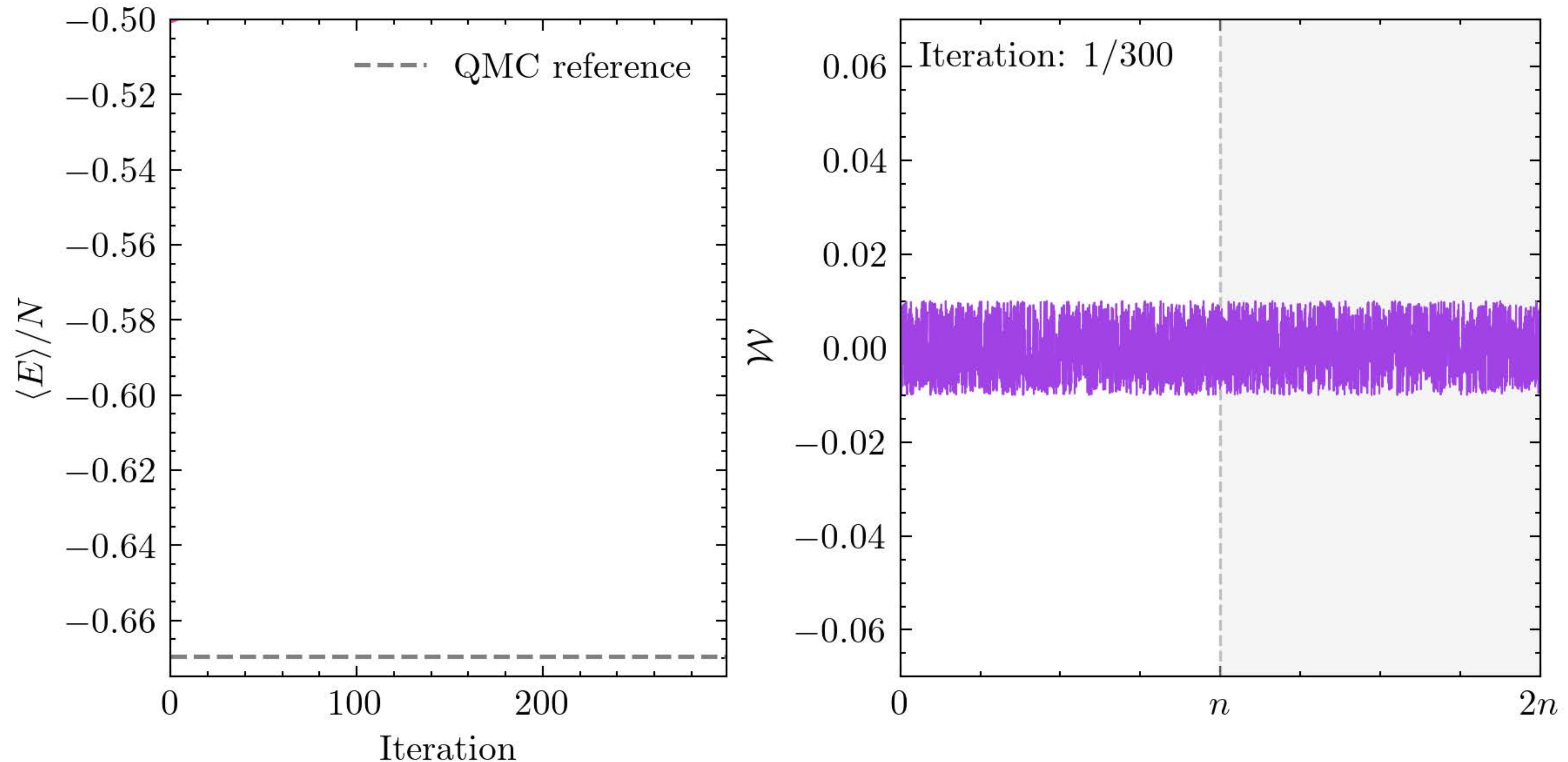


Carleo et al. (2018). *Nature Comm.*, 9(1), 5322.

Nomura et al. (2021). *Phys. Rev. Let.*, 127(6), 060601.

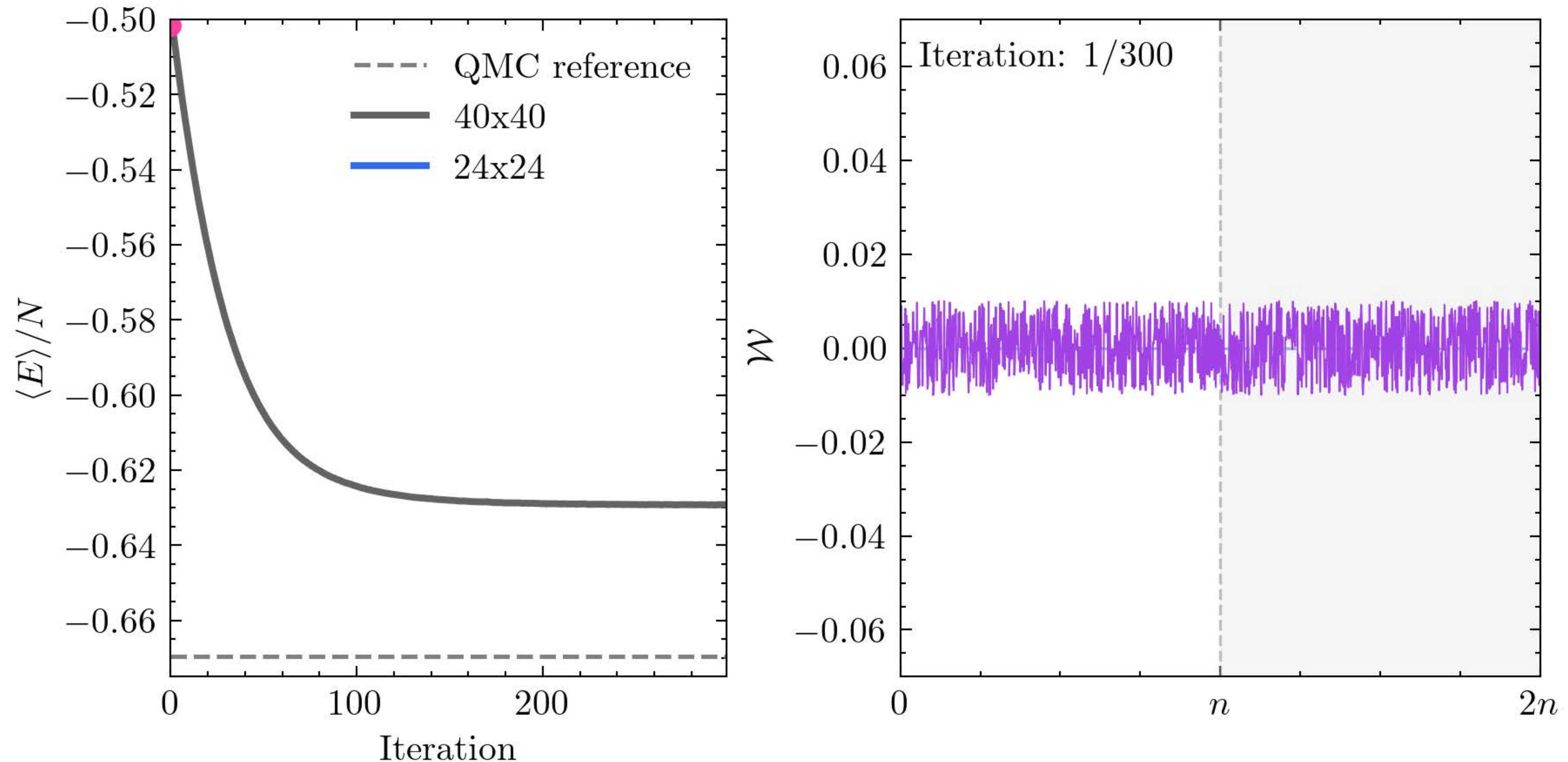
RBM hard to train for big systems

Antiferromagnetic Heisenberg model, $L = 40 \times 40$, $\alpha = 2$



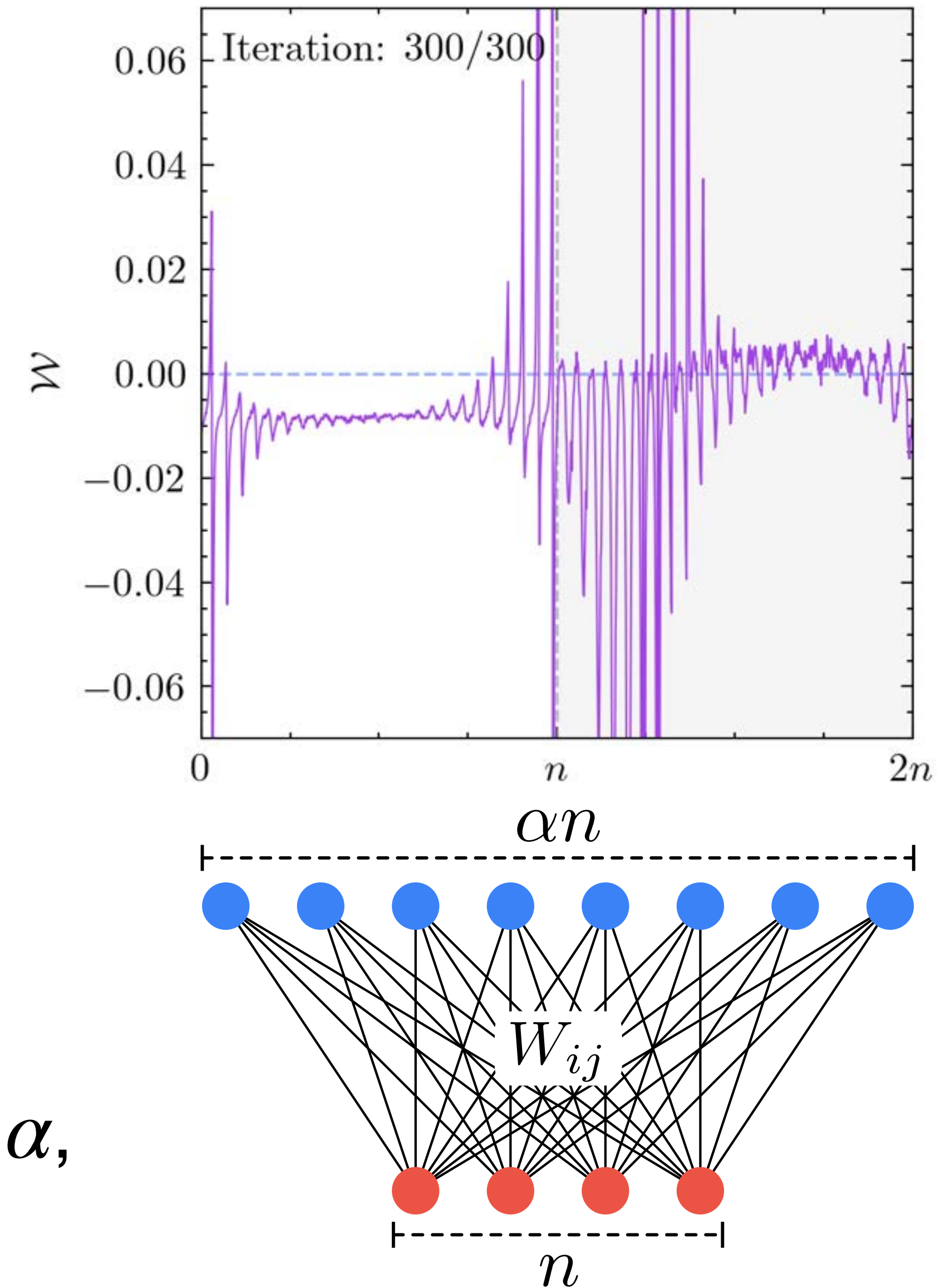
RBM hard to train for big systems

Antiferromagnetic Heisenberg model, $L = 24 \times 24$, $\alpha = 2$



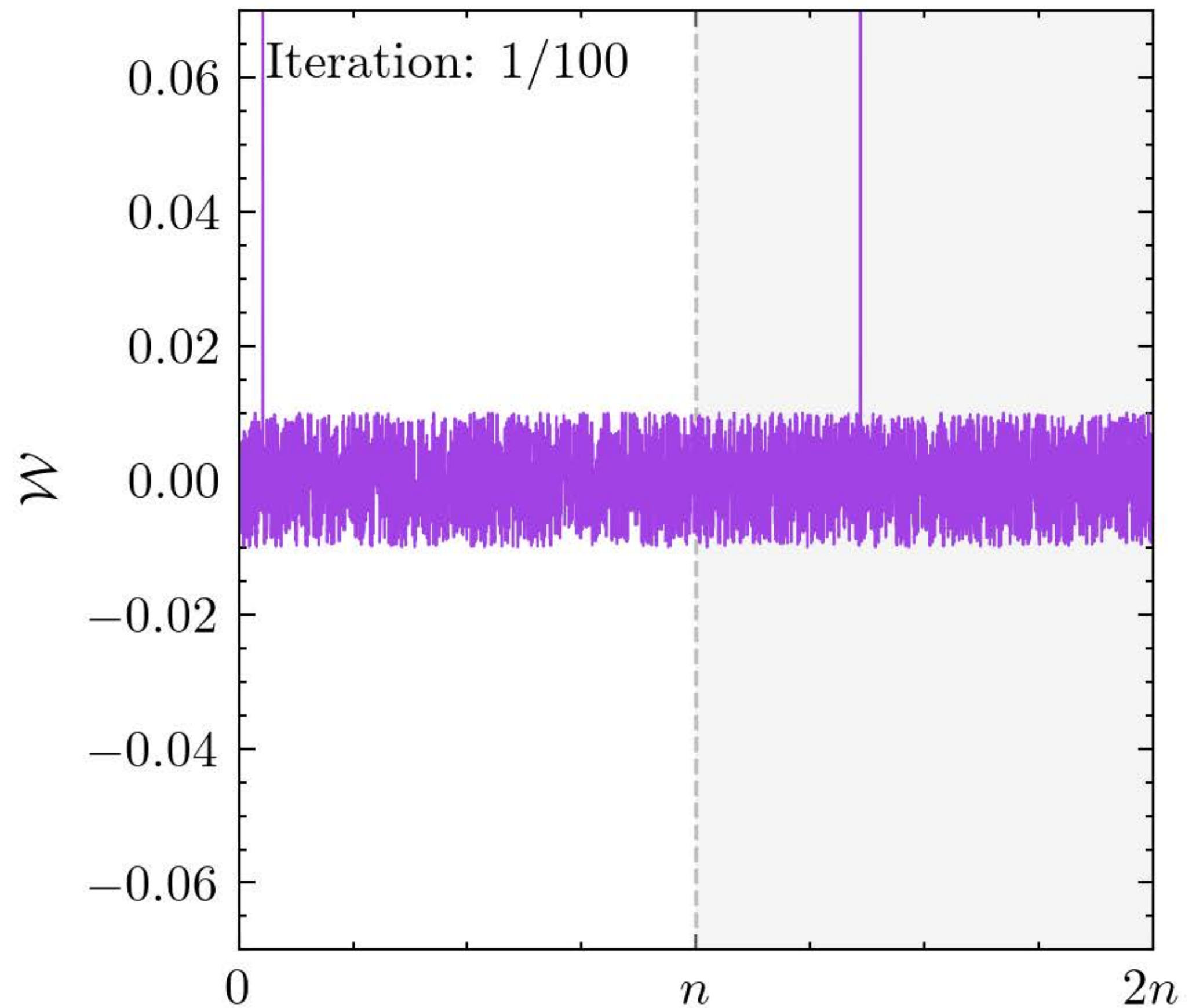
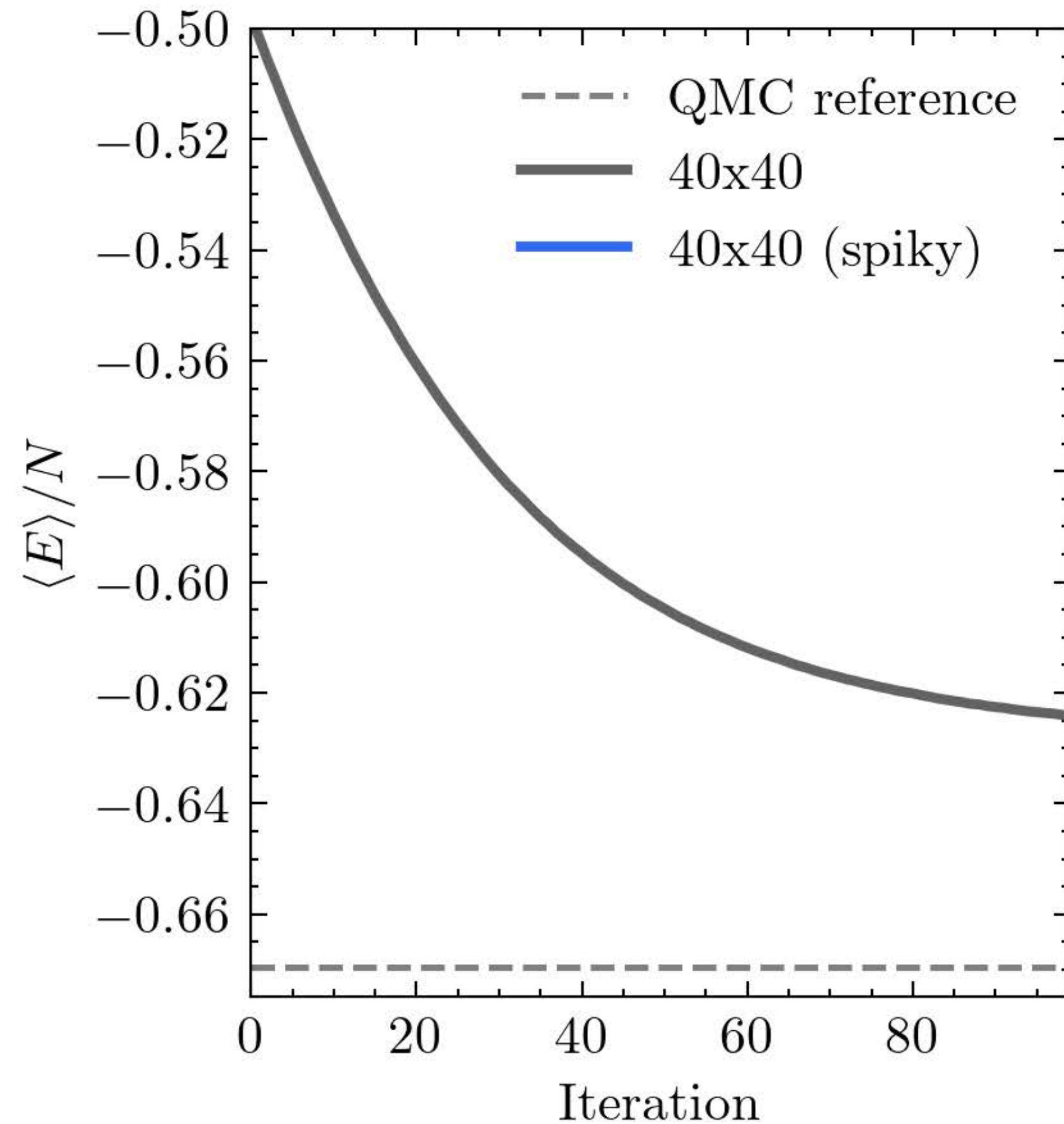
Structured weights

- **For big systems, training gets stuck!**
- Typical initialization: $\mathcal{W}_0 \sim U(-10^{-2}, 10^{-2})$
- **New initialization:**
 - Two variances: $\mathcal{W}_{\text{LOW}} \sim U(-10^{-2}, 10^{-2})$ and $\mathcal{W}_{\text{HIGH}} \sim \mathcal{N}(0,1)$
 - Initialize all with $\mathcal{W}_{\text{LOW}}$, except for **1 weight** per α , $\mathcal{W}_{\text{HIGH}}$, like ‘spikes’



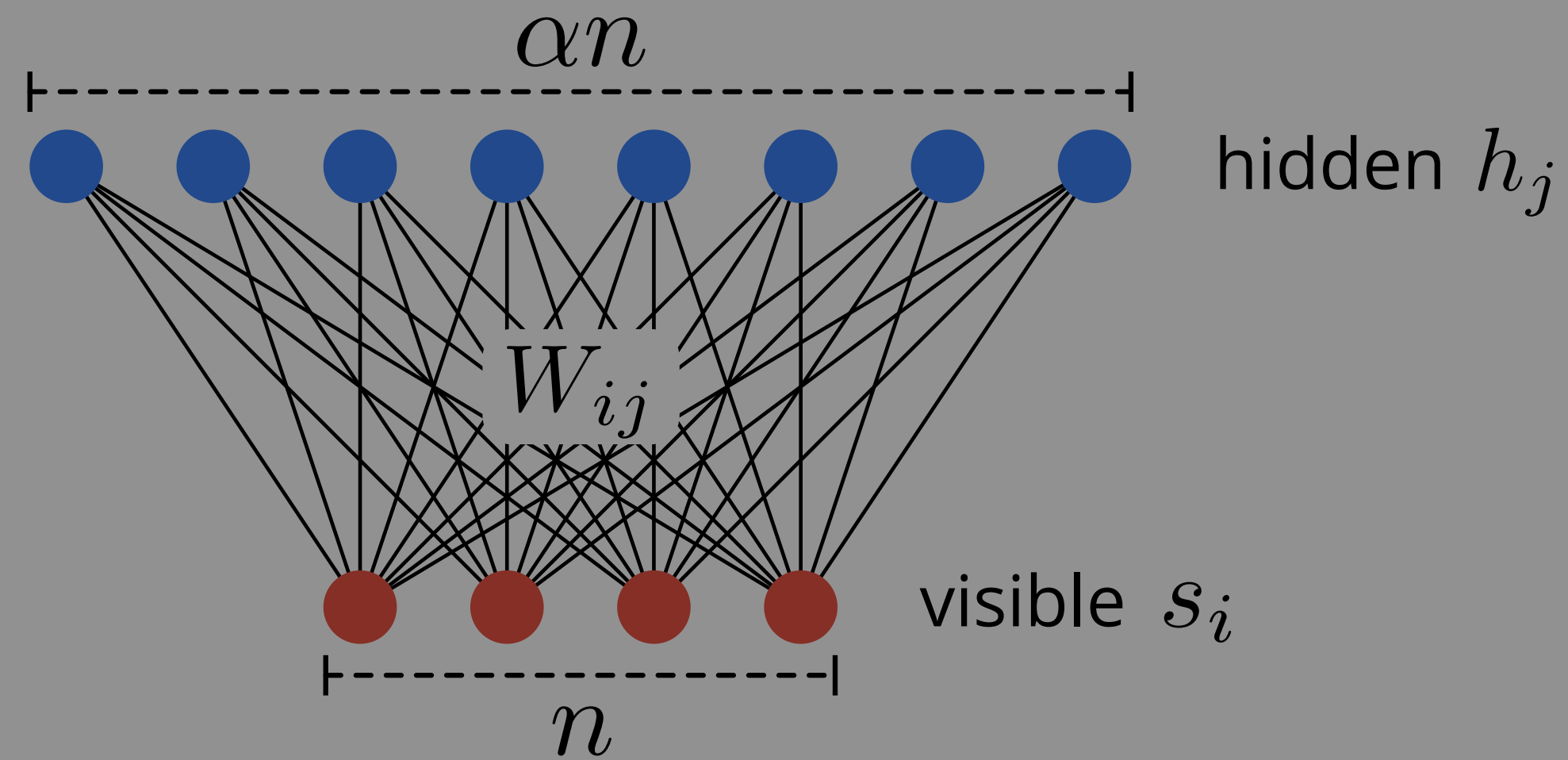
New ‘spiky’ initialization

Antiferromagnetic Heisenberg model, $L = 40 \times 40$, $\alpha = 2$



Outline

1. Restricted Boltzmann Machines



2. GPU Acceleration

Scale up to 10.000 quantum spins

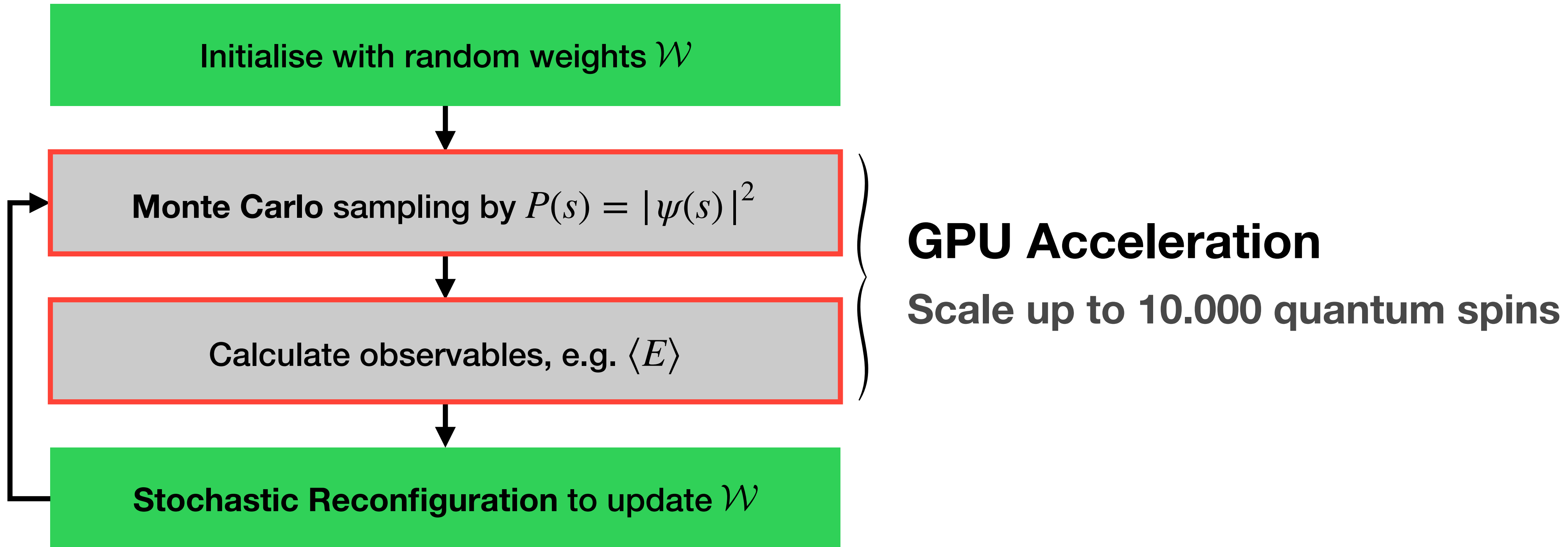
3. Neuromorphic computing

Scale to millions quantum spins

Neural Quantum States

Variational technique

- Ansatz: $\psi_{\mathcal{W}}(s)$, solve: $\hat{H}|\psi\rangle = E|\psi\rangle$



Local energies on GPU

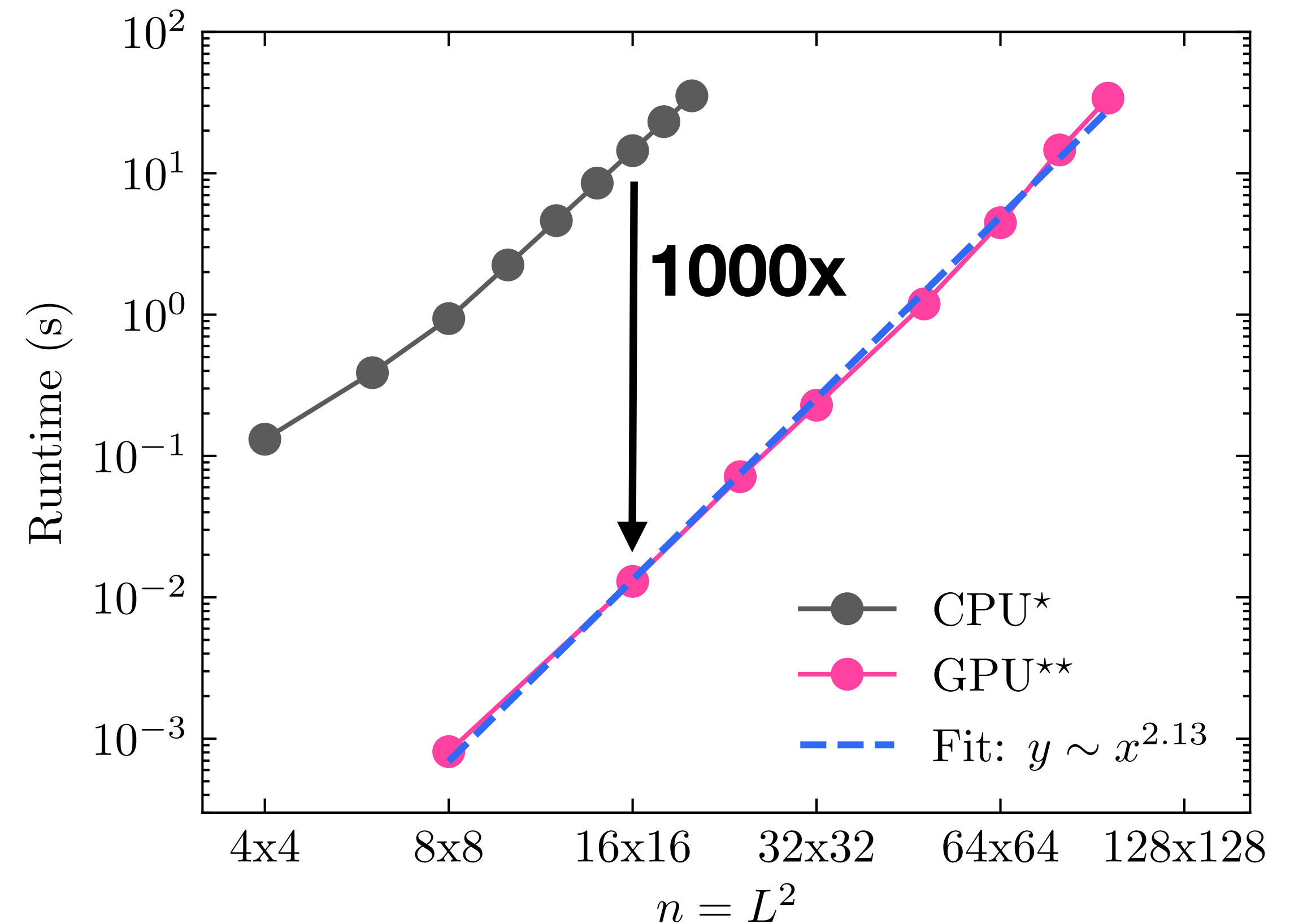
- Parallelization on GPU with Julia & CUDA
- Each GPU thread calculates wavefunction ratio:

$$E_{\text{loc}}(s) = \sum_{\langle i,j \rangle} J s_i s_j - 2 \sum_{\langle i,j \rangle} J \frac{\psi(s')}{\psi(s)}$$

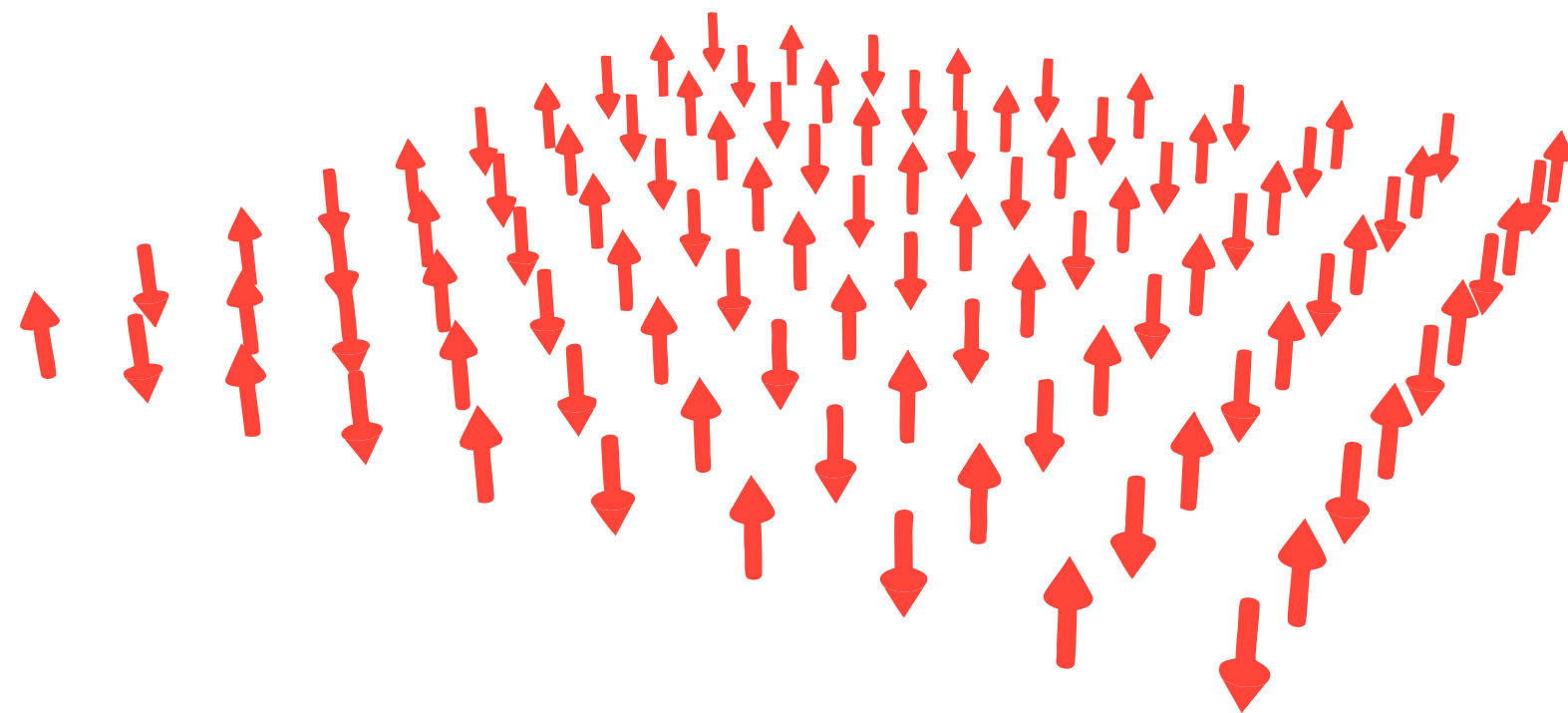
- Use cached: $\theta_i = b_i + \sum_{j=1}^n W_{ij} s_j$

- 1000x speed-up on GPU**

Local energy calculation 16384 samples



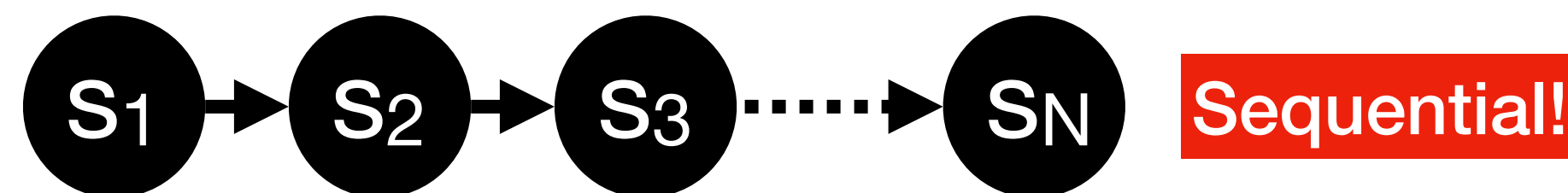
Sampling



Metropolis-Hastings:

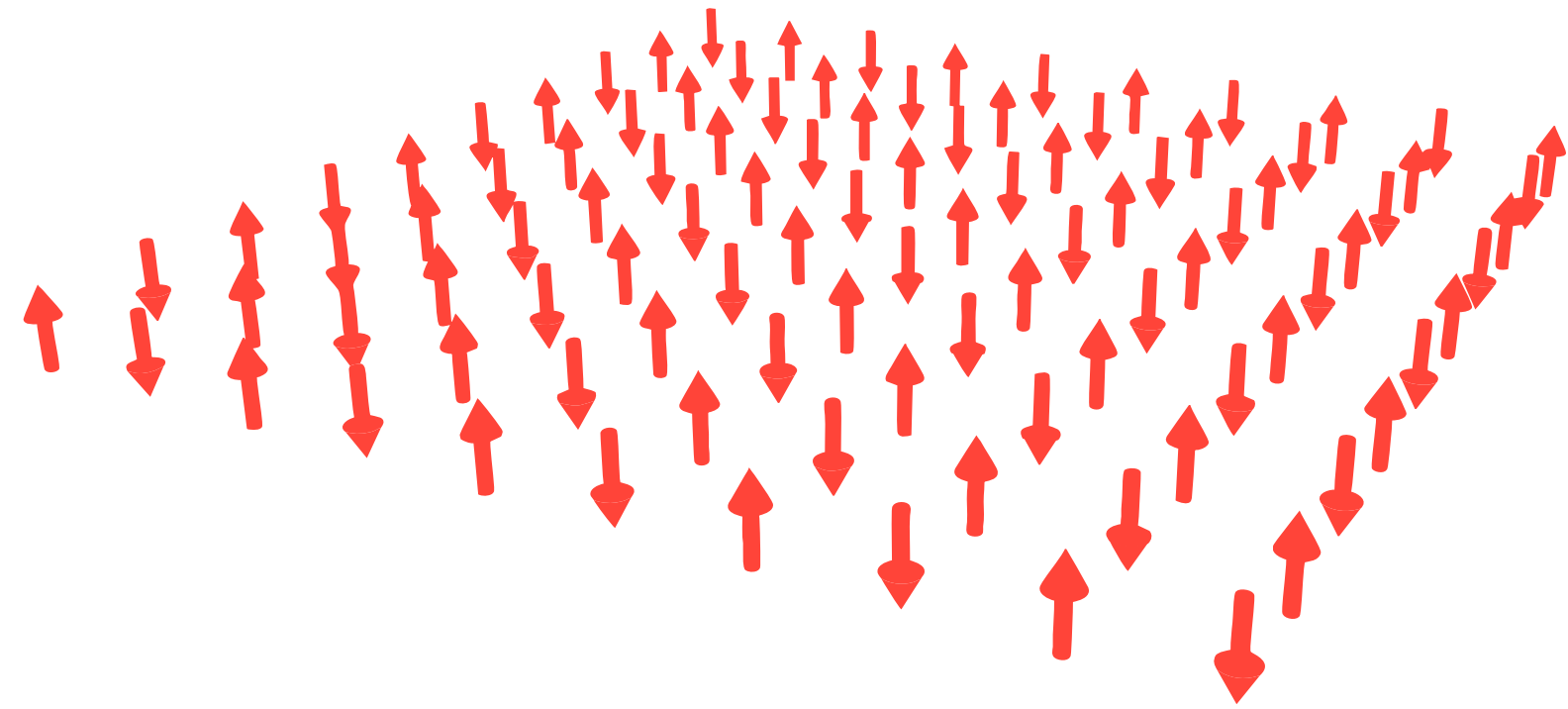
$$p(s \rightarrow s') = \min \left(1, \frac{|\psi(s')|^2}{|\psi(s)|^2} \right)$$

Only one/two flips at a time

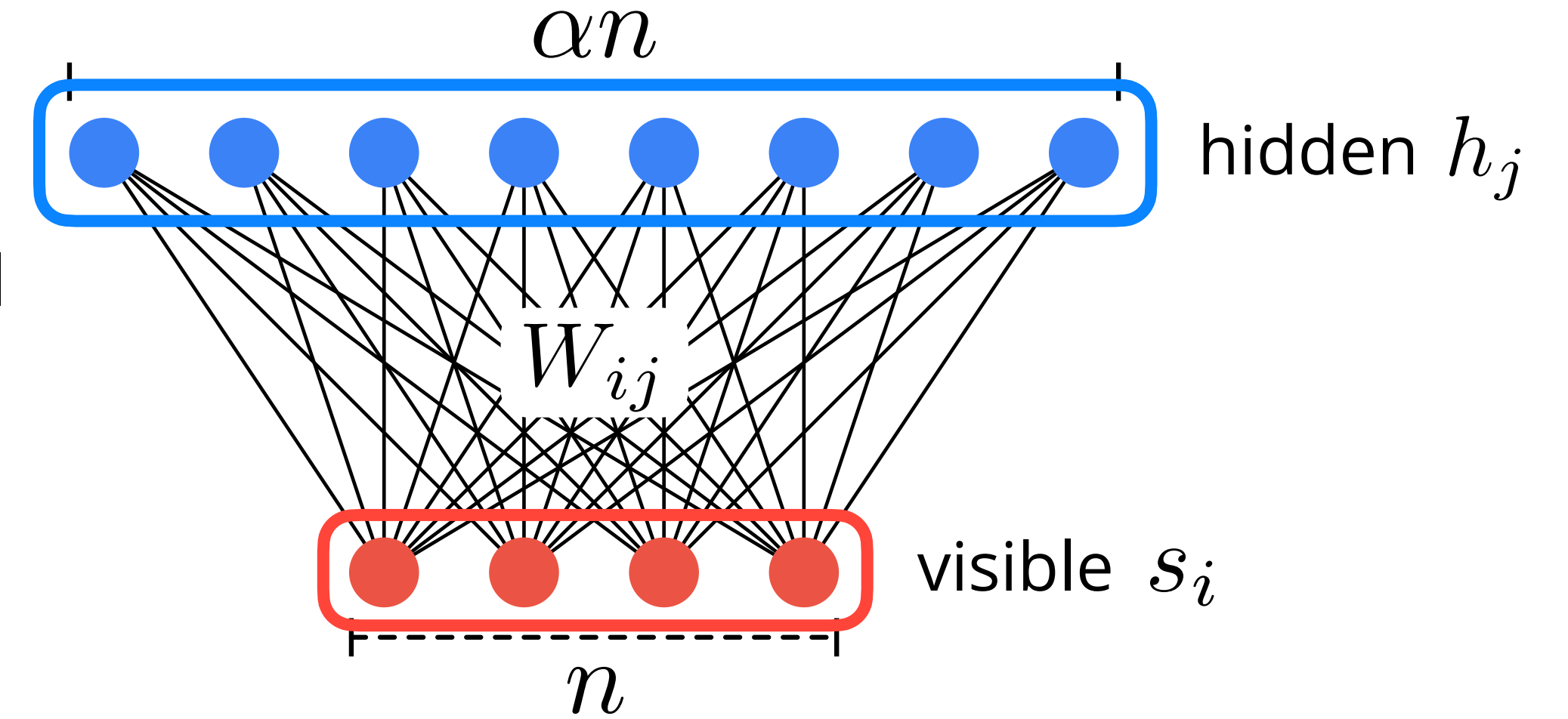


Large quantum systems are hard to study!

Sampling



Map to Ising Model



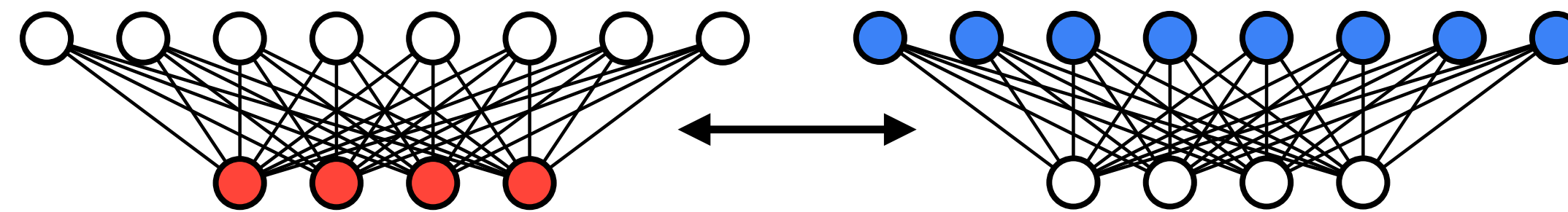
$$\psi(s) = \sqrt{\sum_h e^{\sum_i^{\alpha n} b_i h_i + \sum_{i,j}^{\alpha n, n} W_{ij} h_i s_j}}$$

$$|\psi(s)|^2 = P(s) = \sum_h e^{-H_{\text{ising}}(h,s)} \quad m = [h, s]$$

$$H_{\text{ising}}(m) = - \sum_{i=1}^{\alpha n + n} h_i m_i - \sum_{j=1}^{\alpha n + n} \sum_{i=1}^{j-1} J_{ij} m_i m_j$$

Chromatic Gibbs sampling

Parallel updating of all hidden and visible spins!



$$p(m_i = \pm 1) = \frac{1}{1 + \exp(\pm \Delta E)}$$

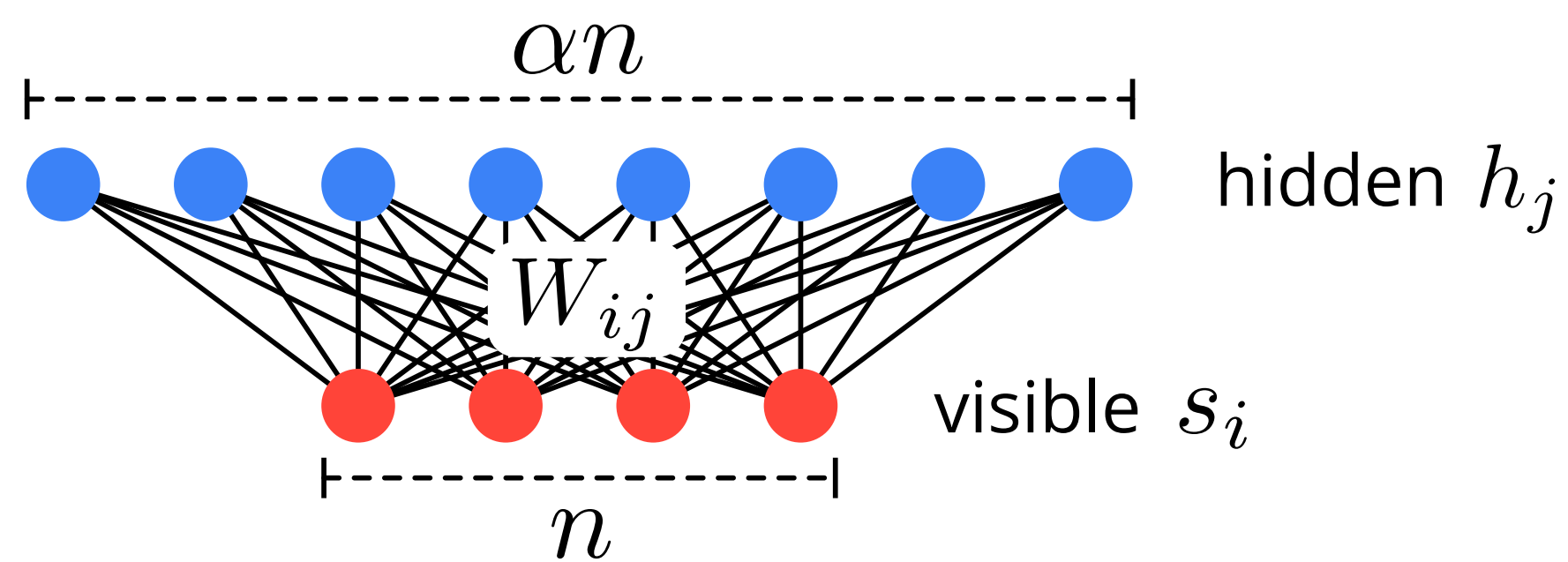
$$\Delta E = H(\dots s_i \dots) - H(\dots -s_i \dots) \stackrel{s}{=} 2s_i \sum_{j=1}^{\alpha n} W_{ij} h_j$$

$$H_{\text{ising}}(m) = - \sum_{i=1}^{\alpha n + n} h_i m_i - \sum_{j=1}^{\alpha n + n} \sum_{i=1}^{j-1} J_{ij} m_i m_j$$

Encoding magnetization constraint is difficult

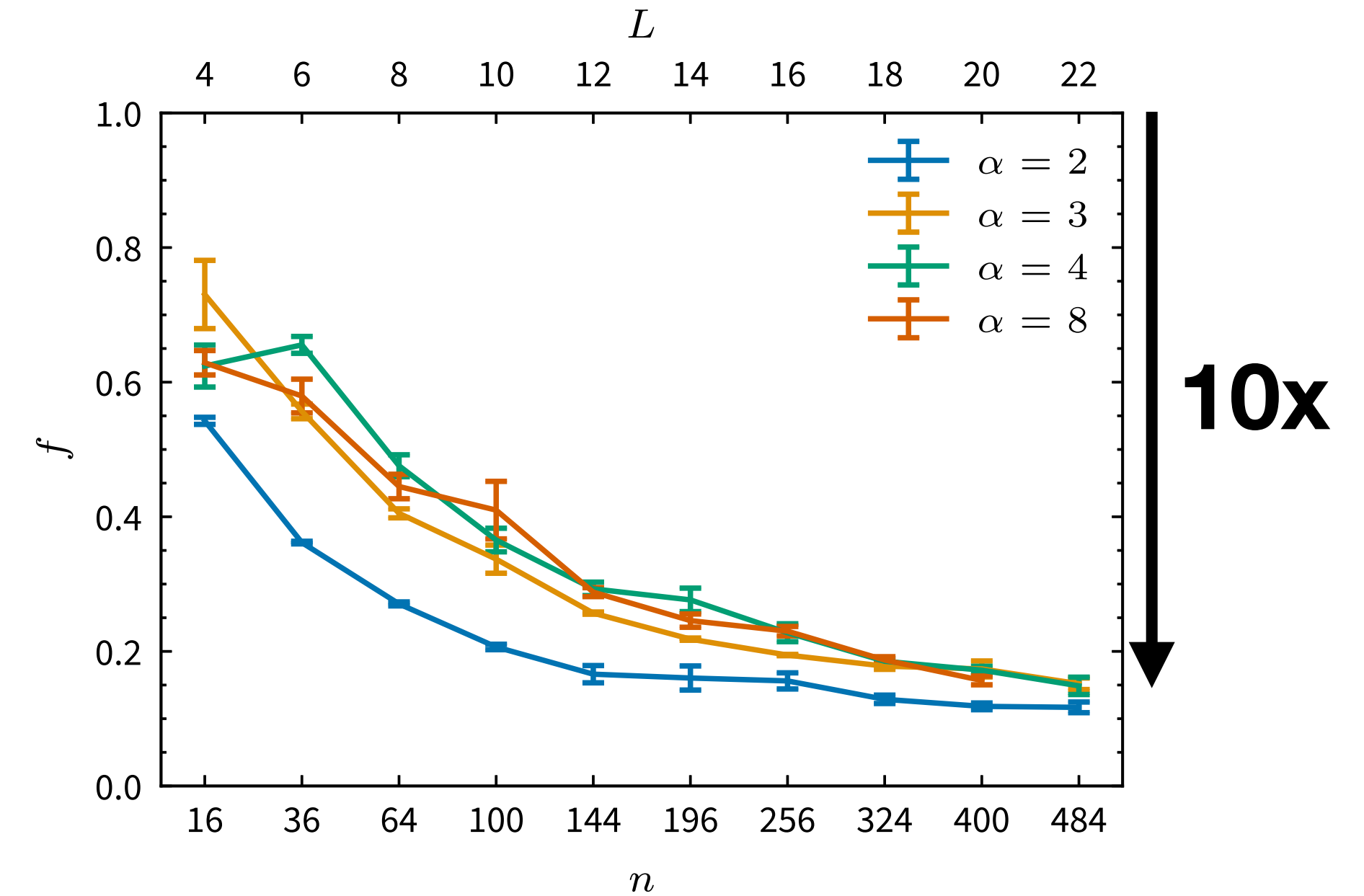
Antiferromagnetic Heisenberg model:

$$\hat{H} = J_{\text{ex}} \sum_{\langle ij \rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j$$



Constraint:

$$\sum_i^n s_i = 0$$



R. Berns et al. *Phys. Rev. Applied* 25, 024085 (2026)

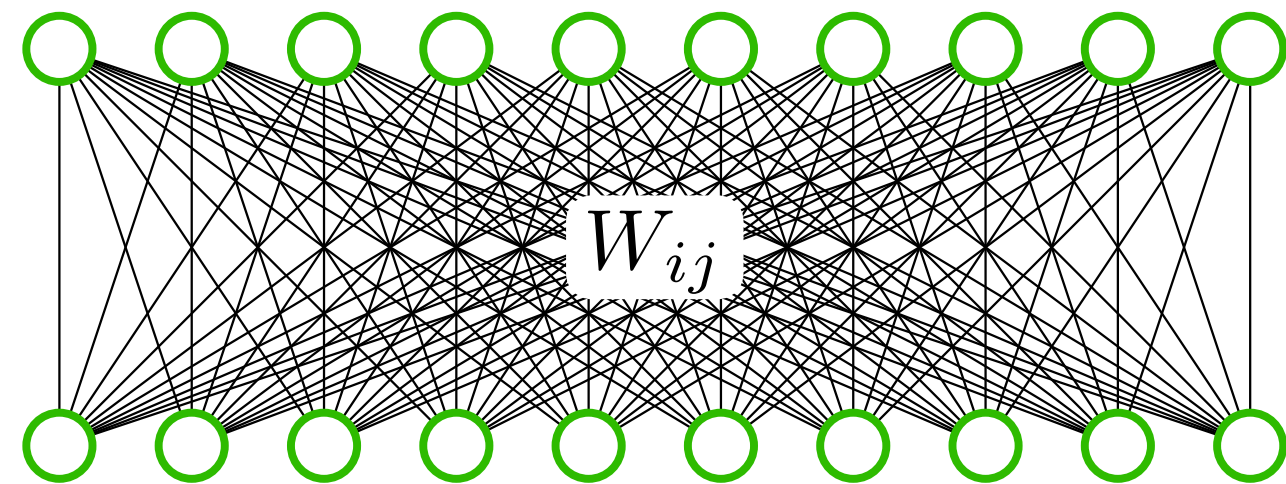
10x more samples without constraint

PArallel sampling with COnstraints (PACO)

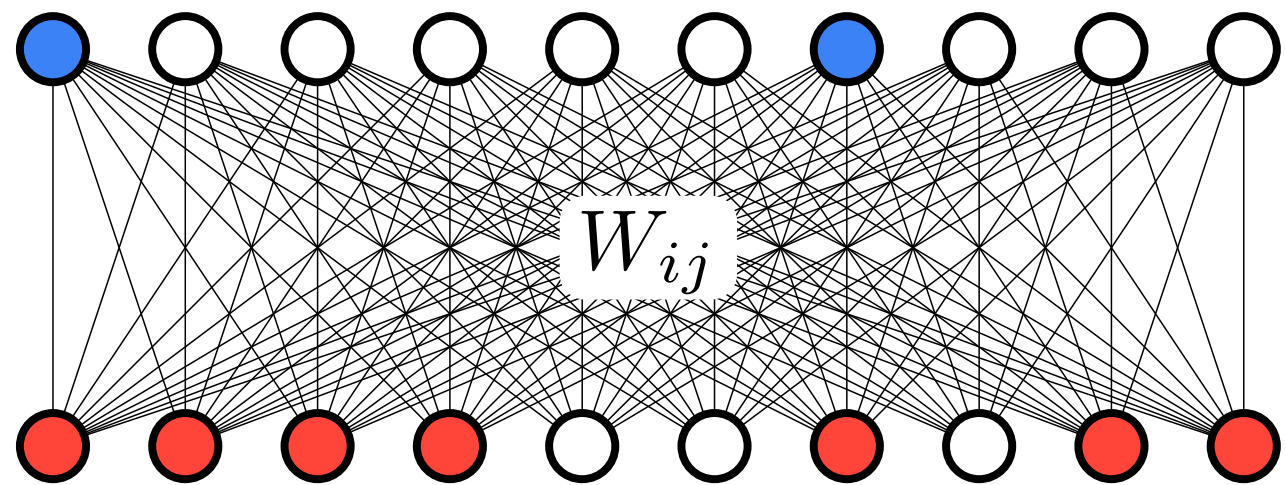
Stratmann et al. (2025) Patent US20250165765A1

(1) Evaluate Boltzmann condition for all spins

$$\exp(-\Delta E(m_i \rightarrow -m_i)) > r, \quad r \in U[0, 1]$$

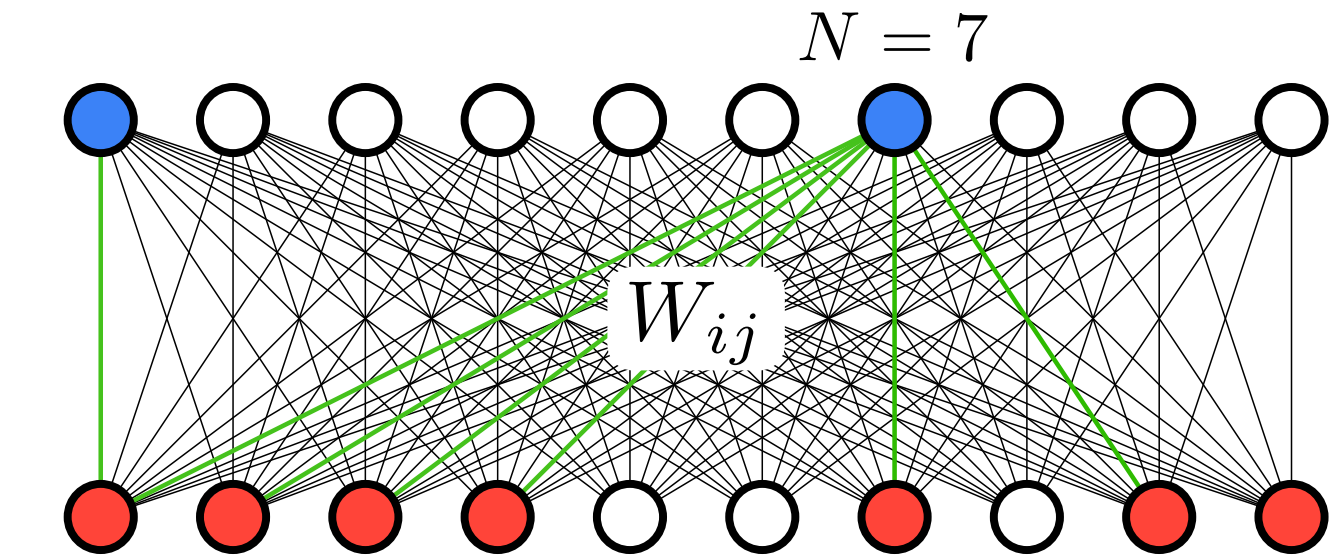


(2) Spins with intention to flip



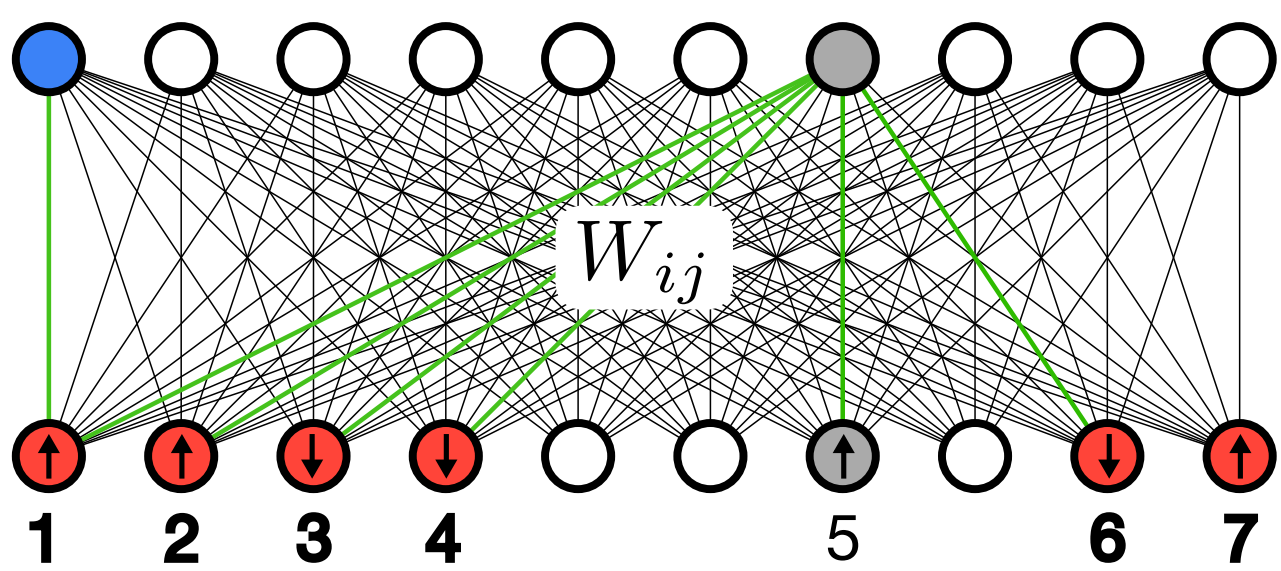
Breaks detailed balance!

(3) Weigh by accepted neighbors: $\frac{C}{N+1} > r', \quad r' \in U[0, 1]$



Restores $\approx$ detailed balance!

(4) Pairing for constraints

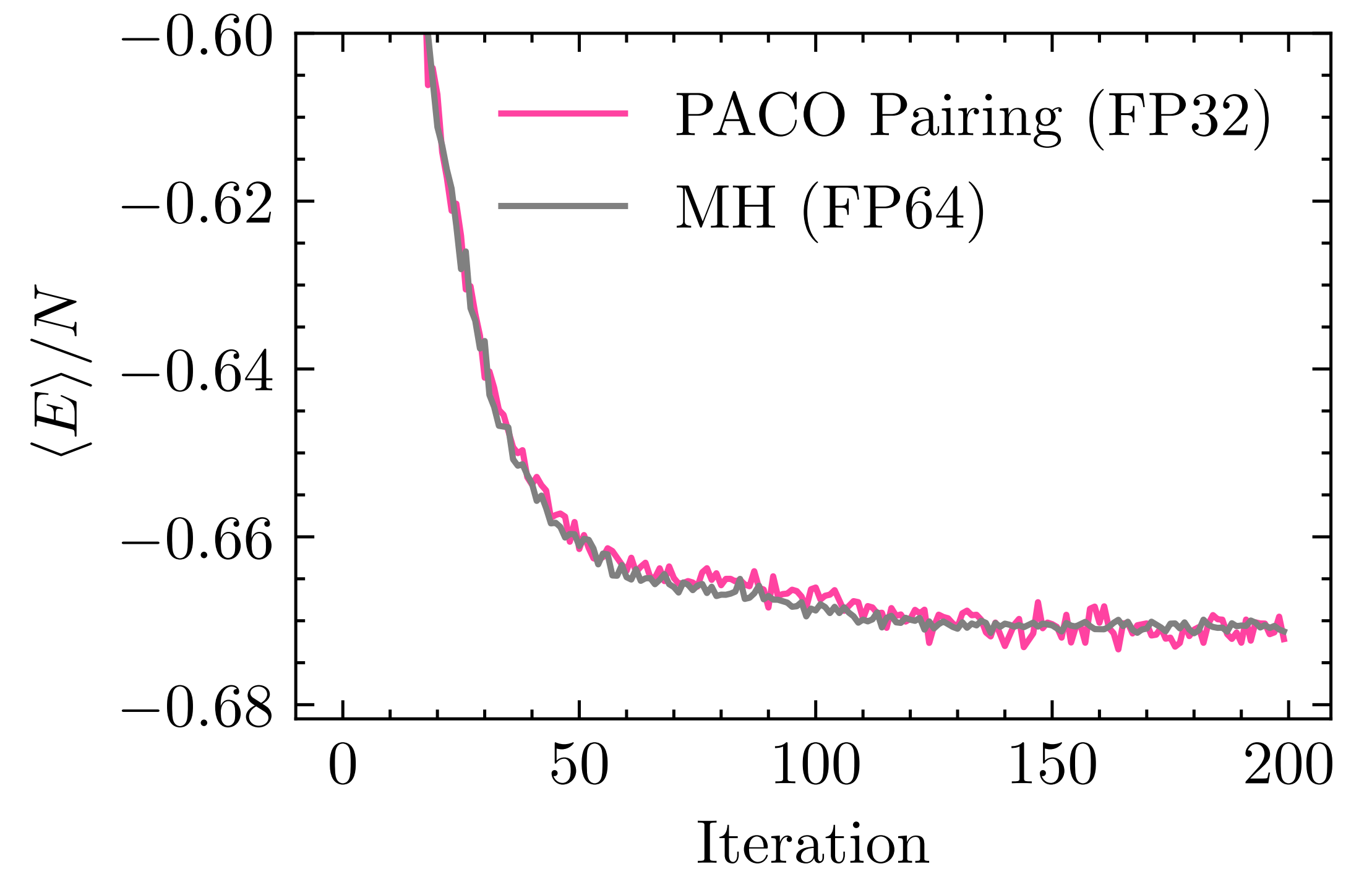
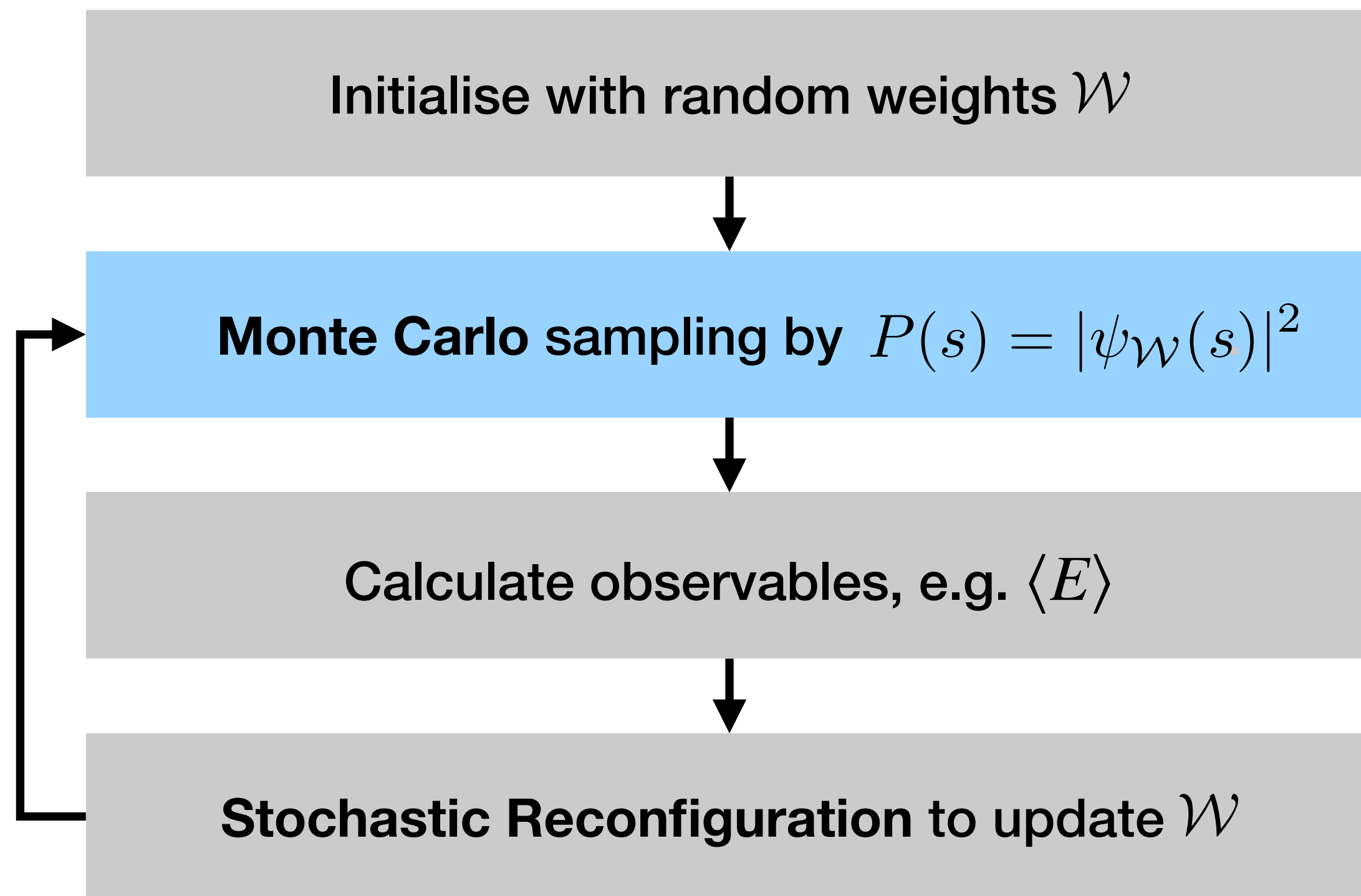


random pairs

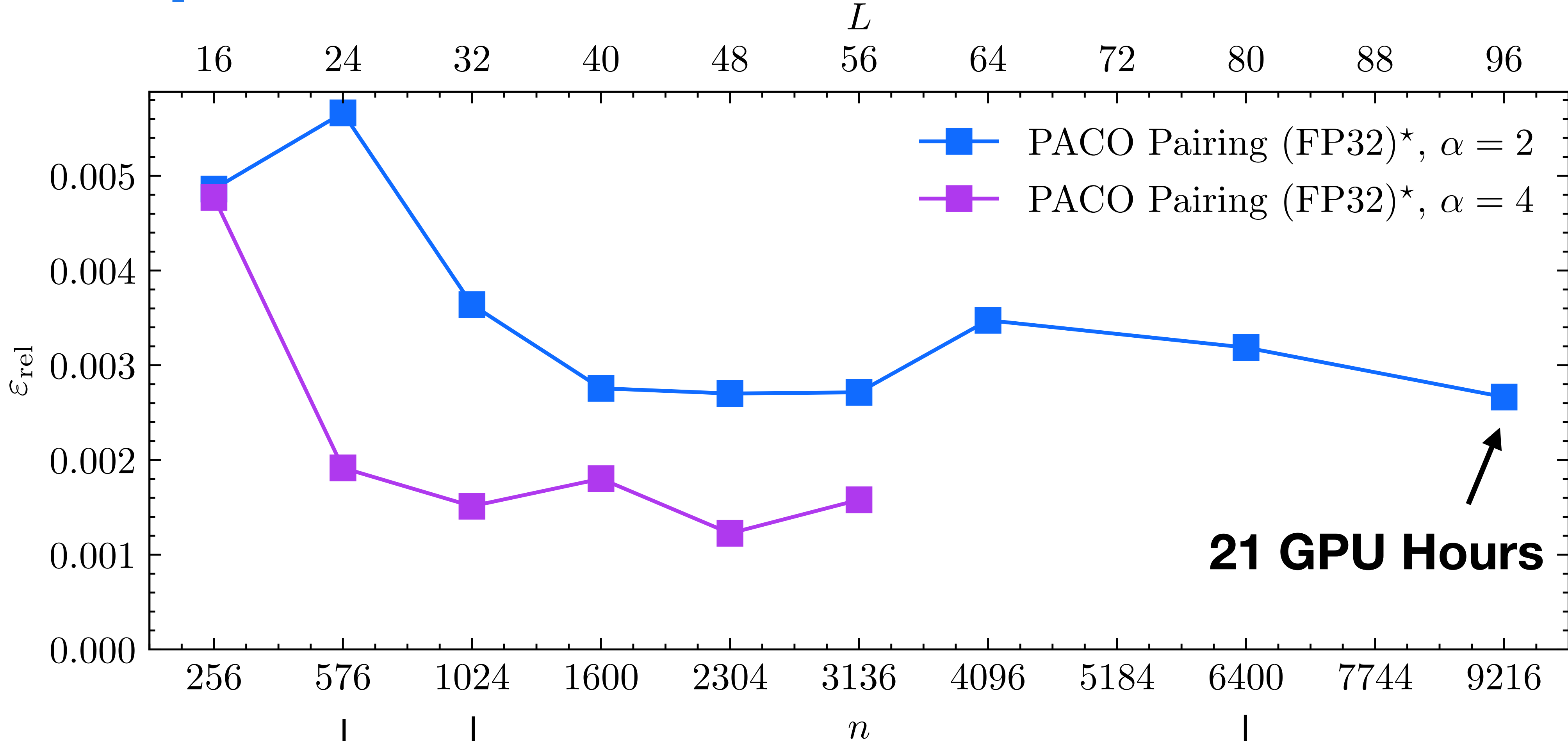
1	7	2	5
3	4	6	

- 1. Group visible spins that want to flip in two blocks (up/down)
- 2. Determine $\#N_b$: number of spins of smallest block
- 3. Flip $\#N_b$ spins in both up and down block

Ground state optimisation



Scale up on GPU

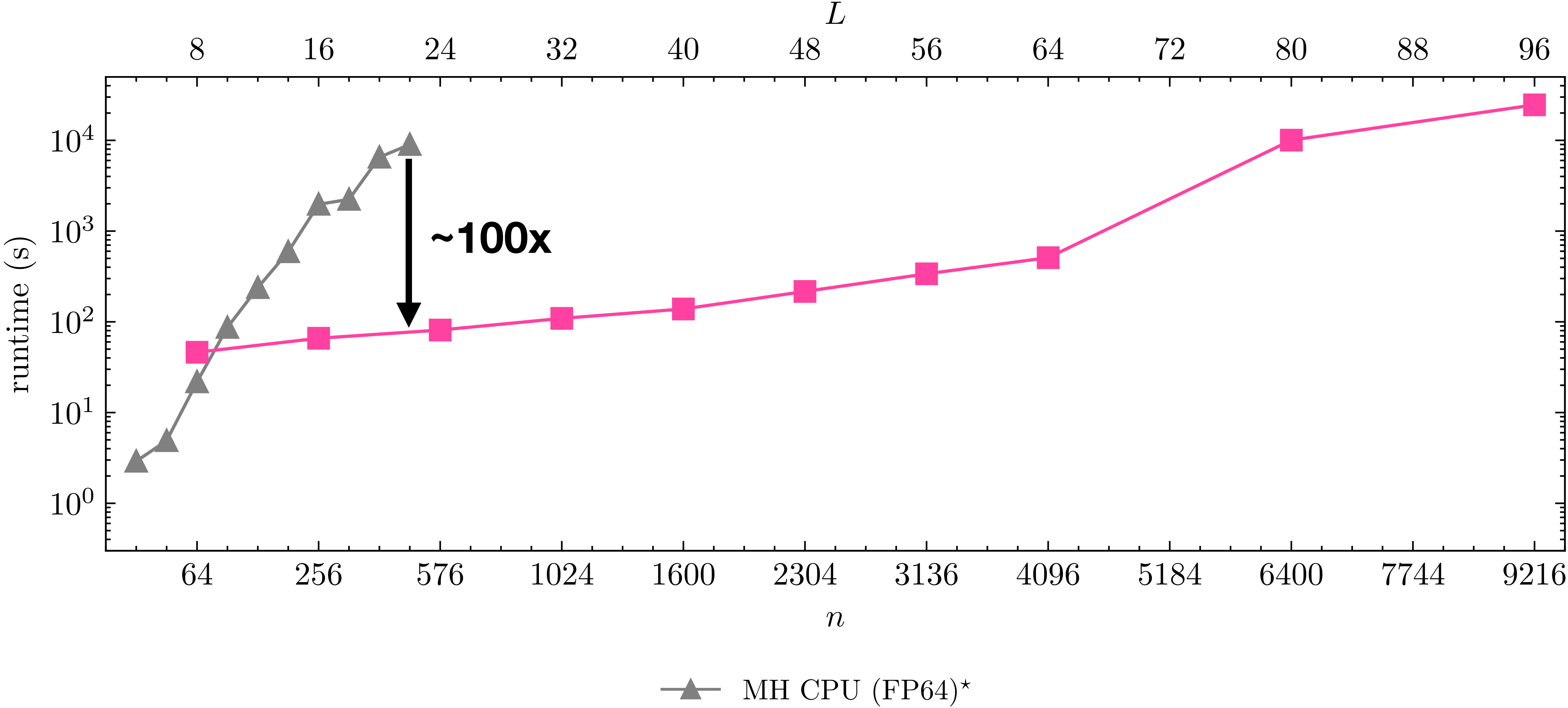


24×24 , *Fabiani et al. (2021) Phys. Rev. Lett. 127, 097202*

32×32 , *Moss et al. (2025) Phys. Rev. B 112, 134450*

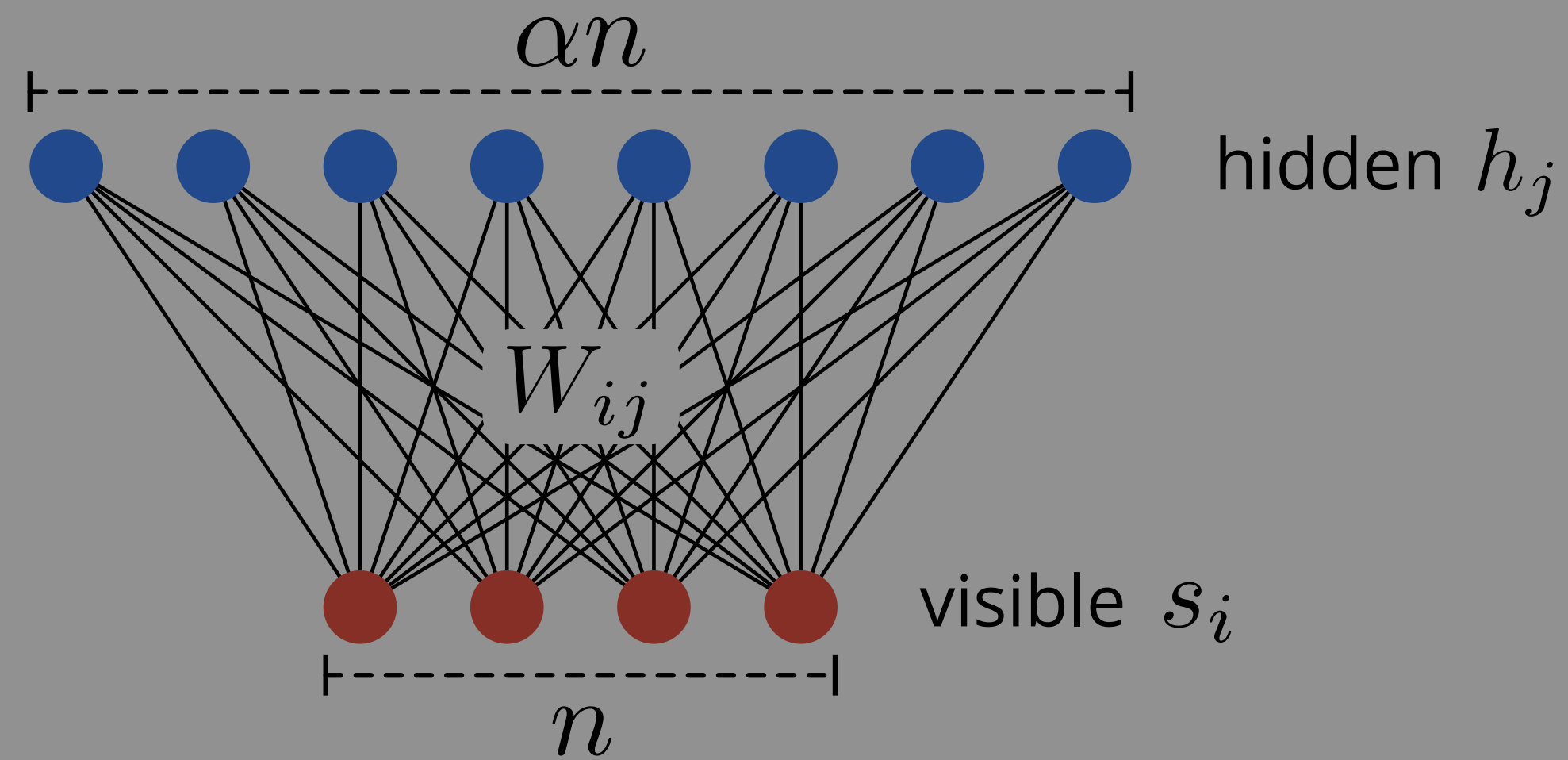
80×80 , *Chowdhury et al. (2025) arXiv:2512.24558*

Total sampling time



Outline

1. Restricted Boltzmann Machines



2. GPU Acceleration

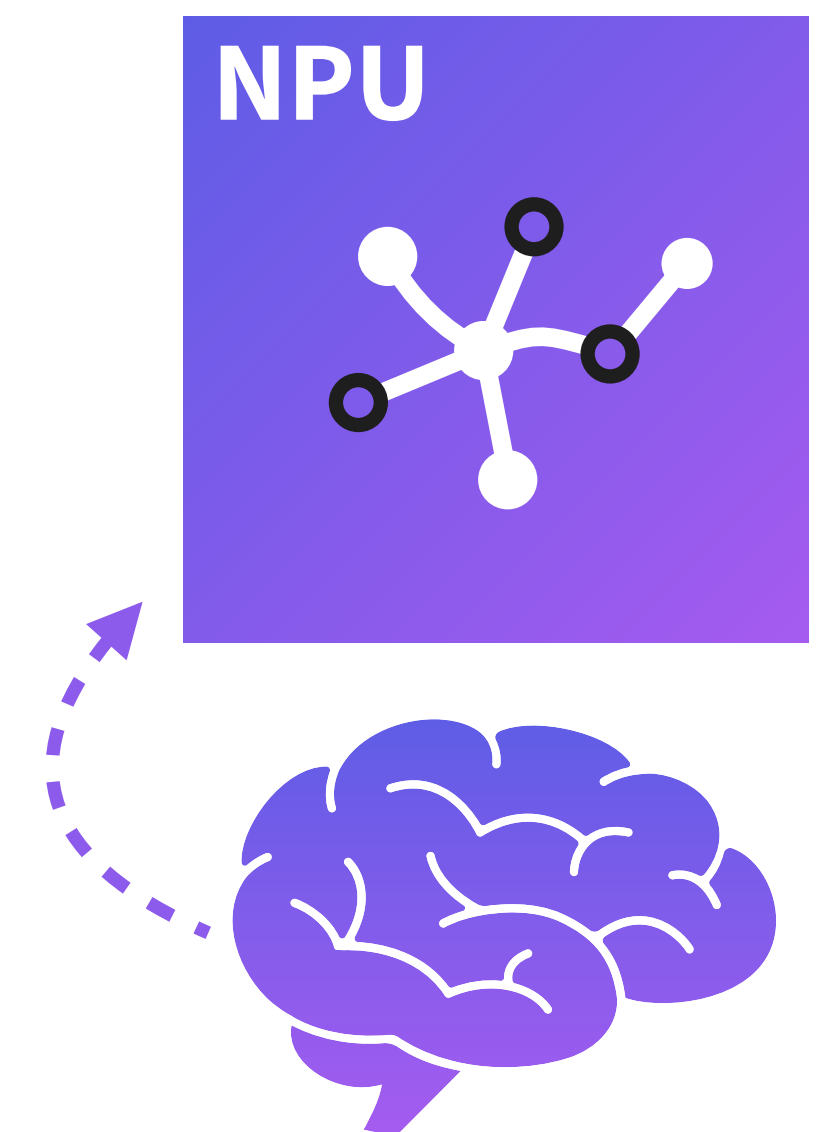
Scale up to 10.000 quantum spins

3. Neuromorphic computing

Scale to millions quantum spins

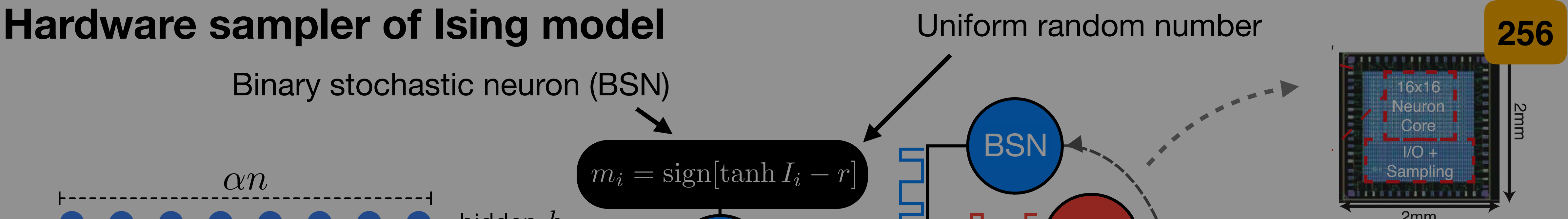
Neuromorphic computing

- Conventional architectures have an inherent memory throughput bottleneck!
- **Neuromorphic hardware:**
 - Brain inspired
 - Memory and processing integrated
 - Massively parallel & scalable compute!
- But limited programmability and often requires **quantization & sparse connectivity**

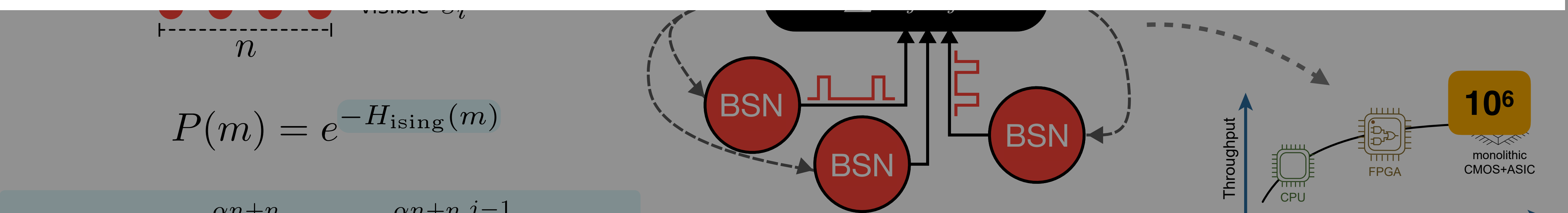


Ising Machines

Hardware sampler of Ising model



Sampling advantage for NQS?



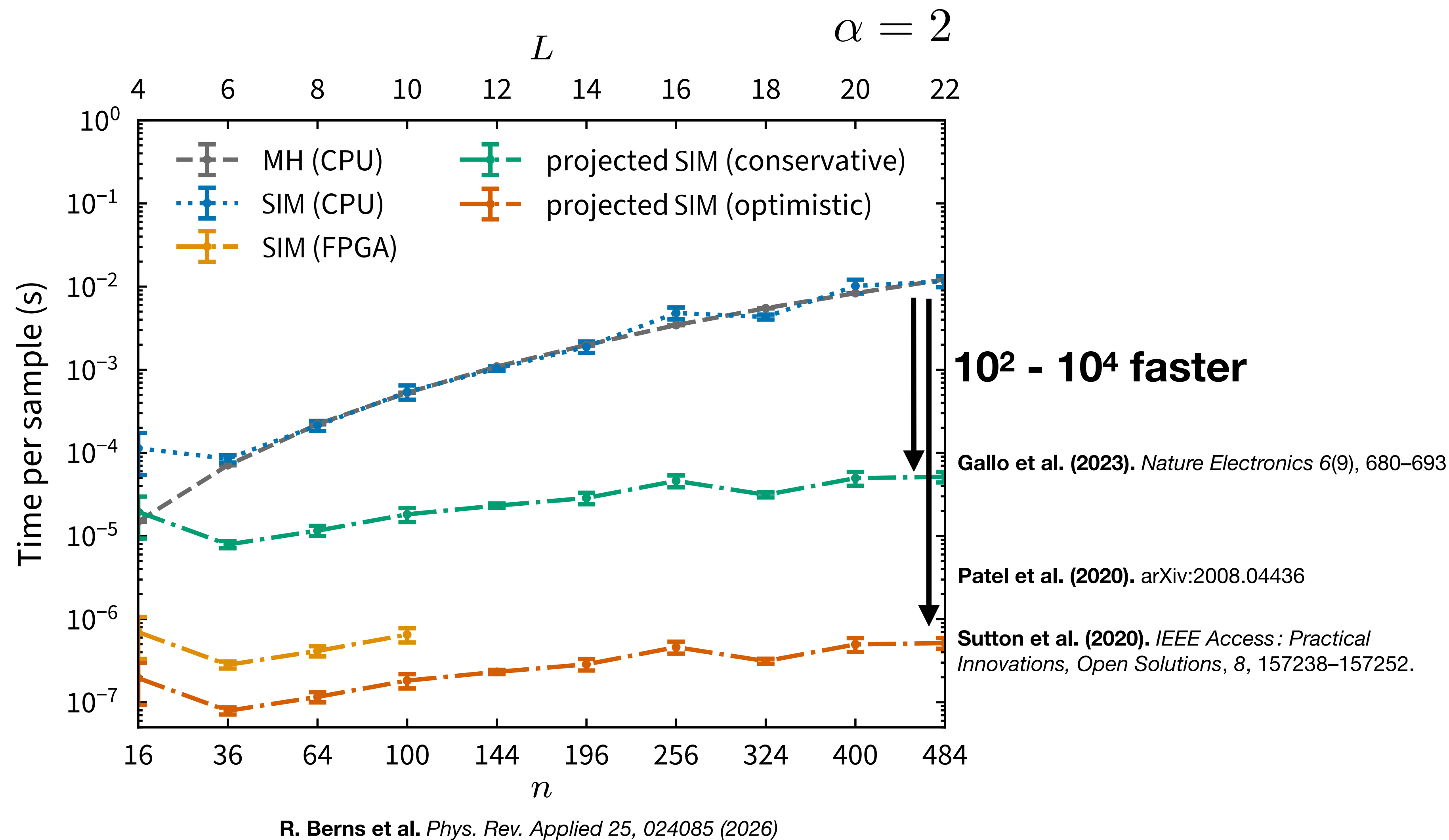
$$H_{\text{ising}}(m) = - \sum_{i=1}^{\alpha n + n} h_i m_i - \sum_{j=1}^{\alpha n + n} \sum_{i=1}^{j-1} J_{ij} m_i m_j$$

Chromatic Gibbs sampling

Chowdhury et al. (2025) arXiv:2512.24558

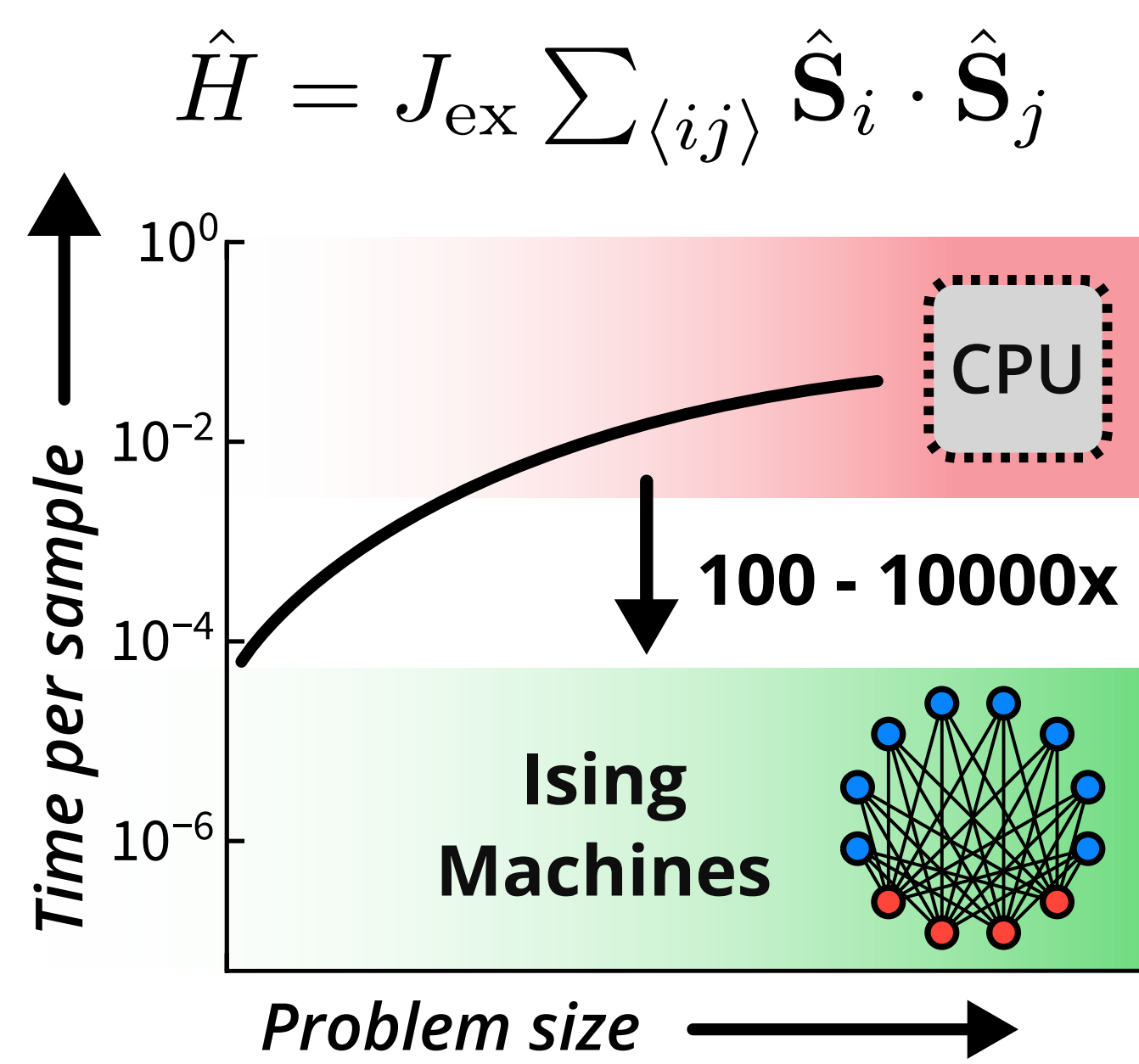
Aadit et al. (2026) ArXiv 2606.25313

Ising Machines



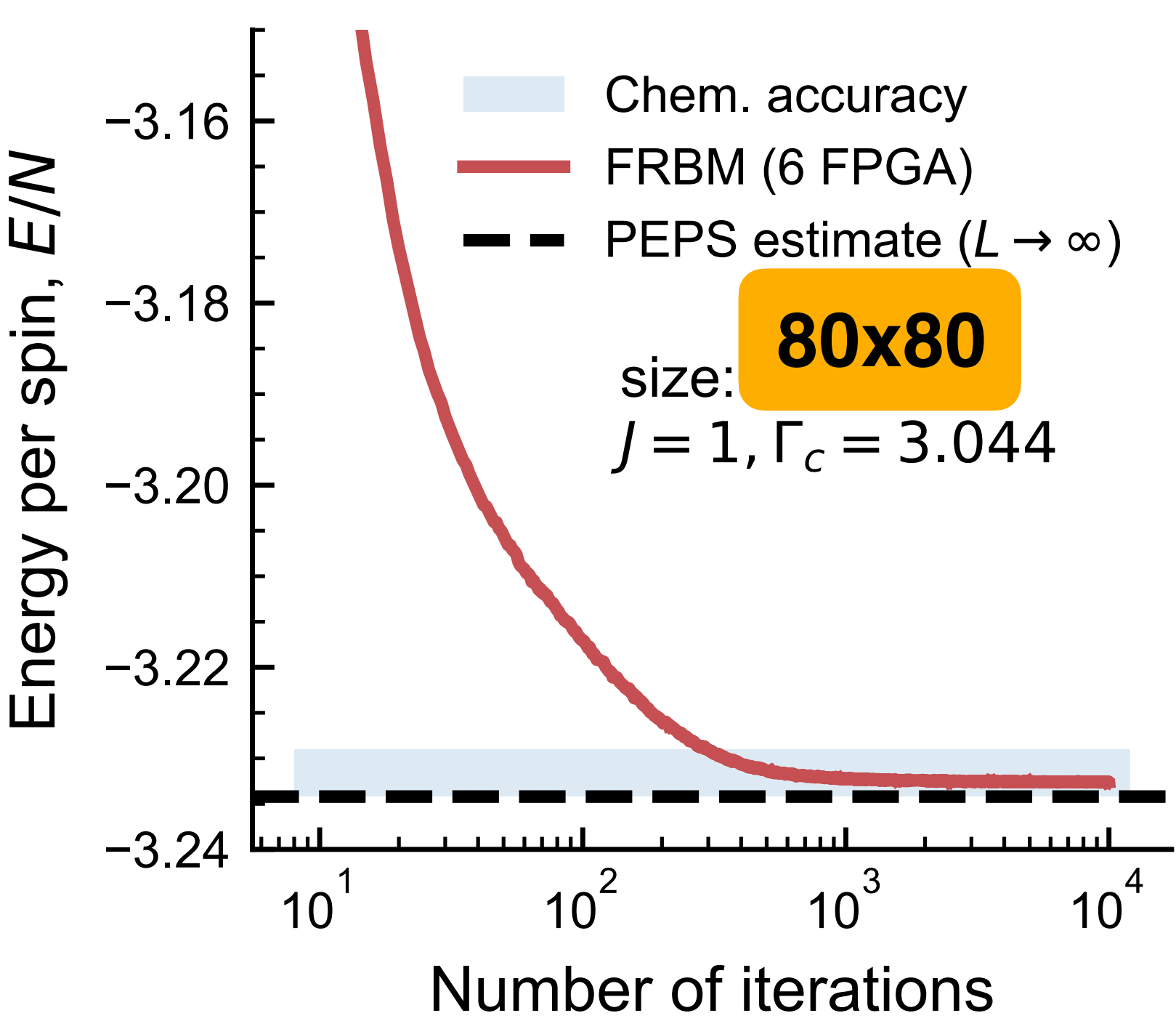
Probabilistic Ising Machines for NQS are exciting!

Predicted sampling advantage



R. Berns et al. *Phys. Rev. Applied* 25, 024085 (2026)

Demonstration Large scale NQS

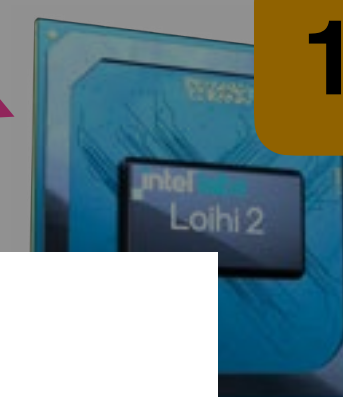
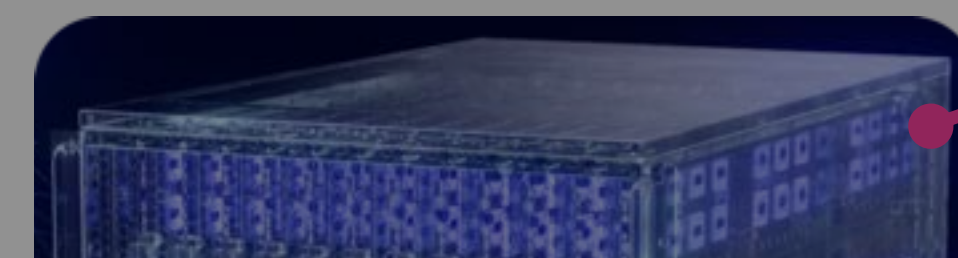


S. Chowdhury et al. *arXiv:2512.24558* (2025)

Large scale distributed spiking hardware

Intel Loihi 2

- Recent dedicated sampling schemes for



10^9

Sampling with constraints on distributed spiking hardware

- However: sparse communication & quantization



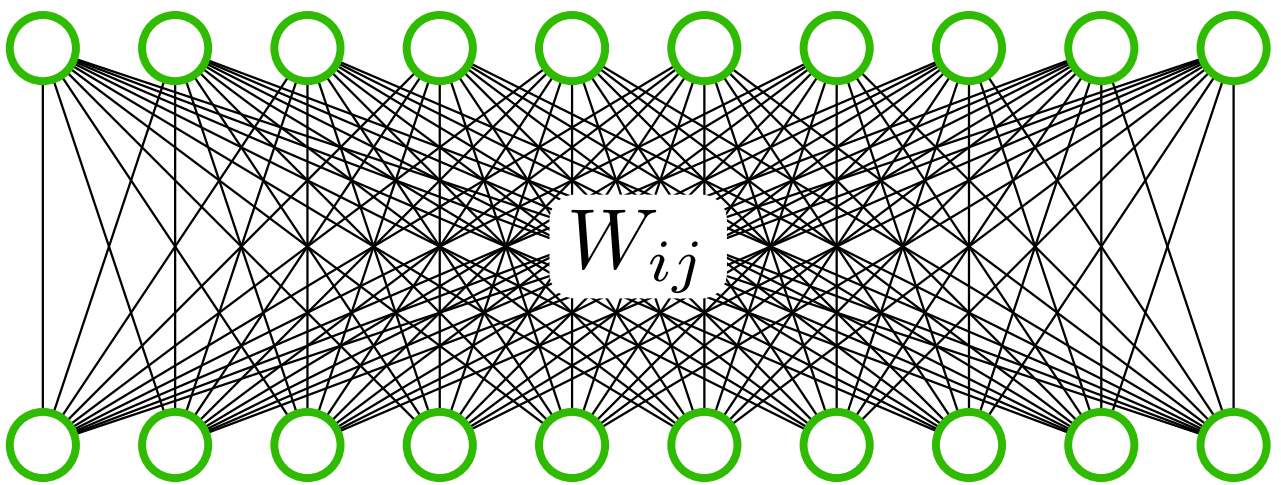
5×10^9

SpiNNaker: Chen et al. (2025) *Nat Commun* 16, 3086.

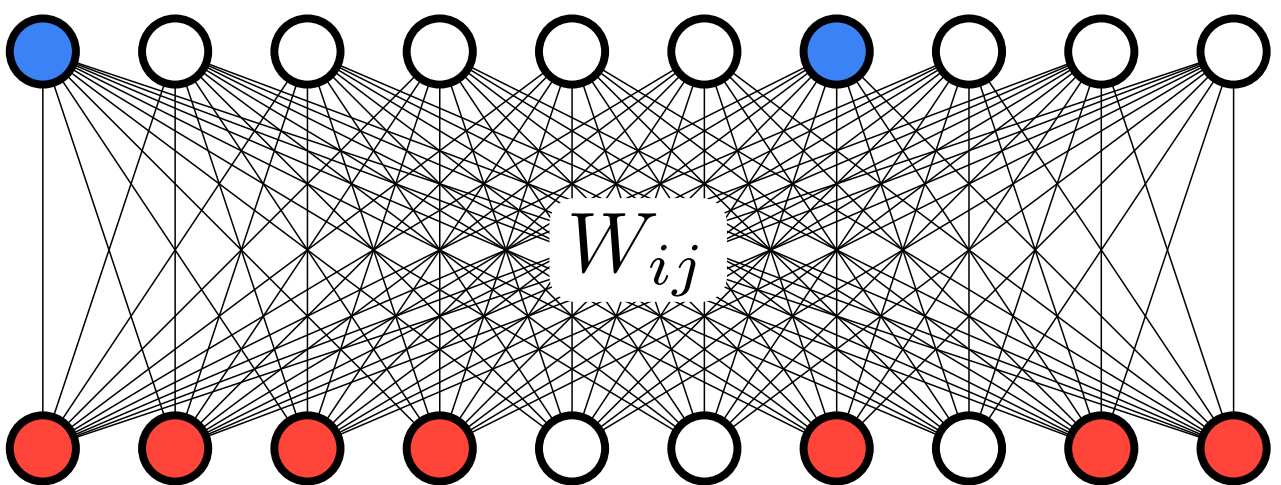
PArallel sampling with COnstraints (PACO)

(1) Evaluate Boltzmann condition for all spins

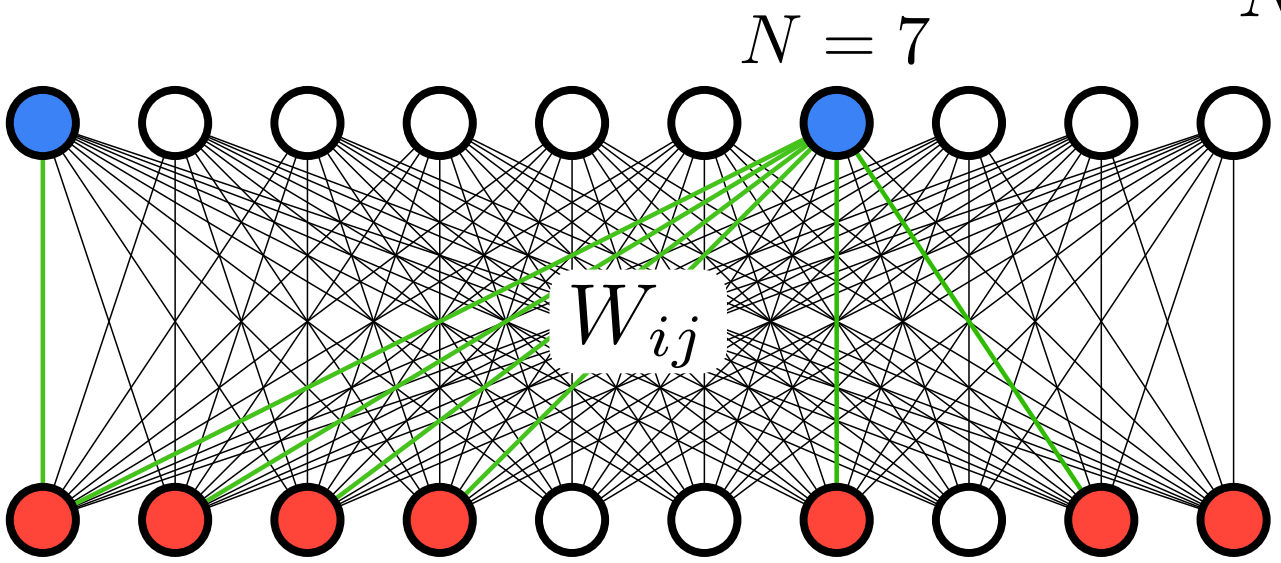
$$\exp(-\Delta E(m_i \rightarrow -m_i)) > r, \quad r \in U[0, 1]$$



(2) Spins with intention to flip



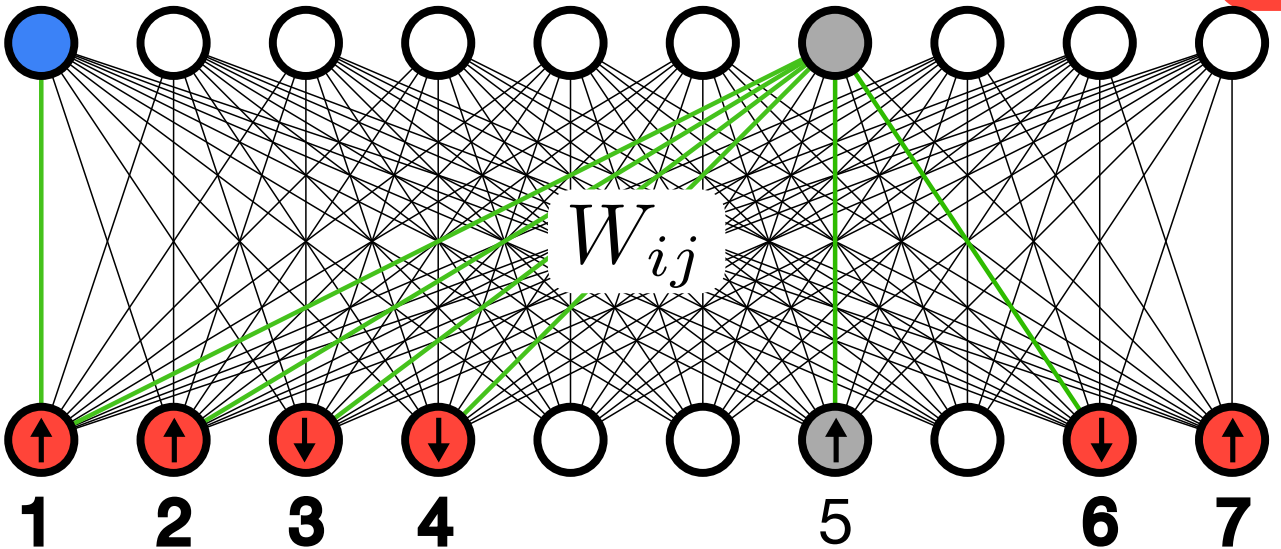
(3) Weigh by accepted neighbors: $\frac{C}{N+1} > r', \quad r' \in U[0, 1]$



$N = 2$

(4A) Pairing for constraints

Global synchronization

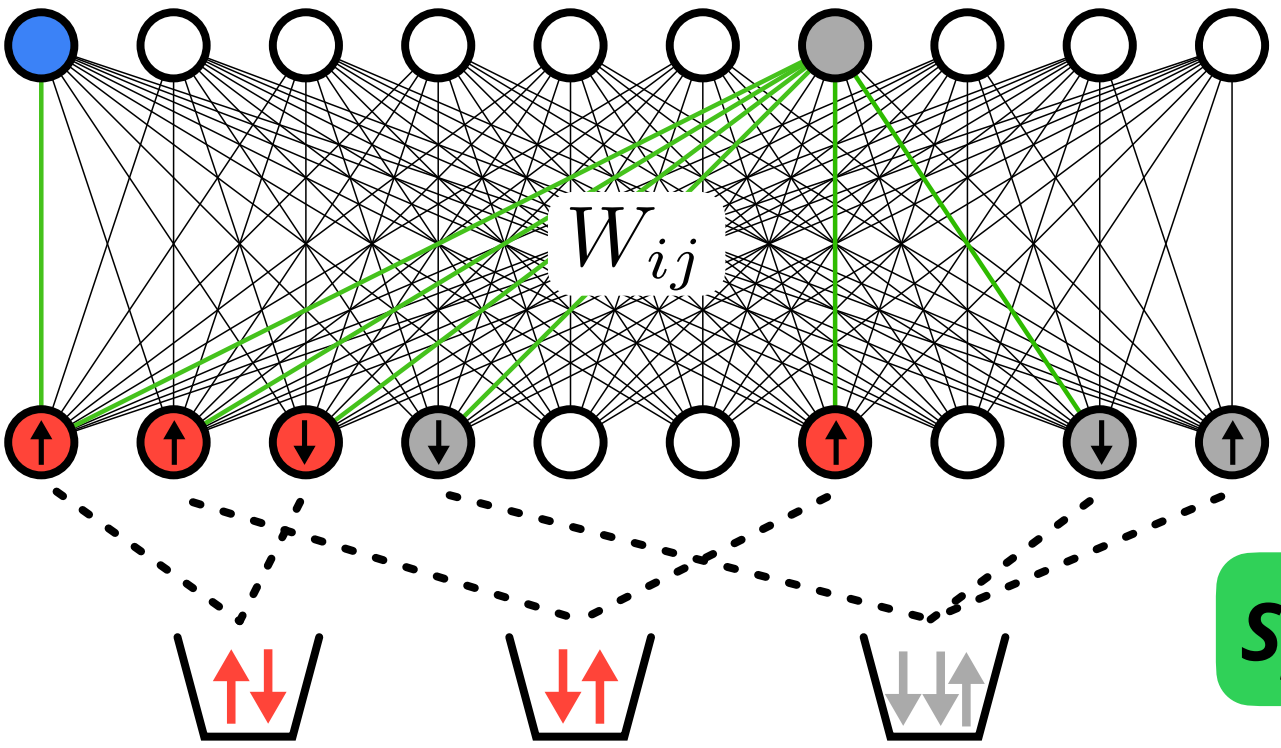


random pairs

1	7	2	5
3	4	6	

1. Group visible spins that want to flip in two blocks (up/down)
2. Determine $\#N_b$: number of spins of smallest block
3. Flip $\#N_b$ spins in both up and down block

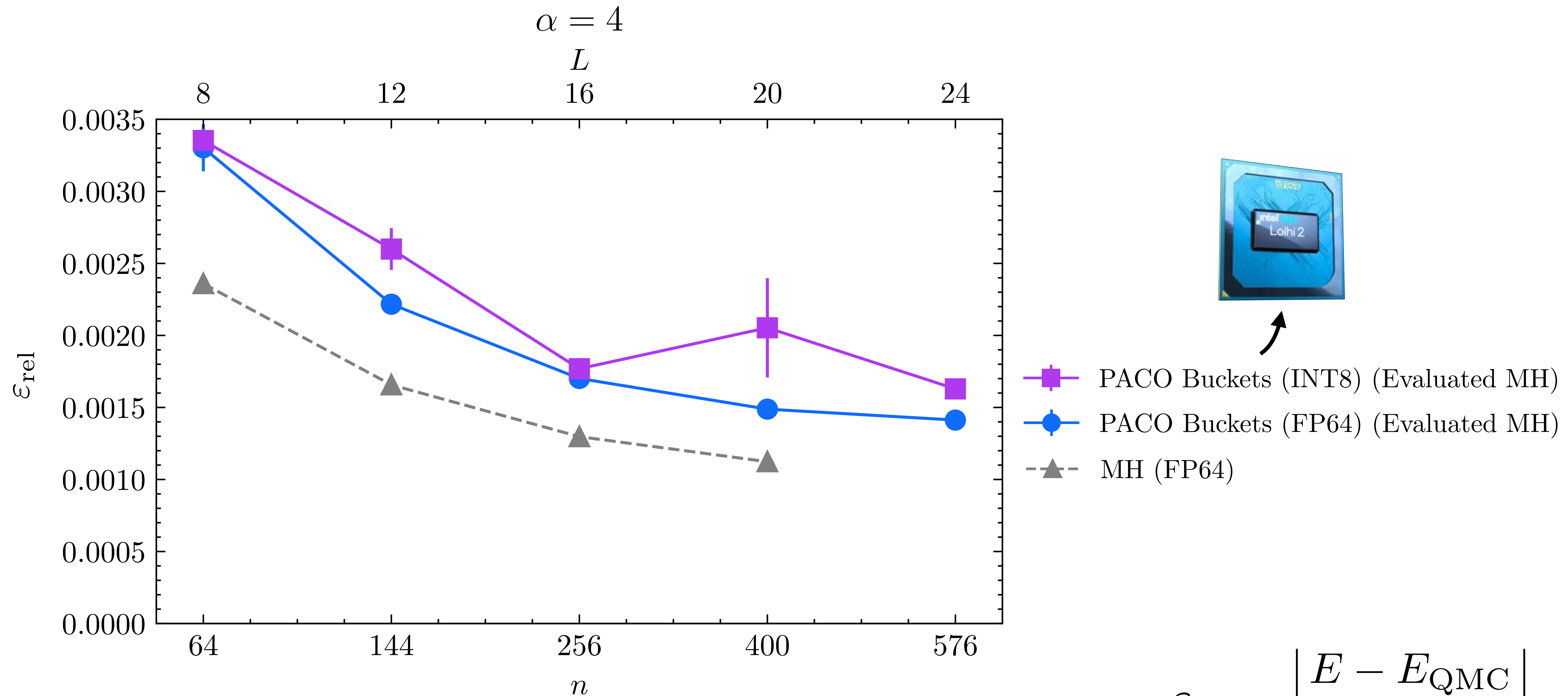
(4B) Buckets for constraints



Sparse communication

1. Create D buckets
2. Each visible spin sends new spin value to **random bucket**
3. If total spin in a bucket vanishes, allow flip of those spins!

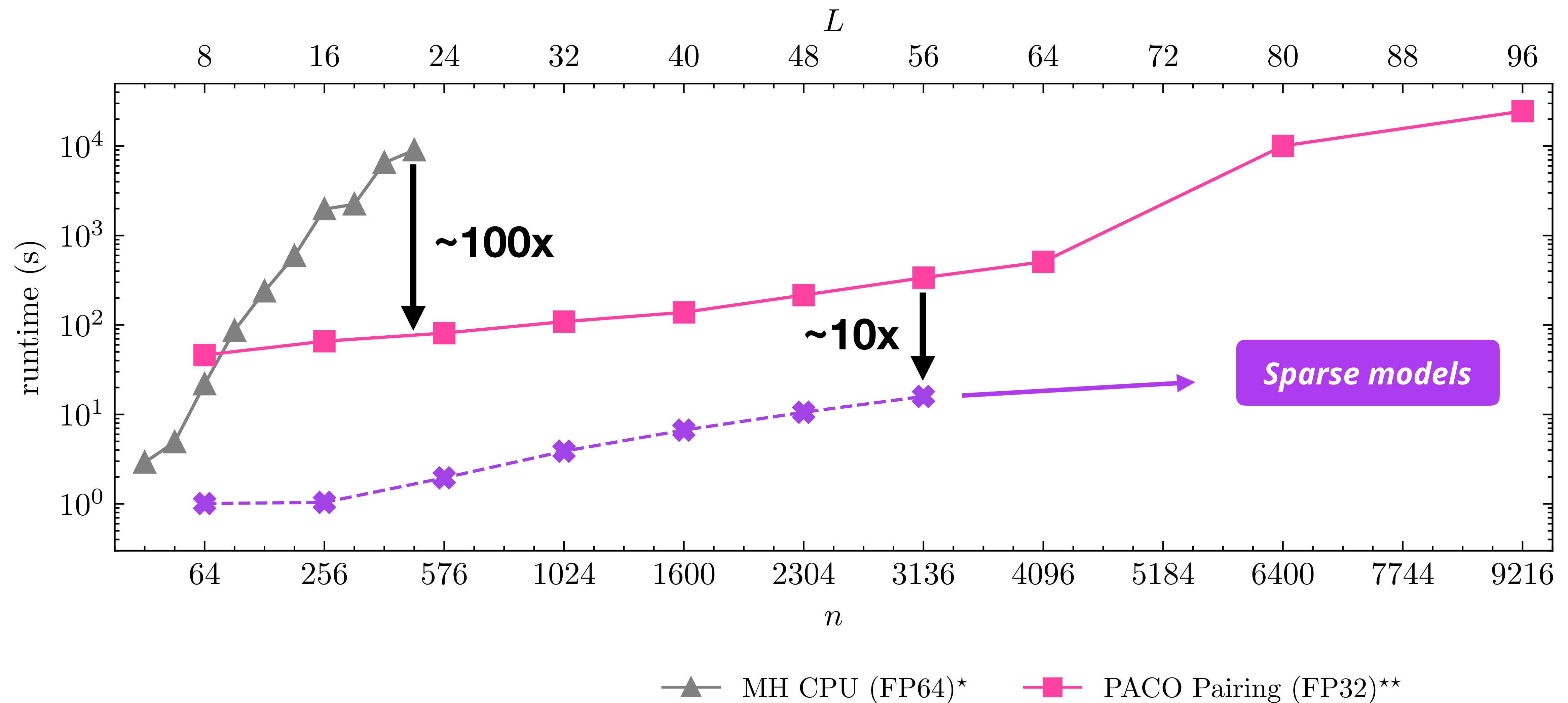
Accuracy for PACO with Buckets



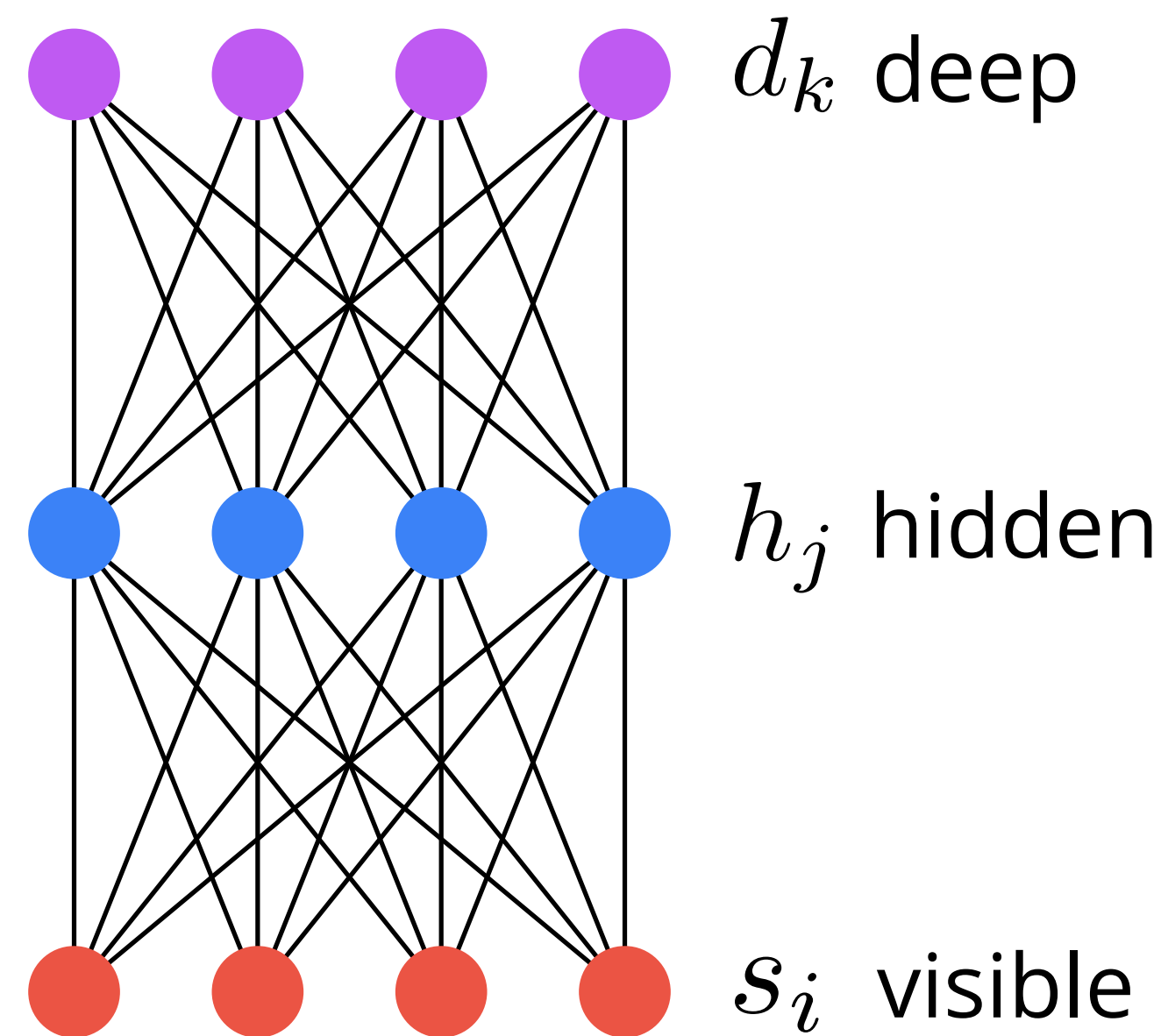
Model & optimization in FP64, Local energies FP32 GPU

$$\varepsilon_{\text{rel}} = \left| \frac{E - E_{\text{QMC}}}{E_{\text{QMC}}} \right|$$

Total sampling time



Outlook: (Sparse) Deep Boltzmann Machines

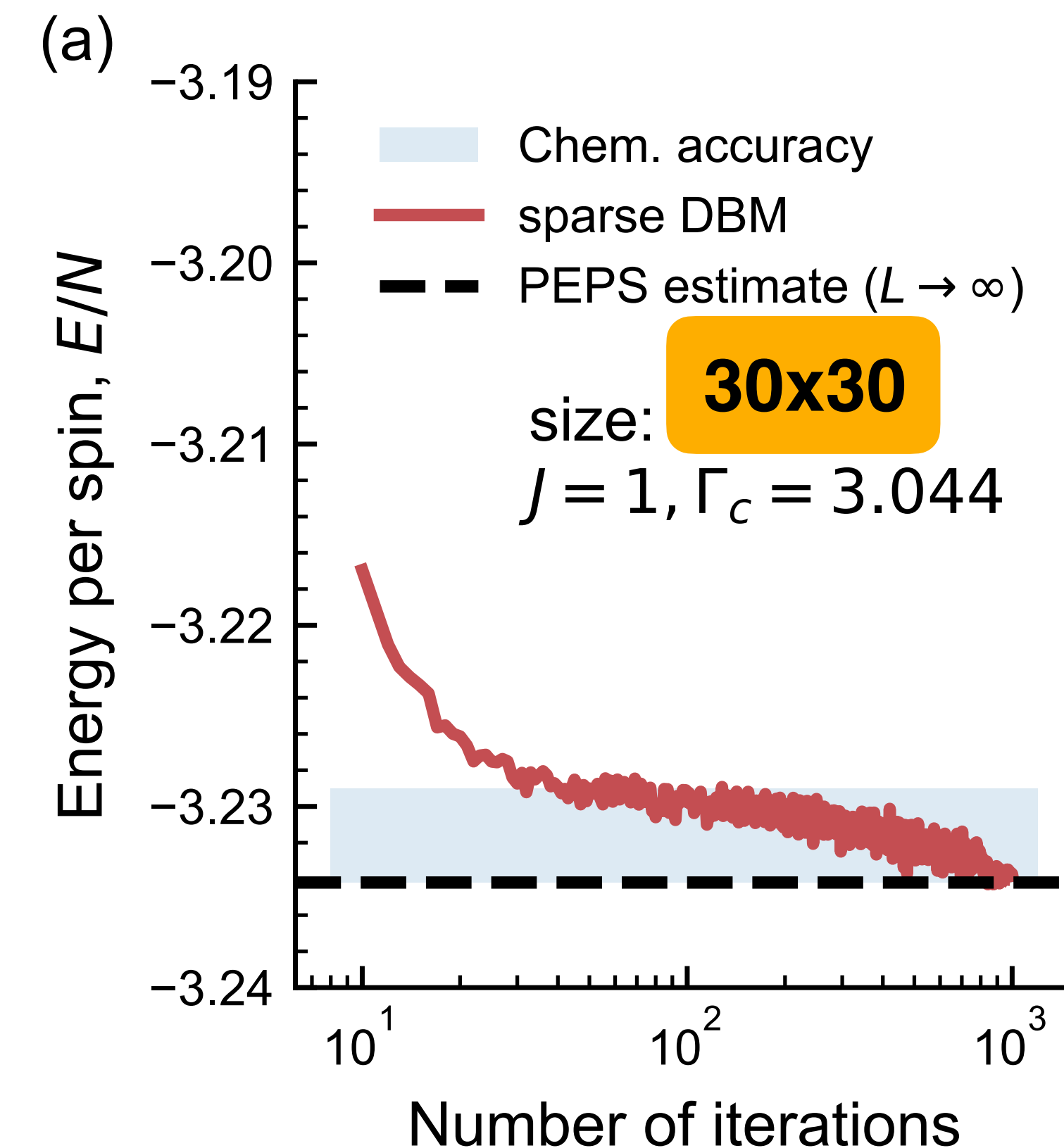


Deep Boltzmann Machine

Carleo et al. (2018). *Nature Communications*, 9(1), 5322.

Niazi et al. (2024). *Nature Electronics*, 7(7), 610–619.

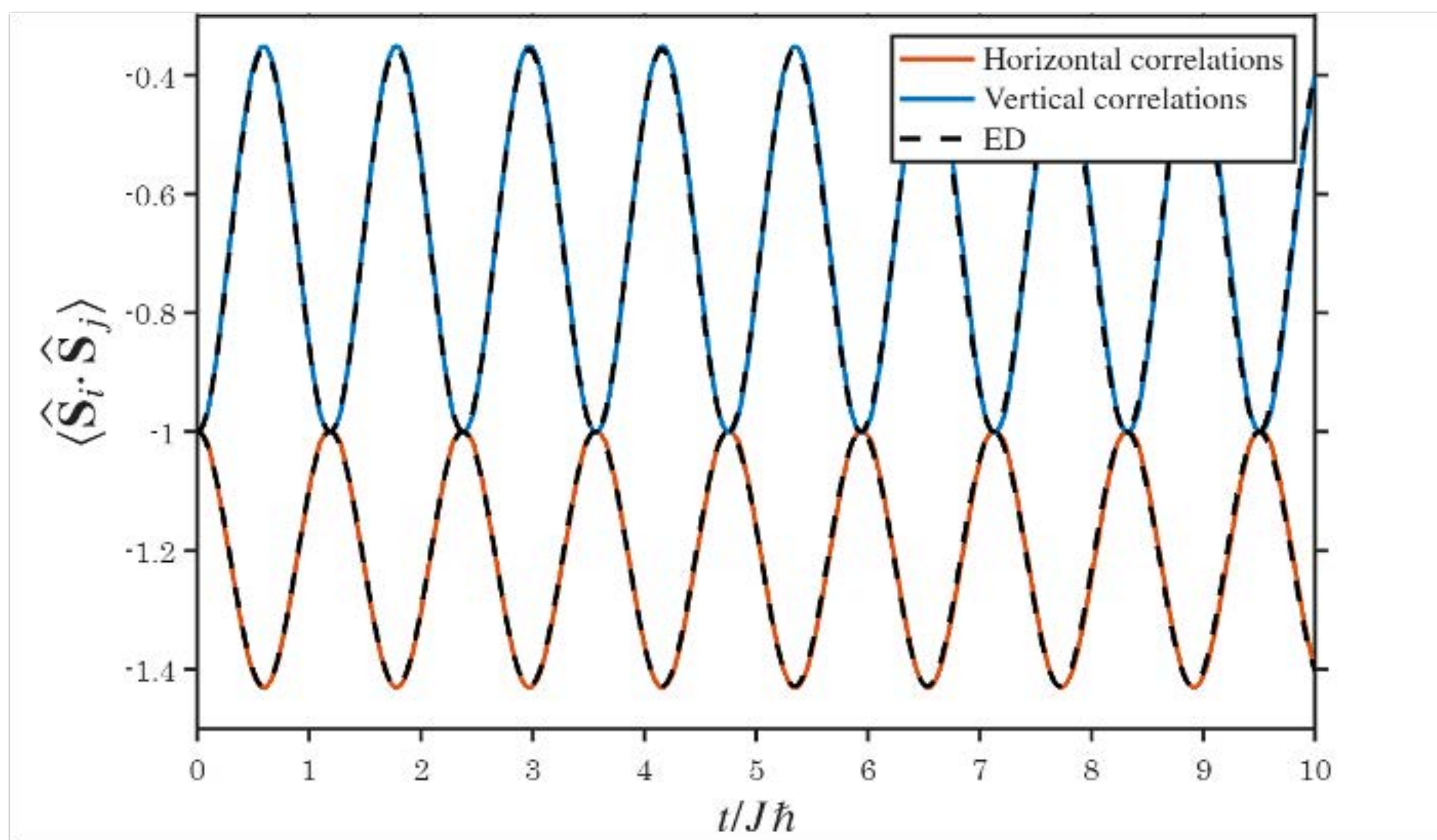
Nomura et al. (2021). *Physical Review Letters*, 127(6), 060601.



S. Chowdhury et al. *arXiv:2512.24558* (2025)

Outlook: Boltzmann Machines for NQS Dynamics

2x2 Heisenberg model



RBM ($\alpha = 2$) x RBM ($\alpha = 2$)

Finnian Rist (2026)

- **Amplitude-phase ansatz:**

$$\psi_{\theta}(S) = \psi_{\mu}(S)e^{i\phi_{\nu}(S)}$$

- **Train dual networks:**

$$\begin{pmatrix} S_{\mu_i \mu_j} & 0 \\ 0 & S_{\nu_i \nu_j} \end{pmatrix} \begin{pmatrix} \dot{\mu}_j \\ \dot{\nu}_j \end{pmatrix} = \begin{pmatrix} F_{\mu_i} \\ F_{\nu_i} \end{pmatrix}$$

Acknowledgement

Radboud Universiteit



- **Johan Mentink**
- **Philipp Küppers**
- Hrvoje Vrcan
- Pim Coenders
- **Finnian Rist**



- **Alessandro Pierro**



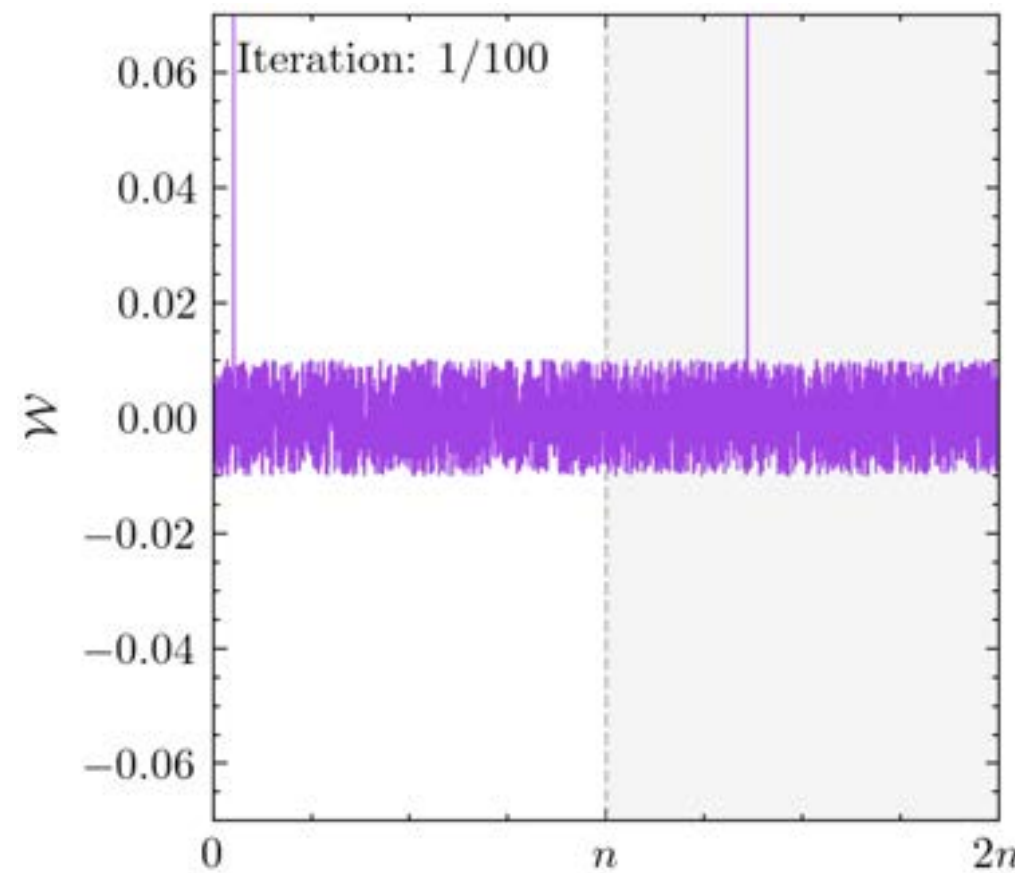
- **Davi Rodrigues**



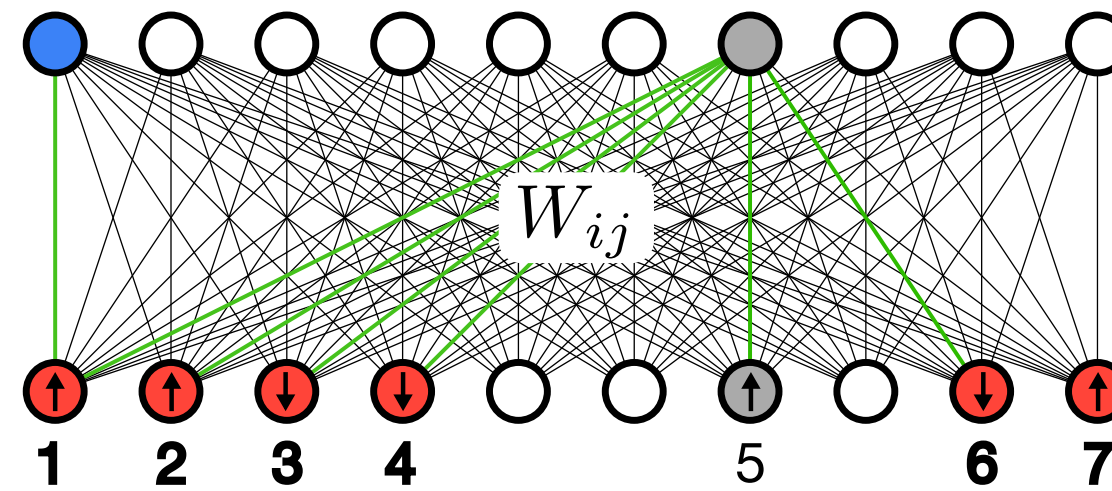
- **Giovanni Finocchio**



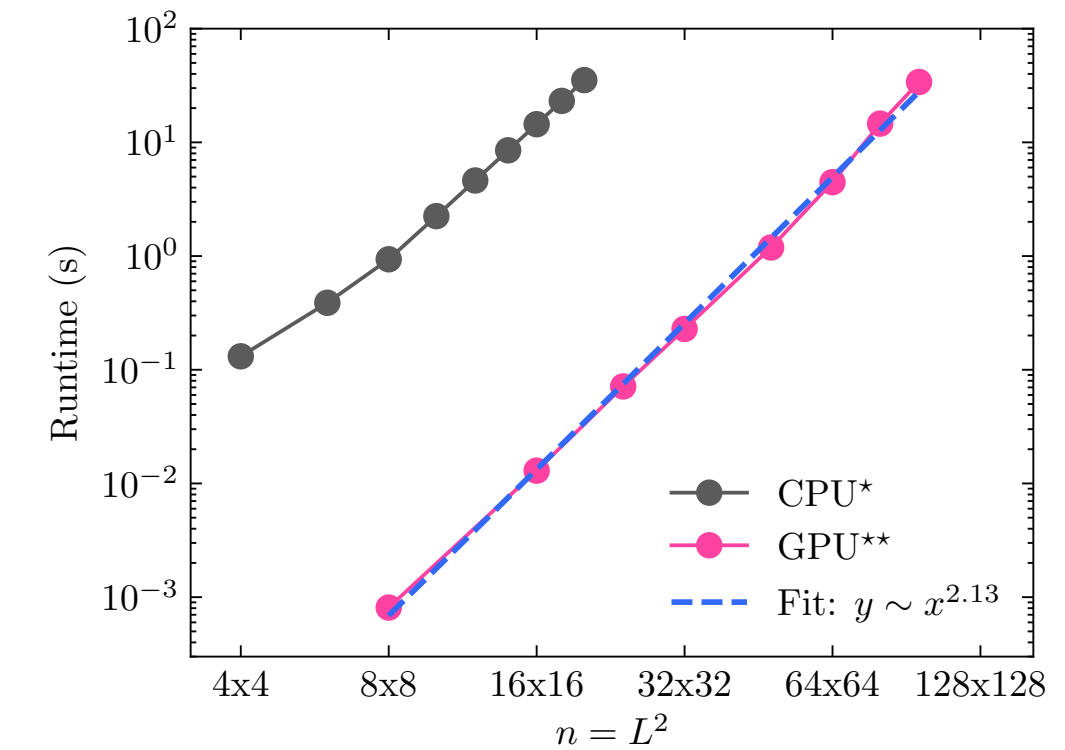
Conclusion



Spiky initialization



PACO



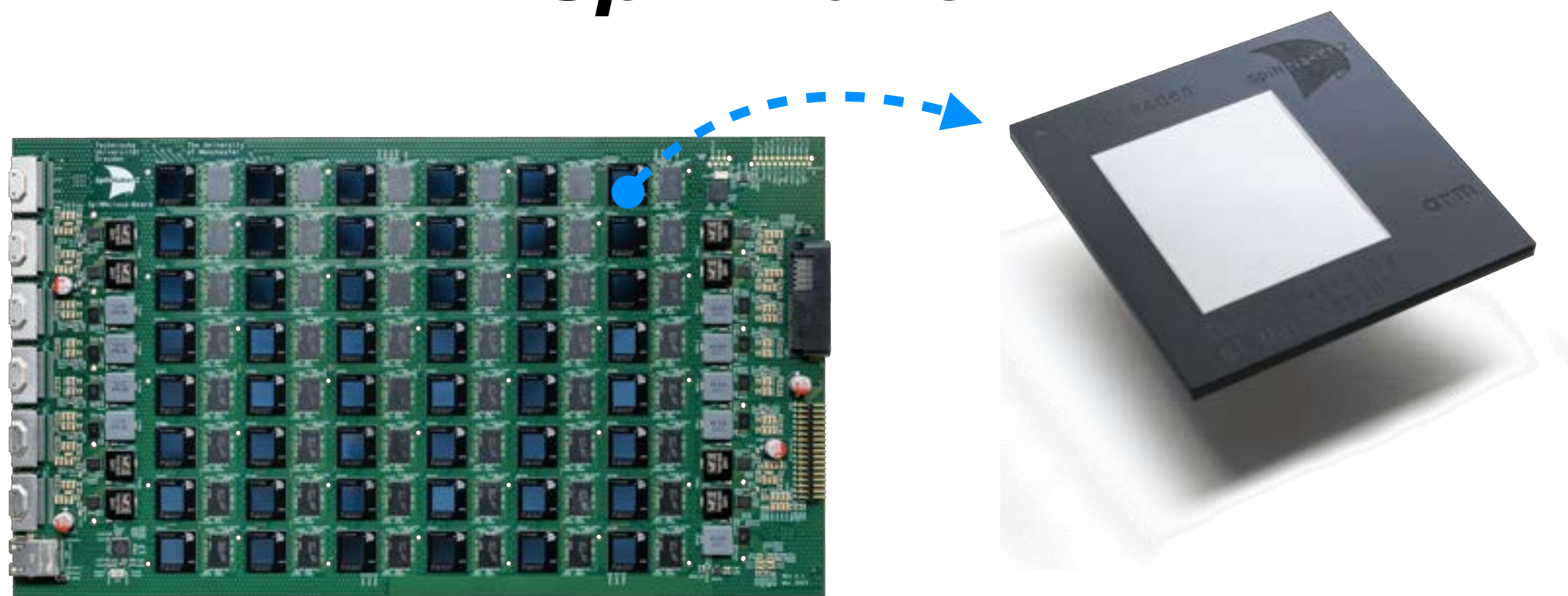
GPU acceleration

- **RBM** powerful due to symmetries, efficient updates and new initialization!
- Quantum simulation with **PACO** sampler on GPUs with almost **10.000 spins**
- INT8 quantization of sampling with competitive accuracy, giving high potential for deployment on **neuromorphic hardware**!
- Projected speed-up on spiking hardware such as Intel Loihi 2.

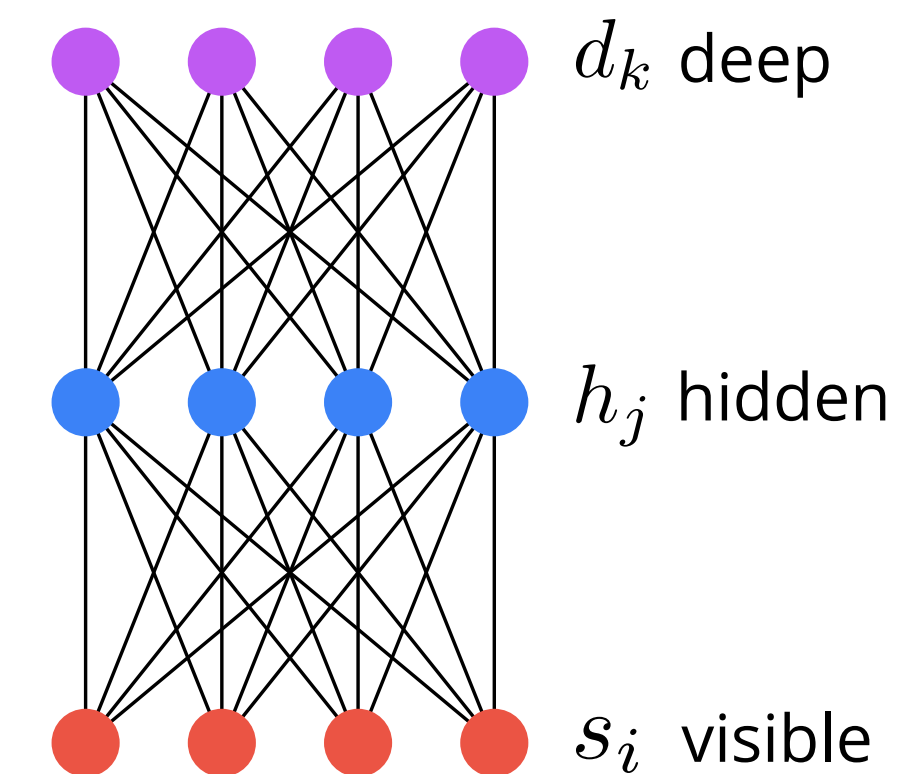
Outlook

- Deploy on SpiNNaker and Intel Loihi 2 to leverage **parallelism & sparsity** with models such as **sparse deep Boltzmann Machines**
- **Buckets** for constraints with chromatic Gibbs sampling
- **Quantum dynamics** of antiferromagnetic textures

SpiNNaker 2



Mayr et al. (2019) arXiv:1911.02385



Deep Boltzmann Machine

Scaling Neural Quantum States beyond 10000 quantum spins

Rutger Berns - Radboud University, The Netherlands

Johan Mentink - Radboud University, The Netherlands

Philipp Küppers - Radboud University, The Netherlands

Alessandro Pierro - LMU Munich, Germany

NeuralTensor – July 9th 2026

Radboud Universiteit

