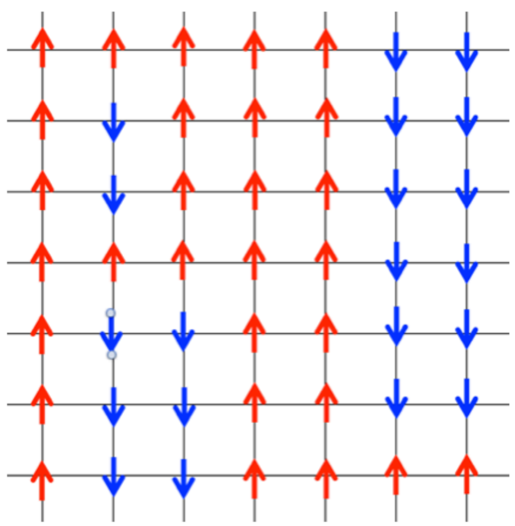
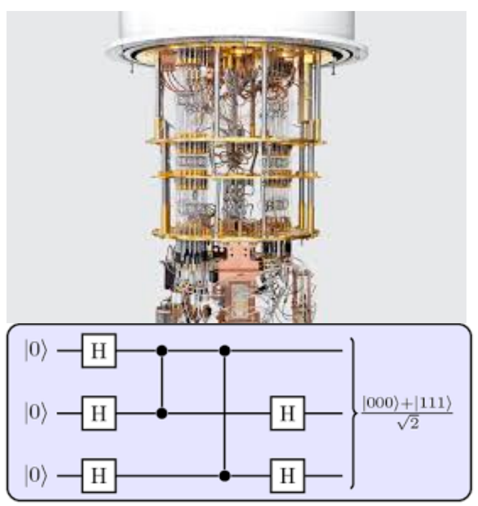
Computation with Tensor Networks





In collaboration with:
Feng Pan, Pengfei Zhou, Sujie Li @ ITP,CAS
Lei Wang @ IOP,CAS
Jinguo Liu @ Harvard

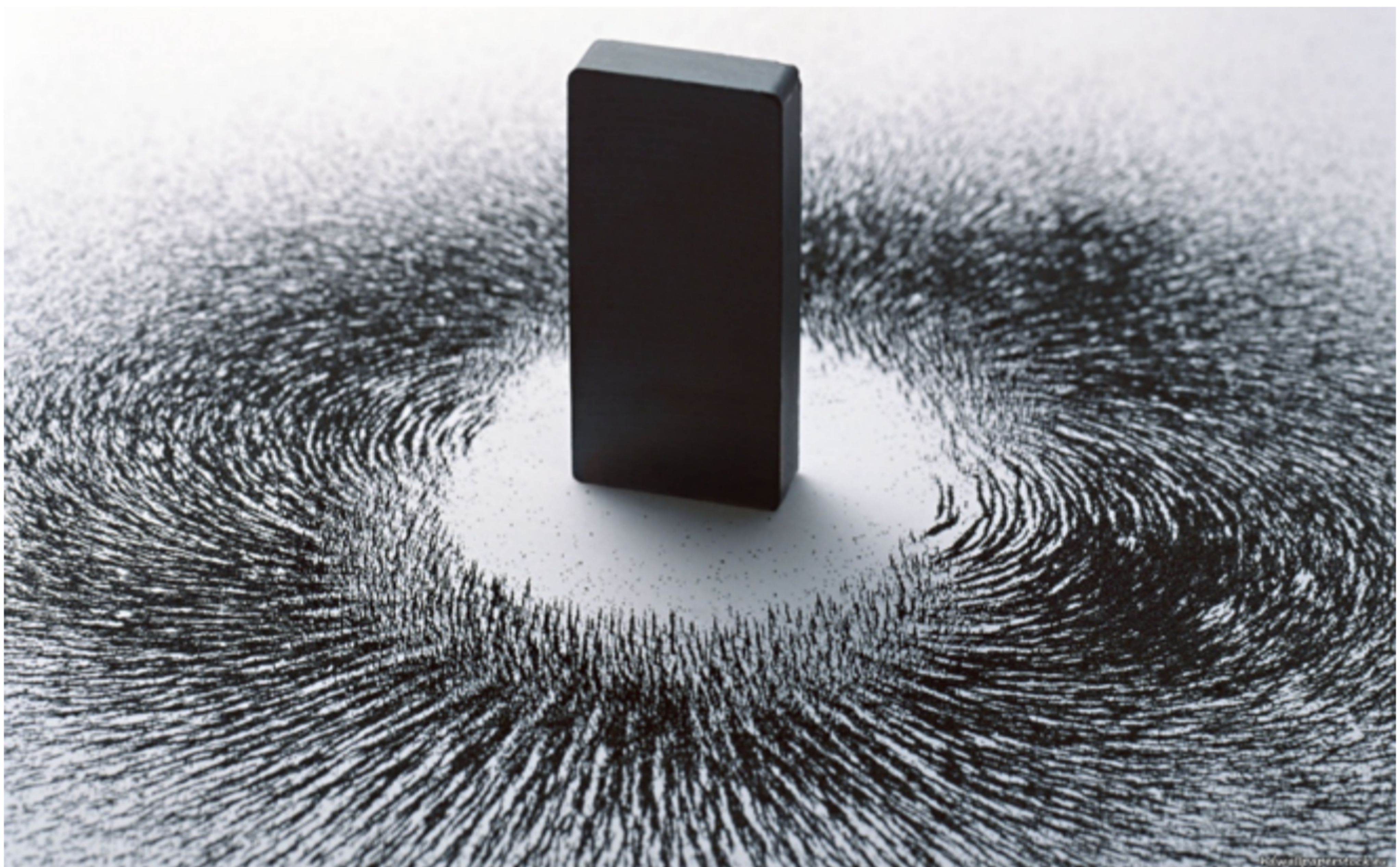
Pan Zhang
ITP,CAS

Ellis-ESA Workshop
2021.05.27

Statistical Mechanics

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

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

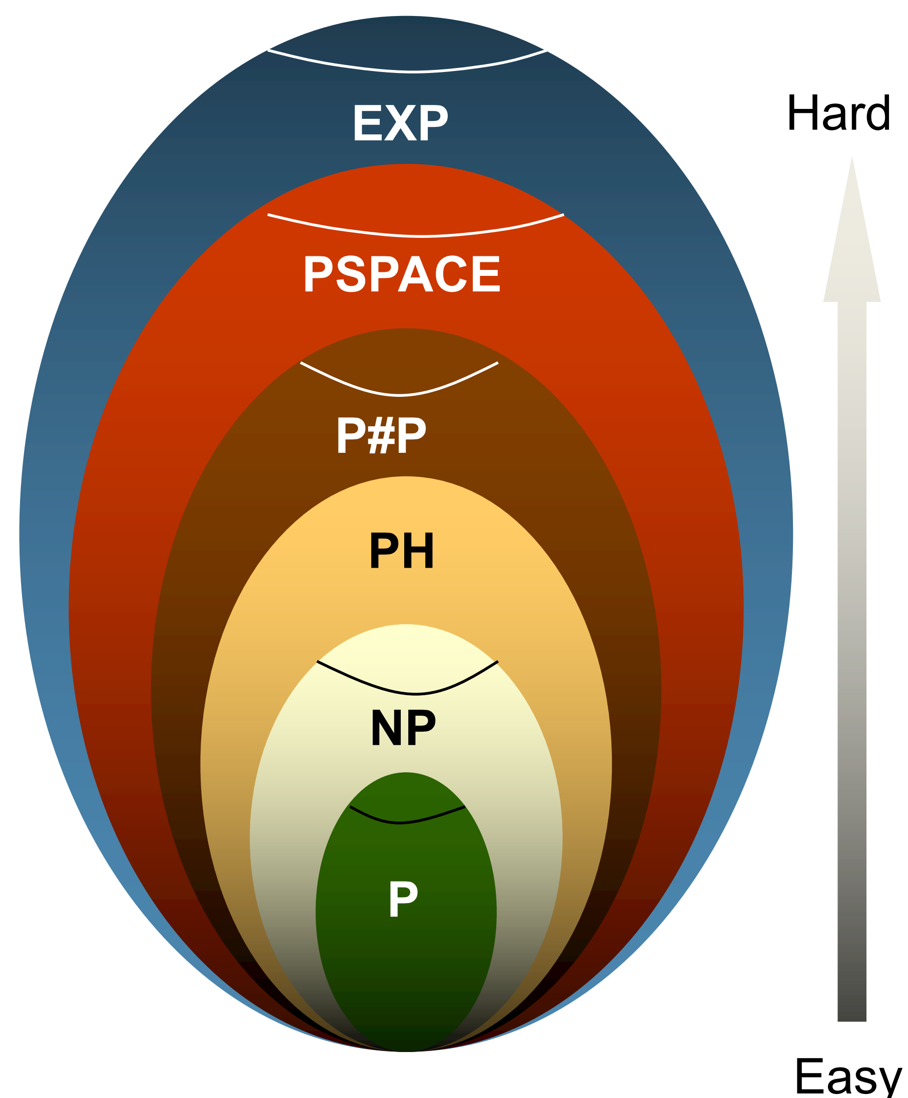


Statistical Mechanics

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

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

- Estimating Free Energy
- Computing statistics
- Unbiased sampling

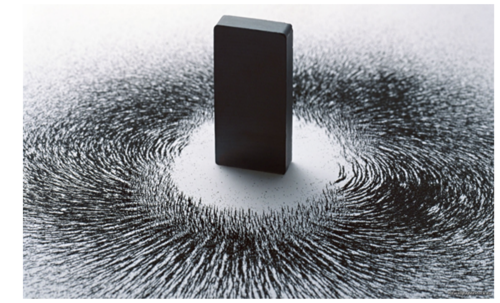


Applications of Statistical Mechanics

- Physics:

  Thermodynamics,

  Phase transitions …



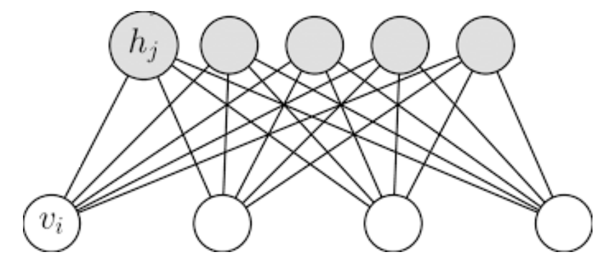
- Combinatorial Optimization

$$P(\mathbf{S}) = \lim_{\beta \to \infty} \frac{1}{Z} e^{-\beta E(\mathbf{S})}$$

- Machine Learning

  Hopfield model,

  Boltzmann machines

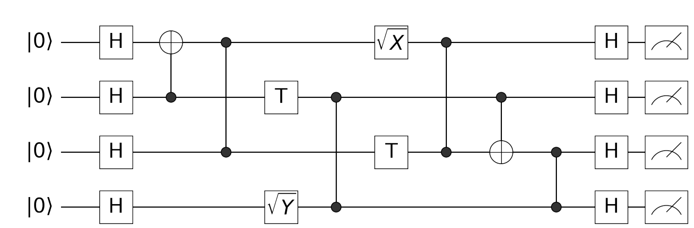


- Statistical Inference

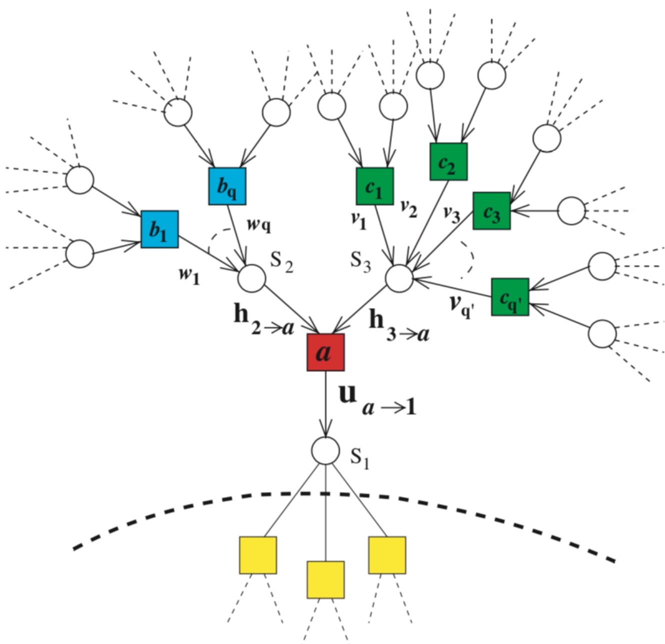
  Bayesian Inference, M.A.P.

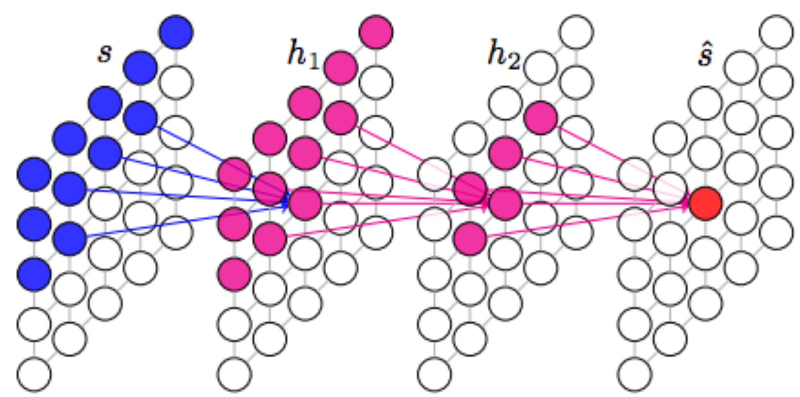
- Quantum computation

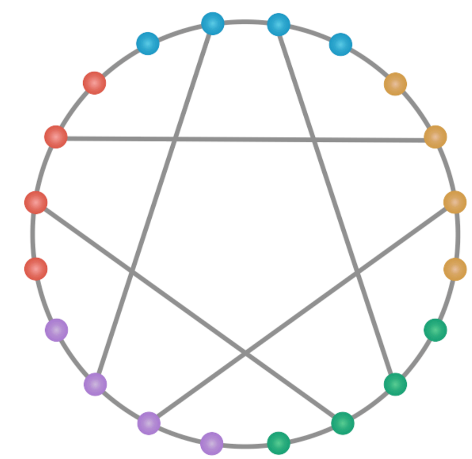
  Stat. Mech. with complexity interactions



My research journey on Statistical Mechanics:
From Mean-Field to Neural Networks Then to Tensor Networks







Mean-field

Variational Mean-field
Belief propagation
Replica symmetry breaking
Survey Propagation
Expectation Propagation

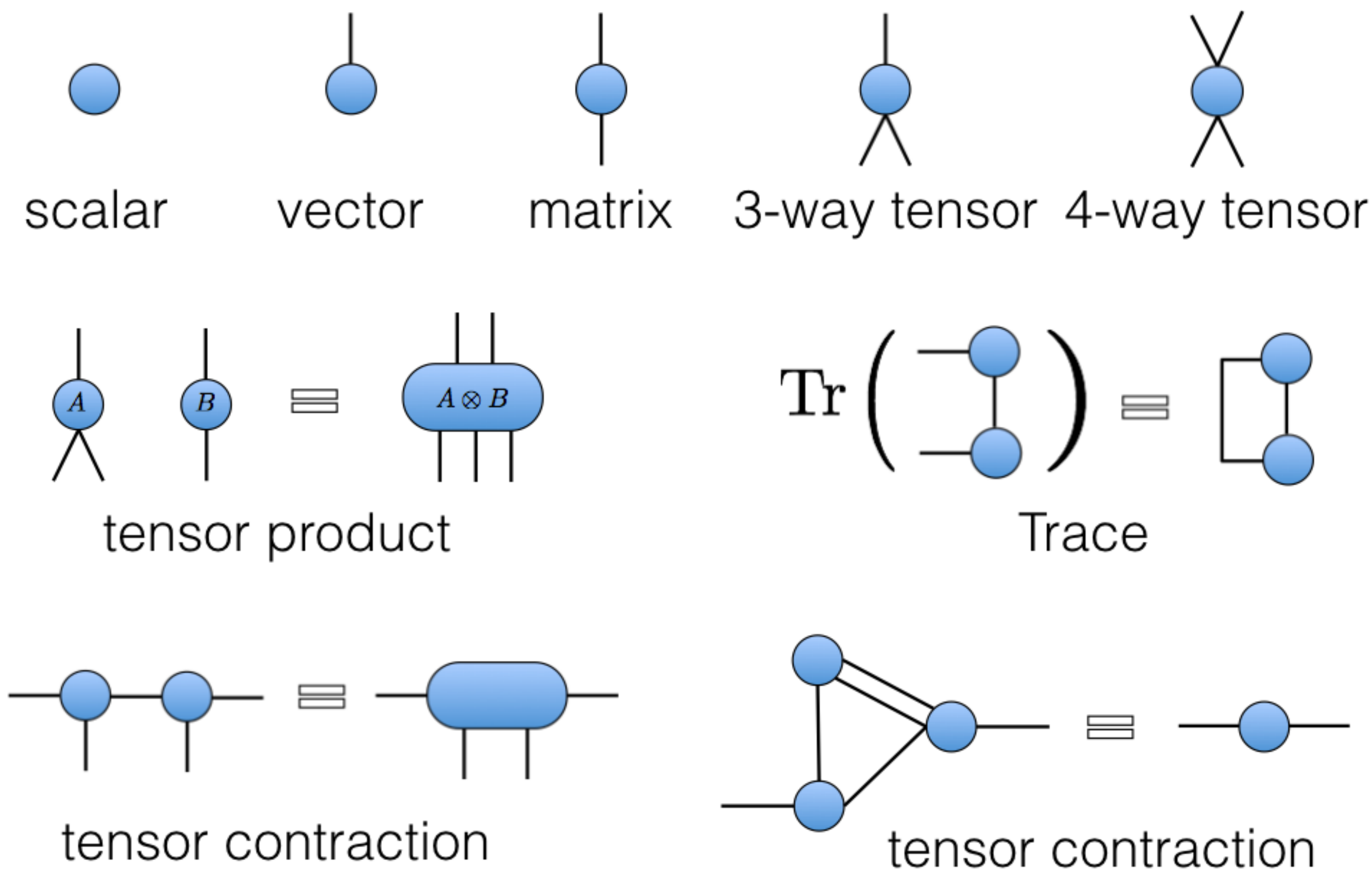
Neural Networks

Variational Autoregerssive Networks
Feedback-set VAN

Tensor Networks

CATN
Tropical Tensor Networks

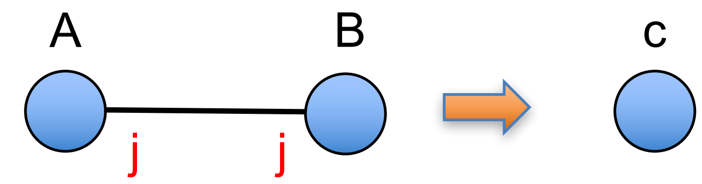
Tensor Network Diagram notations



Einsum notations of tensor network contractions

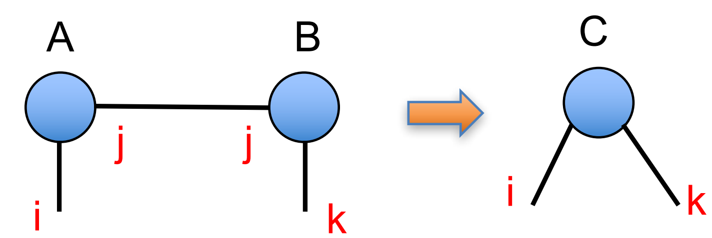
c = einsum(A,B,”j,j”)

$$c = A \cdot B$$



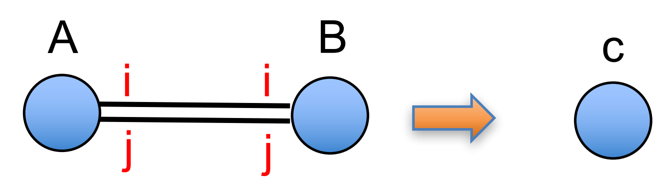
C = einsum(A,B,”ij,jk->ik”)

$$C = AB$$



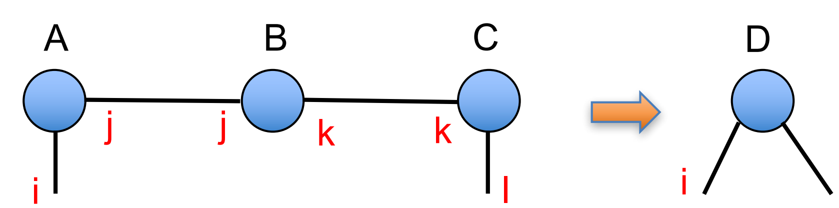
c = einsum(A,B,”ij,ij”)

$$C = \mathrm{Trace}(AB)$$



D = einsum(A,B,C,”ij,jk,kl”)

$$D = ABC$$



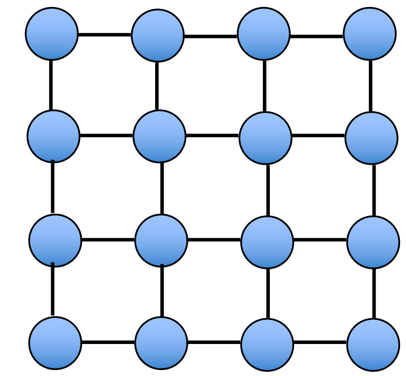
Space complexity: the dimension of the largest tensor

Time complexity: product of dimensions of all unique indices

TN contraction for computing the partition function

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

↑ ↑ ↑ ↓ ↑ ↓ ↓ ↓ ↓ ↓ ↑ ↑



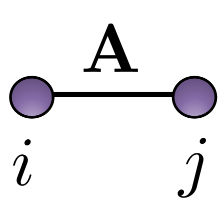
Computing normalization of a discrete probability distribution

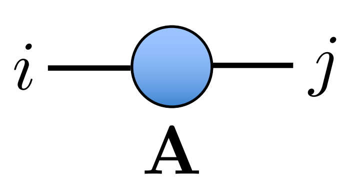
$$P(\mathbf{S}) = \frac{1}{Z}\widetilde{P} = \frac{1}{Z}e^{-\beta E(\mathbf{S})}$$

Any discrete probability distribution is a tensor,

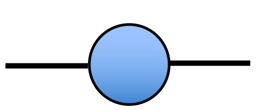
decomposed using tensor networks.

$$Z = \left\|\widetilde{P}\right\|_1 = \widetilde{P}\cdot \mathbf{1}_{2^n}^{\top} = \widetilde{P}\cdot \underbrace{\begin{pmatrix}1\\1\end{pmatrix}\otimes\begin{pmatrix}1\\1\end{pmatrix}\otimes\cdots\otimes\begin{pmatrix}1\\1\end{pmatrix}}_{n},$$



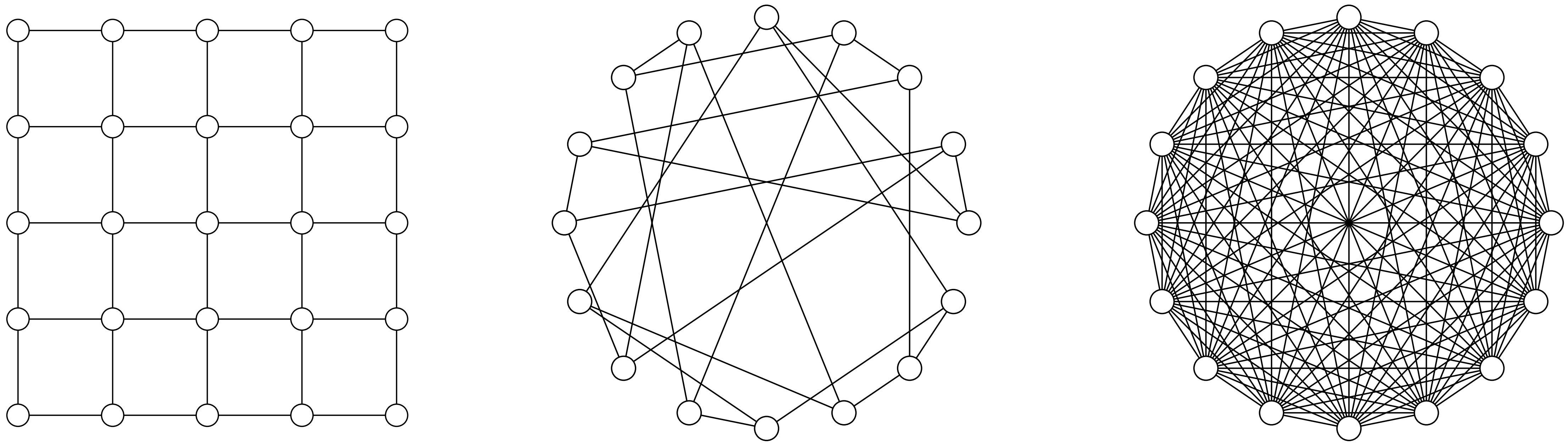


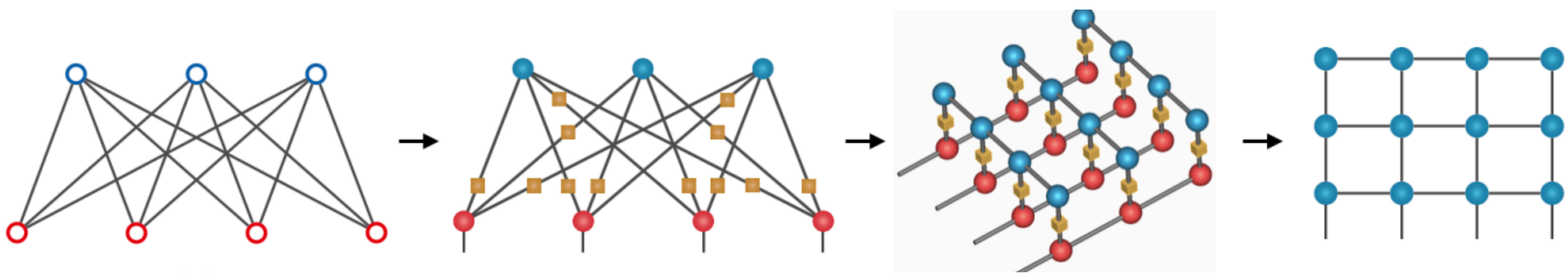
$$Z = \sum_{i,j} A_{ij} = \mathbf{1}^T A \mathbf{1} = [1,1]$$



$$\begin{bmatrix}1\\1\end{bmatrix}$$

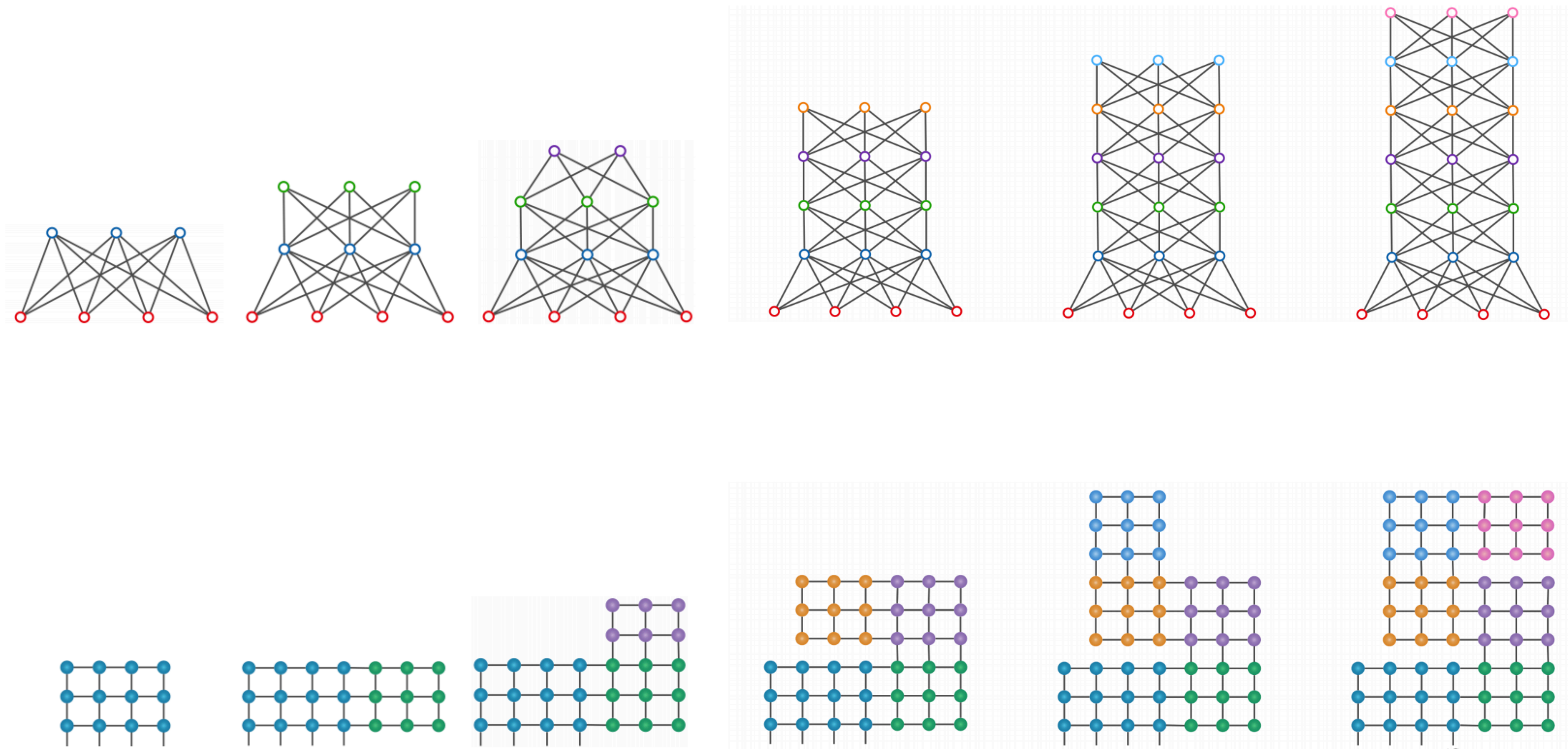
TN Contraction $\longrightarrow$ $Z(\beta)$





S. Li, P. Zhou, F. Pan, PZ, arXiv:2105.04130 (2021)

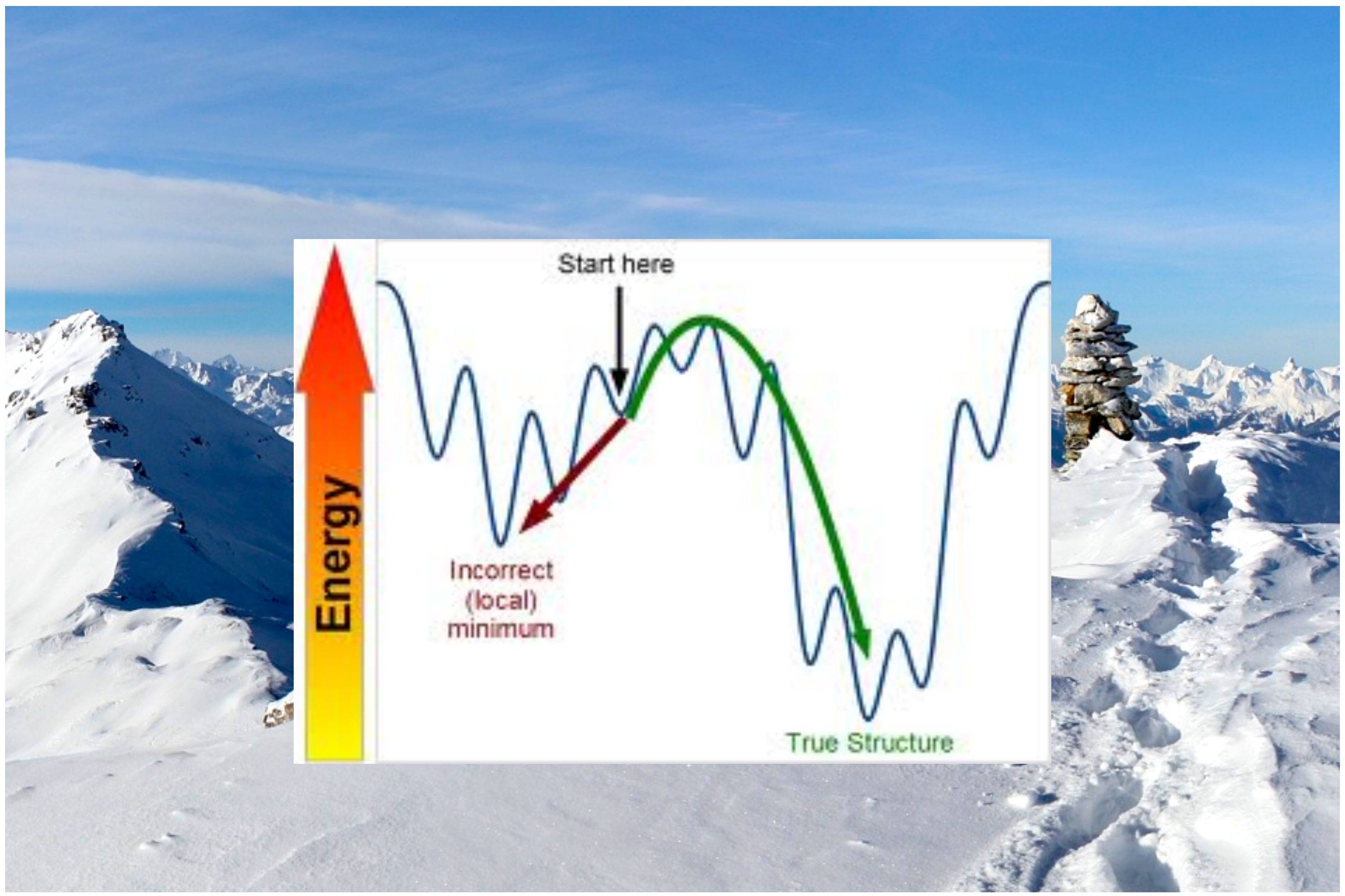
TN Contraction $\longrightarrow$ $Z(\beta)$

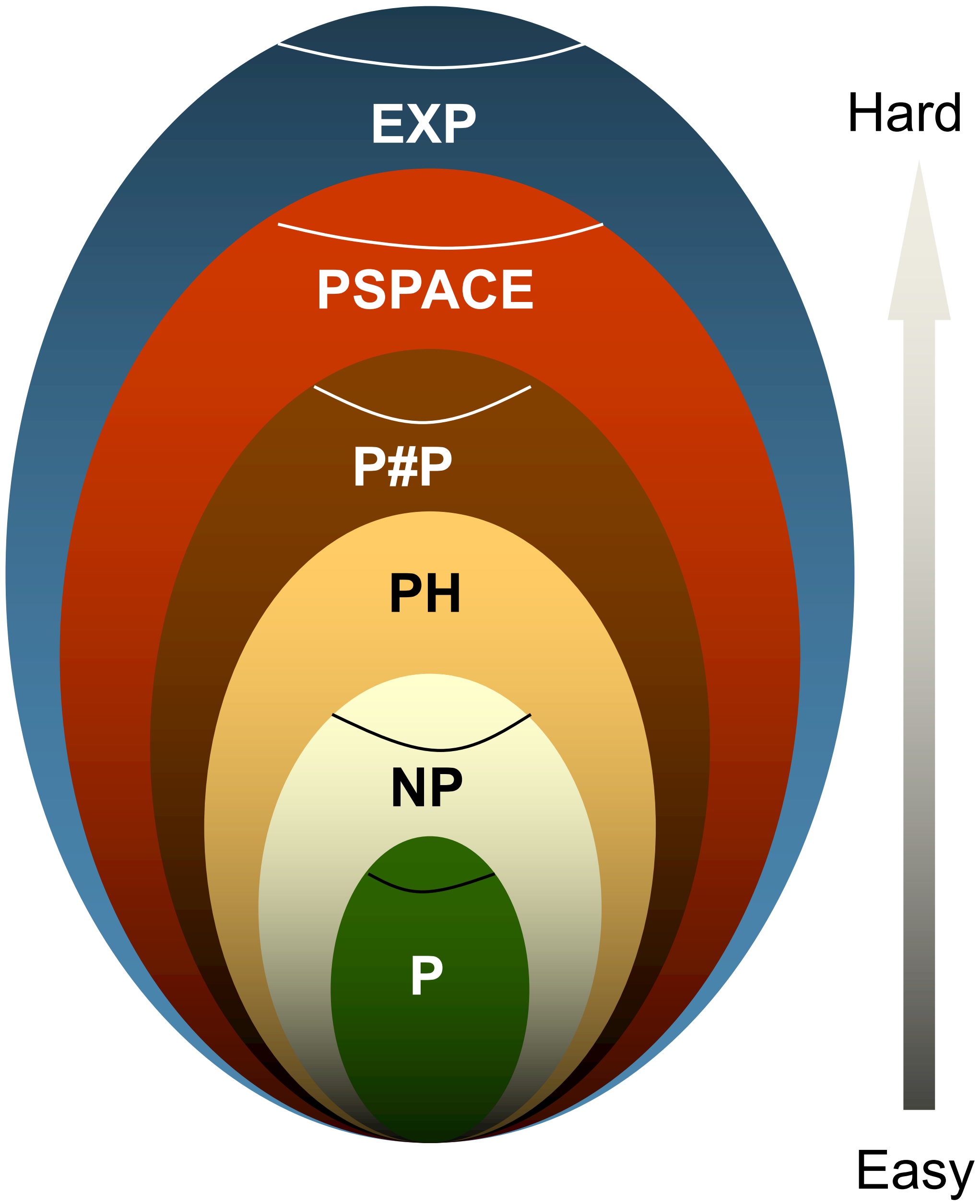


Deep Boltzmann Machines **2D Tensor Network**

S. Li, P. Zhou, F. Pan, PZ, arXiv:2105.04130 (2021)

Towards $T \to 0$: Ground States, combinatorial optimization





Tropical Tensor Network

Replace $(\times, +)$ **operations in linear algebra with** $(\otimes, \oplus)$

$$x \otimes y = x + y \qquad\qquad x \oplus y = \min(x, y)$$

$$F_0 = \lim_{\beta\to\infty} -\frac{1}{\beta} \ln \sum_{\mathbf{s}} e^{-\beta E(\mathbf{s})}$$

$$= \lim_{\beta\to\infty} \left[E_{\min} - \frac{1}{\beta} S(E_{\min}) \right]$$

$$\lim_{\beta\to\infty} -\frac{1}{\beta} \ln(e^{-\beta x} \times e^{-\beta y}) = x \otimes y$$

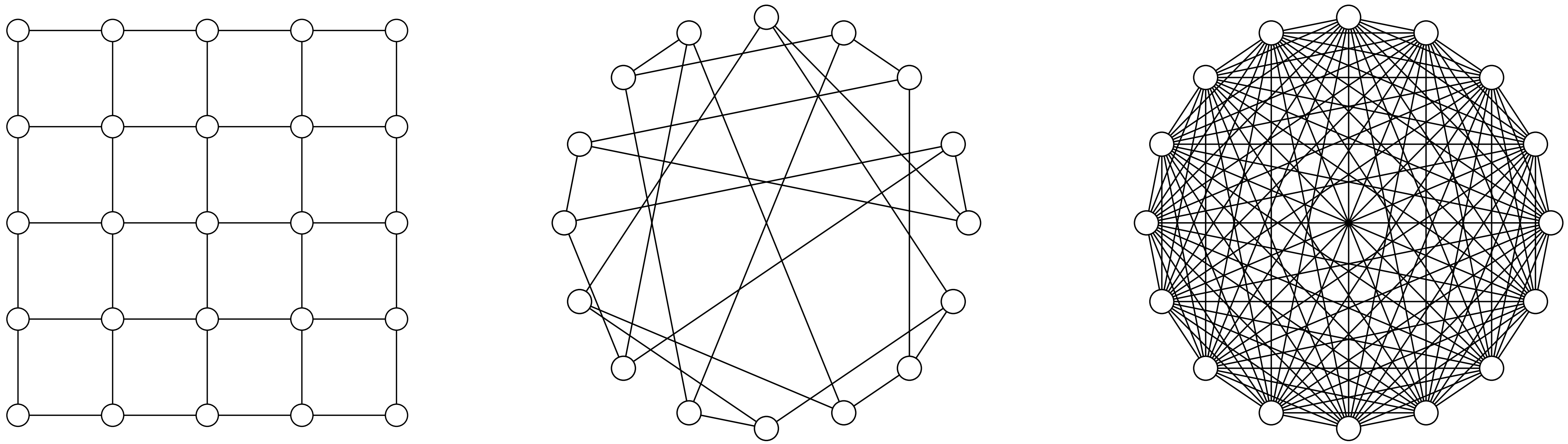
$$\lim_{\beta\to\infty} -\frac{1}{\beta} \ln(e^{-\beta x} + e^{-\beta y}) = x \oplus y$$

ground state energy
$\longrightarrow$ Tropical tensor network contraction

ground state configuration
$\longrightarrow$ Gradient with respect to the field

ground state degeneracy
$\longrightarrow$ Mix tropical with ordinary algebra

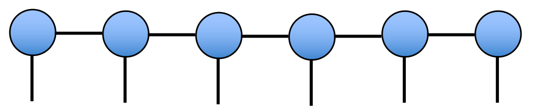
J. Liu, L. Wang, PZ, PRL 126, 090506 (2021)



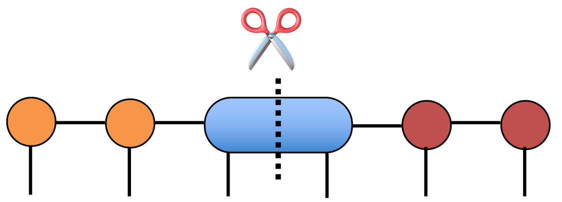
For large TNs: we need approximate contractions

Challenges:

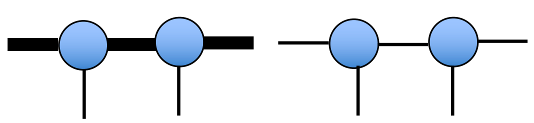
1. Exponential space complexity **Compressed using MPS**



2. Global approximations **Canonical form, DMRG**



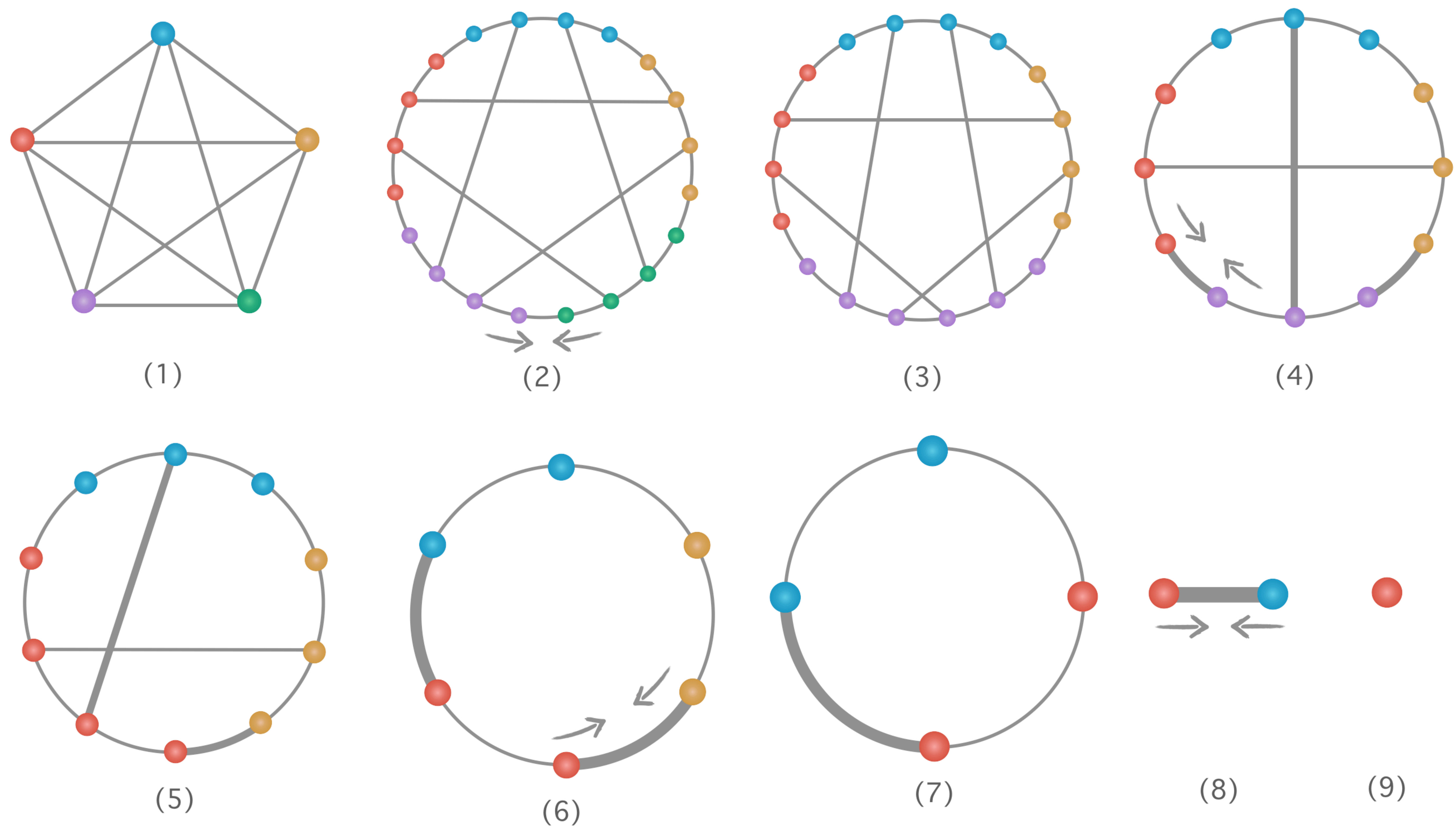
3. Scalability **Tunable bond dimension**



→ **CATN**

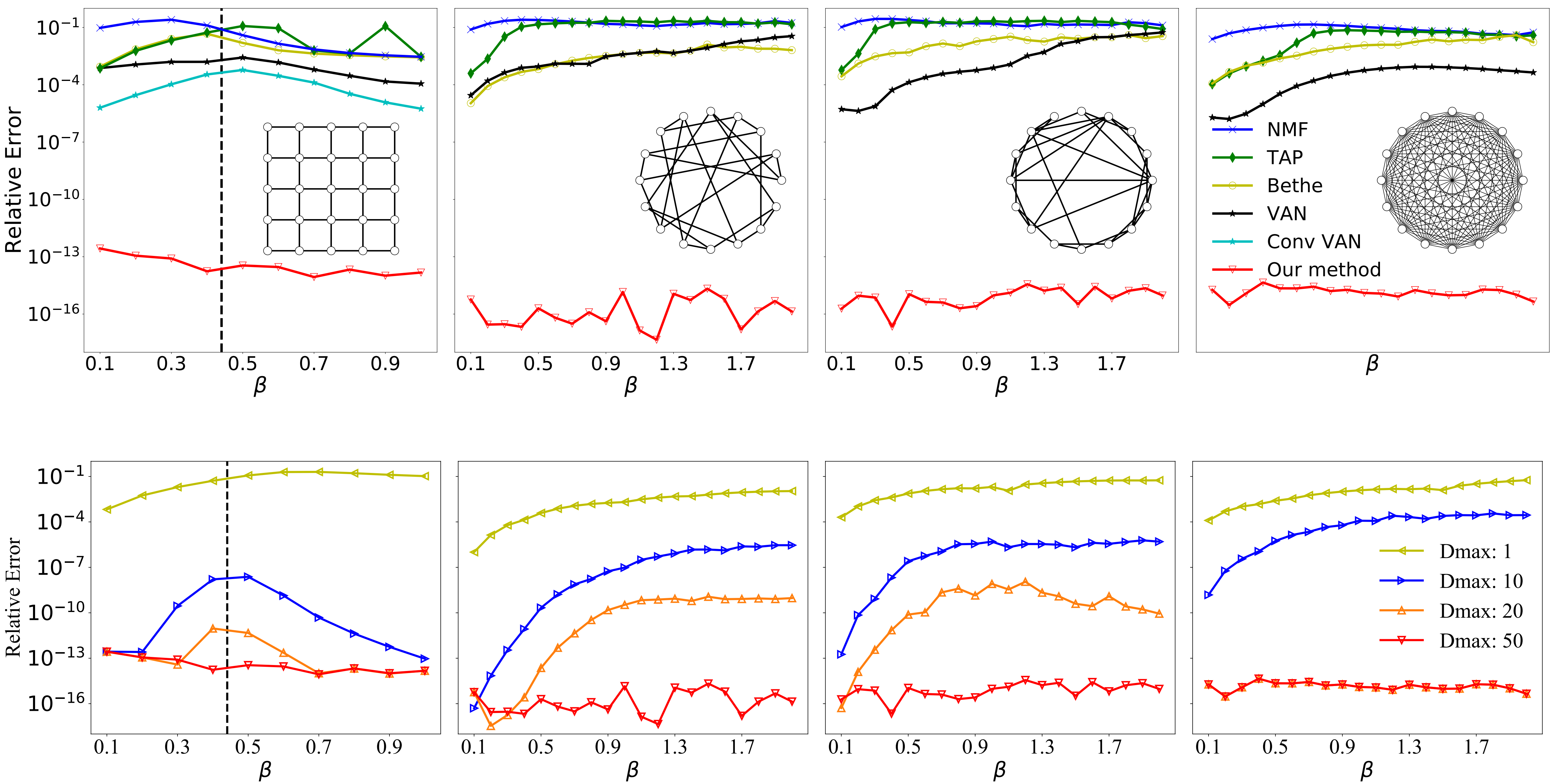
F. Pan, P. Zhou, S. Li , PZ, *Phys. Rev. Lett.* 125,060503 (2020)

CATN: Contracting Arbitrary Tensor Networks



F. Pan, P. Zhou, S. Li , PZ, *Phys. Rev. Lett.* 125,060503 (2020)

Computing free energy of spin glasses



16x16 2D lattice

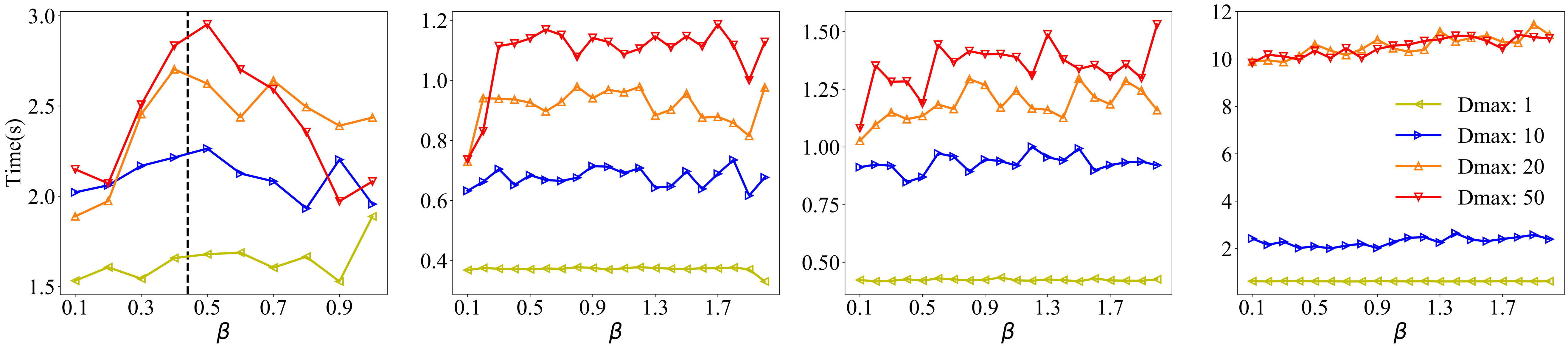
Regular Random graphs n=80, k=3

Small-world n=70, c=4

SK model n=20

F. Pan, P. Zhou, S. Li , PZ, *Phy. Rev. Lett.* 125,060503 (2020)

Time used



16x16 2D lattice

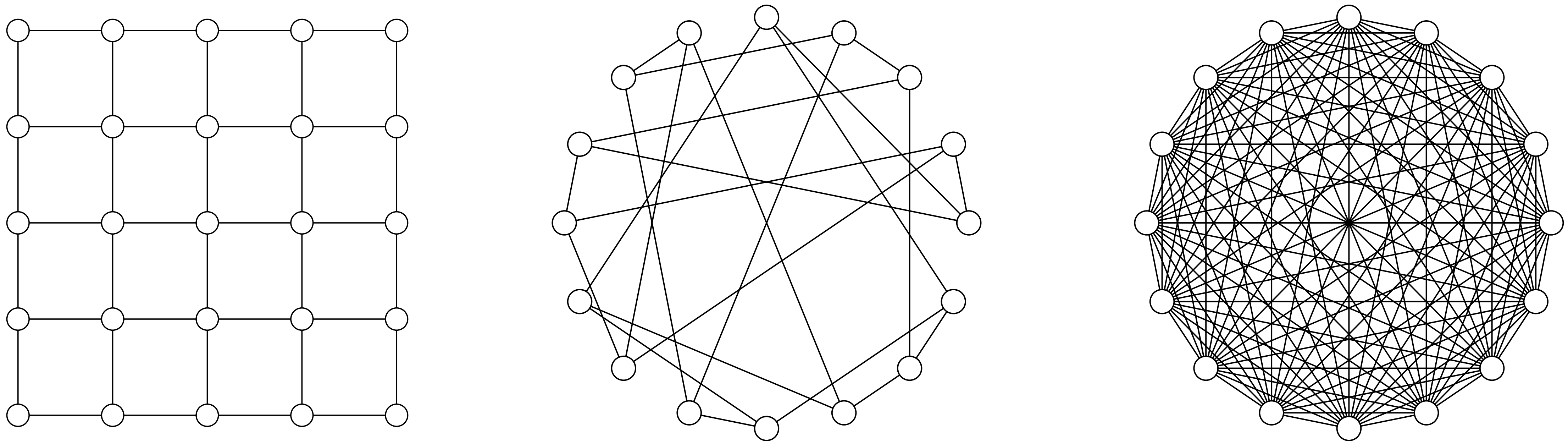
Regular Random graphs
n=80, k=3

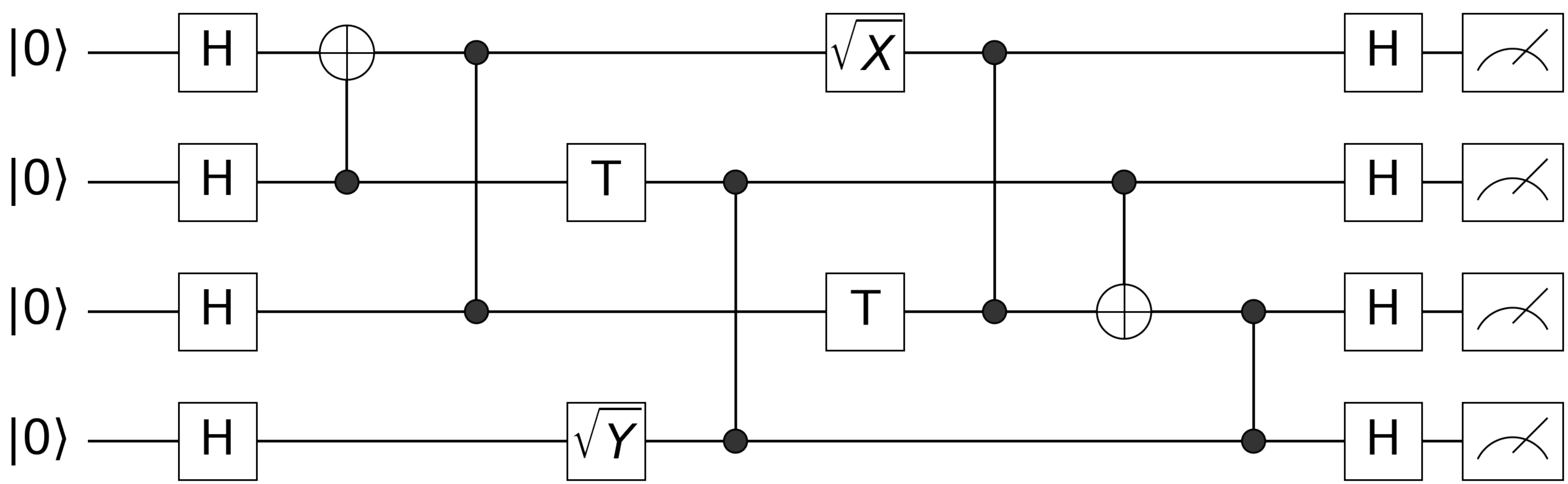
Small-world
n=70, c=4

SK model
n=20

F. Pan, P. Zhou, S. Li , PZ, *Phys. Rev. Lett.* 125,060503 (2020)

From partition function to quantum circuit simulation

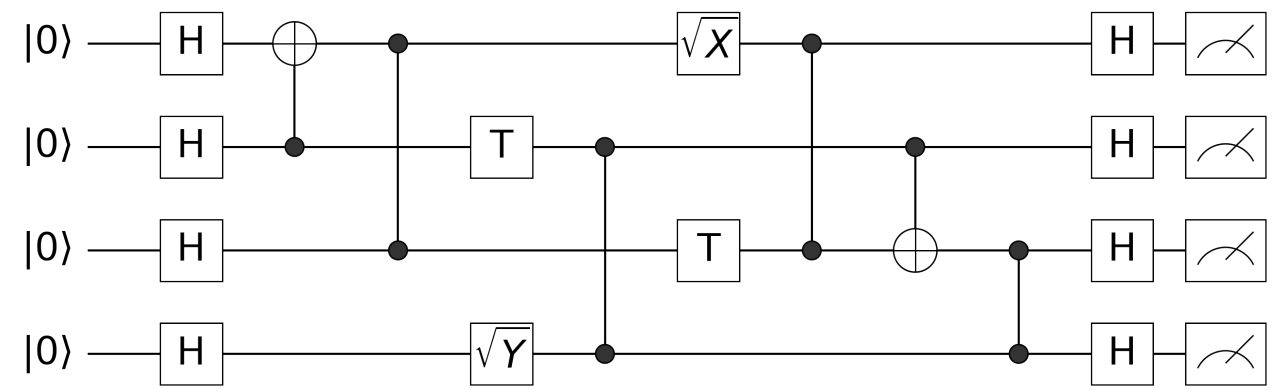




Simulation methods

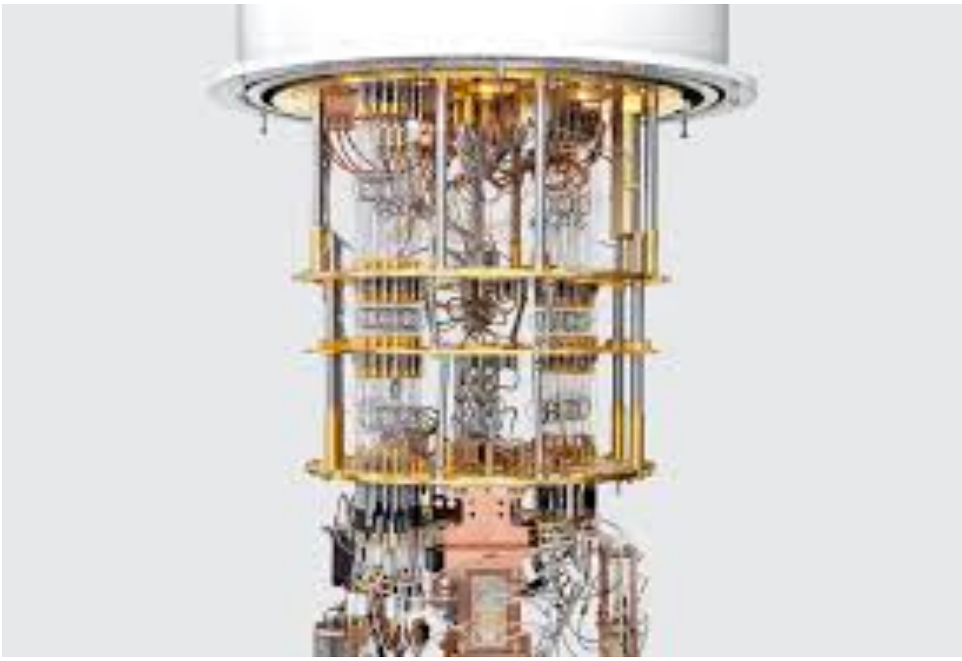
Full amplitude:

- Storing full state-vector [**Yao.jl**, Qiskit, Qulacs, Cirq...]
- Schrödinger-Feynmann
- MPS / Group MPS

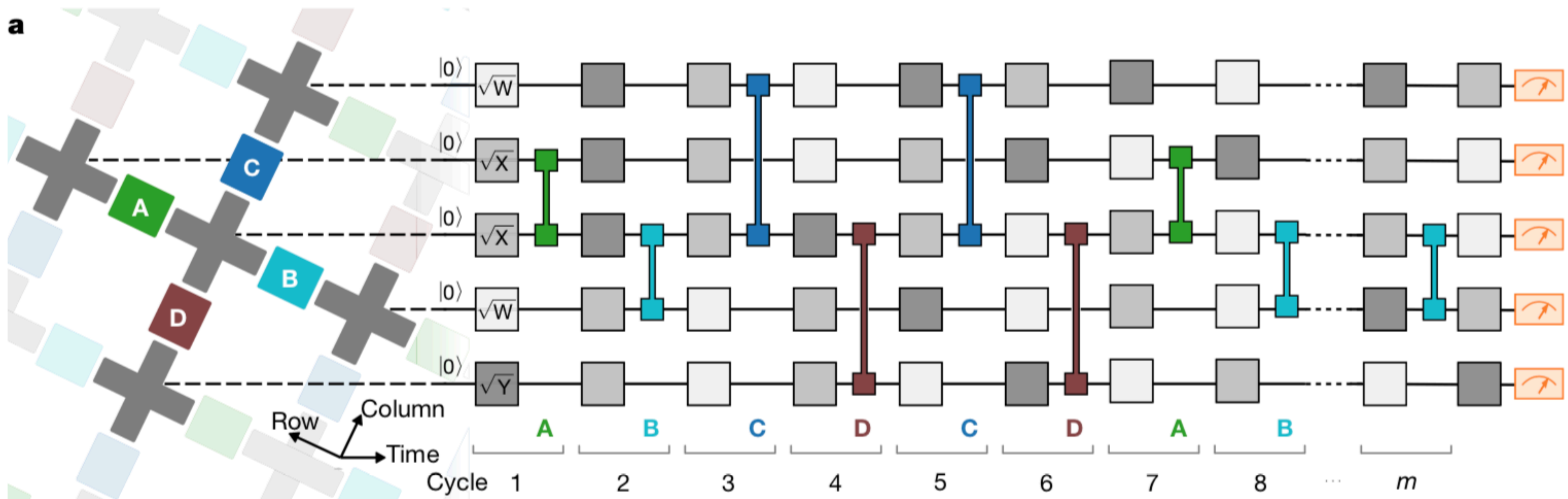


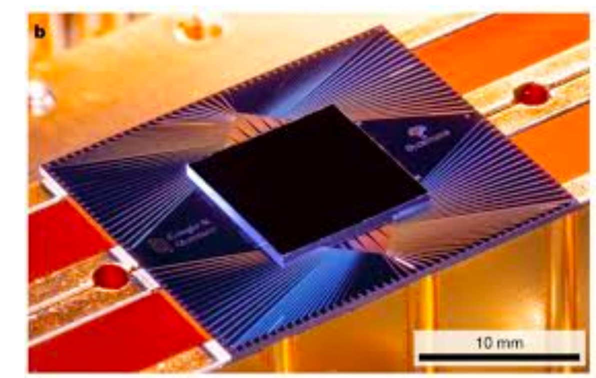
Single/batch amplitude:

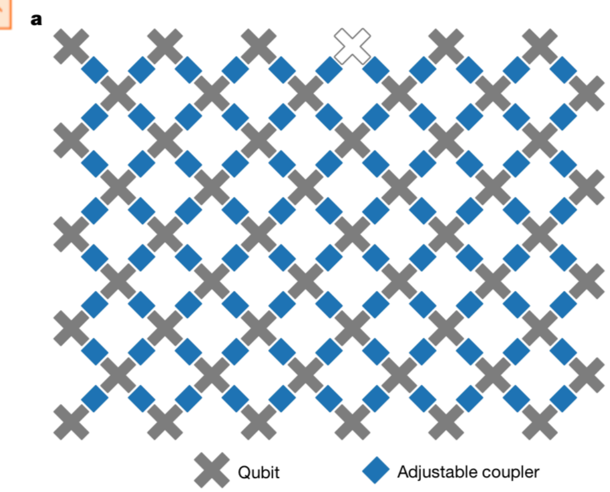
- PEPS based (single amplitude)
- Cotengra (single amplitude)
- Alibaba ACQDP (64-amplitude batch)
- Our method (big-batch)



Google’s Quantum Supremacy experiments





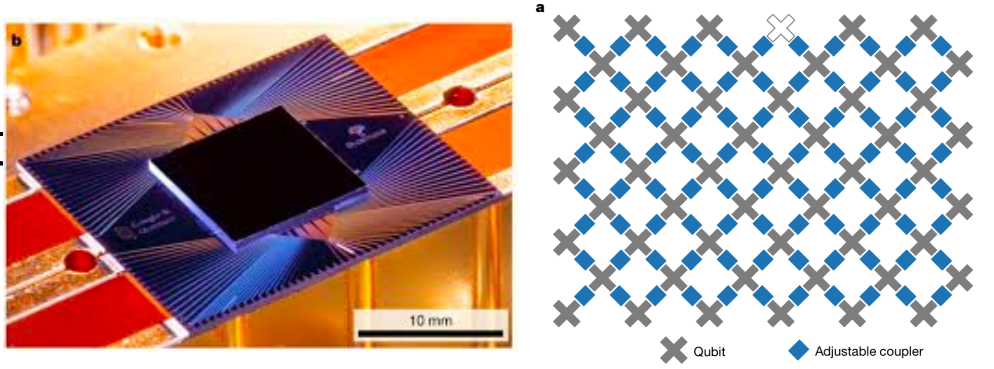


Arute et al, Nature 574 505 (2019)

- 53 qubits, 20 cycles of unitary operations
- 1 million samples within 200 seconds
- Linear Cross Entropy Fidelity (XEB) = 0.002, given by extrapolations
- Google’s (Shrödinger-Feynmann) classic algorithm requires 10,000 years on Summit

Classical simulation of Sycamore

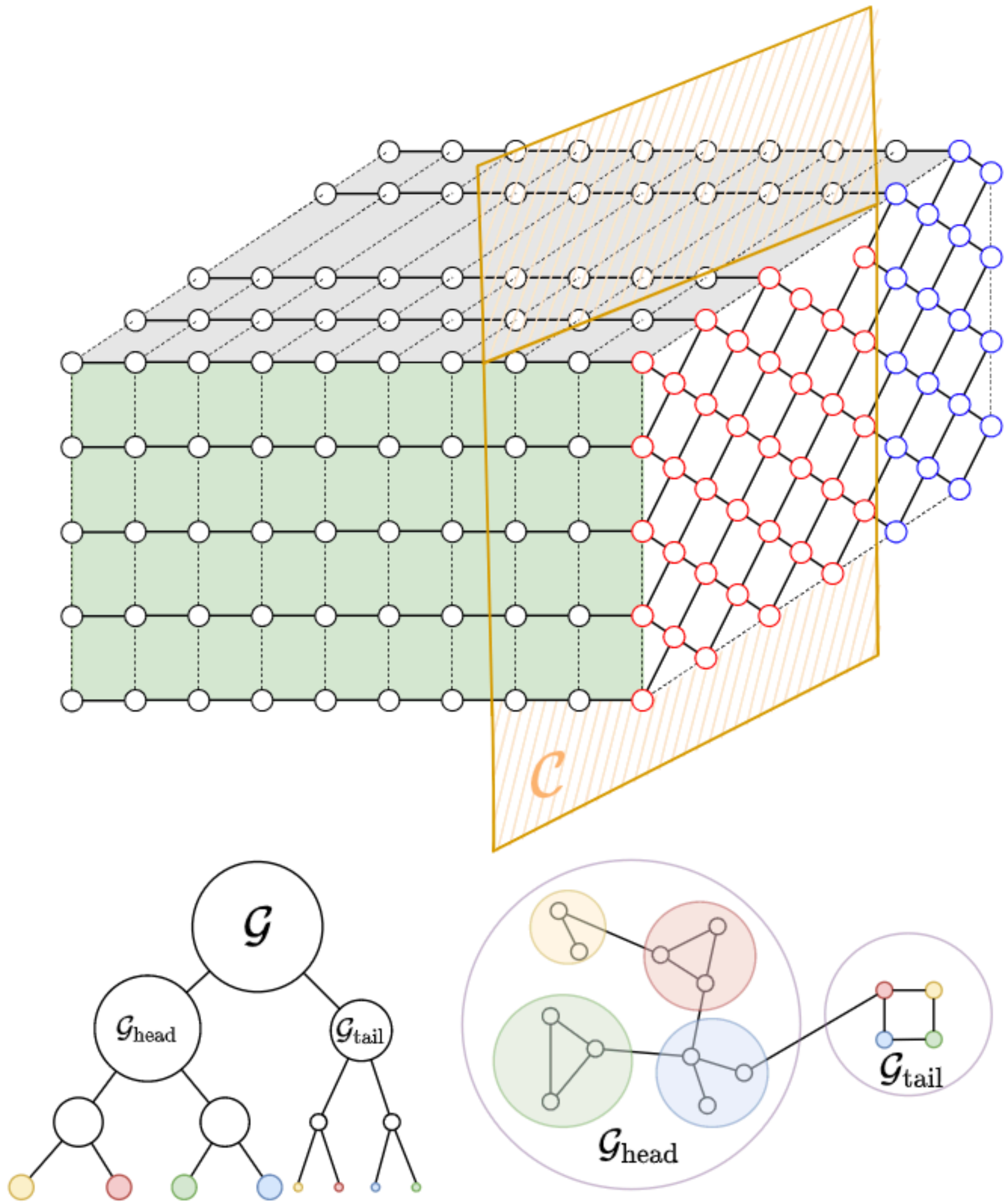
- Google's original estimate [**Arute et. al. 2019**]
  - 10,000 years for simulating the Sycamore circuit with 53 qubits and 20 cycles (using Summit)

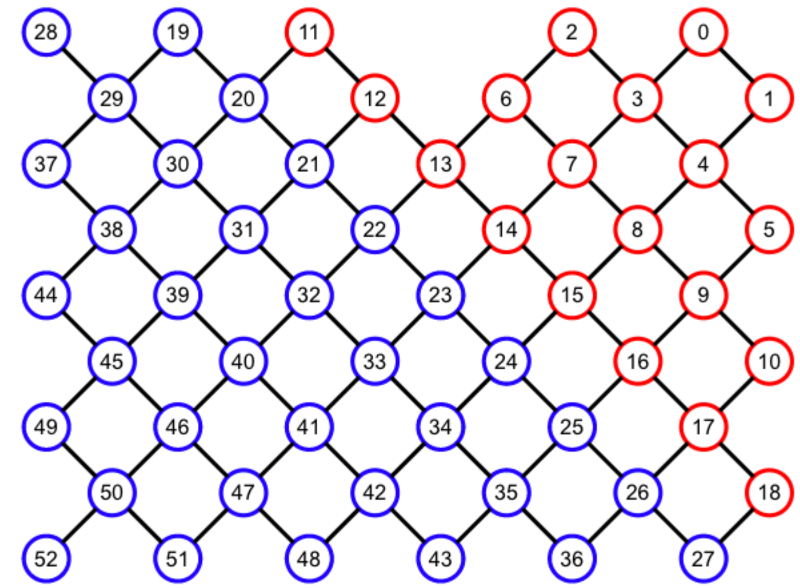


Images from Arute et. al. Nature 2019

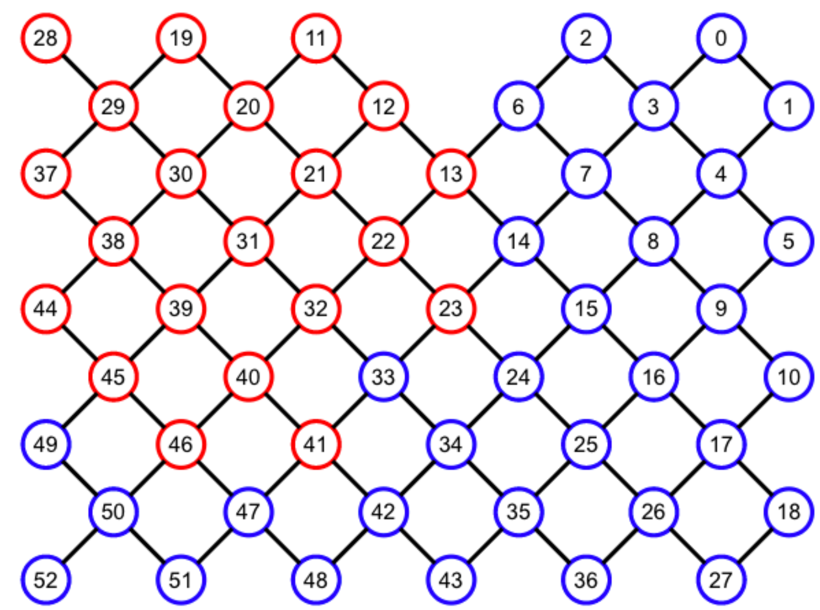
- IBM's estimate [**Pednault et al 2019**]:
  - 2.5 days (with 250PB memory, all memory and hard disks of Summit)
- Cotengra [**Gray/Kourtis 2020**]
  - Balanced Partitioning (`Kahypar`) + Greedy/optimal + Slicing
  - 3000 years for single amplitude (Single GPU)
- Alibaba's simulator [**Huang et. al. 2020**]
  - Hierarchical partitioning (`Kahypar`) + Greedy/optimal + Slicing + sampling
  - 20 days for 1 million samples (with Summit-compatible supercomputer)

Our approach: Big-Head tensor network method





14 cycles



20 cycles

arXiv:2013.03074

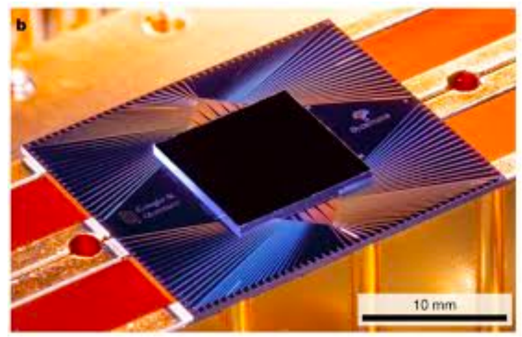
Computational complexity

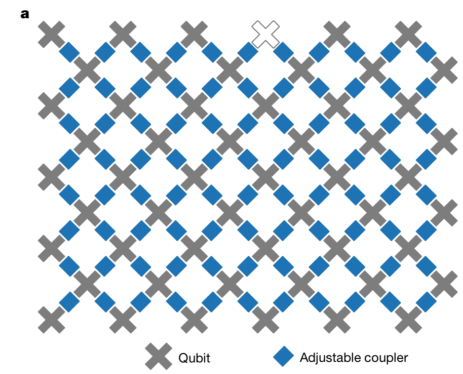
	# bitstrings	Time complexity	Space complexity	Computational time	Computational hardware
Google [1]	10^6	—	—	10,000 years	Summit supercomputer
Cotengra [12]	1	3.10×10^{22}	2^{27}	3,088 years	One NVIDIA Quadro P2000
Alibaba [18]	64	6.66×10^{18}	2^{29}	267 days	One V100 GPU
Ours	**2097152**	4.51×10^{18}	2^{30}	149 Days	One A100 GPU

- **The computational cost of our algorithm in obtaining 2 million amplitudes is smaller than obtaining 64 amplitudes using Alibaba's method.**
- **Google, Cotengra, and Alibaba's results are estimations**
- **We did the computations for the first time.**

arXiv:2013.03074

Simulating Sycamore with 53 qubits, 20 cycles





	Computation hardward	**Time**
Google [Arute et. al., 2019] (**Estiamte**)	Summit Super Computer	10,000 Years
IBM [Pednault et. al., 2019] (**Estimate**)	Summit Super Computer (all disks)	2.5 days
Alibaba [Huang et. al., 2020] (**Estimate**)	Summit Super Computer (compatible)	20 days
Ours [arXiv:2103.03074] (**Computation**)	60 GPUs	5 days

Main results

$$F_{\mathrm{XEB}} = \frac{2^n}{L}\sum_{i=1}^{L} P_U(\mathbf{s}_i) - 1$$

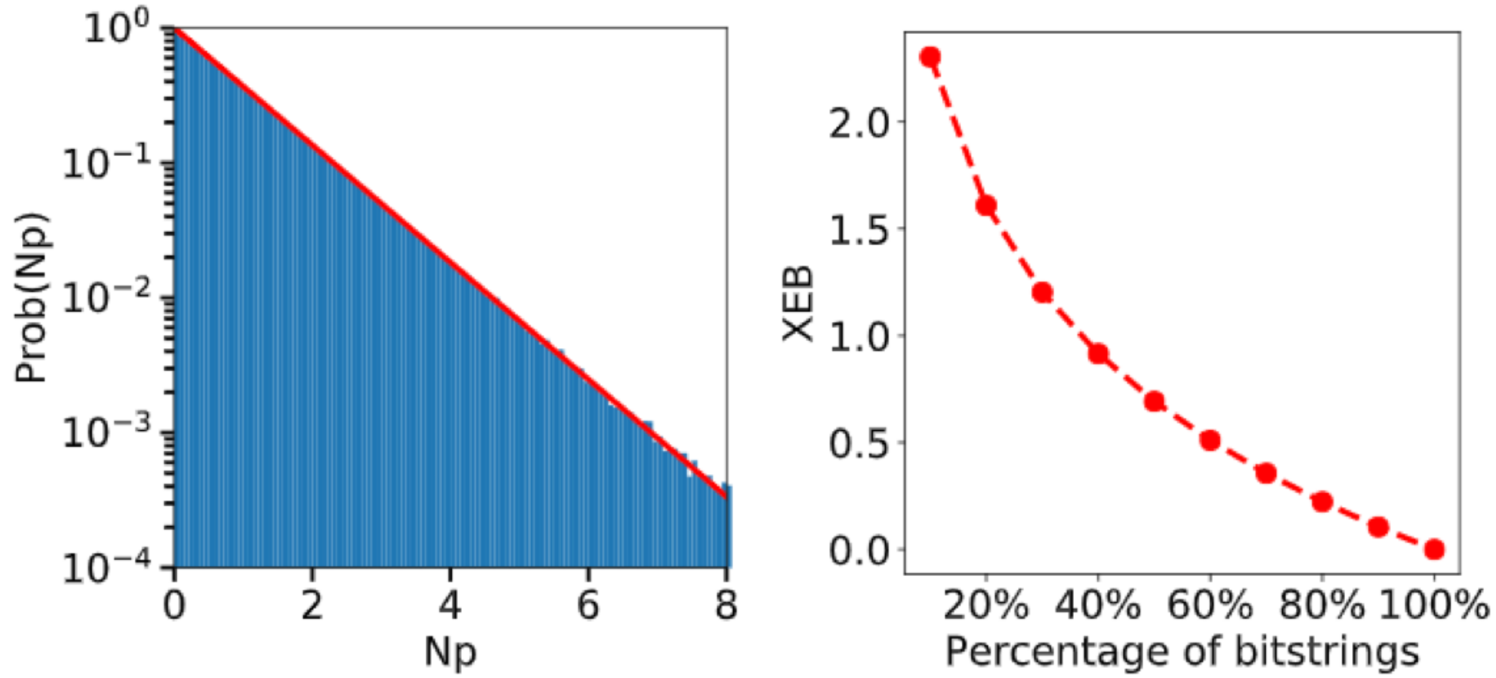


FIG. 2. (Left): Histogram of bitstring probabilities $P_U(\mathbf{s}) = P_U(\mathbf{s}_1;\mathbf{s}_2)$ for $l = 2^{21}$ bitstrings obtained from the Sycamore circuit with $n = 53$ qubits, $m = 20$ cycles, sequence ABCDCDAB, seed 0, and the assignment of partial bitstring $\mathbf{s}_1$ are fixed to $\underbrace{0,0,0,\cdots,0}_{32}$.

- **Obtained 2 million amplitudes and probabilities, following the Porter-Thomas distribution**
- **Sampled 1 million bitstrings, XEB fidelity=0.739, larger than Google's XEB**

arXiv:2013.03074

Discussions

Advantages over Google's hardware:

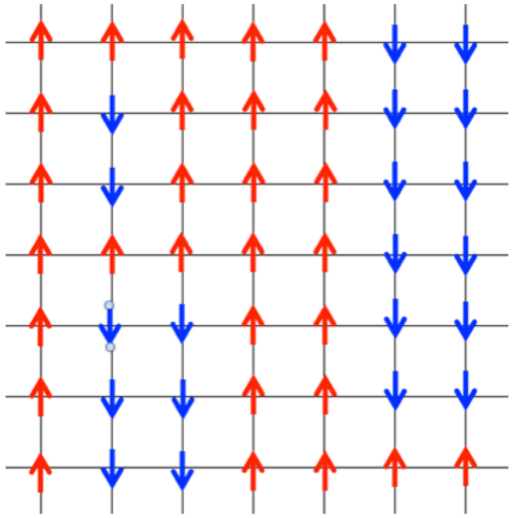
- Exact amplitude computation
- Pass the XEB test with an higher XEB fidelity
- Conditional probability/sampling

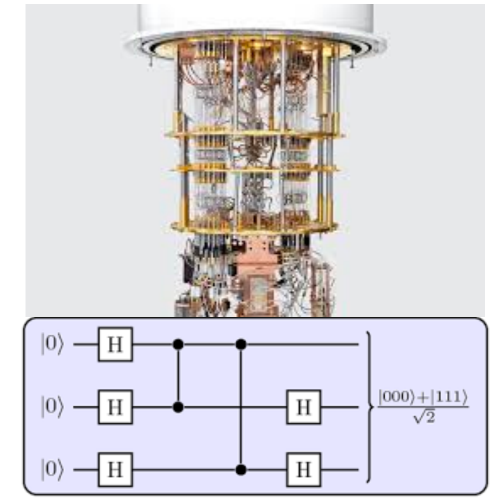
Disadvantages over Google's hardware

- Exponential algorithm, not scalable
- Bitstring Correlations
- XEB dependence

Using tensor networks as bridge, combining classical and quantum computations for solving challenge real-world problems

Computation with tensor networks





Joint probability distribution of microscopic configurations

$$P(\sigma) = \frac{1}{Z}\exp(-\beta E(\sigma))$$

Joint probability distribution of data variables

$$P(\mathrm{Data})$$

Control of quantum states

$$\psi(\sigma)$$

Exponential space

Effective models

Computational power

Tensor Network as a bridge !

Reference:

F. Pan, P. Zhou, S. Li , PZ, PRL 125, 060503 (2020)

J. Liu, L. Wang, PZ, PRL 126, 090506 (2021)

F. Pan, PZ, arXiv:2103.03074 (2021)

S. Li, P. Zhou, F. Pan, PZ, arXiv:2105.04130 (2021)