2023-07-24 Extreme Universe Collaboration - 18th Colloquium

Markov-Chain Monte Carlo in Tensor-Network Representation

Synge Todo / 藤堂眞治 <wistaria@phys.s.u-tokyo.ac.jp>
Department of Physics, University of Tokyo

Agenda

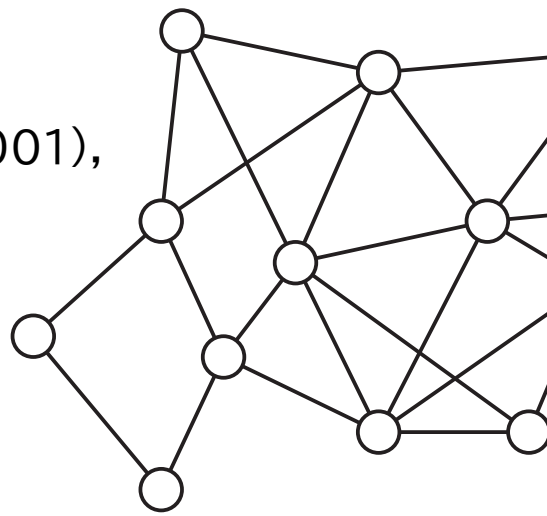
- Introduction
  - Markov-chain Monte Carlo and negative sign problem
  - Tensor network representation and approximate contraction
- Sampling approach in tensor network contraction
  - Projector formulation of tensor network method
  - Sampling projectors
- Monte Carlo in tensor network representation
  - Sequential Monte Carlo in tensor network representation
  - Markov-chain Monte Carlo in tensor network representation
  - Results
- Summary

Stochastic approach in computational physics

- "Our brains are just not wired to do probability problems very well"
  - Persi Diaconis
- Markov chain Monte Carlo - Metropolis et al (1953)
  - can sample from an arbitrary probability distribution
  - perfect sampling ("coupling from the past") - Propp and Wilson (1996)
    - samples perfectly independent samples from Markov chain
  - extended ensemble method - Hukushima and Nemoto (1996), Wang and Landau (2001)
    - realizes equilibrium immediately after quench
    - samples extremely rare events ($\sim 10^{-100}$)
  - $O(N)$ method for long-range interacting system - Fukui and Todo (2009)
    - exact $O(N)$ sampling (and energy measurement) instead of $O(N^2)$
  - MCMC without detailed balance - Suwa and Todo (2010), Michel et al (2014)
    - diffusive dynamics → ballistic dynamics

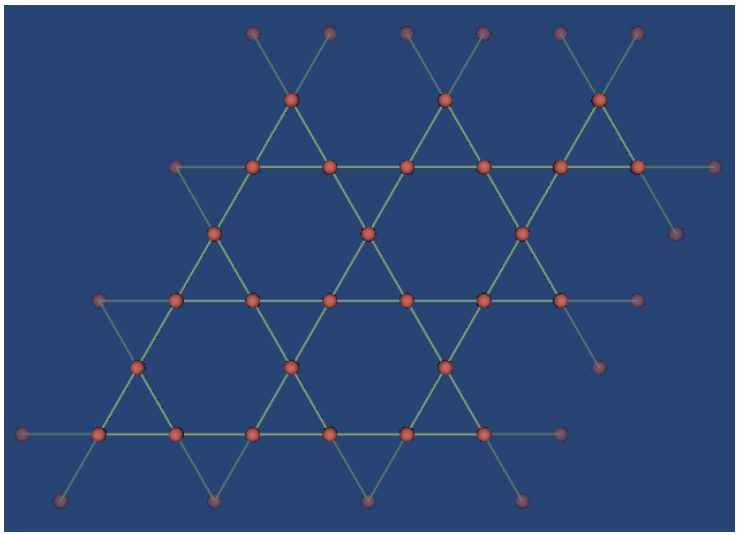
Advances in Markov chain Monte Carlo

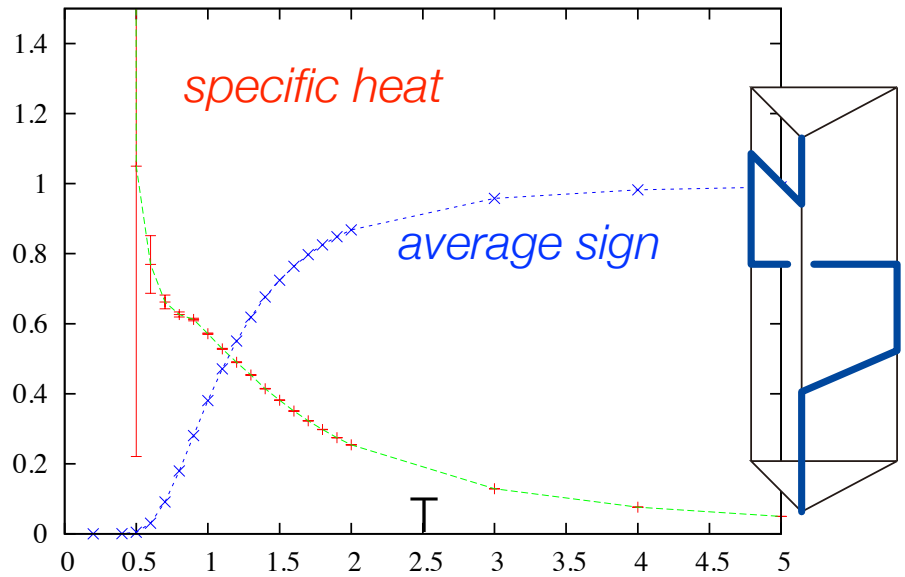
- Representation (definition of “configurations” and “weighs”)
  - path integral representation for quantum Monte Carlo (1976), Bayesian inference (1990)...
- Choice of ensemble
  - extended ensemble method: multicanonical MC (1991, 2001), exchange MC (1996), lifting (2000)...
- Generation of set of candidate configurations
  - non-local (cluster) updates: Swendsen-Wang (1987), Hamiltonian MC (1987), loop (1993), worm (1998)...
- Choice of transition kernel (probabilities)
  - Metropolis, heat bath (Gibbs sampler), over-relaxation (1987), irreversible kernel (2010), event-chain (2013)...
- Algorithm for generating a configuration according to transition probabilities
  - N-fold way (rejection free) (1975), Walker’s method (1977, 2019), order-N algorithm (1995, 2009)...



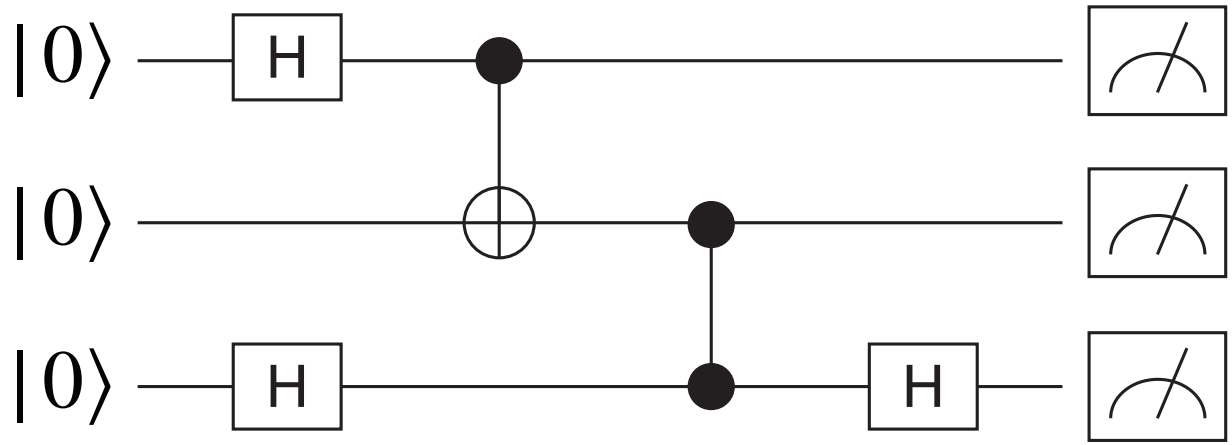
Negative sign problem

- In the path-integral representation for frustrated magnets, fermionic systems, real-time dynamics, the sample weights become negative (or even complex)
  - average sign becomes exponentially small by cancellation at lower temperatures, longer time, and/or larger system sizes
  - specific heat of antiferromagnetic Heisenberg model on kagomé lattice





Unitary dynamics in quantum circuits



- $|000\rangle \Rightarrow \frac{1}{2}(|0\rangle + |1\rangle)|0\rangle(|0\rangle + |1\rangle) = \frac{1}{2}(|000\rangle + |001\rangle + |100\rangle + |101\rangle)$

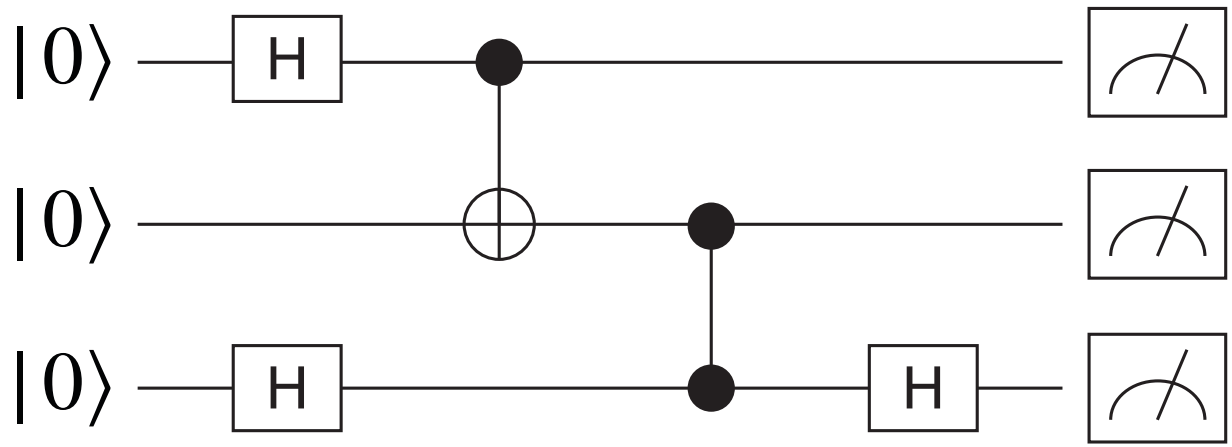
  $\Rightarrow \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle + |111\rangle) \Rightarrow \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle - |111\rangle)$

  $\Rightarrow \frac{1}{2\sqrt{2}}(|00\rangle + |11\rangle)(|0\rangle + |1\rangle) + (|00\rangle - |11\rangle)(|0\rangle - |1\rangle)$

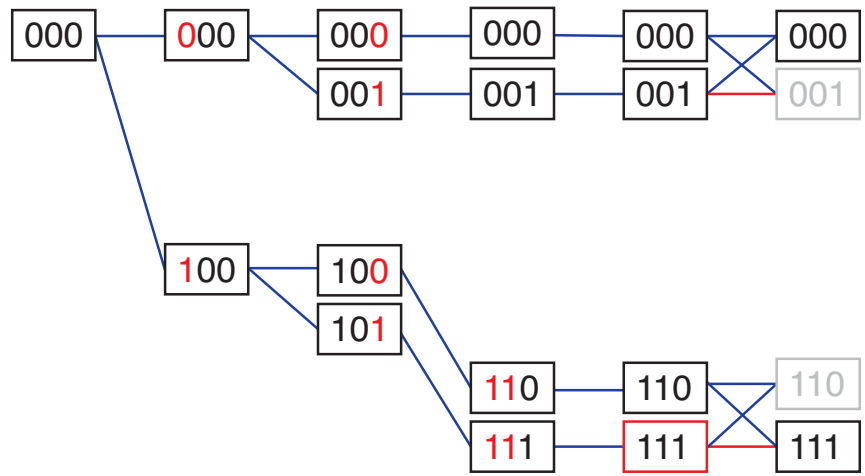
  $= \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |110\rangle + |111\rangle + |000\rangle - |001\rangle - |110\rangle + |111\rangle)$

  $= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

Path sampling of quantum circuits



- State transition diagram



- many states vanishes by interference ⇒ negative (complex) sign problem

Statistical Error of MCMC Measurements

- There is autocorrelation between successive configurations

$$\sigma^2 = \frac{\sigma_0^2(1 + 2\tau_{\mathrm{int}})}{M}$$

  - σ_0^2 : population variance (determined by the ensemble)
  - M : number of Monte Carlo steps
  - τ_{int} : autocorrelation time (determined by the MC dynamics)
  - effective number of independent samples → $M/(1 + 2\tau_{\mathrm{int}})$

- For systems with negative sign problem

$$\sigma^2 = \frac{\sigma_0^2(1 + 2\tau_{\mathrm{int}})}{M\, s^2}$$

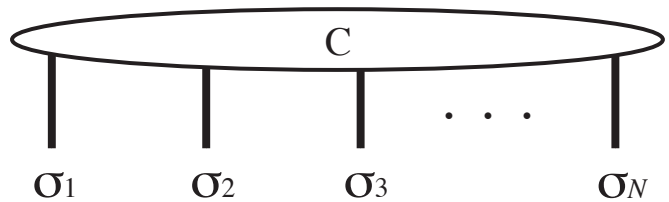
  - s : average sign (exponentially small for lower temperature, longer time, larger system)

Many-body wave function and tensor

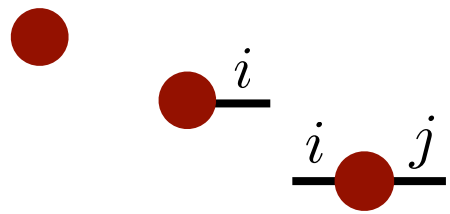
- Wave function of N-qubit (spin-1/2) system

$$|\Psi\rangle = \sum_{\sigma_1,\sigma_2,\cdots,\sigma_N} C_{\sigma_1,\sigma_2,\cdots,\sigma_N} |\sigma_1\sigma_2\cdots\sigma_N\rangle$$

  - linear combination of 2^N states → 2^N coefficients($C_{\sigma_1,\sigma_2,\cdots,\sigma_N}$) should be specified → memory cost $\sim 2^N$
  - C can be regarded as N-leg (rank-N) tensor



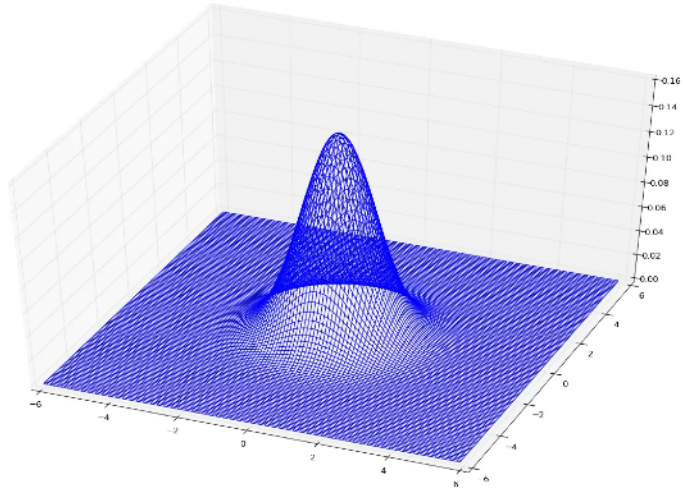
- Tensor = multi-dim array = generalization of vectors/matrices
  - 0-leg tensor → scalar
  - 1-leg tensor → vector
  - 2-leg tensor → matrix
  - ...
  - N-leg tensor → memory/computational cost $\sim \exp(N)$



Tensor representation

- Tensor is ubiquitous
  - probability distribution function

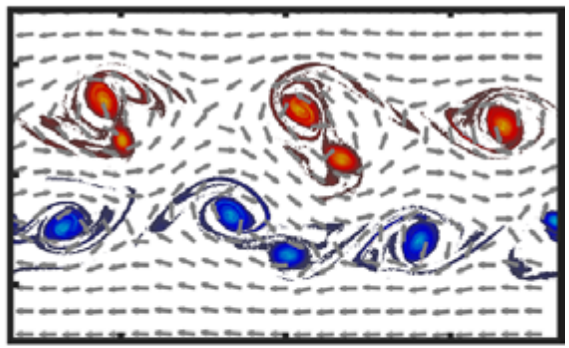
$$P(s_1, s_2, \cdots, s_N)$$

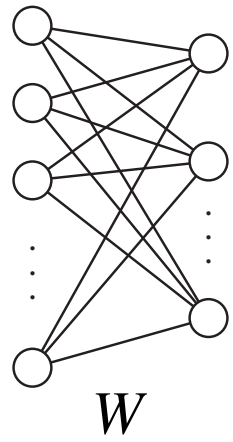


  - multi-dim data
  - grid data, images

$$g(x, y) = g(x_1, x_2, \cdots, x_N, y_1, y_2, \cdots, y_N)$$

    - $(x_1, x_2, \cdots, x_N)$ and $(y_1, y_2, \cdots, y_N)$ are binary number rep. of x and y
    - e.g) 256x256 image → 256x256 matrix or 2^{16} (16-leg) tensor

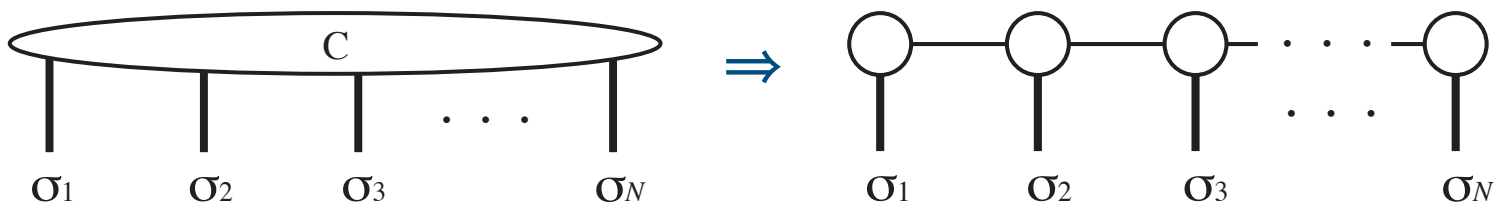




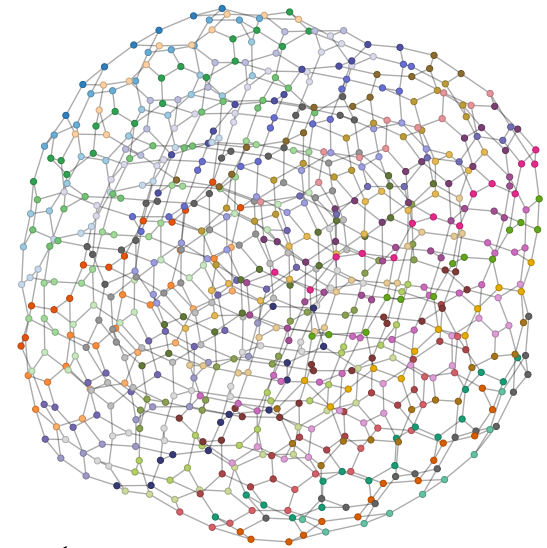
  - neural network
    - weight matrix → tensor with many legs

Tensor decomposition

- Decomposition of a many-leg tensor into product of small tensors

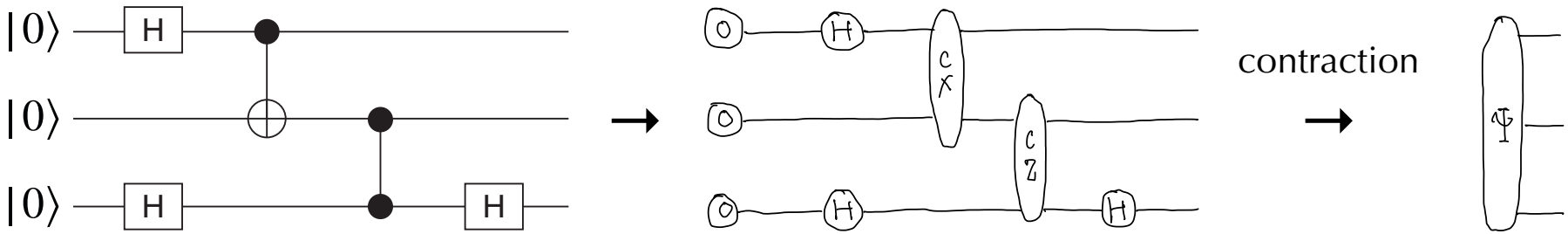


  - taking summation of common indices (contraction)
  - tensor ⇒ tensor decomposition ⇒ tensor network
  - other representations: Boltzmann machine, path-integral, DNN, etc
- Advantage of tensor network
  - data compression $O(\exp(N)) \rightarrow O(N)$
  - freedom in contraction order
  - highly accurate approximation based on SVD
- Tensor network representation
  - approximation of original tensor (e.g. variational w.f.)
  - exact representation
    - quantum states: GHZ state, AKLT state, etc
    - classical/quantum statistical model, quantum circuit, etc

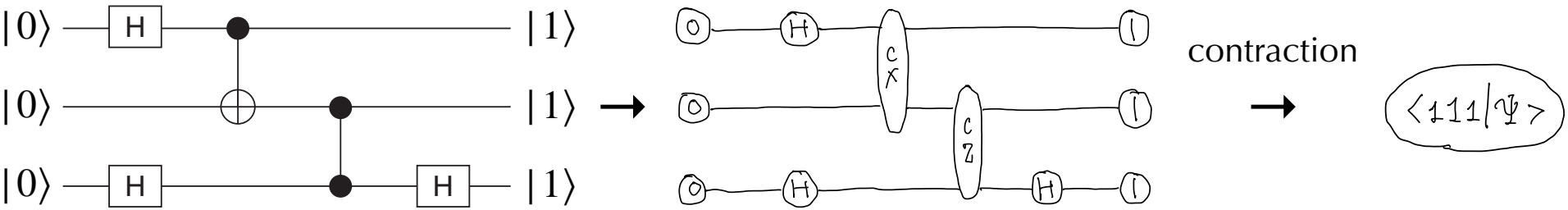


Tensor network representation of quantum circuit

- Tensor network representation of a quantum circuit
  - all bond dimensions are 2



- Taking contraction from the initial state (left to right)
  - equivalent to Schrödinger (state-vector) simulation
- Evaluating an amplitude



  - order of contraction does not change the final result
  - freedom in contraction order → possibility to reduce the cost

Optimization of Contraction Order

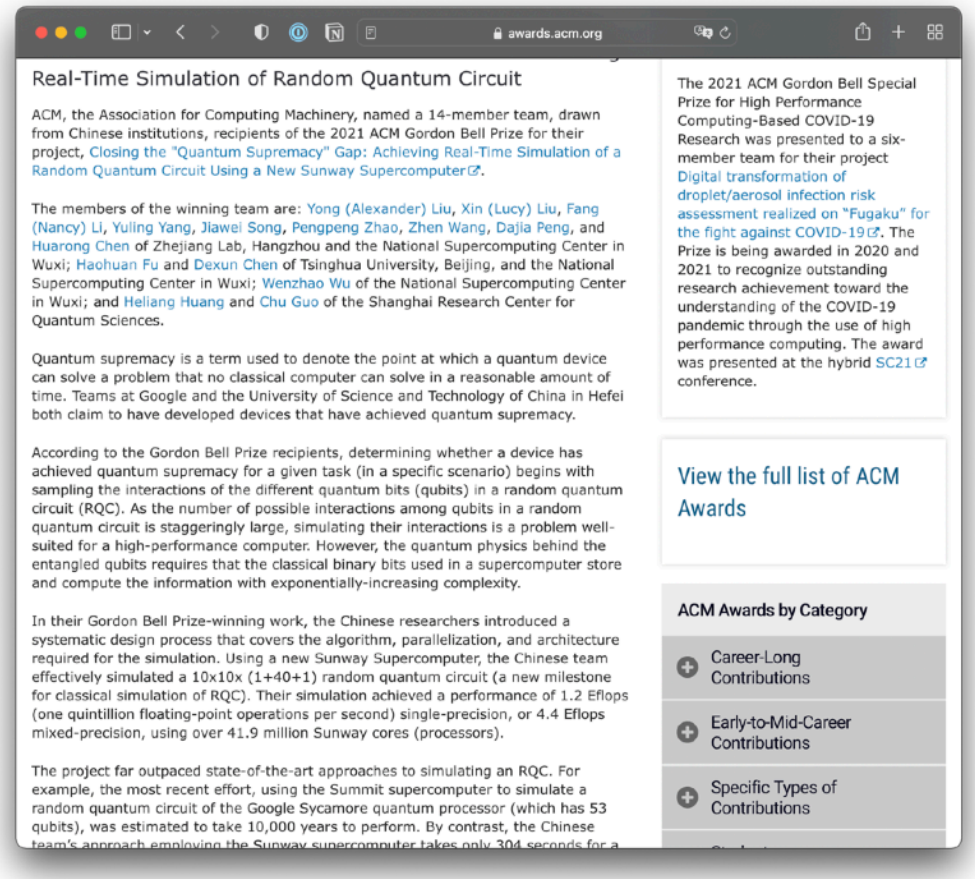
- General guideline for better contraction order
  - avoid tensors with a large number of legs in the middle or at the end of computation
- It is known that finding the optimal contraction order is (at least) #P-hard problem
  - can’t find the best solution in a realistic time
  - many heuristics have been proposed, c.f.,
    - Schutski, R., Khakhulin, T., Oseledets, I., & Kolmakov, D., Simple heuristics for efficient parallel tensor contraction and quantum circuit simulation. Physical Review A, 102(6), 1–11 (2020). https://doi.org/10.1103/PhysRevA.102.062614
    - Gray, J., & Kourtis, S., Hyper-optimized tensor network contraction. Quantum, 5, 1–22 (2021). https://doi.org/10.22331/Q-2021-03-15-410

Slicing tensor network

- Slicing
  - for a subset of bonds in the tensor network
    - fix the bonds to some value
    - these bonds disappear from the tensor network
  - for each fixed value
    - perform contraction independently
  - take summation on fixed-value patterns at the outermost
- Advantage
  - memory cost of contraction of sliced tensor network becomes smaller
  - contraction of each sliced network can be performed in parallel

State-of-the-art tensor network simulation

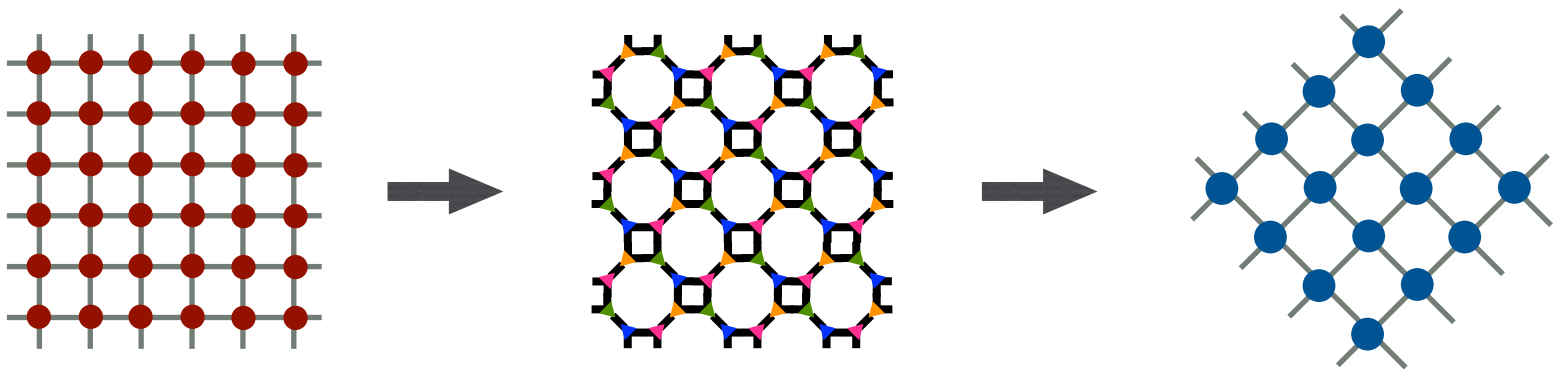
- Y. A. Liu et al., Closing the “quantum supremacy” gap: Achieving real-Time simulation of a random quantum circuit using a new sunway supercomputer. International Conference for High Performance Computing, Networking, Storage and Analysis, SC (2021)
  - Gordon Bell Prize Winner in 2021



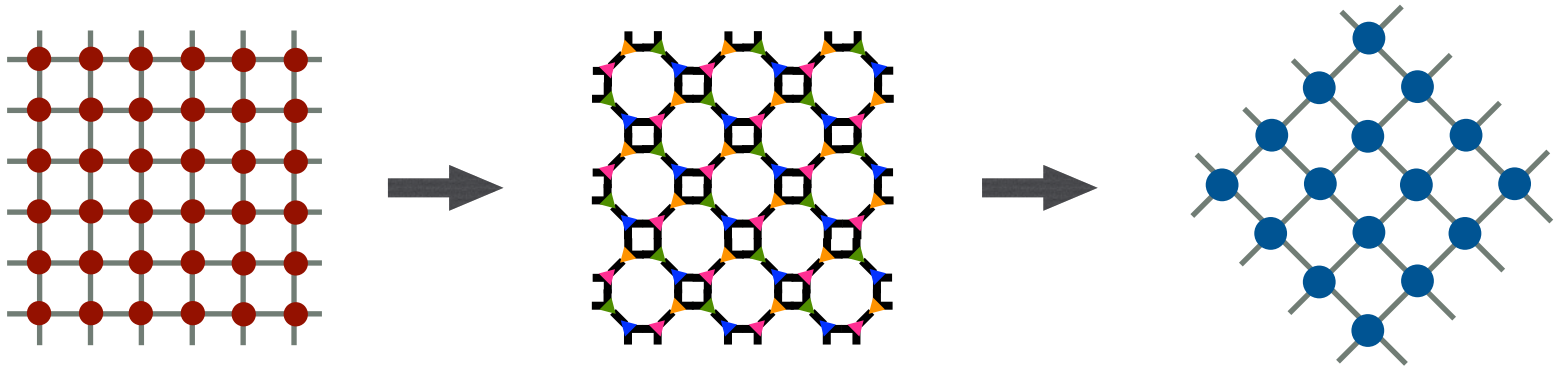
https://awards.acm.org/bell

Tensor network methods in statistical physics

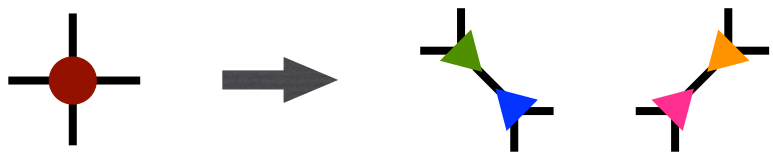
- "Renormalization" (or "Lagrangian") approach
  - coarse graining of tensor network representation of the partition function
  - transfer matrix, tensor network renormalization (TRG), higher-order tensor network renormalization (HOTRG), etc
- "Variational" (or "Hamiltonian") approach
  - tensor-network approximation of strongly correlated many-body quantum states
  - DMRG, PEPS, MERA, etc
- "Exact" contraction of tensor network can not be done in two and higher dimensions
  - low-rank approximation based on eigenvalue/singular value decomposition
  - accuracy of approximation is controlled by "bond dimension": D (or χ)



Tensor renormalization group



- Low-rank approximation based on SVD



computational cost: $O(D^5)$

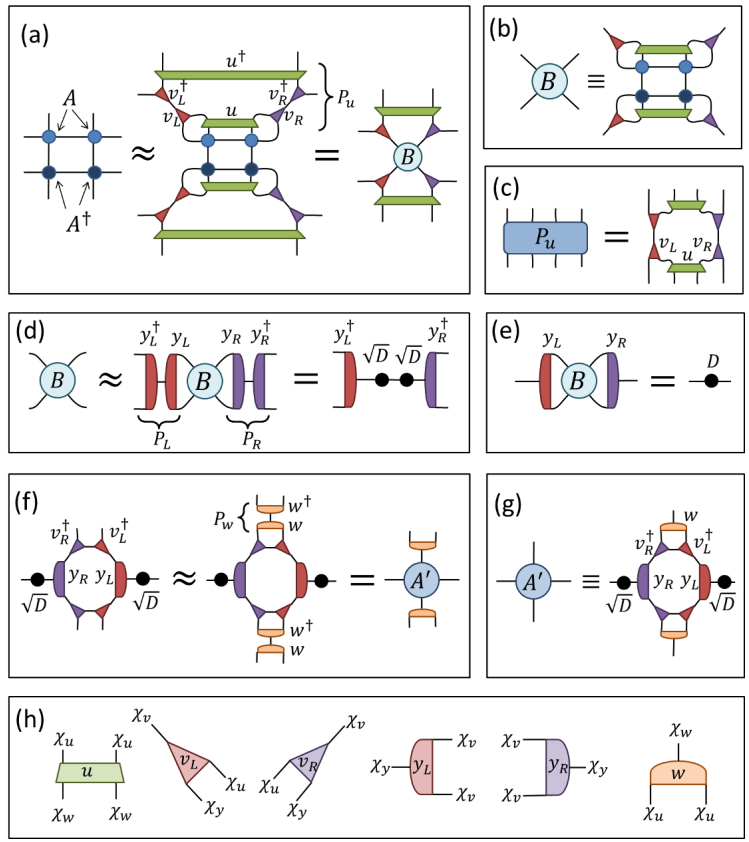
memory cost: $O(D^3)$

- Improvement of accuracy by considering the “environment” effects or removing local correlations
  - second order renormalization (SRG), mean-field SRG, etc
  - tensor network renormalization (TNR), loop TNR, Gilt, etc
  - computational cost increases significantly

Levin & Nave (2007)

More advanced tensor-network methods

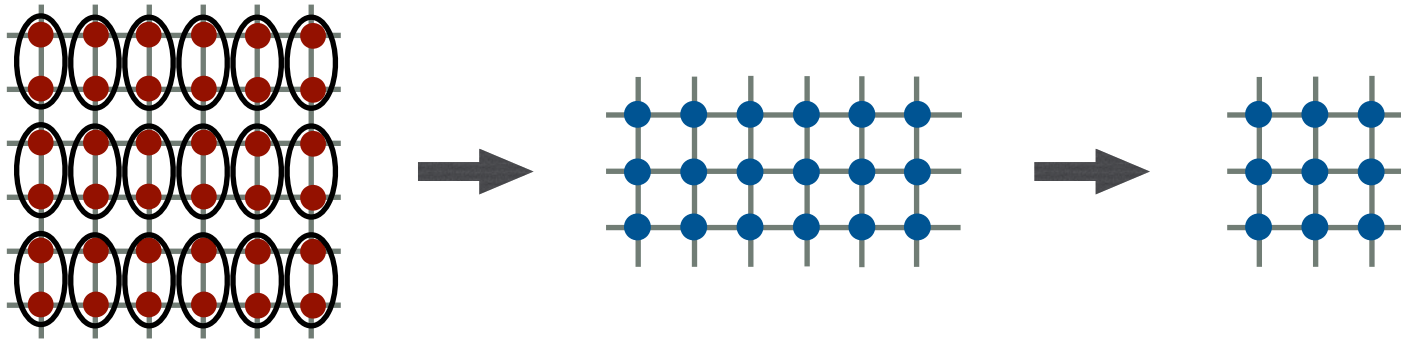
- Improving accuracy
  - including environment effect
    - SRG (2009), CTMRG (1996, 2009)
  - removing local correlations
    - TNR (2015), loop TNR (2017), Gilt (2018)
  - computational cost increases significantly
  - improve accuracy without increasing complexity
    - BTRG (2022)
- Reducing complexity
  - TRG using few-leg tensors
    - ATRG (2020), CATN (2020)
- Generalization to higher-dimensions
  - HOTRG (2012), ATRG (2020), CATN (2020)
- Fermions
  - Grassmann tensor network (2010, 2021)



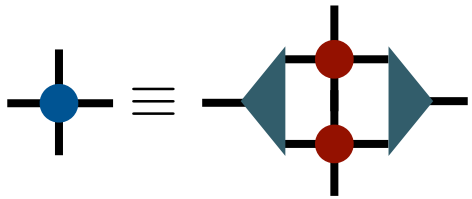
G. Evenbly (2015)

Higher-order tensor renormalization group

- Coarse graining in each direction using HOSVD instead of SVD



- Squeeze tensors using “isometries”



- More accurate than TRG in two dimensions
  - computational cost for 2D: $O(D^7)$
- Works in higher dimensions
  - computational cost in d-dimensions: $O(D^{4d-1})$

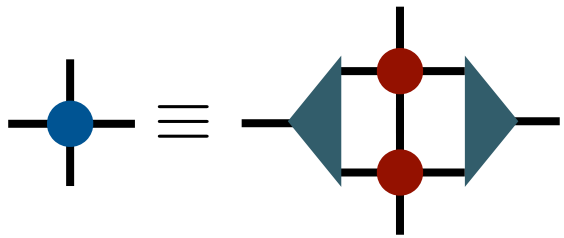
Xie et al (2012)

New tensor network algorithms: ATRG and BTRG

- Anisotropic Tensor Renormalization Group (ATRG)
  - D. Adachi, T. Okubo, S. Todo, Phys. Rev. B 102 054432 (2020) Open Access
  - Works in any dimensions with smaller cost: $O(D^{2d+1}) \ll O(D^{4d-1})$
    - same cost as TRG in two dimensions
  - More accurate than TRG
- Bond-weighted Tensor Renormalization Group (BTRG)
  - D. Adachi, T. Okubo, S. Todo, Phys. Rev. B 105, L060402 (2022) Open Access
  - Works in two dimensions with the same cost as TRG
  - More accurate than TRG and HOTRG

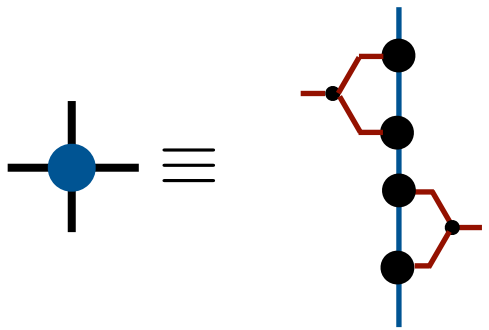
Central idea of ATRG

- Prevent contracting large tensors
  - in HOTRG



$O(D^7)$

  - in ATRG

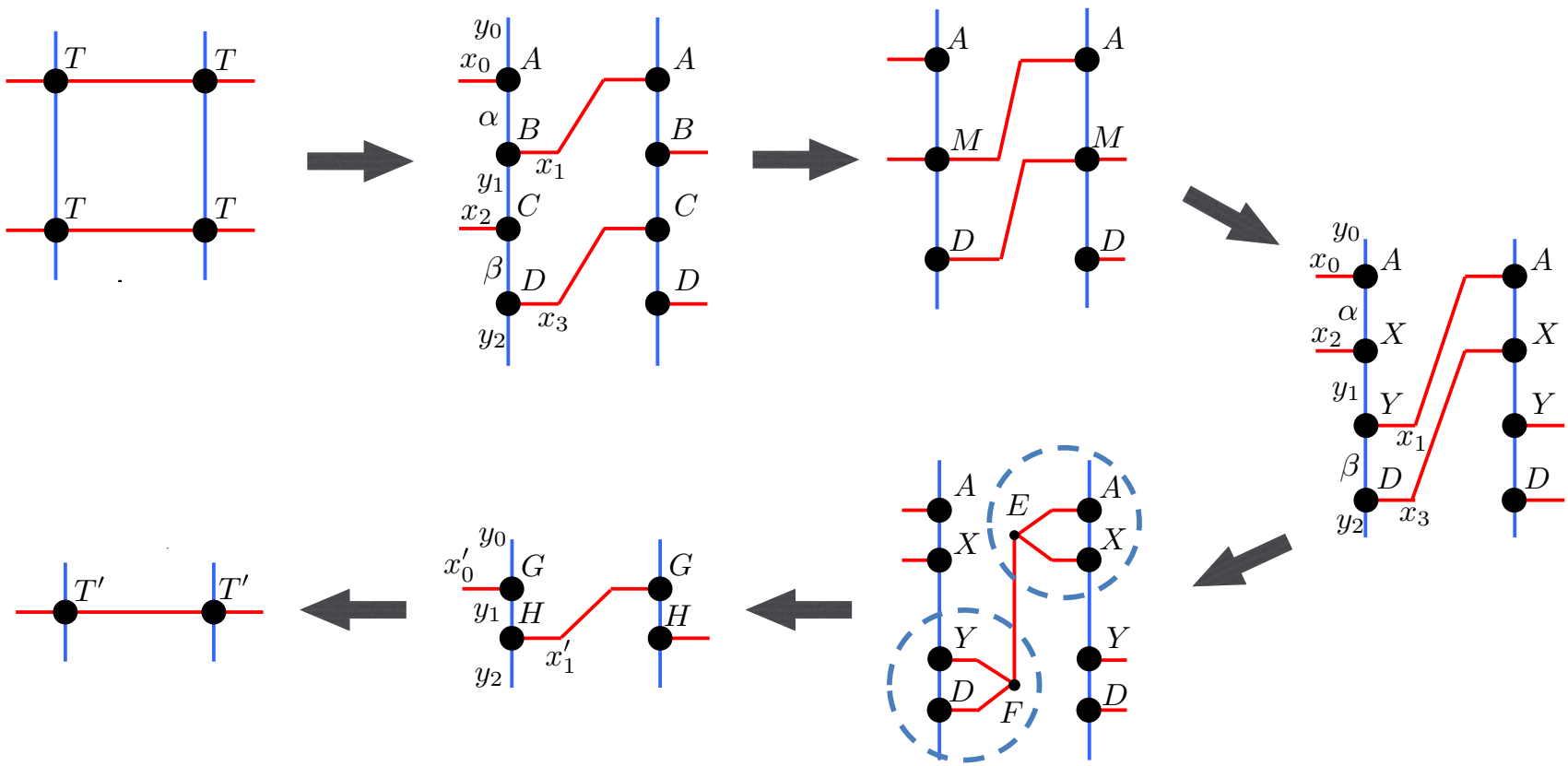


$O(D^5)$

    - decompose the local tensor into small pieces and interchange the position before coarse graining

D. Adachi, T. Okubo, S. Todo, Phys. Rev. B 102 (2020) 054432

One Renormalization Step of ATRG



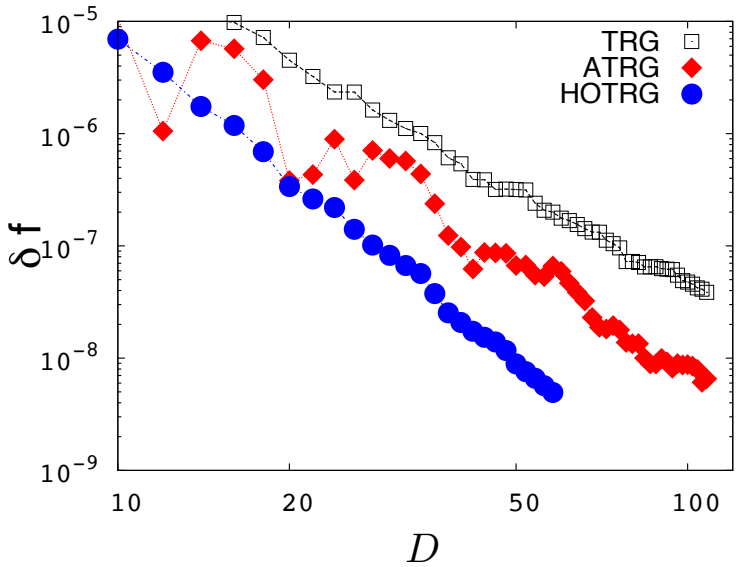
- computational cost: $O(D^5)$
- memory: $O(D^3)$
- generalization to higher dimensions is straightforward

D. Adachi, T. Okubo, S. Todo, Phys. Rev. B 102 (2020) 054432

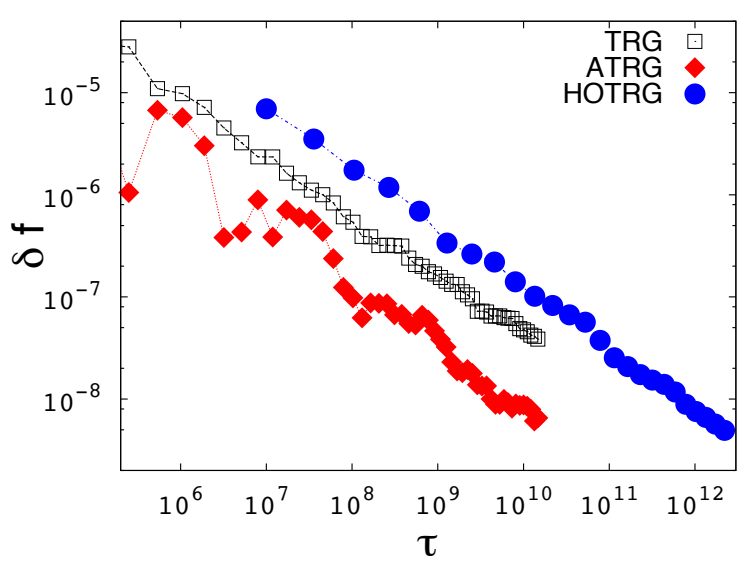
Benchmark Test of ATRG in 2D

- Free energy of square lattice Ising model

comparison at same *bond dimension*



comparison at same *computational cost*



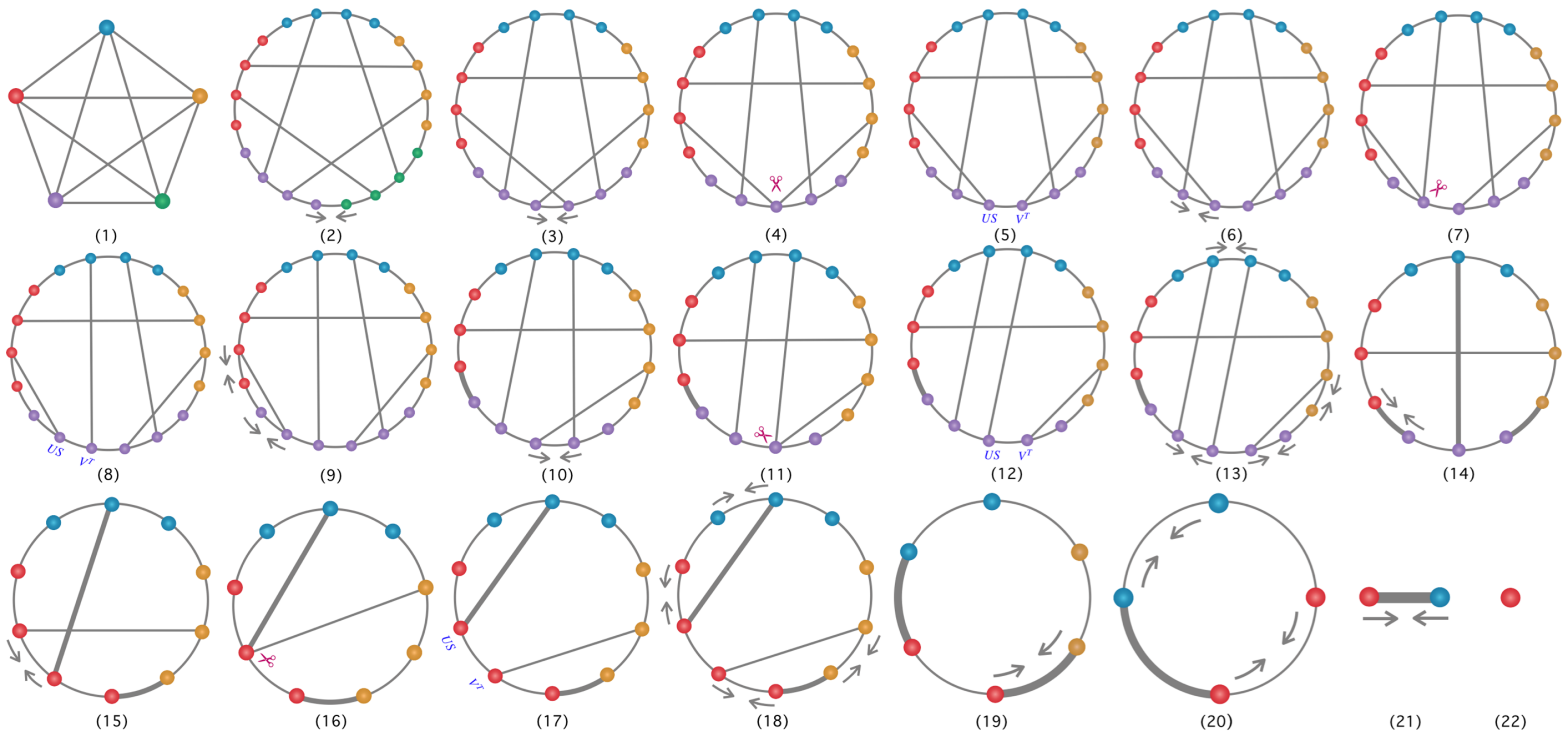
$$\tau \equiv \begin{cases} D^5 & \text{TRG and ATRG} \\ D^7 & \text{HOTRG} \end{cases}$$

- ATRG outperforms TRG at the same bond dimension
- ATRG outperforms HOTRG at the same computational cost

D. Adachi, T. Okubo, S. Todo, Phys. Rev. B 102 (2020) 054432

Contracting arbitrary tensor networks

- Combination of
  - contraction of tensor network with optimized order
  - low-rank approximation using SVD (if bond dimension exceeds D)
  - avoid many-leg tensors and keep MPS form of three-leg tensors



F. Pan, P. Zhou, S. Li, P. Zhang, Phys. Rev. Lett. 125, 60503 (2020)

Markov-chain Monte Carlo

- Various update algorithms
  - local, cluster, worm, event-chain, hybrid (HMC), etc
  - extended ensemble methods
- Pros
  - consistent in long-time limit (if balance condition and ergodicity are satisfied)
  - computational cost increases only linearly (in many cases)
  - trivial parallelization
- Cons
  - slow convergence of statistical error ($\sim 1/\sqrt{M}$)
  - effective number of samples decreases as auto-correlation increases

$$\sigma^2 = \frac{\sigma_0^2(1 + 2\tau_{\mathrm{int}})}{M}$$

  - negative sign for frustrated quantum magnets, fermions, real-time dynamics

Tensor network renormalization group

- Many variants
  - tensor renormalization group (TRG) by Levin and Nave (2007)
  - higher-order tensor renormalization group (HOTRG) by Xie et al (2012)
  - tensor network renormalization (TNR) by Evenbly and Vidal (2015)
  - anisotropic tensor renormalization group (ATRG) by Adachi et al (2020)
  - bond-weighted tensor renormalization group (BTRG) by Adachi et al (2022)
- Pros
  - (super-?)exponentially accurate for large bond dimension D
- Cons
  - systematic error (bias) due to finite bond dimension D
  - computational cost increases rapidly as D^{α}
  - few methods for higher dimensions

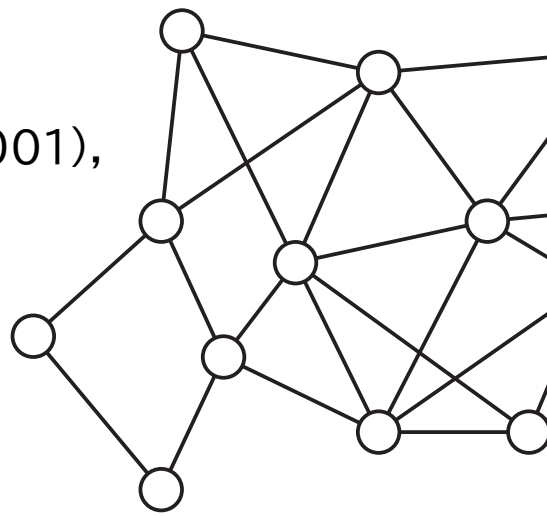
Approximate contraction of tensor network

- "Exact" contraction of tensor network can not be done in two and higher dimensions
  - low-rank approximation based on singular value decomposition
    - systematic bias due to low-rank approximation
  - accuracy of approximation is controlled by cutoff (or bond dimension: D)
    - convergence is fast but not smooth (even not monotonic) in many cases

- How can we eliminate the systematic bias in approximate contraction?
  - combination with Markov chain Monte Carlo?

Advances in Markov chain Monte Carlo

- Representation (definition of “configurations” and “weighs”)
  - path integral representation for quantum Monte Carlo (1976), Bayesian inference (1990), tensor network representation
- Choice of ensemble
  - extended ensemble method: multicanonical MC (1991, 2001), exchange MC (1996), lifting (2000)...
- Generation of set of candidate configurations
  - non-local (cluster) updates: Swendsen-Wang (1987), Hamiltonian MC (1987), loop (1993), worm (1998)...
- Choice of transition kernel (probabilities)
  - Metropolis, heat bath (Gibbs sampler), over-relaxation (1987), irreversible kernel (2010), event-chain (2013)...
- Algorithm for generating a configuration according to transition probabilities
  - N-fold way (rejection free) (1975), Walker’s method (1977, 2019), order-N algorithm (1995, 2009)...



Agenda

- Introduction
  - Markov-chain Monte Carlo and negative sign problem
  - Tensor network representation and approximate contraction

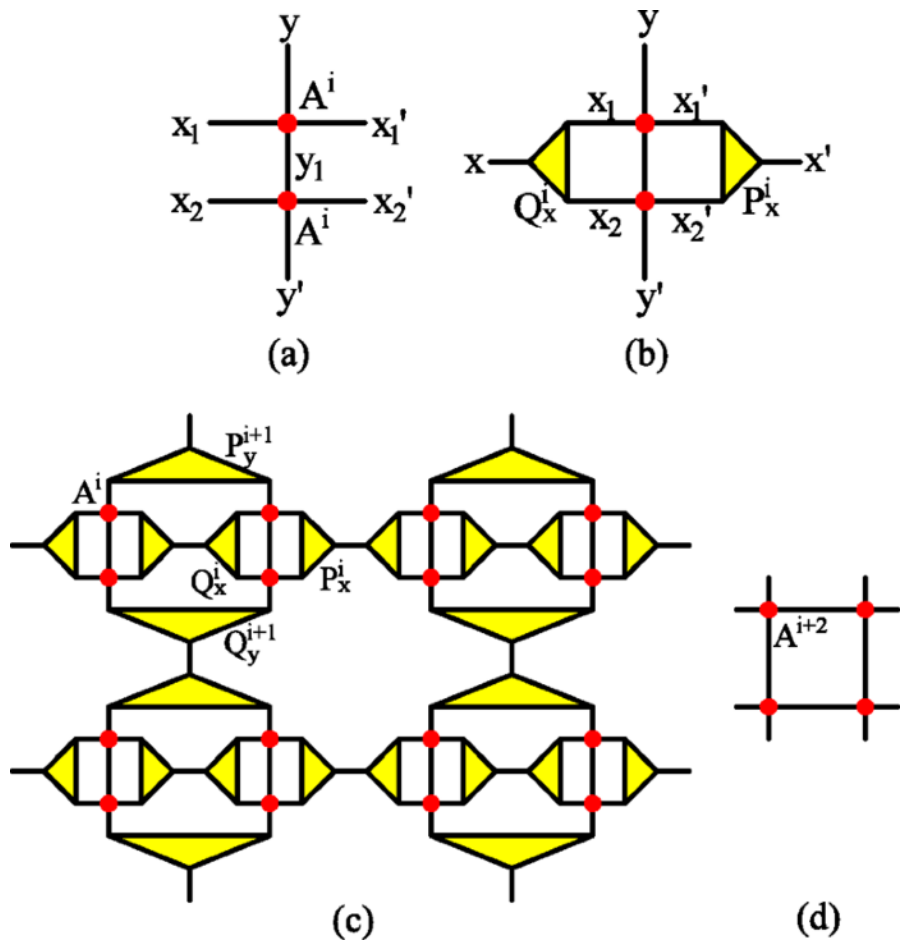
- Sampling approach in tensor network contraction
  - Projector formulation of tensor network method
  - Sampling projectors

- Monte Carlo in tensor network representation
  - Sequential Monte Carlo in tensor network representation
  - Markov-chain Monte Carlo in tensor network representation
  - Results

- Summary

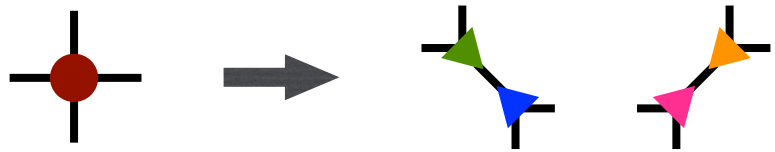
Projector formulation of tensor network methods

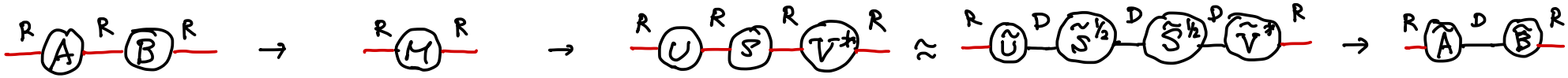
- HOTRG (Xie et al 2012)



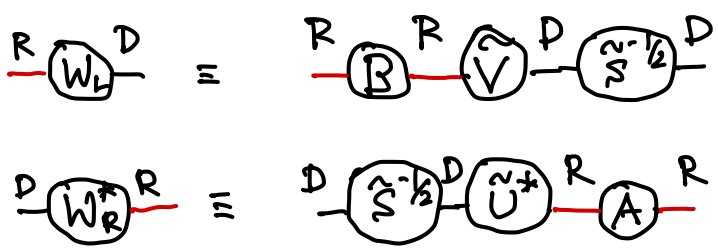
Projector formulation of tensor network methods

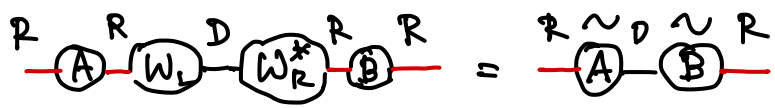
- TRG (Levin-Nave 2007)
  - original truncation based on SVD





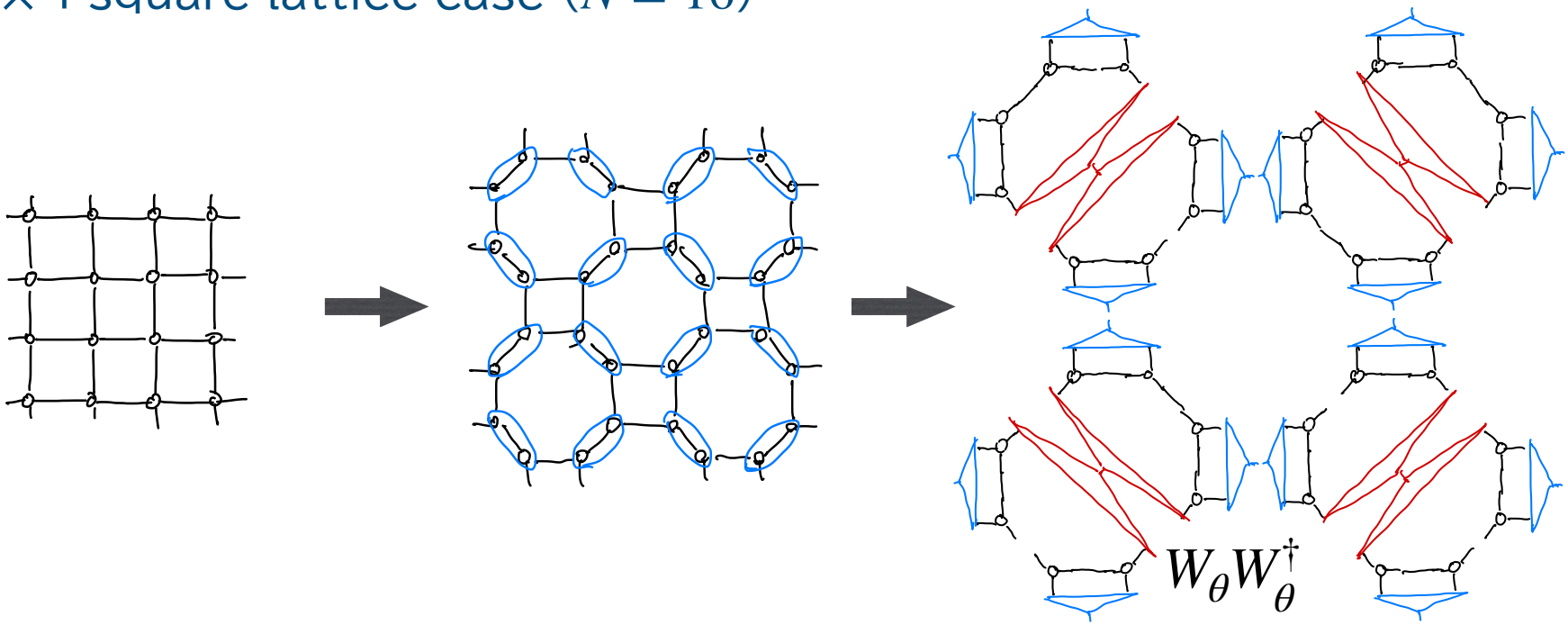
  - truncation based on projector





Levin-Nave TRG in projector formulation

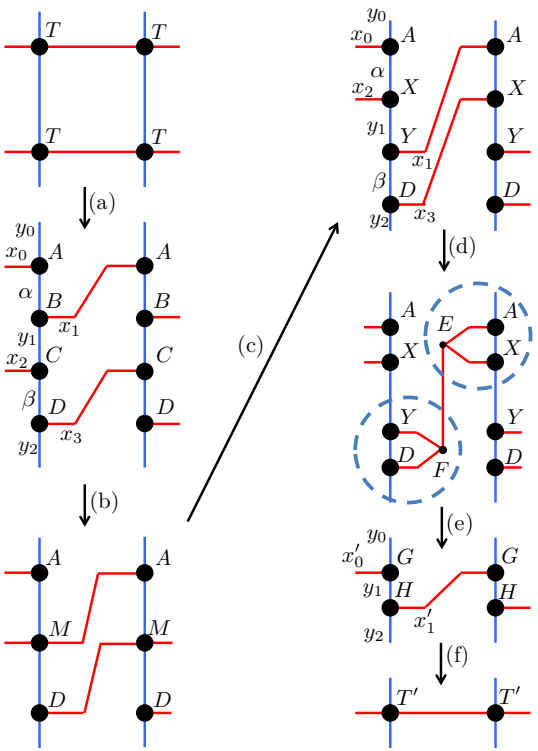
- 4×4 square lattice case ($N = 16$)

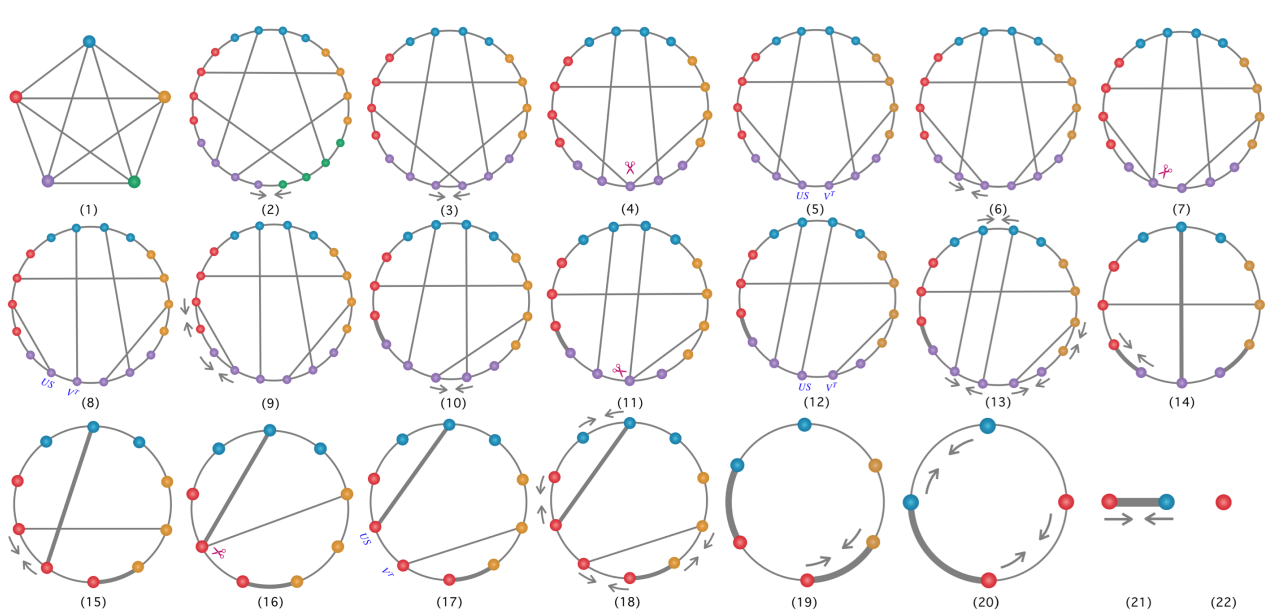


- approximate partition function
  - contraction of tensor network of $2N$ initial tensors and $(N-4)$ projectors
  - contraction graph of depth $\sim \log N$

Projector formulation of tensor network methods

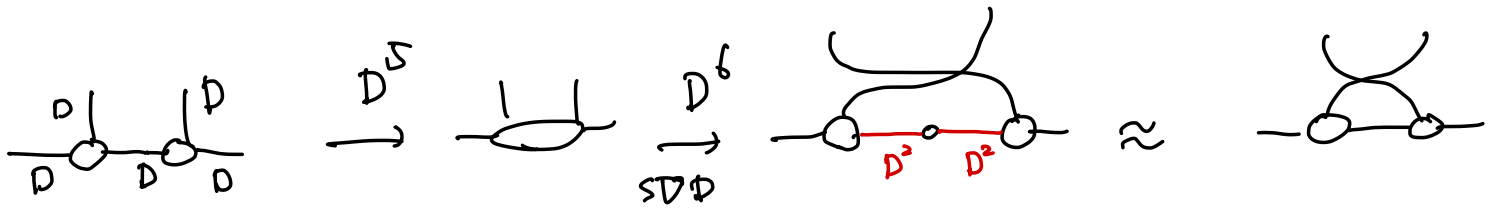
- ATRG (Adachi et al 2020) and CATN (Pan et al 2020)



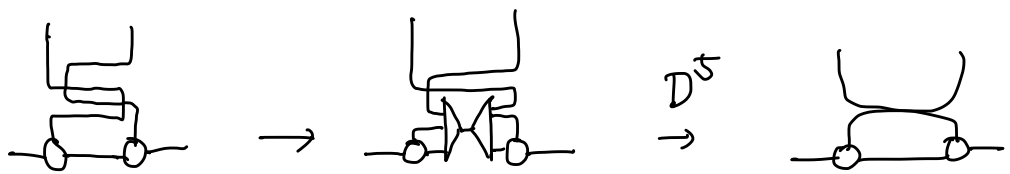


Projector formulation of tensor network methods

- ATRG (Adachi et al 2020) and CATN (Pan et al 2020)
  - leg swap based on SVD



  - leg swap based on projector



- Any tensor network renormalization methods can be reformulated using projectors (?)

Unbiased Monte Carlo for the age of tensor networks

Andrew J. Ferris
ICFO—Institut de Ciencies Fotoniques, Parc Mediterrani de la Tecnologia, 08860 Barcelona, Spain
(Dated: July 6, 2015)

A new unbiased Monte Carlo technique called Tensor Network Monte Carlo (TNMC) is introduced based on sampling all possible renormalizations (or course-grainings) of tensor networks, in this case matrix-product states. Tensor networks are a natural language for expressing a wide range of discrete physical and statistical problems, such as classical and quantum systems on a lattice at thermal equilibrium. By simultaneously sampling multiple degrees of freedom associated with each bond of the tensor network (and its renormalized form), we can achieve unprecedented low levels of statistical fluctuations which simultaneously parallel the impressive accuracy scaling of tensor networks while avoiding completely the variational bias inherent to those techniques, even with small bond dimensions. The resulting technique is essentially an aggressive multi-sampling technique that can account for the great majority of the partition function *in a single sample*. The method is quite general and can be combined with a variety of tensor renormalization techniques appropriate to different geometries and dimensionalities.

Random choice of projectors (Ferris 2015)

- Choose D vectors from R ($=D^2$) basis vectors
  - $N_p = \binom{R}{D}$ candidate projectors
  - basis vectors are generated by SVD and chosen randomly according to its singular value
- Stochastic projector is generated as a linear combination of D projectors
  - such that on average,

$$\frac{1}{N_p}\sum_{\theta} W_\theta W_\theta^\dagger = E$$

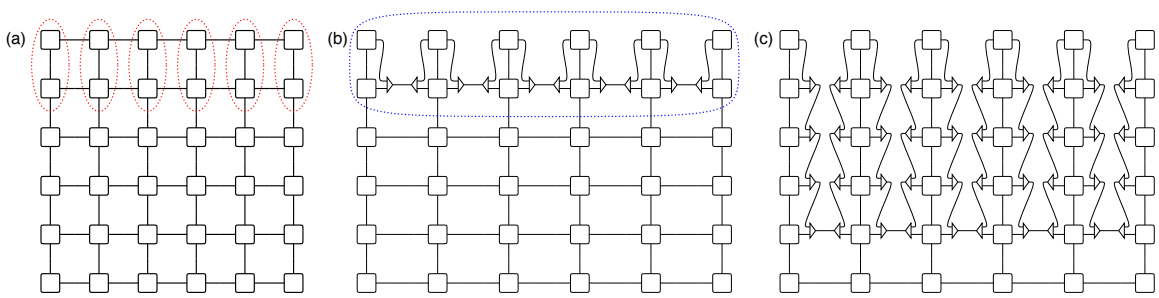
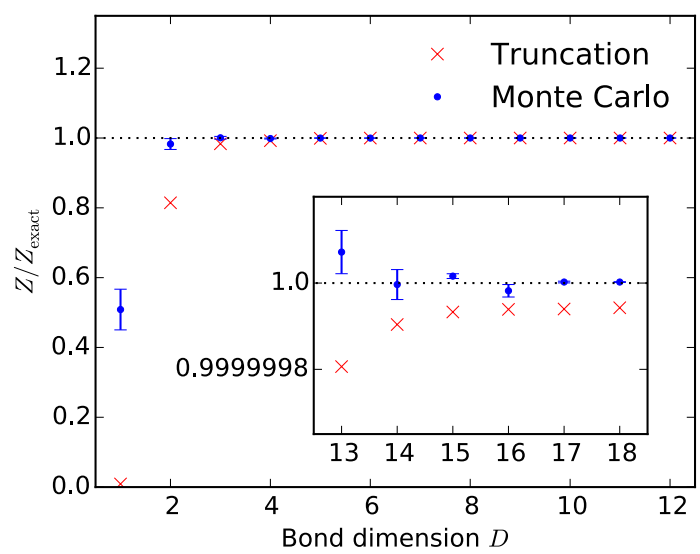


FIG. 1: (a) A two-dimensional partition function expressed as a tensor network. Boundary-MPS contraction proceeds by combining the top row of tensors with the next (red circles), and then (b) projecting the combined horizontal bonds to a subspace of maximal dimension D (triangular tensors). Typically, the projectors are chosen to maximize the fidelity of resulting effective state of the upper-half of the system, outlined in blue. (c) After repetition, the end result is a tensor network that is contractible with cost linear in system size.



partition function of six-vertex model (16x16)

Summary of arXiv:1507.00767

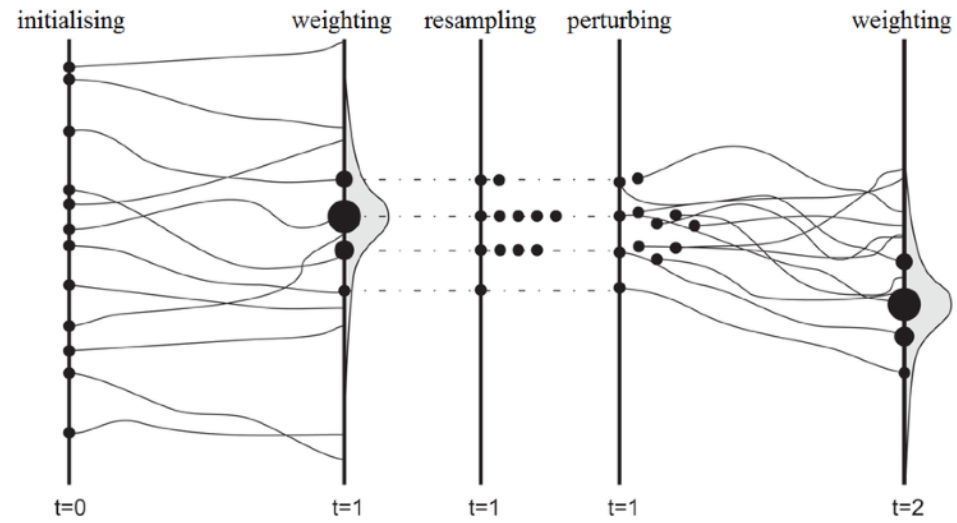
- “Perfect” sampling of tensor network contraction
  - can be applied to other tensor network renormalization variances (e.g. HOTRG, ATRG)
  - only works for finite systems
  - unbiased? → yes (for partition function)
    - NB: biased for free energy, expectation value of physical quantities
  - suffers from exponentially large variance
    - proposed method = sequential Monte Carlo without resampling
    - variance of weight of “walkers” is multiplicative and increases exponentially

Agenda

- Introduction
  - Markov-chain Monte Carlo and negative sign problem
  - Tensor network representation and approximate contraction
- Sampling approach in tensor network contraction
  - Projector formulation of tensor network method
  - Sampling projectors
- Monte Carlo in tensor network representation
  - Sequential Monte Carlo in tensor network representation
  - Markov-chain Monte Carlo in tensor network representation
  - Results
- Summary

Sequential Monte Carlo

- aka
  - green's function Monte Carlo
  - transfer matrix Monte Carlo
  - population Monte Carlo
  - Monte Carlo filter
  - particle filter
  - bootstrap filter
  - SISR (sequential importance sampling with resampling)



https://www.researchgate.net/figure/Sequential-Monte-Carlo-scheme_fig2_322302619

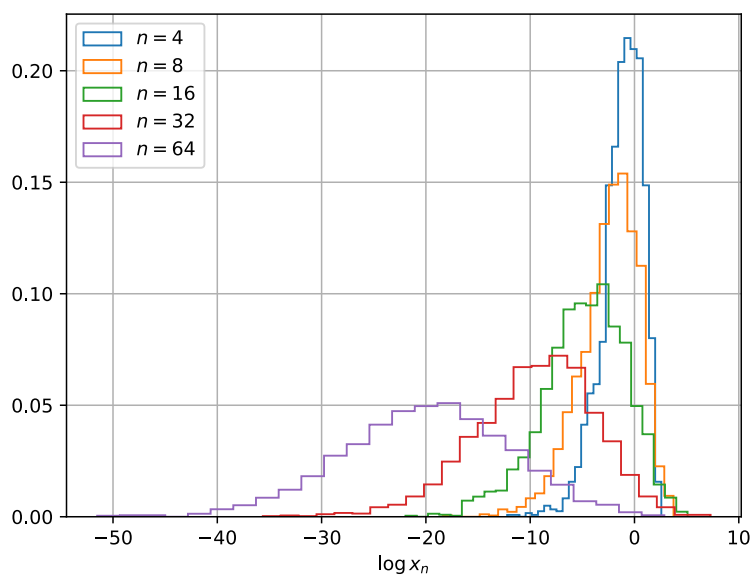
- Central idea of SMC
  - approximate probability distribution by (weighted) ensemble of particles
  - by using Markov chain
  - control variance by resampling

A simple example

$$x_0 = 1$$

$$x_k = \xi_k x_{k-1} \quad (k = 1,2,\cdots,n)$$

- ξ_k is sampled independently randomly from uniform distribution between 0 and 2
- expectation value: $\langle x_n \rangle = 1$
- variance increases rapidly for large n

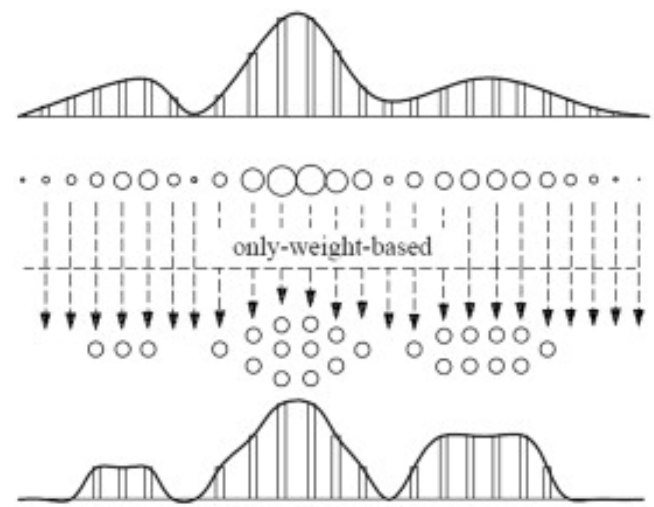


Resampling

- Simple sequential importance sampling becomes unstable for large steps
  - weight of each walker is updated randomly by weight factors: $W = w_1 w_2 w_3 \cdots$
  - random walk diffusion in logarithmic scale
  - weight degeneracy: weight variance (discrepancy between weights) grows exponentially and only a few walkers dominate
- Resampling is necessary to stabilize the algorithm
  - resampling:

$$P_i \simeq \sum_k W_k \delta_{i,i_k} \Rightarrow \sum_k \delta_{i,\tilde{i}_k}$$

  - after resampling, all walkers share the same weight: $\sum_k W_k / N_w$



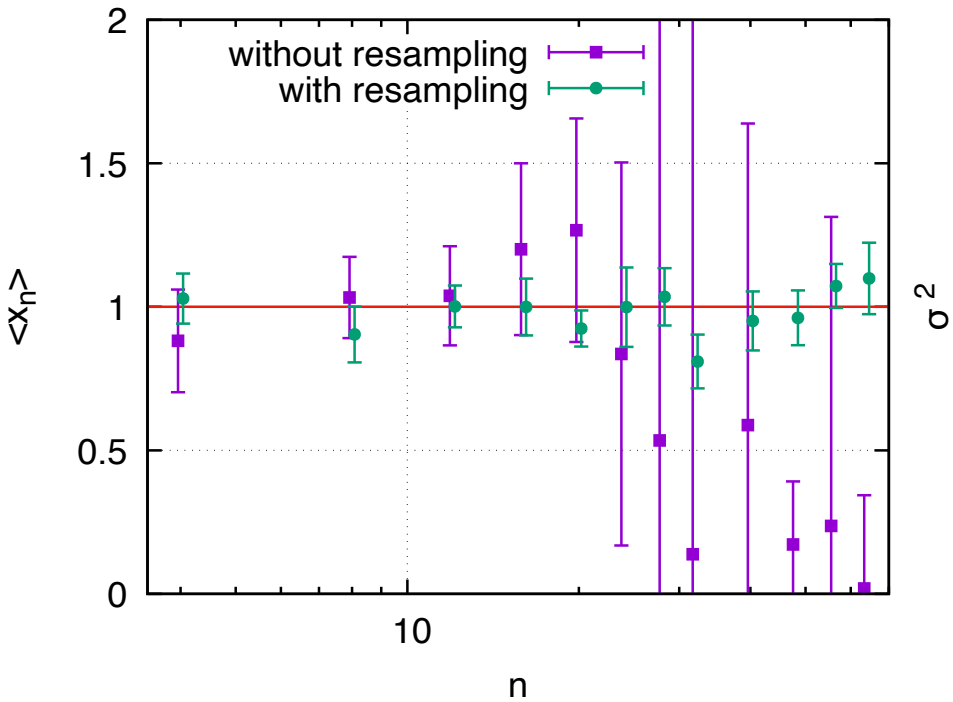
Li-Stattar-Sun (2012)

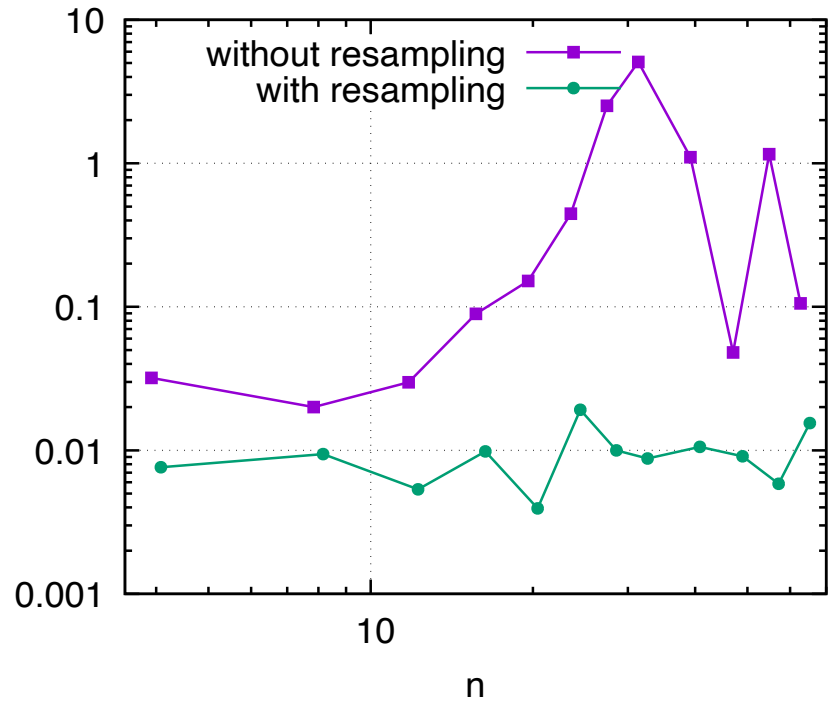
Effect of resampling

$$x_0 = 1$$

$$x_k = \xi_k x_{k-1} \quad (k = 1,2,\cdots,n)$$

- ξ_k is sampled independently randomly from uniform distribution between 0 and 2





Tensor network sequential Monte Carlo

- Markov chain
  - sequential selection of projectors $p(\theta_1, \theta_2, \cdots, \theta_n) = p(\theta_1)p(\theta_2)\cdots p(\theta_n)$
- Walker
  - each Markov chain sequence
  - a number of Markov chains run simultaneously
- Weight
  - partition function with present projector configuration
- Resampling
  - necessary for controlling the variance between walkers

- Systematic error in physical quantities
  - $\sim 1/M$ (M: number of walkers)
- Memory cost
  - M walkers should be simulated simultaneously

Markov-chain Monte Carlo approach

- Determine projector candidates from SVD during the conventional (deterministic) TRG
  - projectors becomes independent with each other and can be sampled independently

$$p(\theta_1, \theta_2, \cdots, \theta_n) = p(\theta_1)p(\theta_2)\cdots p(\theta_n)$$

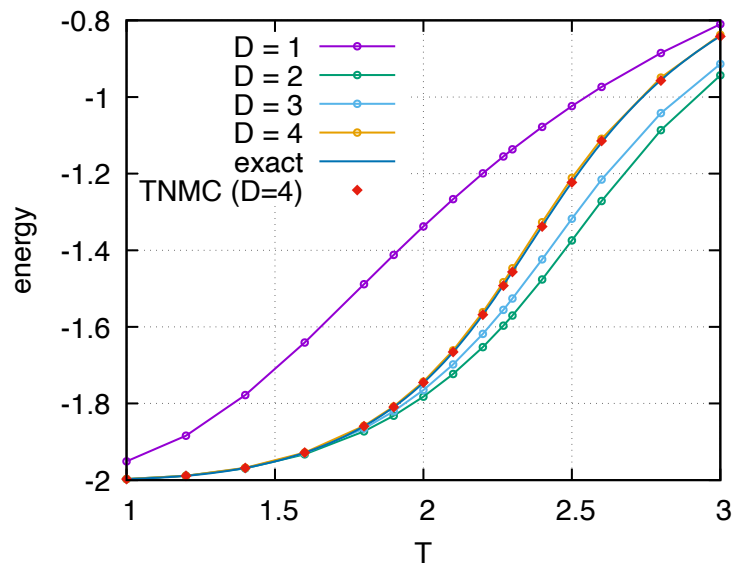
  - (exact) tensor network representation of partition function

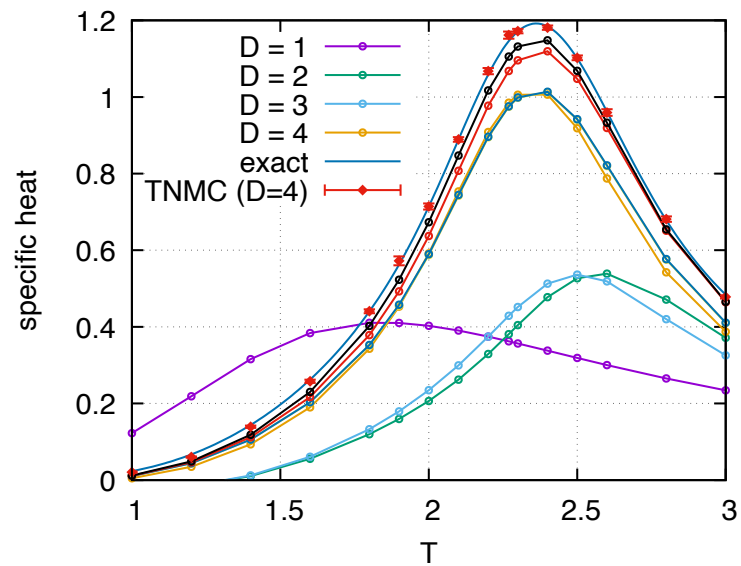
$$Z = \sum_{\{\theta_i\}} g(\theta_1, \theta_2, \cdots, \theta_n)p(\theta_1, \theta_2, \cdots, \theta_n) = \sum_{\{\theta_i\}} g(\theta_1, \theta_2, \cdots \theta_n)p(\theta_1)p(\theta_2)\cdots p(\theta_n)$$

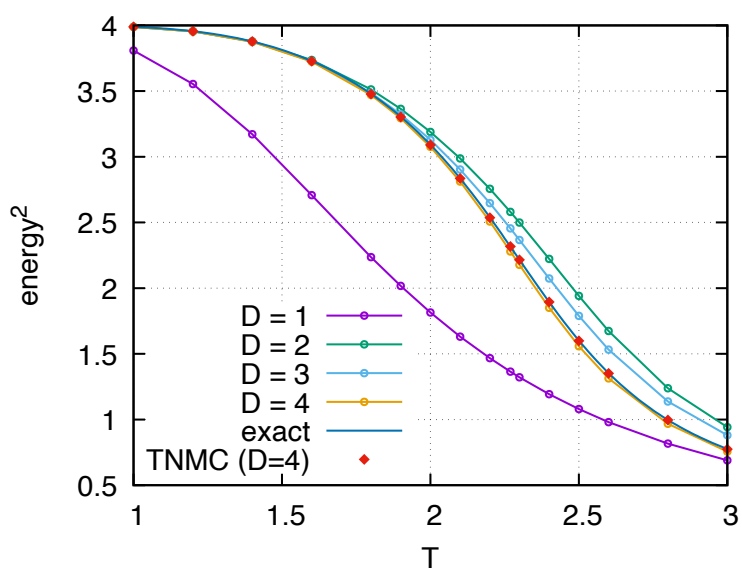
- Sample projectors $\{\theta_i\}$ using Markov-chain Monte Carlo
  - propose new θ_i according to $p(\theta_i)$
  - Metropolis update with $P = \min(1, g(\theta_1, \theta_2, \cdots, \theta_i', \cdots, \theta_n)/g(\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_n))$
  - update of weights is $O(\log N)$ and includes matmul only
  - SVDs are required during initialization stage only
- Physical quantities
  - can be evaluated by using impurity tensor technique (without systematic bias)

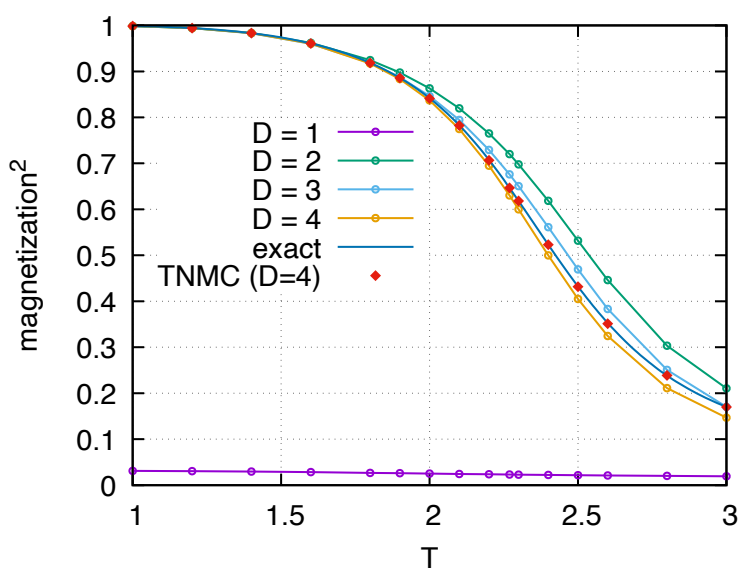
Comparison with Levin-Nave TRG + impurity tensor

- Square-lattice Ising model ($L = 8$)





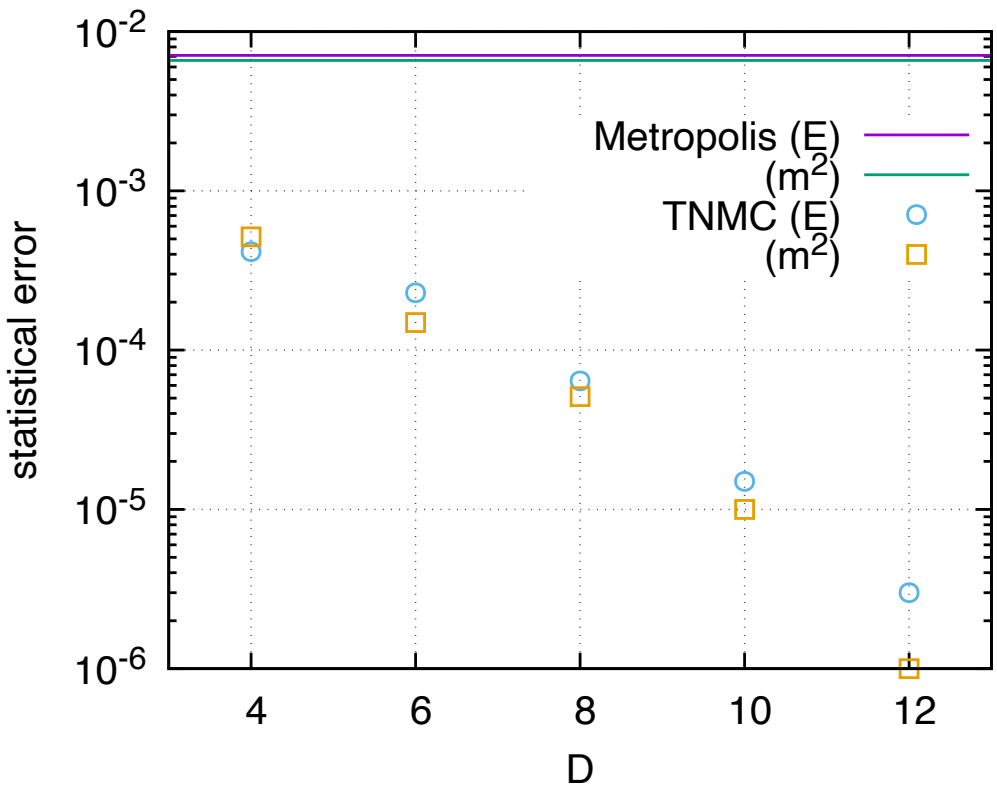


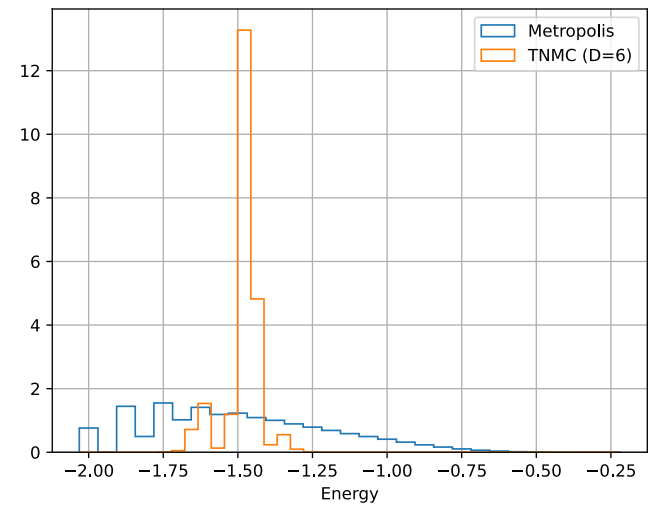


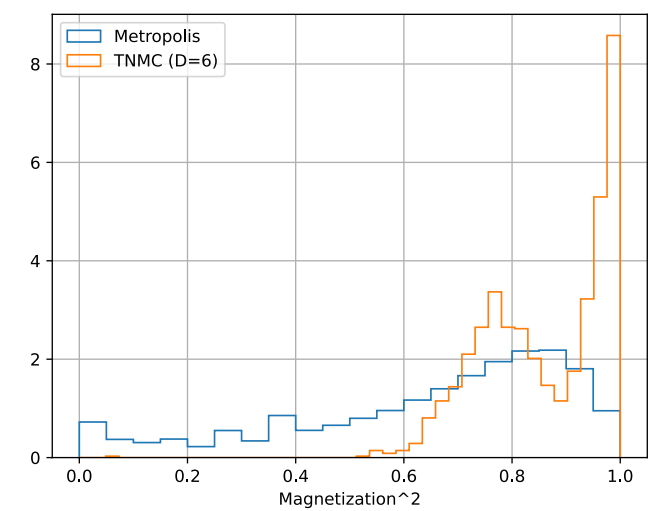
Comparison with Metropolis algorithm

- Square-lattice Ising model ($L = 8$, $T = T_c$, $N_{\mathrm{mcs}} = 8192$)
  - statistical error is smaller by orders of magnitude
  - statistical error decreases exponentially as D is increased

$$\sigma^2 = \frac{\sigma_0^2(1 + 2\tau_{\mathrm{int}})}{M}$$





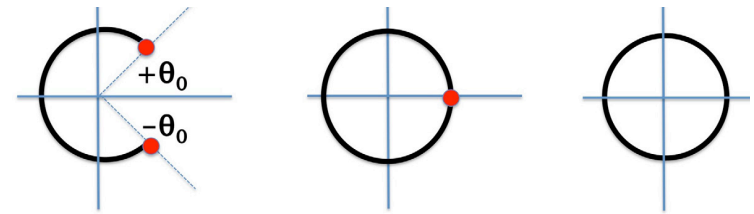


Ising model in imaginary external field

- Square lattice Ising model

$$H = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i$$

Yang-Lee zeros on complex plane of fugacity z



T > T_c T = T_c T < T_c

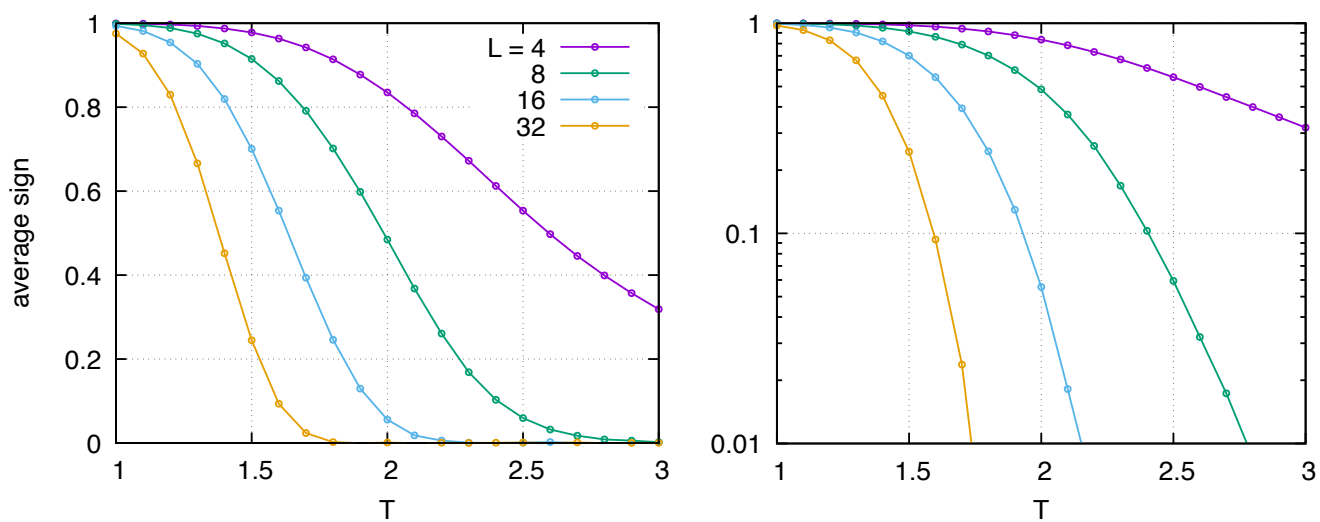
- pure imaginary external field

$$h = i\pi/2\beta \quad \Rightarrow \quad z = e^{-2\beta h} = -1$$

- non-positive Boltzmann weight (m: total magnetization)

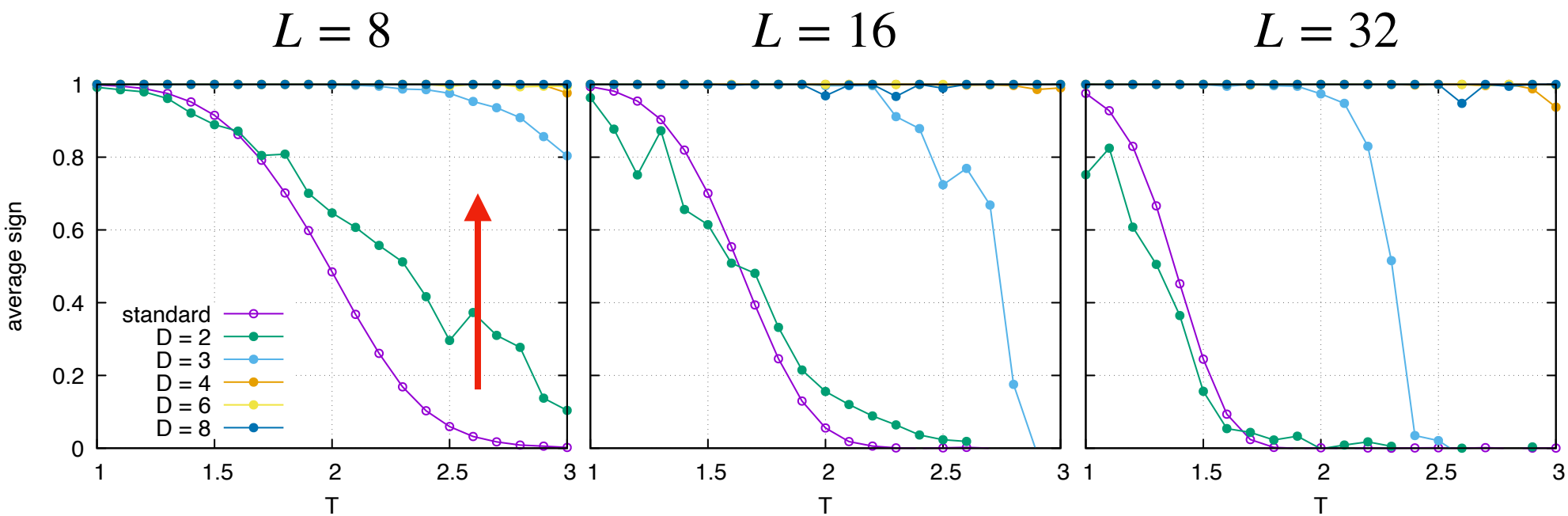
$$W = e^{\beta \sum \sigma_i \sigma_j} \times (-1)^{m/2} \qquad (m\text{: total magnetization})$$

- Standard Markov chain Monte Carlo suffers from severe negative sign problem



Ising model in imaginary external field

- Our proposed method also has negative signs for small D
  - $D = 2$ results are almost similar to the standard method
  - NB: negative signs can appear for small D even if the original model is free from negative sign
- However, average sign is improved drastically as we increase D



Summary

- Markov chain Monte Carlo in tensor network representation
  - reducing statistical error using approximate tensor network contraction
  - removing systematic bias by sampling singular vectors (projectors) using MCMC

    → avoid divergence of statistical error, negative signs and systematic bias
- Computational complexity of one Monte Carlo update
  - $O(D^{\alpha} N \log N)$
  - matmul only (no SVD) during MCMC sampling → ideal for modern GPGPU or HPC
- Similar formulation is possible for sequential Monte Carlo with resampling
  - advantage in calculating free energy/partition function
- Applications: HOTRG (2012), ATRG (2020), quantum spin models, fermions, lattice QCD, quantum circuits, etc

S. Todo (in preparation)