Sampling spin systems on the lattice efficiently with transformers

Dawid Zapolski
with Piotr Białas, Piotr Korcyl, Tomasz Stebel, Adam Stefański

Faculty of Physics, Astronomy and Applied Computer Science
Jagiellonian University in Kraków



Polish National Science Center (NCN)
Grant No. 2021/43/D/ST2/03375

Background – what problem do we want to solve?

Sample from a discrete Boltzmann distribution

$$p(\mathbf{s}) = \frac{e^{-\beta E(\mathbf{s})}}{Z}$$

$p(\mathbf{s})$ – target probability

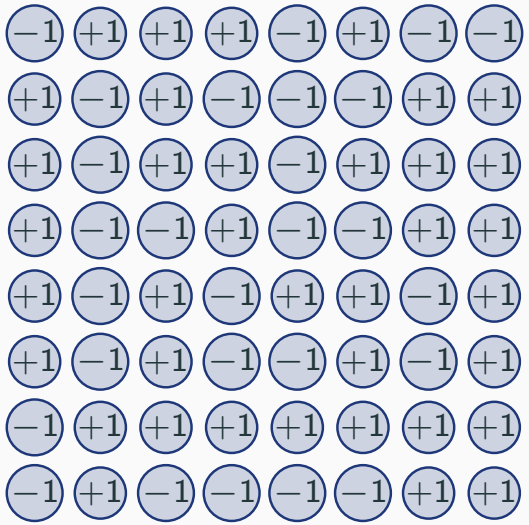
$Z = \sum_{\mathbf{s}} e^{-\beta E(\mathbf{s})}$

Spin models we simulate

$$E(\mathbf{s}) = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j$$

$s_k \in \{-1, 1\}$

$\langle i, j \rangle$ – sum over neighbours.



Spin models we simulate

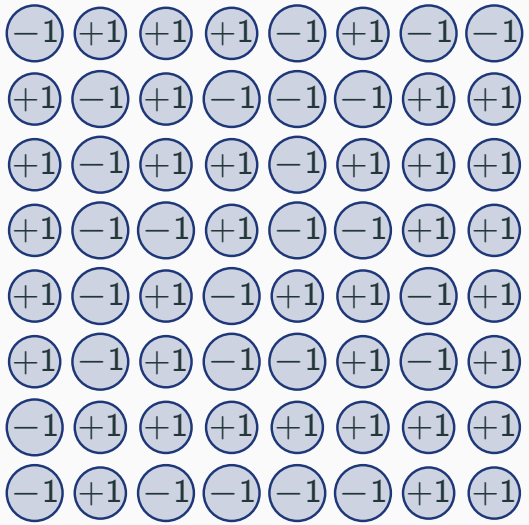
$$E(\mathbf{s}) = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j$$

$s_k \in \{-1, 1\}$

$\langle i, j \rangle$ – sum over neighbours.

Two models we explore:

- Ising model, $J_{ij} = 1$,
- Edwards-Anderson model (EA), $J_{ij} \in \{-1, 1\}$



Traditionally, Markov chain Monte Carlo methods used here but they struggle in some conditions

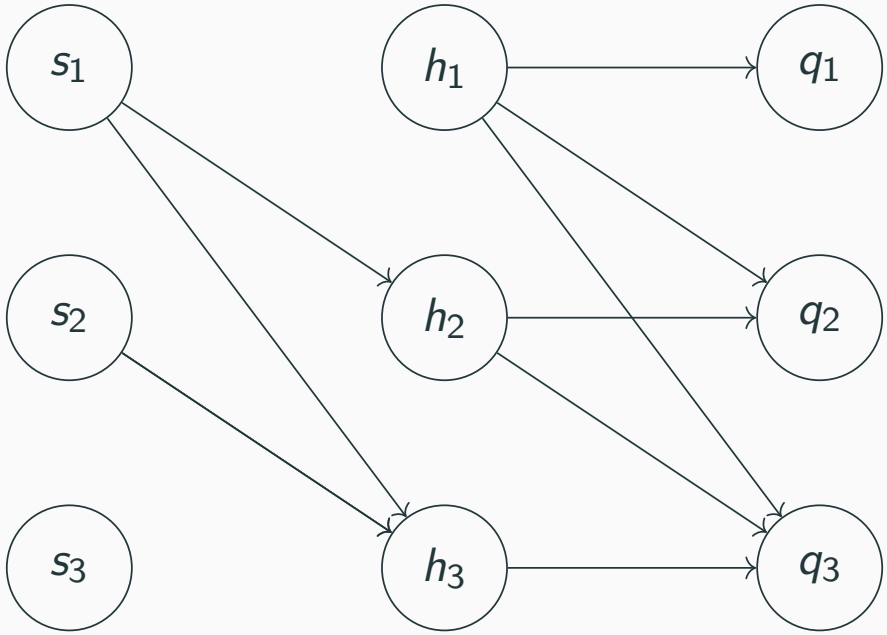
Close to critical point, these methods fail (critical slowing down)

Neural networks: alternative approach

Variational Autoregressive Networks as a pioneering work

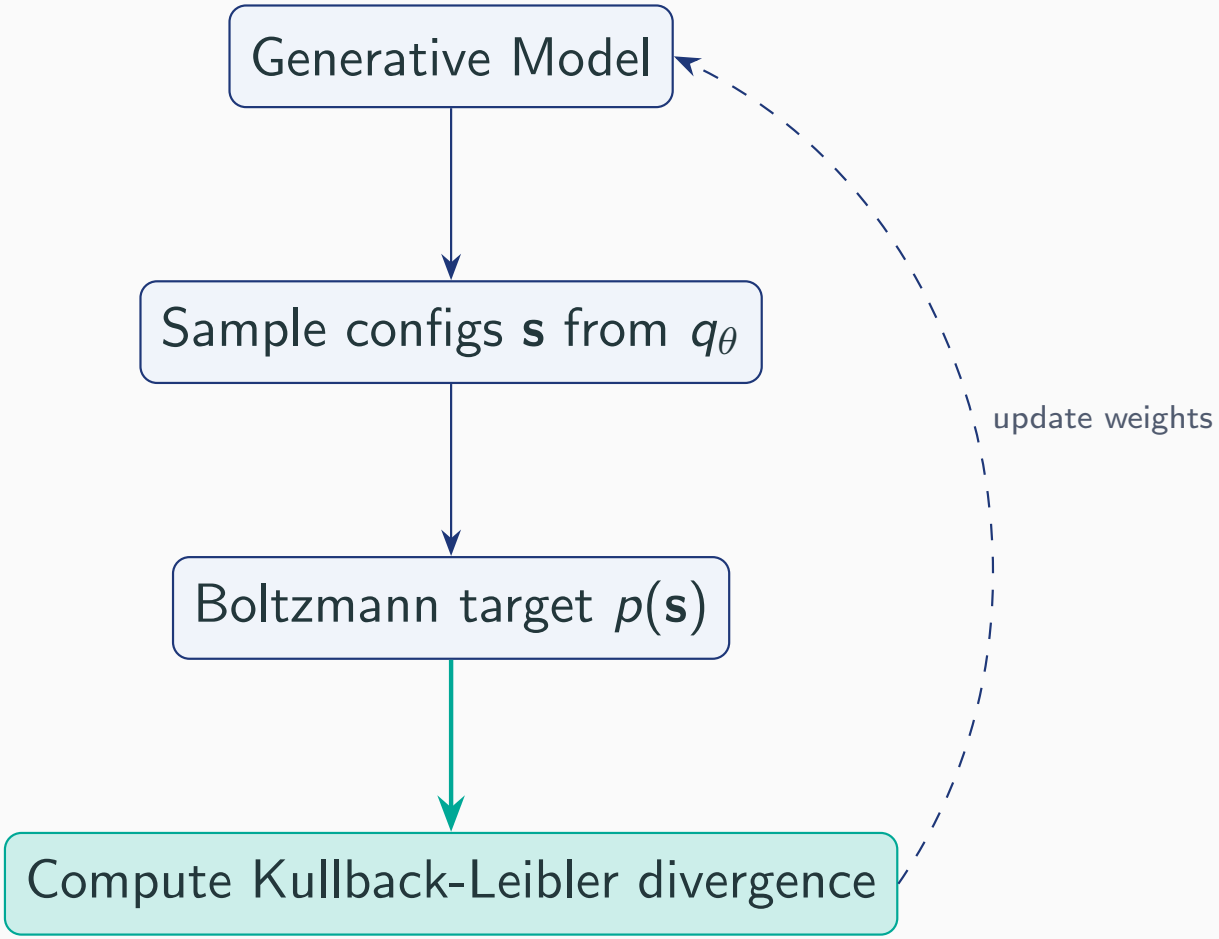
Generate a configuration spin by spin, simulating probability of the whole configuration as

$$q_\theta(\mathbf{s}) = \prod_i q_\theta(s_i|s_{<i})$$



D. Wu, L. Wang, P. Zhang, *Solving statistical mechanics using variational autoregressive networks*, Phys. Rev. Lett. **122**, 080602 (2019).

Training like in reinforcement learning
no external configurations are needed



Training with reversed Kullback-Leibler divergence

$$D_{KL}(q_\theta|p) = \sum_{\mathbf{s}} q_\theta(\mathbf{s}) \log \frac{q_\theta(\mathbf{s})}{p(\mathbf{s})}$$
$$\approx \frac{1}{N} \sum_{i=0}^{N} \log \frac{q_\theta(\mathbf{s}_i)}{p(\mathbf{s}_i)}, \quad \mathbf{s}_i \sim q_\theta$$

q_θ – probability given by the neural networks

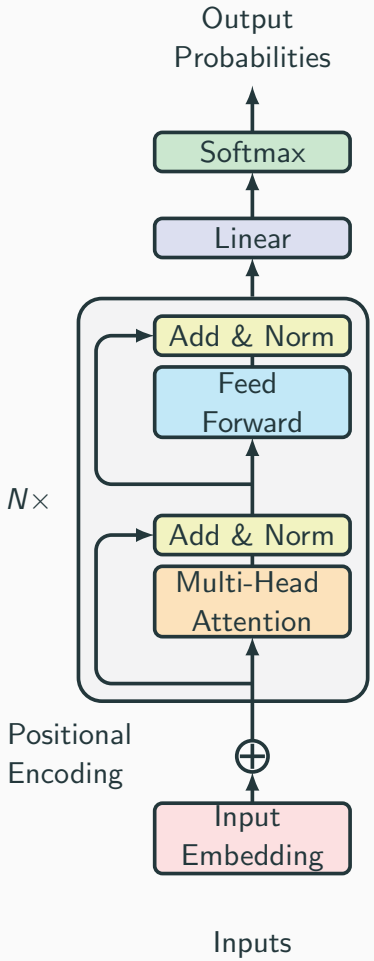
p – target probability

Effective sample size defines a metric describing the quality of training

$$\mathrm{ESS} = \frac{\langle \hat{w}(\mathbf{s}) \rangle_{q_\theta}^2}{\langle \hat{w}(\mathbf{s})^2 \rangle_{q_\theta}}, \quad \text{where} \quad \hat{w}(\mathbf{s}) = \frac{e^{-\beta E(\mathbf{s})}}{q_\theta(\mathbf{s})}$$

Our work

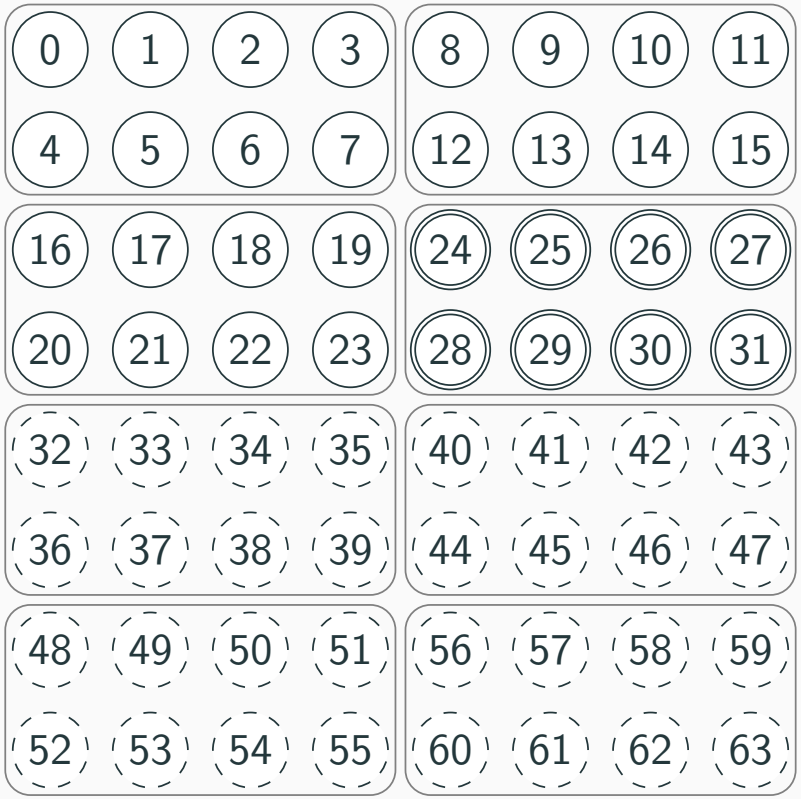
Transformer-based sampler: expressive but computationally demanding



- Decoder-only transformer
- Model based on GPT-2
- Just one or two blocks

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, *Attention is all you need*, Advances in Neural Information Processing Systems **30** (NeurIPS 2017).

Our solution – divide lattice into patches



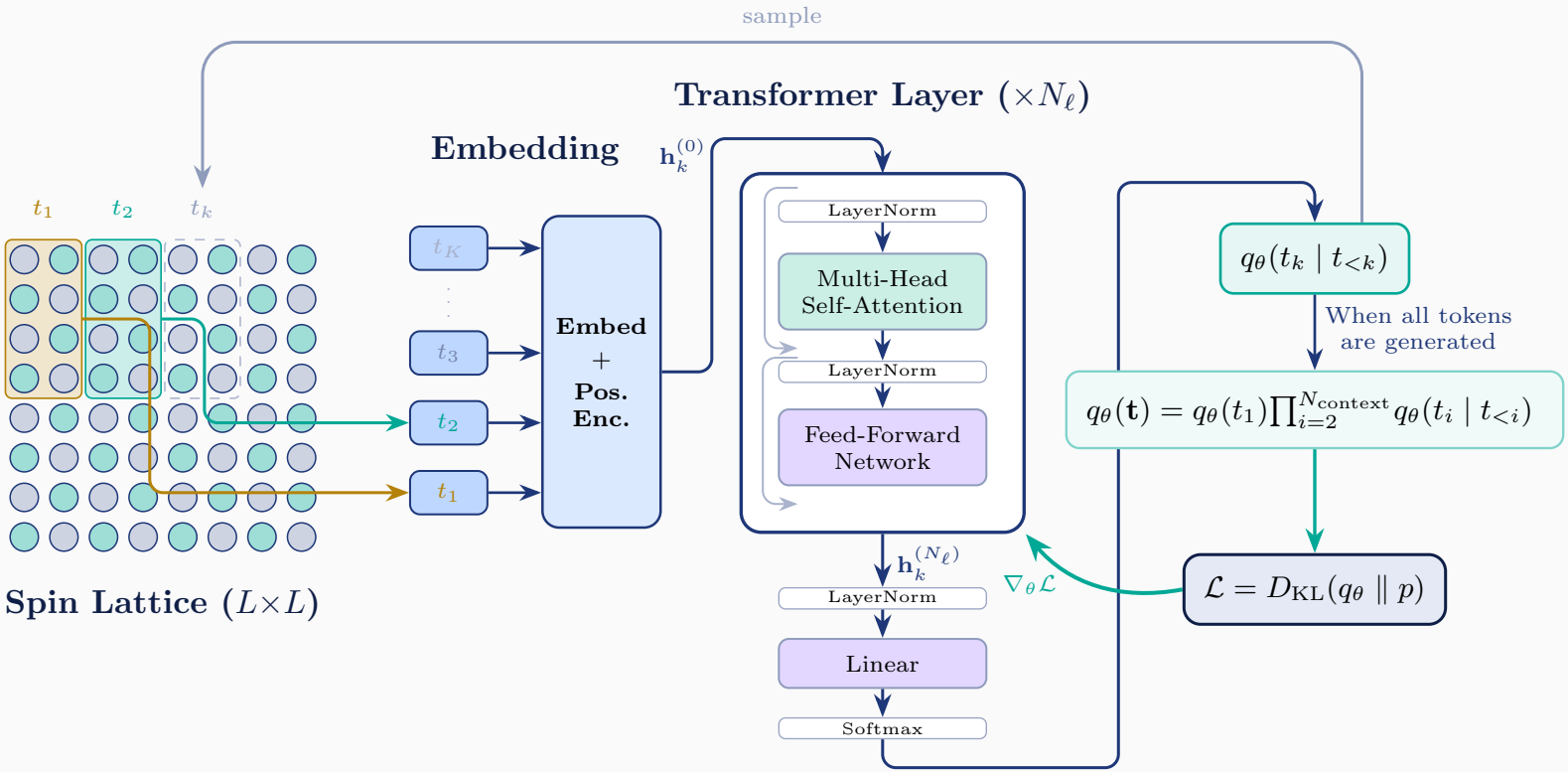
Map patches to tokens

$$0 \leftrightarrow \begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{pmatrix} \quad 1 \leftrightarrow \begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{pmatrix}$$

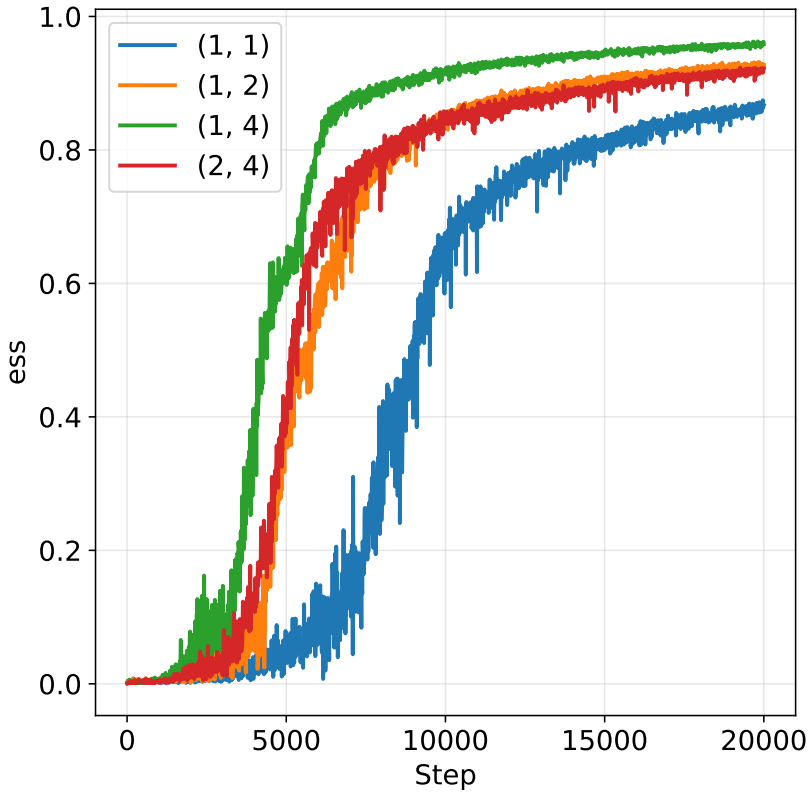
$$2 \leftrightarrow \begin{pmatrix} -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{pmatrix} \quad 4 \leftrightarrow \begin{pmatrix} -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 \end{pmatrix}$$

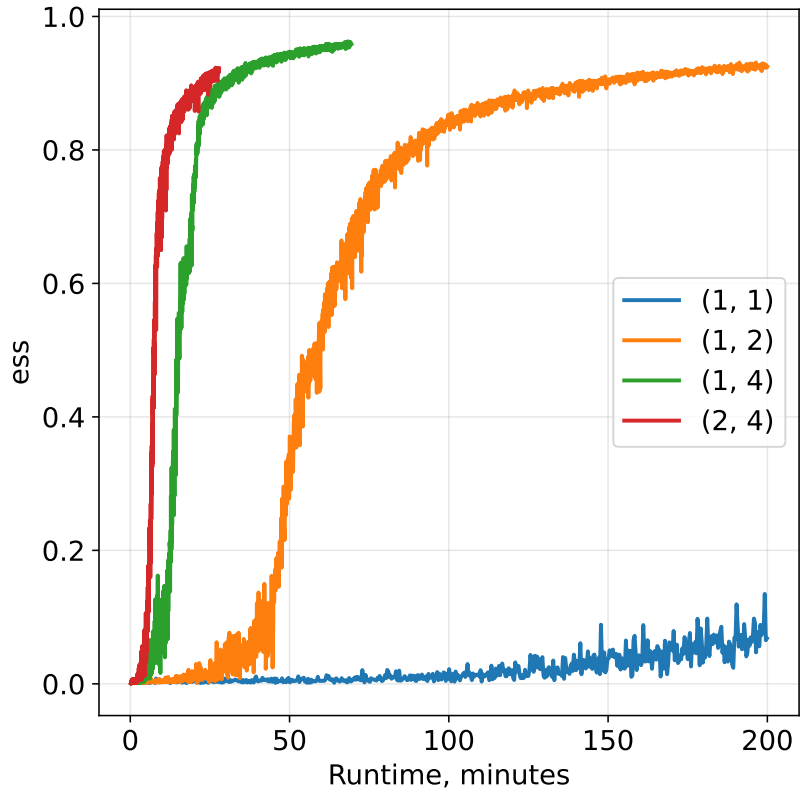
$$31 \leftrightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix}$$

How does it work?



Results – influence of the patch size





32×32 Ising, β_c

Limits on the patch size

$$O\left(n_{\text{embed}} \times 2^{A} \times \frac{L^2}{A}\right) + O\left(n_{\text{embed}}^2 \times \frac{L^2}{A}\right) + O\left(\frac{L^4}{A^2} \times n_{\text{embed}}\right)$$

A – area of the patch L – system size

Limits on the patch size

$$O\left(n_{\text{embed}} \times 2^{A} \times \frac{L^2}{A}\right) + O\left(n_{\text{embed}}^2 \times \frac{L^2}{A}\right) + O\left(\frac{L^4}{A^2} \times n_{\text{embed}}\right)$$

A – area of the patch L – system size

In practice, $10 \lesssim A \lesssim 12$ seems optimal for $100 \leq L \leq 200$

Results – comparison with other methods

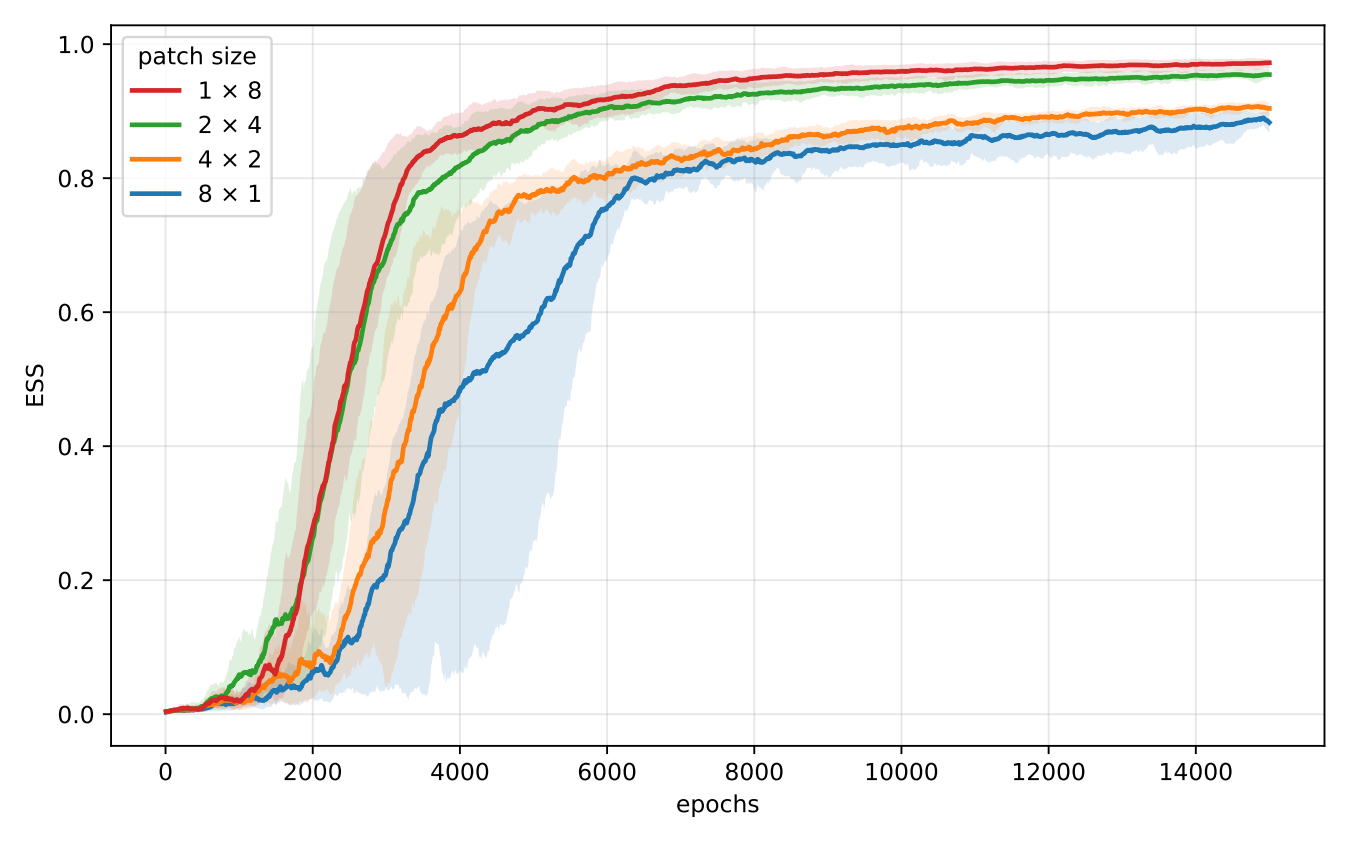
algorithm	system size L	time of training	ESS (↑)
HAN	128	48h	$< 10^{-3}$
RiGCS ($\beta = 0.44$)	128	14.5h (A100 GPU)	0.03
tVAN one layer, 2×4	128	228h	0.43
tVAN two layers, 2×4	128	267h	0.68

Ising model at β_c

Our model (tVAN) trained on 4× AMD MI250x (8 GPUs with 60GiB VRAM each)

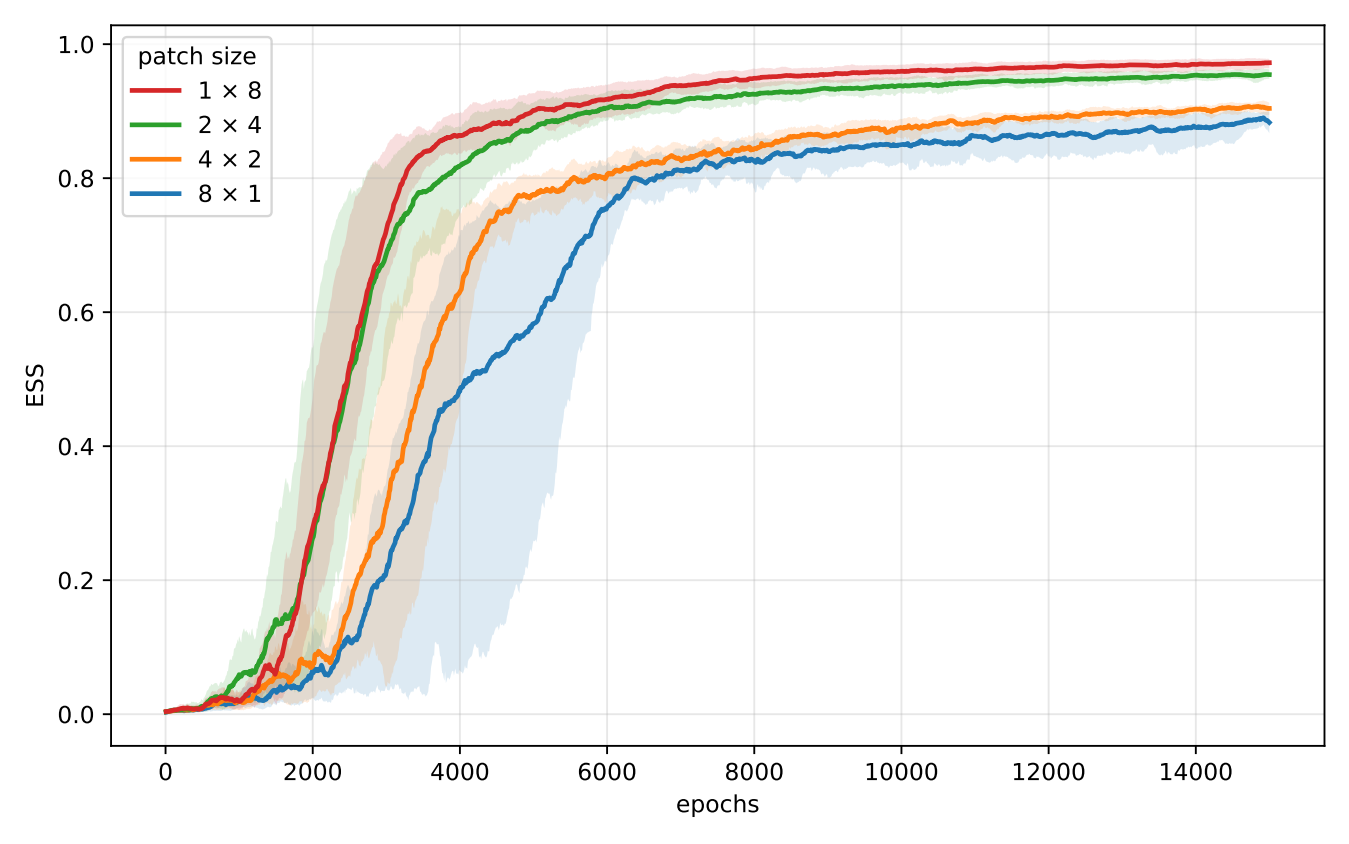
P. Białas, P. Korcyl, T. Stebel, *Hierarchical autoregressive neural networks for statistical systems*, Comput. Phys. Commun. **281**, 108502 (2022).
A. Singha et al., *Multilevel Generative Samplers for Investigating Critical Phenomena*, ICLR 2025, pp. 25450–25483.

Shape of the patch influences results as well



64×64 Ising, β_c

Shape of the patch influences results as well

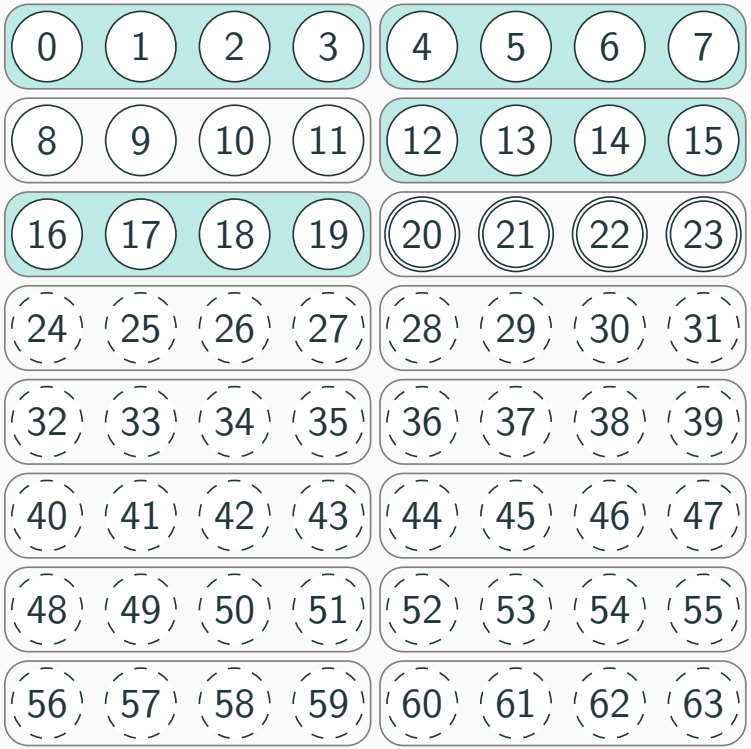


64×64 Ising, β_c

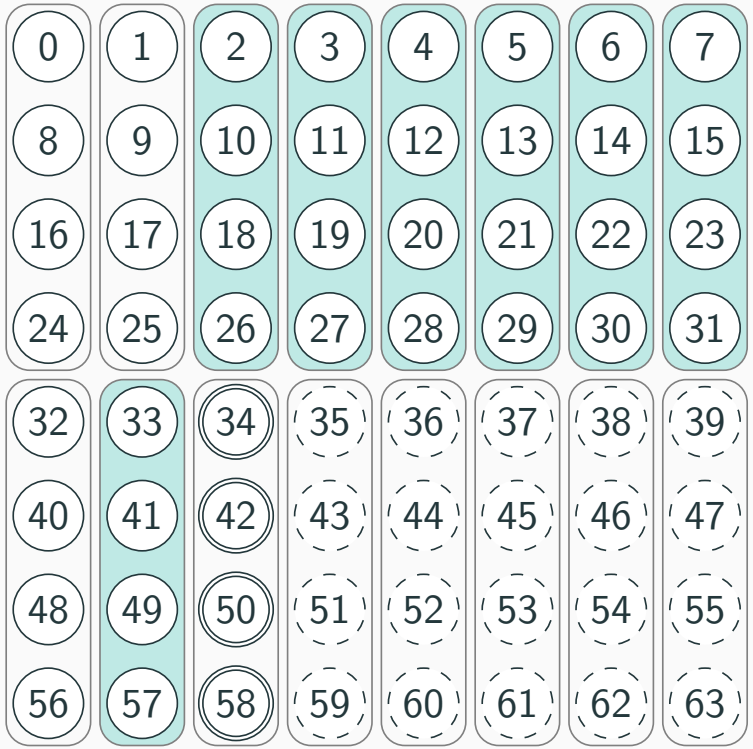
Hypothesis:

Patches in the direction of sampling reduce effective context length

Effective context



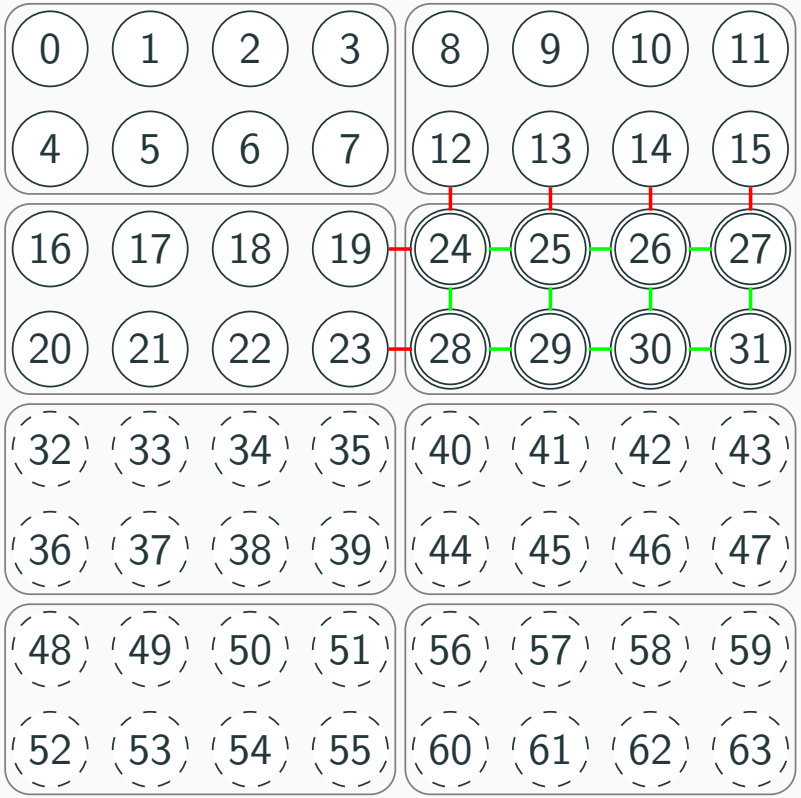
1×4 patches



4×1 patches

Can we do even better?
Adding physics as a prior

Prior probabilities



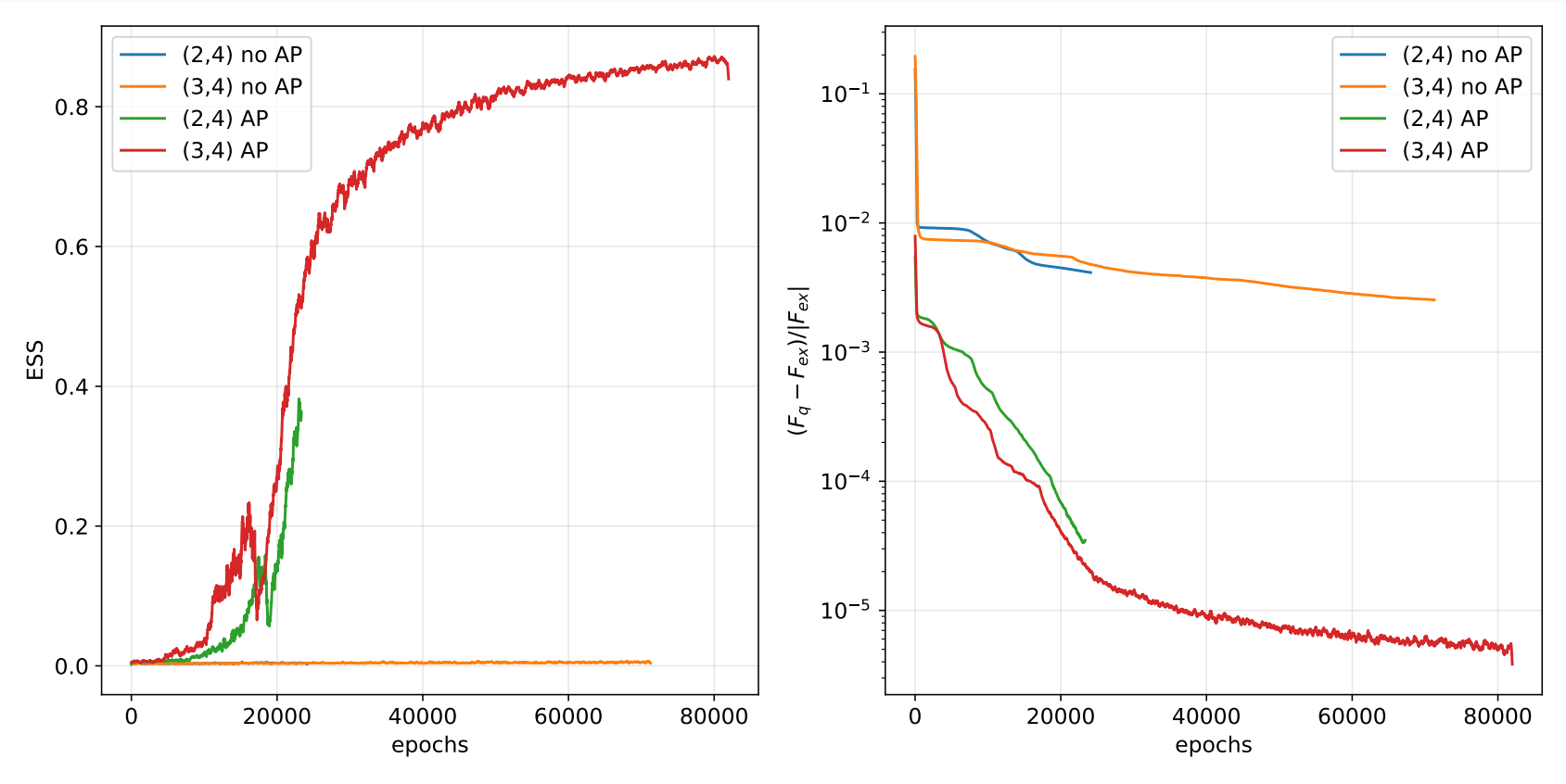
$$E_i^j = -\sum_{\text{internal}} J_{kl} s_k s_l - \sum_{\text{neigh.of } i} J_{kl} s_k s_l$$

$$q(t_i^j | t_{<i}) = \text{softmax}^j\left(-\beta E_i^j + f^j(t_{<i})\right),\ f^j \text{ – from transformer}$$

I. Biazzo, D. Wu, G. Carleo, *Sparse Autoregressive Neural Networks for Classical Spin Systems*, Mach. Learn.: Sci. Technol. **5** (2024), 025074, doi:10.1088/2632-2153/ad5783.

P. Białas, P. Korcyl, T. Stebel, D. Zapolski, *Variational Autoregressive Networks with probability priors*, arXiv:2605.16020 (2026).

Results – added probabilities (AP)



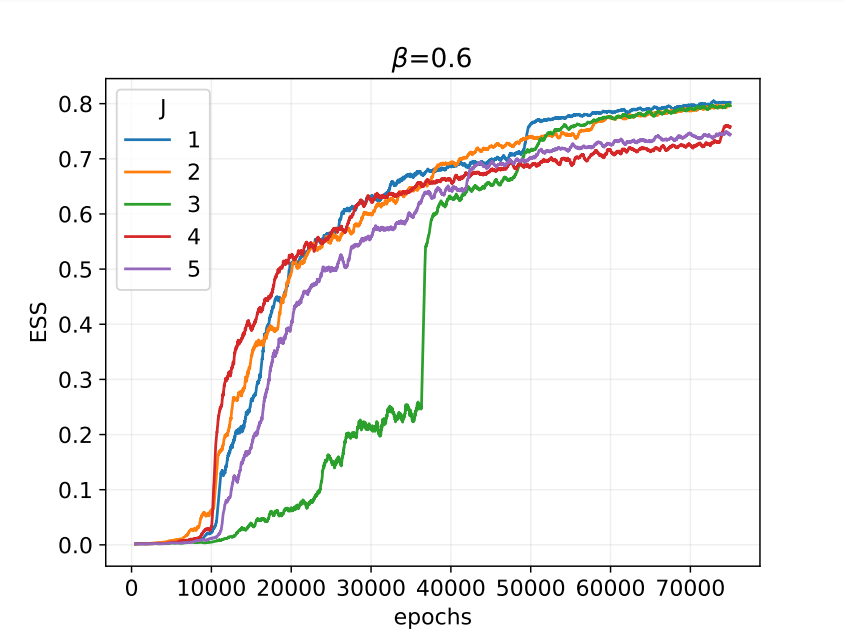
120×120 Ising, β_c, all runs took 48 hours

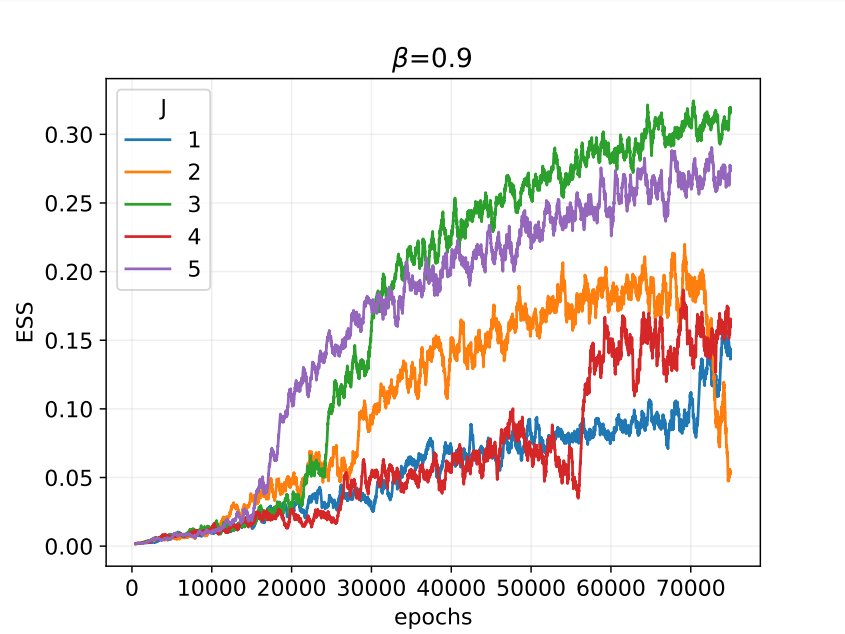
Results – added probabilities (AP)

algorithm	system size L	time of training	$(F_q - F)/\lvert F\rvert$	ESS
HAN	128	48h	1.5×10^{-4}	$< 10^{-3}$
RiGCS ($\beta = 0.44$)	128	14.5h (single A100 GPU)	1.1×10^{-4}	0.03
tVAN one layer, 2×4	128	228h	2.5×10^{-5}	0.43
tVAN two layers, 2×4	128	267h	1.1×10^{-5}	0.68
tVAN one layer + AP, 2×4	128	82h	5.5×10^{-6}	0.84
tVAN one layer + AP, 3×4	180	192h	8.8×10^{-6}	0.59

Ising model at β_c

Results: EA spin glass





64×64 EA, without beta annealing

Conclusions

- Transformers with patches make sampling more efficient than previous models

Conclusions

- Transformers with patches make sampling more efficient than previous models
- Adding simple priors to the model can significantly speed up training time