計算科学・量子計算における情報圧縮

Data Compression in Computational Science and Quantum Computing

2022.12.15

#9: テンソル繰り込み

Tensor renormalization

理学系研究科　大久保　毅

Graduate school of science, **Tsuyoshi Okubo**

I put (URLs of) recordings of previous lectures on ITC-LMS.
You can also download lecture slide from ITC-LMS.

Today's topic

1. Computational science, quantum computing, and data compression
2. Review of linear algebra
3. Singular value decomposition
4. Application of SVD and generalization to tensors
5. Entanglement of information and matrix product states
6. Application of MPS to eigenvalue problems
7. Tensor network representation
8. Data compression in tensor network
9. Tensor renormalization
10. Quantum mechanics and quantum computation
11. Simulation of quantum computers
12. Quantum-classical hybrid algorithms and tensor network
13. Quantum error correction and tensor network

Outline

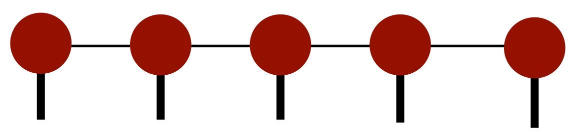
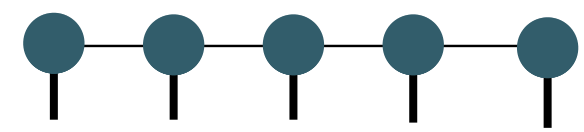
- Tensor renormalization
  - Tensor network representation of a scalar
  - Tensor renormalization
    - Tensor renormalization group (TRG)
    - Anisotropic TRG and Bond-weighted TRG
  - Tensor network renormalization around a critical point
    - Corner double line structure and short-range correlation
    - Entanglement Filtering and tensor network renormalization

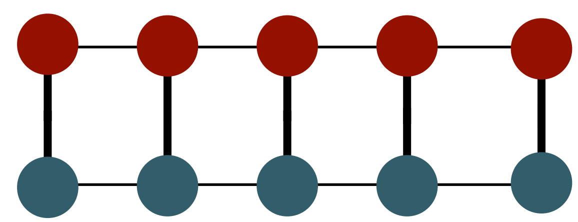
Tensor network representation of a scalar

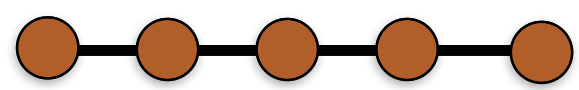
Tensor network representation of a scalar

Example: inner product of two TNSs

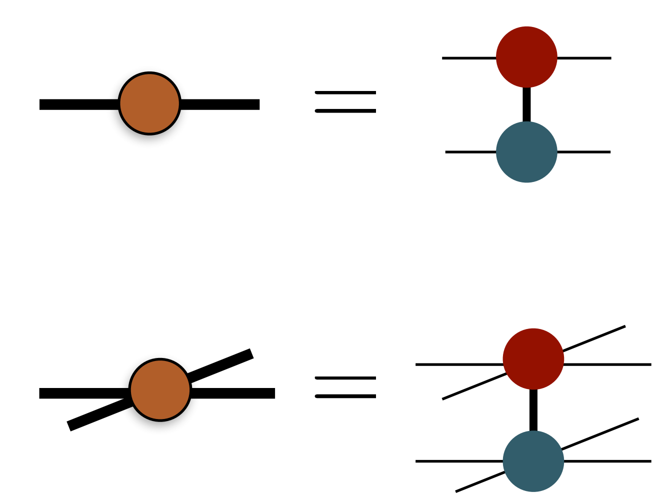
MPS

$\vec{v} =$  $\vec{u} =$ 

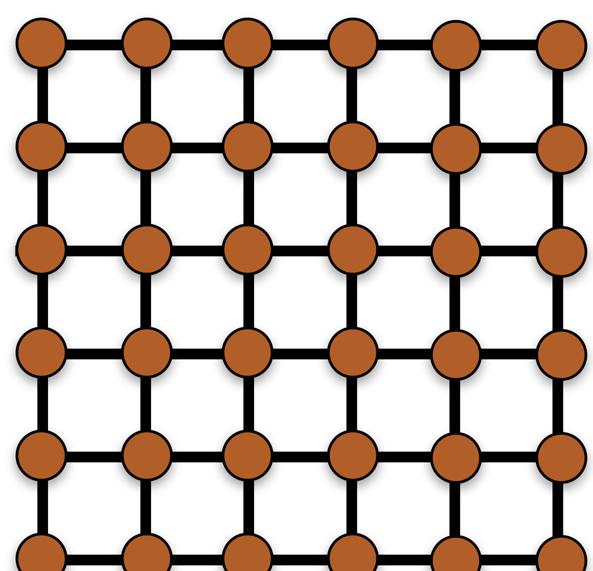
$\vec{v} \cdot \vec{u}^* =$ 

$=$ 

Double layer tensor



TPS (in two dimension)

$\vec{v} \cdot \vec{u}^* =$ 

Statistical mechanics and canonical ensemble

Canonical ensemble:
（カノニカル分布）

$$P(\Gamma) \propto e^{-\beta \mathcal{H}(\Gamma)}$$

Γ ：State (e.g. {S_1, S_2, ... S_L})

$P(\Gamma)$: Probability to appear state Γ

$\beta = \dfrac{1}{k_B T}$: Inverse temperature

$\mathcal{H}$ ：Hamiltonian

Partition function（分配関数）

＝Normalization factor of the canonical ensemble

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

Relation to the free energy in thermodynamics

$$F = -k_B T \ln Z$$

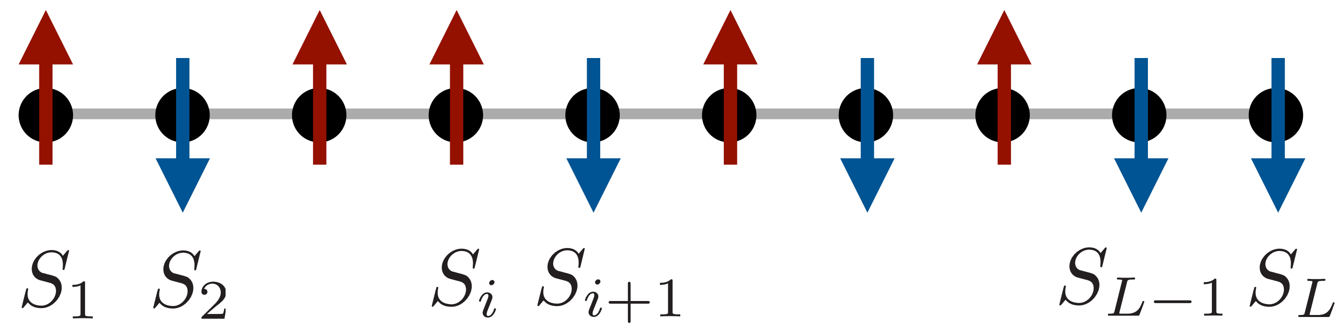
$\boxed{\text{log of the partition function＝Free energy}}$

Tensor network representation of partition function

Classical Ising model on a chain

$$\mathcal{H} = -J \sum_{i=1}^{L-1} S_i S_{i+1}$$

$$S_i = 1, -1$$



$1 = \uparrow$ $\quad -1 = \downarrow$

Partition function:

$$Z = \sum_{\{S_i = \pm 1\}} e^{\beta J \sum_i S_i S_{i+1}}$$

$$= \sum_{\{S_i = \pm 1\}} \prod_{i=1}^{L-1} e^{\beta J S_i S_{i+1}}$$

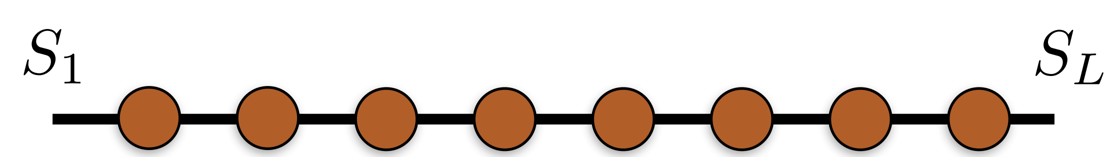
$$= \sum_{S_1 = \pm 1, S_L = \pm 1} \left(T^{L-1}\right)_{S_1, S_L}$$

Transfer matrix
(転送行列)

$$T_{S_i, S_{i+1}} = e^{\beta J S_i S_{i+1}}$$

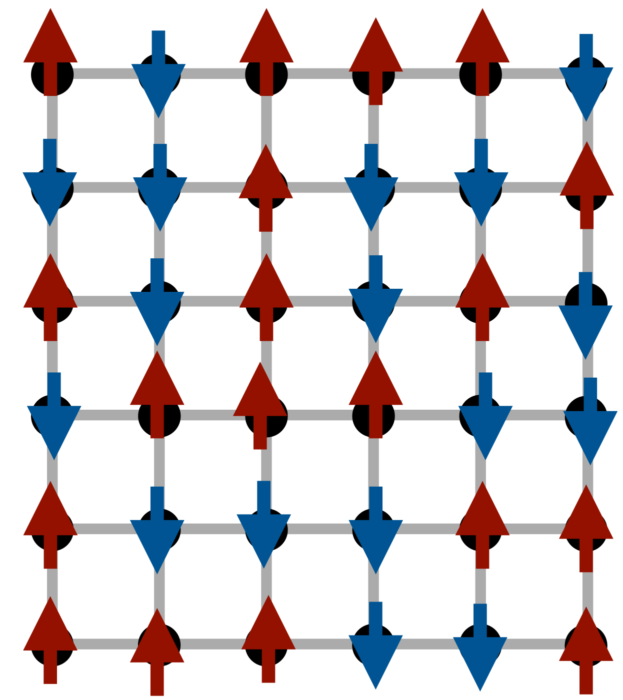
$$T = \begin{array}{c} \\ \\ \end{array} \begin{array}{cc} +1 & -1 \\ \begin{pmatrix} e^{\beta J} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J} \end{pmatrix} & \end{array} \begin{array}{c} \\ +1 \\ -1 \end{array}$$

$$\sum_{S_1 = \pm 1, S_L = \pm 1}$$



Tensor network representation in two dimension

Classical Ising model on the square lattice



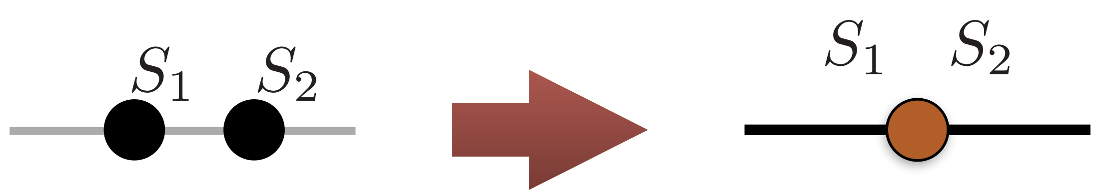
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j \quad (S_i = \pm 1 = \uparrow, \downarrow)$$

$$Z = \sum_{\{S_i = \pm 1\}} e^{\beta J \sum_{\langle i,j \rangle} S_i S_j}$$

We can use a tensor instead of the transfer matrix.

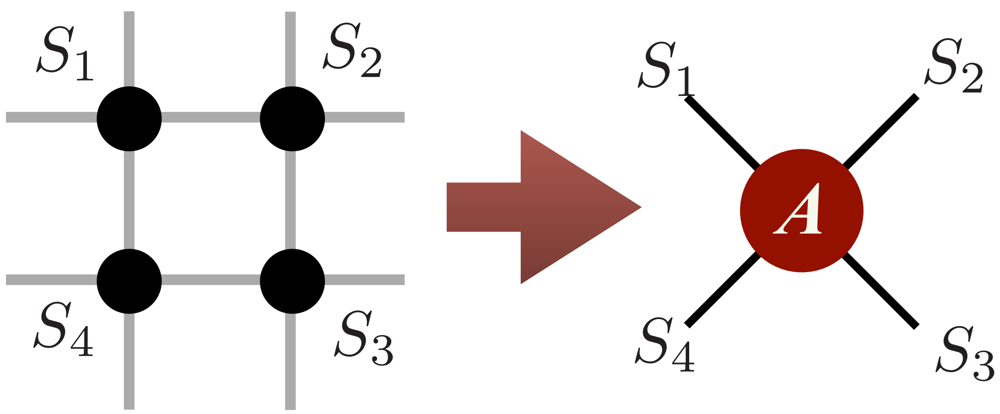
$$e^{\beta J S_1 S_2} = T_{S_1 S_2}$$

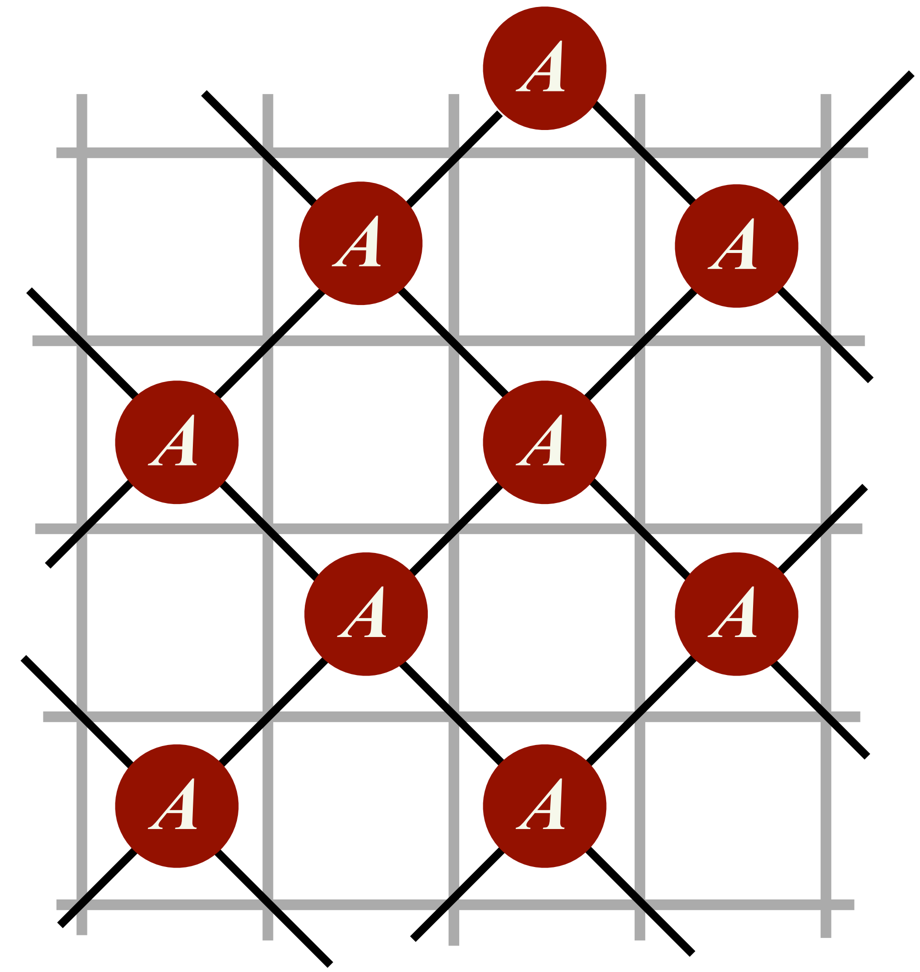
Tensor?



Tensor network representation in two dimension

$$e^{\beta J(S_1S_2+S_2S_3+S_3S_4+S_4S_1)} = A_{S_1S_2S_3S_4}$$



$Z =$ 

Partition function = Tensor network of tensor A

Square lattice Ising model→Square lattice tensor network rotating 45 degrees.

*We can construct a tensor network where tensors are on the nodes of original lattice.

Calculation cost of "classical" tensor network

Cost of tensor network contraction:

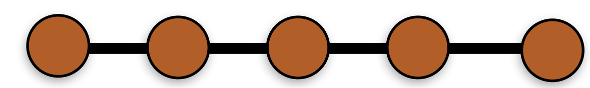
d-dimensional cubic lattice $N = L^d$

Chain: $O(ND^2)$ (Open)
$O(ND^3)$ (Periodic)

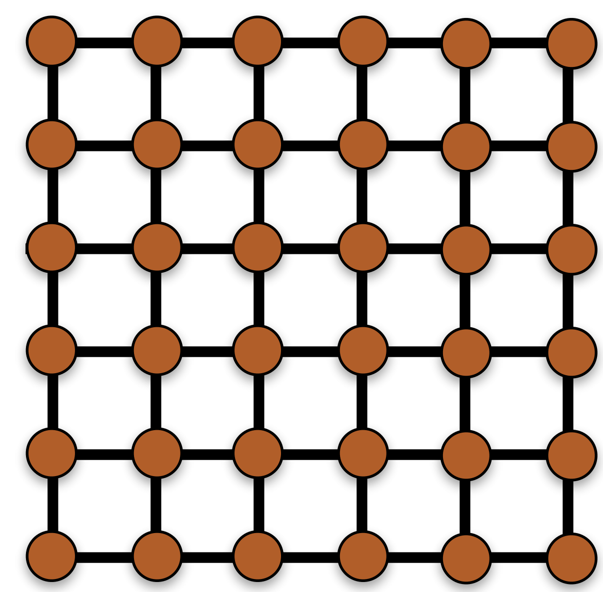
Square: $O(D^L)$ (Open)
$O(D^{2L})$ (Periodic)

d-dimensional cubic: $O(D^{L^{d-1}})$

Chain



Square lattice



It is impossible to perform exact contraction.

We need efficient approximations for the contraction.

Tensor network renormalization

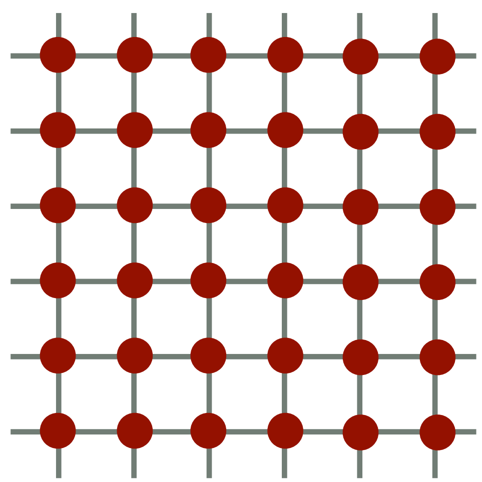
Tensor renormalization
（テンソル繰り込み）

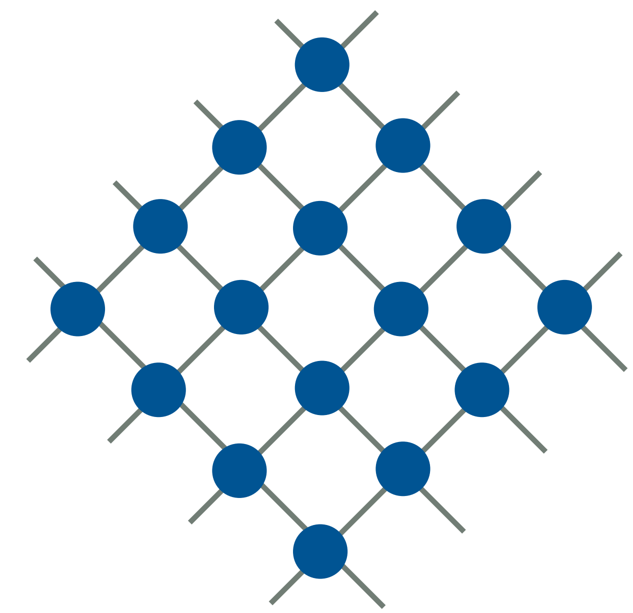
- Approximate calculation of a tensor network contraction by using "coarse graining" （粗視化）of the network
  - Coarse graining $\longleftrightarrow$ Real space renormalization
  - （粗視化） $\longleftrightarrow$ （実空間繰り込み）
- It can be applicable to (basically) any lattices, and the idea (algorithm) is independent on "models" represented by tensor networks.
  - Potential application to wide range of the science.

Outline of tensor renormalization

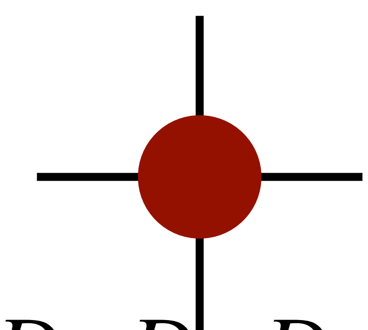
Scaler represented by $L \times L$ tensors

$(L \times L)/2$ tensors



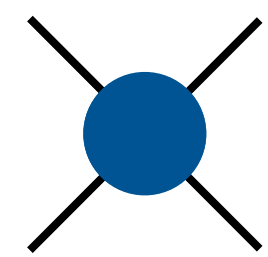


Corse graining (Renormalization) into $\sqrt{2}$ times longer scale.



A: $D \times D \times D \times D$

Approximation

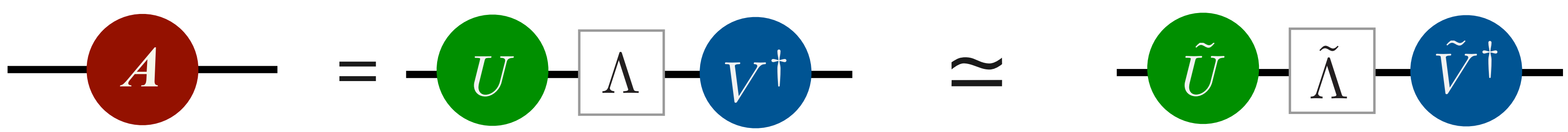


$\tilde{A}$: $D \times D \times D \times D$

Reduce the number of tensors keeping their size constant

Key technique: low rank approximation by SVD

Best low-rank approximation of a matrix = SVD



$A: M \times N$
(M ≦ N)

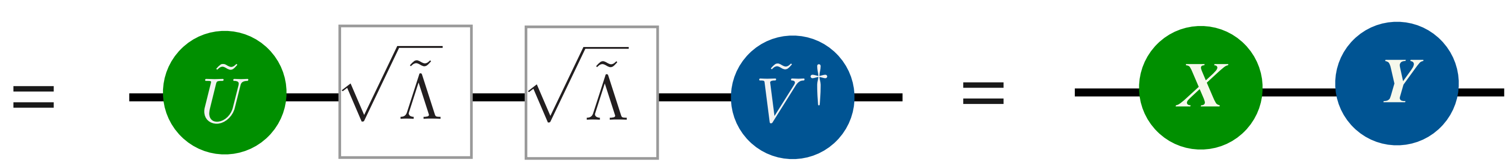
$\Lambda: M \times M$
(Diagonal matrix)

$U, V: (\mathrm{M},\mathrm{N}) \times M$

$\tilde{\Lambda}: R \times R$
(Keeping the R largest singular values)

$\tilde{U}, \tilde{V}: (\mathrm{M},\mathrm{N}) \times R$

In addition,



$\sqrt{\tilde{\Lambda}}$:Diagonal matrix
those elements are $\sqrt{\lambda}$

$X = \tilde{U}\sqrt{\tilde{\Lambda}}$ $:\mathrm{M} \times R$

$Y = \sqrt{\tilde{\Lambda}}\tilde{V}^\dagger$ $:\mathrm{R} \times M$

By SVD, we can decompose a matrix into a product of "small" matrices.

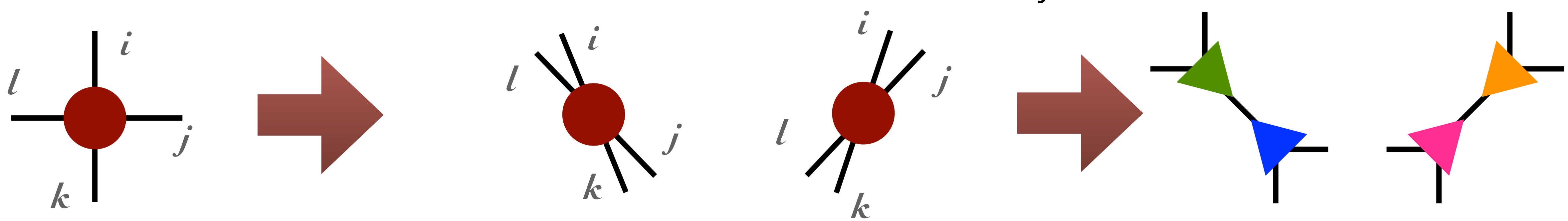
Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

1. Decomposition

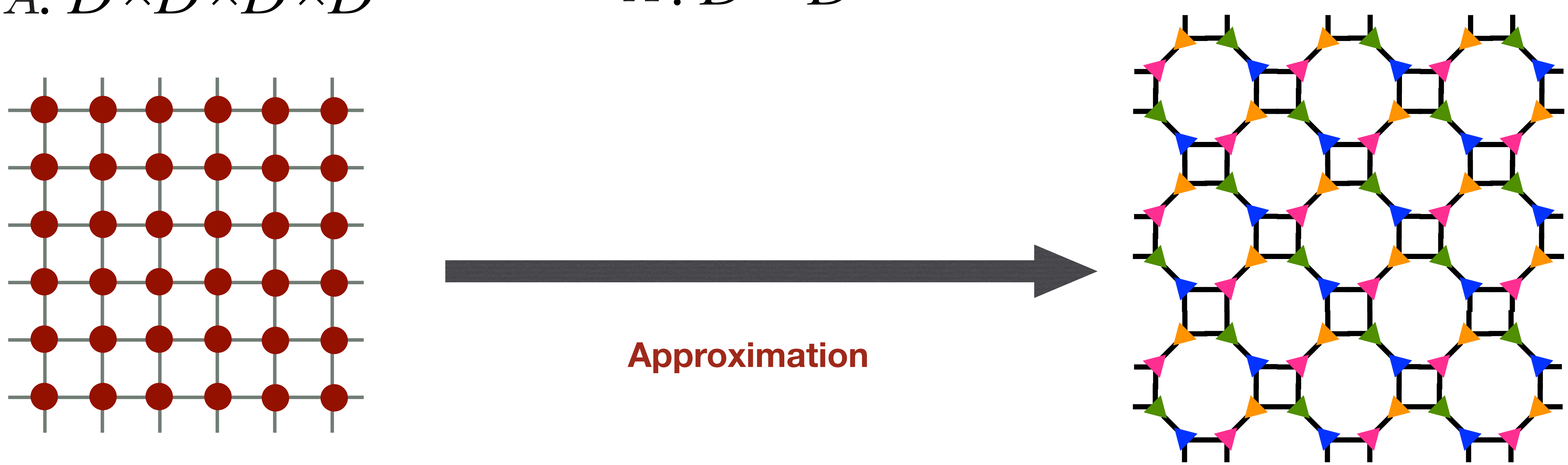
Regard a tensor as a matrix

D-rank approximation by SVD



$A_{i,j,k,l}$ $\quad A_{(i,l),(j,k)}$ $\quad A_{(i,j),(k,l)}$

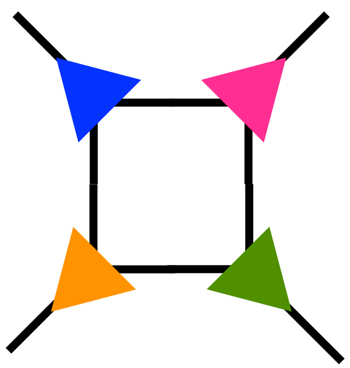
A: $D\times D\times D\times D$ $\qquad A: D^2\times D^2$



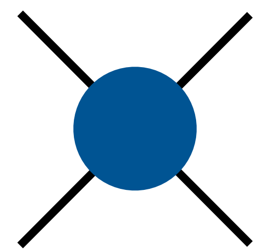
Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

2. Coarse graining

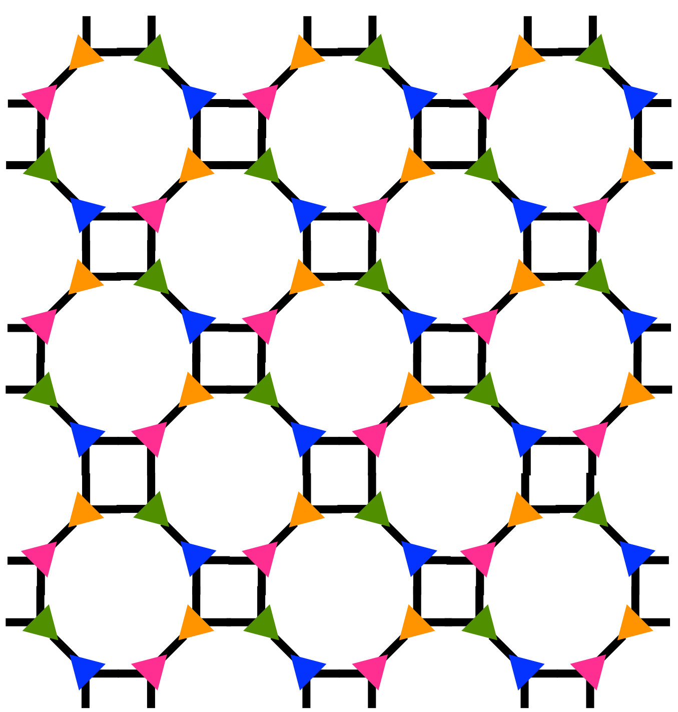


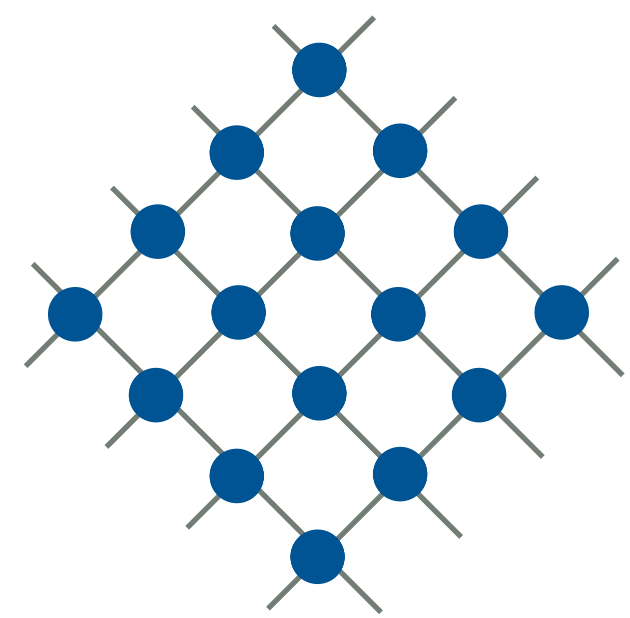
Contraction of inner indices



$$\tilde{A} : D \times D \times D \times D$$

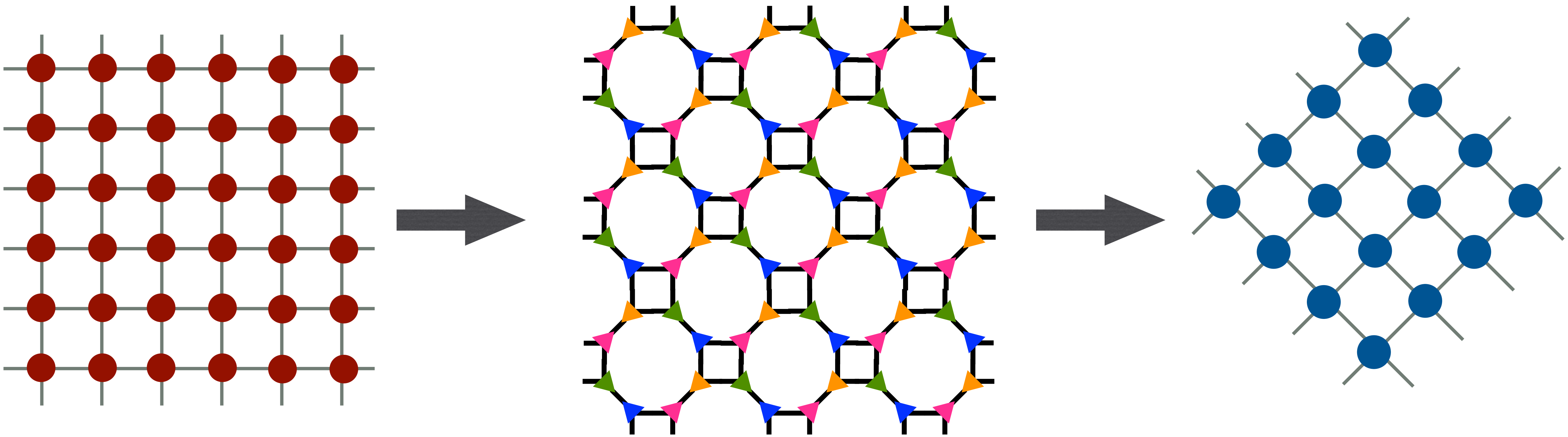
In total, two original tensors are coarse grained into a new tensor.





Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)



Calculation cost: SVD= $O(D^6)$

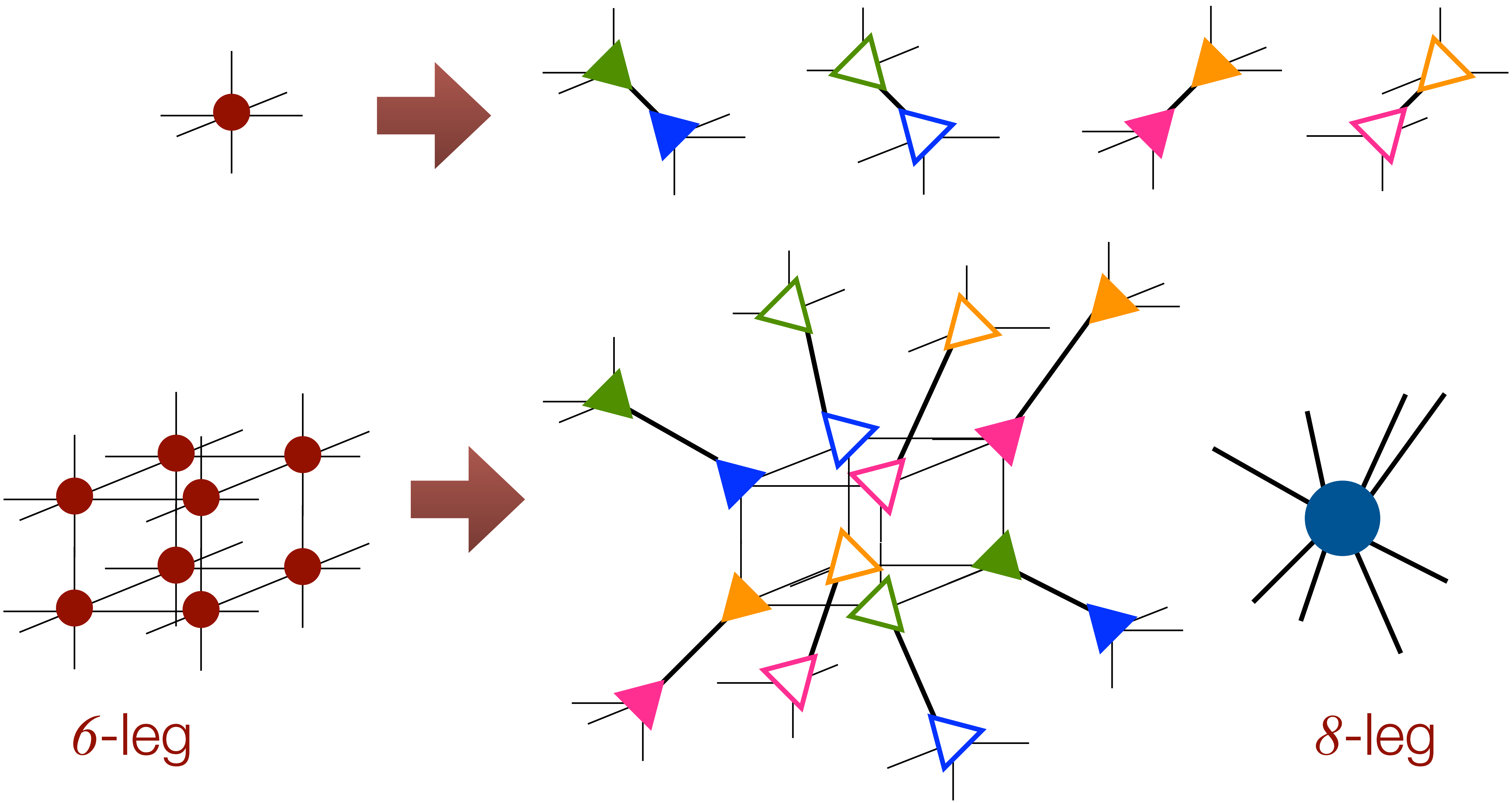
Contraction= $O(D^6)$ (per tensor)

*By one TRG step, # of tensors is reduced by 1/2.

We can calculate the contraction in polynomial cost!

Tensor renormalization group for higher dimensions

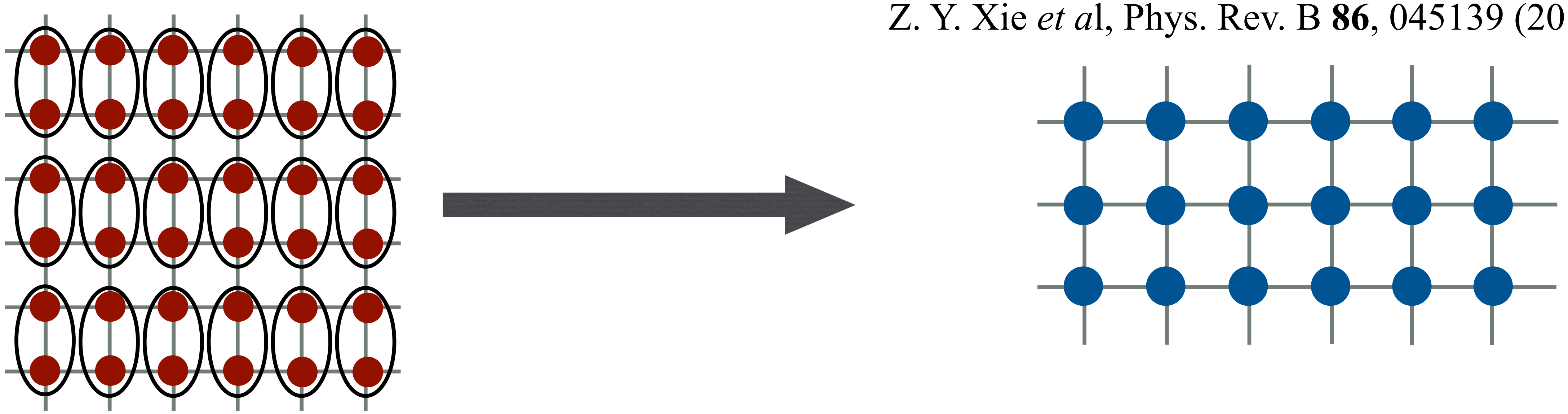
Simple generalization of TRG to cubic lattice (three dimension)



Tensor renormalization group by using HOSVD

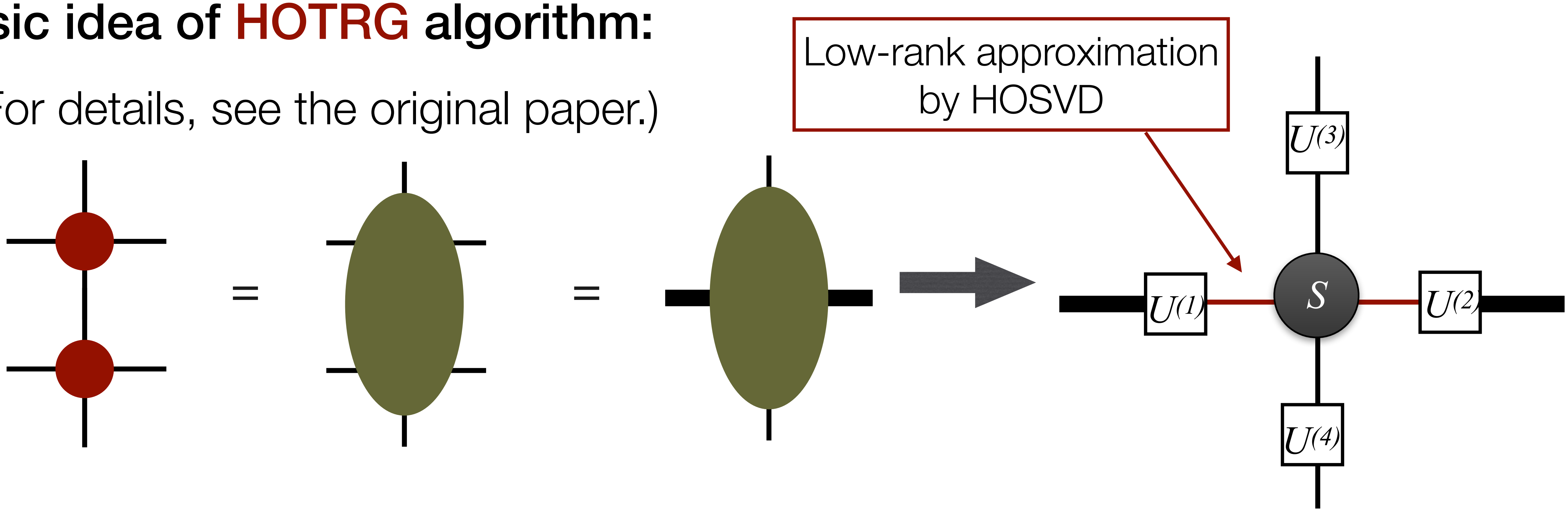
Anisotropic coarse graining by using HOSVD instead of SVD

Z. Y. Xie *et a*l, Phys. Rev. B **86**, 045139 (2012)



Basic idea of HOTRG algorithm:

(For details, see the original paper.)



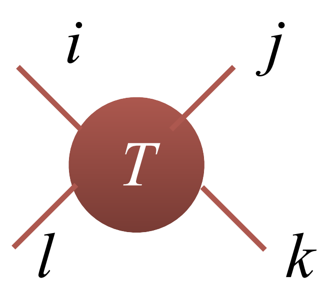
(Reference of HOSVD)

Tucker decomposition: generalization of SVD

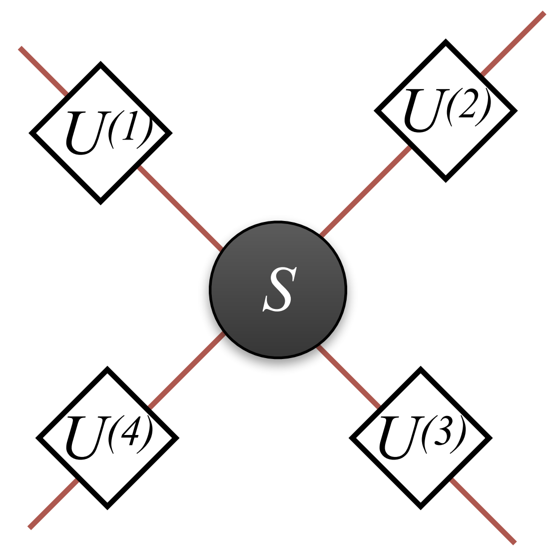
Review: T. G. Kolda et al, SIAM Review **51**, 455 (2009)

Tucker decomposition:

(Tucker (1963))



=



$U^{(i)}$: Factor matrix (usually unitary)

S :Core tensor

$$T_{ijkl} = \sum_{i'=1}^{I}\sum_{j'=1}^{J}\sum_{k'=1}^{K}\sum_{l'=1}^{L} S_{i'j'k'l'} U^{(1)}_{ii'} U^{(2)}_{jj'} U^{(3)}_{kk'} U^{(4)}_{ll'}$$

*If S is "diagonal", Tucker decomposition becomes CP decomposition.

Low "rank" approximation

$$T_{ijkl} = \sum_{i'=1}^{I'}\sum_{j'=1}^{J'}\sum_{k'=1}^{K'}\sum_{l'=1}^{L'} \tilde{S}_{i'j'k'l'} \tilde{U}^{(1)}_{ii'} \tilde{U}^{(2)}_{jj'} \tilde{U}^{(3)}_{kk'} \tilde{U}^{(4)}_{ll'}$$

$I' < I, \quad J' < J, \quad K' < K, \quad L' < L$

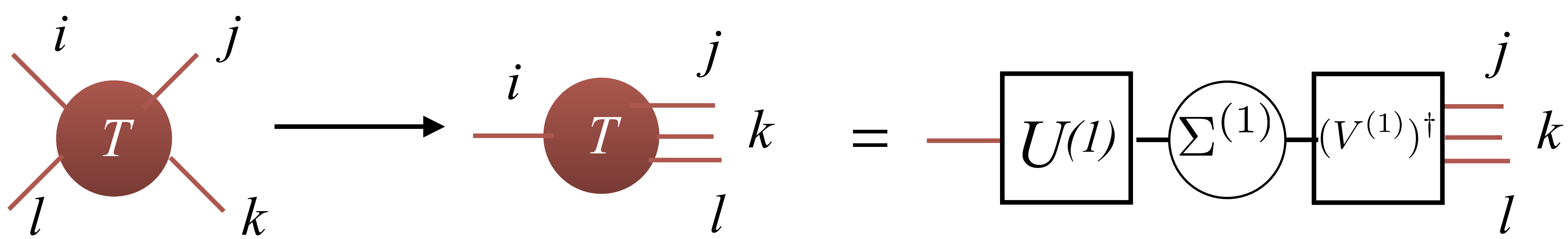
rank-(I',J', K', L') approximation

(Reference of HOSVD)

Higher order SVD (HOSVD)

L. De Lathauwer et al, SIAM J. Matrix Anal. & Appl., **21**, 1253 (2000)

Define a factor matrix from matrix SVD:



Core tensor is calculated as

$$S_{i'j'k'l'} \equiv \sum_{ijkl} T_{ijkl} (U^{(1)})^{\dagger}_{i'i} (U^{(2)})^{\dagger}_{j'j} (U^{(3)})^{\dagger}_{k'k} (U^{(4)})^{\dagger}_{l'l}$$

Properties of the core tensor

$$S^{*}_{:,i_n=\alpha,:,:} \cdot S_{:,i_n=\beta,:,:} = \begin{cases} 0 & (\alpha \neq \beta) \\ (\sigma^{(n)}_{\alpha})^2 & (\alpha = \beta) \end{cases}$$

Dot product

$$A \cdot B \equiv \sum_{i,j,k,l} A_{ijkl} B_{ijkl}$$

Generalization of the diagonal matrix Σ in matrix SVD.

* Low-rank approximation based on HOSVD is not optimal.

Power of the HOTRG

Advantage:

- HOTRG does not change the network structure.
  - We can easily generalize it to higher dimensions.
- Low-rank approximation is based on the cluster of two tensors.
  - At the approximation, we take into account more information.
  - More efficient than TRG where SVD is done for a single tensor.

Disadvantage:

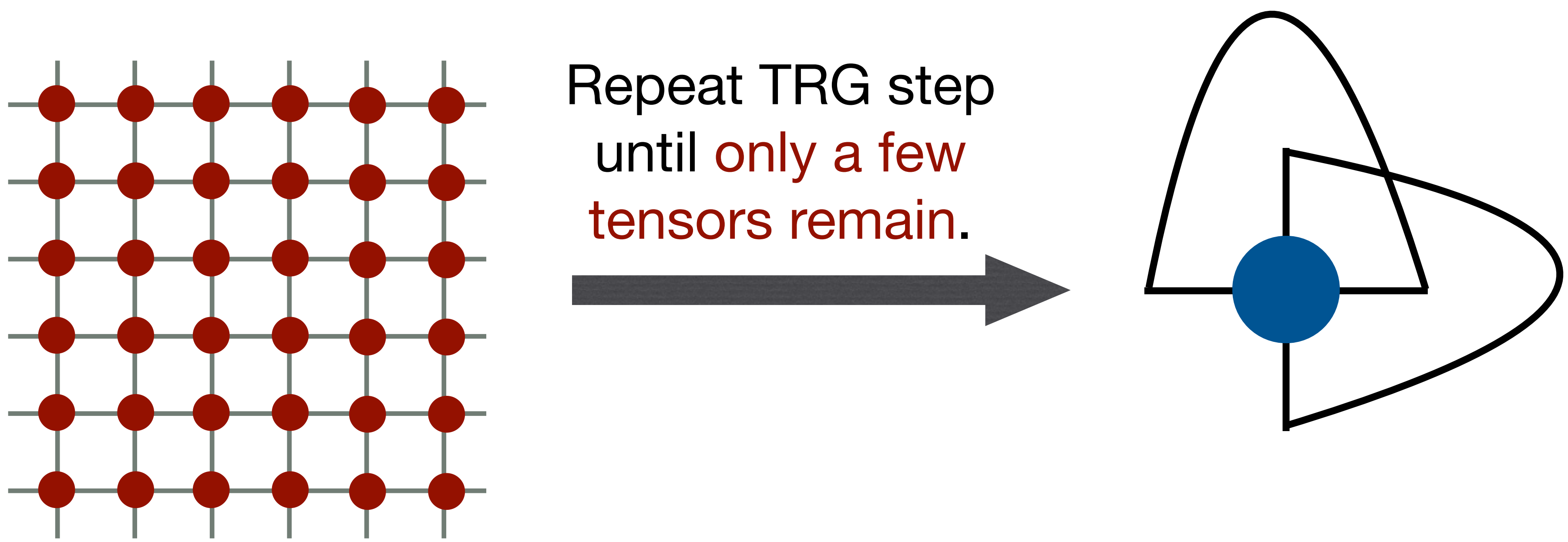
- HOTRG needs higher cost than TRG.
  - $\mathrm{O}(D^7)$ in HOTRG $\longleftrightarrow$ $\mathrm{O}(D^6)$ in TRG

Application to a classical partition function

Partition function

(Periodic boundary condition)

$Z =$



We can easily calculate physical quantities from Z.

Free energy: $F = -k_B T \ln Z$

Energy: $E = -\dfrac{\partial \ln Z}{\partial \beta}$

Specific heat: $C = \dfrac{1}{k_B T^2} \dfrac{\partial^2 \ln Z}{\partial \beta^2}$

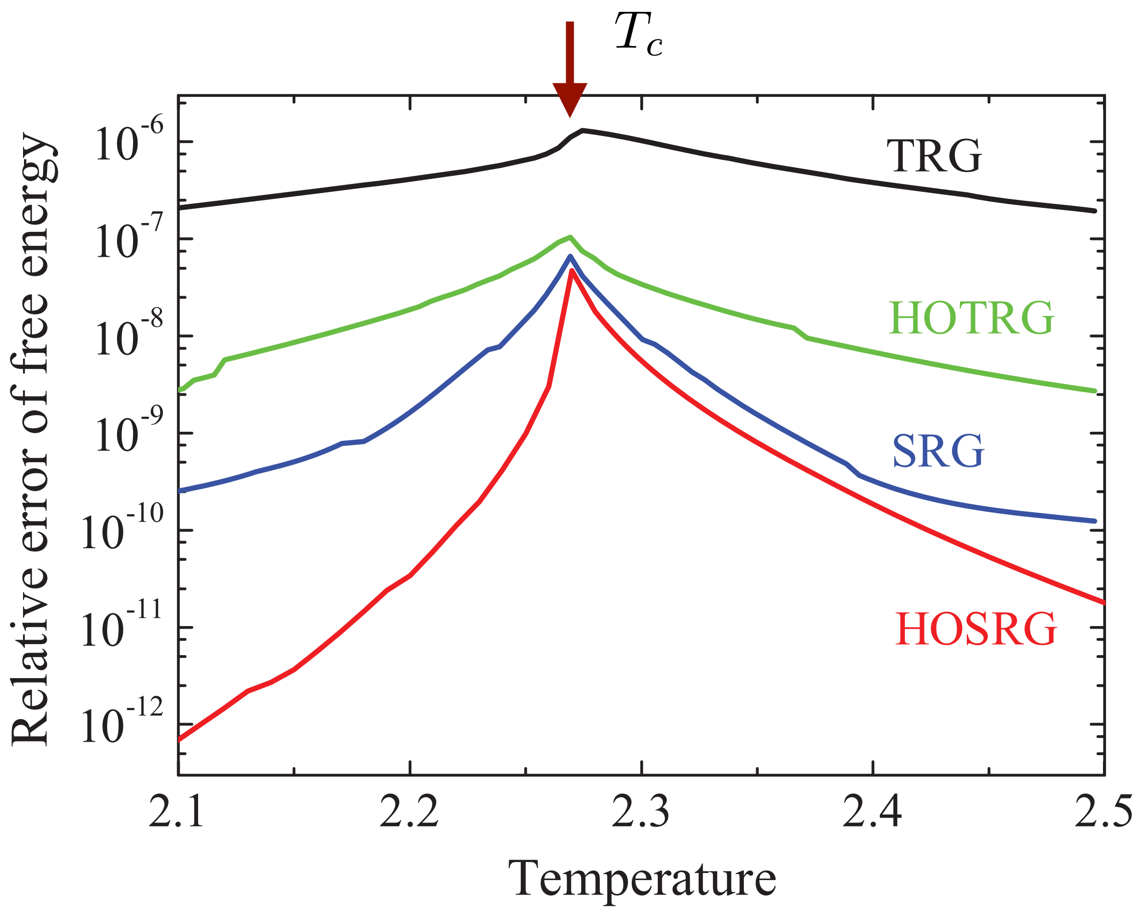
Example of calculation

Ising model in infinite size

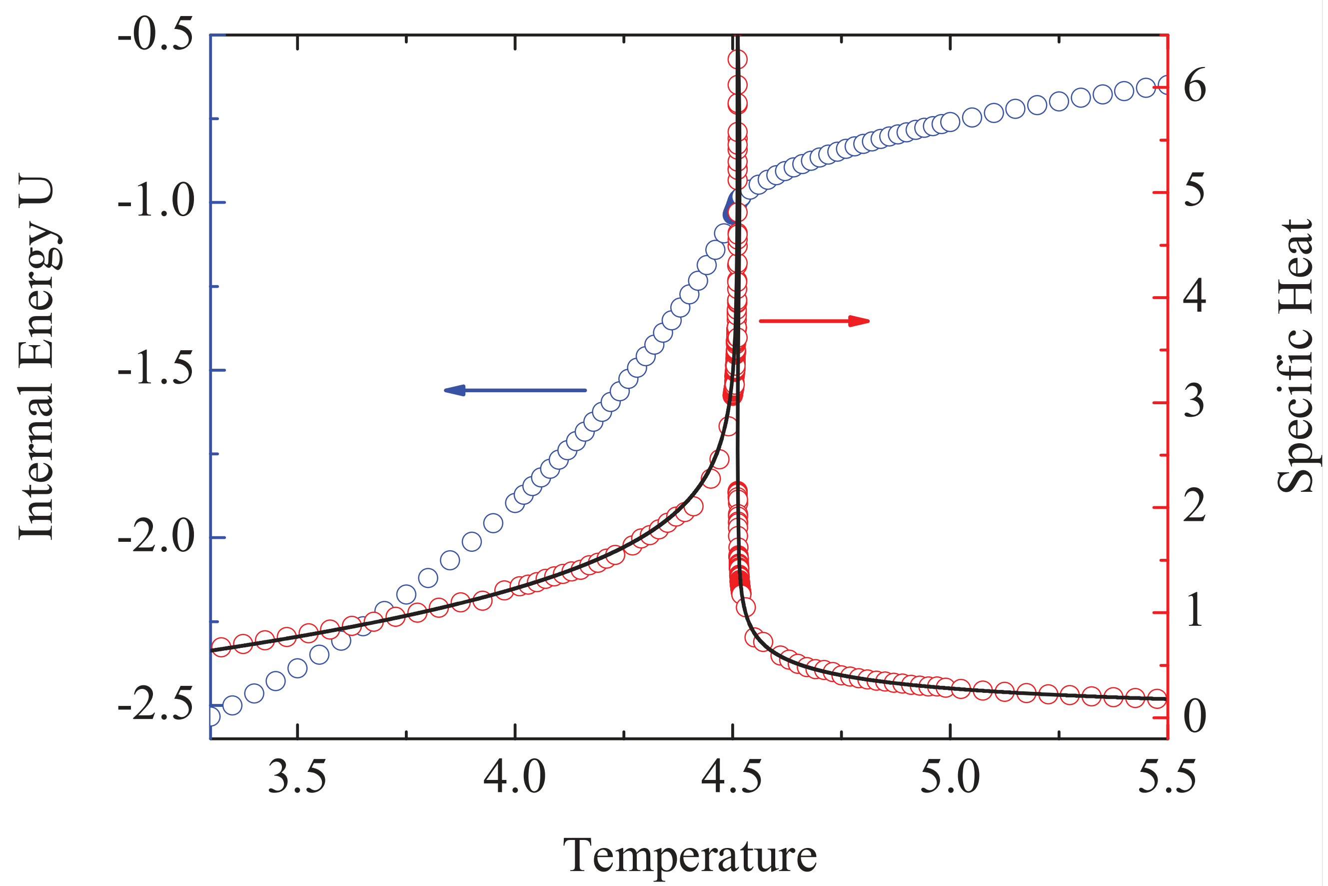
$$\mathcal{H} = -J\sum_{\langle i,j\rangle} S_i S_j$$

Z. Y. Xie *et a*l, Phys. Rev. B **86**, 045139 (2012)

Error of free energy for 2D Ising model



Energy and specific heat of 3D Ising model



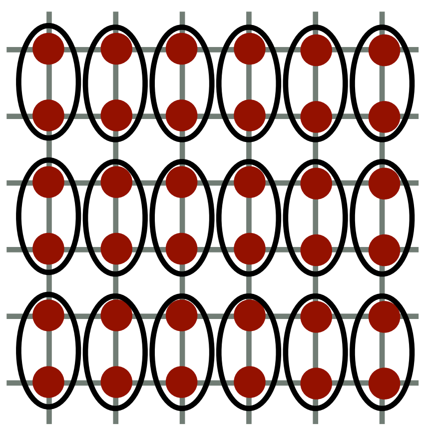
$$T_c/J = \frac{2}{\ln\left(1+\sqrt{2}\right)} \simeq 2.269$$

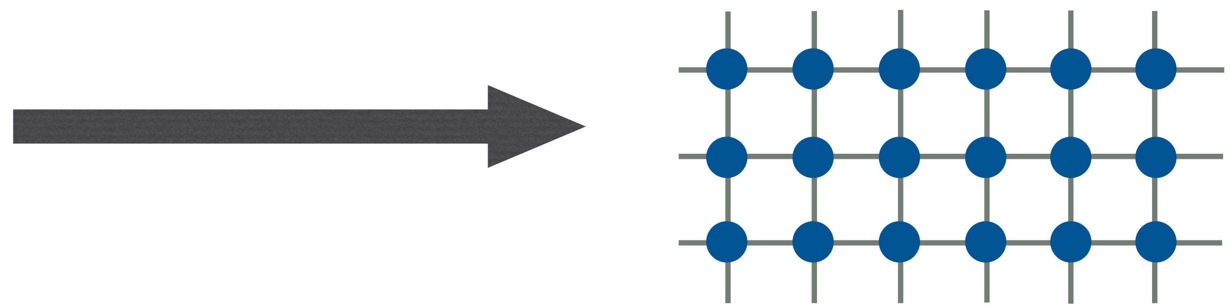
Recent improvements on TRG

Higher Oder Tensor Renormalization Group (HOTRG)

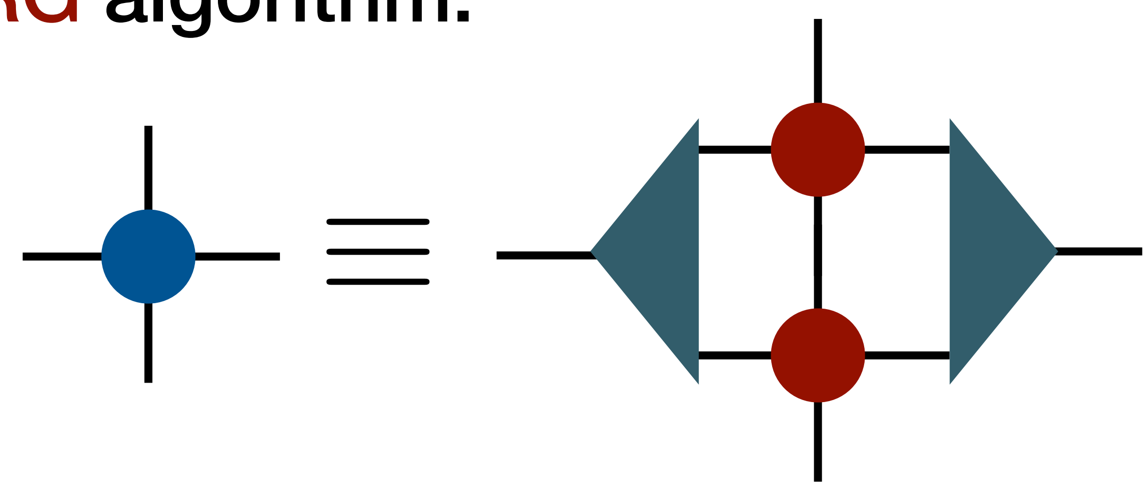
Anisotropic coarse-graining by using HOSVD instead of SVD

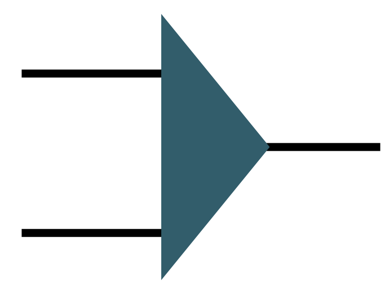
Z. Y. Xie *et a*l, Phys. Rev. B **86**, 045139 (2012)



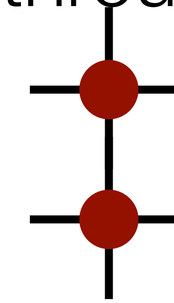


HOTRG algorithm:





Isometry is defined through HOSVD of



Better accuracy than TRG, although,

$\boxed{\text{Computation cost: } O(D^7) > O(D^5) \quad \text{(TRG)}}$

Application to high dimensions

Interests in

- 3d classical systems
- 2d and 3d quantum systems
- Much higher dimensions...

→ We want to perform tensor network RG for high dimensions!

However,

TRG: Not easy to generalize to high dimensions.

HOTRG: Easy to generalize to high dimensions, but its cost is $\underline{O(D^{4d-1})}$

$d=3 : O(D^{11})$

$d=4 : O(D^{15})$

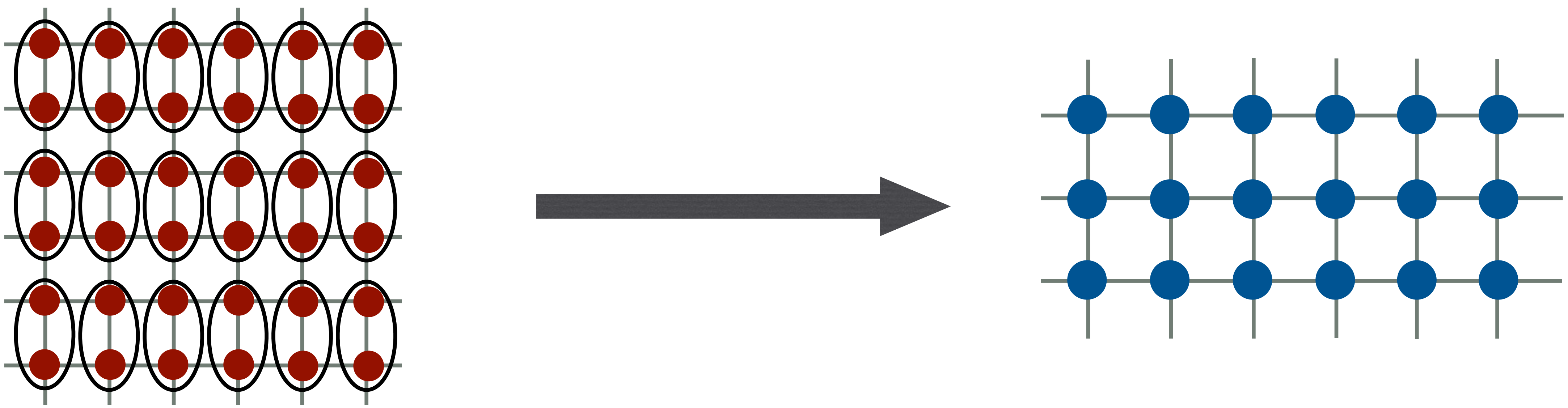
Is it possible to construct lower cost algorithm ?

Anisotropic TRG = ATRG

D. Adachi, T. Okubo and S. Todo, Physical Review B **102**, 054432 (2020).

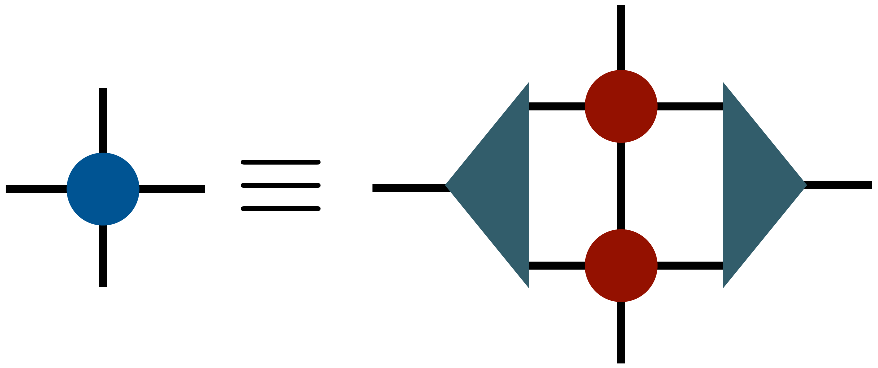
Central idea of Anisotropic TRG

In ATRG, we coarse-grain tensors anisotropically as HOTRG:

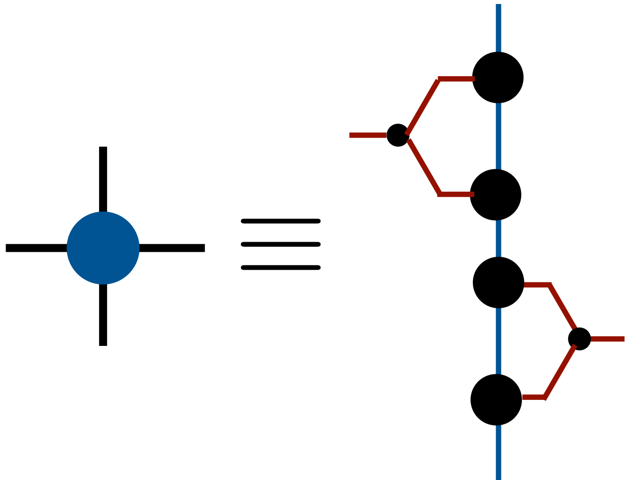


In order to reduce the computation cost, we decompose the local tensor into small pieces before performing coarse-graining.

HOTRG $O(D^7)$



ATRG $O(D^5)$

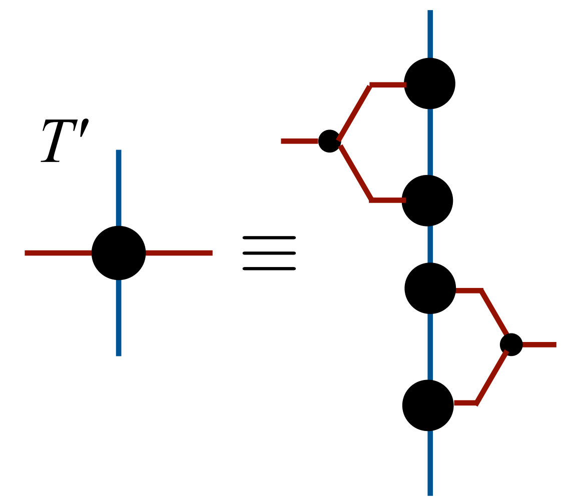


Summary of 2D ATRG

Memory storage: $O(D^3)$

* We do not explicitly create 4-leg tensor.

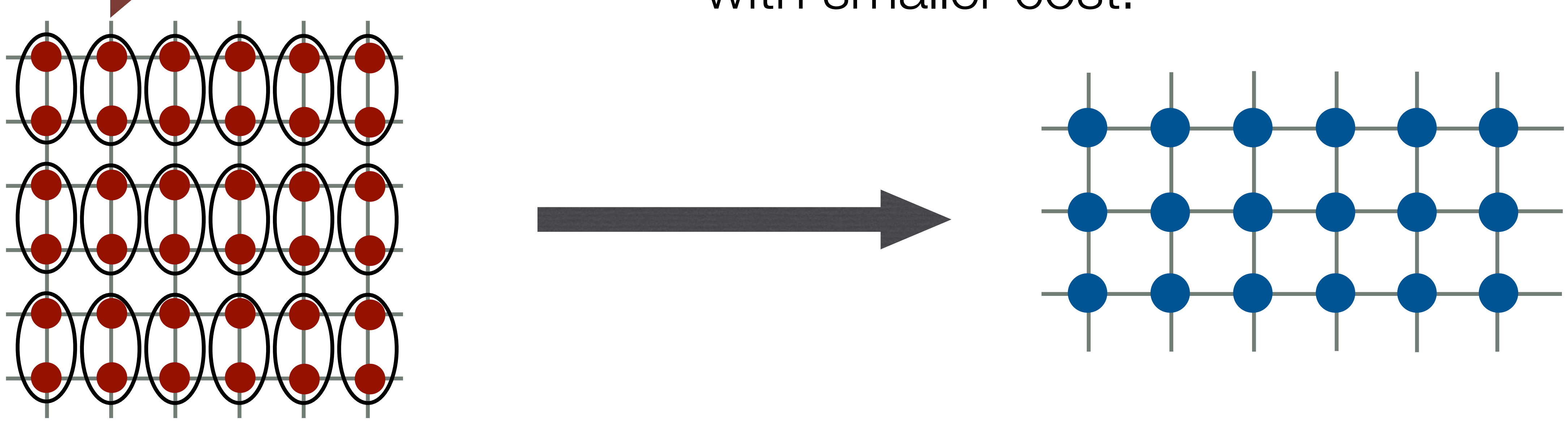
(We need only 3-leg tensors!)



Computation time: $O(D^5)$

* By using *partial SVD* technique, such as the Arnoldi method, we can reduce SVD cost.

We can perform HOTRG like anisotropic coarse-graining with smaller cost!



Benchmarks

D. Adachi, T. Okubo, and S. Todo, PRB (2020)

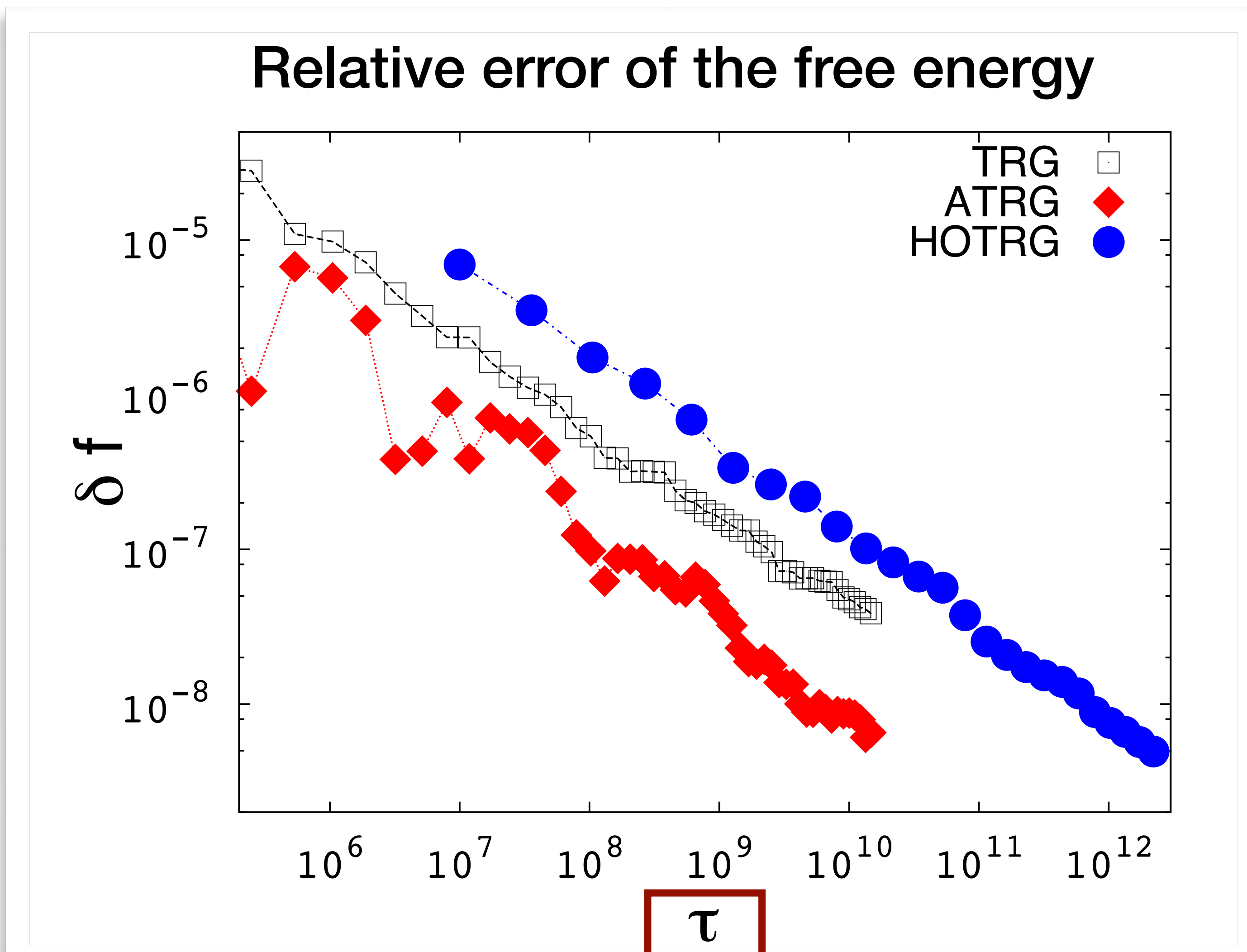
The computation costs are different between ATRG and HOTRG.

TRG, ATRG: $O(D^5)$

HOTRG: $O(D^7)$

Leading order computation time:

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



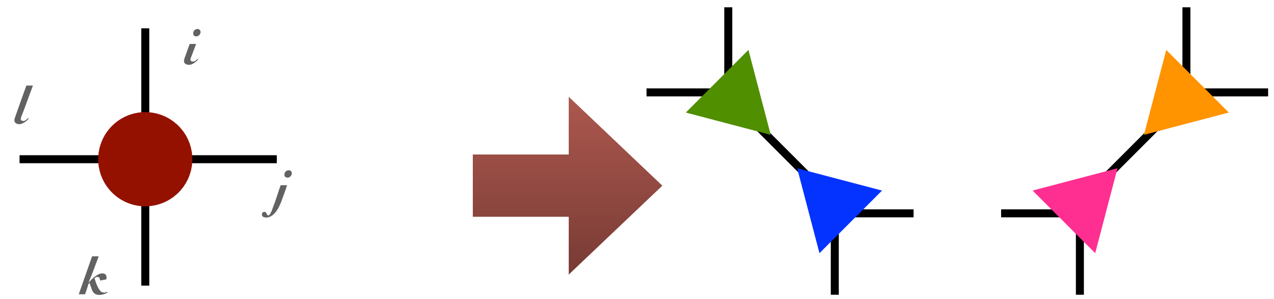
ATRG is the best!

(ATRG is ~10^2 faster than HOTRG!)

Bond-weighted TRG = BTRG

D. Adachi, T. Okubo, and S. Todo, Phys. Rev. B, **105**, L060402 (2022).

Disadvantage of local SVD truncation



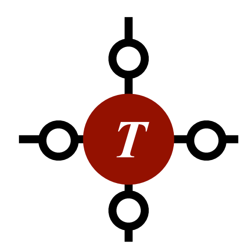
$A_{i,j,k,l}$

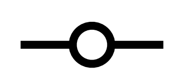
Truncation by SVD

Local SVD is not necessarily the best "D-rank" approximation
for the whole TN (the partition functions).

We need to consider the "environment"
for better truncations.

As an approximated environment, we may use "weights" of bonds as
a mean-field environment.

$\tilde{T} =$ 

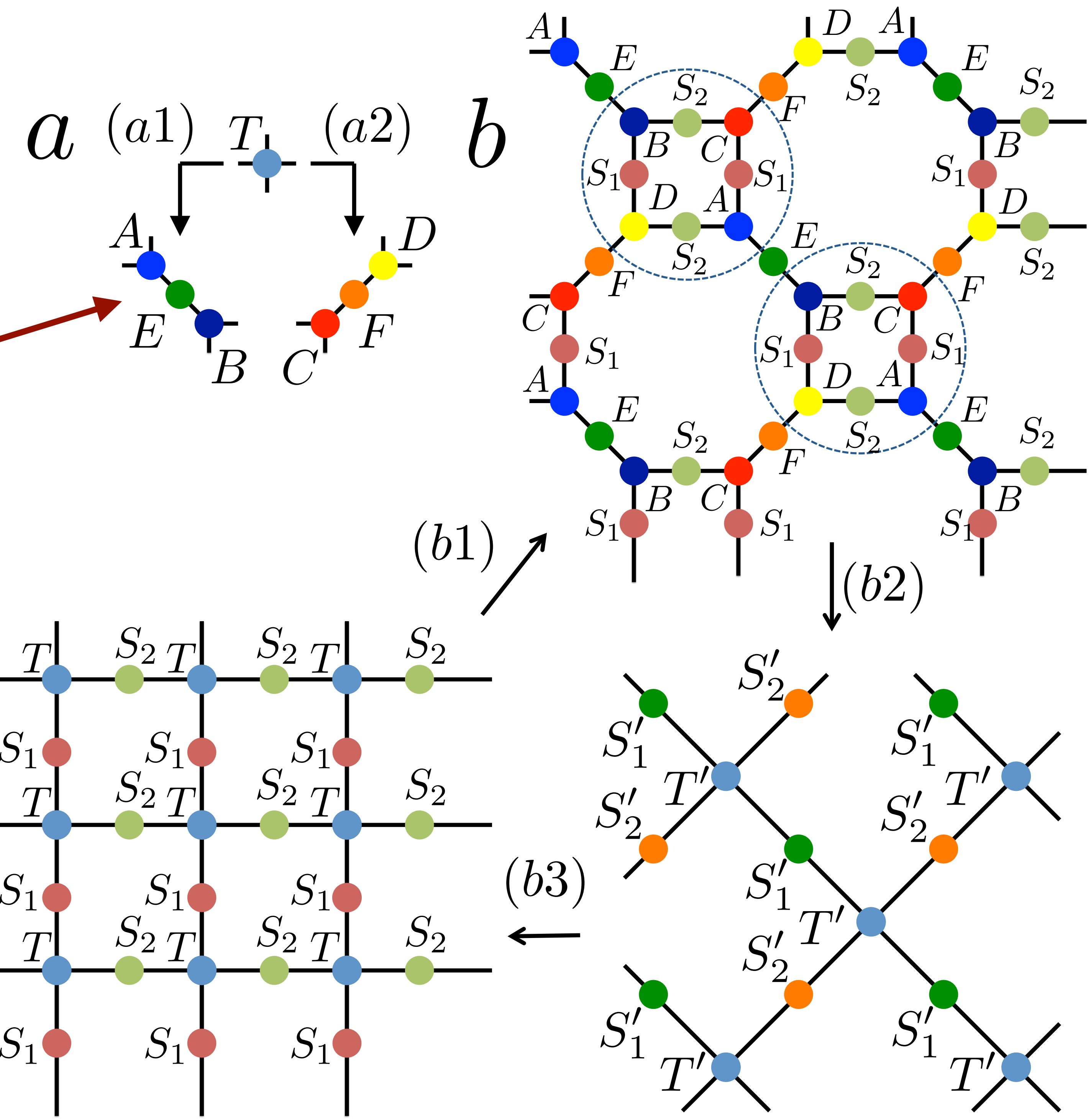
 : Diagonal non-negative matrix

*You may find a similarity to the TEBD algorithm in MPS.

Recipe of BTRG:

By using SVD,
we divide T into three parts.

$$T = U\sigma V^\dagger$$
$$= (U\sigma^{\frac{1-k}{2}})\sigma^k(\sigma^{\frac{1-k}{2}}V^\dagger)$$



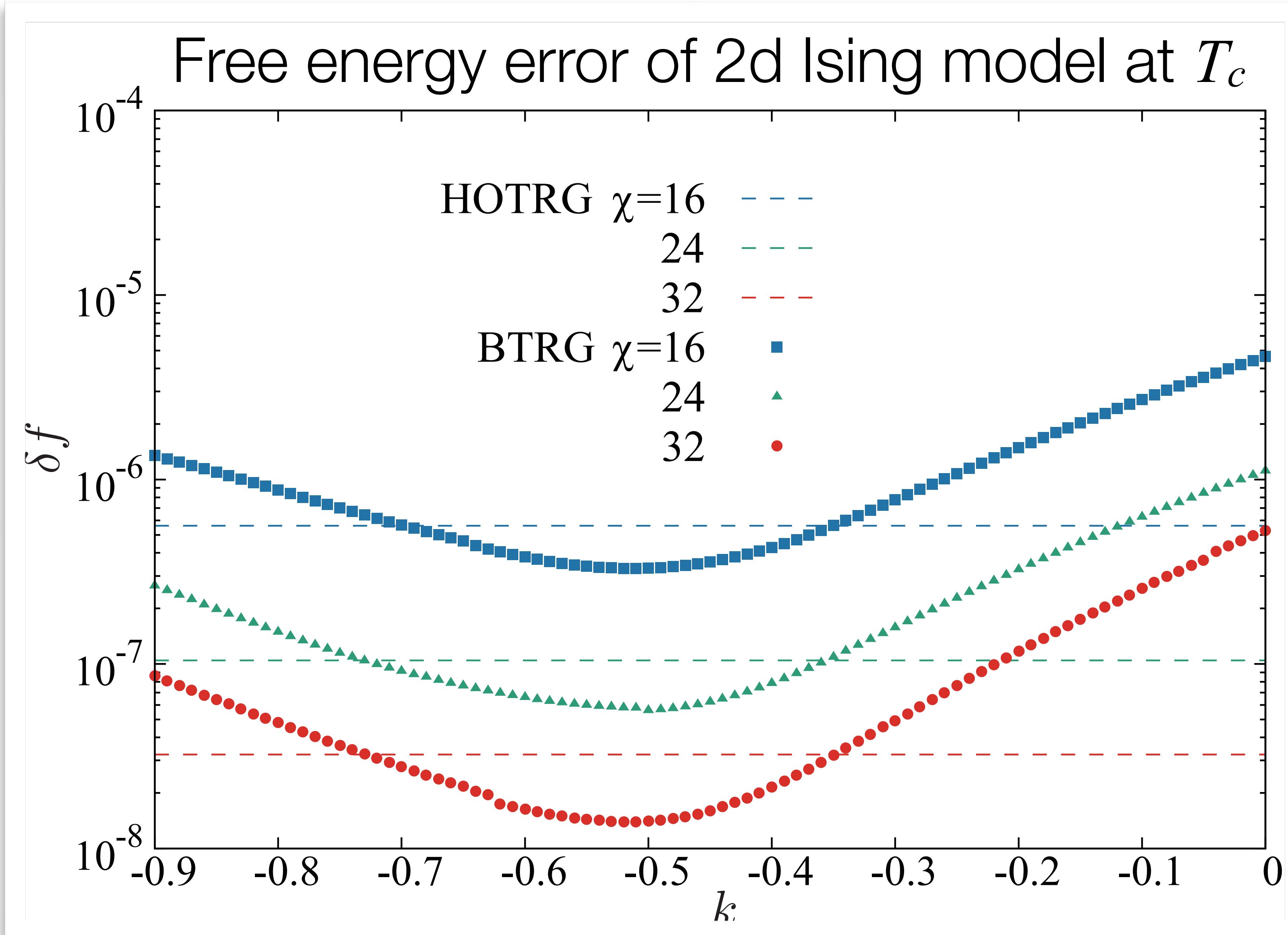
σ^k : bond weight S

$k = 0$: Conventional TRG

$k < 0$: New tensor contains mean-field environment

It might improve the accuracy without increase of cost.

Benchmark



We see

1. At $k=-0.5$, the error takes its minimum.
2. Around the minimum, accuracy is better than HOTRG.

Tensor network renormalization at the critical point

— When the accuracy of TRG becomes worse?

Correlation (entanglement) within a tensor

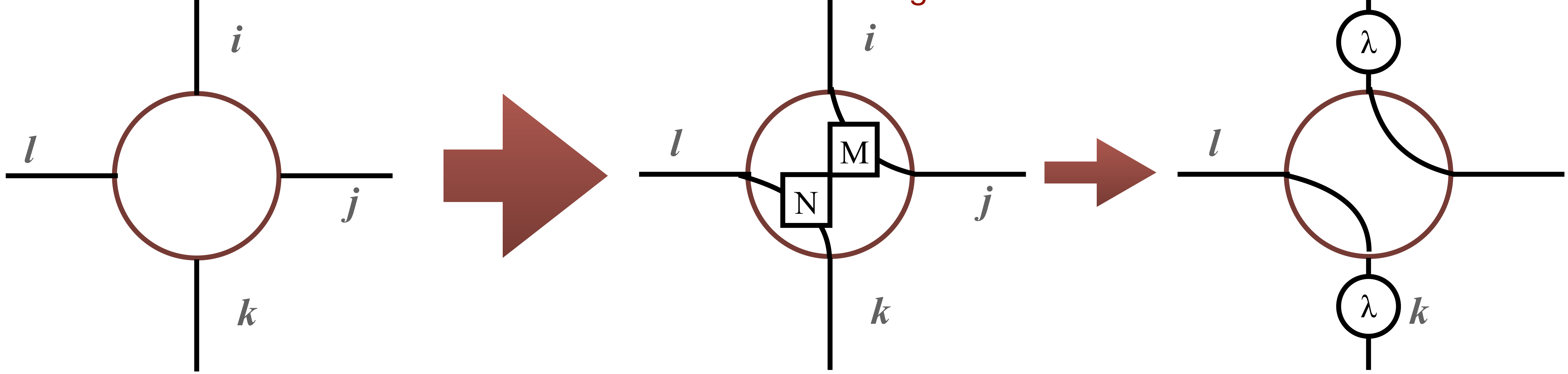
General tensor

$$A_{ijkl}$$

Eg. *Correlation* **in (i,j) and (k,j)**

$$A_{ijkl} = M_{ij} N_{kl} \quad \Rightarrow \quad A_{ijkl} = \lambda_i^{(M)} \lambda_k^{(N)} \delta_{ij} \delta_{kl}$$

Diagonalize M and N matrices.



New rule for representation of the correlation:

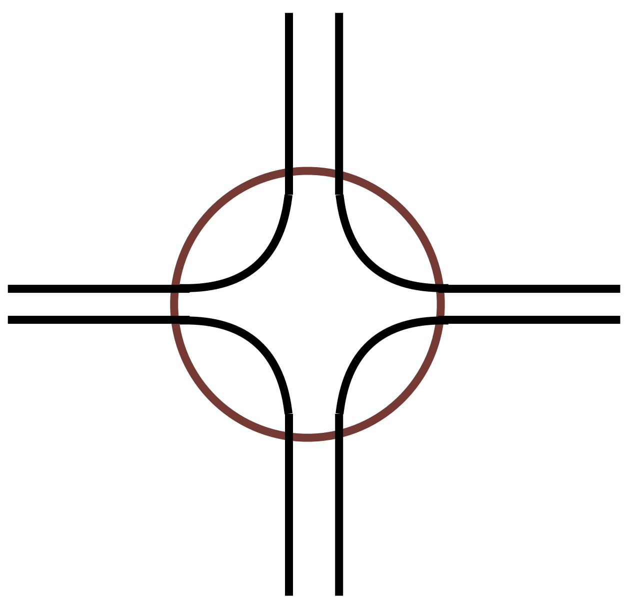
$$i \;\text{———}\; j = \delta_{ij}$$

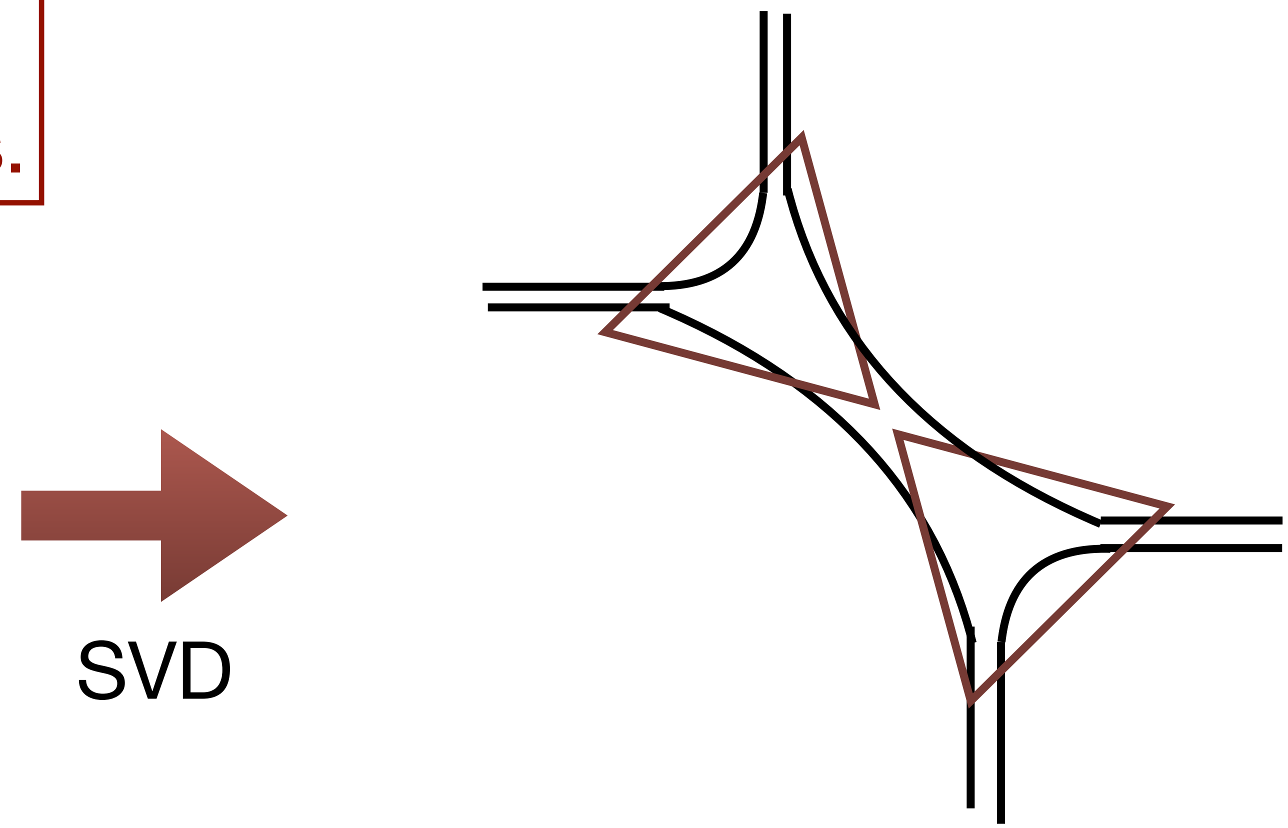
(+ we neglect eigenvalues in the diagram.)

Fixed point of TRG: Corner Double Line tensor (固定点)

Corner Double Line (CDL) tensor:

There are correlations among the nearest legs.





Original bond dimension = D

➡ Single line: bond dimension $\sqrt{D}$

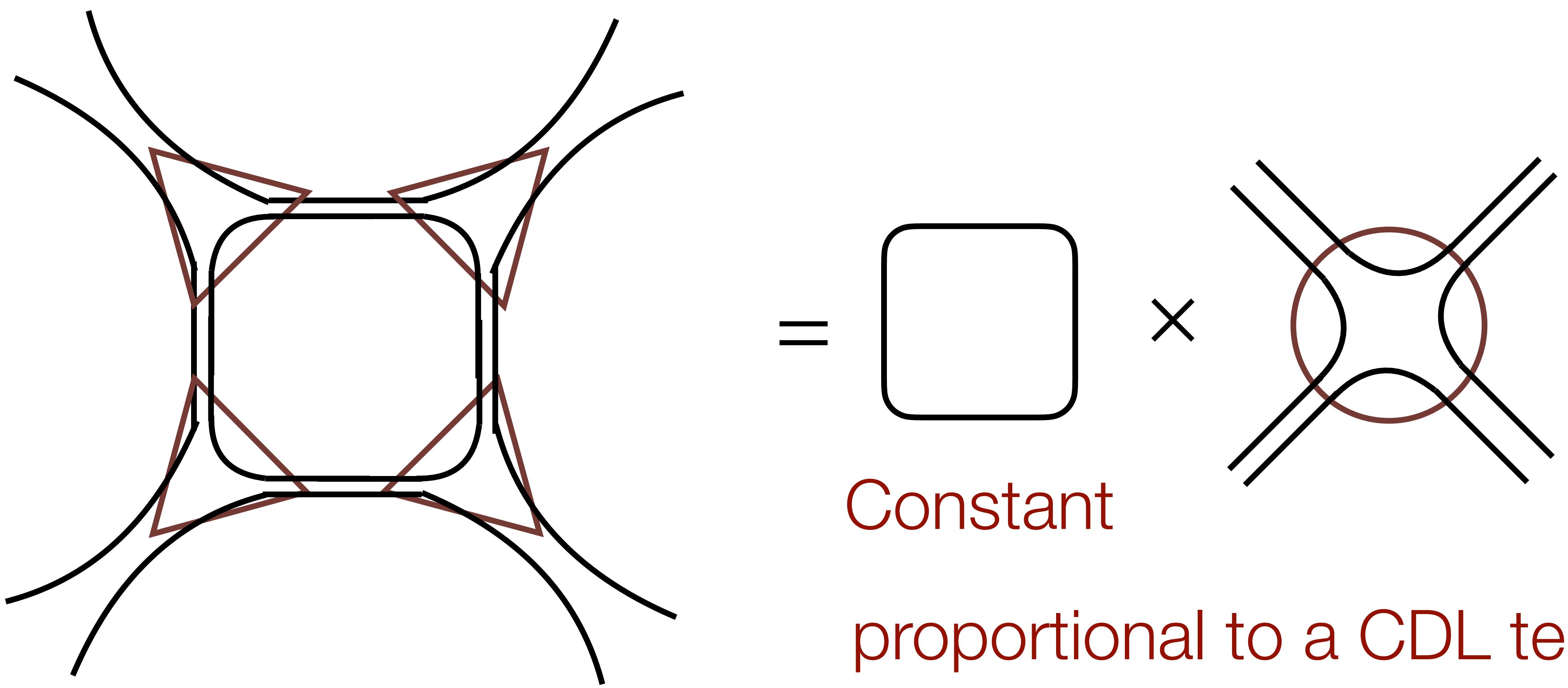
Degree of freedoms connecting two tensors.

Two lines = D

➡ No truncation error at SVD (Original rank = D)

Fixed point of TRG: Corner Double Line tensor

Contraction of four tensors in TRG:



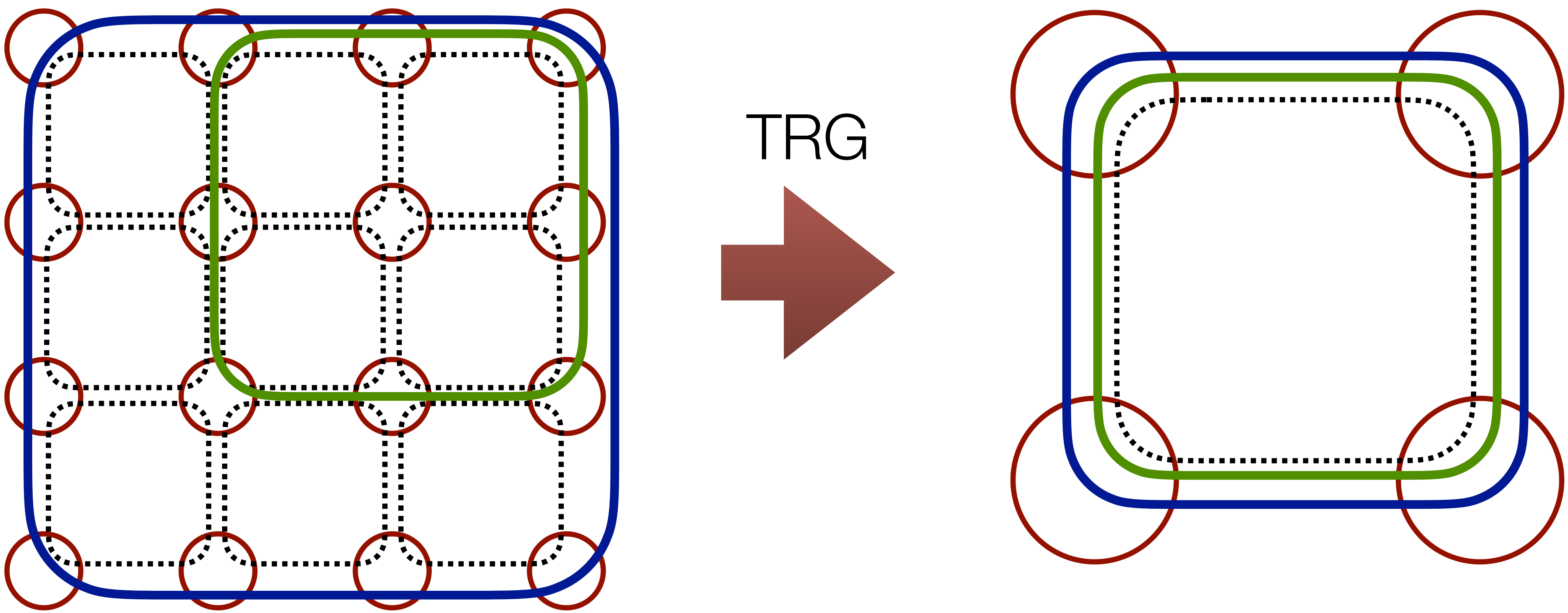
CDL tensor is a fixed point of TRG (and also HOTRG).

CDL tensor remains as CDL tensor along TRG.

Problems in TRG: accumulation of correlations

Correlation in several scales

Correlations remains after TRG.



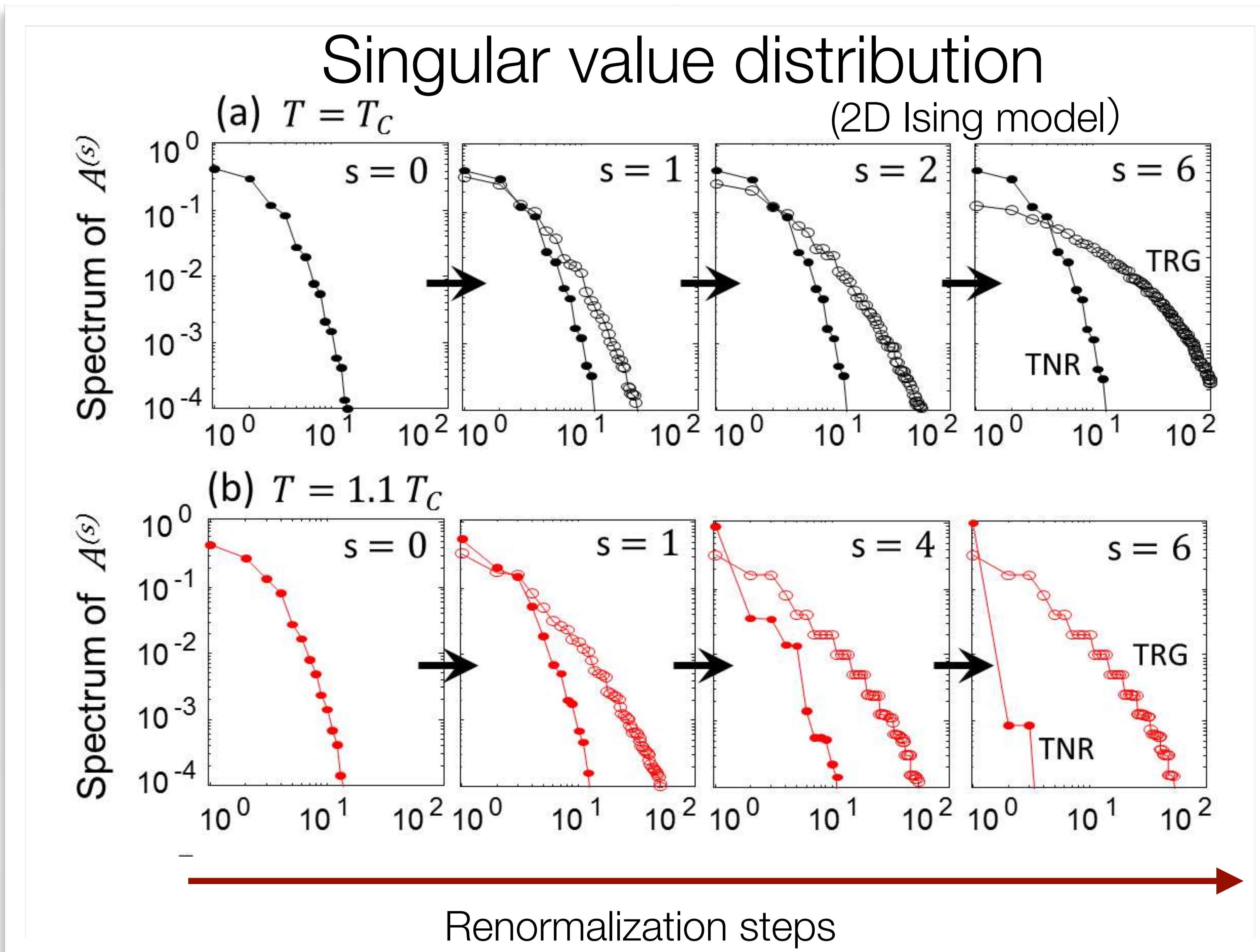
Ideal renormalization:

Correlation in shorter scales should be removed.
Only the correlation in the present scale exists.

TRG：

Correlations in all scales remain.

Problem in TRG: increase of truncation error



G. Evenbly and G. Vidal
Phys. Rev. Lett. **115**, 180405 (2015)

In TRG, the width of the singular value distribution increases along renormalization.

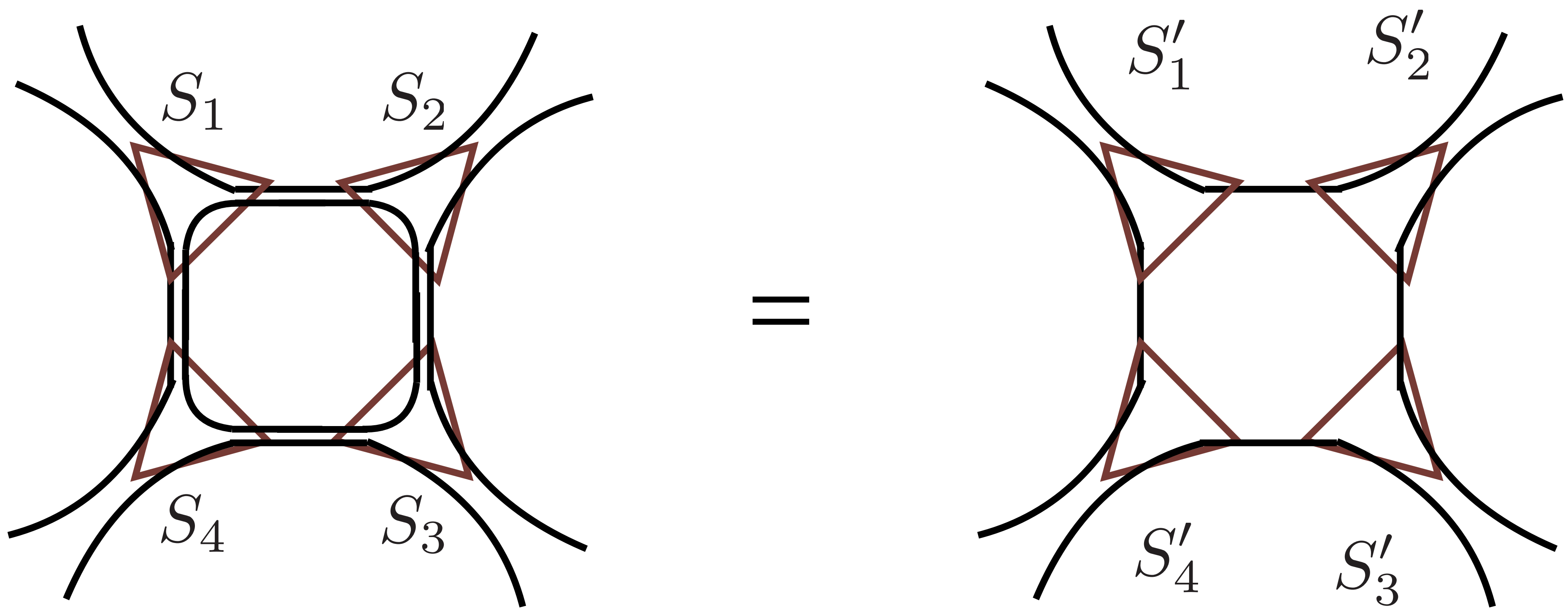
Increase of truncation error (decrease of accuracy)

Improvement of TRG : Entanglement Filtering

Try to remove CDL structure at renormalization steps.

Z.-C. Gu and X.-G Wen, Phys. Rev. B 80, 155131 (2009)

Idea:



S: $D \times D \times D$

S': $D \times D' \times D'$

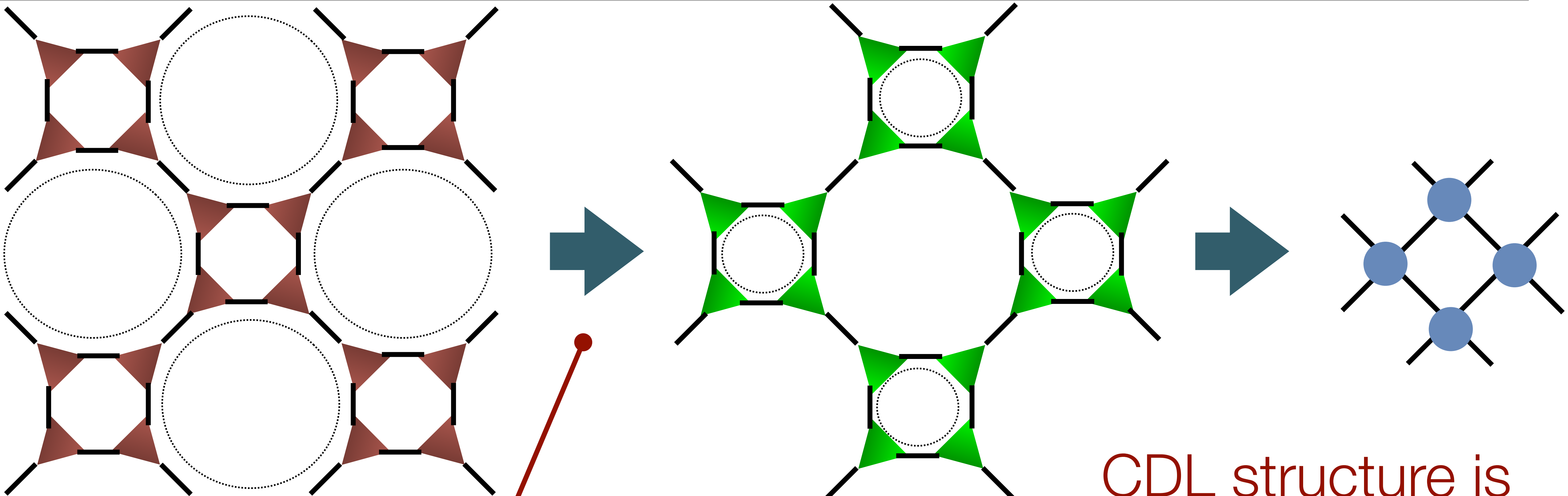
$$\mathbf{S'} = \square^{1/4}\,\mathbf{S}$$

$$D' \sim \sqrt{D}$$

Insert this "approximation" into the TRG algorithm.

Tensor Entanglement Filtering Renormalization

Z.-C. Gu and X.-G Wen, Phys. Rev. B 80, 155131 (2009)

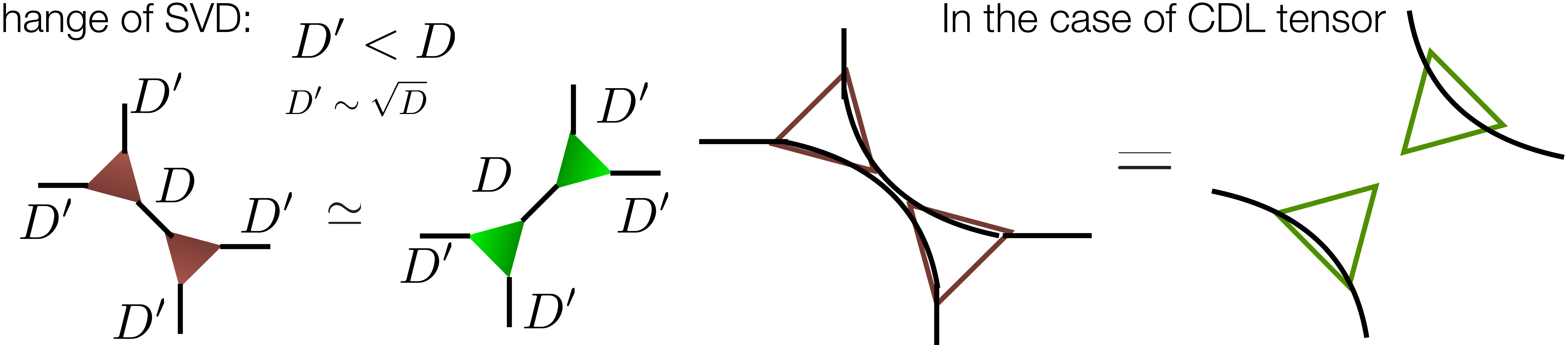


S' (after approximation)

CDL structure is perfectly removed!

Change of SVD: $D' < D$

$D' \sim \sqrt{D}$



In the case of CDL tensor

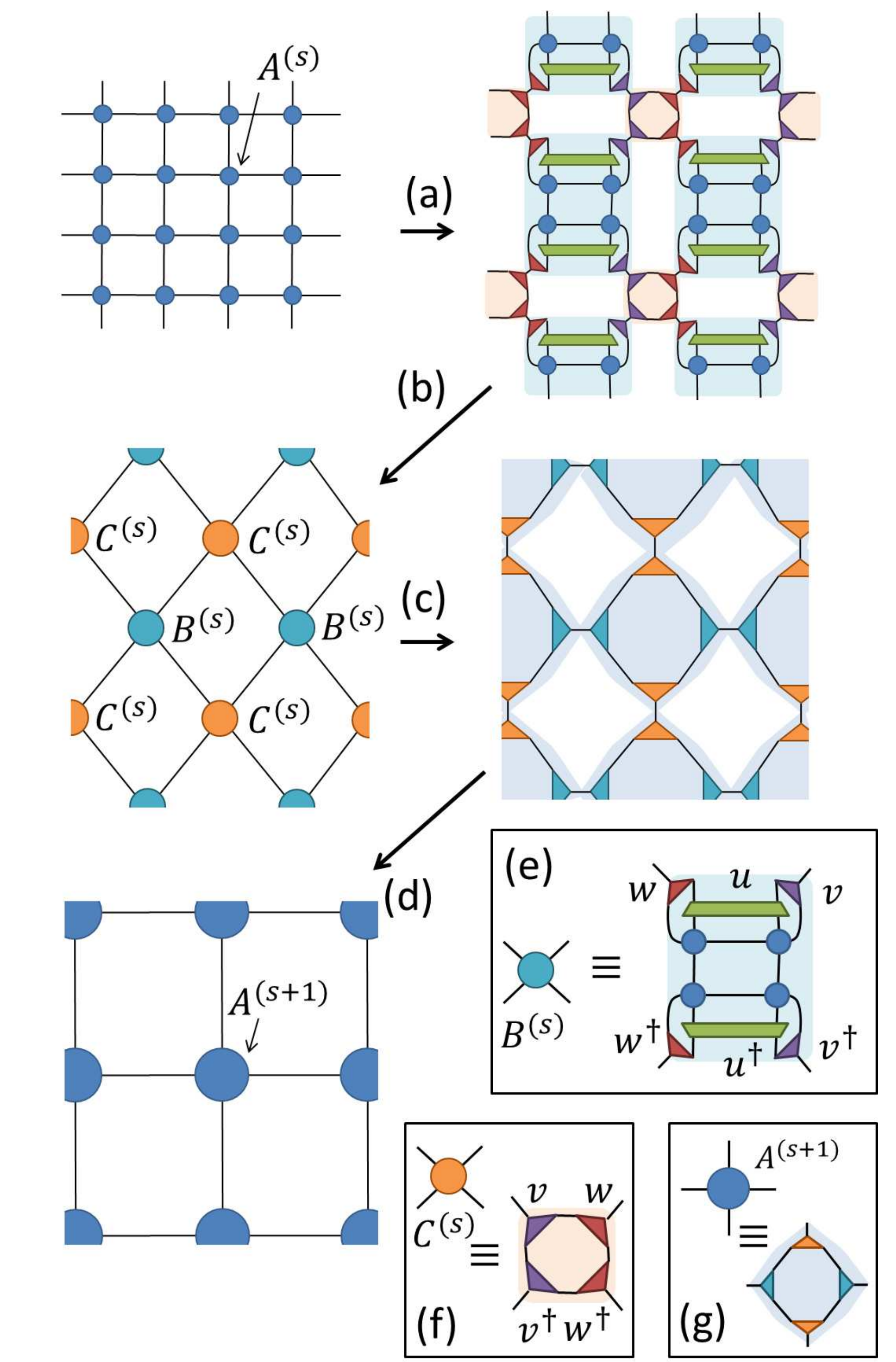
Remaining problem in TEFR

- TEFR works well at far from the critical point.
  - Because it can remove CDL structure.
- In the vicinity of the critical point, the accuracy is still poor.
  - Because the actual entanglement is not necessarily a perfect CDL structure.
- To improve further, we must consider the entanglement structure beyond the CDL tensor.

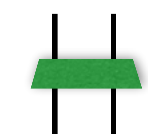
Progress: Tensor Network Renormalization

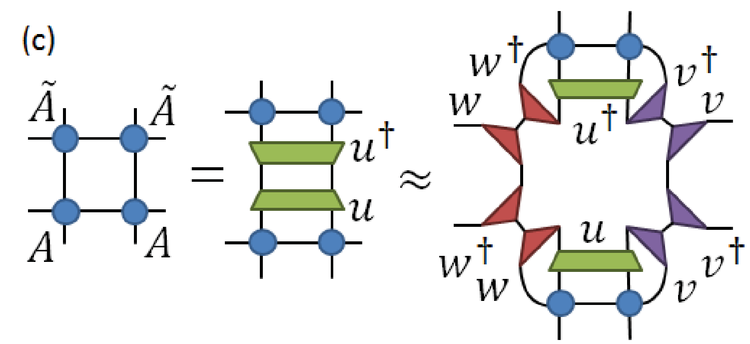
G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405(2015).

Tensor Network Renormalization



Point of TNR

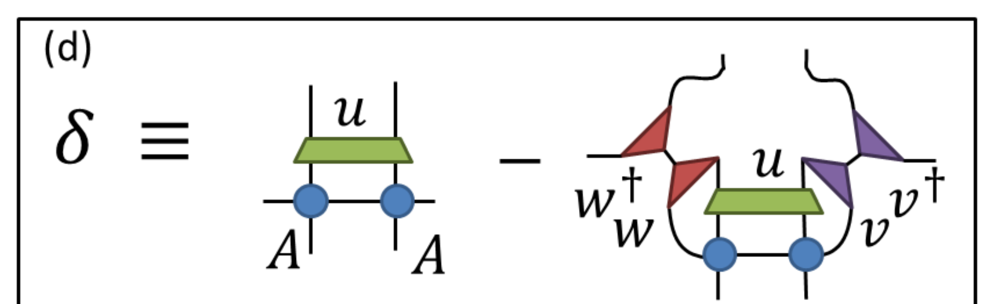
Use of a disentangler(Unitary tensor) 



It can remove short range entanglement efficiently.

(Not only CDL)

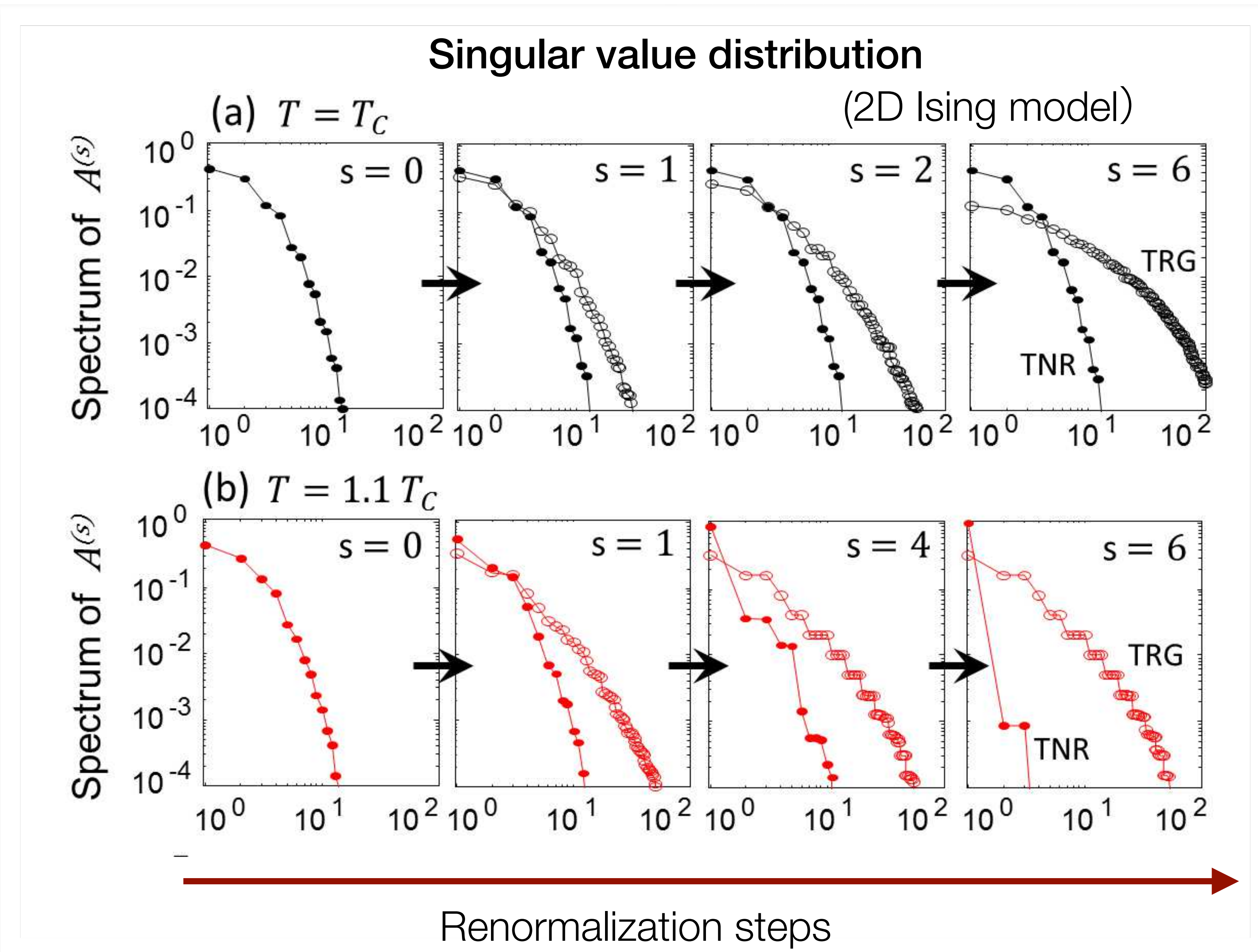
Approximation by using two-tensor cluster:

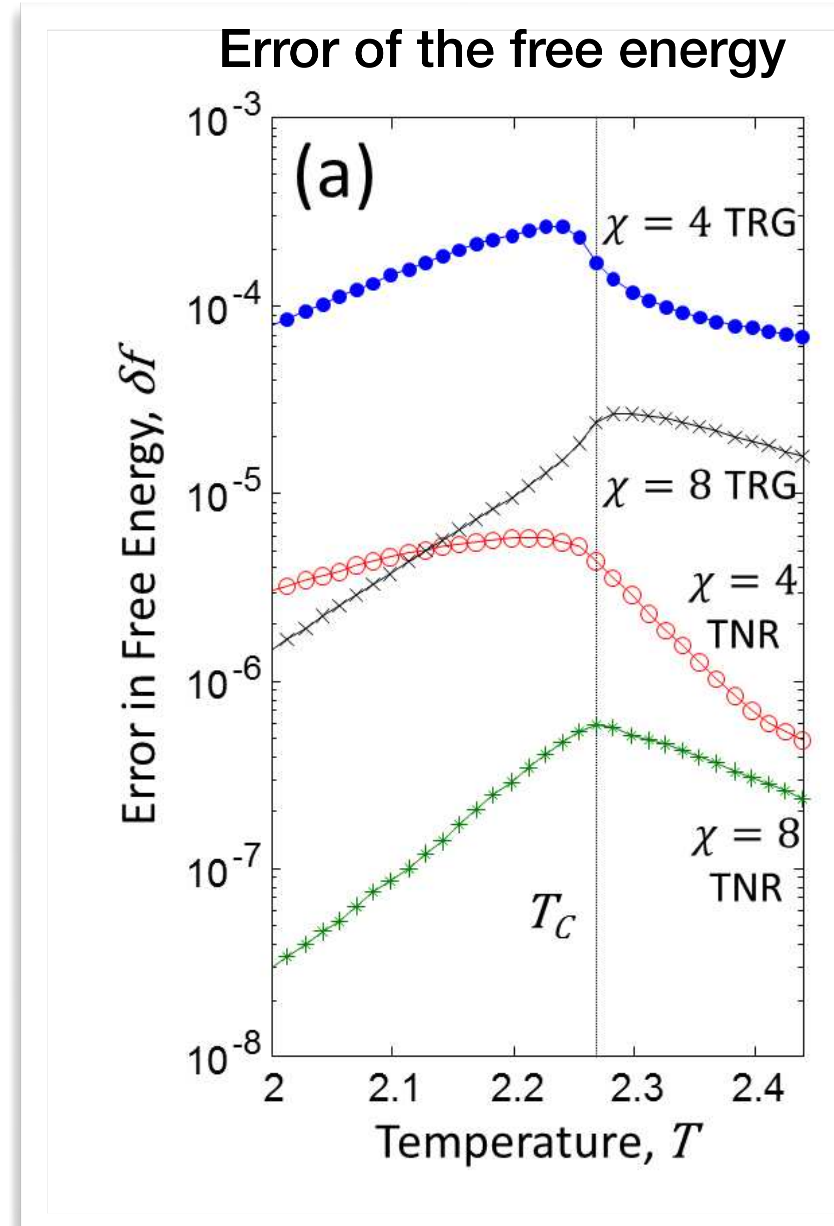


Better accuracy than the simple SVD of single tensor

Power of TNR

G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015) ,arXiv: 1412.0732v2 (free energy).





In TNR:

- The singular value distribution is narrower than that of TRG.
- It is almost unchanged at Tc.
  - Indicating scale invariance of the critical system.

Interesting topics in tensor network renormalization

- Try to find efficient algorithm to remove "short range" entanglement
    - TNR, Loop-TNR, GILT, Gauge fixing

TNR: G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015)

Loop-TNR: S. Yang, Z.-C. Gu and , X.-G. Wen, Phys. Rev. Lett. **118**, 110504 (2017)

GILT: M. Hauru, C. Delcamp. S. Mizera Phys. Rev. B **97**, 045111 (2018)

Gauge fixing: G. Evenbly, Phys. Rev. B **98**, 085155 (2018)

- Application to lattice QCD
    - TRG with Grassmann algebra

Z.-C. Gu, F. Verstraete, and X.-G. Wen, arXiv:1004.2563

S. Takeda, and Y. Yoshimura PTEP **2015**, 043B1 (2015).

- Property at the criticality
    - Relation between TNR and MERA
    - Relation to Conformal invariance

G. Evenbly and G. Vidal,Phys. Rev. Lett. **115**, 200401 (2015)

G. Evenbly, Phys. Rev. B **95**, 045117 (2017)

Next (Dec. 22)

Notice

- No classes on Nov. 3, Nov. 17, and Nov. 22
- Classes will also be held on Jan. 5 and Jan. 19

1. Computational science, quantum computing, and data compression
2. Review of linear algebra
3. Singular value decomposition
4. Application of SVD and generalization to tensors
5. Entanglement of information and matrix product states
6. Application of MPS to eigenvalue problems
7. Tensor network representation
8. Data compression in tensor network
9. Tensor network renormalization
10. Quantum mechanics and quantum computation
11. Simulation of quantum computers
12. Quantum-classical hybrid algorithms and tensor network
13. Quantum error correction and tensor network