Title: Introduction to Tensor Network methods - 3

Date: Jun 20, 2017 11:00 AM

URL: http://pirsa.org/17060070

Abstract:

LECTURES 3 and 4

Tensor Renormalization Group (TRG)

TRG as a computational scheme

- TRG Algorithm
- Non-critical systems
- critical systems
- D=3 dimensions
- Transfer matrix

TRG and the Renormalization Group flow

- The Renormalization Group
- TRG flow
- Tensor Network Renormalization (TNR)

Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

partition function

Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

partition function

local Boltzmann weights $W^{(ij)} \equiv e^{-\beta H^{(ij)}}$

Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

partition function

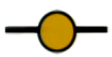
local Boltzmann weights $W^{(ij)} \equiv e^{-\beta H^{(ij)}}$

(1) define matrix

$$W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$$

(Ising model)

W



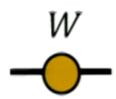
Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

partition function

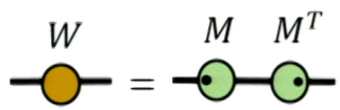
local Boltzmann weights $W^{(ij)} \equiv e^{-\beta H^{(ij)}}$

(1) define matrix $W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$ (Ising model)



(2) decompose

$$W = MM^T$$



$$M = \begin{pmatrix} \cosh\beta & \sinh\beta \\ \cosh\beta & -\sinh\beta \end{pmatrix}$$

(Ising model)

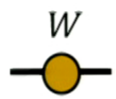
Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

partition function

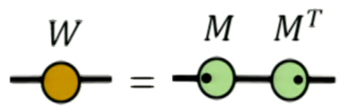
local Boltzmann weights $W^{(ij)} \equiv e^{-\beta H^{(ij)}}$

(1) define matrix $W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$ (Ising model)



(2) decompose

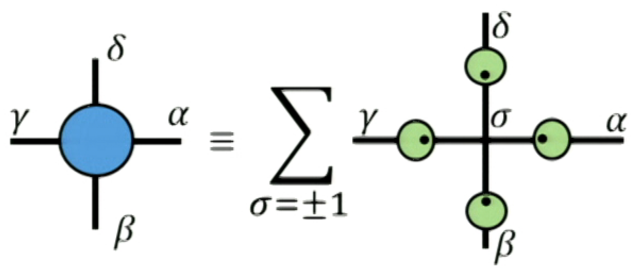
$W = MM^T$



$M = \begin{pmatrix} \cosh\beta & \sinh\beta \\ \cosh\beta & -\sinh\beta \end{pmatrix}$

(Ising model)

(3) define tensor A with components $A_{\alpha\beta\gamma\delta}$

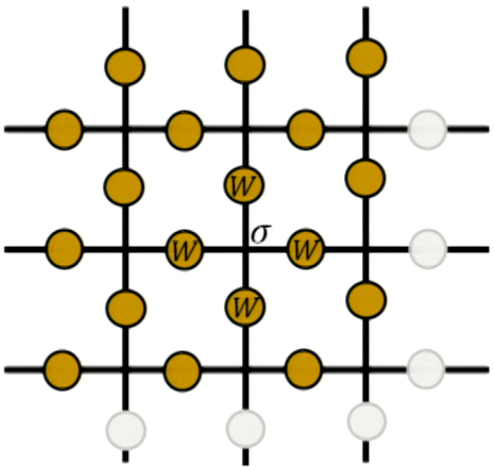


Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

$N = 3 \times 3 = 9$ spins

$$W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$$

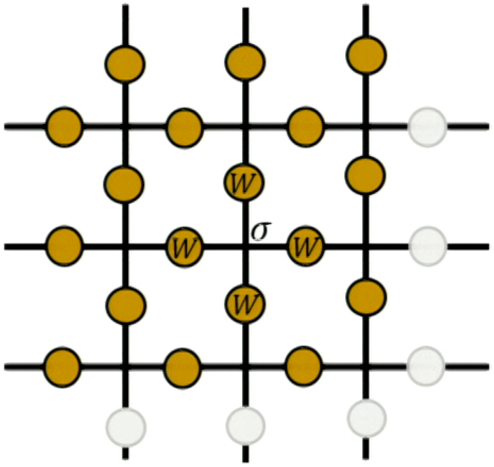
$Z =$ 

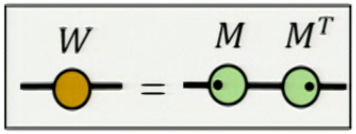
Partition function as a tensor network

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

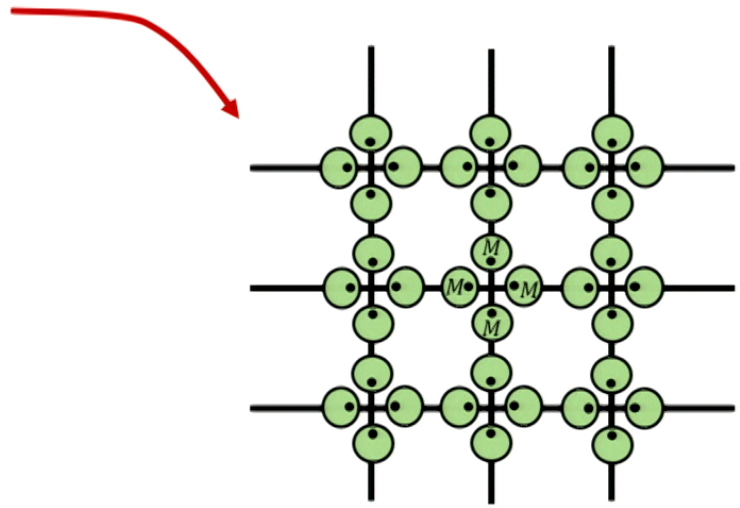
$N = 3 \times 3 = 9$ spins

$$W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$$

$Z =$ 



$$M = \begin{pmatrix} \cosh\beta & \sinh\beta \\ \cosh\beta & -\sinh\beta \end{pmatrix}$$



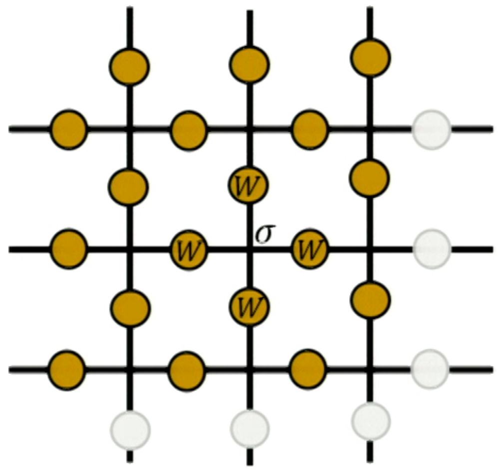
Partition function as a tensor network

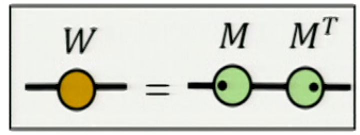
$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

$N = 3 \times 3 = 9$ spins

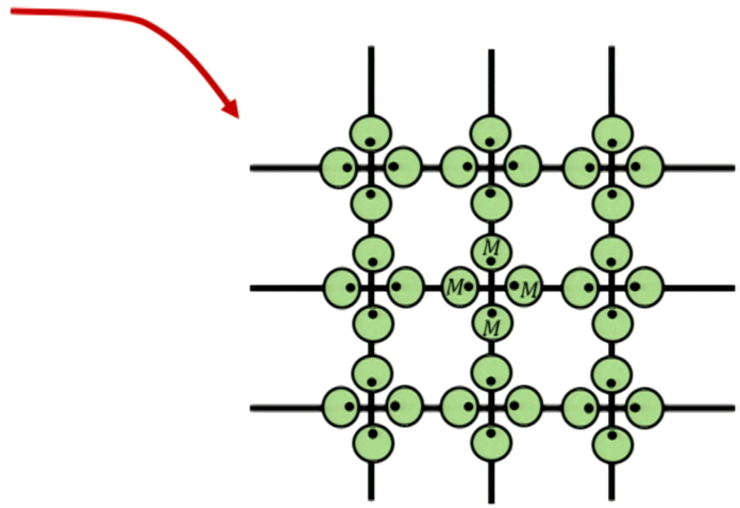
$$W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$$

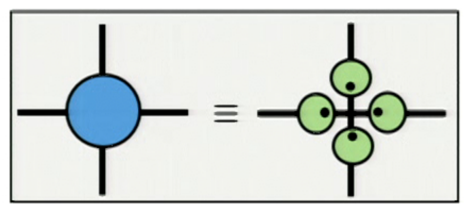
$Z =$





$$M = \begin{pmatrix} \cosh\beta & \sinh\beta \\ \cosh\beta & -\sinh\beta \end{pmatrix}$$





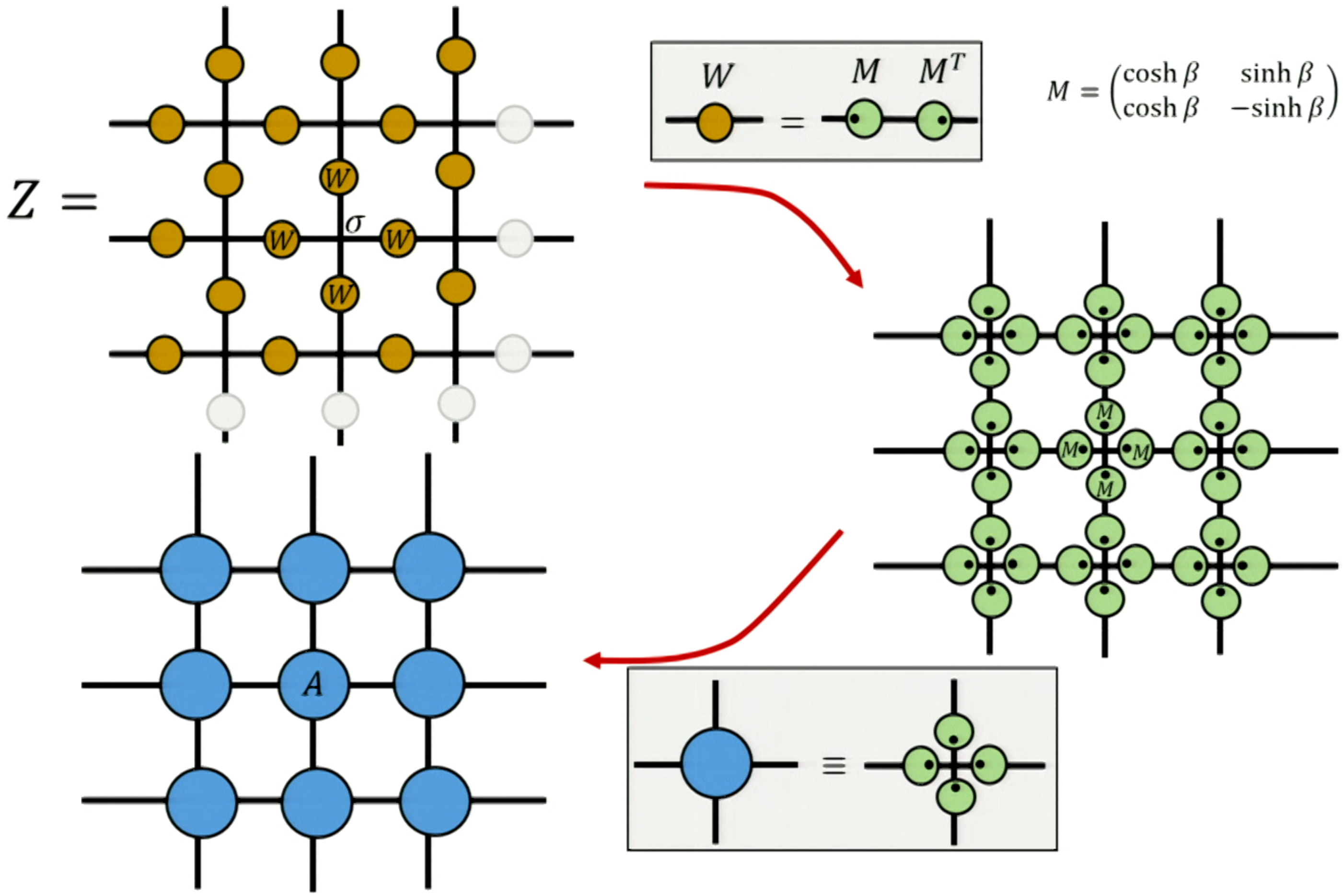
Partition function as a tensor network

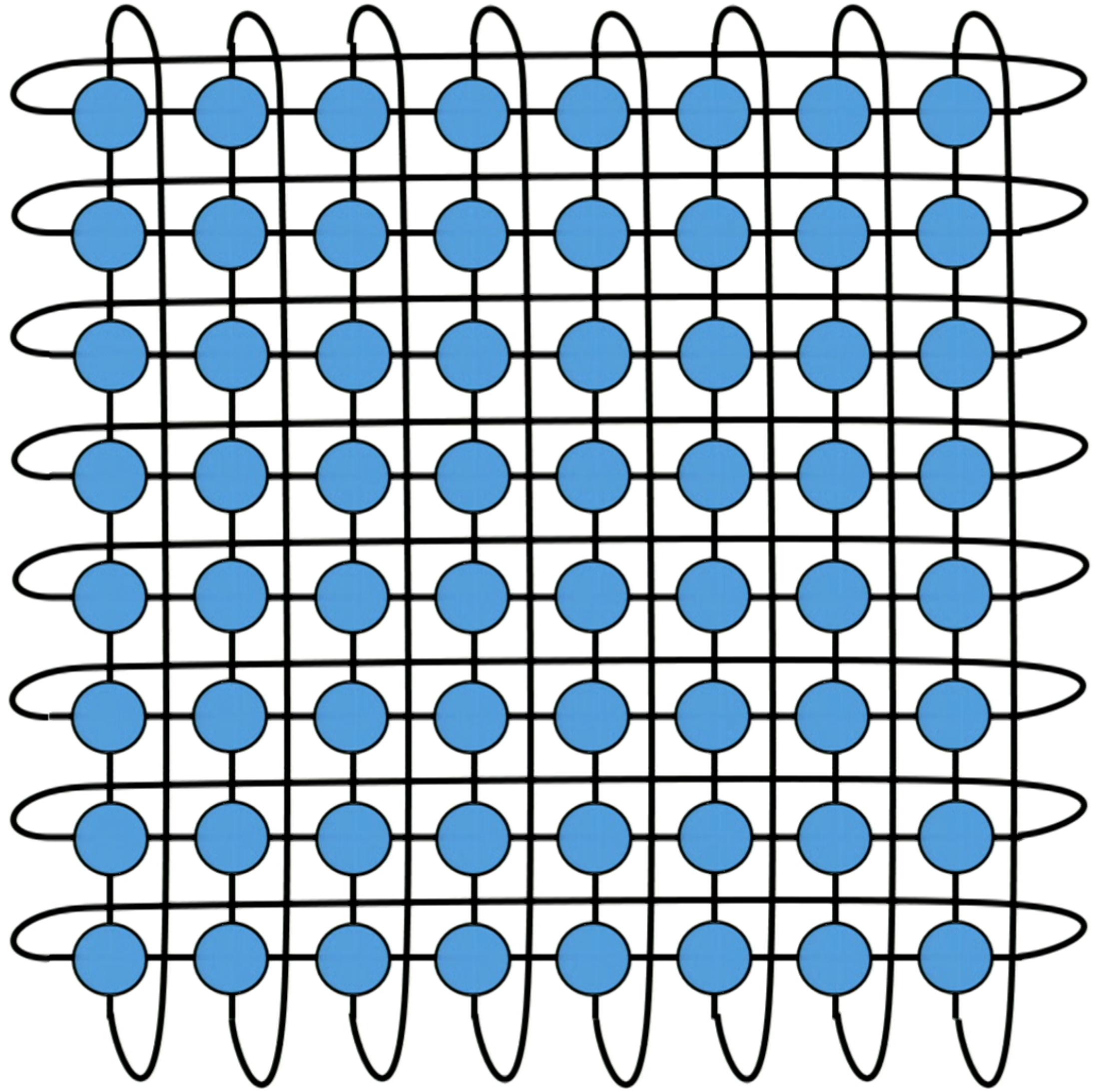
$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

$N = 3 \times 3 = 9$ spins

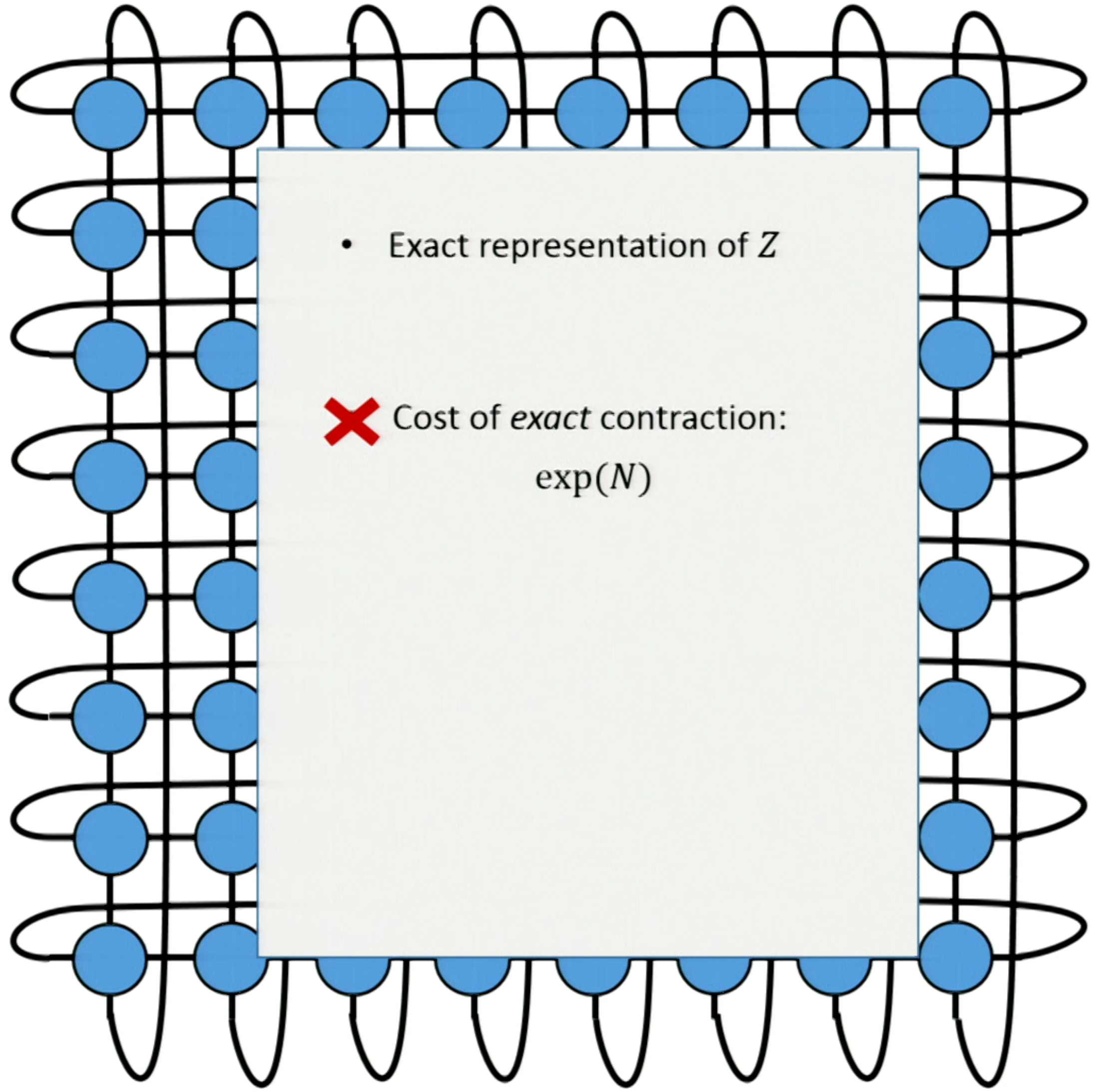
$$W = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}$$

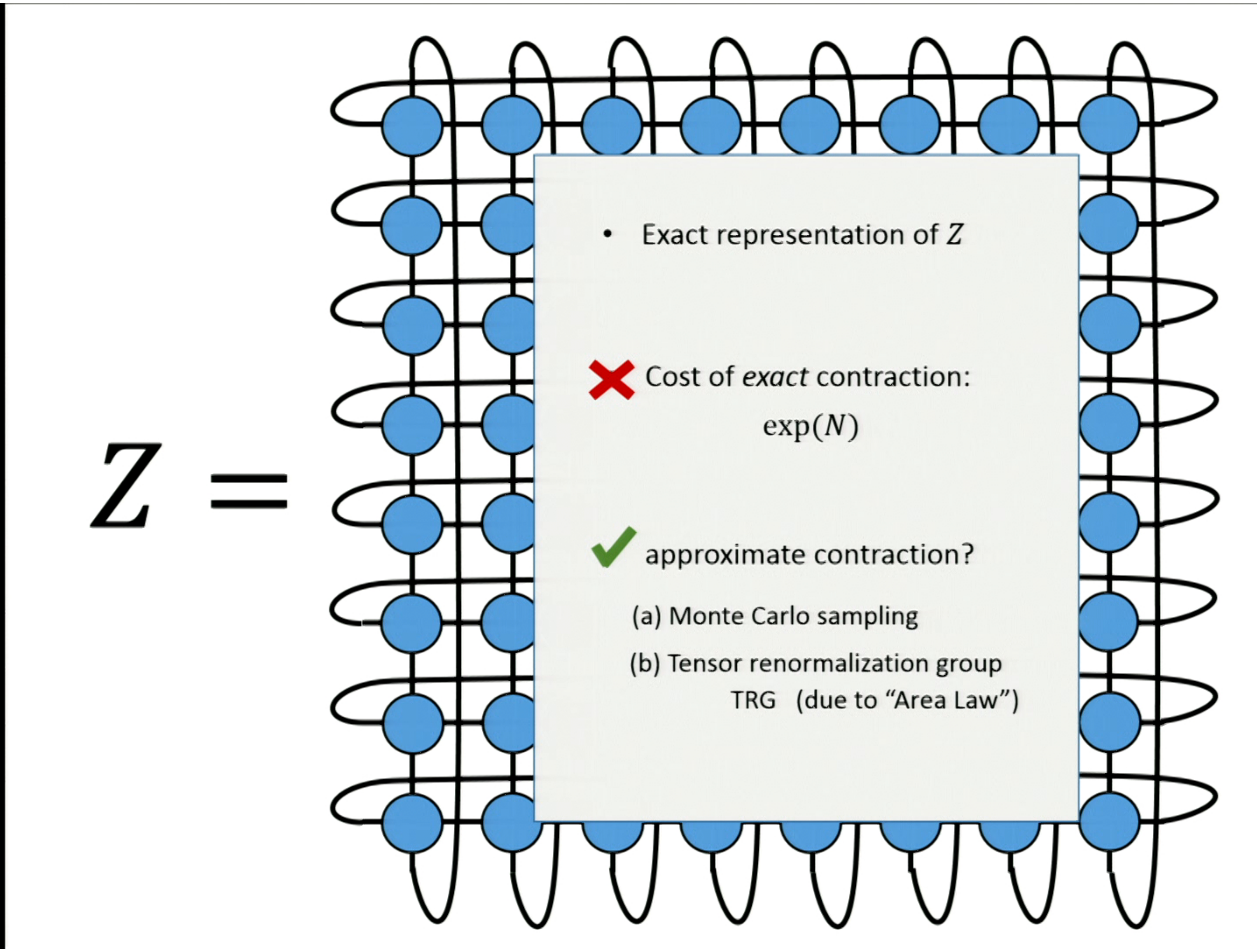
$$M = \begin{pmatrix} \cosh\beta & \sinh\beta \\ \cosh\beta & -\sinh\beta \end{pmatrix}$$

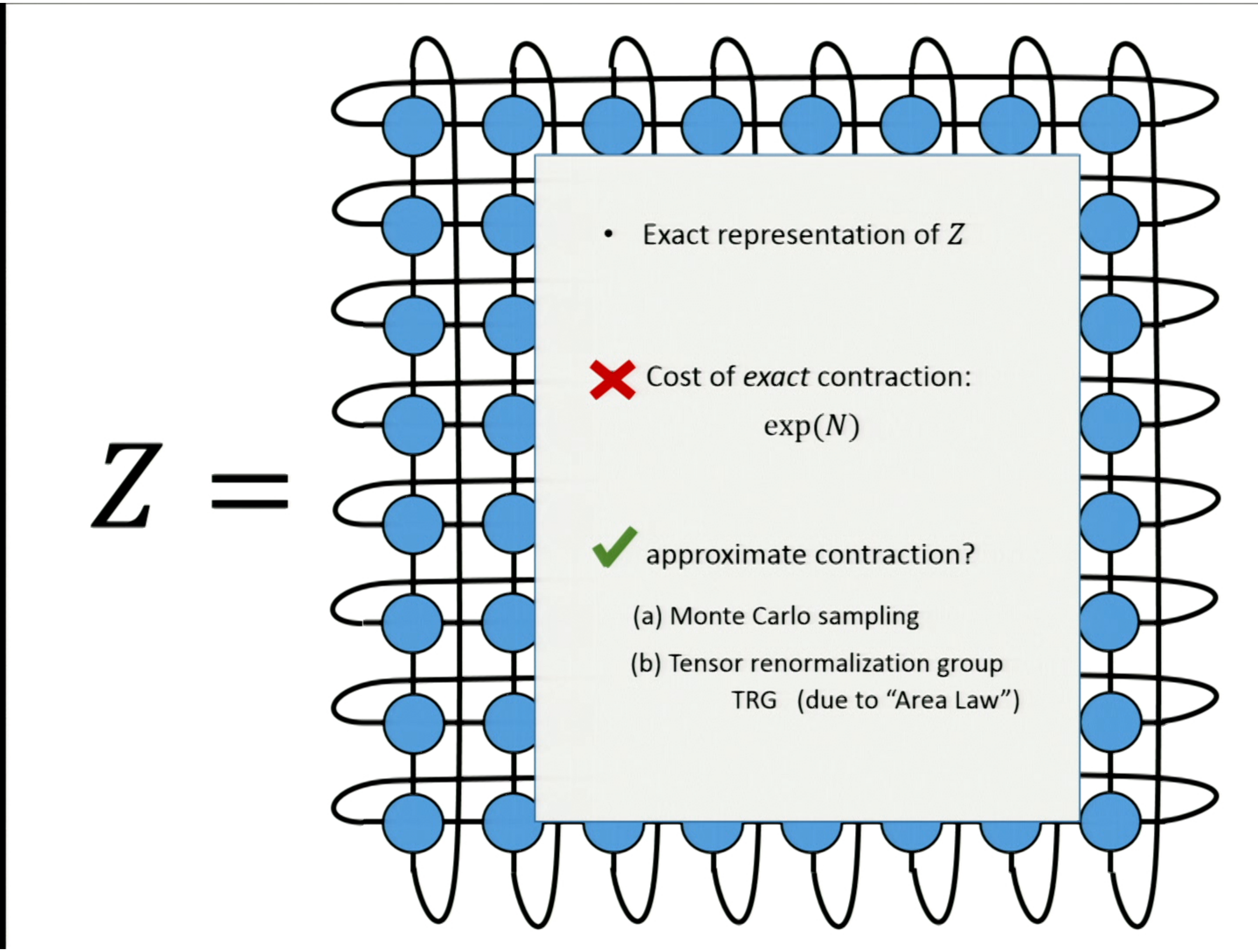


$Z =$ 

$$Z =$$

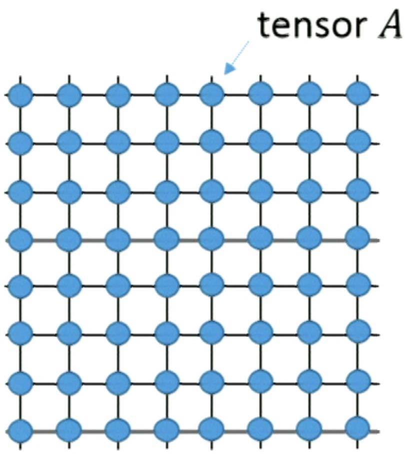



$Z =$
• Exact representation of Z
Cost of *exact* contraction:
$\exp(N)$
approximate contraction?
(a) Monte Carlo sampling
(b) Tensor renormalization group
TRG (due to "Area Law")


$Z =$
• Exact representation of Z
Cost of *exact* contraction:
$\exp(N)$
approximate contraction?
(a) Monte Carlo sampling
(b) Tensor renormalization group
TRG (due to "Area Law")

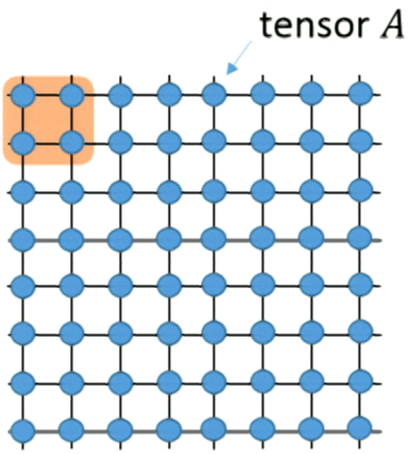
simplify by coarse-graining

general strategy:



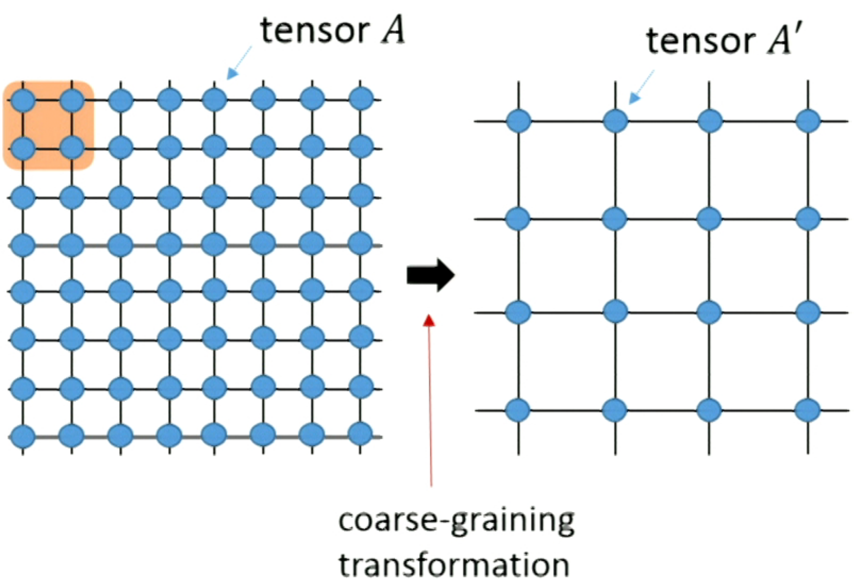
simplify by coarse-graining

general strategy:



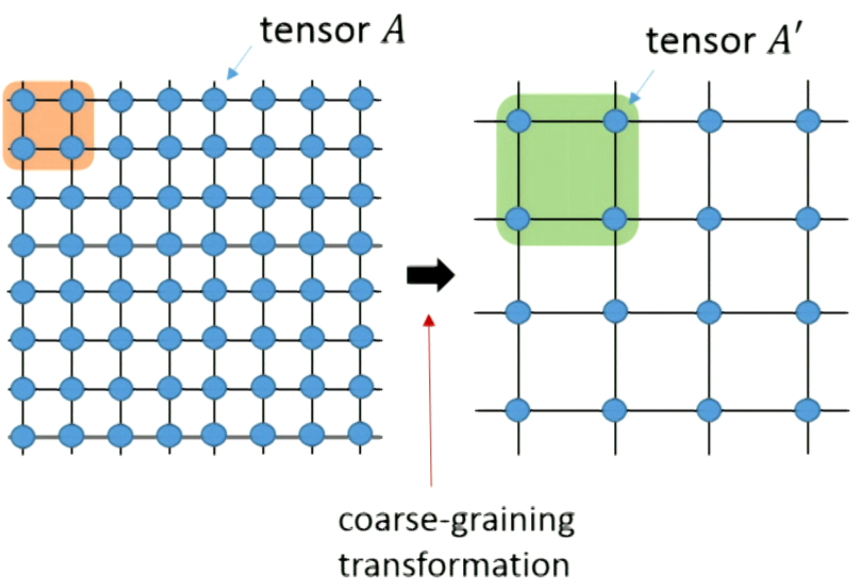
simplify by coarse-graining

general strategy:



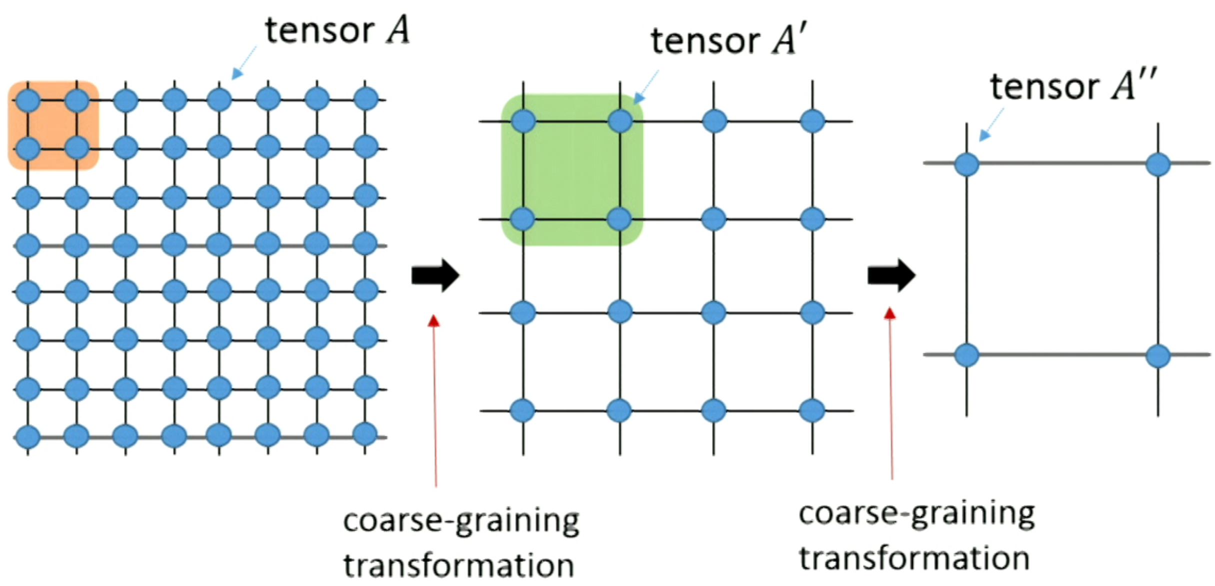
simplify by coarse-graining

general strategy:



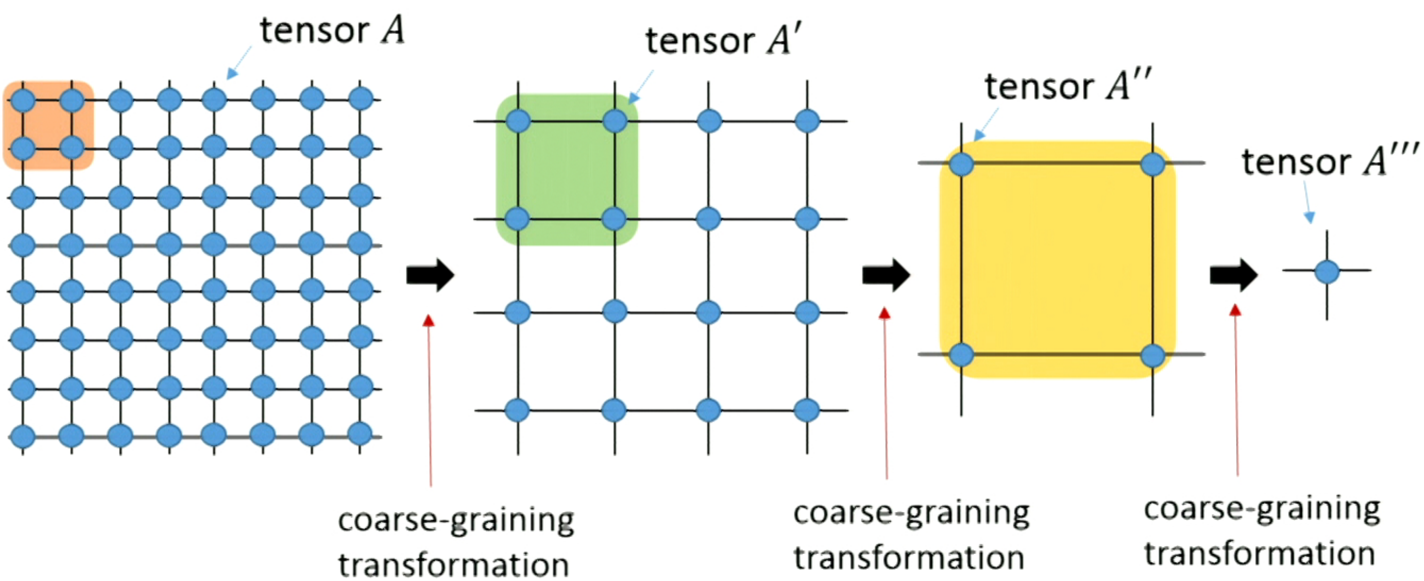
simplify by coarse-graining

general strategy:



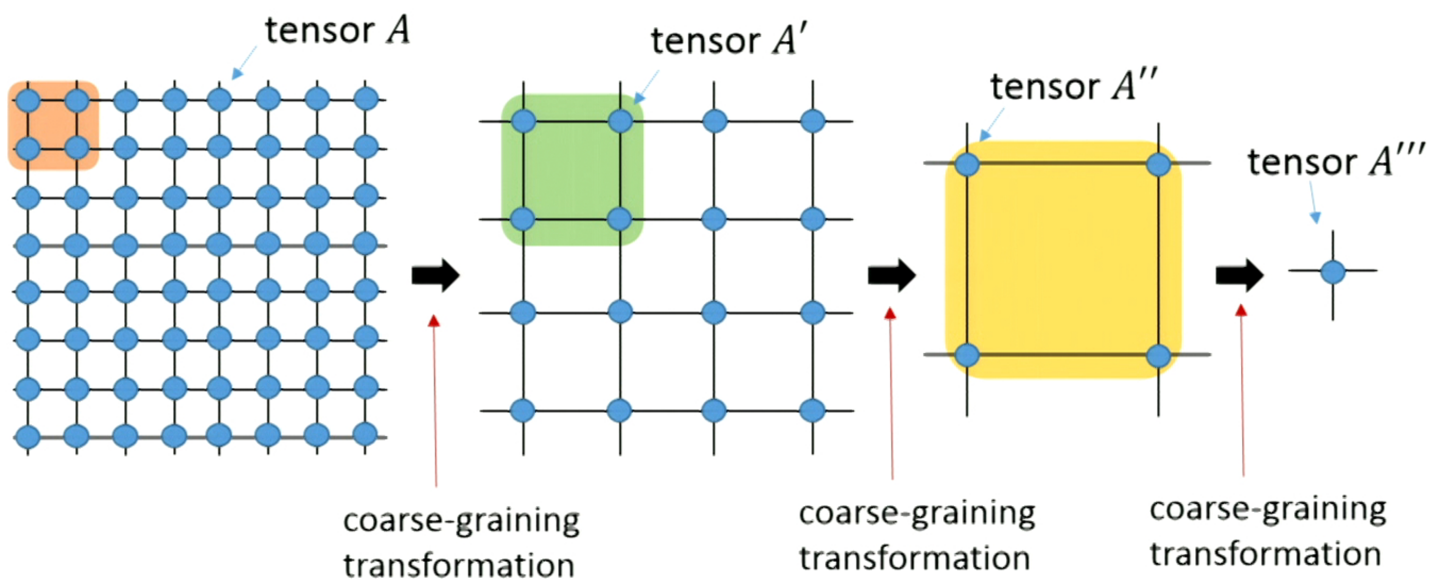
simplify by coarse-graining

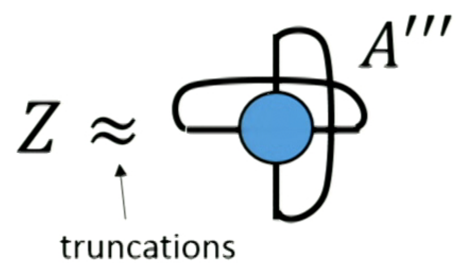
general strategy:



simplify by coarse-graining

general strategy:





Tensor Renormalization Group (TRG) Levin, Nave, 2006

2 simple steps

Tensor Renormalization Group (TRG)

Levin, Nave, 2006

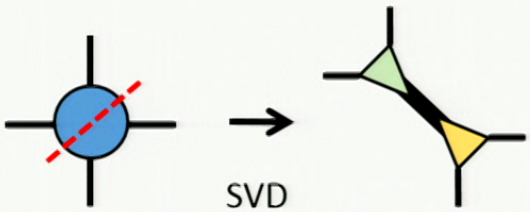
2 simple steps

STEP 1: SPLIT tensors

Tensor Renormalization Group (TRG) Levin, Nave, 2006

2 simple steps

STEP 1: SPLIT tensors

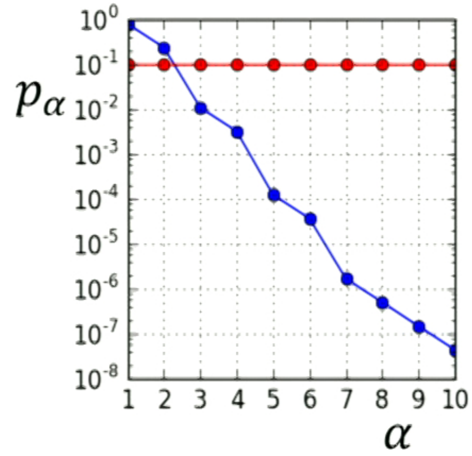


Tensor Renormalization Group (TRG)   Levin, Nave, 2006

2 simple steps

STEP 1: SPLIT tensors

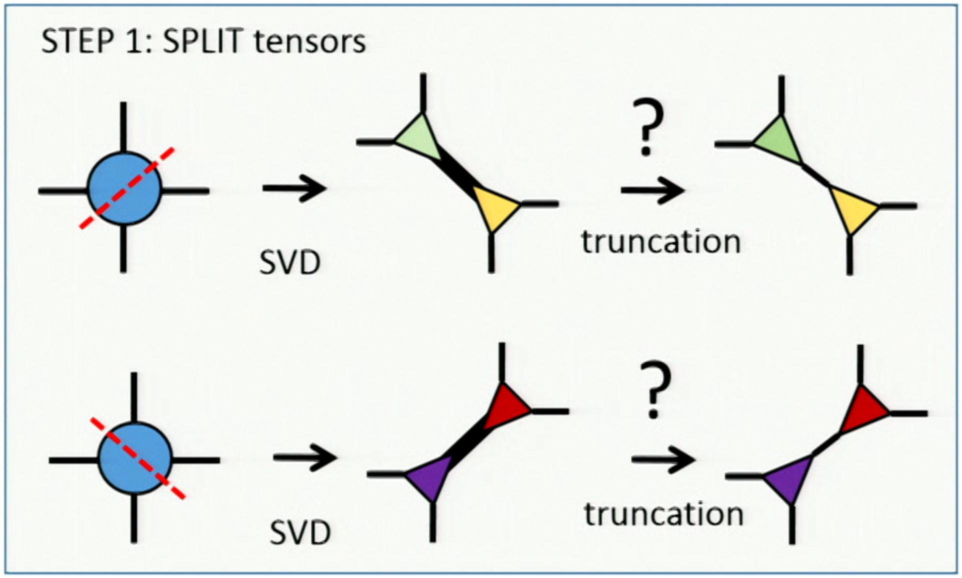
examples of singular values:



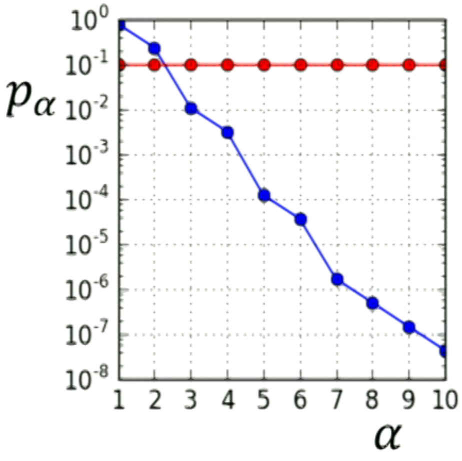
Tensor Renormalization Group (TRG)

Levin, Nave, 2006

2 simple steps



examples of singular values:

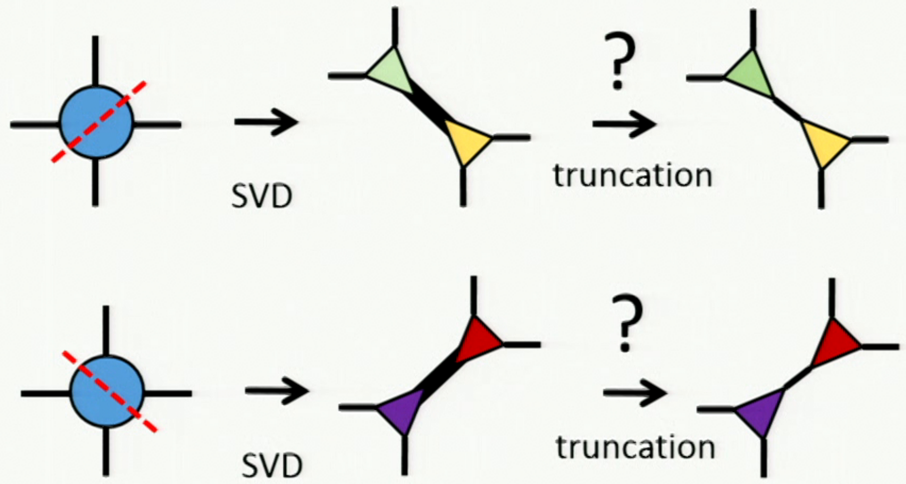


Tensor Renormalization Group (TRG)

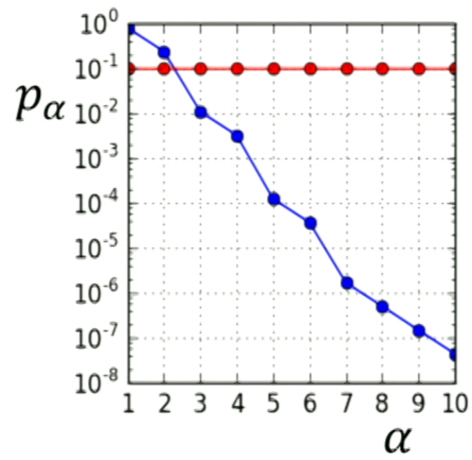
Levin, Nave, 2006

2 simple steps

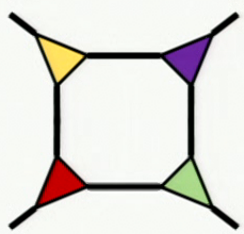
STEP 1: SPLIT tensors



examples of singular values:



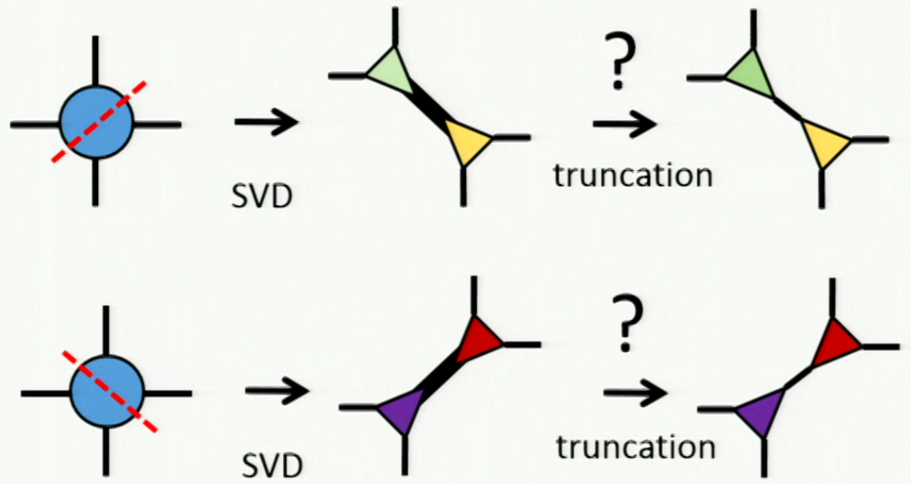
STEP 2: JOIN tensors



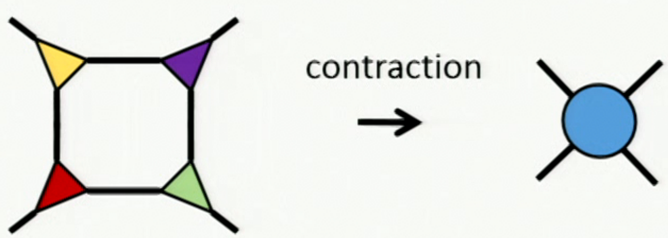
Tensor Renormalization Group (TRG)   Levin, Nave, 2006

2 simple steps

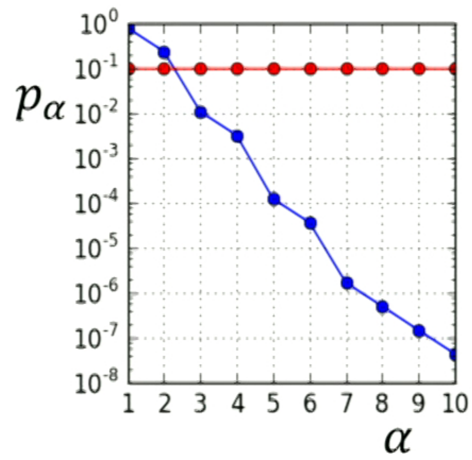
STEP 1: SPLIT tensors



STEP 2: JOIN tensors

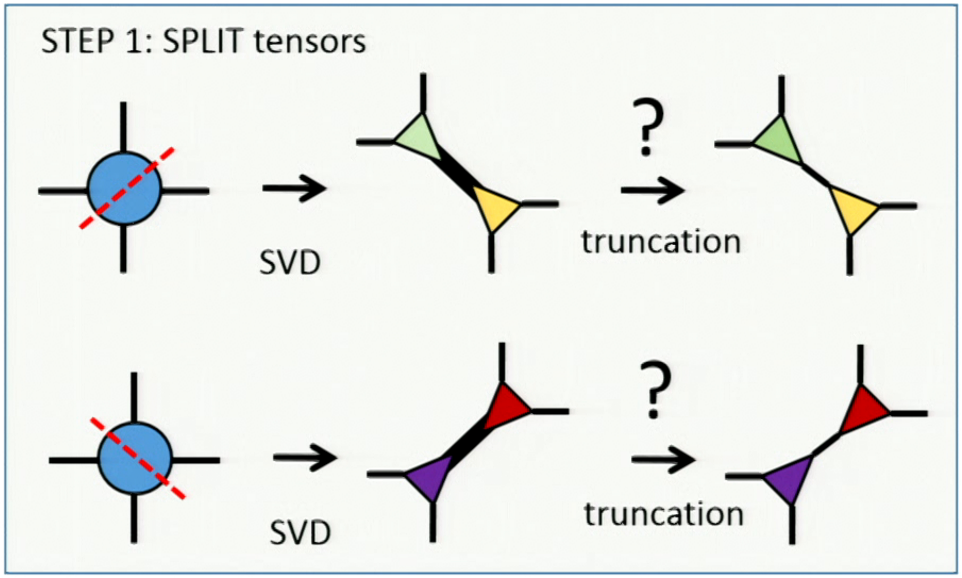


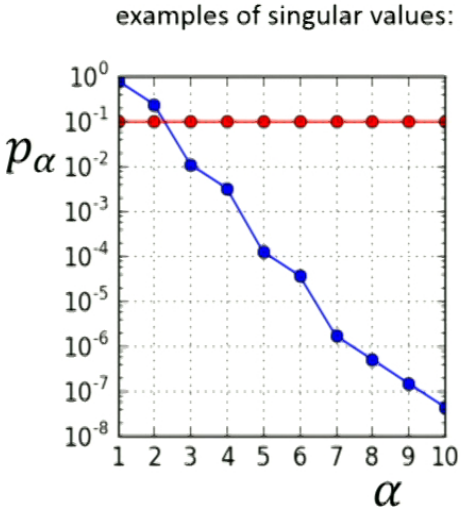
examples of singular values:



Tensor Renormalization Group (TRG) Levin, Nave, 2006

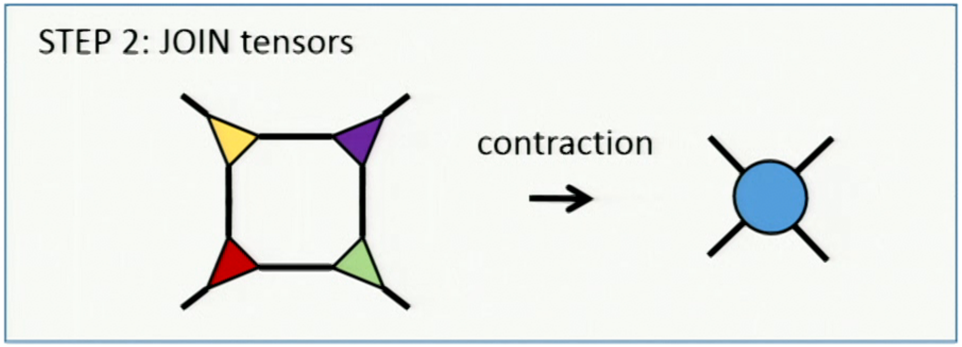
2 simple steps





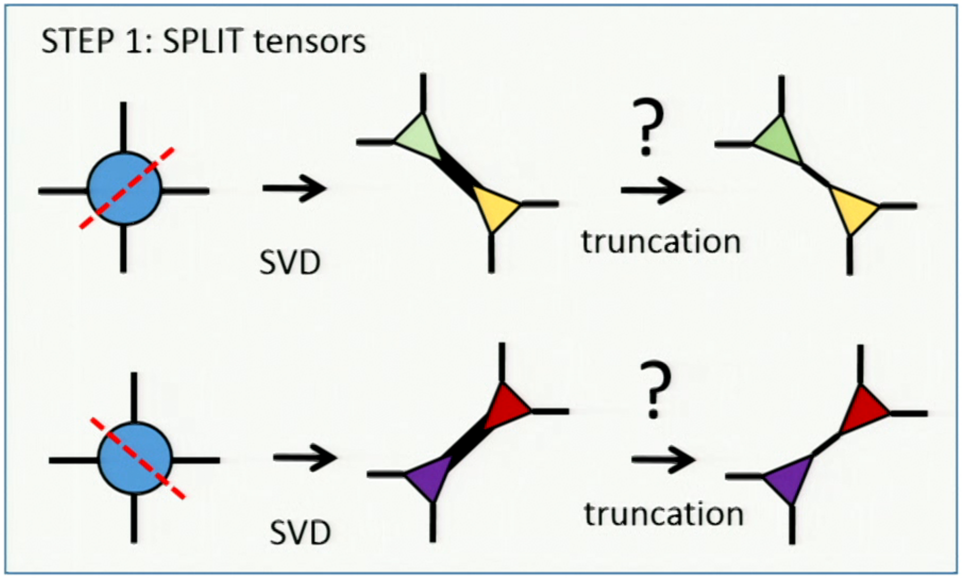
$$\text{cost} \sim \chi^6$$

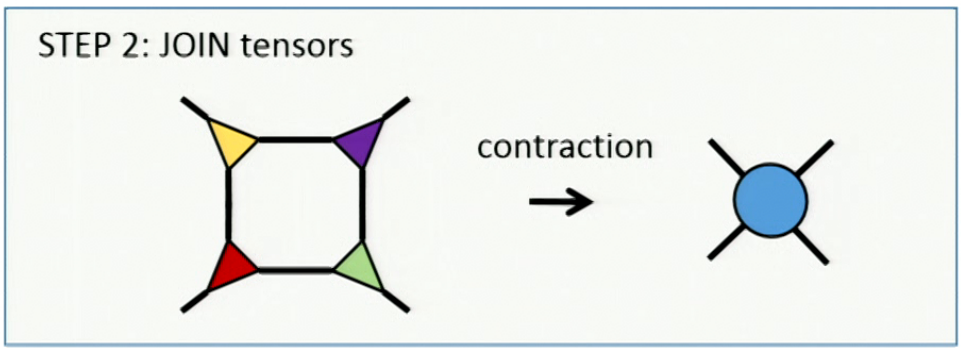
$(\chi \approx 100)$



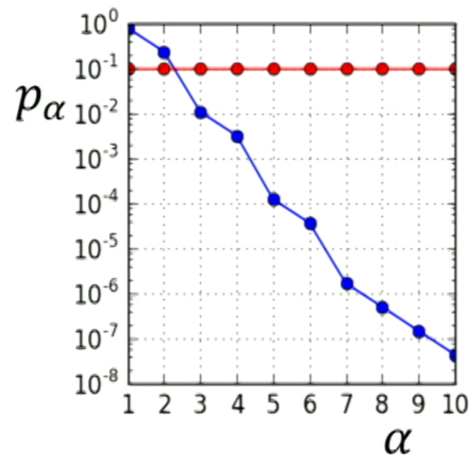
Tensor Renormalization Group (TRG) Levin, Nave, 2006

2 simple steps





examples of singular values:



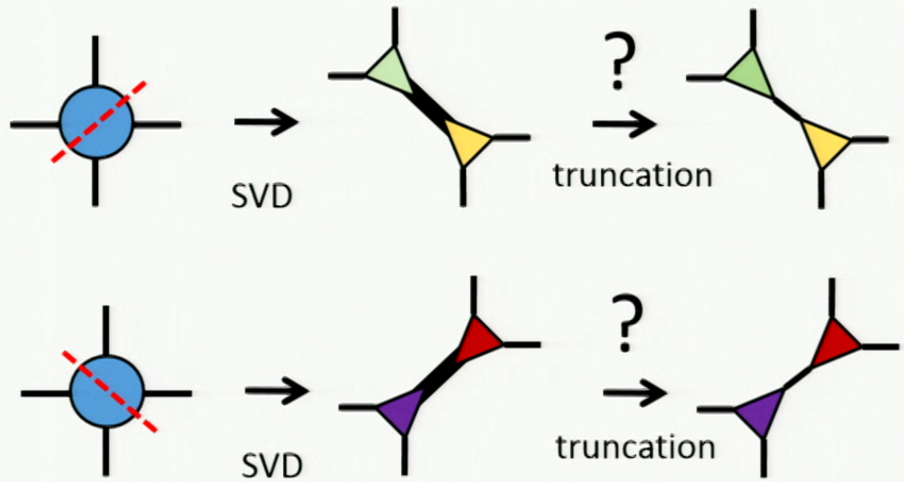
$$\text{cost} \sim \chi^6$$

$(\chi \approx 100)$

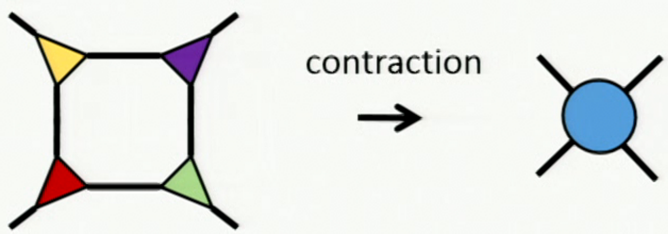
Tensor Renormalization Group (TRG)   Levin, Nave, 2006

2 simple steps

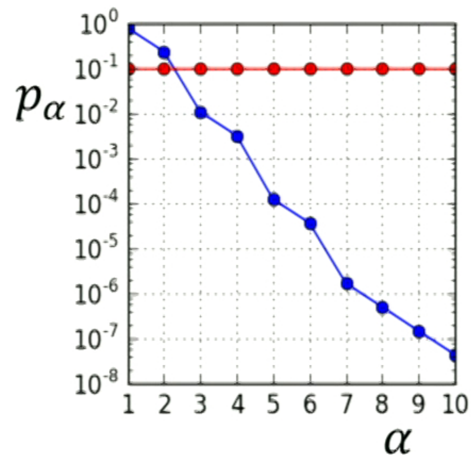
STEP 1: SPLIT tensors



STEP 2: JOIN tensors



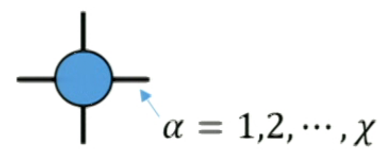
examples of singular values:



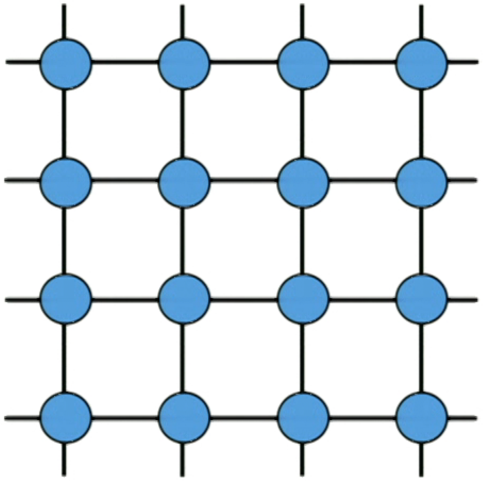
$$\text{cost} \sim \chi^6$$

$(\chi \approx 100)$

χ bond dimension
= # singular values kept

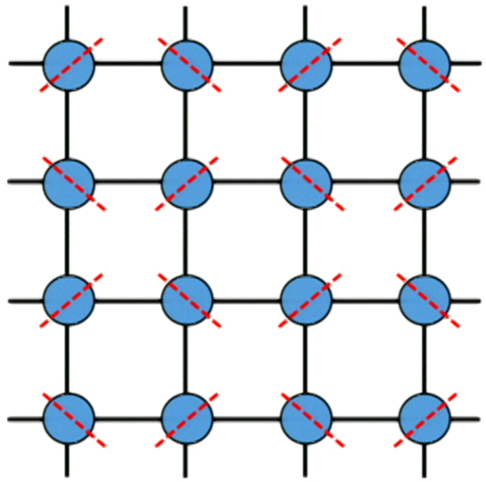


Tensor Renormalization Group (TRG) Levin, Nave, 2006



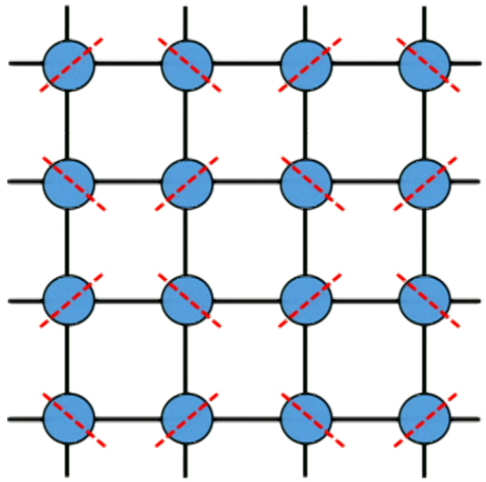
Tensor Renormalization Group (TRG)

Levin, Nave, 2006

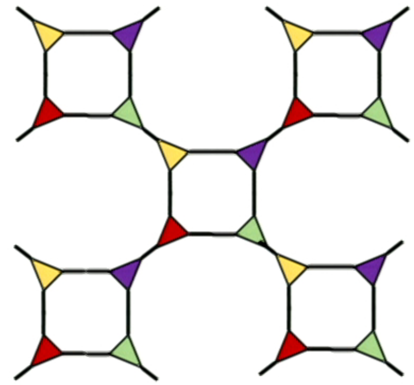


Tensor Renormalization Group (TRG)

Levin, Nave, 2006

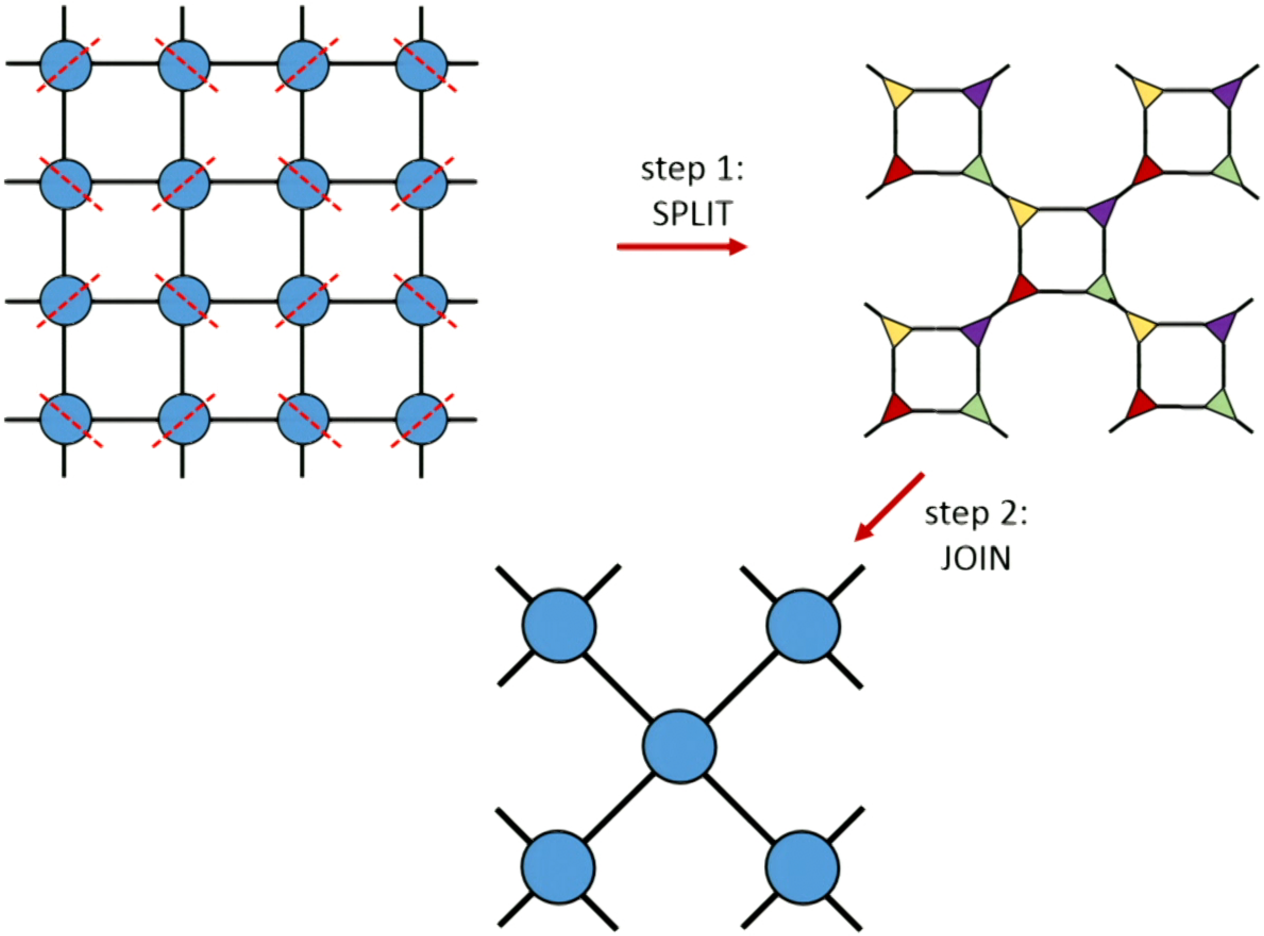


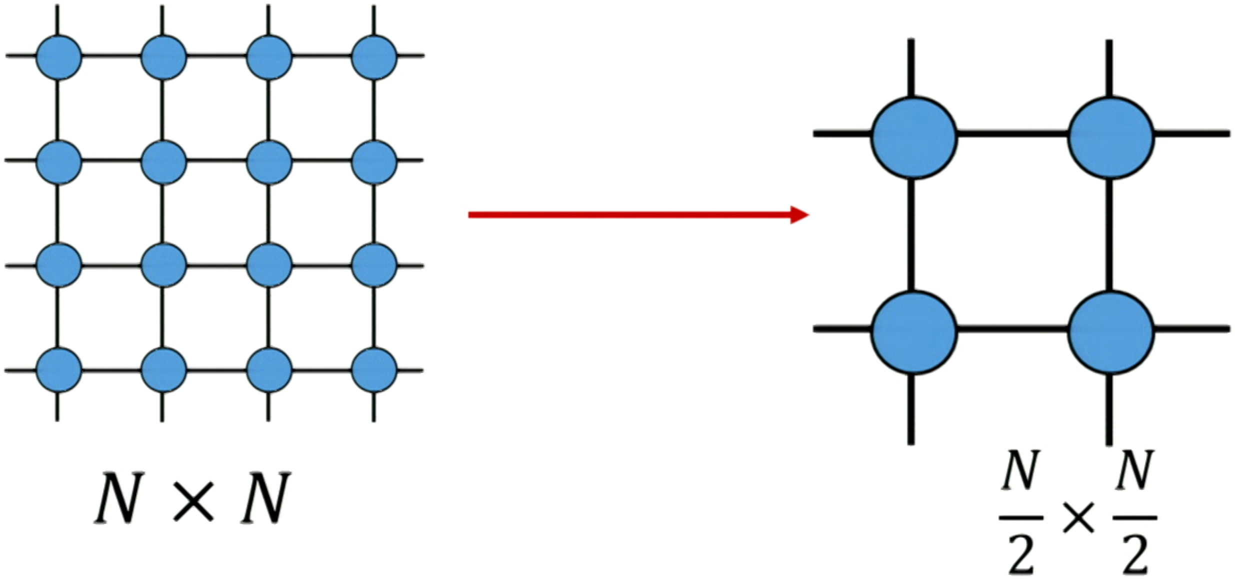
step 1:
SPLIT

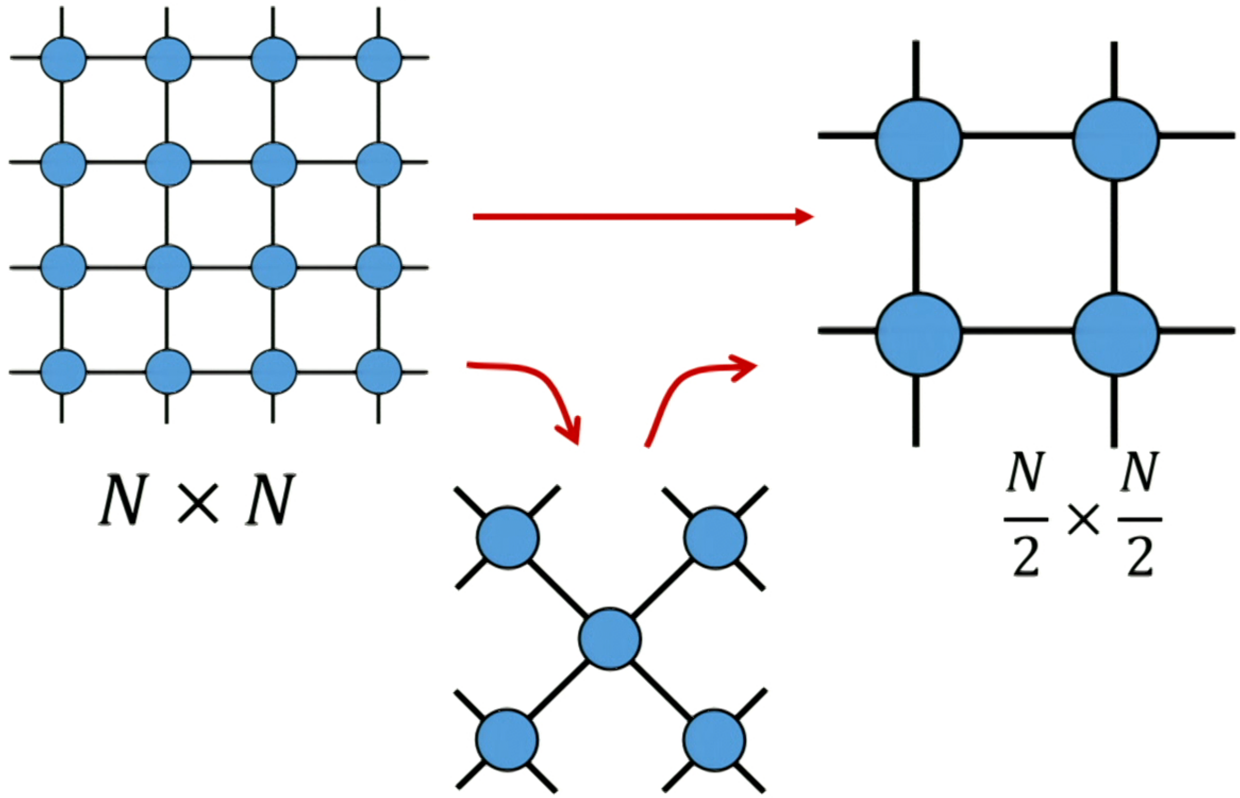


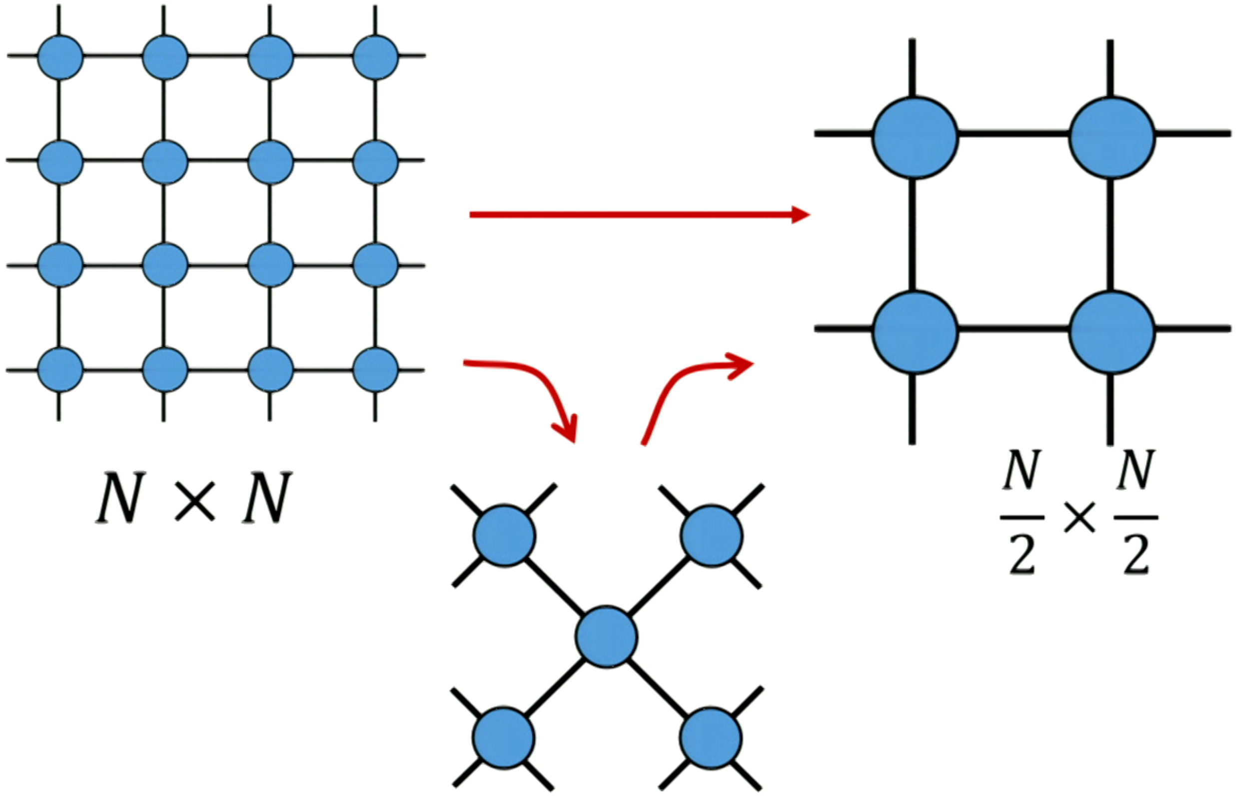
Tensor Renormalization Group (TRG)

Levin, Nave, 2006



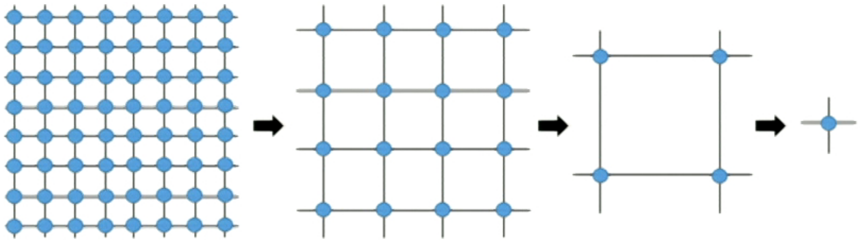

$N \times N$
$\frac{N}{2} \times \frac{N}{2}$


$N \times N$
$\frac{N}{2} \times \frac{N}{2}$



How many times do we do this?

How many times do we do this?

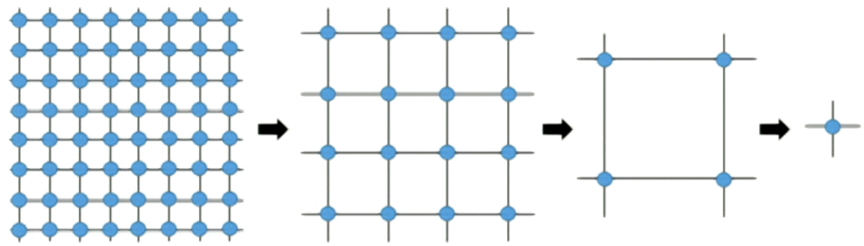


$$N \times N \quad \rightarrow \quad \frac{N}{2} \times \frac{N}{2} \quad \rightarrow \quad \cdots \quad \rightarrow \quad 1 \times 1$$

$\log_2 N$

coarse-graining steps

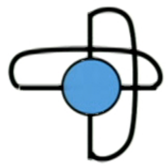
How many times do we do this?



$$N \times N \quad \rightarrow \quad \frac{N}{2} \times \frac{N}{2} \quad \rightarrow \quad \cdots \quad \rightarrow \quad 1 \times 1$$

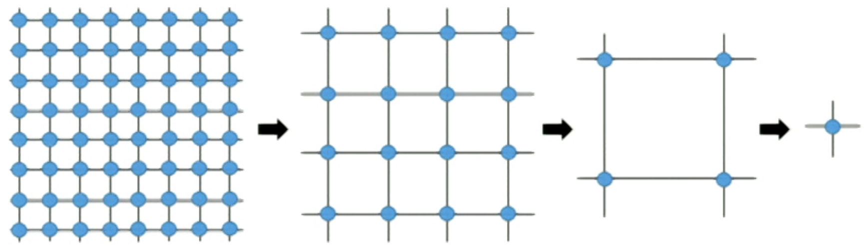
$\underbrace{\qquad\qquad}_{\log_2 N}$

$\log_2 N$
coarse-graining steps

$Z \approx$ 

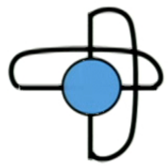
truncations

How many times do we do this?



$$N \times N \quad \rightarrow \quad \frac{N}{2} \times \frac{N}{2} \quad \rightarrow \quad \cdots \quad \rightarrow \quad 1 \times 1$$

$\log_2 N$
coarse-graining steps

$Z \approx$ 

truncations

$$\text{cost} \sim (\chi_\epsilon)^6 \log_2 N$$

of computing Z
with truncation
error ϵ

Performance of TRG (1) non-critical partition function

Performance of TRG (1) non-critical partition function

singular values of tensor $A^{(\tau)}$

coarse-graining steps

1 2 3 4 τ

Performance of TRG (1) non-critical partition function

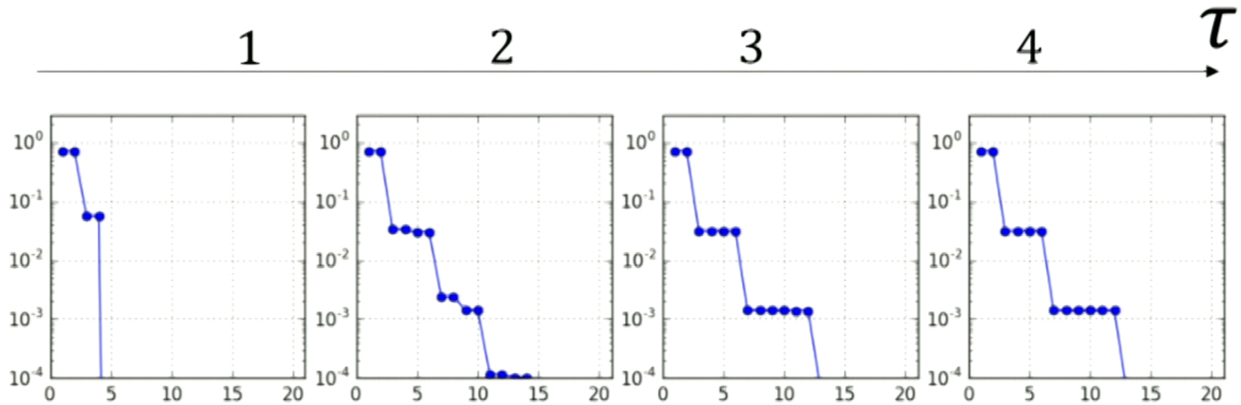
singular values of tensor $A^{(\tau)}$

coarse-graining steps

low temperature

$\beta = 0.8$

$(\beta_c \approx 0.4406)$



Performance of TRG (1) non-critical partition function

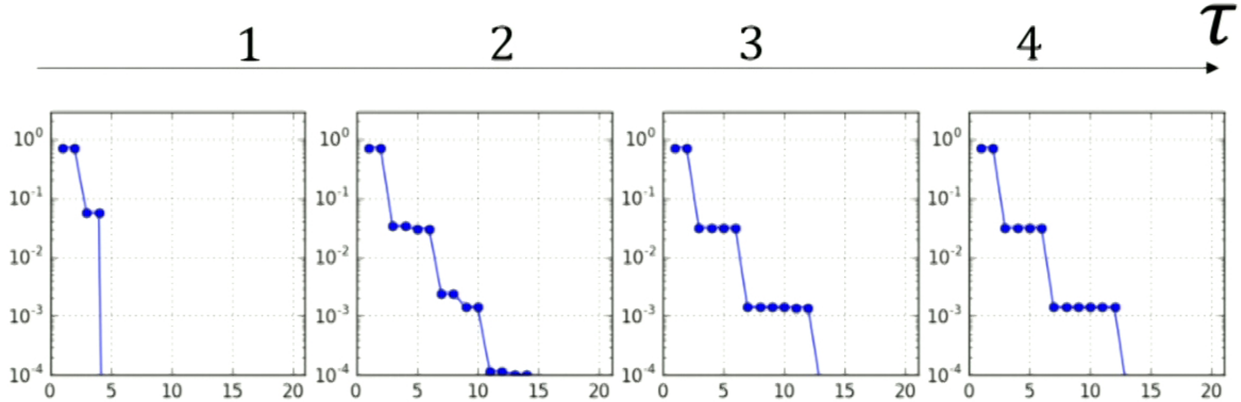
singular values of tensor $A^{(\tau)}$

coarse-graining steps

low temperature

$\beta = 0.8$

($\beta_c \approx 0.4406$)



- singular values decay very fast
- spectrum converges to some fixed spectrum
- we can iterate for ever – very large systems!!!

Performance of TRG (1) non-critical partition function

singular values of tensor $A^{(\tau)}$

coarse-graining steps

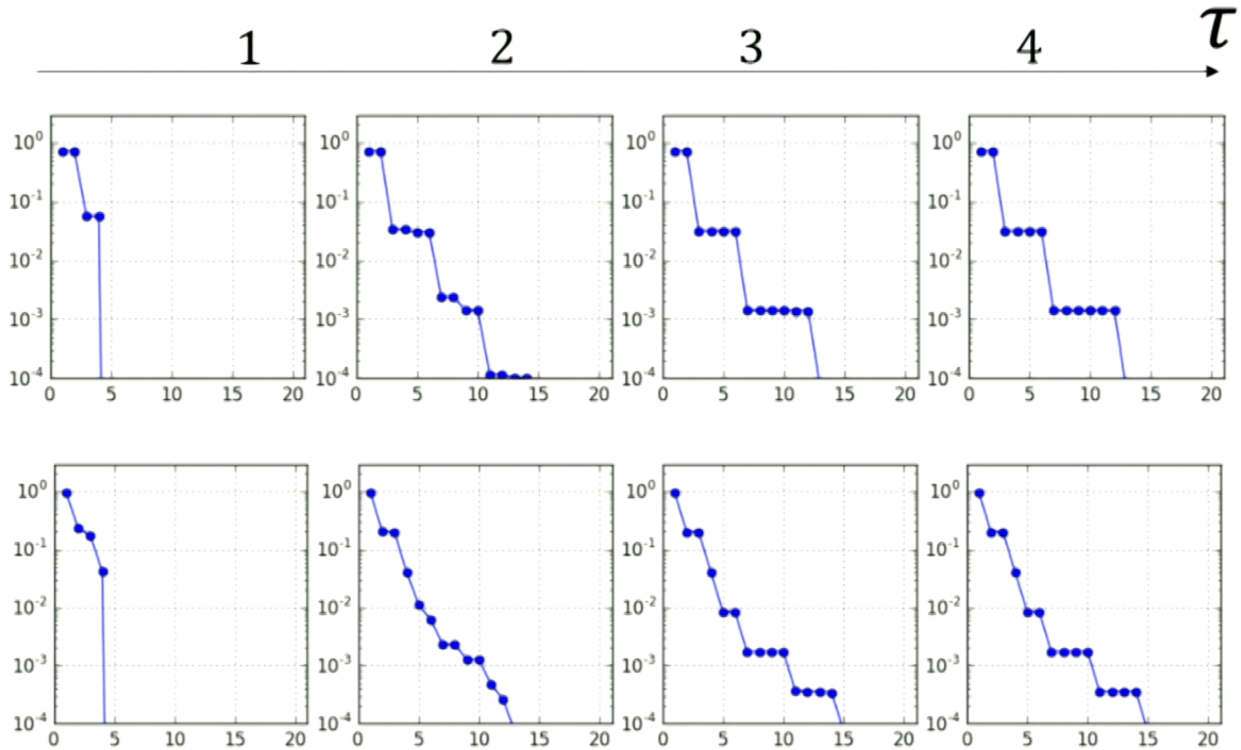
low temperature

$\beta = 0.8$

$(\beta_c \approx 0.4406)$

high temperature

$\beta = 0.2$

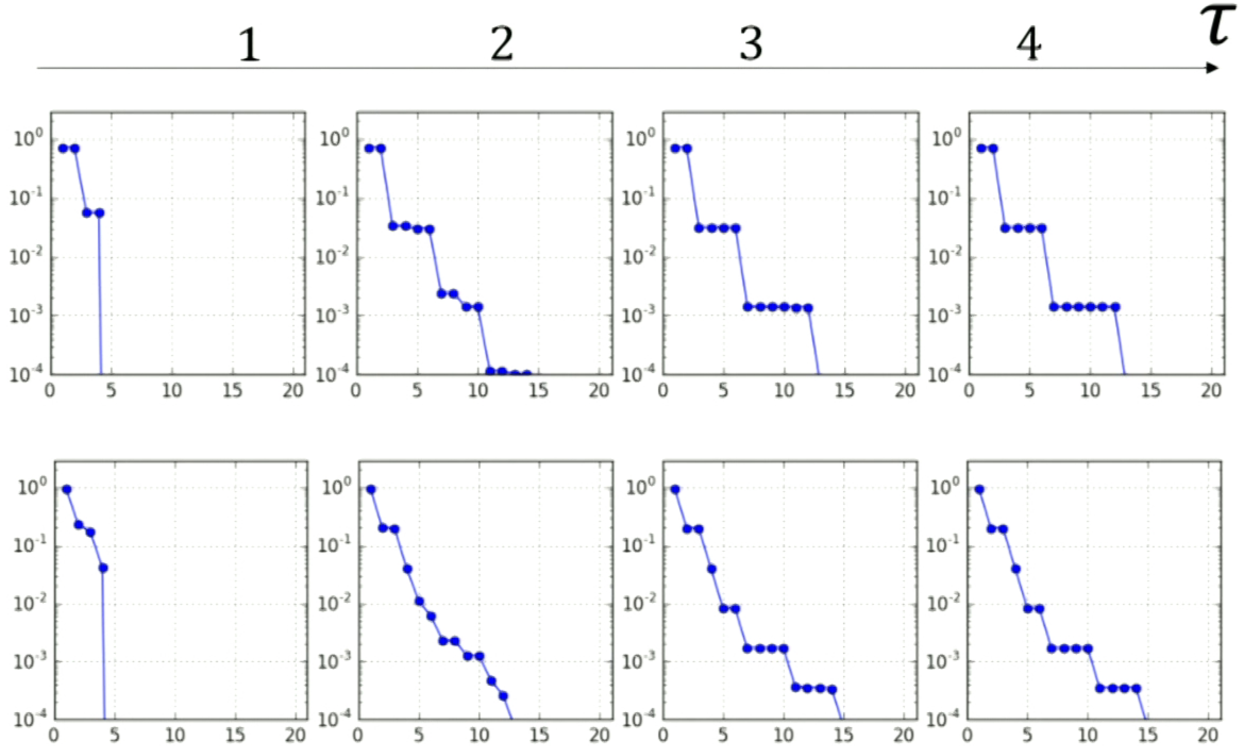


- singular values decay very fast
- spectrum converges to some fixed spectrum
- we can iterate for ever – very large systems!!!

Performance of TRG (1) non-critical partition function

singular values of tensor $A^{(\tau)}$

coarse-graining steps



low temperature

$\beta = 0.8$

$(\beta_c \approx 0.4406)$

high temperature

$\beta = 0.2$

- singular values decay very fast
- spectrum converges to some fixed spectrum
- we can iterate for ever – very large systems!!!

cost $\sim \chi^6 \log N$

$\chi \sim const(\beta)$

'area law'

Performance of TRG (2) critical partition function

Performance of TRG (2) critical partition function

singular values of tensor $A^{(\tau)}$

coarse-graining steps

$2 \quad 3 \quad 4 \quad (\cdots) \quad 8 \quad \tau$

Performance of TRG (2) critical partition function

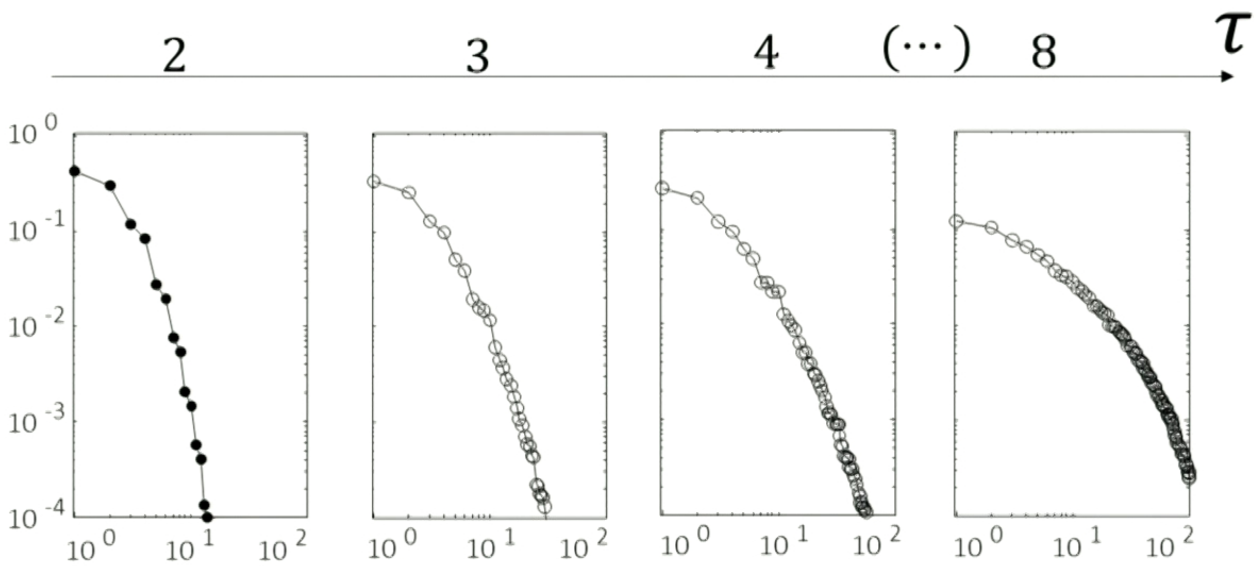
singular values of tensor $A^{(\tau)}$

critical temperature

$\beta = \beta_c$

$(\beta_c \approx 0.4406)$

coarse-graining steps



Performance of TRG (2) critical partition function

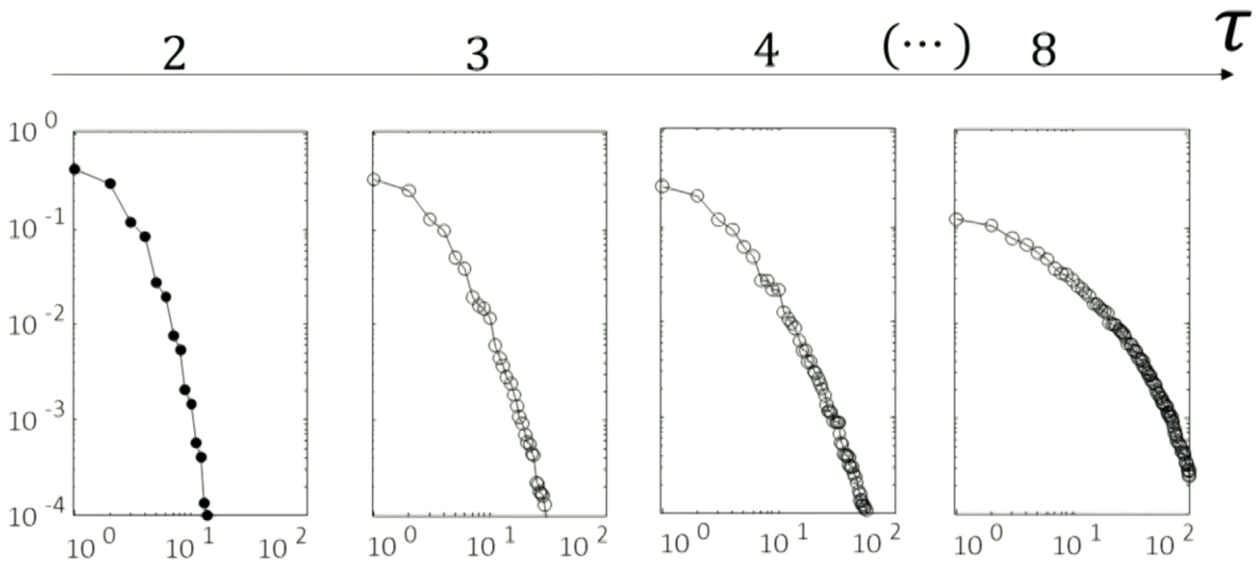
singular values of tensor $A^{(\tau)}$

coarse-graining steps

critical temperature

$\beta = \beta_c$

$(\beta_c \approx 0.4406)$



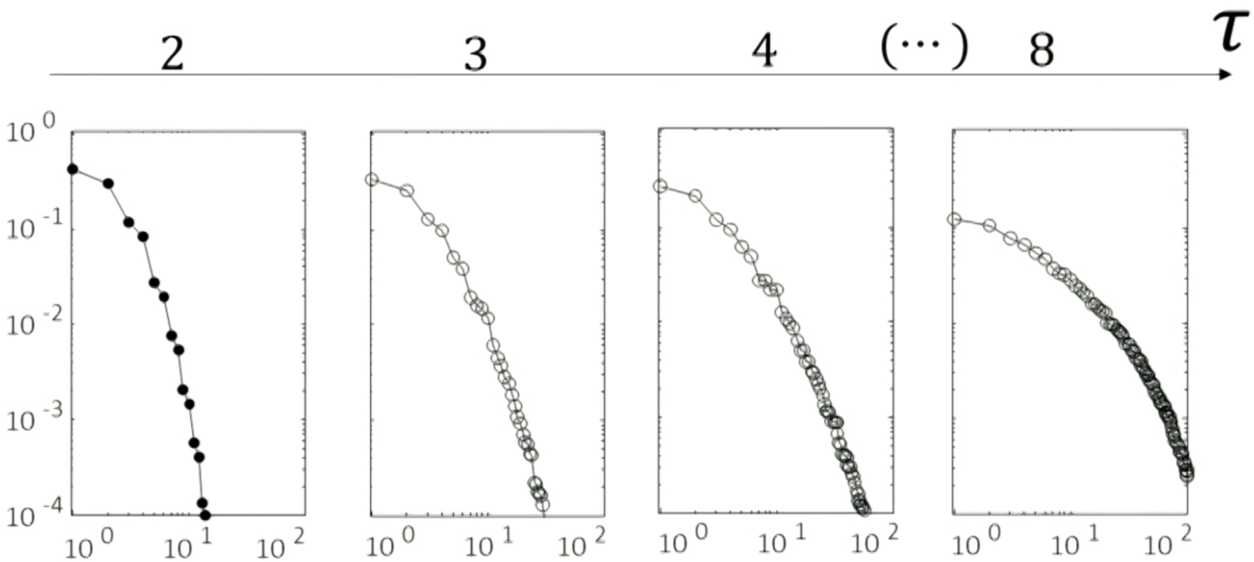
- singular values do not decay as fast (but spectrum is not flat!)
- spectrum does not converges to some fixed spectrum
- each iteration is more expensive (if we want to keep accuracy fixed, we need to increase bond dimension χ)

cost $\sim \chi^6$

Performance of TRG (2) critical partition function

singular values of tensor $A^{(\tau)}$

coarse-graining steps



critical temperature

$\beta = \beta_c$

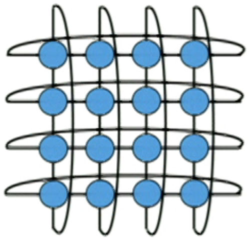
$(\beta_c \approx 0.4406)$

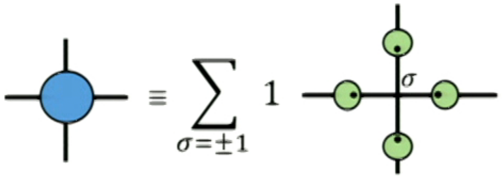
- singular values do not decay as fast (but spectrum is not flat!)
- spectrum does not converges to some fixed spectrum
- each iteration is more expensive (if we want to keep accuracy fixed, we need to increase bond dimension χ)

cost $\sim \chi^6$ $\chi \sim e^{\tau}$ $\text{cost} \sim \exp(\tau)$

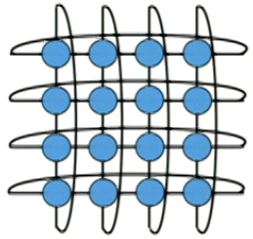
Expectation values and correlation functions

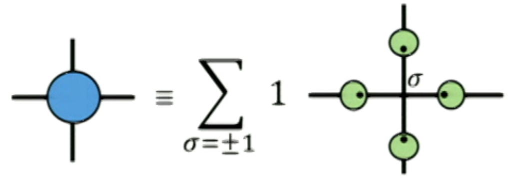
Expectation values and correlation functions

partition function $Z \equiv \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} =$ 



Expectation values and correlation functions

partition function $Z \equiv \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$ $=$ 

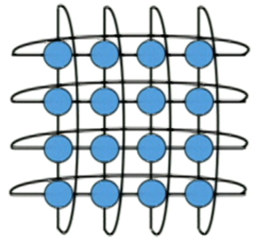


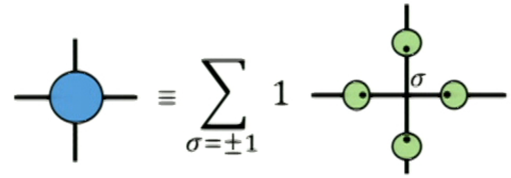
magnetization

$$\langle \sigma^{(1)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} e^{-\beta H(\{\sigma\})}$$

Expectation values and correlation functions

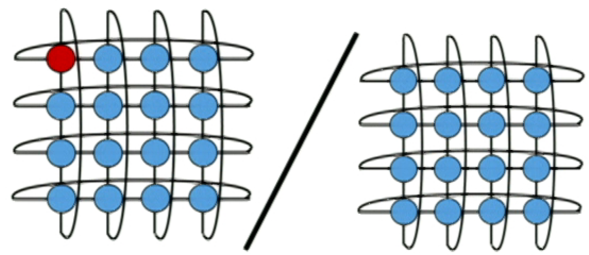
partition function $Z \equiv \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} =$





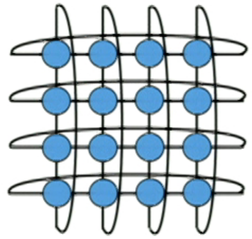
magnetization

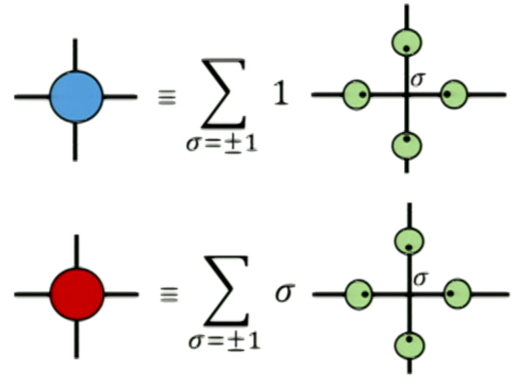
$$\langle \sigma^{(1)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} e^{-\beta H(\{\sigma\})} =$$



Expectation values and correlation functions

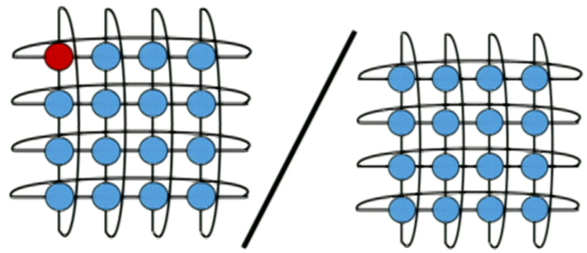
partition function $Z \equiv \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$ =





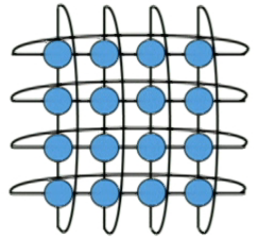
magnetization

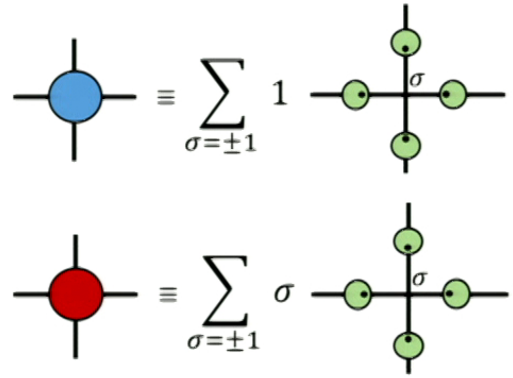
$$\langle \sigma^{(1)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} e^{-\beta H(\{\sigma\})} =$$



Expectation values and correlation functions

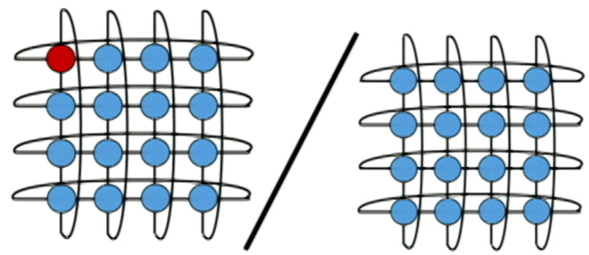
partition function $Z \equiv \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$ =





magnetization

$$\langle \sigma^{(1)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} e^{-\beta H(\{\sigma\})} =$$

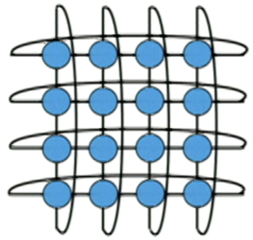


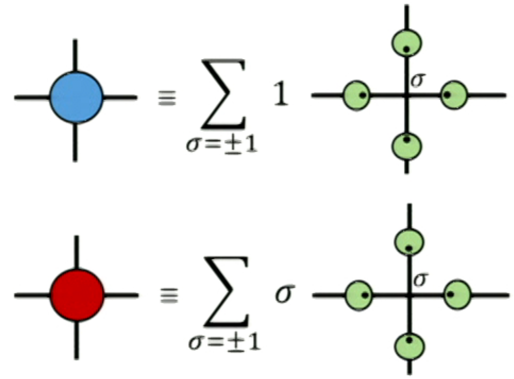
correlator

$$\langle \sigma^{(1)} \sigma^{(7)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} \sigma^{(7)} e^{-\beta H(\{\sigma\})}$$

Expectation values and correlation functions

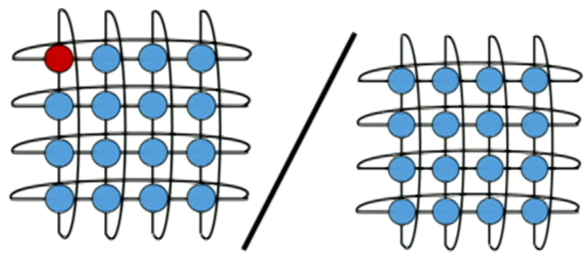
partition function $Z \equiv \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$ $=$





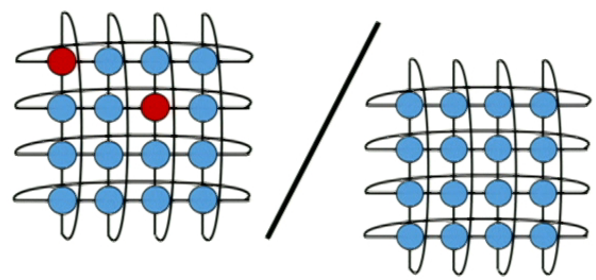
magnetization

$$\langle \sigma^{(1)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} e^{-\beta H(\{\sigma\})} =$$



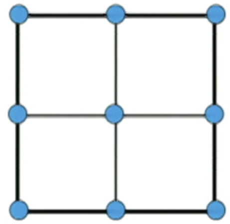
correlator

$$\langle \sigma^{(1)} \sigma^{(7)} \rangle \equiv \frac{1}{Z} \sum_{\{\sigma\}} \sigma^{(1)} \sigma^{(7)} e^{-\beta H(\{\sigma\})} =$$



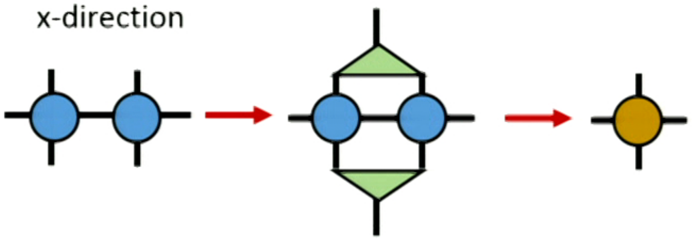
Performance of in D=3 dimensions

Performance of in D=3 dimensions

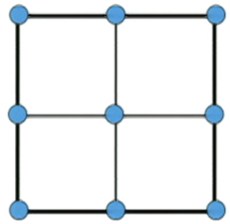


D=2 TRG is essentially equivalent to

coarse-graining
x-direction

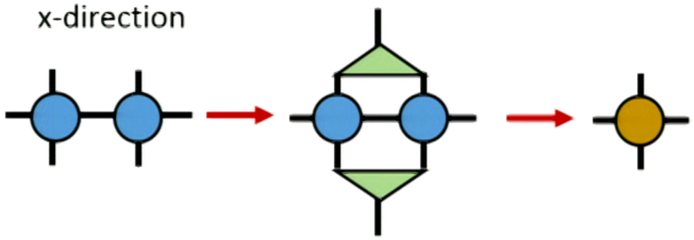


Performance of in D=3 dimensions

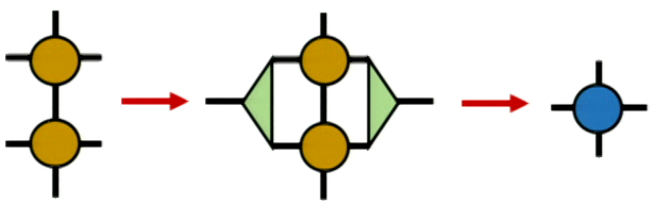


D=2 TRG is essentially equivalent to

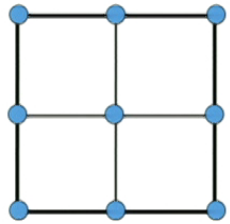
coarse-graining
x-direction



coarse-graining
y-direction

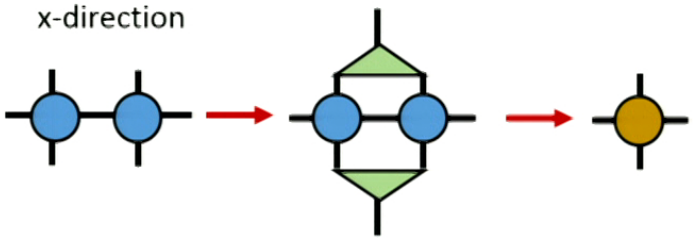


Performance of in D=3 dimensions

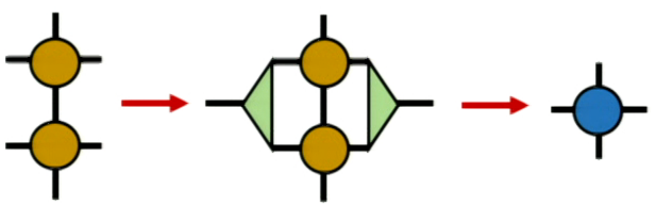


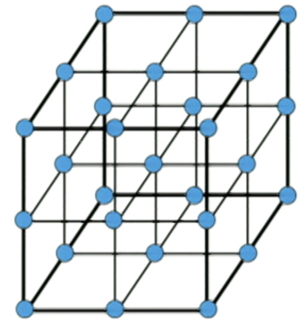
D=2 TRG is essentially equivalent to

coarse-graining
x-direction



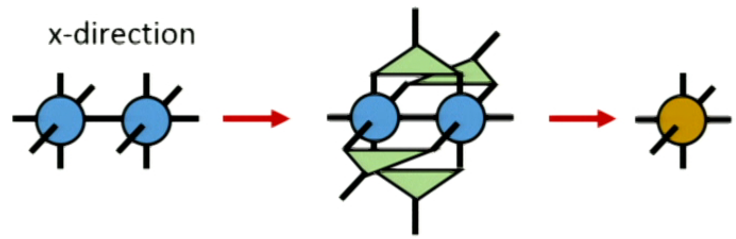
coarse-graining
y-direction



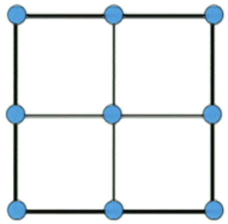


TRG can be generalized to D=3 dimensions

x-direction

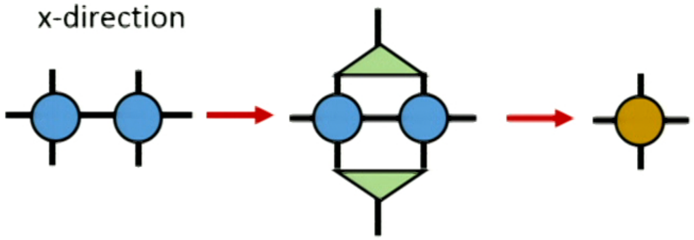


Performance of in D=3 dimensions

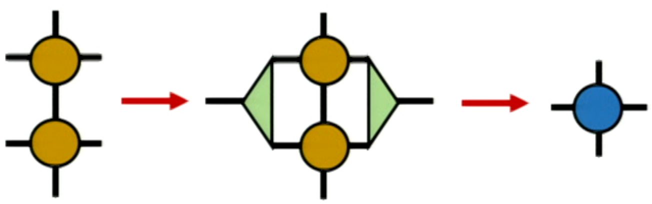


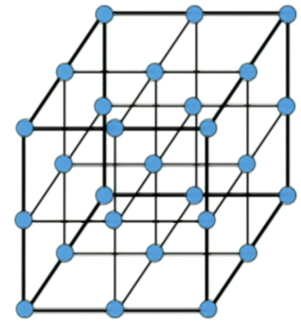
D=2 TRG is essentially equivalent to

coarse-graining
x-direction



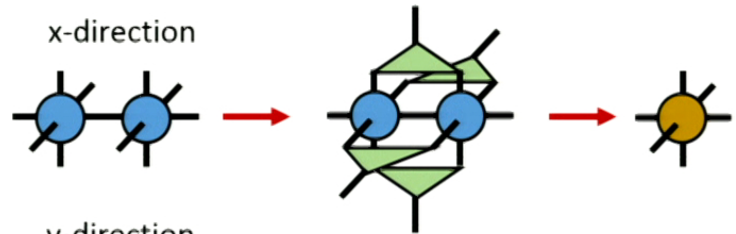
coarse-graining
y-direction



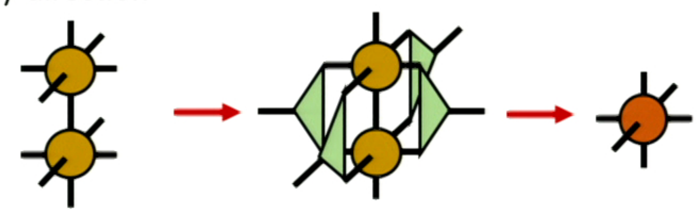


TRG can be generalized to D=3 dimensions

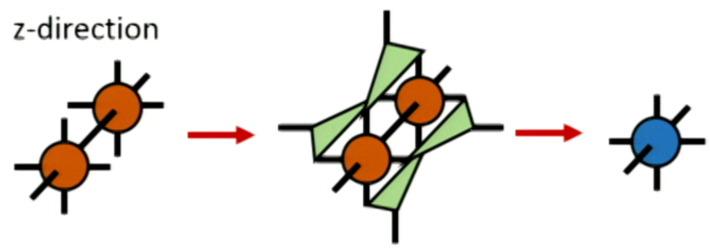
x-direction



y-direction

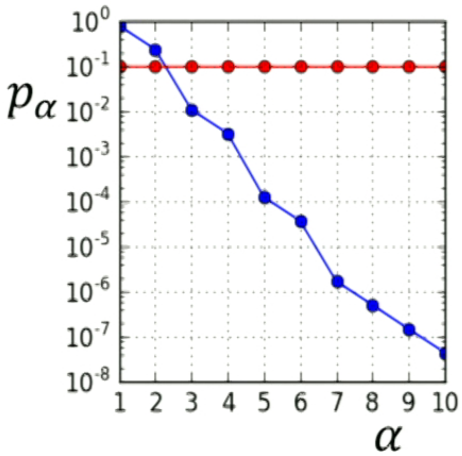


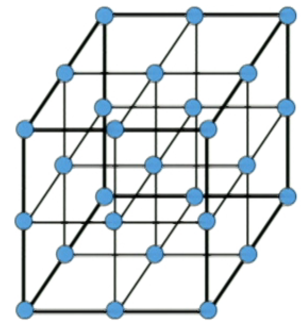
z-direction



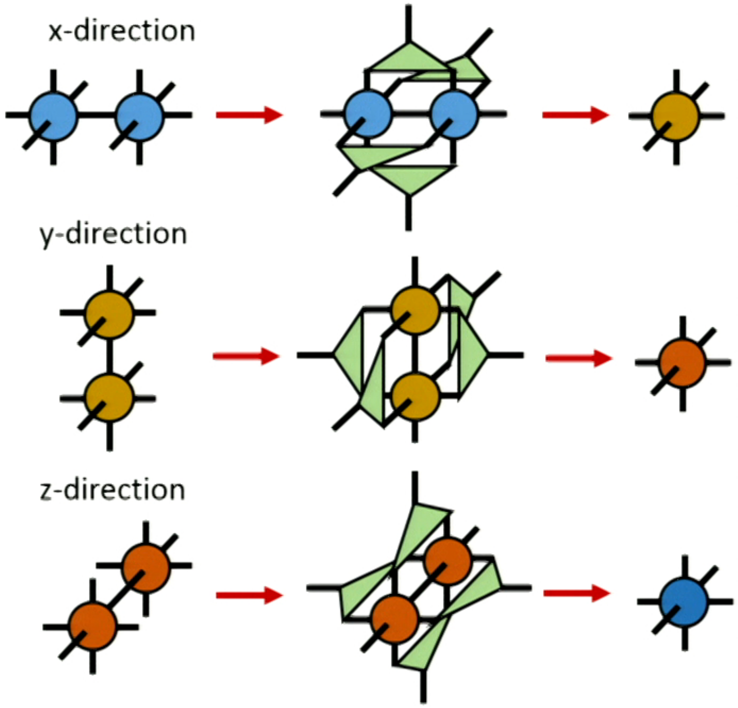
Performance of in D=3 dimensions

examples of singular values:



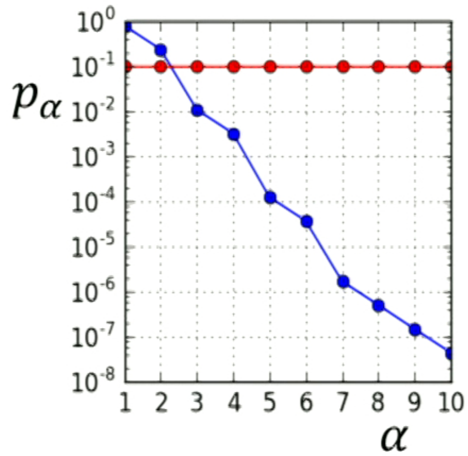


TRG can be generalized to D=3 dimensions



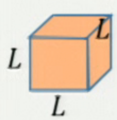
Performance of in D=3 dimensions

examples of singular values:

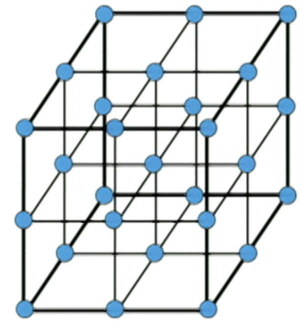


$$S_{L^D} \sim L^{D-2}$$

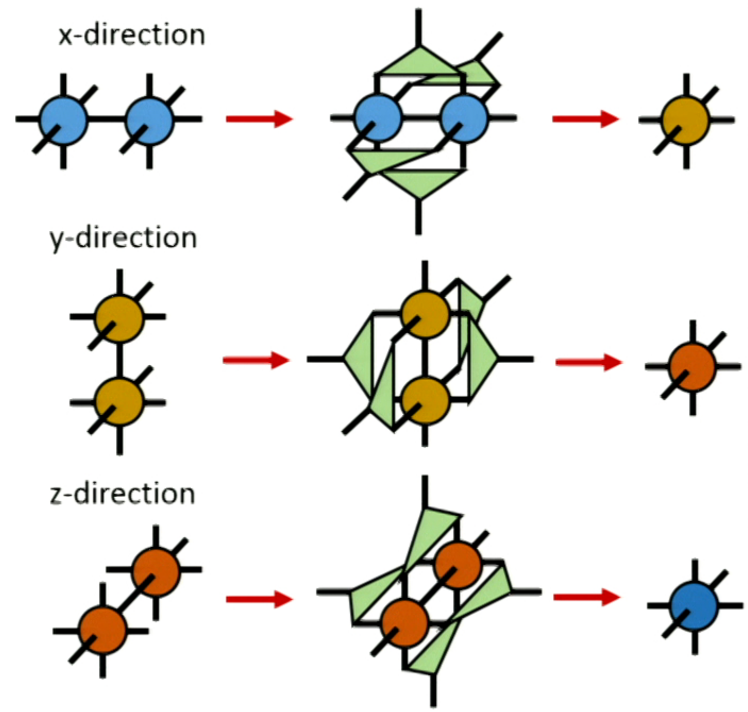
$D = 3$ spacetime dimensions



$$S_{L^3} \sim L$$

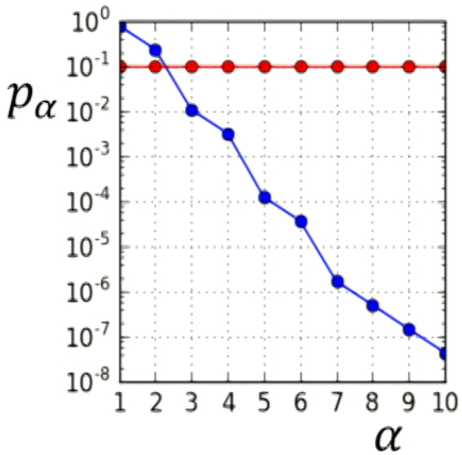


TRG can be generalized to D=3 dimensions



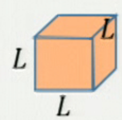
Performance of in D=3 dimensions

examples of singular values:



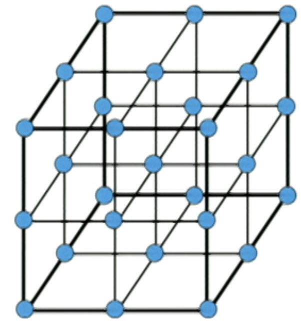
$S_{L^D} \sim L^{D-2}$

$D = 3$ spacetime dimensions

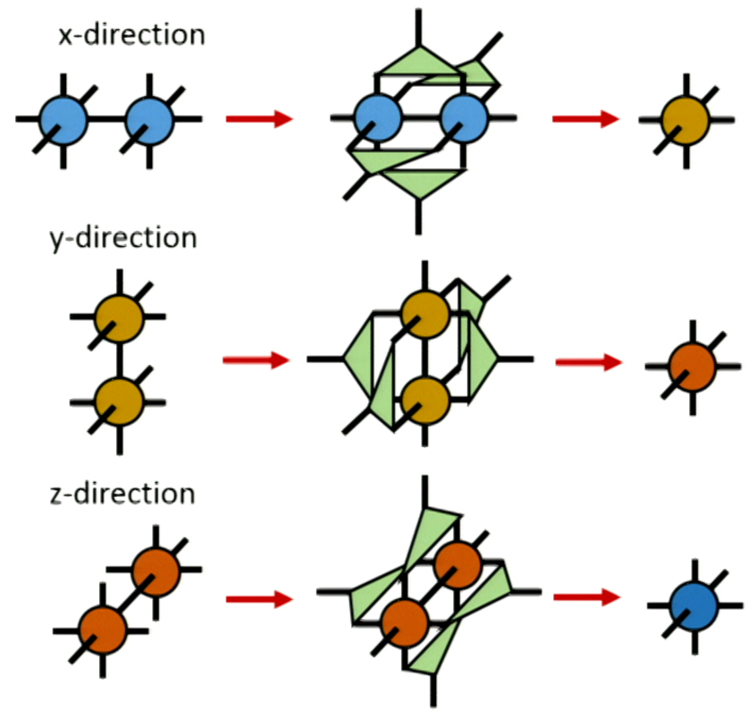


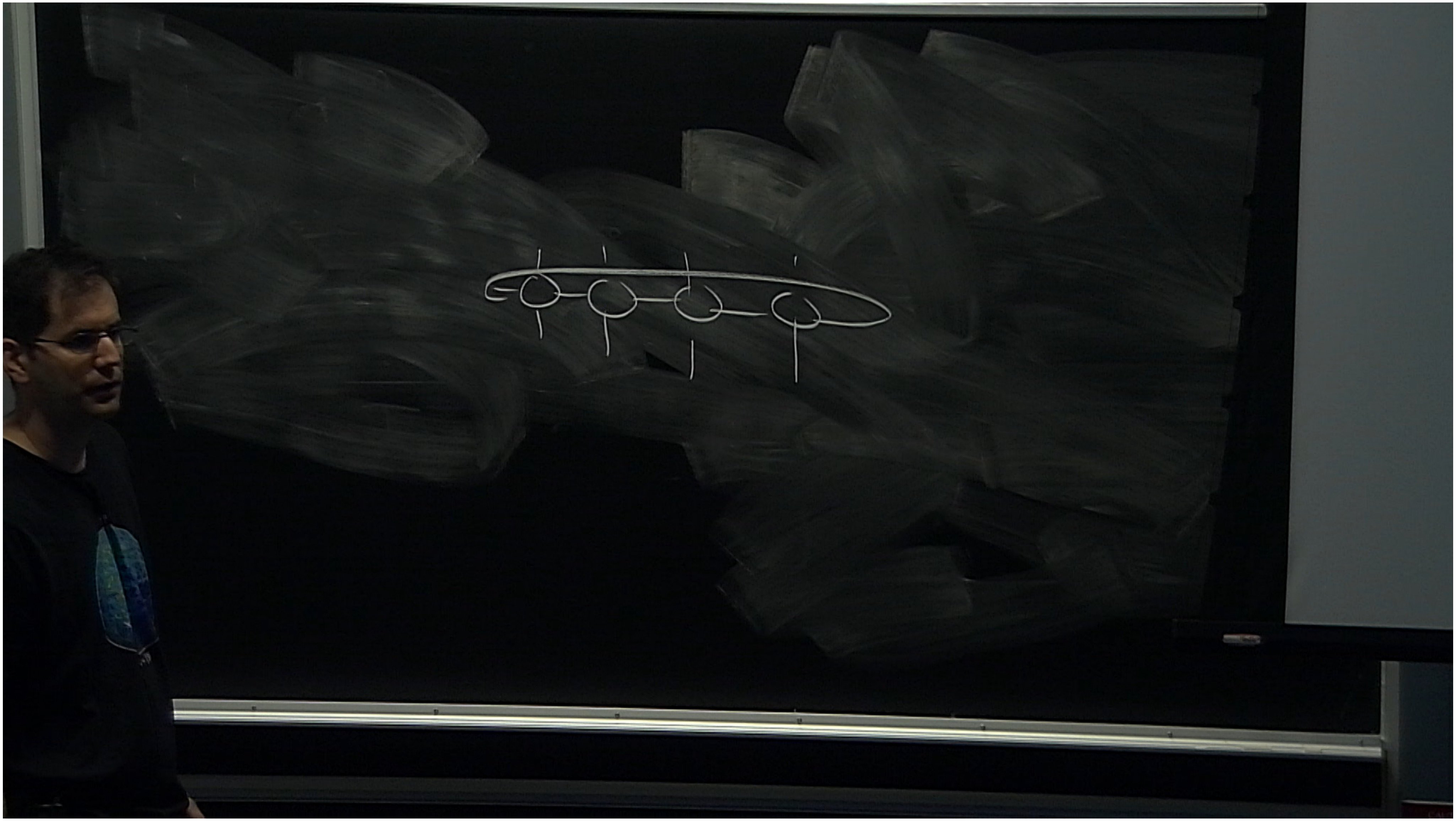
$S_{L^3} \sim L$

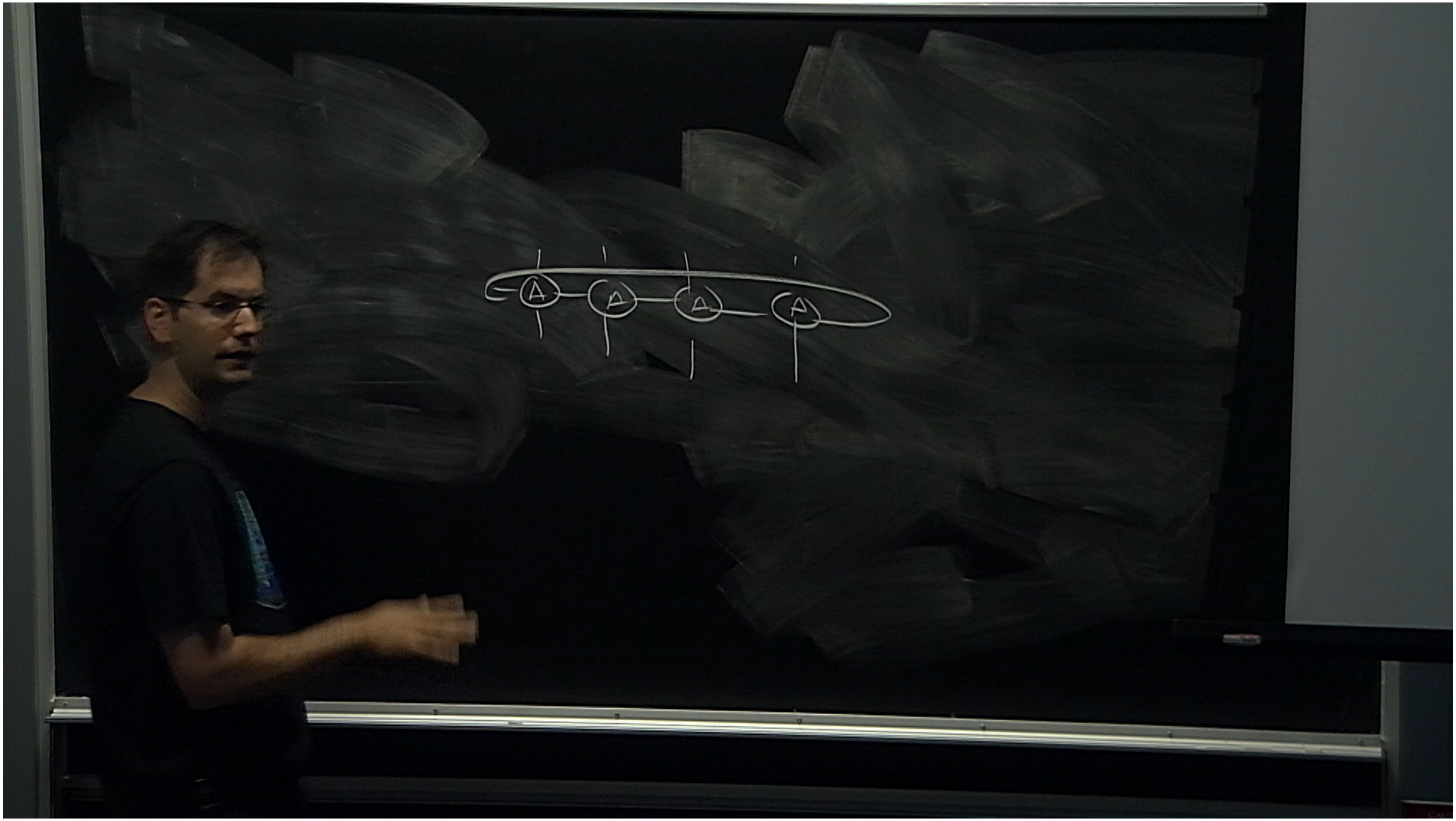
$$\text{cost} \sim \exp S_{L^3} \sim \exp L$$



TRG can be generalized to D=3 dimensions

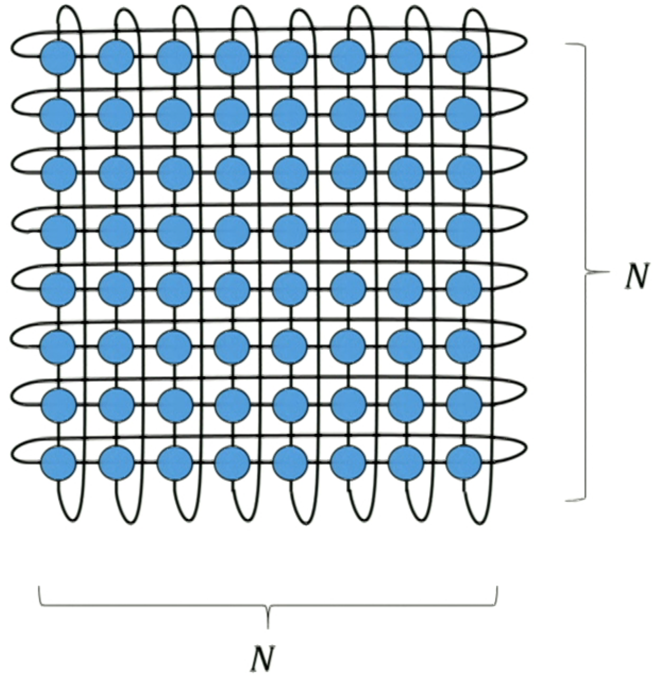





A
A
A
A

Back to D=2 TRG: Transfer matrix

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} =$$

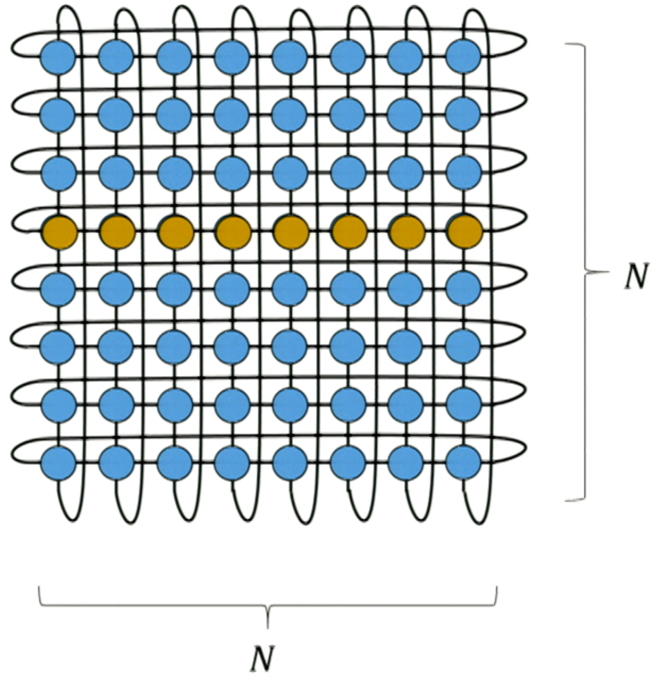


partition function

Back to D=2 TRG: Transfer matrix

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} =$$

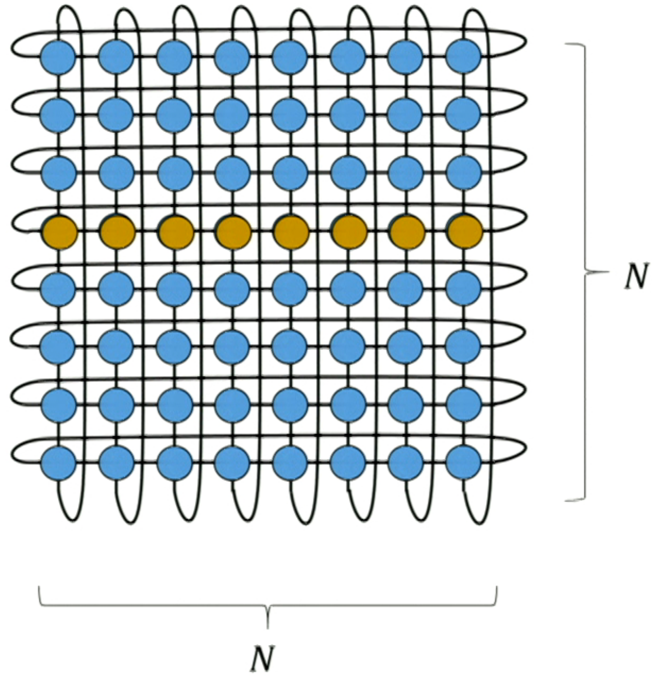
partition function



Back to D=2 TRG: Transfer matrix

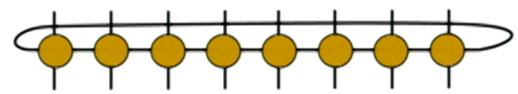
$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} =$$

partition function



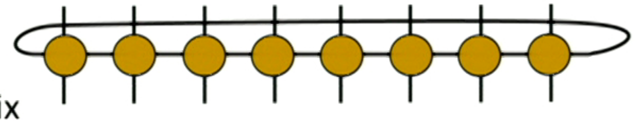
$$Z = tr\ (M^N)$$

transfer matrix

$M =$ 

Back to D=2 TRG: Transfer matrix

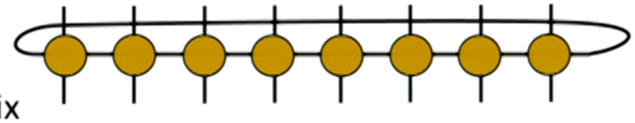
$$Z = tr\,(M^N)$$

$M =$ 

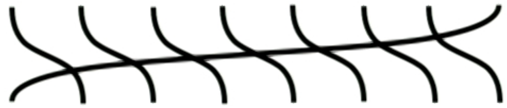
transfer matrix

Back to D=2 TRG: Transfer matrix

$$Z = tr\,(M^N)$$

$M =$ 

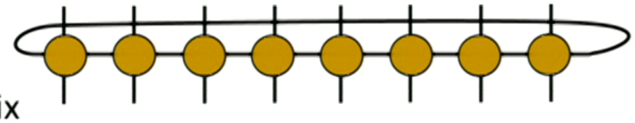
transfer matrix

$T =$ 

translation operator

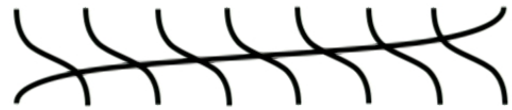
Back to D=2 TRG: Transfer matrix

$$Z = tr\,(M^N)$$

$M =$ 

transfer matrix

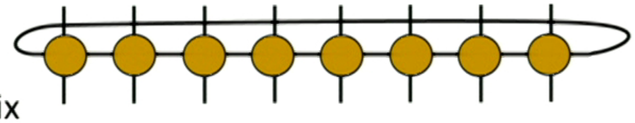
$[M,T] = 0$

$T =$ 

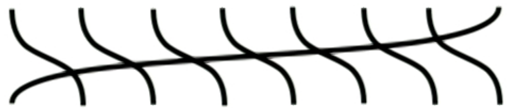
translation operator

Back to D=2 TRG: Transfer matrix

$$Z = tr\,(M^N)$$

$M =$ 

transfer matrix

$T =$ 

translation operator

$$[M, T] = 0$$

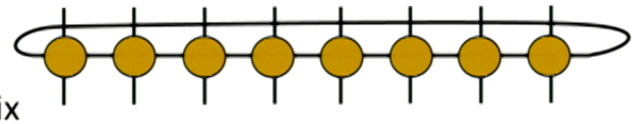
at a critical point:

$$Z \approx Z^{CFT}$$

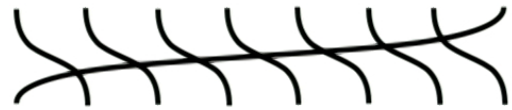
partition function of a conformal field theory (CFT)

Back to D=2 TRG: Transfer matrix

$$Z = tr\,(M^N)$$

$M =$ 

transfer matrix

$T =$ 

translation operator

$$[M, T] = 0$$

at a critical point:

$$Z \approx Z^{CFT}$$

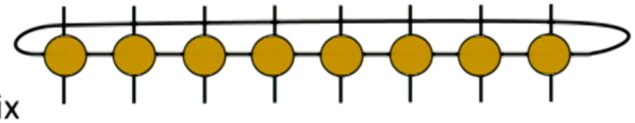
finite size corrections

partition function of a conformal field theory (CFT)

Back to D=2 TRG: Transfer matrix

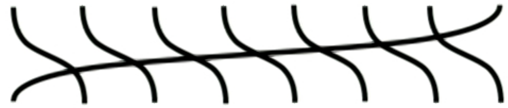
$$Z = tr\left(M^N\right)$$

$M =$



transfer matrix

$T =$



translation operator

$$[M, T] = 0$$

at a critical point:

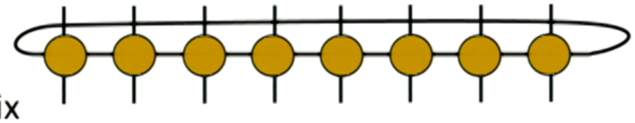
$Z \approx Z^{CFT}$ ← finite size corrections

partition function of a conformal field theory (CFT)

$M \approx M^{CFT}$ ← finite size corrections

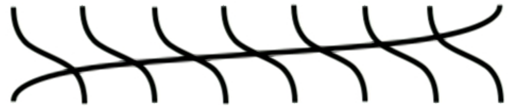
Back to D=2 TRG: Transfer matrix

$$Z = tr\left(M^N\right)$$

$M =$ 

transfer matrix

$[M, T] = 0$

$T =$ 

translation operator

at a critical point:

$Z \approx Z^{CFT}$ ← finite size corrections

partition function of a conformal field theory (CFT)

$M \approx M^{CFT}$ ← finite size corrections

continuous phase transition ⟷ CFT

universal properties

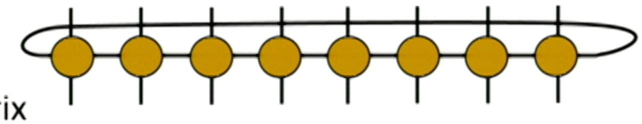
conformal data

e.g. critical exponents

e.g. scaling dimensions Δ_α, conformal spins s_α

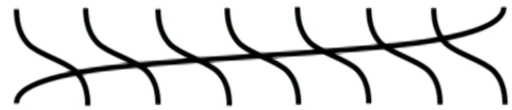
Back to D=2 TRG: Transfer matrix

$$Z = tr\left(M^N\right)$$

$M =$ 

transfer matrix

$[M, T] = 0$

$T =$ 

translation operator

at a critical point:

$Z \approx Z^{CFT}$ — partition function of a conformal field theory (CFT)

finite size corrections

$M \approx M^{CFT}$

finite size corrections

continuous phase transition ⟷ CFT

universal properties

e.g. critical exponents

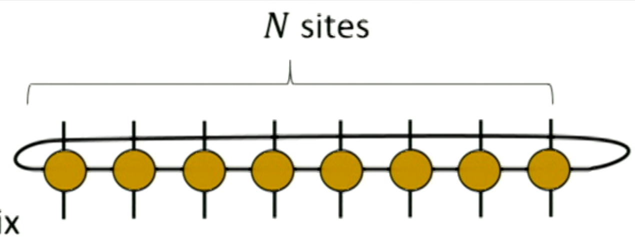
conformal data

e.g. scaling dimensions Δ_α, conformal spins s_α

eigenvalues of M^{CFT} $\quad m_\alpha^{CFT} = ae^{-b\Delta_\alpha}$

eigenvalues of T^{CFT} $\quad t_\alpha^{CFT} = ce^{-ds_\alpha}$

$M =$



transfer matrix

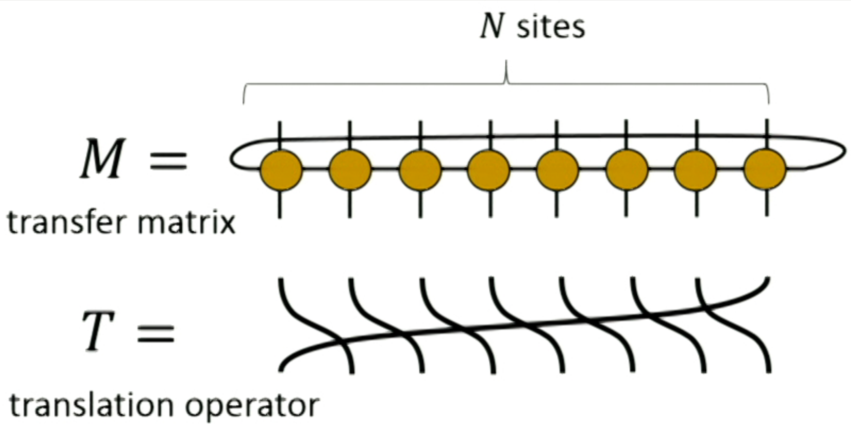
$T =$

translation operator

lattice

eigenvalues of transfer matrix M

$$m_\alpha = a \exp[b\Delta_\alpha + O(1/N)]$$



lattice

eigenvalues of transfer matrix M

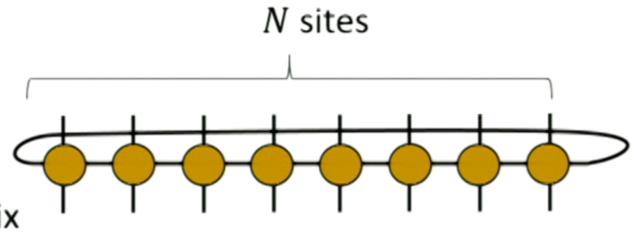
$$m_\alpha = a\exp[b\Delta_\alpha + O(1/N)]$$

CFT

eigenvalues of M^{CFT}

$$m_\alpha^{CFT} = a\exp[b\Delta_\alpha]$$

$M =$



transfer matrix

$T =$

translation operator

lattice

eigenvalues of transfer matrix M

$$m_\alpha = a \exp[b\Delta_\alpha + O(1/N)]$$

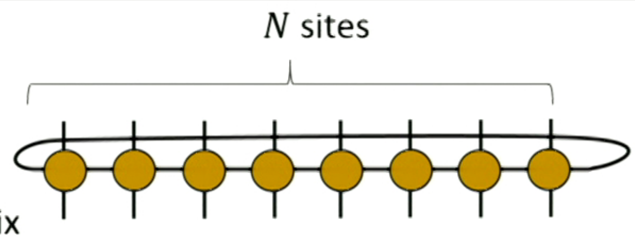
CFT

eigenvalues of M^{CFT}

$$m_\alpha^{CFT} = a \exp[b\Delta_\alpha]$$

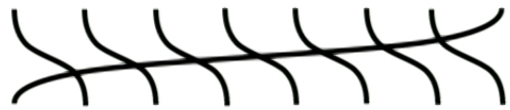
goal: use eigenvalues m_α to compute scaling dimensions Δ_α

$M =$



transfer matrix

$T =$



translation operator

lattice

eigenvalues of transfer matrix M

$$m_\alpha = a\exp[b\Delta_\alpha + O(1/N)]$$

CFT

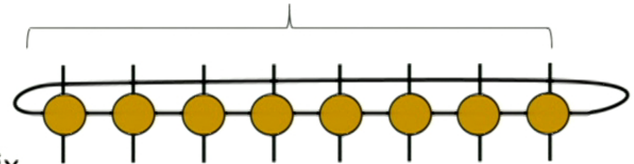
eigenvalues of M^{CFT}

$$m_\alpha^{CFT} = a\exp[b\Delta_\alpha]$$

goal: use eigenvalues m_α to compute scaling dimensions Δ_α

problem: finite-size corrections

N sites

$M =$ 

transfer matrix

$T =$ 

translation operator

lattice

eigenvalues of transfer matrix M

$$m_\alpha = a\exp[b\Delta_\alpha + O(1/N)]$$

goal: use eigenvalues m_α to compute scaling dimensions Δ_α

problem: finite-size corrections

CFT

eigenvalues of M^{CFT}

$$m_\alpha^{CFT} = a\exp[b\Delta_\alpha]$$

larger $N \Longrightarrow$ smaller finite-size errors

without TRG $N \approx 20 - 30$

with TRG $N \approx 100s$