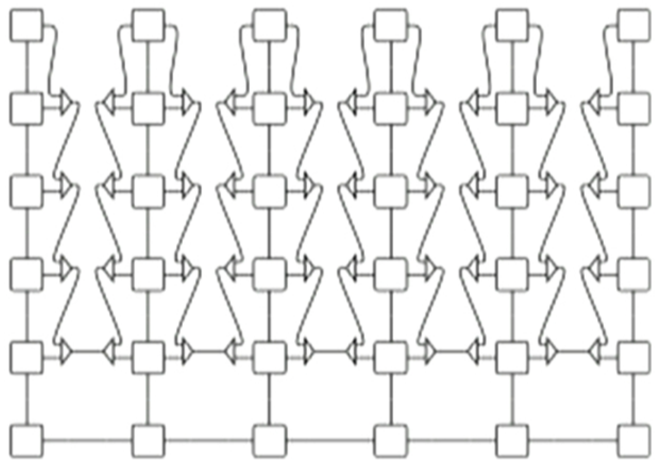
Title: Monte Carlo for the age of Tensor Networks

Date: Oct 06, 2015 01:00 PM

URL: http://pirsa.org/15100073

Abstract: <p>Modern numerical methods have revolutionized the practice of science, creating a third discipline between traditional theory and experiment. Perhaps the most widely known and successful technique has been the Monte Carlo method in general, and the Metropolis algorithm in particular. In this talk, I will present a new way of performing unbiased Monte Carlo simulations based on highly-accurate tensor network contractions. The resulting technique inherits the legendary precision of tensor networks without any of the variational bias. From a Monte Carlo point-of-view, the method can be seen as an aggressive multi-sampling technique where each sample may account for the vast majority of the entire partition function resulting a a drastic reduction in sample-to-sample variance (in contrast to standard
configuration-based Monte Carlo, where only a small subset of possible configurations are sampled). The presented results are all classical, though applications to quantum systems and the sign problem will be discussed.</p>

Monte Carlo for the age of Tensor Networks



Andy Ferris
ICFO (Spain)

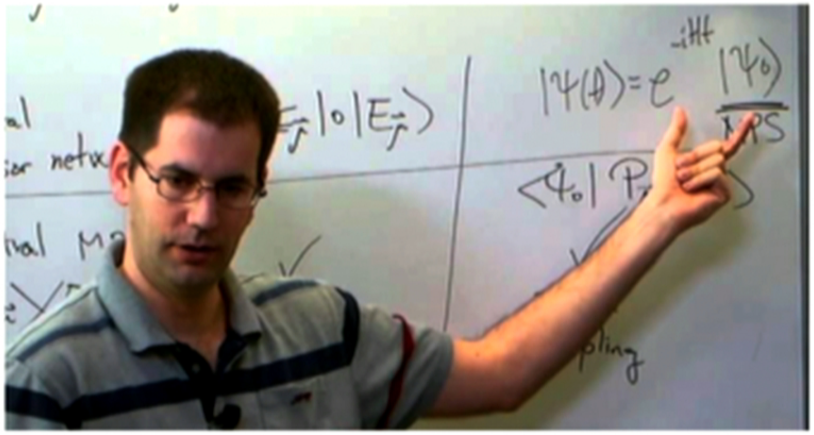
Motivation (to science)

Numerics has become a core way of performing science, between theory and experiment.

Constantly, we discover new challenging problems requiring improved tools and algorithms

Some generic tools can be widely applied
(e.g. Monte Carlo)

Motivation (to me, 5 years ago)



Guifre Vidal

= the boss

(University of Queensland, Australia, 2010)

Motivation

Monte Carlo

Unbiased (i.e. "exact")

Error estimate

Easy to parallelize

Slow convergence, $N^{-1/2}$

Sign problem

Basis choices

Motivation

Monte Carlo	**Tensor Networks**
Unbiased (i.e. "exact") Error estimate Easy to parallelize	Converge rapidly Very precise Frustration, fermions, dynamics
Slow convergence, $N^{-1/2}$ Sign problem Basis choices	Bias (variational error) Require large bond-dimension No error estimate

Motivation

Monte Carlo	**Tensor Networks**
Unbiased (i.e. "exact") Error estimate Easy to parallelize	Converge rapidly Very precise Frustration, fermions, dynamics
Slow convergence, $N^{-1/2}$ Sign problem Basis choices	Bias (variational error) Require large bond-dimension No error estimate

Motivation

Monte Carlo

~~Unbiased (i.e. "exact")~~

~~Error estimate~~

Easy to parallelize

Difficult to optimize

Slow convergence, $N^{-1/2}$

~~Sign problem~~

Basis choices

Tensor Networks

~~Converge rapidly~~

~~Very precise~~

Frustration, fermions, ~~dynamics~~

Bias (variational error)

Require large bond-dimension

No error estimate

Motivation

Monte Carlo

~~Unbiased (i.e.~~

~~Error e~~

Easy to

Difficult

Slow conve

~~Sign prob~~

Basis choices

Tensor Networks

~~nverge rapidly~~

~~precise~~

, fermions,

~~mics~~

tional error)

large bond-
dimension

No error estimate

New approach

- Previously, tried to improve (variational) tensor networks using (variational) Monte Carlo to accelerate calculations.
- Now, the reverse: Do (unbiased) Monte Carlo and using ideas of how to do (direct) renormalization of tensor networks.

Monte Carlo

- Central idea is to sample a subset of a sum

$$\sum_i z_i$$

$$\text{error} = \sqrt{\frac{\text{Var}(z_i)}{N_{\text{samples}}}}$$

An example

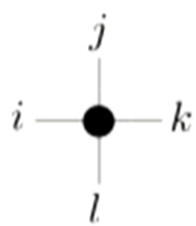
One such sum is the partition function of a classical statistical system

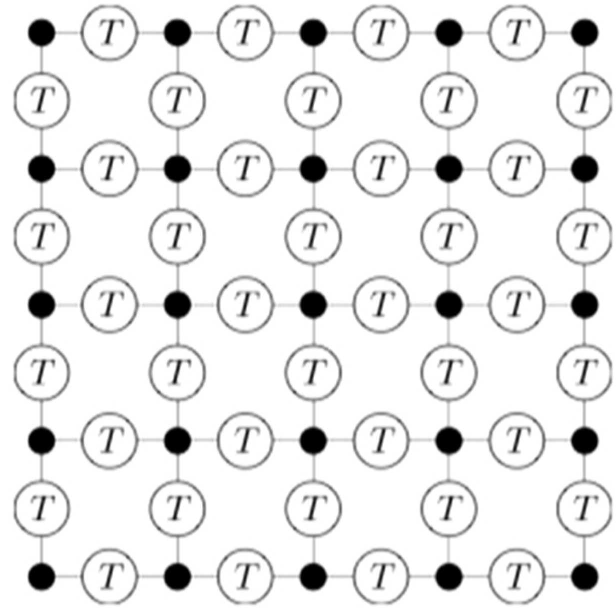
$$Z = \sum_{\mathbf{s}} e^{-\beta E(\mathbf{s})} \qquad\qquad (\beta = 1/k_B T)$$

Tensor network for Z

$$Z = \sum_{\mathbf{s}} \prod_{<i,j>} e^{-\beta E(s_i,s_j)}$$

$$T(s_1, s_2) = e^{-\beta E(s_i,s_j)}$$

 $= \quad \delta_{ijkl}$

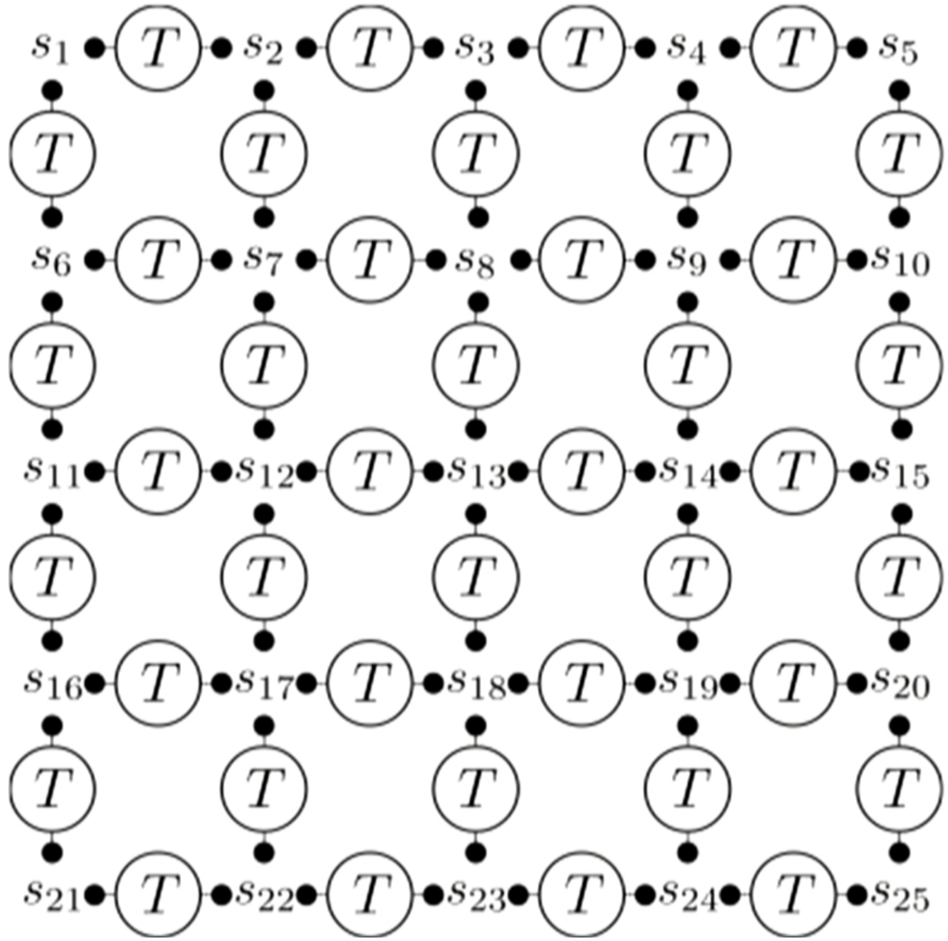


Configuration-based Monte Carlo

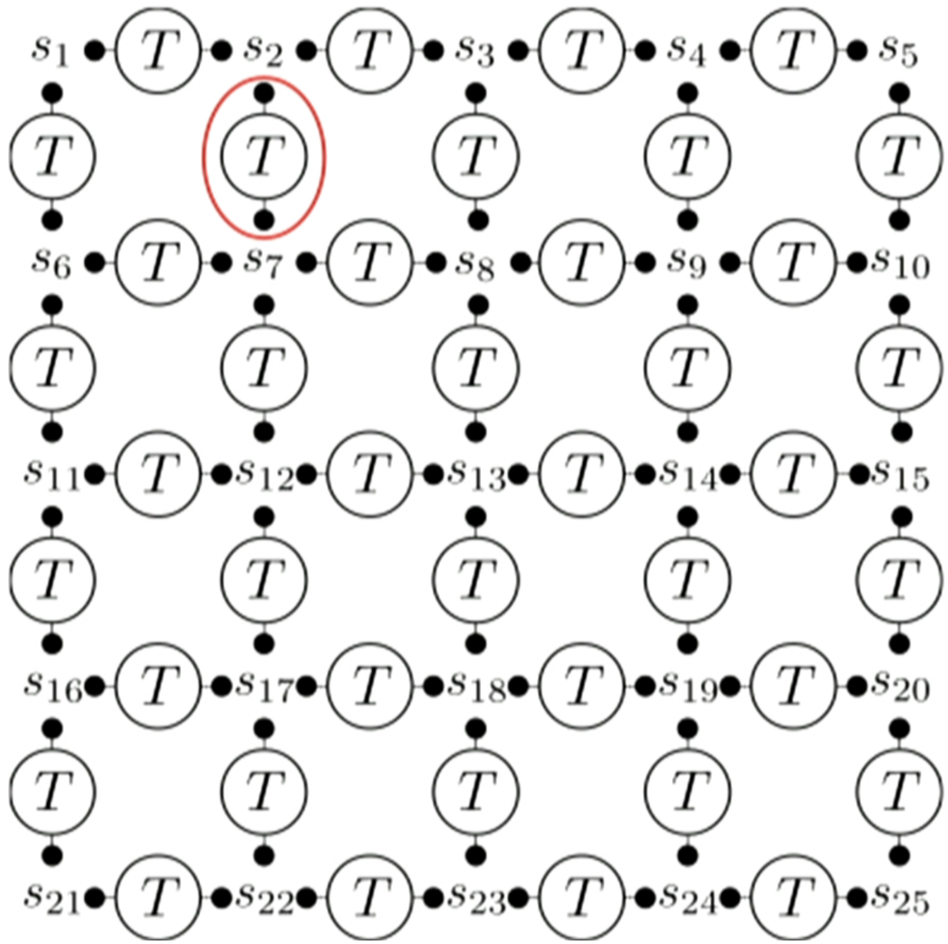
- Typically, the partition function is used to define the probability of a given configuration.
  - Importance sampling
  - Markov-chain algorithm for updating configurations
    - Metropolis algorithm, loop updates, cluster updates...
- From typical configurations we collect data for expectation values

$$\overline{E} = \frac{\sum_{\mathbf{s}} E(\mathbf{s}) p(\mathbf{s})}{\sum_{\mathbf{s}} p(\mathbf{s})}$$

Tensor network of a configuration



Tensor network of a configuration



Accuracy vs. number of samples

$$\Delta E = \sqrt{\frac{\mathrm{Var}(E)}{N_{\mathrm{samples}}}}$$

Accuracy vs. number of samples

$$\Delta E = \sqrt{\frac{\mathrm{Var}(E)}{N_{\mathrm{samples}}}}$$

Number of **independent** samples

- cluster updates
- loop updates
- worm algorithm
- etc...

Accuracy vs. number of samples

$$\Delta E = \sqrt{\frac{\mathrm{Var}(E)}{N_{\mathrm{samples}}}}$$

Number of **independent** samples

- cluster updates
- loop updates
- worm algorithm
- etc…

Accuracy vs. number of samples

$$\Delta E = \sqrt{\frac{\mathrm{Var}(E)}{N_{\mathrm{samples}}}}$$

Sample-to-sample variance

- importance sampling
- partial summation

Number of **independent** samples

- cluster updates
- loop updates
- worm algorithm
- etc…

Tensor Network Monte Carlo

New idea: Perform multi-sampling.

For each bond in the tensor network we keep some subset of $D > 1$ indices (and discard the remainder).

To do this efficiently, ideas from TN renormalization will have to be employed

Tensor Network Monte Carlo

New idea: Perform multi-sampling.

For each bond in the tensor network we keep some subset of $D > 1$ indices (and discard the remainder).

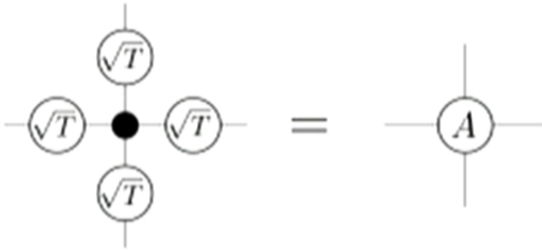
To do this efficiently, ideas from TN renormalization will have to be employed

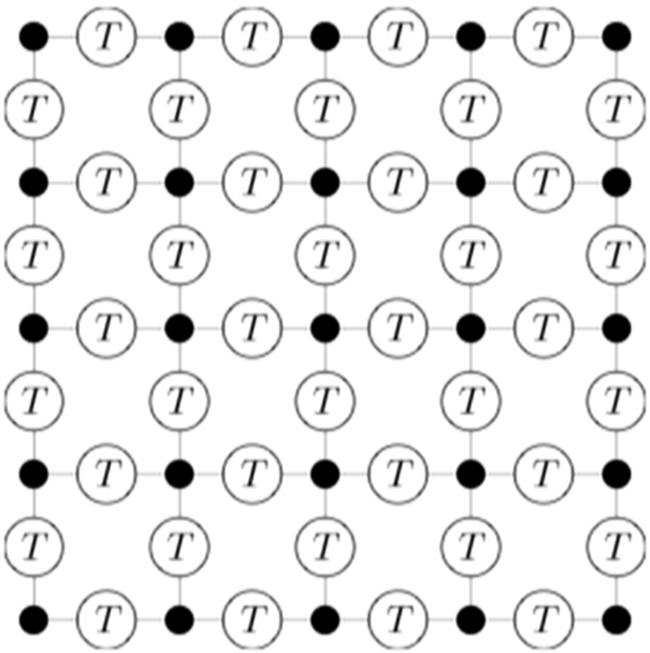
Renormalization

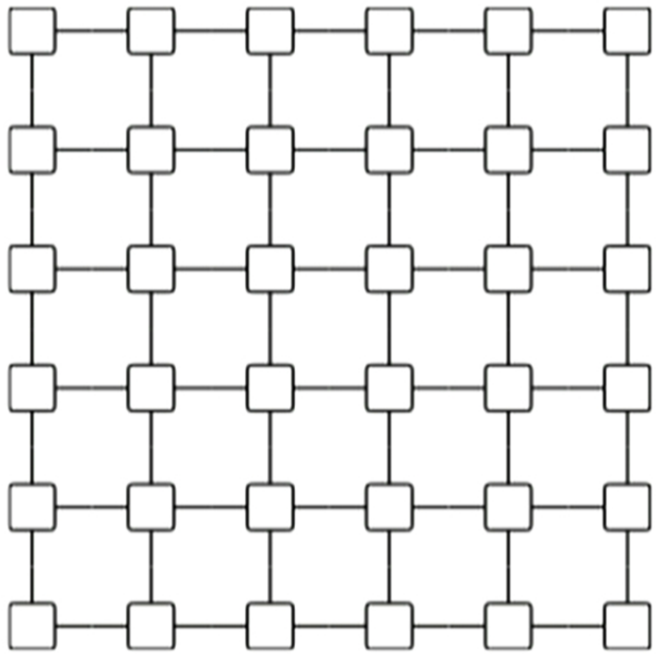
- "Renormalization" is a word that means many different things
  - – Removing divergences from, e.g., QED
  - – Momentum-space renormalization (e.g. Wilson)
  - – **Real-space renormalization** (e.g. Kadanoff, etc…)
- Here, renormalization means approximating one or more tensors as a simpler tensor (with lower bond dimension) – "*BLOCKING*"

Blocking schemes

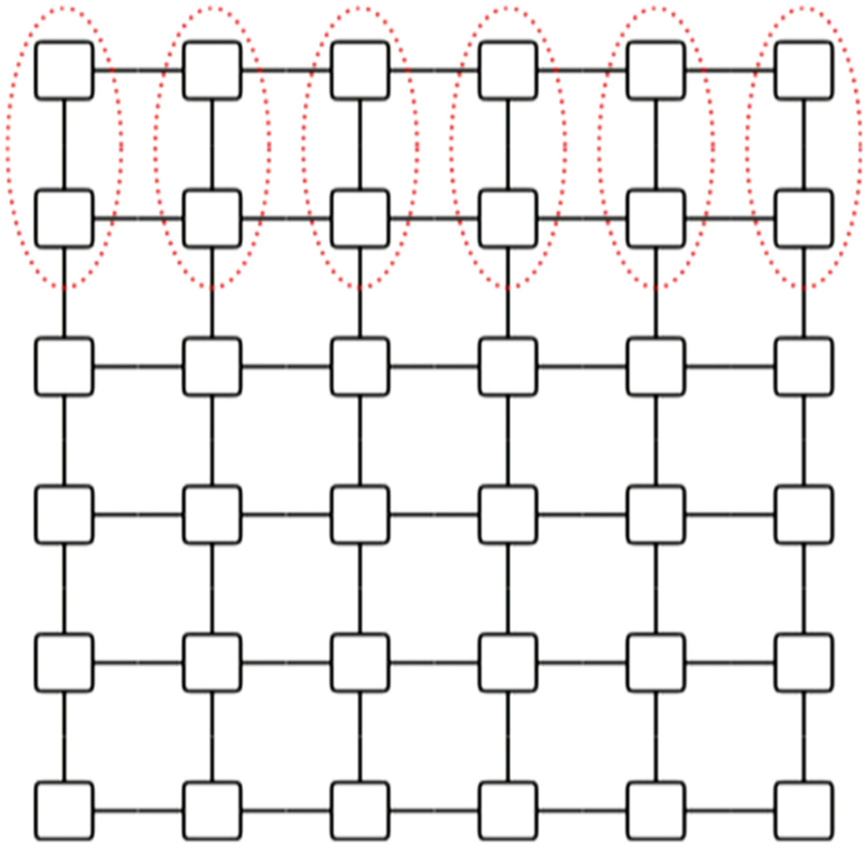
First rewrite the tensors



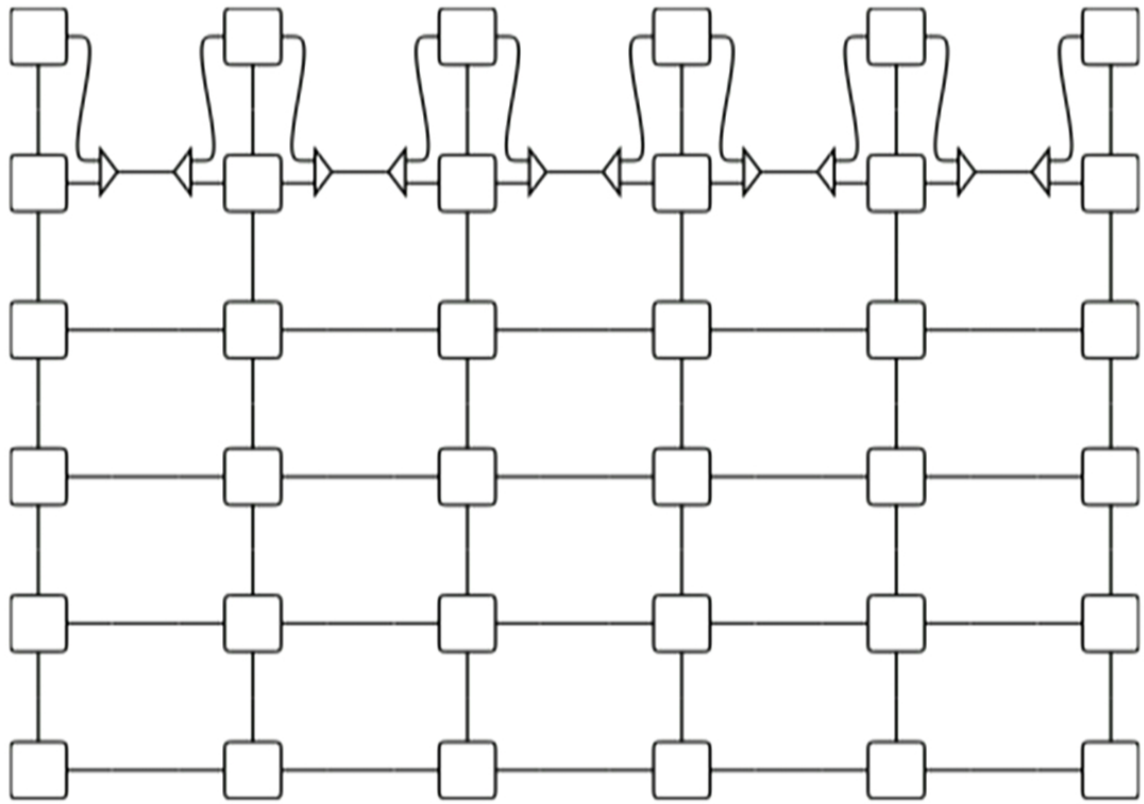




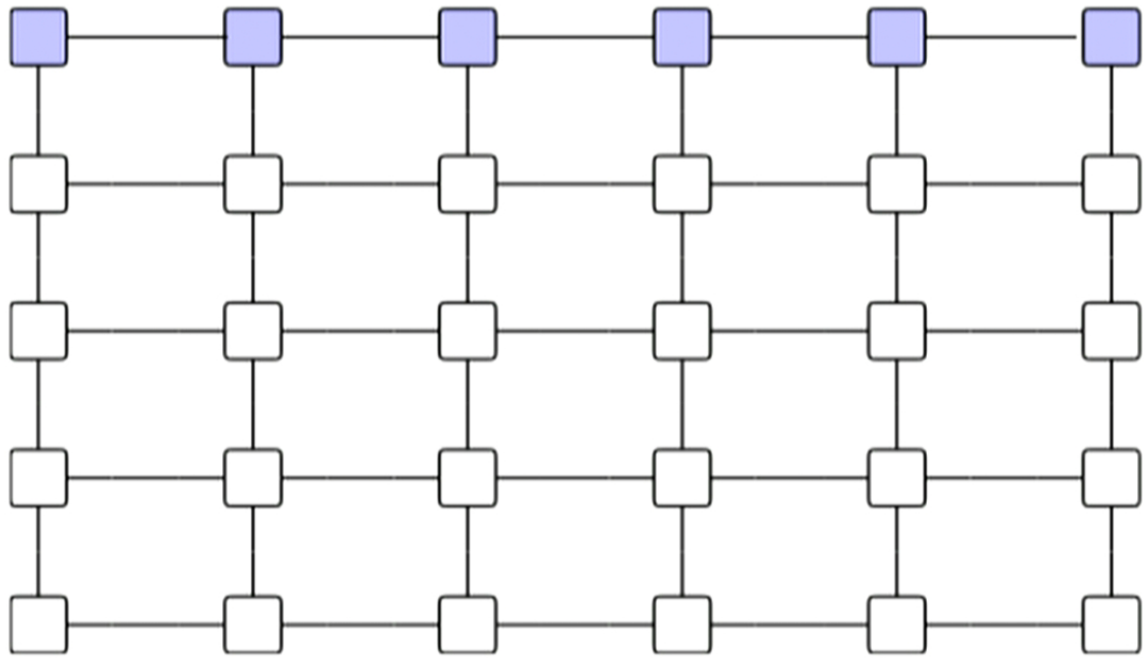
Projecting bonds



Projecting bonds

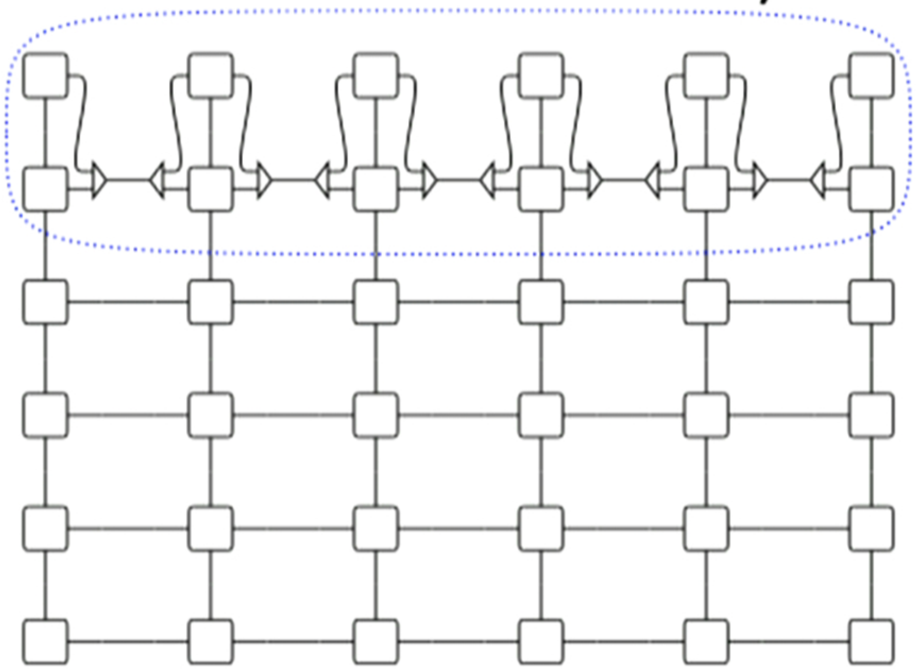


Projecting bonds



Cost function

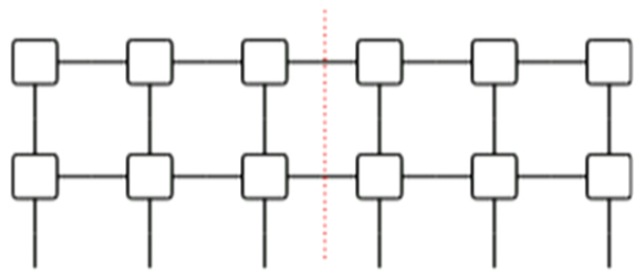
- The error of the projection can be quantified using the 2-norm of the “boundary” state



Optimal truncation

$$\text{Error} = \Big\| |\text{MPS}\rangle - |\text{MPS}'\rangle \Big\|_2^2$$

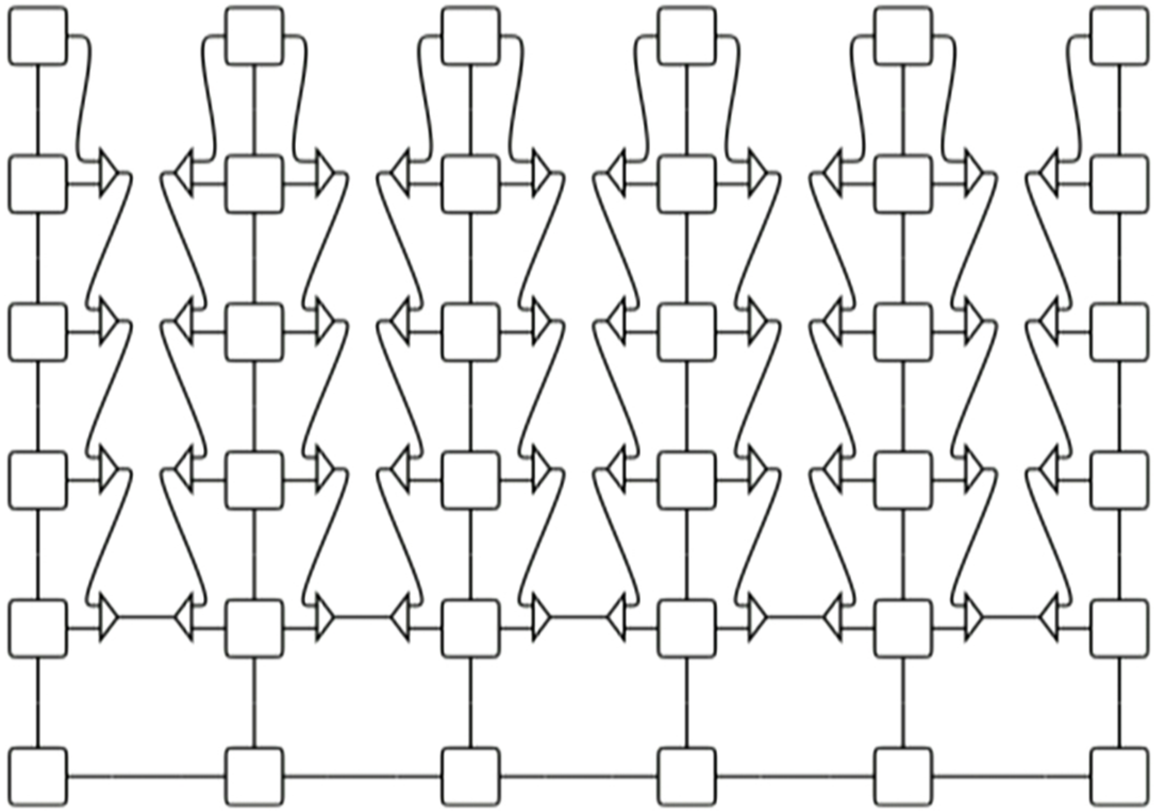
$$|\text{MPS}\rangle = \sum_i S_i |L_i\rangle |R_i\rangle$$



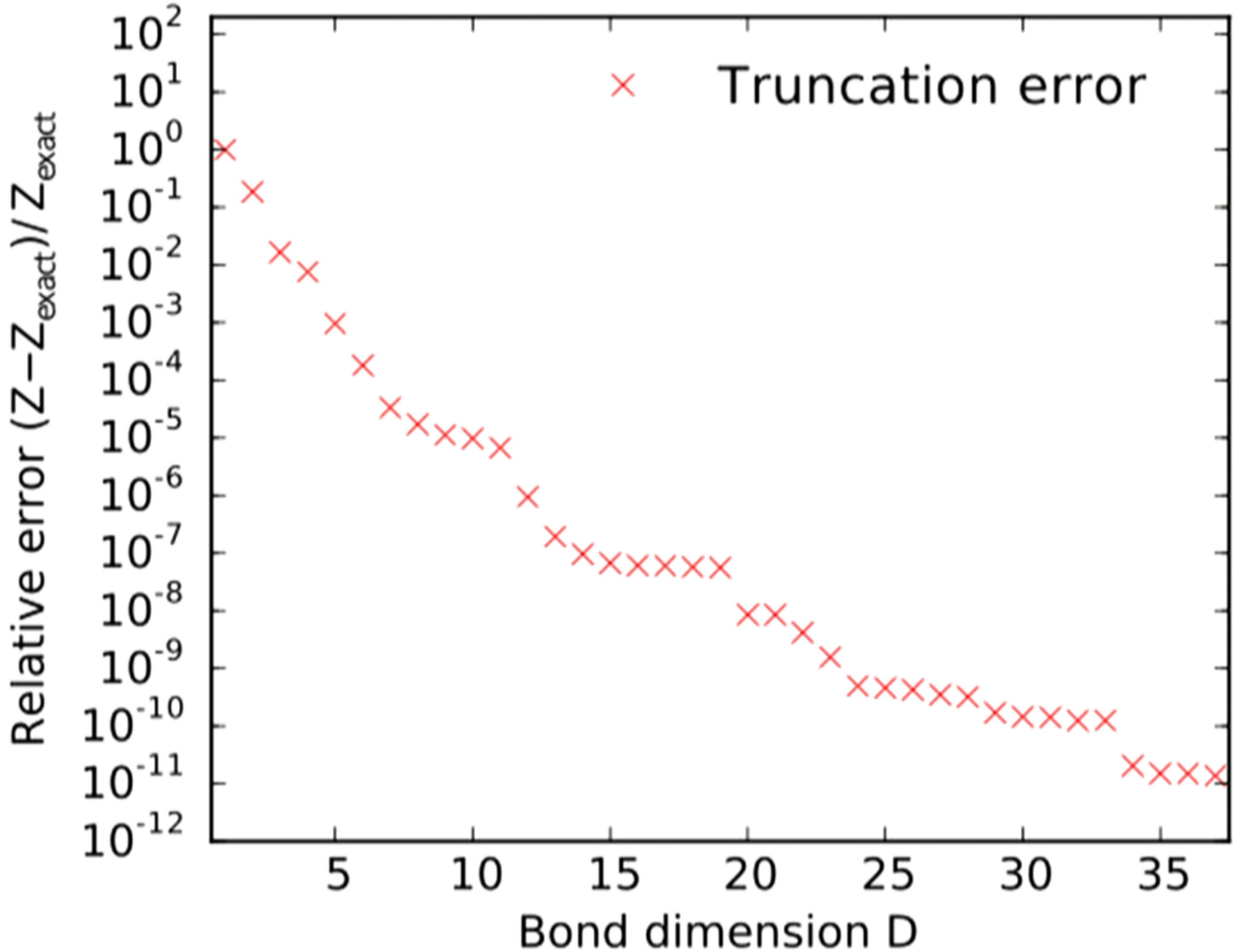
Keep biggest Schmidt coefficients

$$\text{Error} = \sum_{i>D} S_i^2$$

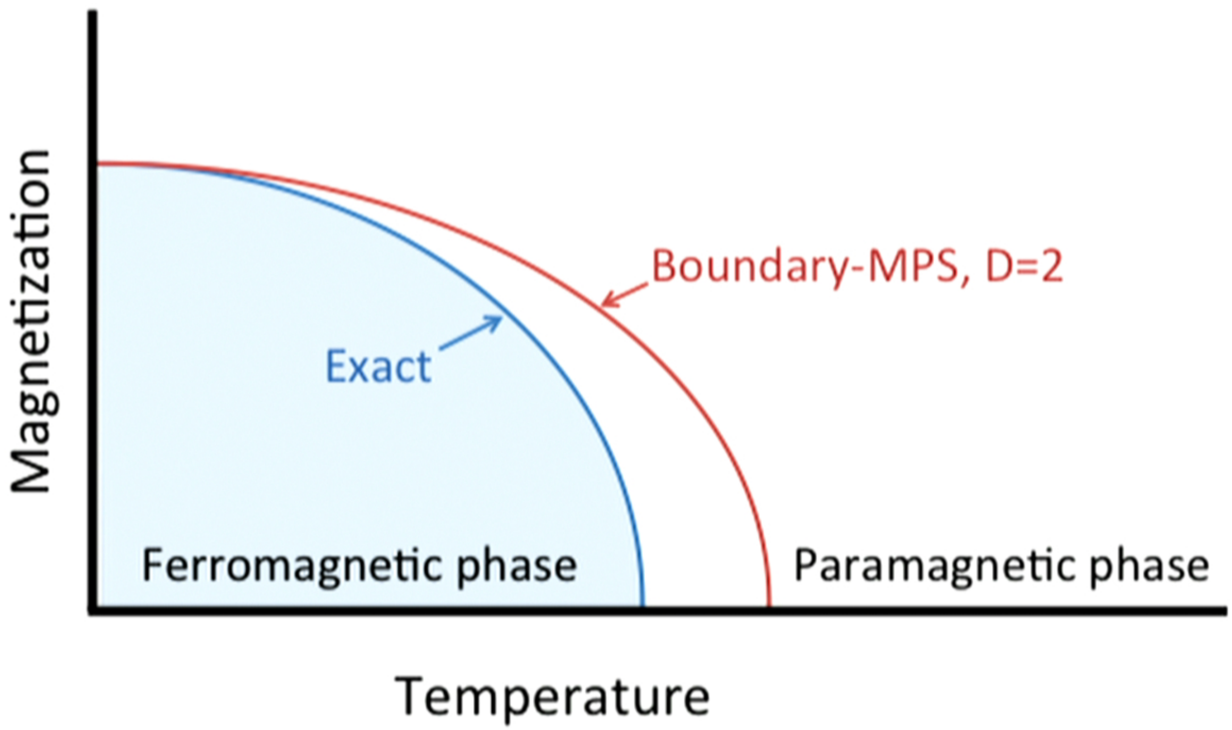
Boundary-MPS renormalization



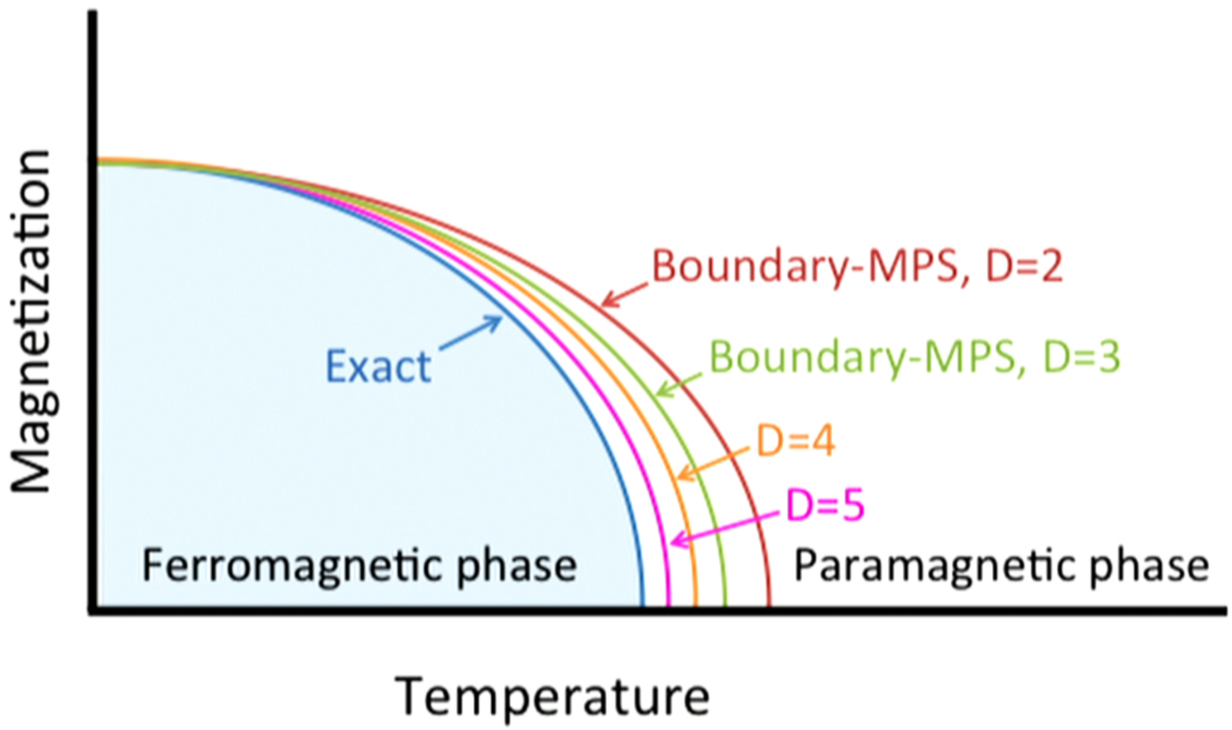
Accuracy vs. bond dimension



Downside: Variational error



Downside: Variational error



Tensor Network Monte Carlo

CORE IDEA

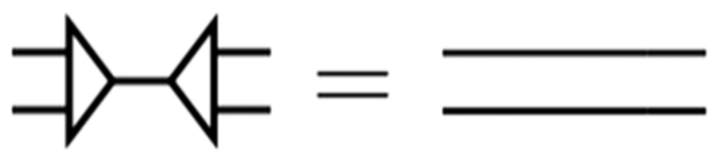
Randomly select which subspace to keep during truncation steps using Monte Carlo

Tensor network Monte Carlo

For *unbiased* Monte Carlo, we need that the result converges exactly in the limit of large number of samples.

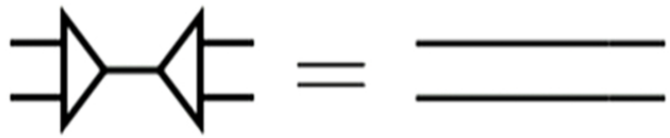
On average, the projectors should do nothing:

$$\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}$$



Subspace selection

$$\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}$$



There are many such possibilities. We want to find one that minimizes the *expectation value* of the error.

$$\mathrm{Error} = \left\langle \Big\| |\mathrm{MPS}\rangle - |\mathrm{MPS}'\rangle \Big\|_2^2 \right\rangle$$

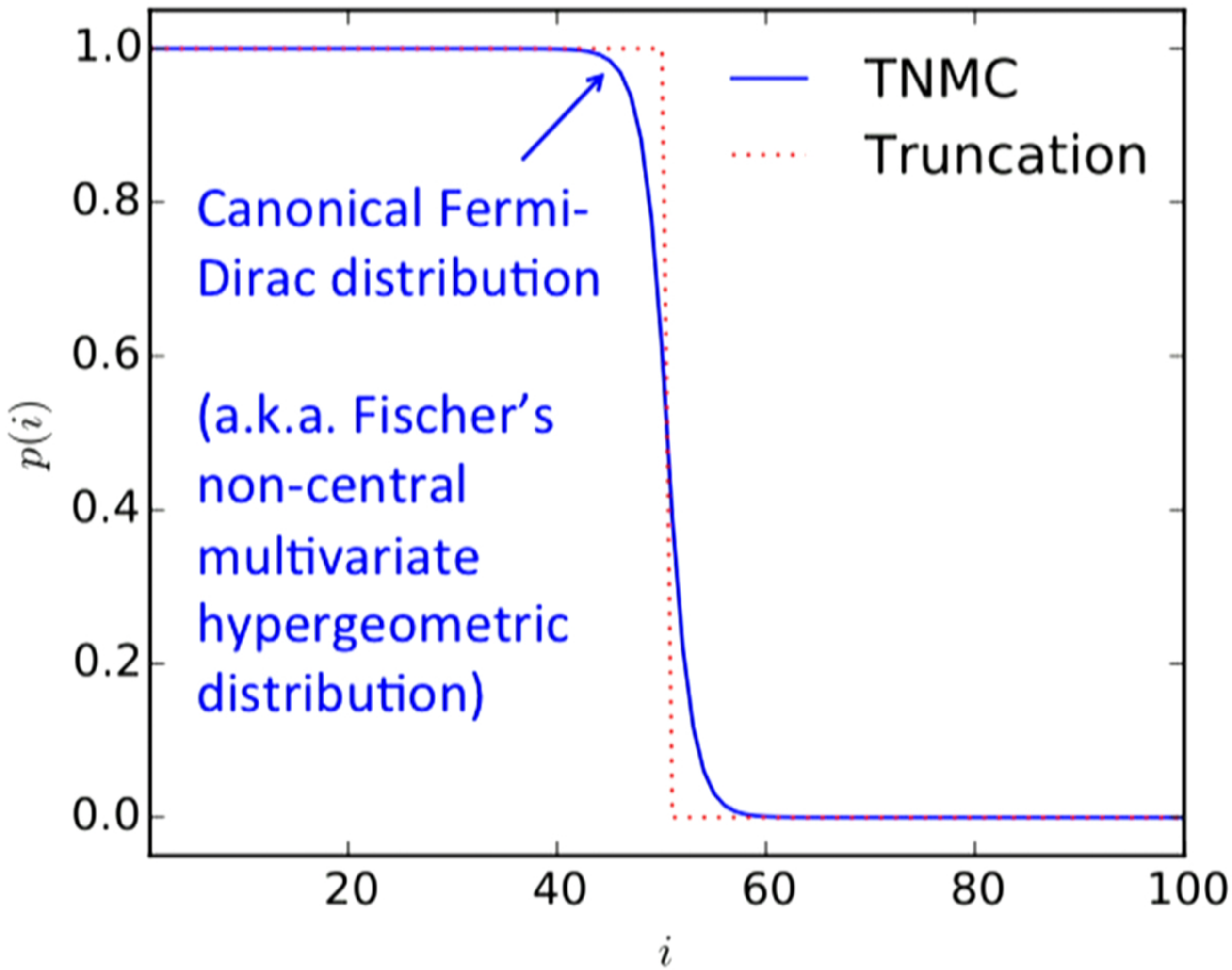
$$\sim \left\langle \sum_{i\notin s} S_i^2 \right\rangle$$

Subspace selection

- Select large singular values more often, small one less often.
- Never select the same index twice.
- Multi-sampling
  - Probability of sampling a collection is the product of the probability of the individual parts

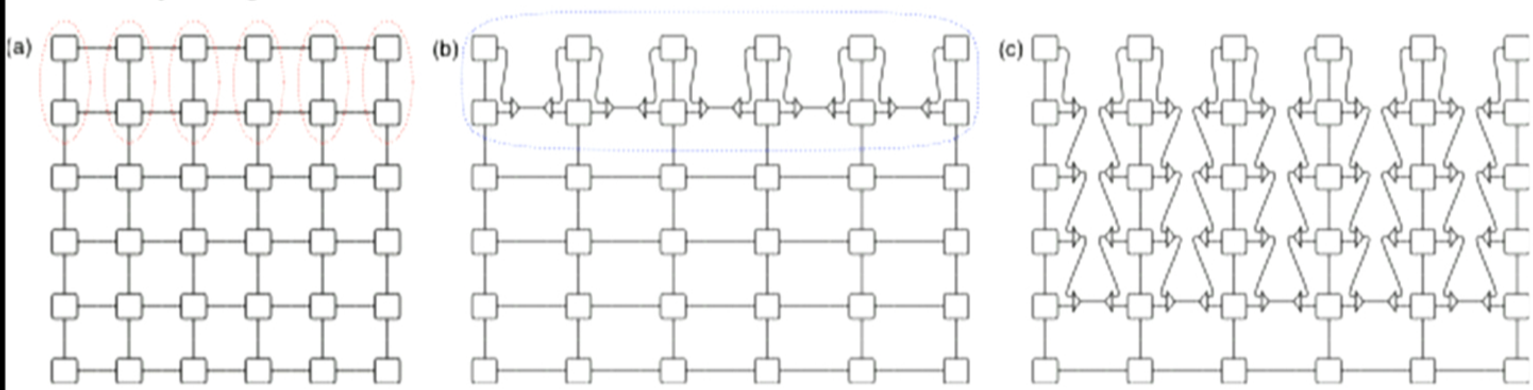
$$p(i_1, i_2) = p(i_1)p(i_2) \qquad (i_1 \neq i_2)$$

Subspace selection

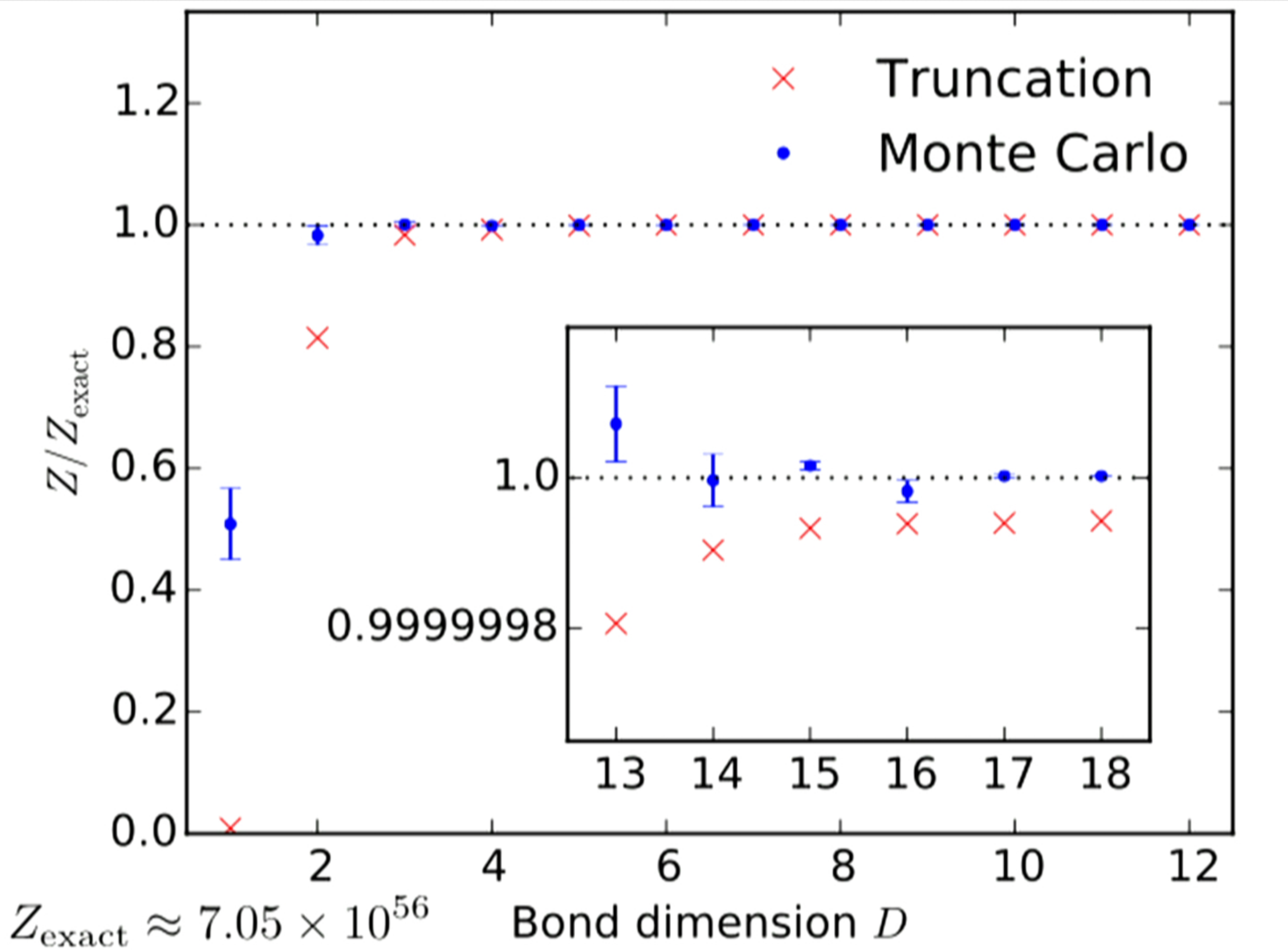


“Perfect” sampling procedure

- In each full-system sample, we calculate the partition function with randomly selected projectors.



- The results for Z are averaged over many, fully-independent samples


Truncation
Monte Carlo
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Z/Z_{exact}
1.0
0.9999998
13
14
15
16
17
18
2
4
6
8
10
12
$Z_{\mathrm{exact}} \approx 7.05 \times 10^{56}$
Bond dimension D

Observation #1

Tensor network Monte Carlo inherits the accuracy of tensor network methods.

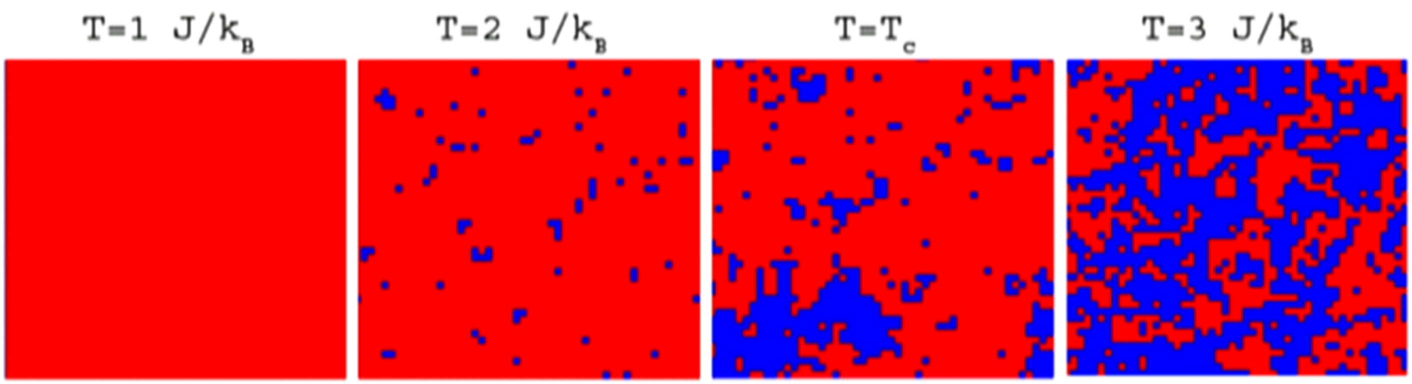
Unprecedented small sample-to-sample variance.

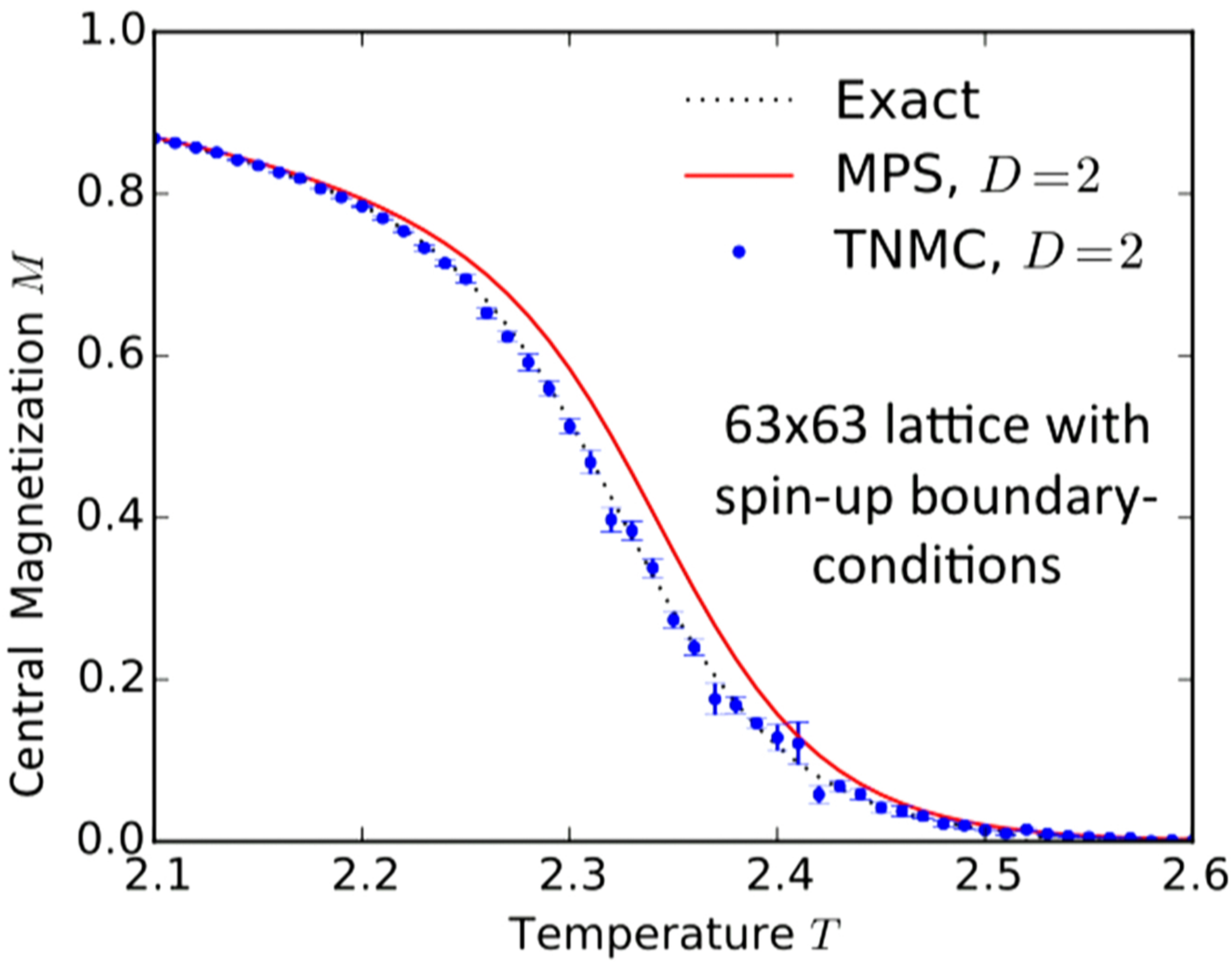
Accuracy vs. cost improves more rapidly than $N^{-1/2}$ of Monte Carlo

Results: 2D classical Ising Model

$$E(\mathbf{s}) = -J \sum_{<i,j>} s_i s_j$$

$$s_i \in \{+1, -1\}$$





1.0
0.8
0.6
0.4
0.2
0.0
2.1
2.2
2.3
2.4
2.5
2.6
Central Magnetization M
Temperature T
Exact
MPS, $D = 2$
TNMC, $D = 2$
63x63 lattice with spin-up boundary-conditions

Observation #2

Tensor network Monte Carlo is an unbiased Monte Carlo method

None of the variational bias of tensor network methods

Sample-to-sample variance is greater where the variational technique struggles

Markov Chain Sampling

- Shown results are for small systems
- For “perfect” sampling, need each sample to account for majority of Z to obtain a good projection basis and well-behaved sampling
  - Bond dimension should increase with system size
- Markov-Chain sampling will overcome this limitation
  - E.g. standard Metropolis algorithm accounts for a tiny fraction of Z each sample

Generalizations

- TNMC sampling can be applied to other tensor renormalization schemes
  - MPS (2D classical), PEPS/TNS (3D classical...)
  - TRG and HOTRG (2D, 3D...)
  - etc...

- *d*-dimensional quantum systems can be represented by *(d+1)*-dimensional partition functions called path integrals
  - Also, Projector MC for zero-temperature q. systems

Sign problem

- Quantum systems can have sign problem for Monte Carlo
  - Path integral may have negative signs, or phases
- TNMC naturally samples large parts of the path integral, summing over positive and negative terms within each sample
  - Sign-problem resistant? (with sufficient D)
  - LTRG has been demonstrated to work...

Correct hybrid method

Monte Carlo	**Tensor Networks**
Unbiased (i.e. "exact") Error estimate Easy to parallelize	Converge rapidly Very precise ? Frustration, fermions, ? ?? dynamics ??
~~Slow convergence,~~ $N^{-1/2}$?? Sign problem ?? ~~Basis choices~~	~~Bias (variational error)~~ ~~Require large bond-dimension~~ ~~No error estimate~~

Conclusions

Tensor Network MC may appeal to:

- Monte Carlo community for faster and more accurate Monte Carlo
- Tensor network community for zero-bias calculations and error estimation