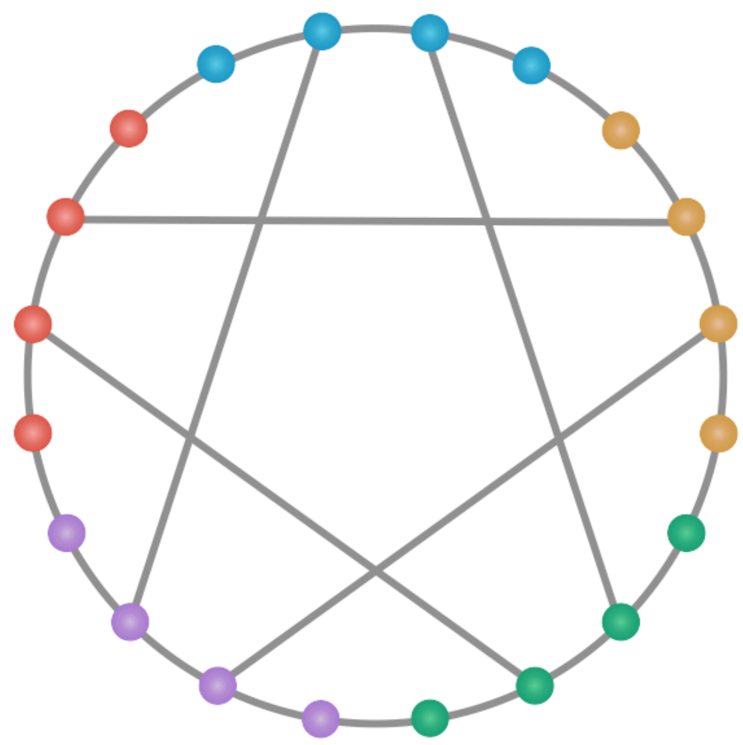
Contracting Arbitrary Tensor Networks



In collaboration with

@ITP,CAS: Sujie Li, Feng Pan, Pengfei Zhou

Pan Zhang
ITP,CAS

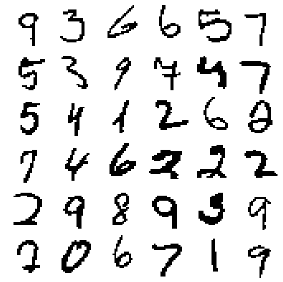
Workshop on Tensor
Network States 2019
Dec. 4, 2019

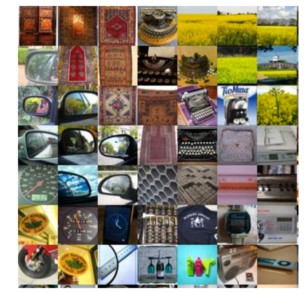
Tensor Networks

- In physics: wave functions
- Out of physics:
  - Revealing internal low-rank structures (CP, Tucker, TT rank)
  - Compression data, optimization
  - Learning: (kerneled) classification, generative modeling
  - Inference in graphical models
  - Simulating quantum circuits

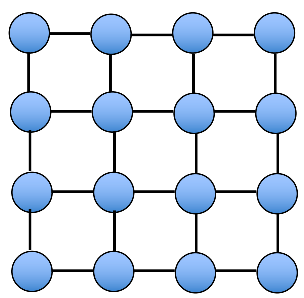
Applying tensor networks to

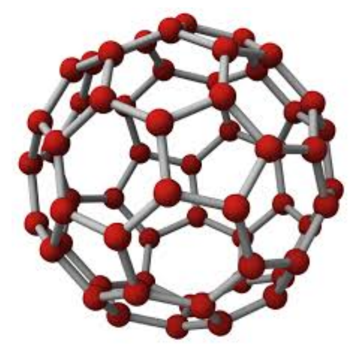
- Machine learning: representing data distribution



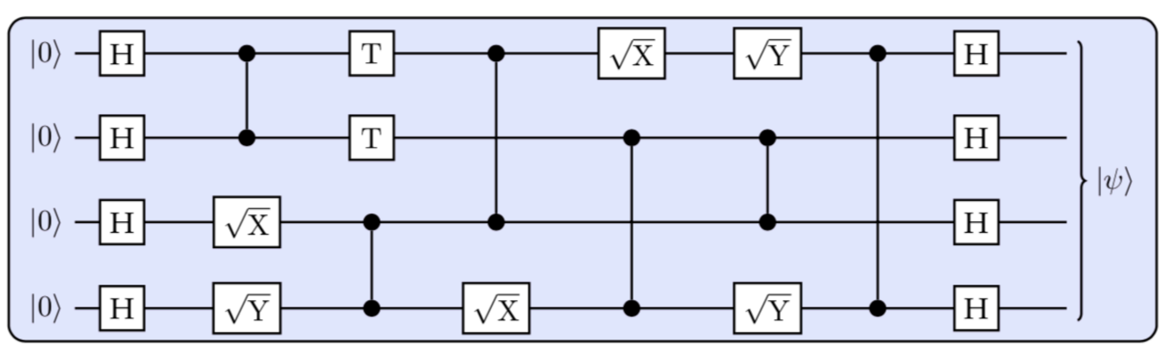


- Graphical model: representing a Boltzmann distribution



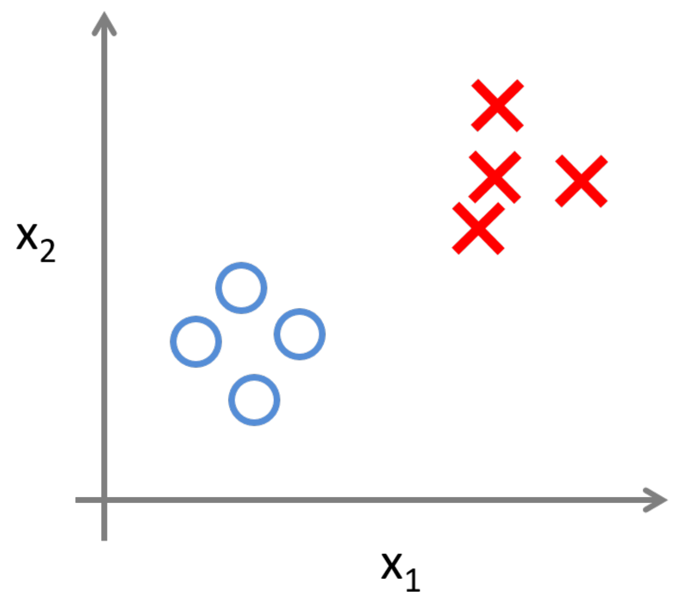


- Quantum circuit simulations: a graphical model with complex temperature



Supervised and unsupervised learning

Supervised Learning

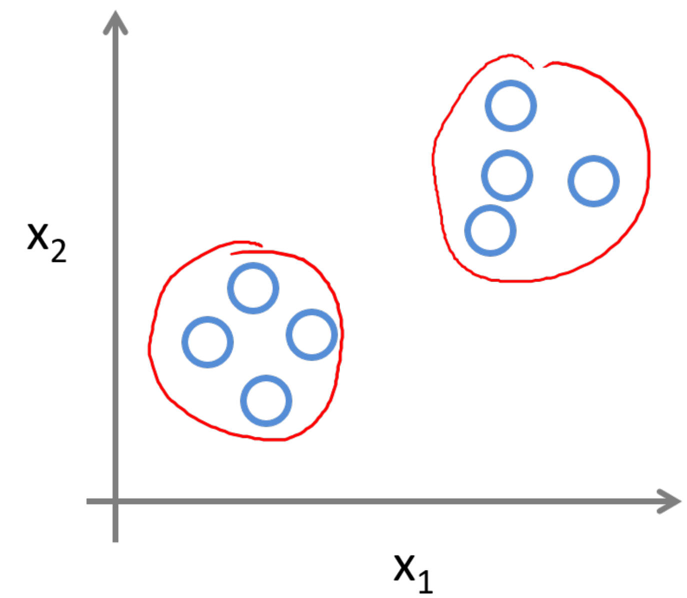


Predicting Labels

$$p(\mathbf{y}|\mathbf{x})$$

Classification

Unsupervised Learning

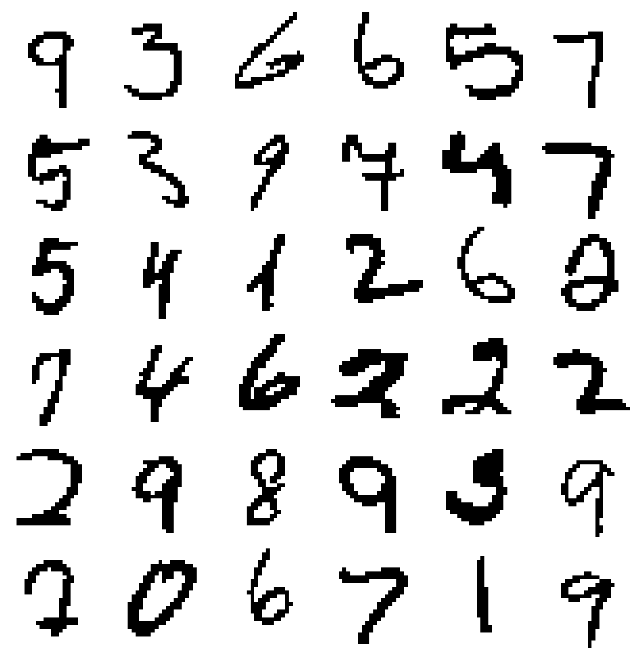


Finding structures in the data

$$P_{\text{data}}(\mathbf{x}) \propto \sum_{\mathbf{x}^{(i)} \in \text{data}} \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

Generative model

Feature mapping to Hilbert space

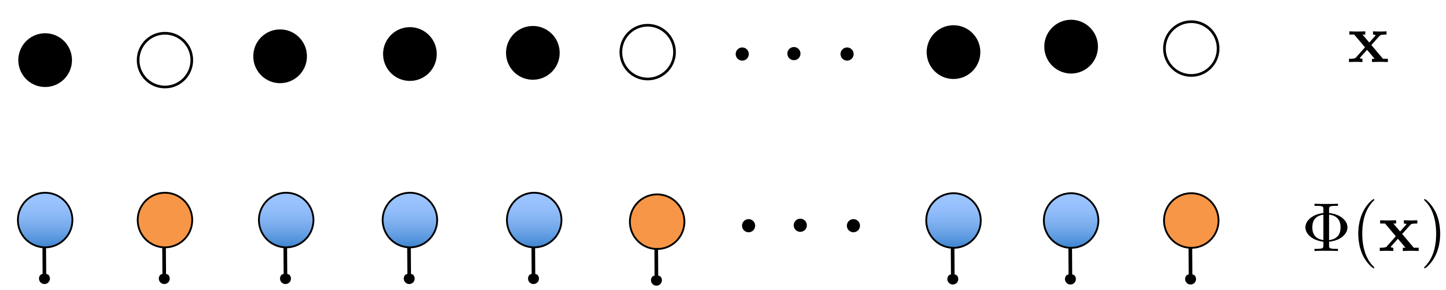


MNIST handwritten digits with labels 0,1,...,9

60,000 training images

10,000 test images

$6 \in \{1,0\}^{28\times 28}$ Zig-zagling →



● $0 \rightarrow \begin{pmatrix}1\\0\end{pmatrix}$

○ $1 \rightarrow \begin{pmatrix}0\\1\end{pmatrix}$

$$\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\otimes\cdots\otimes\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}$$

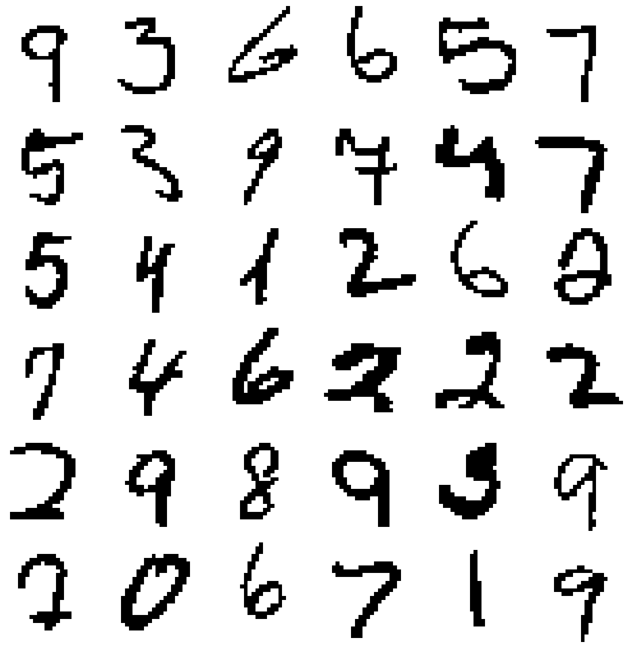
$$\text{Dimension: } 2^{28\times 28} = 2^{784}$$

Rebentrost, Mohseni, Lloyd, Phys. Rev. Lett. 113, 130503 (2014)

Havlíček, Córcoles, Temme, Harrow, Kandala, Chow, Gambetta, Nature 567, 209 (2019)

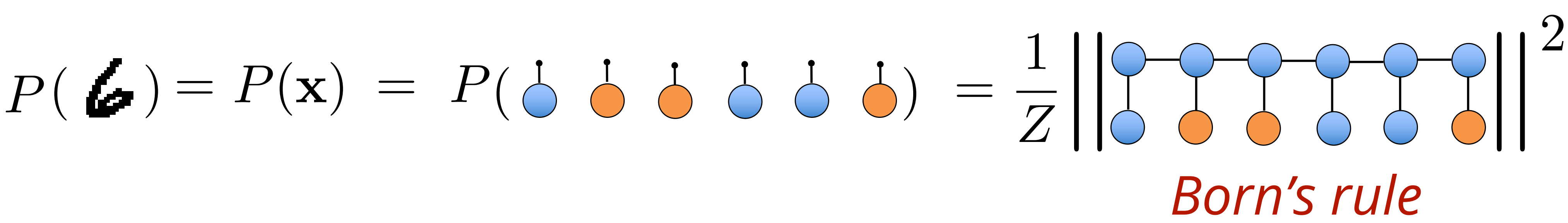
Stoudenmire, Schwab, NIPS (2016)

Parametrizing the joint distribution: MPS Born Machine

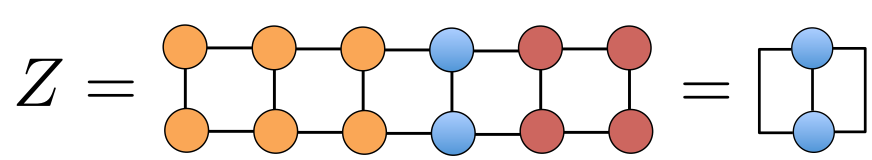


The largest possible space for binary data

The right space for computing the "partition function"

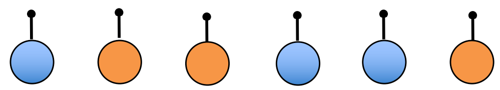


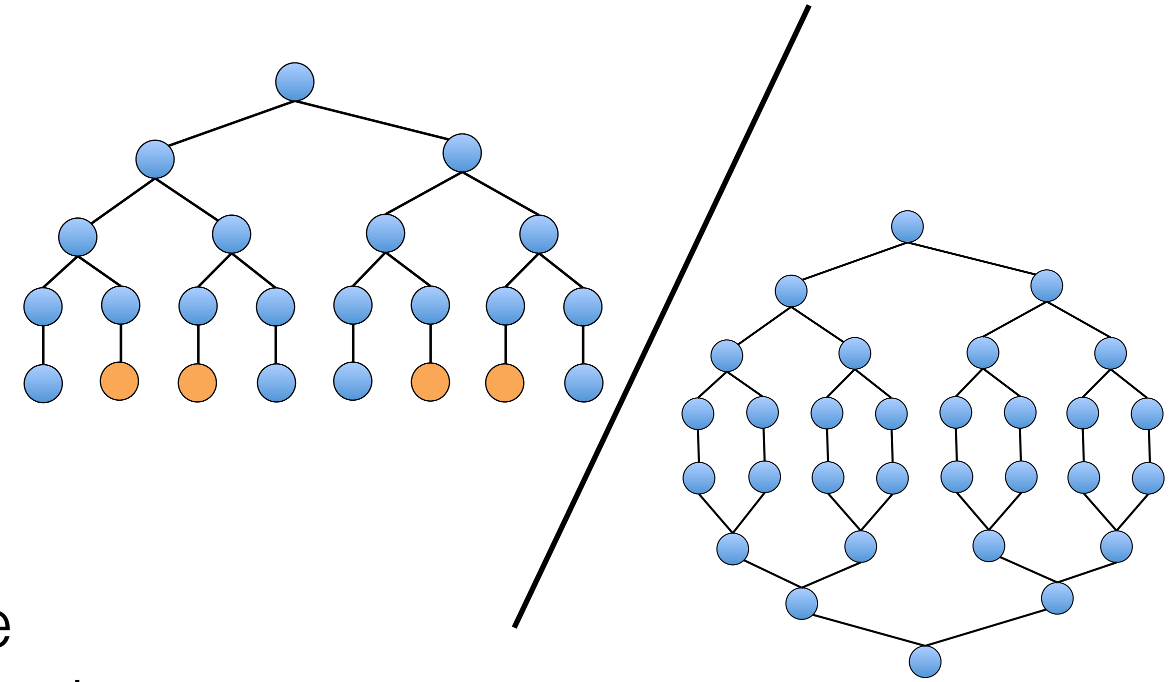
Z can be computed exactly !



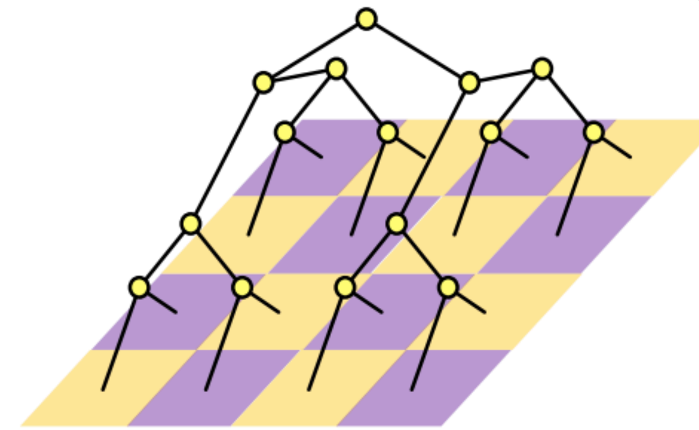
Han, Wang, Fan, Wang, PZ, Phys. Rev. X 8, 031012 (2018)

Tree Tensor Network Born machine

$P(\mathbf{x}) = P($  $) =$



- Features:
  - Analogous to MPS: trackable likelihood, direct sampling, and canonical forms.
  - Good prior for 2-D images
  - Better correlation length (in practice)
- Limitations:
  - Higher computational complexity than MPS Born machine



(a)

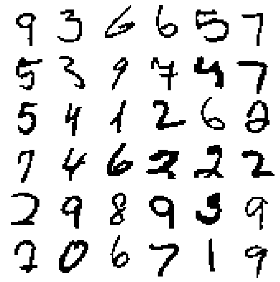
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

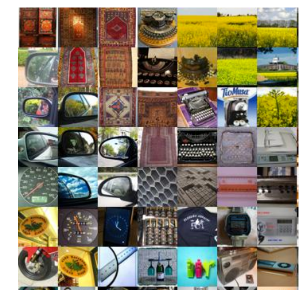
1	3	9	11
2	4	10	12
5	7	13	15
6	8	14	16

Cheng, Wang, Xiang, PZ, Phys. Rev. B 99, 155131 (2019)

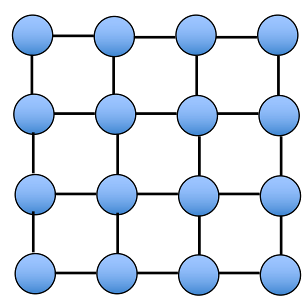
Applying tensor networks to

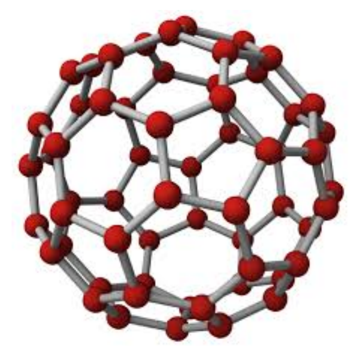
- Machine learning: representing data distribution



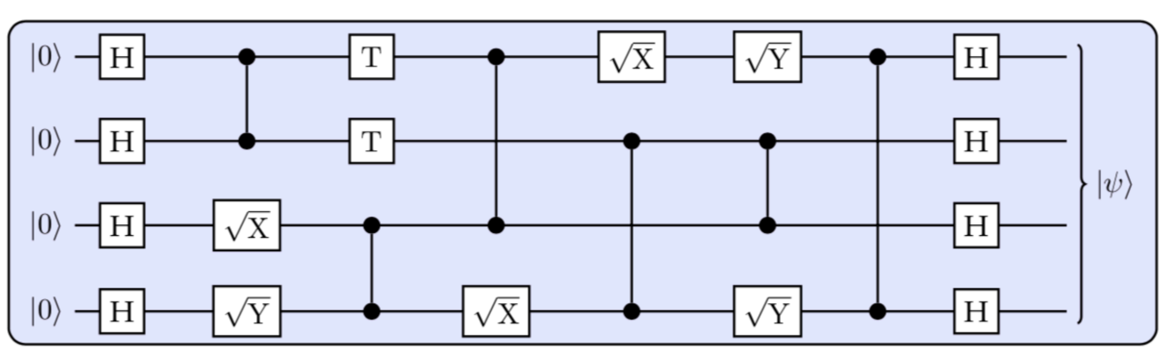


- Graphical model: representing a Boltzmann distribution





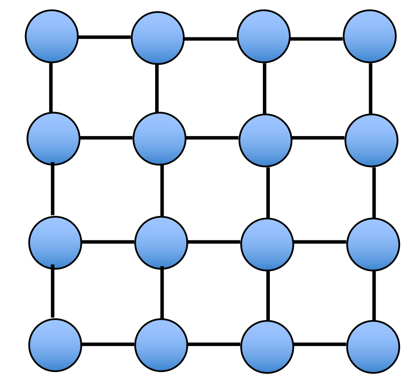
- Quantum circuit simulations: a graphical model with complex temperature



Graphical models

Example: Ising (spin glass) models

$\mathbf{S} = \{+1, -1\}^n$ ↑ ↑ ↑ ↓ ↑ ↓ ↓ ↓ ↓ ↓ ↑ ↑

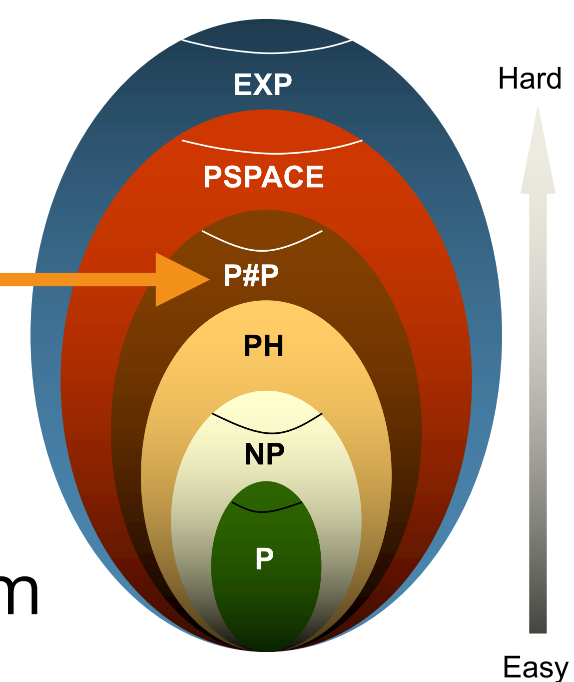


Computing partition function of the 2-D Ising model

$$P(\mathbf{S}) = \frac{1}{Z} e^{-\beta E(\mathbf{S})} \qquad Z = \sum_{\mathbf{S}} e^{-\beta E(\mathbf{S})}$$

- Estimating the free energy:
- Computing macroscopic observables
- Sampling the Boltzmann distribution
- … …

#P-hard problem



Mean-field approximations

- Statistical properties of variables are functions of self-consistent *fields (marginal probabilities)*
- Converting many-body problems to single/few - body problems
- Fields are functions of *means* of neighborhood variables
- Results to self-consistent (message passing) equations:
  - Variational mean-field (Product distribution)
  - Bethe approximation / belief propagation
  - Thouless-Anderson-Palmer equations
  - Kikuchi loop series expansions
  - Expectation Propagation
  - ……

Van der Waals 1873

Weiss, Pierre 1907

Bethe 1935

Kikuchi 1951

Plefka 1982

Yedidia et. al., 2001

Mezard, Parisi 2001

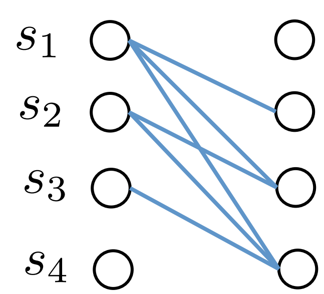
Improve mean-field methods using Neural networks

- Representing joint distribution using chain rule of conditional probabilities.

$$q(\mathbf{s}) = \prod_i q(s_i|\mathbf{s}_{j<i})$$

$$\begin{aligned} q(s_1, s_2, s_3, s_4) &= q(s_4|s_3, s_2, s_1)q(s_3, s_2, s_1) \\ &= q(s_4|s_3, s_2, s_1)q(s_3|s_2, s_1)q(s_2, s_1) \\ &= q(s_4|s_3, s_2, s_1)q(s_3|s_2, s_1)q(s_2|s_1)q(s_1) \end{aligned}$$

Fully Visible Belief Network [Frey 1998]



$$\hat{s}_1 = \mathrm{sigmoid}(0) = q(s_1 = 1)$$

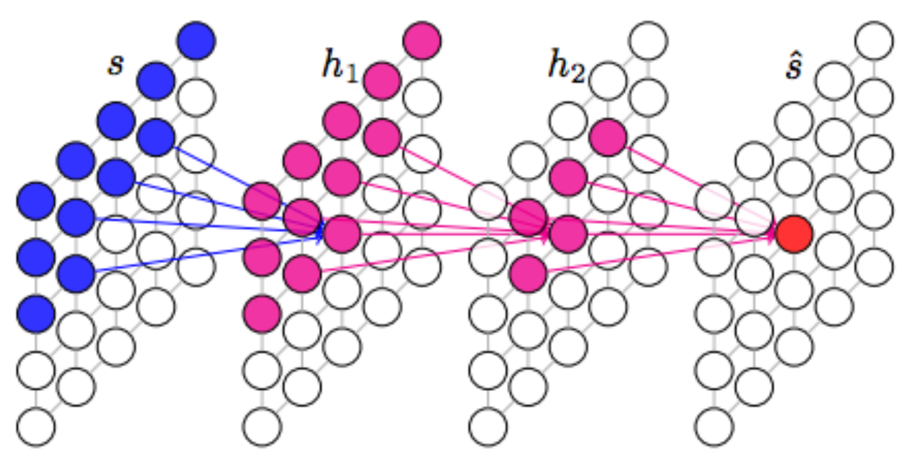
$$\hat{s}_2 = \mathrm{sigmoid}(w_{12}s_1) = q(s_2 = 1|s_1)$$

$$\hat{s}_3 = \mathrm{sigmoid}(w_{13}s_1 + w_{23}s_2) = q(s_3 = 1|s_2, s_1)$$

$$\hat{s}_4 = \mathrm{sigmoid}(w_{14}s_1 + w_{24}s_2 + w_{34}s_4) = q(s_4 = 1|s_3, s_2, s_1)$$

$$q(s1, s2, s3, s4) = \hat{s}_1\hat{s}_2\hat{s}_3\hat{s}_4$$

- Variational Autoregressive Networks (VAN)



D. Wu, L. Wang, PZ, PRL 122, 080602 (2019)

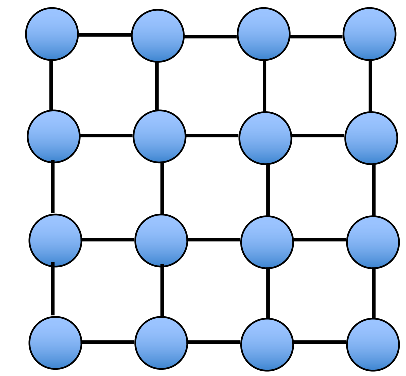
Can Tensor Networks do better ?

Converting graphical models to tensor networks

Example: Ising (spin glass) models

$\mathbf{S} = \{+1, -1\}^n$

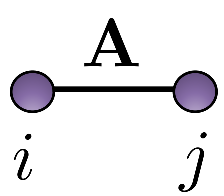
$\uparrow \uparrow \uparrow \downarrow \uparrow \downarrow \downarrow \downarrow \downarrow \downarrow \uparrow \uparrow$

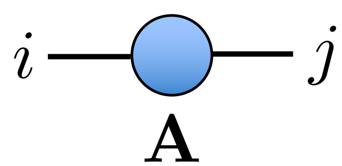


Computing partition function of the 2-D Ising model

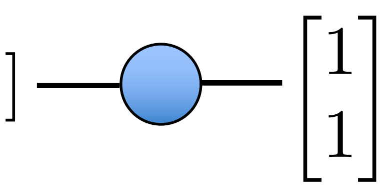
$$P(\mathbf{S}) = \frac{1}{Z} e^{-\beta E(\mathbf{S})} \qquad Z = \sum_{\mathbf{s}} e^{-\beta E(\mathbf{S})}$$

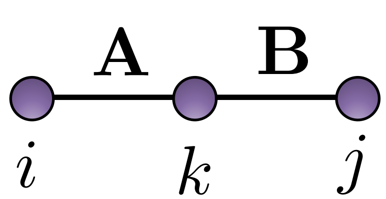
Probability distribution is a tensor, hence can be represented by a tensor network in general.

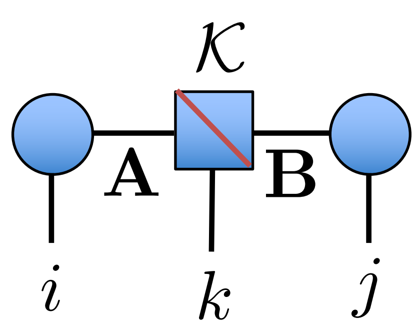




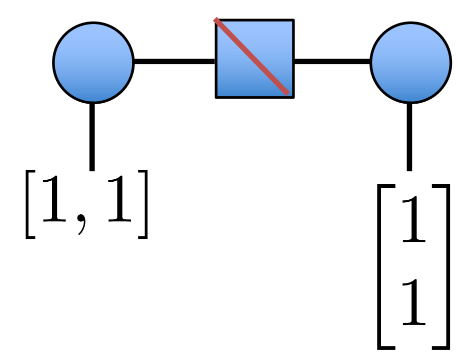
$$Z = \sum_{i,j} A_{ij} = \mathbf{1}^T A \mathbf{1} = \ [1, 1]$$



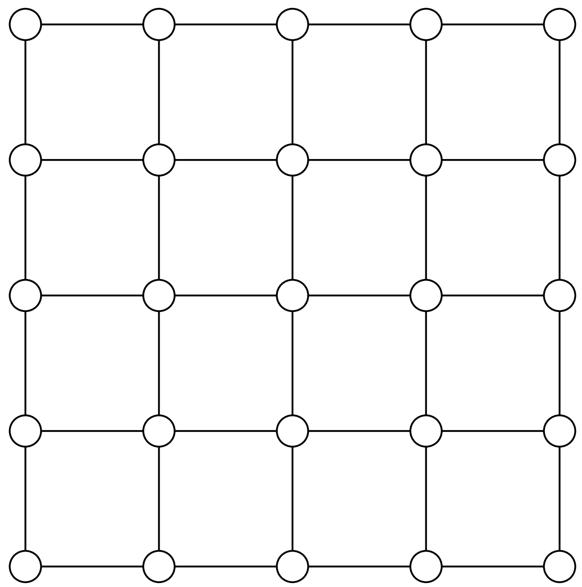


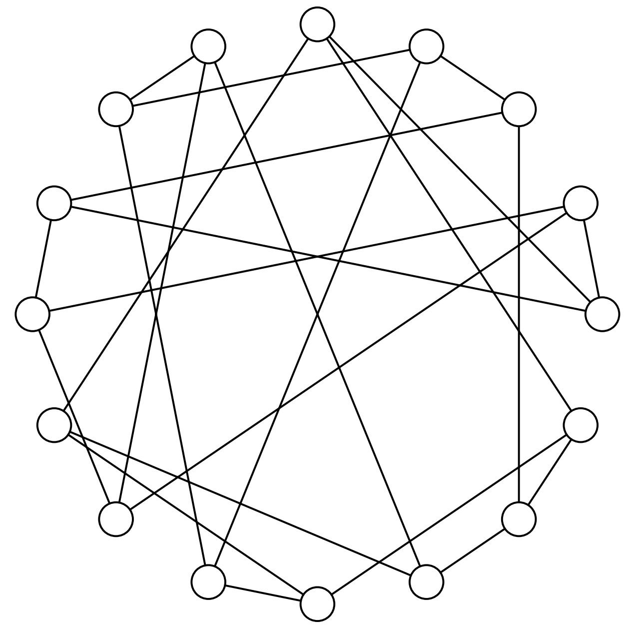


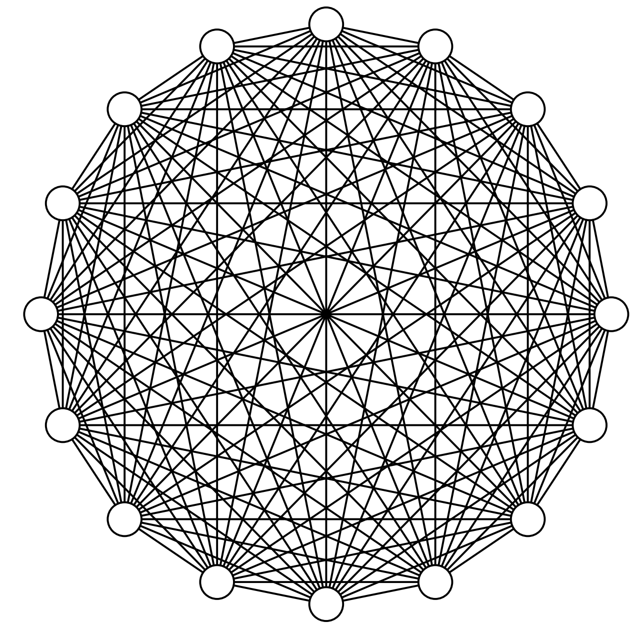
$$Z = \mathbf{1}^{\mathrm{T}} \mathbf{A}\mathbf{B}\mathbf{1} =$$

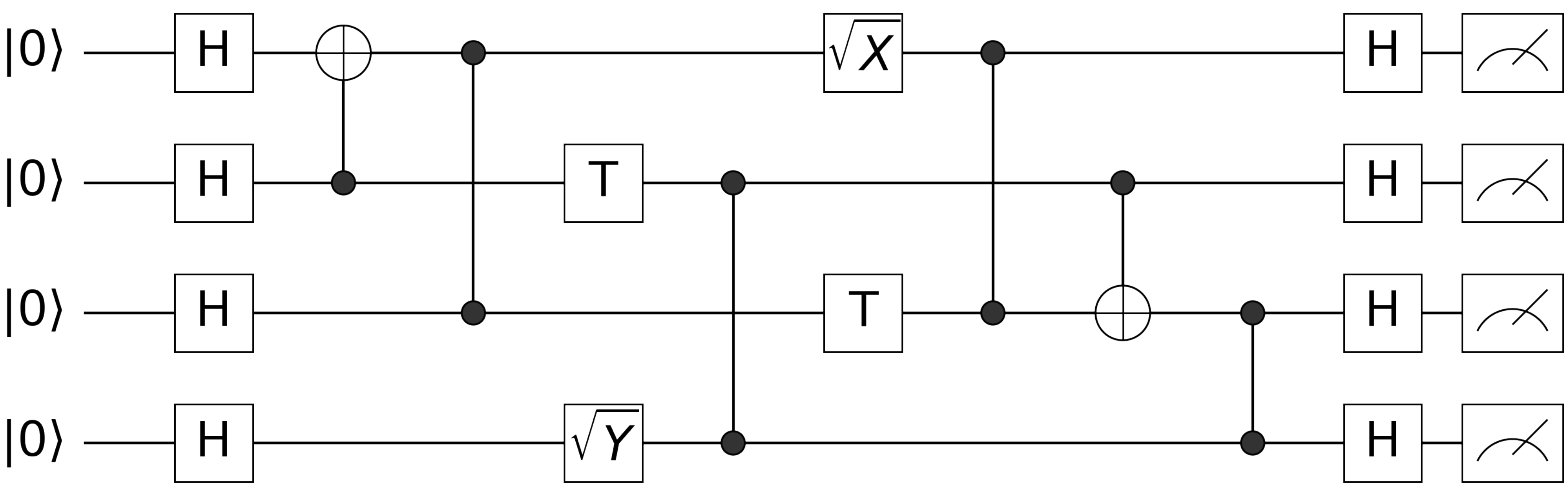


Tensor networks we would like to contract









Questions:

How to deal with large intermediate tensors during the contraction process?

Stored and compressed using MPS

How to do accurate approximations?

Canonical form, SVD

Can TNs handle strong entanglements induced by long-range interactions?

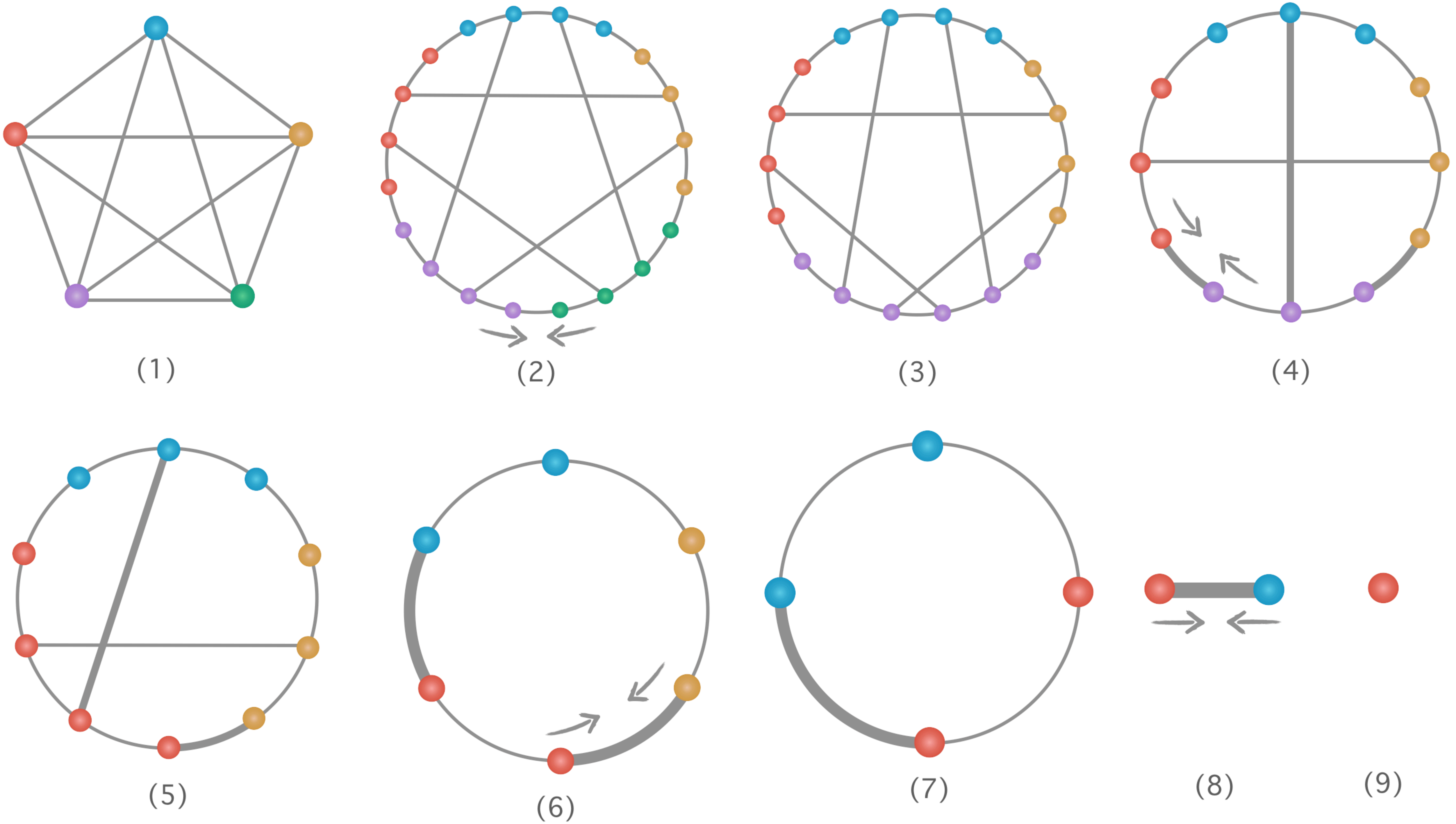
No guarantee,
but could be better than other methods

Scalability of approximations?

trade off between
accuracy and bond dimensions

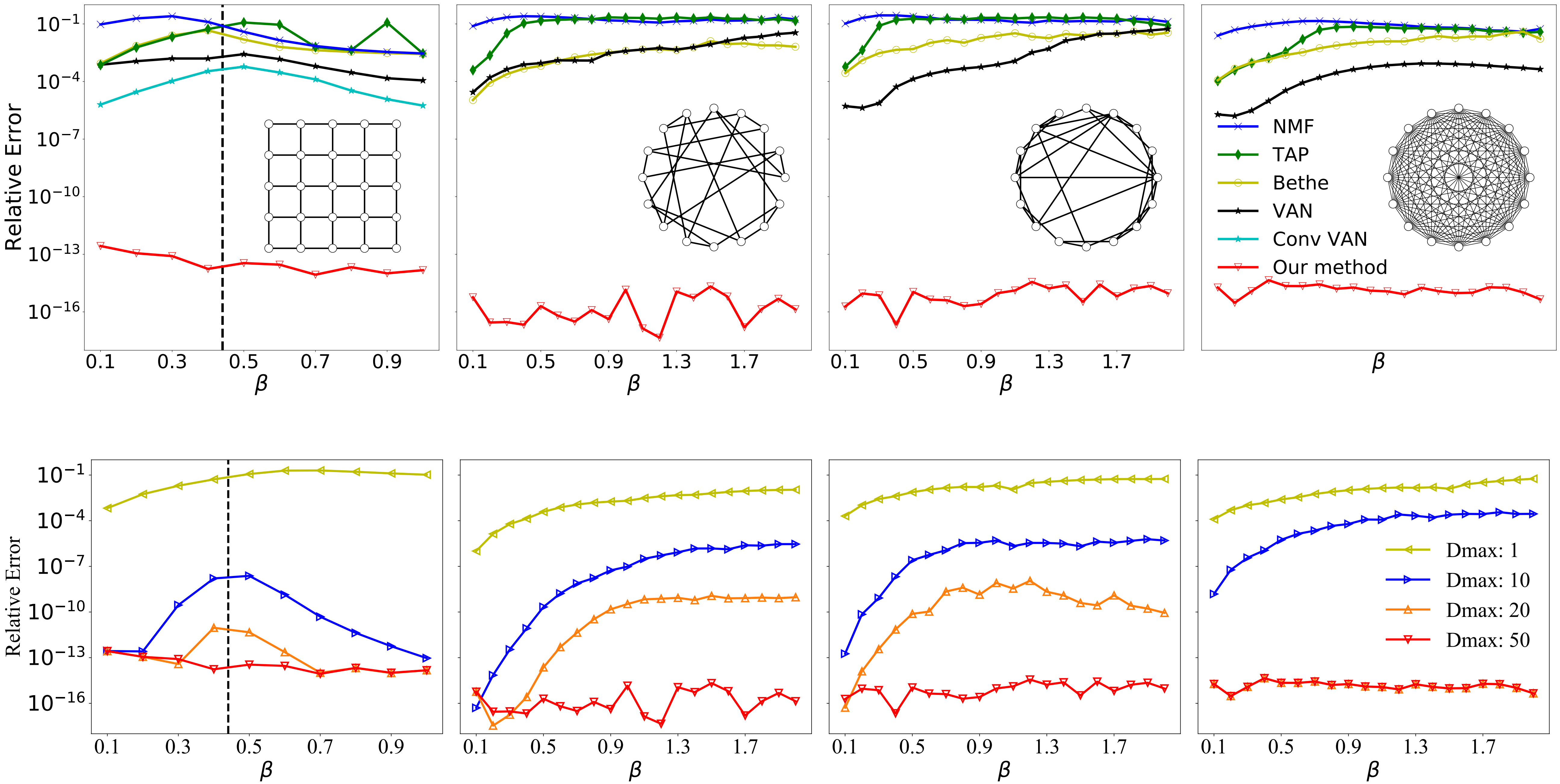
→ **MPS calculus**

MPS calculus



F. Pan, P. Zhou, S. Li , PZ, arXiv:1912.03014

Computing free energy of spin glasses



16x16 2D lattice

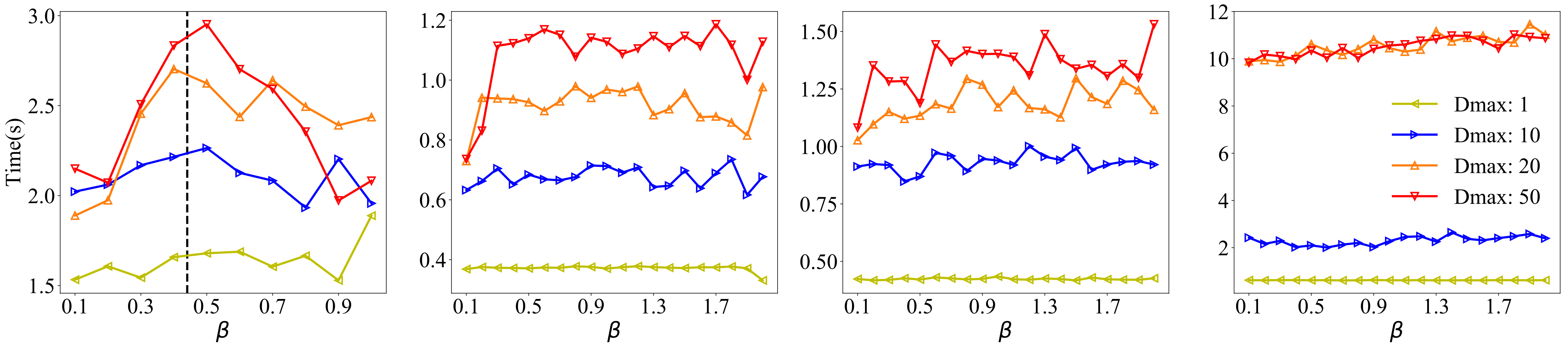
RRG graphs n=80, k=3

Small-world n=70, c=4

SK model n=20

F. Pan, P. Zhou, S. Li , PZ, arXiv:1912.03014

Time used



16x16 2D lattice

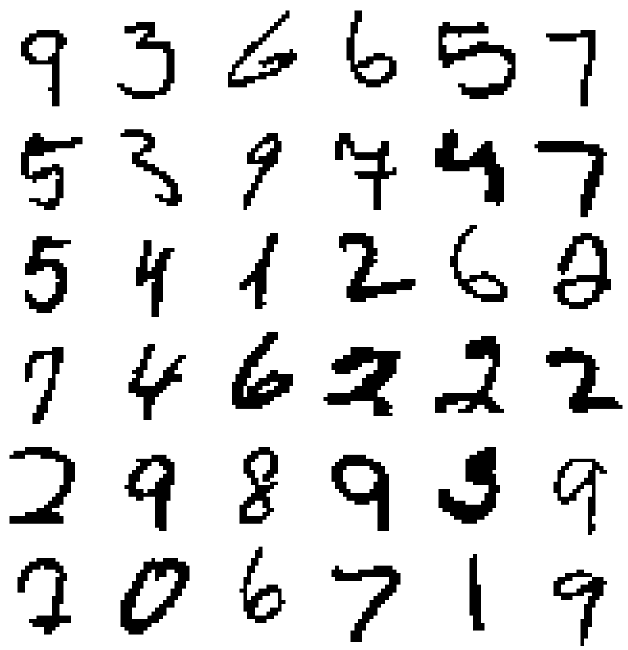
RRG graphs n=80, k=3

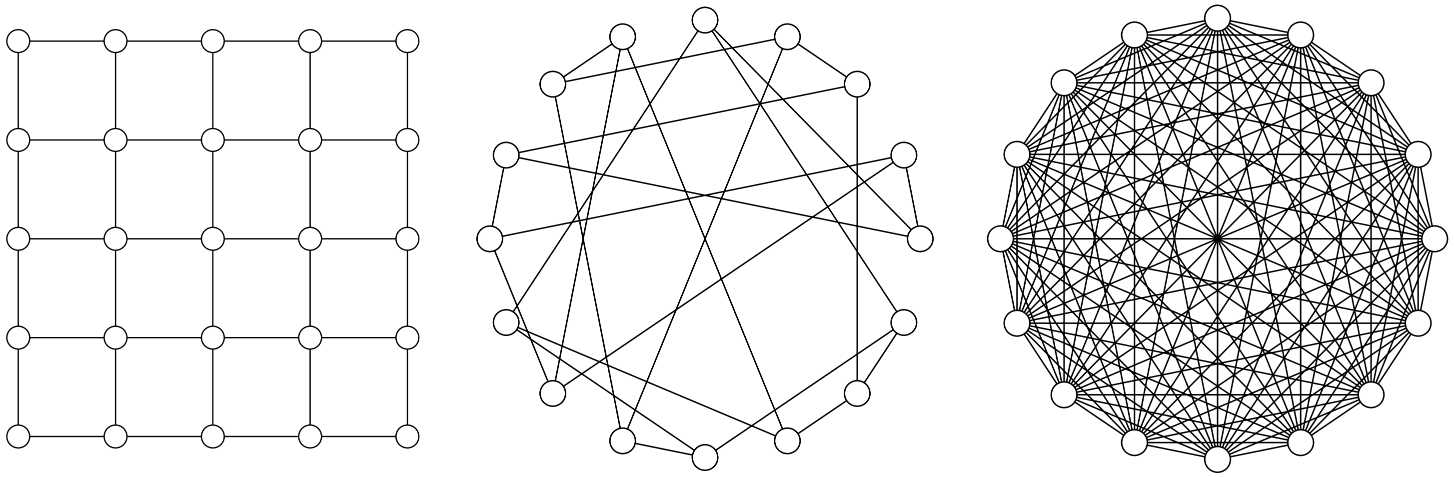
Small-world n=70, c=4

SK model n=20

F. Pan, P. Zhou, S. Li , PZ, arXiv:1912.03014

From inference to learning





$$P_{\text{data}}(\mathbf{x}) \propto \sum_{\mathbf{x}^{(i)} \in \text{data}} \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

$$P_{\text{BM}}(\mathbf{x}) = \frac{1}{Z} e^{-\beta \sum_{(ij)} J_{ij} x_i x_j}$$

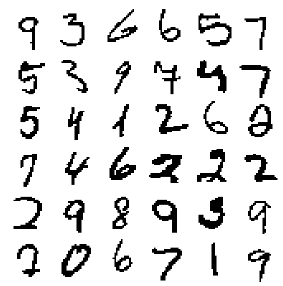
$$D_{\text{KL}}(P_{\text{data}} \| P_{\text{GM}}) = \sum_{\mathbf{x}} P_{\text{data}}(\mathbf{x}) \log \frac{P_{\text{data}}(\mathbf{x})}{P_{\text{GM}}(\mathbf{x})}$$

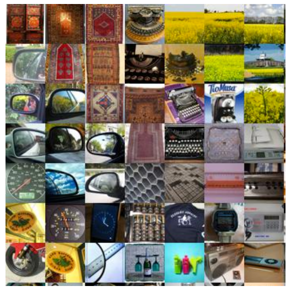
$$\begin{aligned} \widehat{J} &= \arg\min_{\mathbf{J}} D_{\text{KL}}(P_{\text{data}} \| P_{\text{GM}}) \\ &= \arg\min_{\mathbf{J}} E + \boxed{\log Z} \end{aligned}$$

The Loss function can be estimated using tensor networks
Gradients can be computed using back propagation

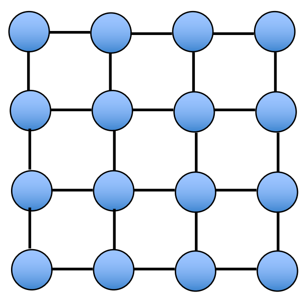
Applying tensor networks to

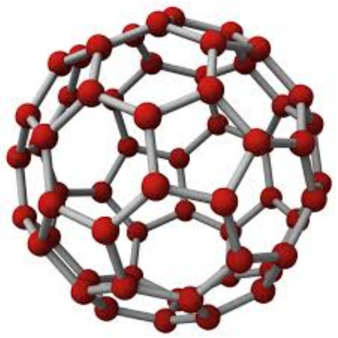
- Machine learning: representing data distribution



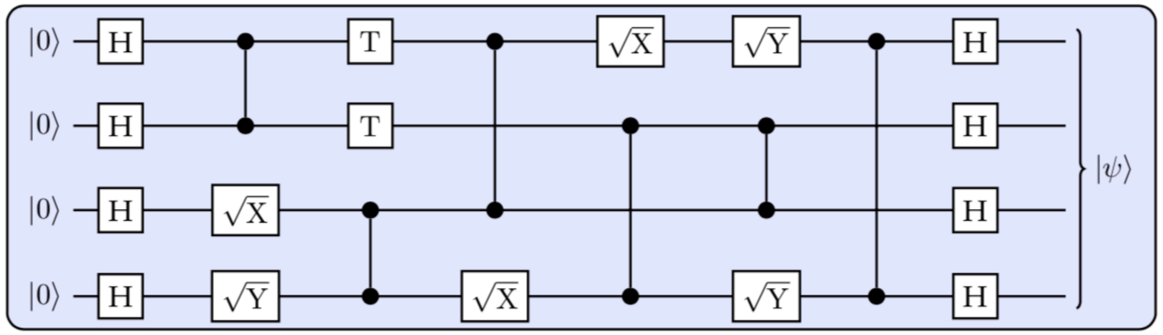


- Graphical model: representing a Boltzmann distribution

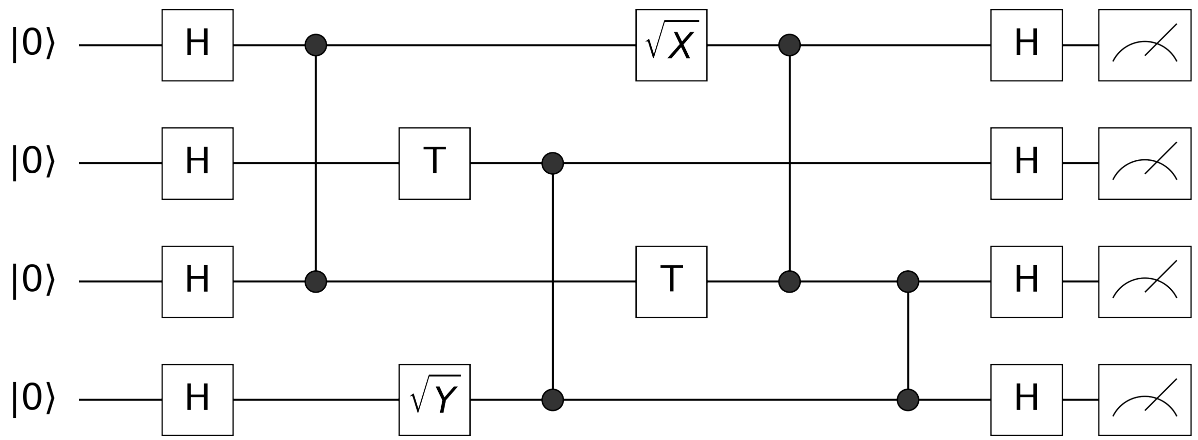




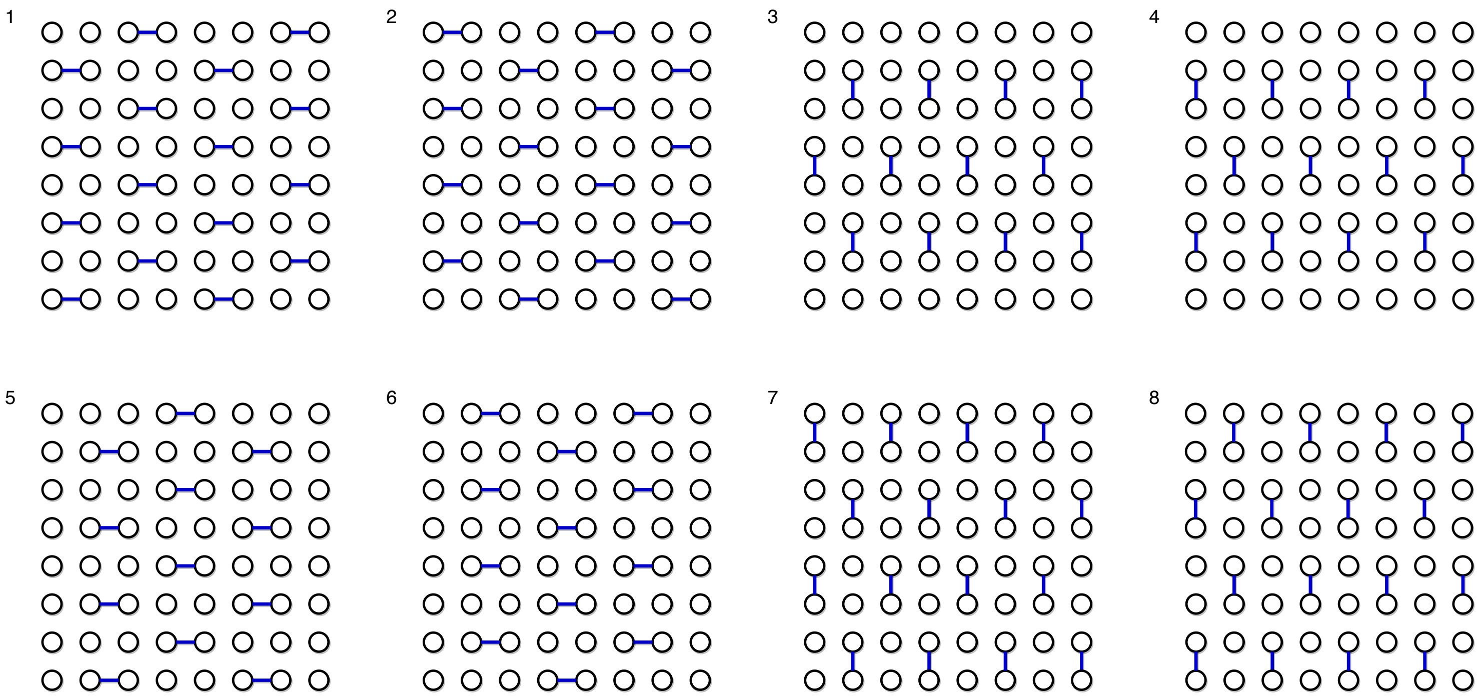
- Quantum circuit simulations: a graphical model with a complex temperature



Simulate shallow quantum circuits



1. Apply a Hadamard gate to each quit
2. Apply CZ chosen sequentially from 8 configurations
3. Apply *T*, $\sqrt{X}$, or $\sqrt{Y}$ to each qubit
4. Repeat step 2 and 3 to L layers
5. Apply a Hadamard gate to each quit



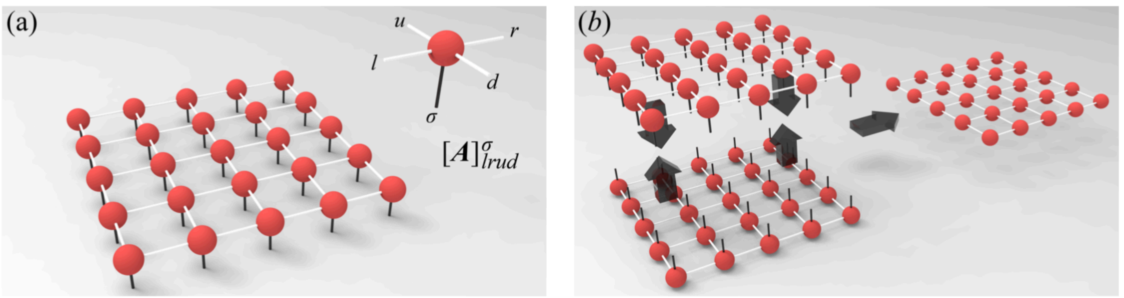
Robeva et al, arXiv:1710.01437

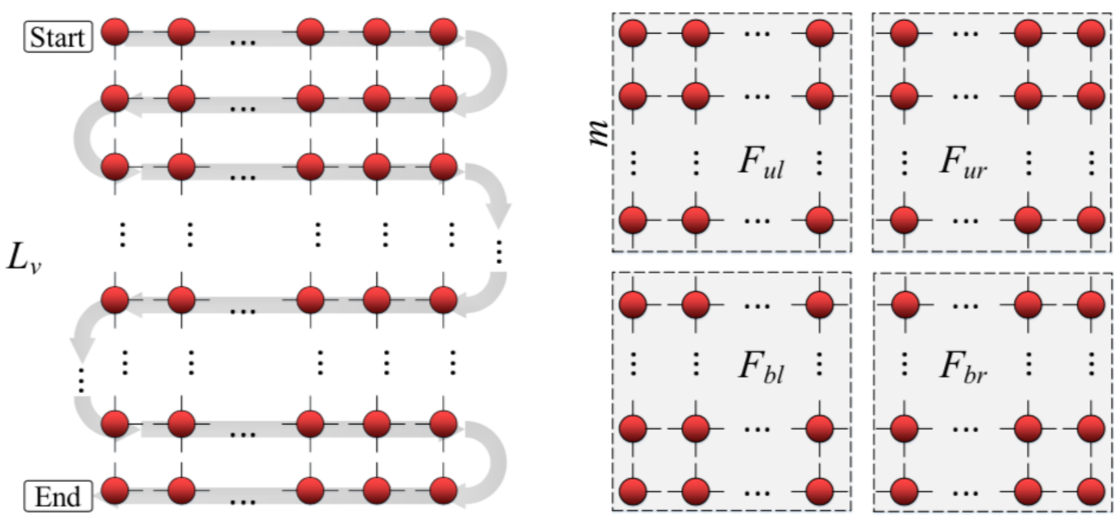
Boixo et al, arXiv:1712.05384

Chen et al, arXiv:1805.01450

Guo et al, PRL 123, 190501 (2019)

Performance comparison

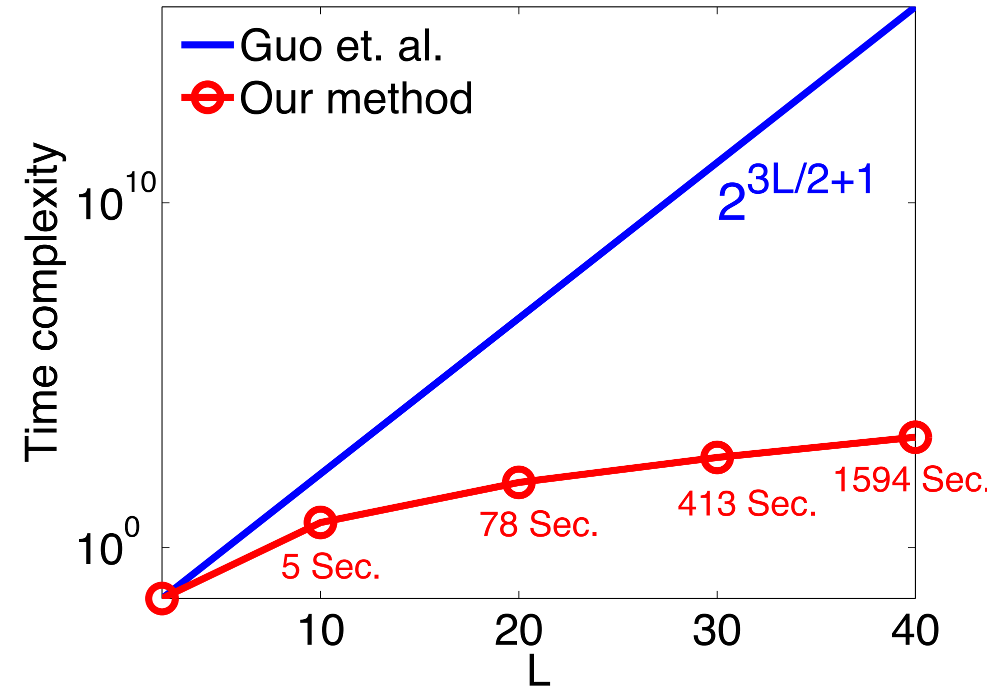


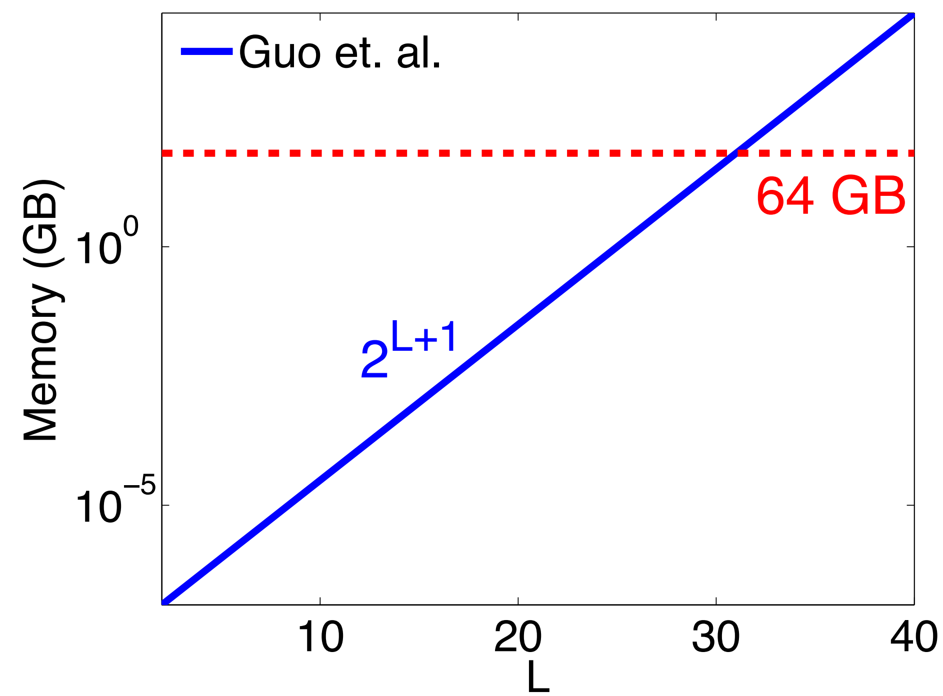


Guo et al, Physical Review Letter 123, 190501 (2019)

Time complexity: $2^{3L/2+1}$

Space complexity: 2^{L+1}



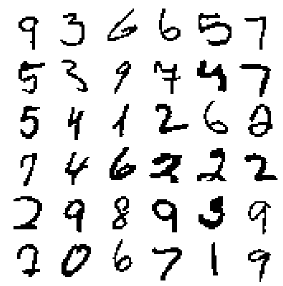


F. Pan, P. Zhou, S. Li , PZ, arXiv:1912.03014

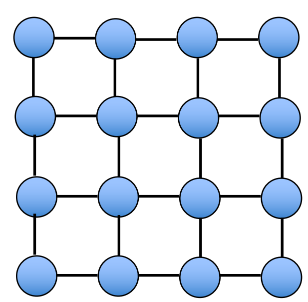
Random circuits are generated by **Jinguo Liu** using **Yao.jl**

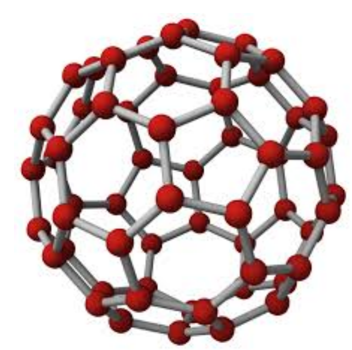
We hope more advanced algorithms could fully release the computational power of tensor networks in wide applications

- Machine learning: representing data distribution

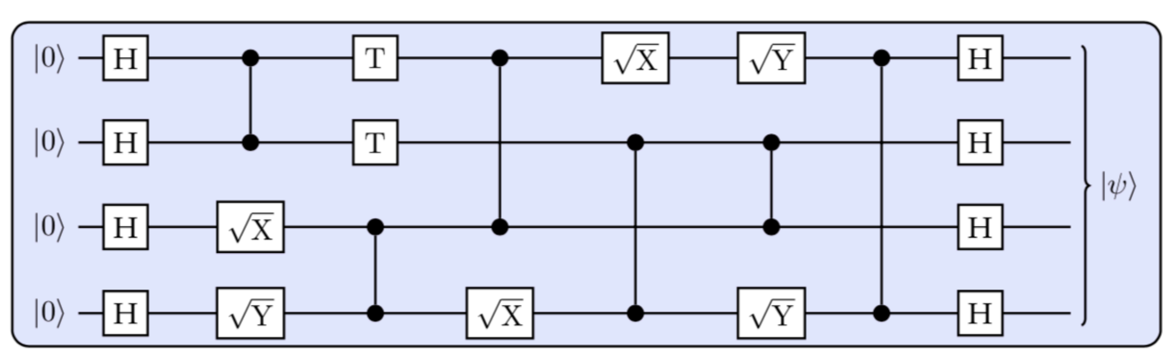


- Graphical model: representing a Boltzmann distribution





- Quantum circuit simulations



Thanks for your attention