

計算科学における情報圧縮

Information Compression in Computational Science

2018.1.11

#13: テンソルネットワーク繰り込みと低ランク近似の応用

Tensor network renormalization

理学系研究科 物理学専攻 大久保 毅

Department of Physics, **Tsuyoshi Okubo**

Outline

(Main goals)

- Tensor network representation of a scalar
 - Partition functions in statistical mechanics
 - Exponentially large computational cost
- Tensor network renormalization
 - Tensor Renormalization Group (TRG) in two dimension
 - Generalization to higher dimensions
- Tensor network renormalization around critical point
 - Fixed point of TRG: Corner double line tensors
- Report problems

(Share problems)

(Understand idea of TRG)

(Understand "entanglement structure")

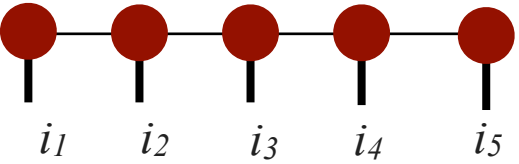
Tensor network representation of a scalar

Tensor network state: approximation for a vector

G.S. wave function: $|\Psi\rangle = \sum_{\{i_1, i_2, \dots, i_N\}} \Psi_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle$

Vector (or N-rank tensor): $\Psi_{i_1 i_2 \dots i_N} =$  # of Elements = a^N

“Tensor network”
decomposition

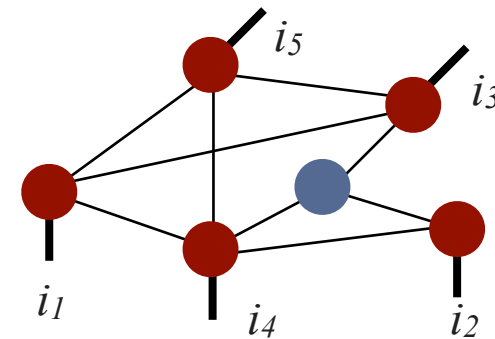
* Matrix Product State (MPS) $A_1[i_1] A_2[i_2] \dots A_N[i_N] =$ 

$A[m]$: Matrix for state m

* General network $\text{Tr} X_1[i_1] X_2[i_2] X_3[i_3] X_4[i_4] X_5[i_5] Y$

X, Y : Tensors

Tr : Tensor network contraction



By choosing a “good” network, we can express G.S. wave function efficiently.

ex. MPS: # of elements = $2ND^2$

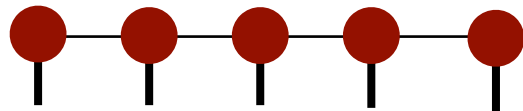
D : dimension of the matrix A

Exponential \rightarrow Linear

*If D does not depend on N ...

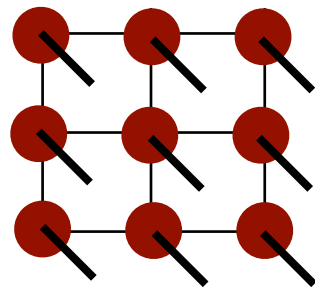
Examples of TNS

MPS:



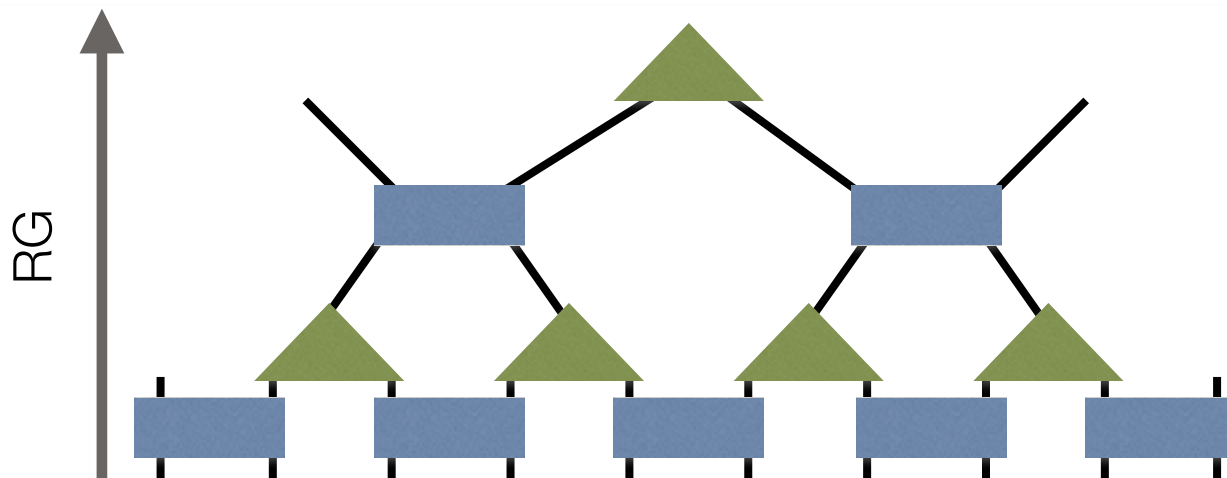
Good for 1-d gapped systems

PEPS, TPS:



For higher dimensional systems
Extension of MPS

MERA:

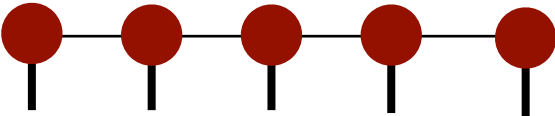



Scale invariant systems

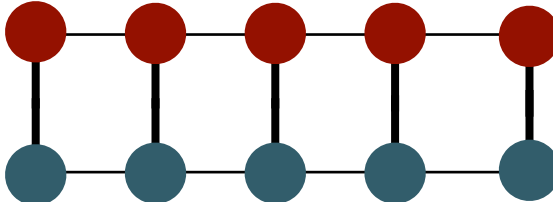
Tensor network representation of a scalar


Example: inner product of two TNSs

MPS

$$\vec{v} =$$


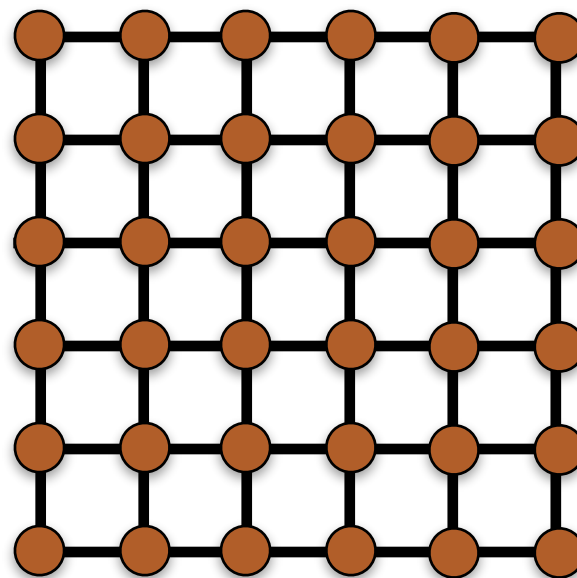
$$\vec{u} =$$


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


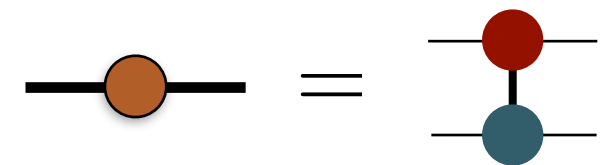
$$=$$


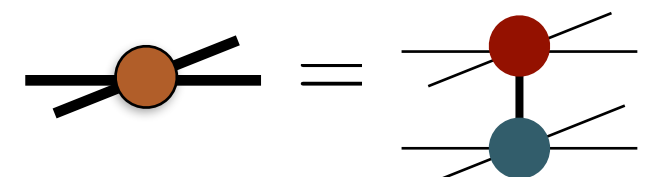
TPS (in two dimension)

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



Double layer tensor





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. $\{S_1, S_2, \dots, S_L\}$)

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

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

Partition function (分配関数) \mathcal{H} : Hamiltonian

= Normalization factor of the canonical ensemble

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

Relation to the free energy in thermodynamics

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

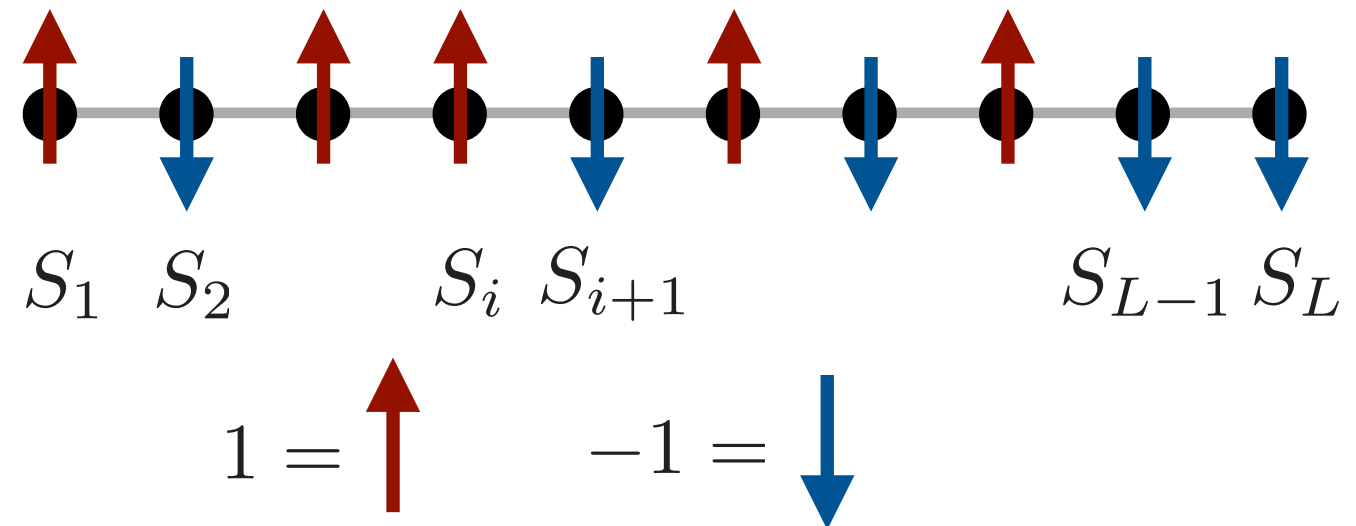
log of the partition function = Free energy

Tensor network representation of partition function

Classical Ising model on a chain

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

$S_i = 1, -1$



Partition function:

$$\begin{aligned} Z &= \sum_{\{S_i = \pm 1\}} e^{\beta J \sum_i S_i S_{i+1}} \\ &= \sum_{\{S_i = \pm 1\}} \prod_{i=1}^{L-1} e^{\beta J S_i S_{i+1}} \\ &= \sum_{S_1 = \pm 1, S_L = \pm 1} (T^{L-1})_{S_1, S_L} \end{aligned}$$

Transfer matrix
(転送行列)

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

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

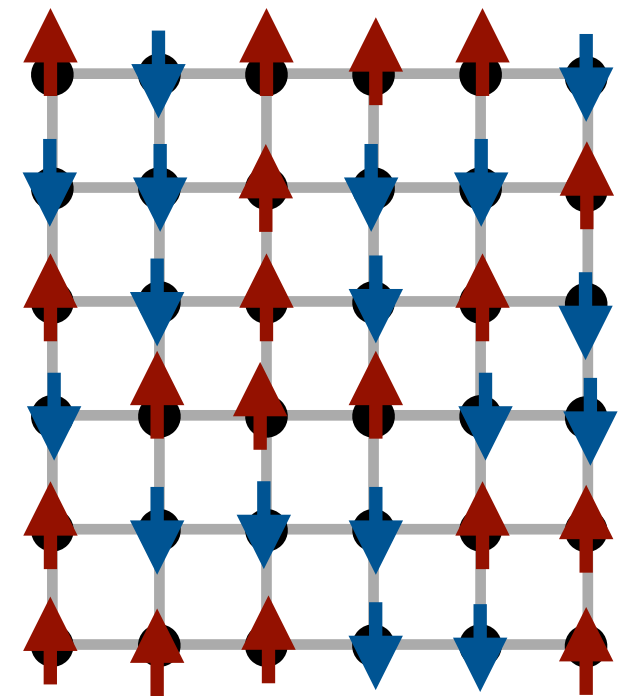
$$\sum_{S_1 = \pm 1, S_L = \pm 1} \text{---} S_1 \text{---} \text{---} \text{---} \text{---} S_L$$

Tensor network representation in two dimension

Classical Ising model on the square lattice

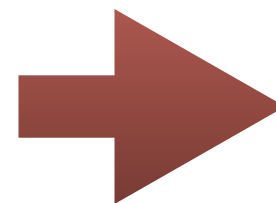
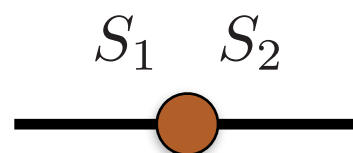
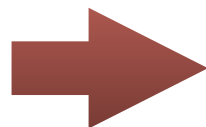
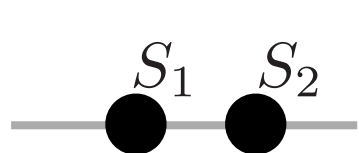
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j \quad (S_i = \pm 1 = \uparrow, \downarrow)$$

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



We can use a tensor instead of the transfer matrix.

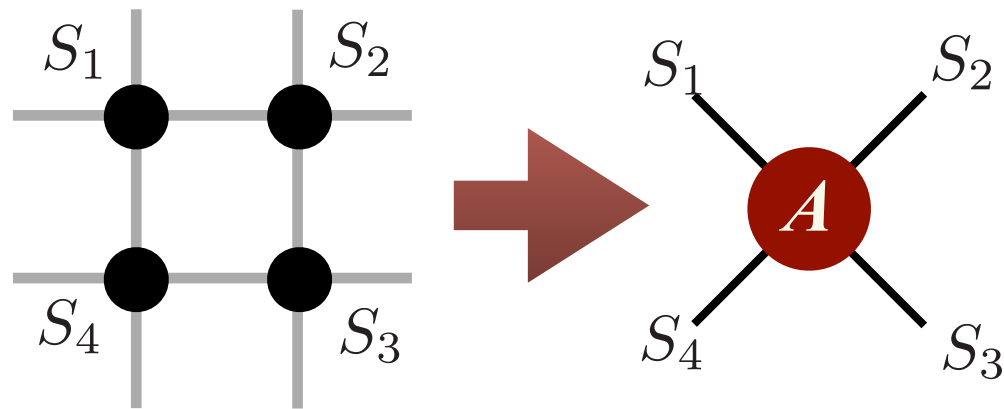
$$e^{\beta J S_1 S_2} = T_{S_1 S_2}$$



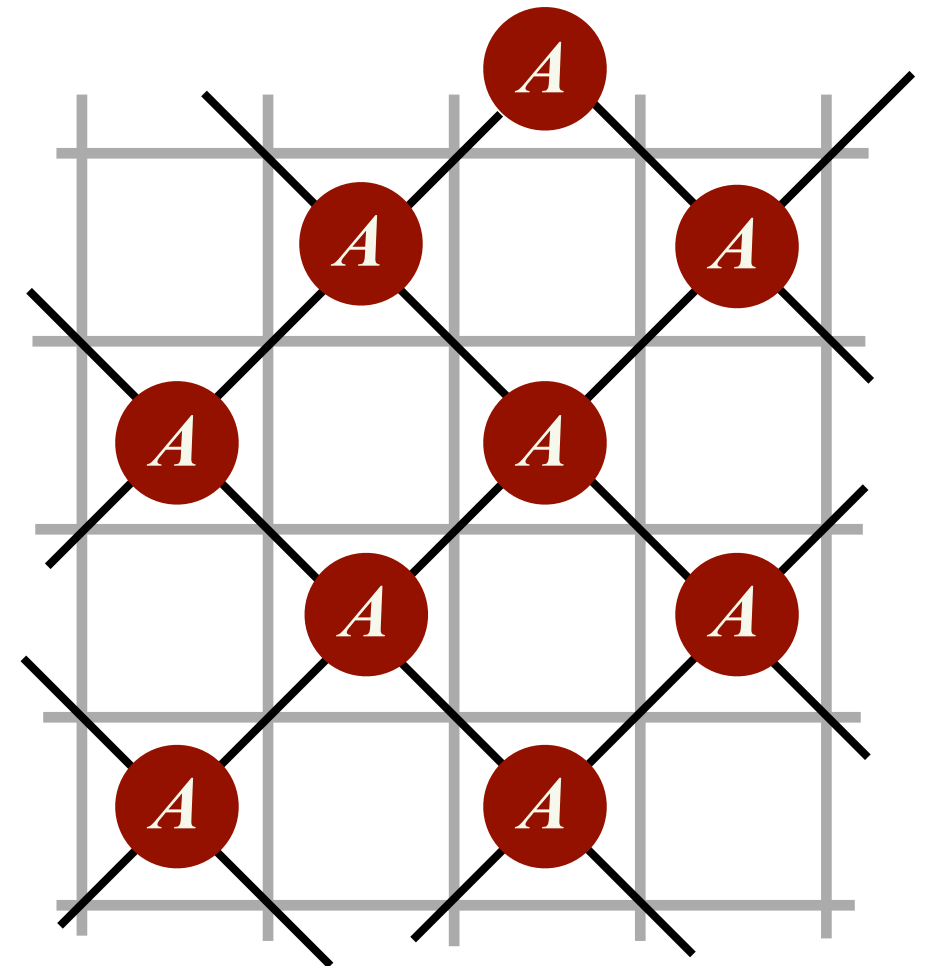
Tensor?

Tensor network representation in two dimension

$$e^{\beta J(S_1 S_2 + S_2 S_3 + S_3 S_4 + S_4 S_1)} = A_{S_1 S_2 S_3 S_4}$$



$Z =$



Partition function = Tensor network of tensor A

Square lattice Ising model \rightarrow 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:

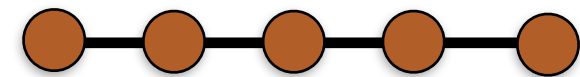
d-dimensional cubic lattice $N = L^d$

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

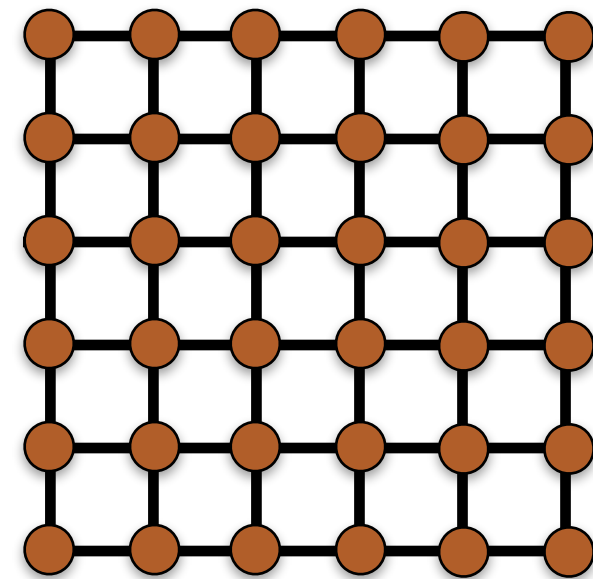
Square: $O(D^L)$ (Open)
 $O(D^{2L})$ (Periodic)

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

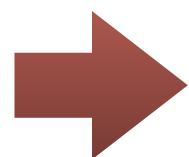
Chain



Square lattice



It is **impossible** to perform exact contraction.



We need **efficient approximations** for the contraction.

Tensor network renormalization

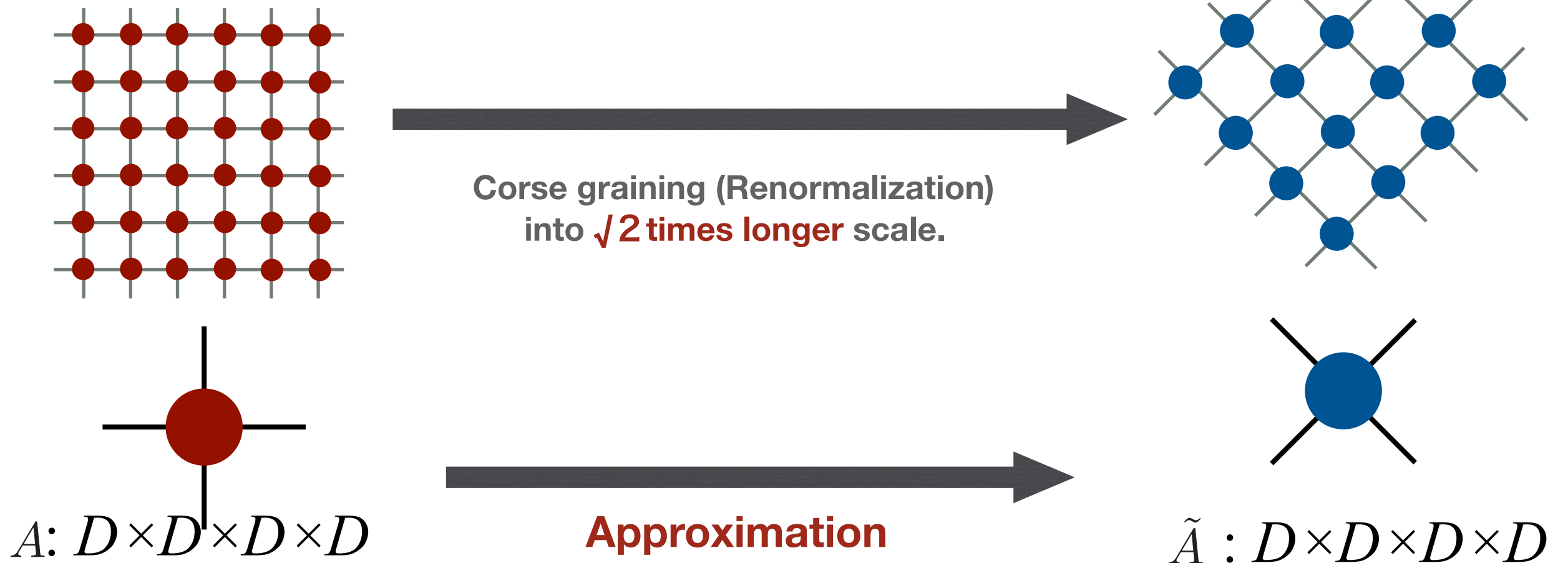
Tensor network renormalization (テンソルネットワーク繰り込み)

- **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 network renormalization

Scalar represented
by $L \times L$ tensors

$(L \times L)/2$ tensors



Reduce the number of tensors
keeping their size constant

Key technique: low rank approximation by SVD

Best low-rank approximation of **a matrix** = SVD

$$\begin{array}{ccccc}
 \text{---} \textcircled{A} \text{---} & = & \text{---} \textcircled{U} \text{---} \boxed{\Lambda} \text{---} \textcircled{V^\dagger} \text{---} & \simeq & \text{---} \textcircled{\tilde{U}} \text{---} \boxed{\tilde{\Lambda}} \text{---} \textcircled{\tilde{V}^\dagger} \text{---} \\
 A : M \times N & & \Lambda : M \times M & & \tilde{\Lambda} : R \times R \\
 (M \leq N) & & (\text{Diagonal matrix}) & & (\text{Keeping the } R \text{ largest} \\
 & & U, V : (M, N) \times M & & \text{singular values}) \\
 & & & & \tilde{U}, \tilde{V} : (M, N) \times R
 \end{array}$$

In addition,

$$\begin{array}{ccccc}
 = & \text{---} \textcircled{\tilde{U}} \text{---} \boxed{\sqrt{\tilde{\Lambda}}} \text{---} \boxed{\sqrt{\tilde{\Lambda}}} \text{---} \textcircled{\tilde{V}^\dagger} \text{---} & = & \text{---} \textcircled{X} \text{---} \textcircled{Y} \text{---} \\
 \sqrt{\tilde{\Lambda}} : \text{Diagonal matrix} & & X = \tilde{U} \sqrt{\tilde{\Lambda}} : M \times R & & \\
 \text{those elements are } \sqrt{\lambda} & & Y = \sqrt{\tilde{\Lambda}} \tilde{V}^\dagger : R \times M & &
 \end{array}$$

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

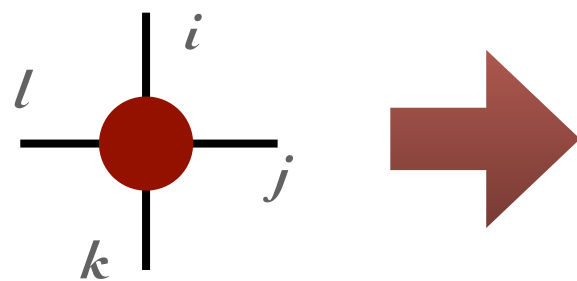
Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)

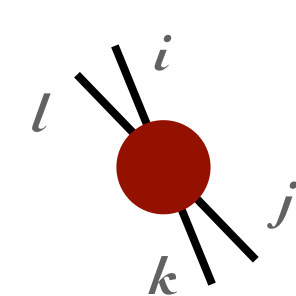
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

1. Decomposition

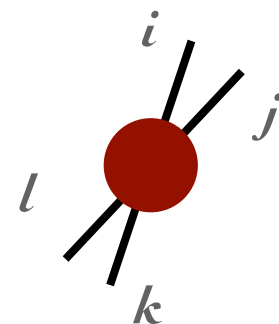
Regard a tensor as a **matrix**



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

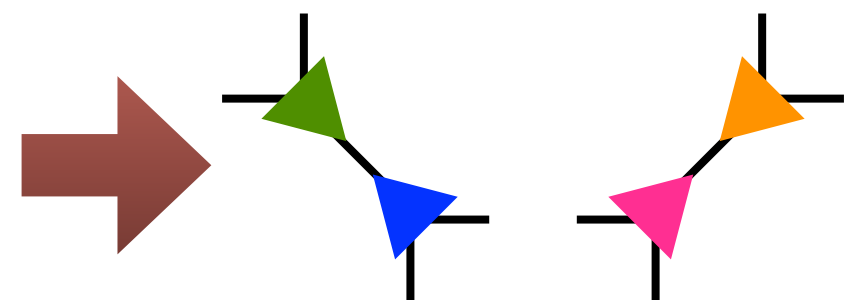


$$A_{(i,l),(j,k)}$$

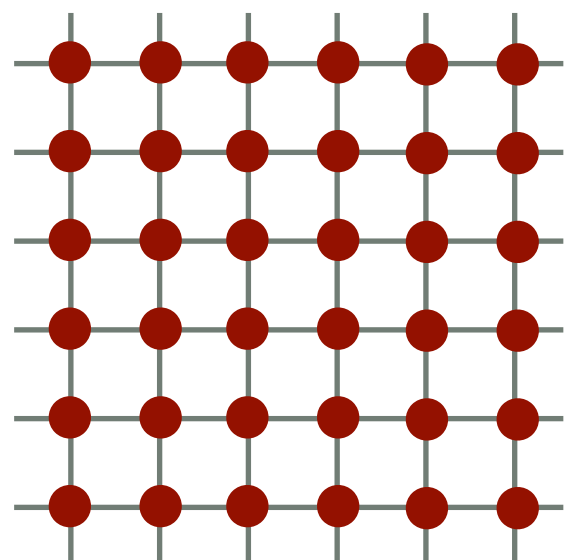


$$A_{(i,j),(k,l)}$$

D-rank approximation
by **SVD**



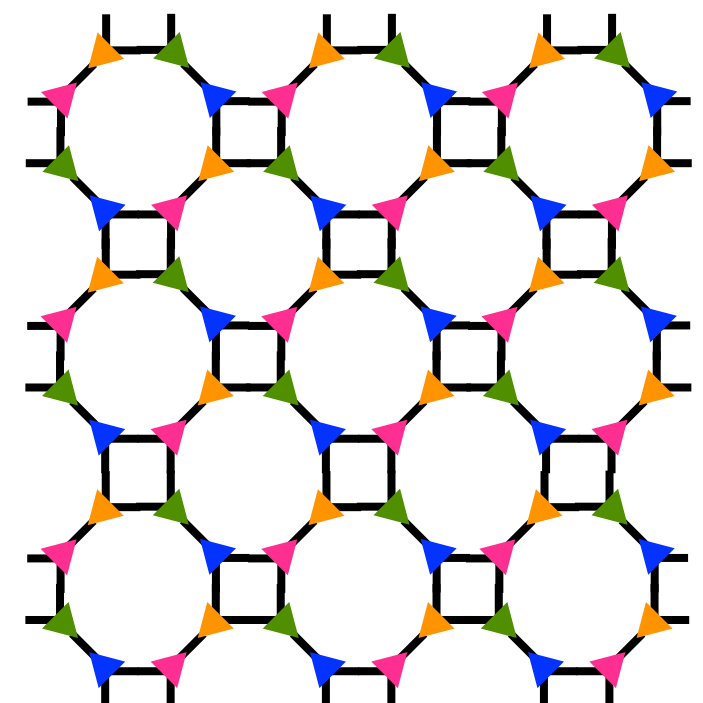
$$A: D \times D \times D \times D$$



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



Approximation

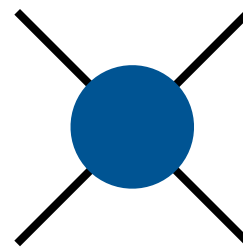
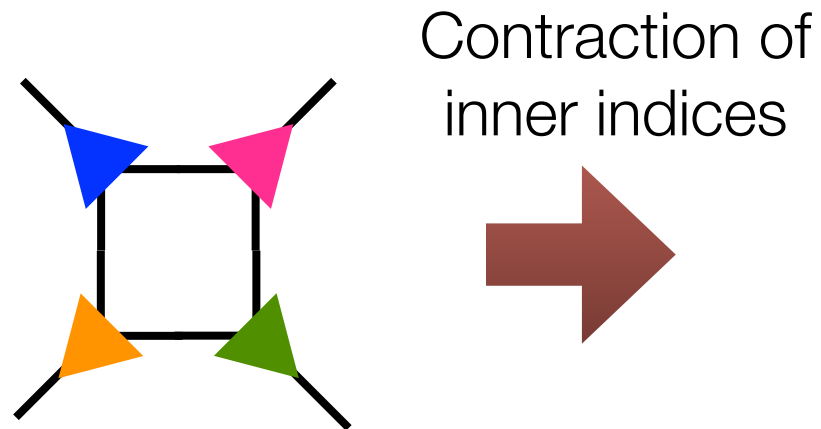


Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)

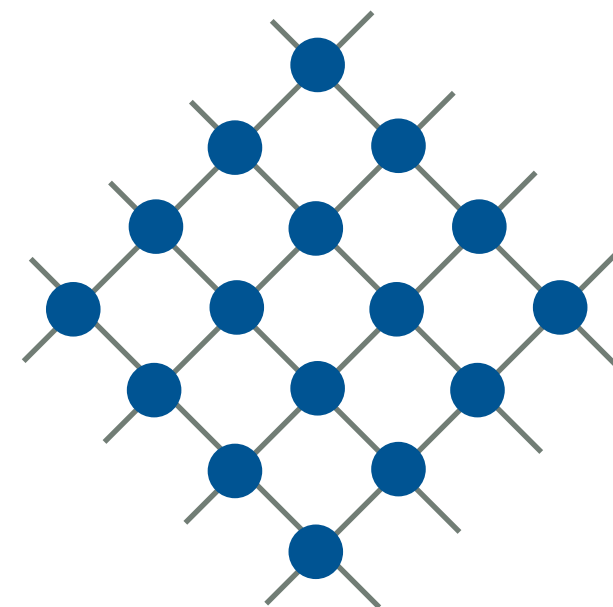
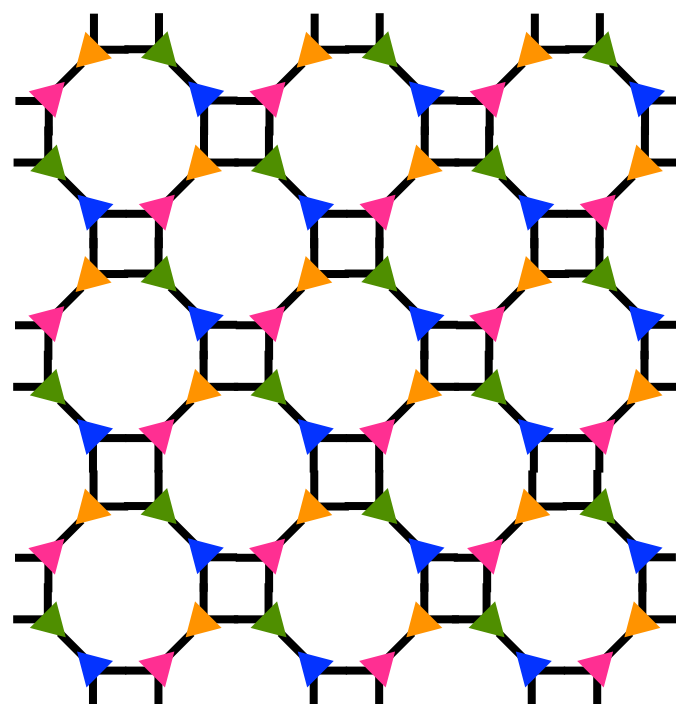
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

2. Coarse graining



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

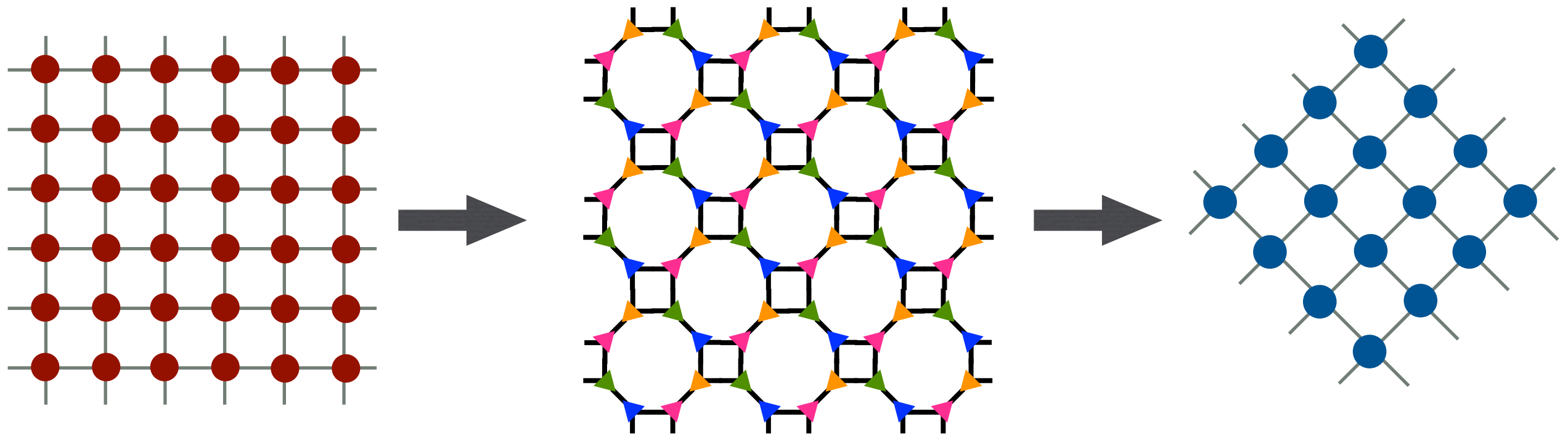
In total, **two original tensors** are coarse grained into **a new tensor**.



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)



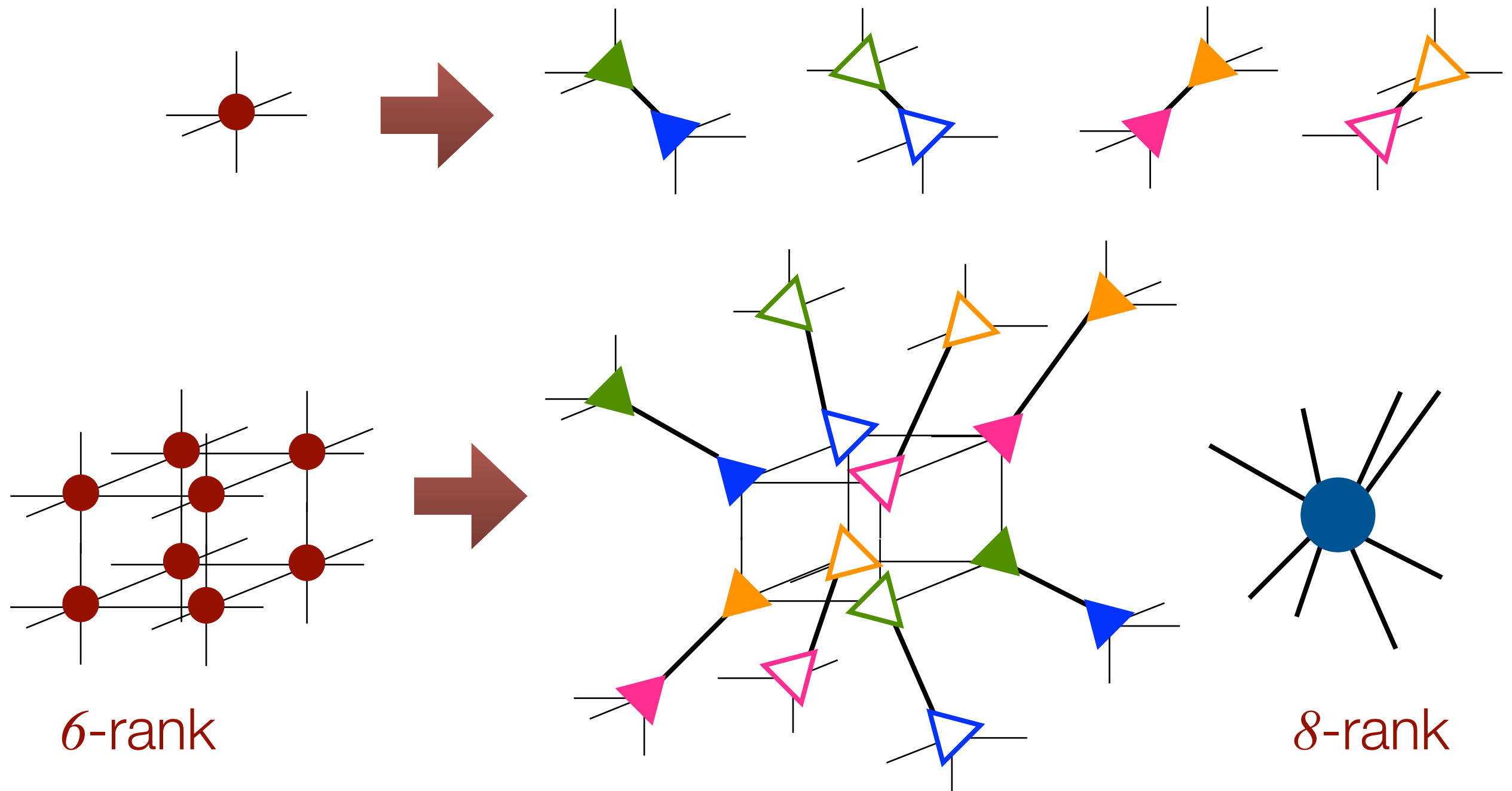
Calculation cost: $\text{SVD} = O(D^6)$ (per tensor)
Contraction = $O(D^6)$

*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

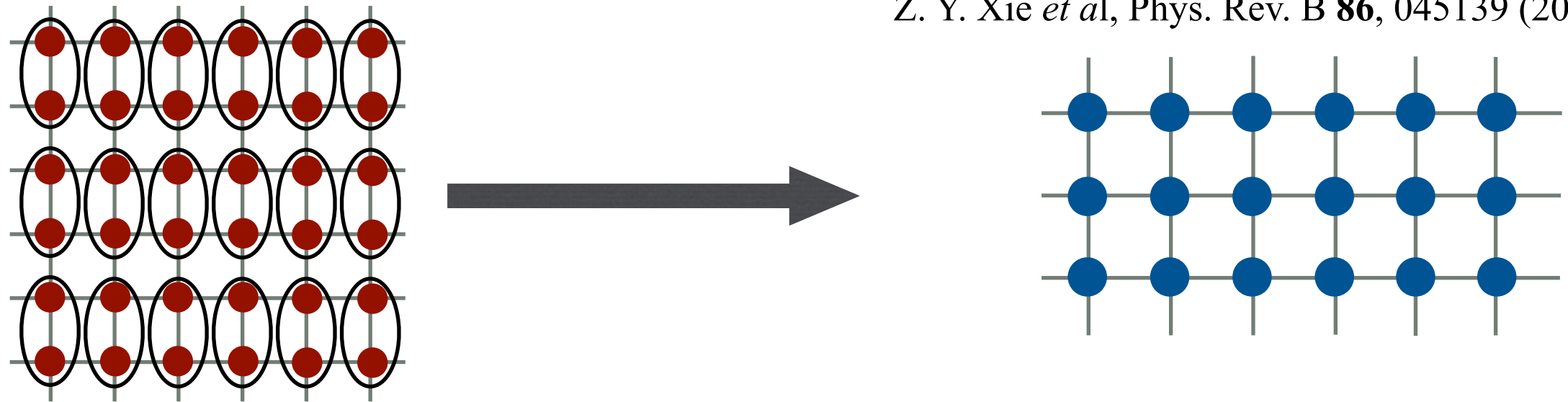
Simple generalization of TRG to cubic lattice (three dimension)



Tensor renormalization group by using HOSVD

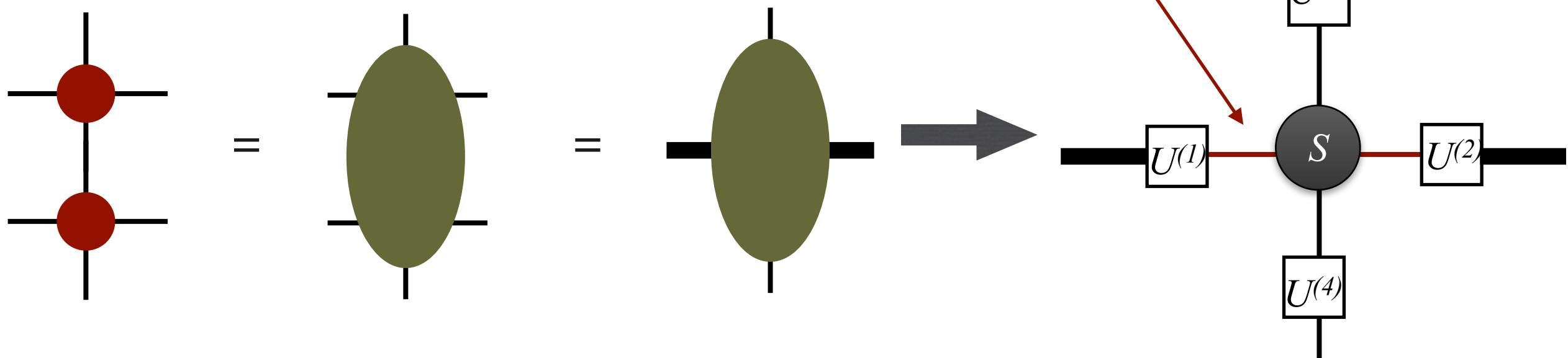
Anisotropic coarse graining by using **HOSVD** instead of SVD

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



Basic idea of **HOTRG** algorithm:

(For details, see the original paper.)

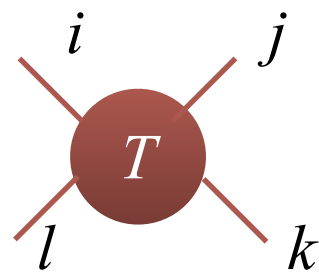


(Reference of HOSVD)

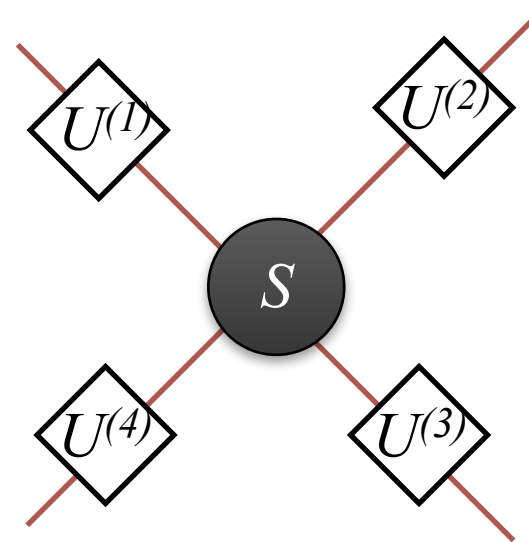
Tucker decomposition: generalization of SVD

Review: T. G. Kolda et al, SIAM Review **51**, 455 (2006)

Tucker decomposition:



=



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

S : Core tensor

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

Low "rank" approximation

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

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

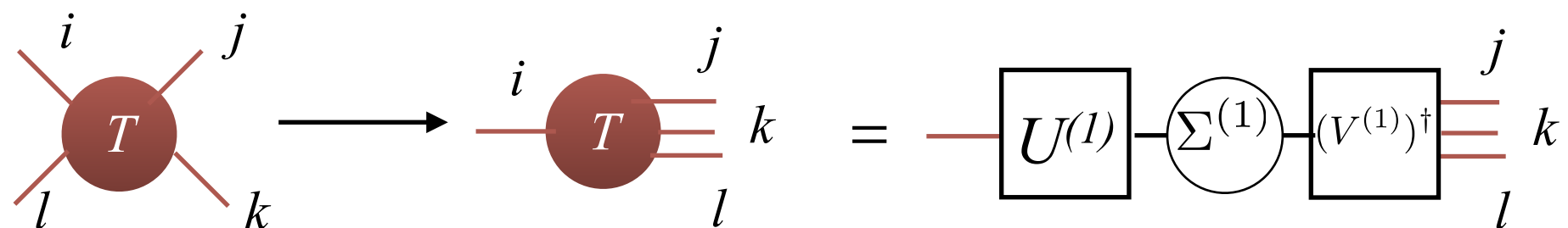
rank- (I', J', K', L') approximation

(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

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

Properties of the core tensor

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

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**.

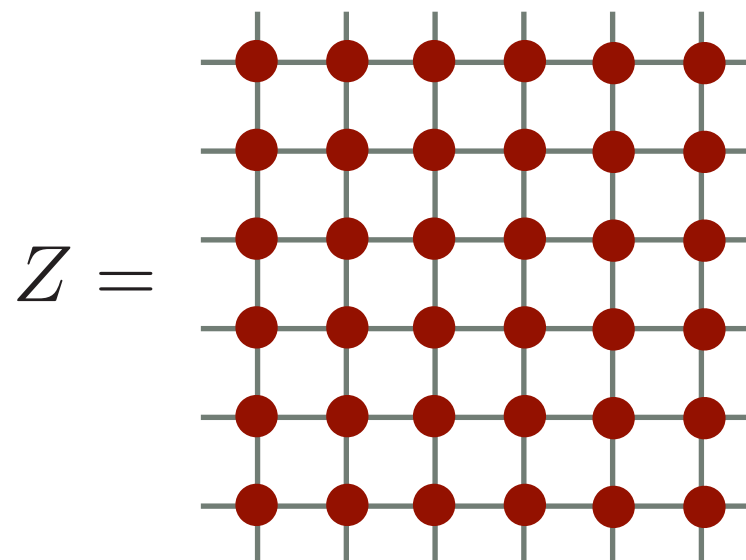
Disadvantage:

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

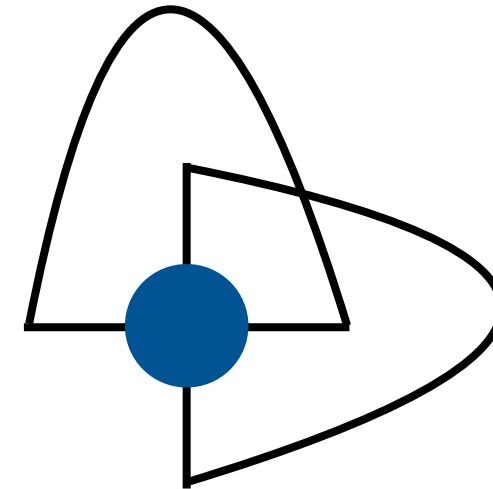
Application to a classical partition function

Partition function

(Periodic boundary condition)



Repeat TRG step
until **only a few**
tensors remain.



We can easily calculate physical quantities from Z .

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

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

(Use difference approximation)

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

(Use difference approximation)

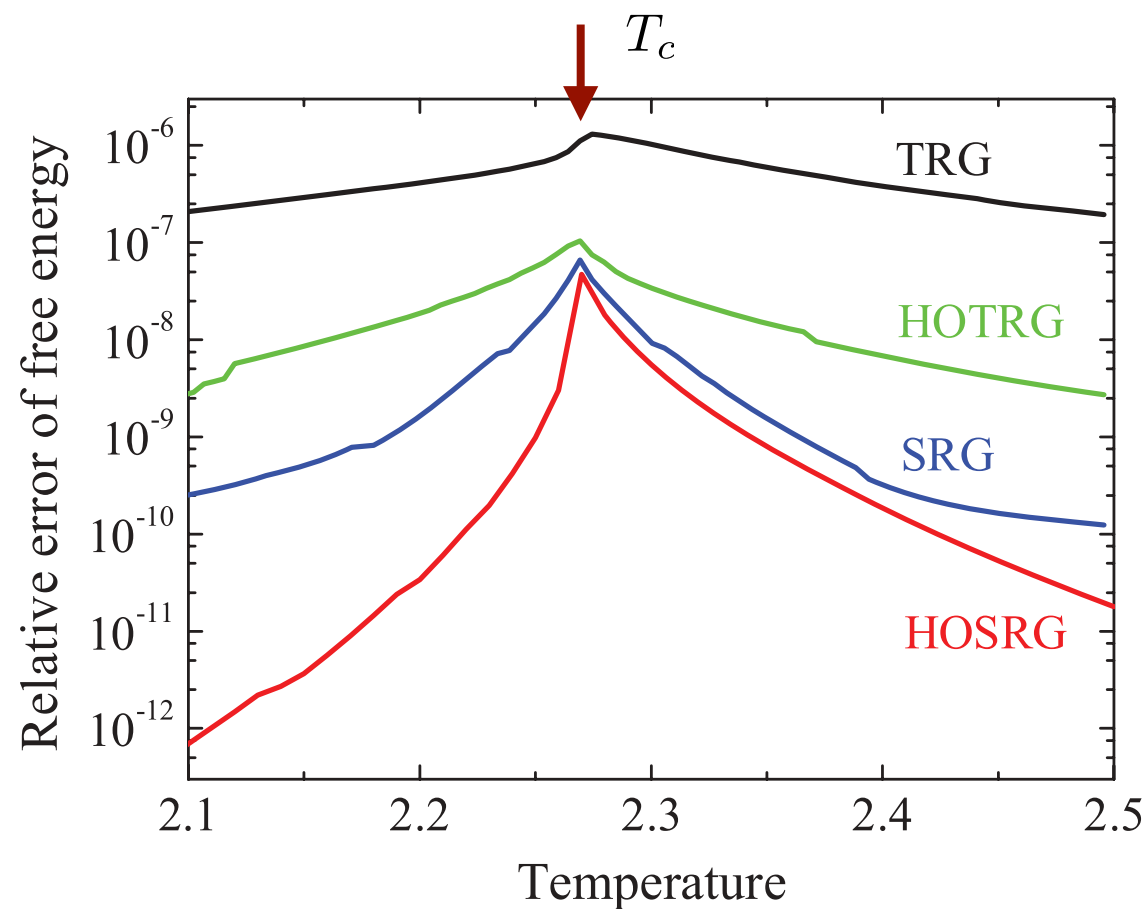
Example of calculation

Ising model in **infinite size**

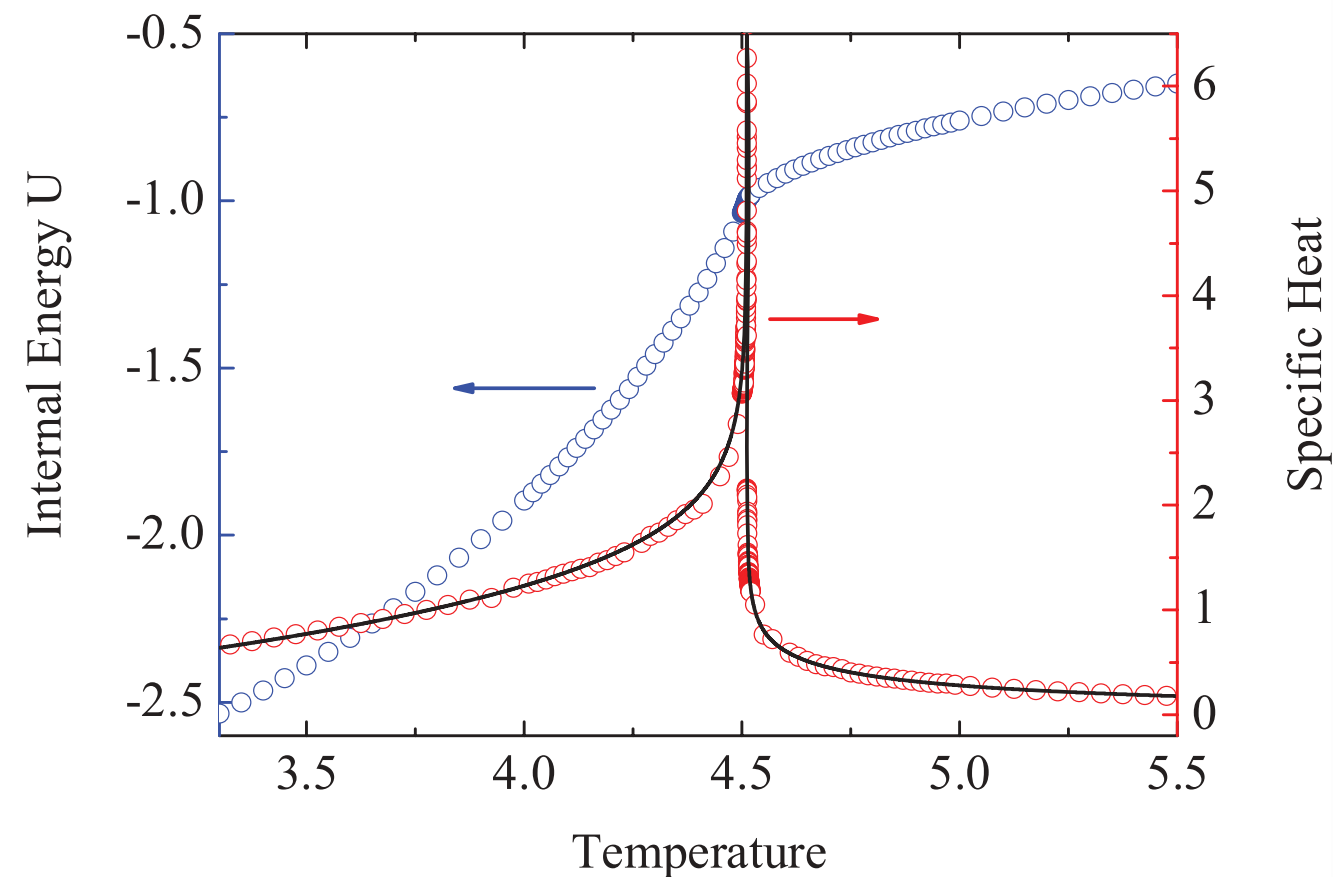
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

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

Error of free energy for 2D Ising model



Energy and specific heat of **3D** Ising model



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

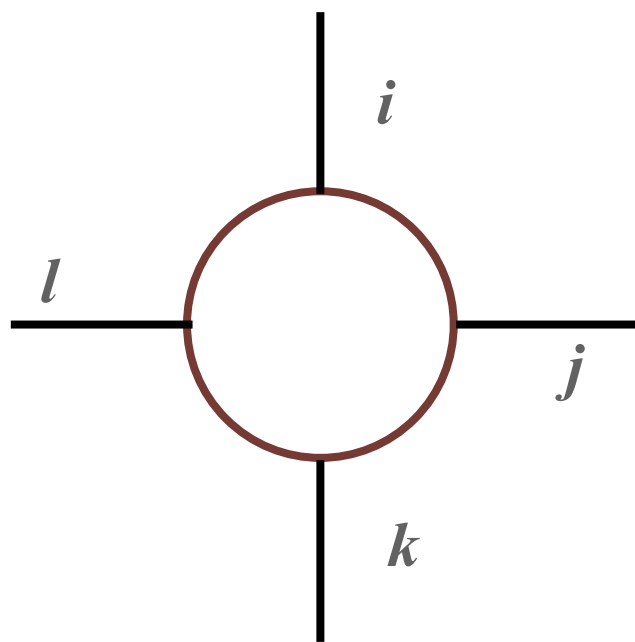
Tensor network renormalization at the critical point

- When the accuracy of TRG becomes worse?

Correlation (entanglement) within a tensor

General tensor

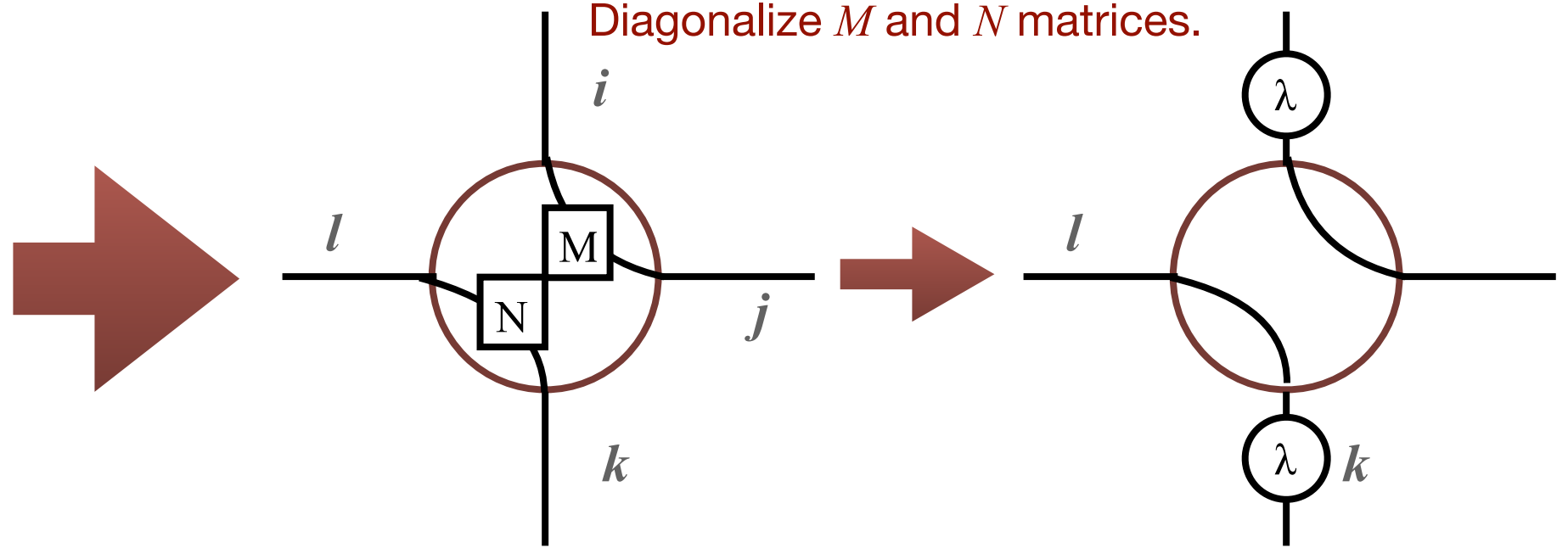
A_{ijkl}



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

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

Diagonalize M and N matrices.



New rule for representation of the correlation:

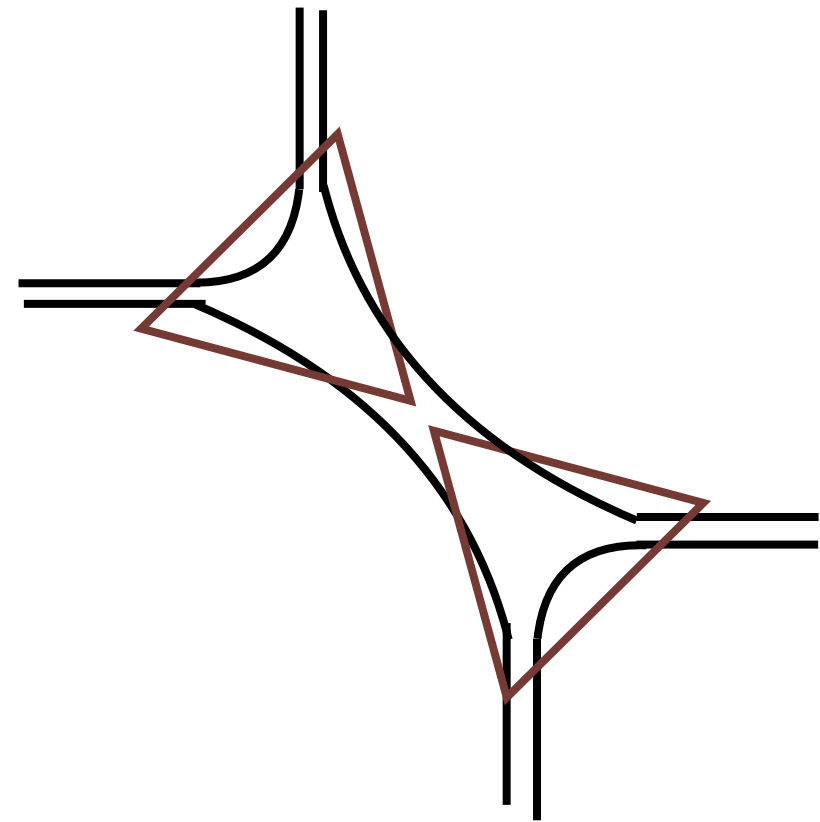
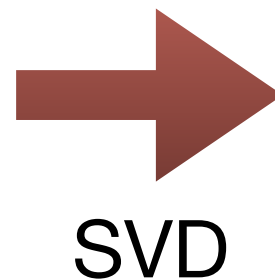
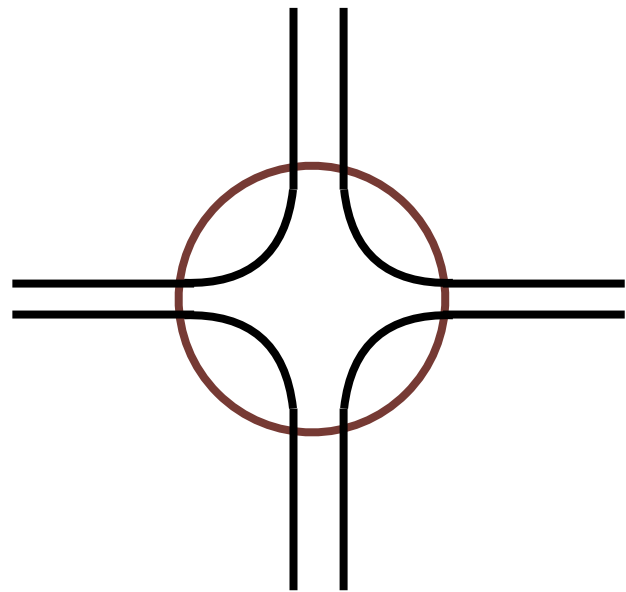
$$i \text{ — } j = \delta_{ij}$$

(+ we neglect eigenvalues in the graph.)

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}

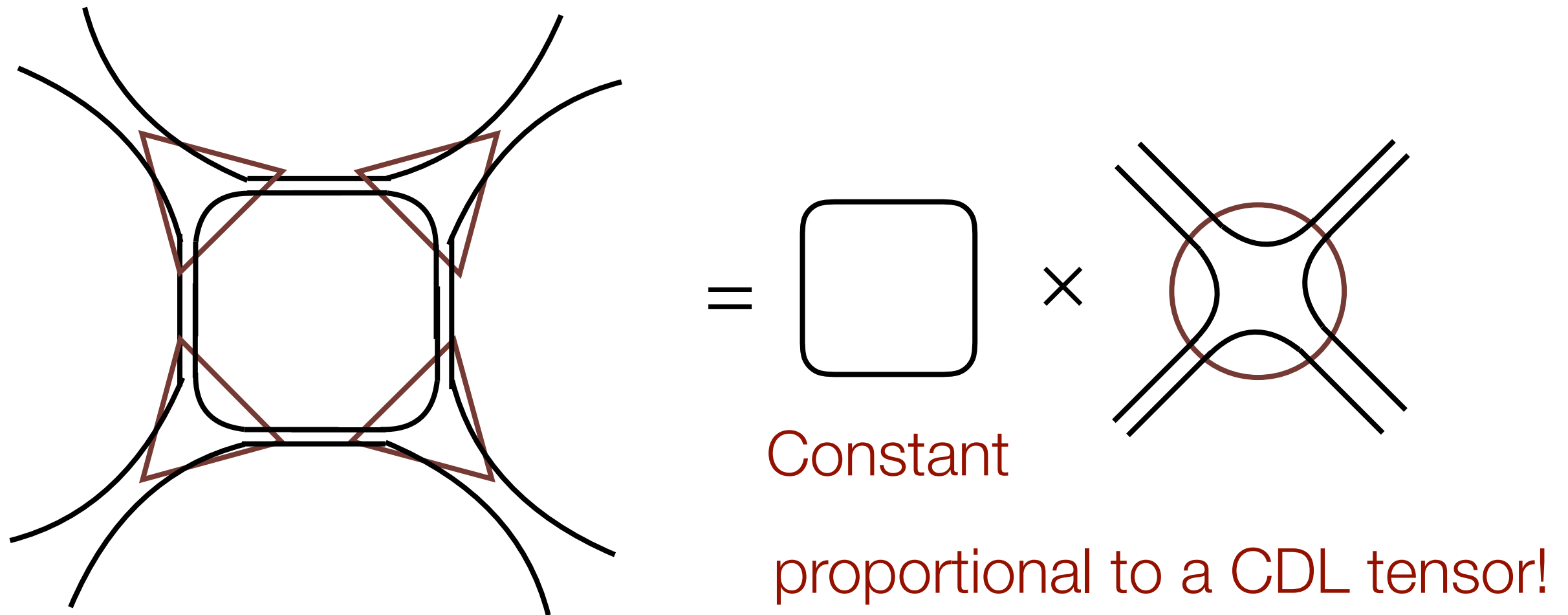
Degree of freedoms
connecting two tensors.

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:

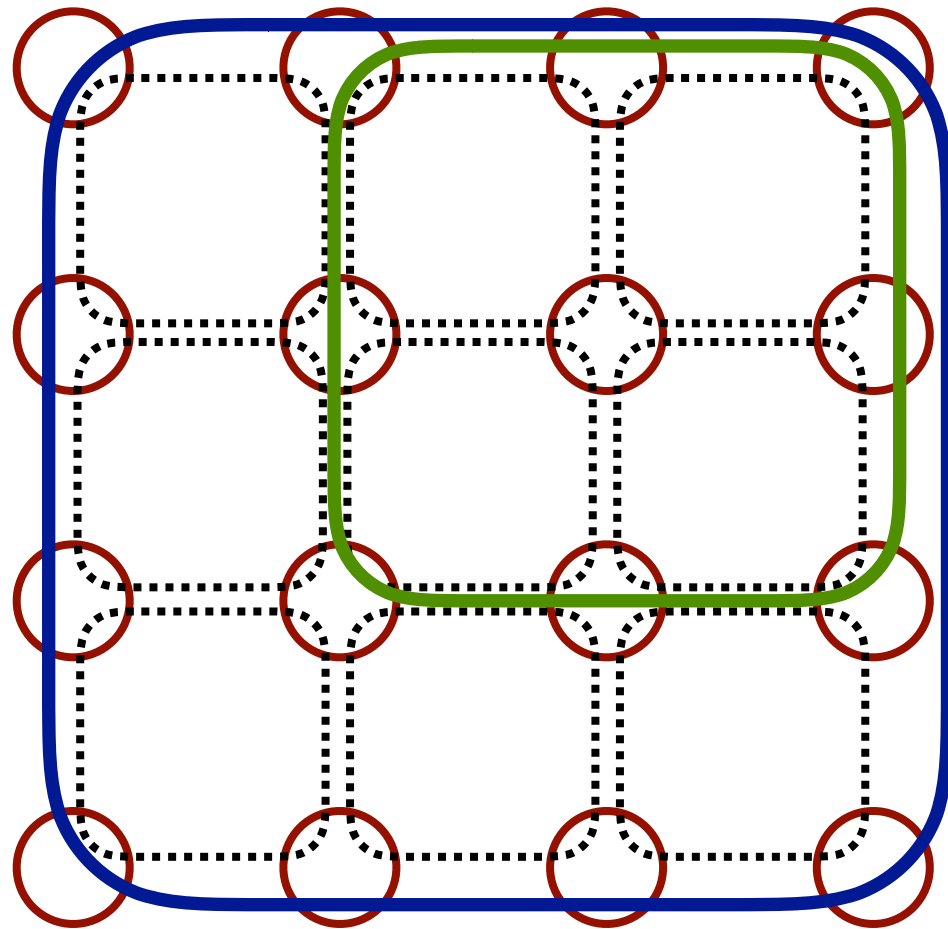


CDL tensor is **a fixed point** of TRG (and also HOTRG).

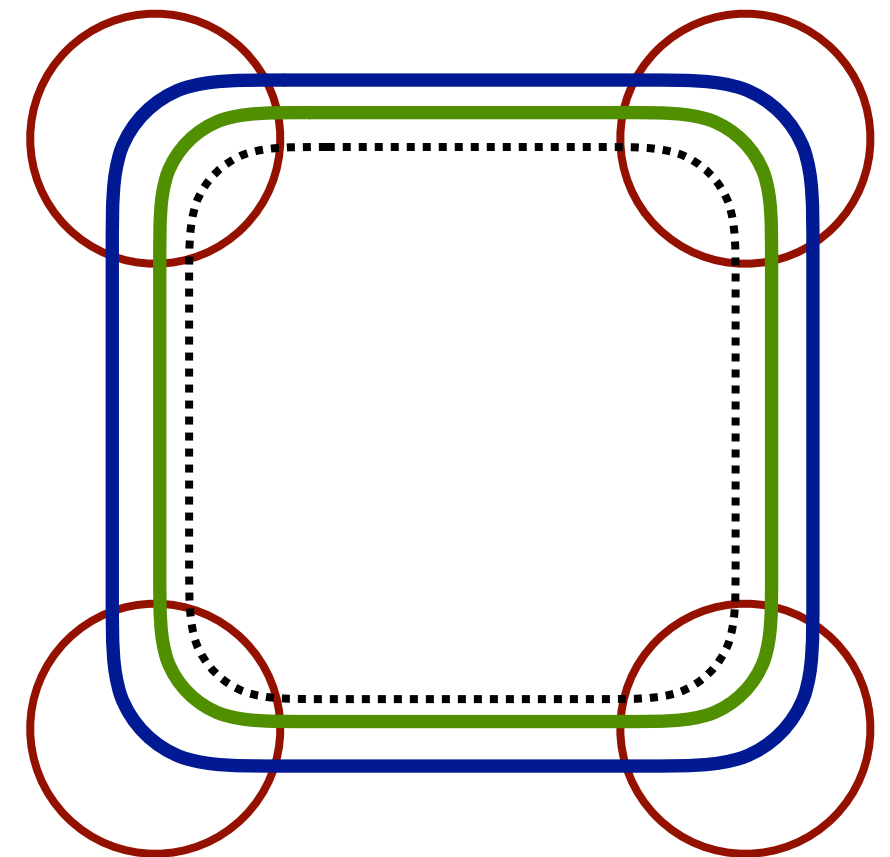
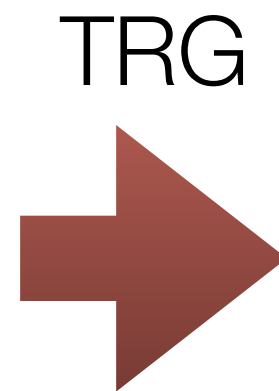
CDL tensor remains as CDL tensor along TRG.

Problems in TRG: accumulation of correlations

Correlation in several scales



Correlations **remains** after TRG.



Ideal renormalization:

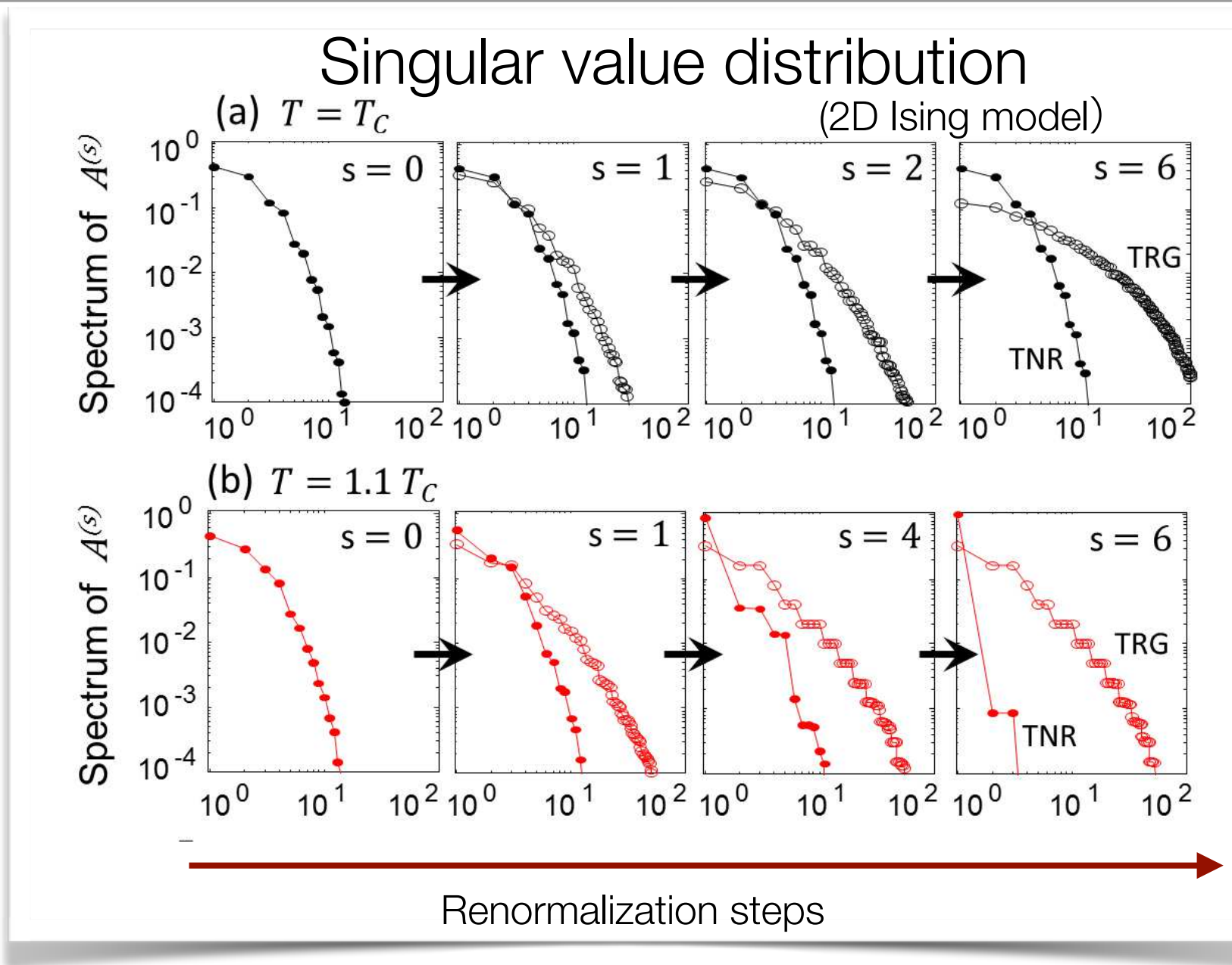
Correlation in shorter scales **should be removed**.
Only the correlation in the present scale exists.

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.

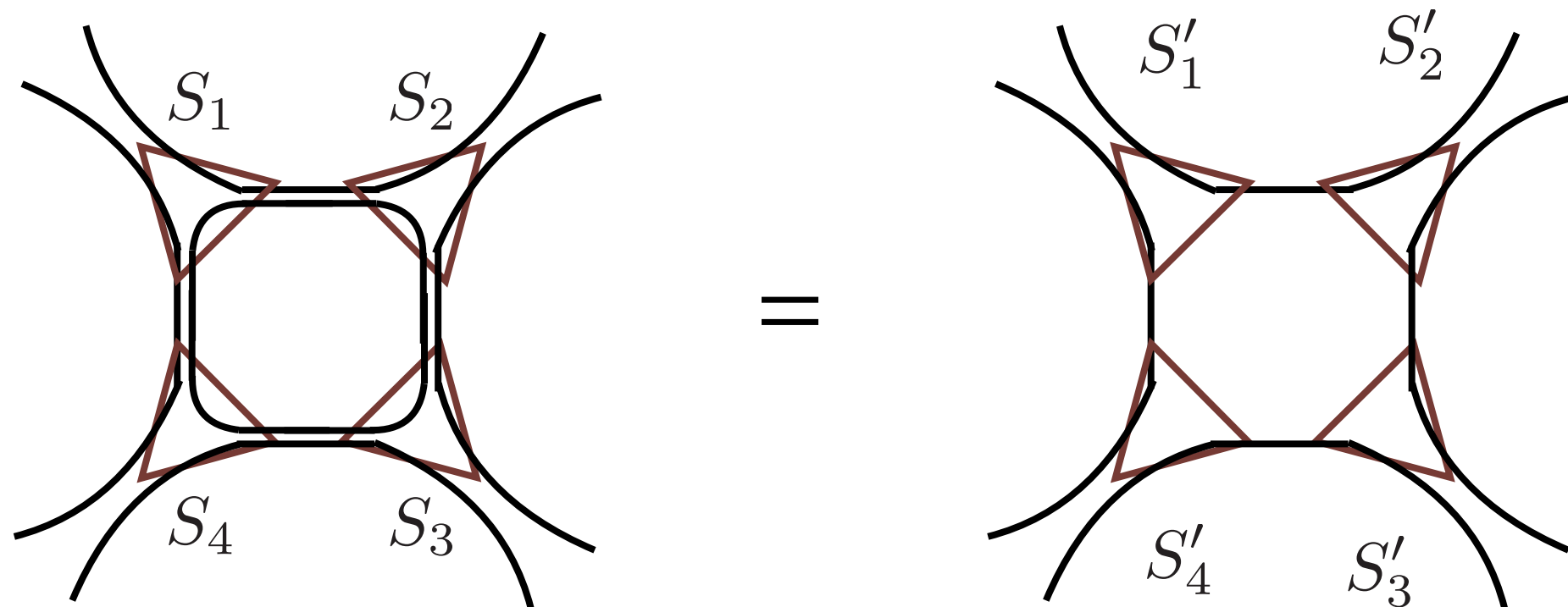
➡ Increase of truncation error (decrease of accuracy)

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:



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

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

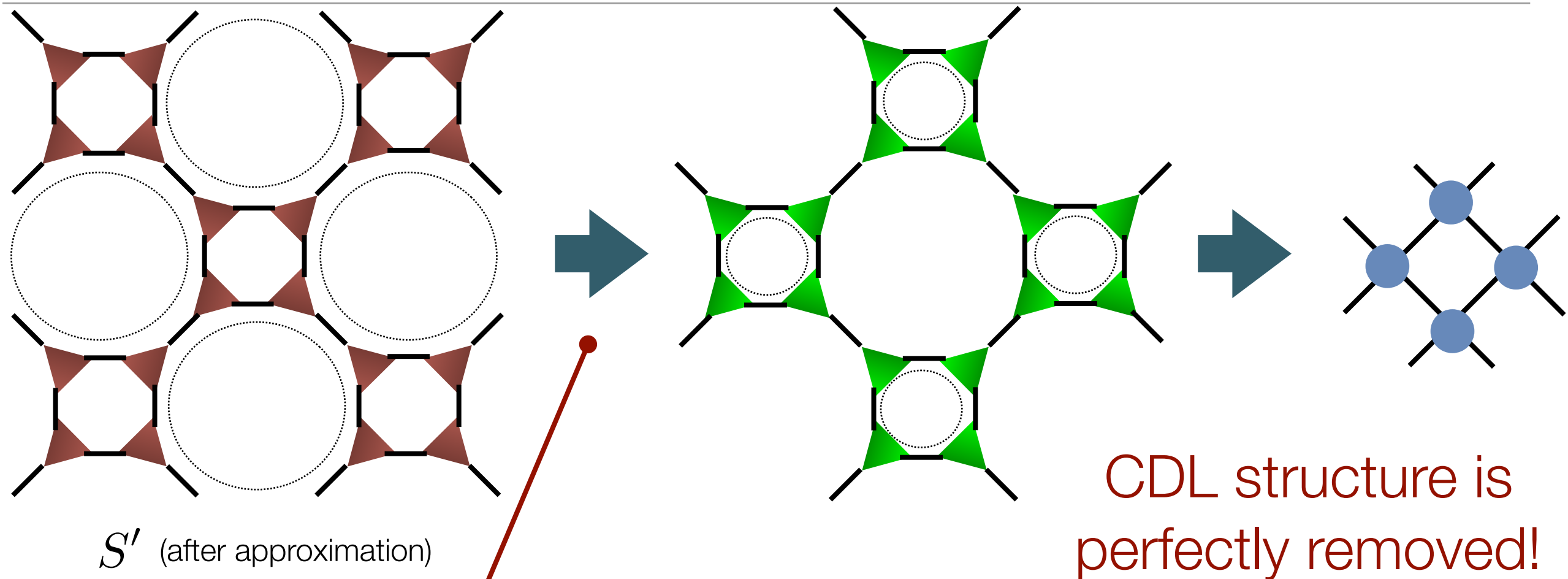
$$S' = \square^{1/4} S$$

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

Insert this "approximation" into the TRG algorithm.

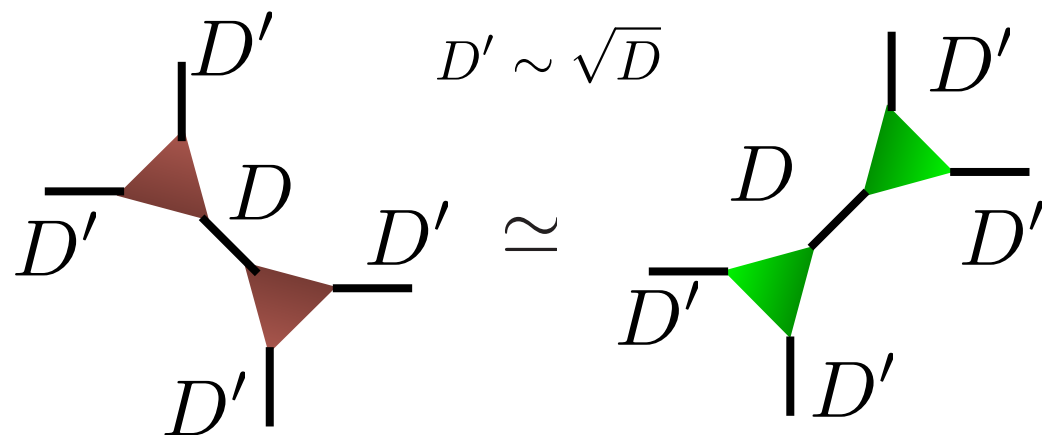
Tensor Entanglement Filtering Renormalization

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

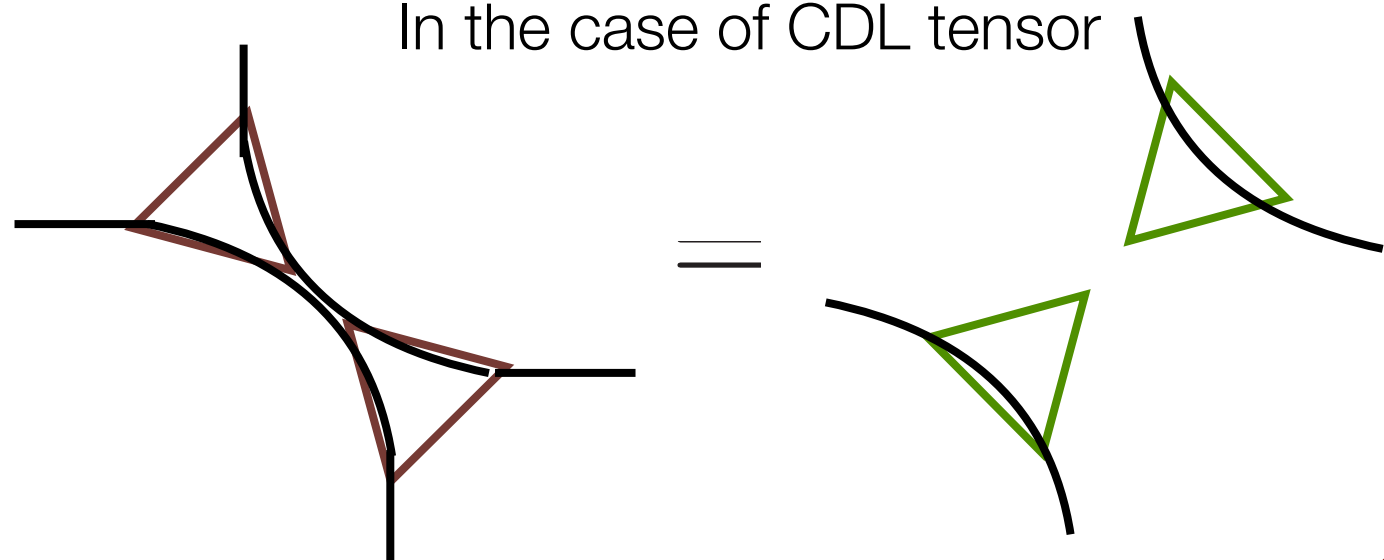


Change of SVD: $D' < D$

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



In the case of CDL tensor



Remaining problem in TEFR

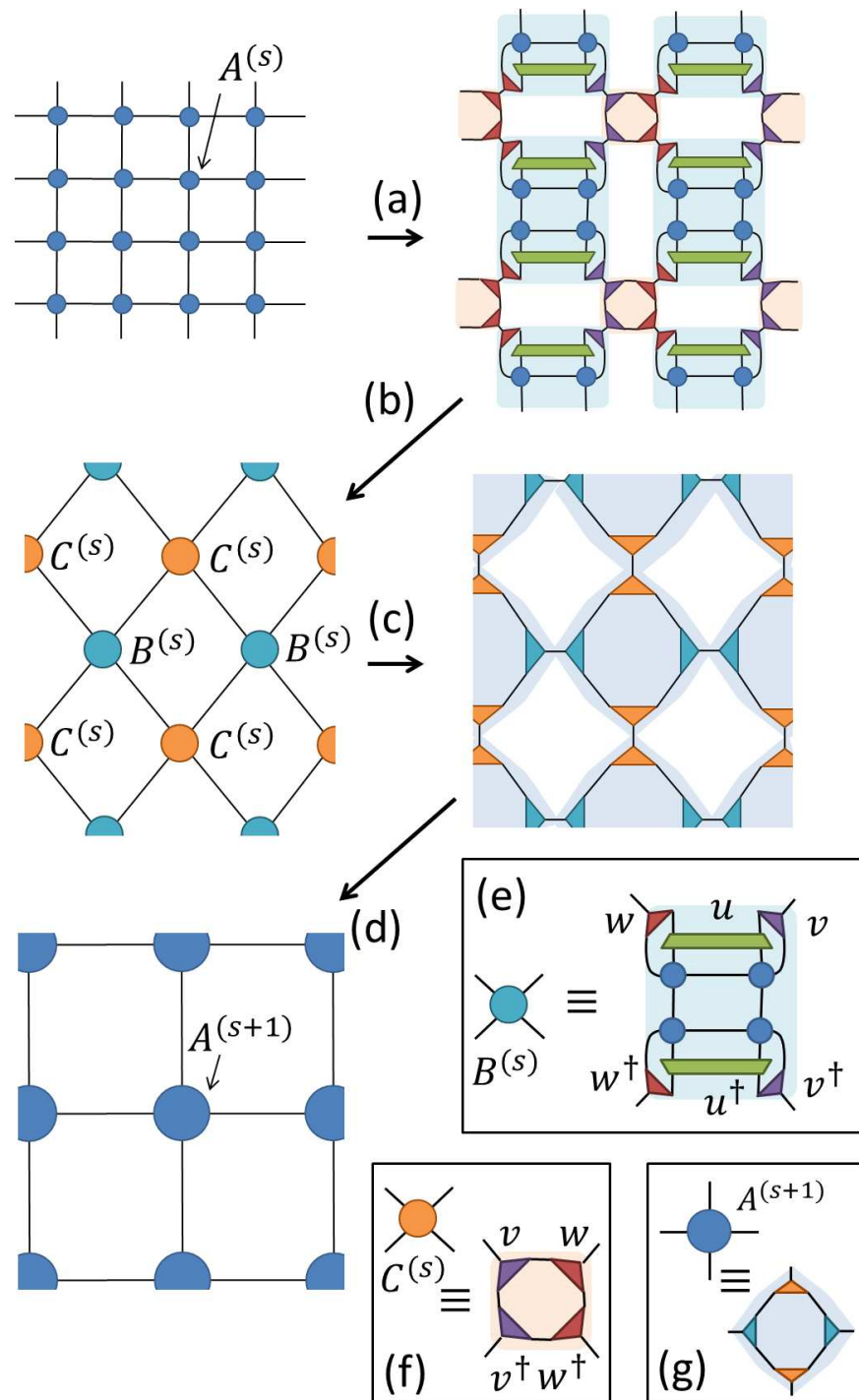
- TEFR works well far from the critical point.
 - Because it can remove CDL structure.
- In the vicinity of the critical point, the accuracy is still poor.
 - Because the actual entanglement is not necessarily perfect CDL structure.
- In order to improve further, we need to consider the entanglement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

Recent progress: Tensor Network Renormalization

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

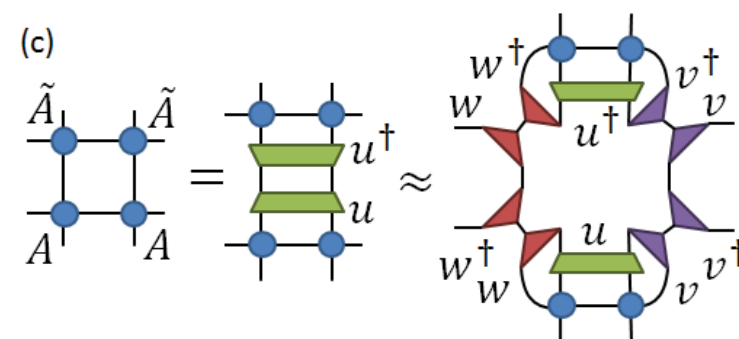
arXiv:1412.0732.

Tensor Network Renormalization



Point of TNR

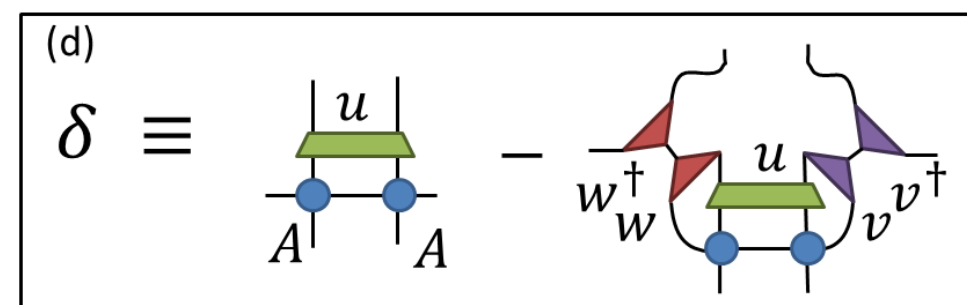
Use of a **disentangler** (Unitary tensor)



It can remove **short range entanglement** efficiently.

(Not only CDL)

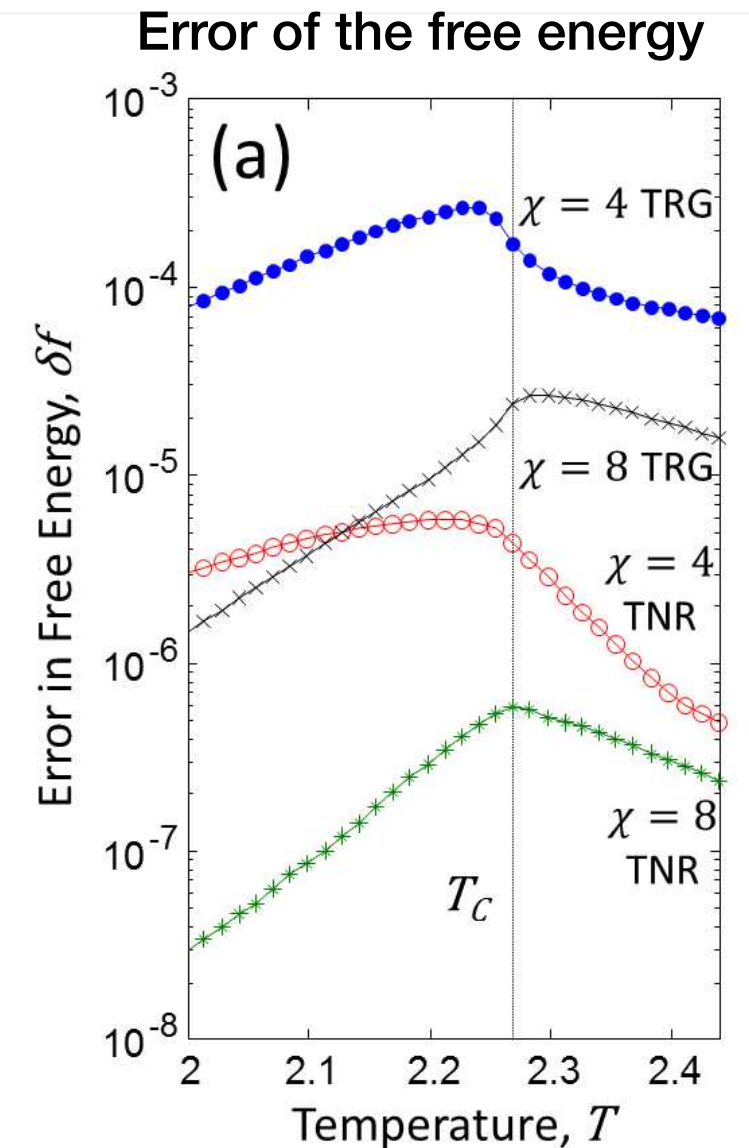
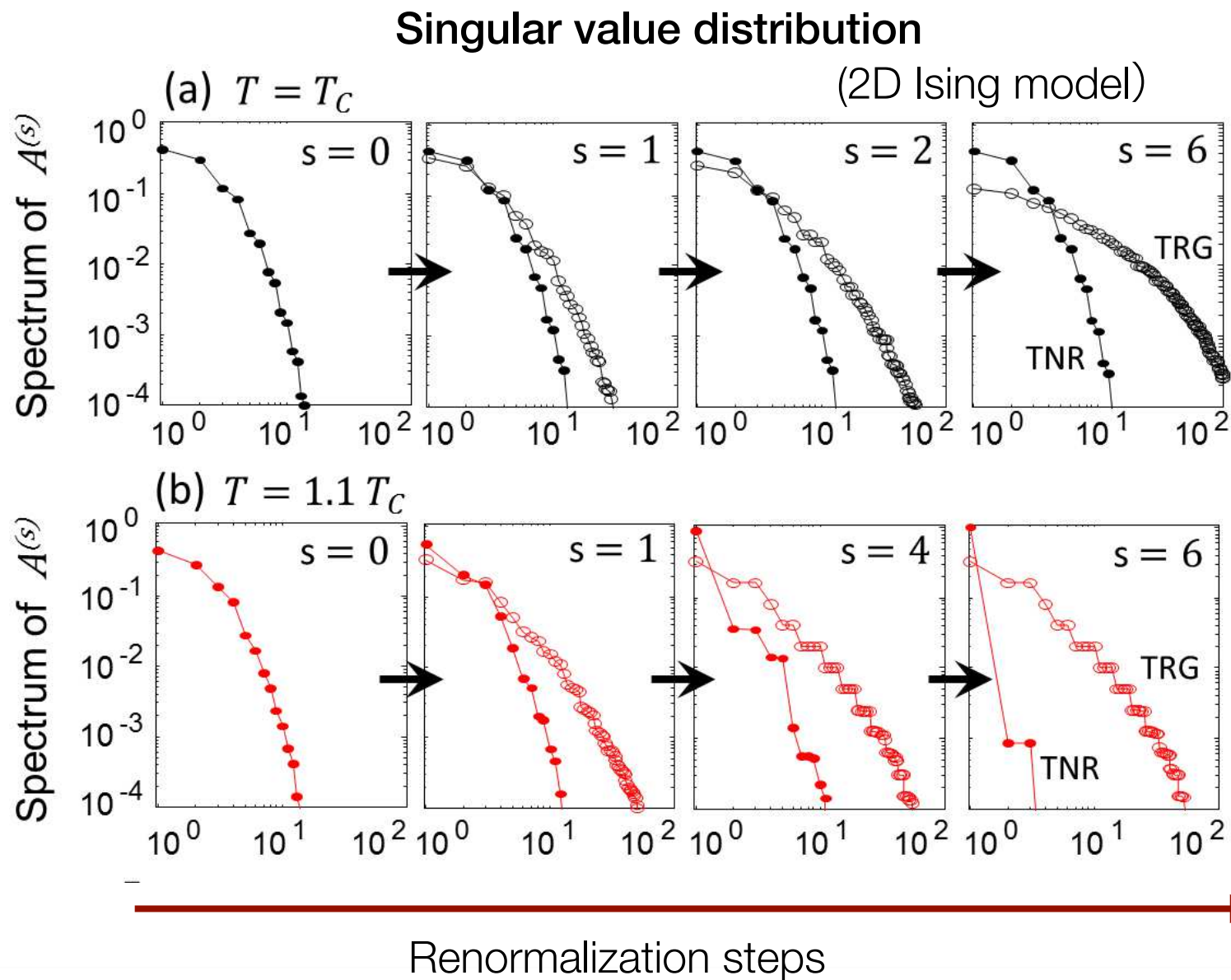
Approximation by using two-tensor cluster:



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 T_c .
 - 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

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 arXiv:1709.07460

- 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)

Report problems

- Please solve both of the problems:
 1. Report problem (EE in TPS)
 2. Report problem (TRG)
- Please include your name and student id in your report.
- Please submit through ITC LMS
(If you have any troubles, please send us email:
t-okubo@phys.s.u-tokyo.ac.jp)
- Deadline is 2018/1/31 (23:30)

Report problem (EE in TPS)

1. Explain the upper bound of entanglement entropy of MPS (compulsory)
 1. Consider enough large vector with dimension m^N and suppose it is represented by MPS with bond dimension χ .
(The vector is normalized.)
 2. Divide the system into two part (for example at the center of MPS), and represent the reduced density matrix by matrices of MPS.
 3. Show the rank of the reduced density matrix by using χ .
(Please include "derivation" of the result.)
 4. Explain the upper bound of the entanglement entropy calculated from the reduced density matrix.

You can use the following fact:

$$-\text{Tr } \rho \log \rho \leq \log(\text{dim} \rho) \quad \text{(*Corrected after the lecture)}$$

ρ : (reduced) density matrix

Report problem (EE in TPS)

2. Explain the upper bound of entanglement entropy of TPS (optional)

1. By using similar argument to the case of MPS, explain the upper bound of EE of TPS (on d -dimensional cubic lattice).
2. Explain general relation between upper bound of EE and tensor network structure.

Report problem (TRG)

1. Calculate the singular value spectrum obtained by TRG (compulsory)
 1. Consider classical Ising model on the square lattice.
 2. Perform TRG in several steps for a few temperatures with bond dimension D .
 3. Plot singular values at the SVD (Please normalize it as $\sum_i \sigma_i = 1$.)
 4. By changing temperatures and the bond dimensions, discuss the shape of singular value distribution.(Recommendation: try around T_c and far from T_c)

This can be done by sample python code Report_TRG.py

Usage: `python Report_TRG.py -D D -n n -T T`

output:

```
okubo$ python Report_TRG.py -D 8 -n 4 -T 2.0
T= 2.0
L= 16
D= 8
free energy density= -2.05699380572
```

Report problem (TRG)

1. Calculate the singular value spectrum obtained by TRG (compulsory)

Usage: `python Report_TRG.py -D D -n n -T T`

D : bond dimension

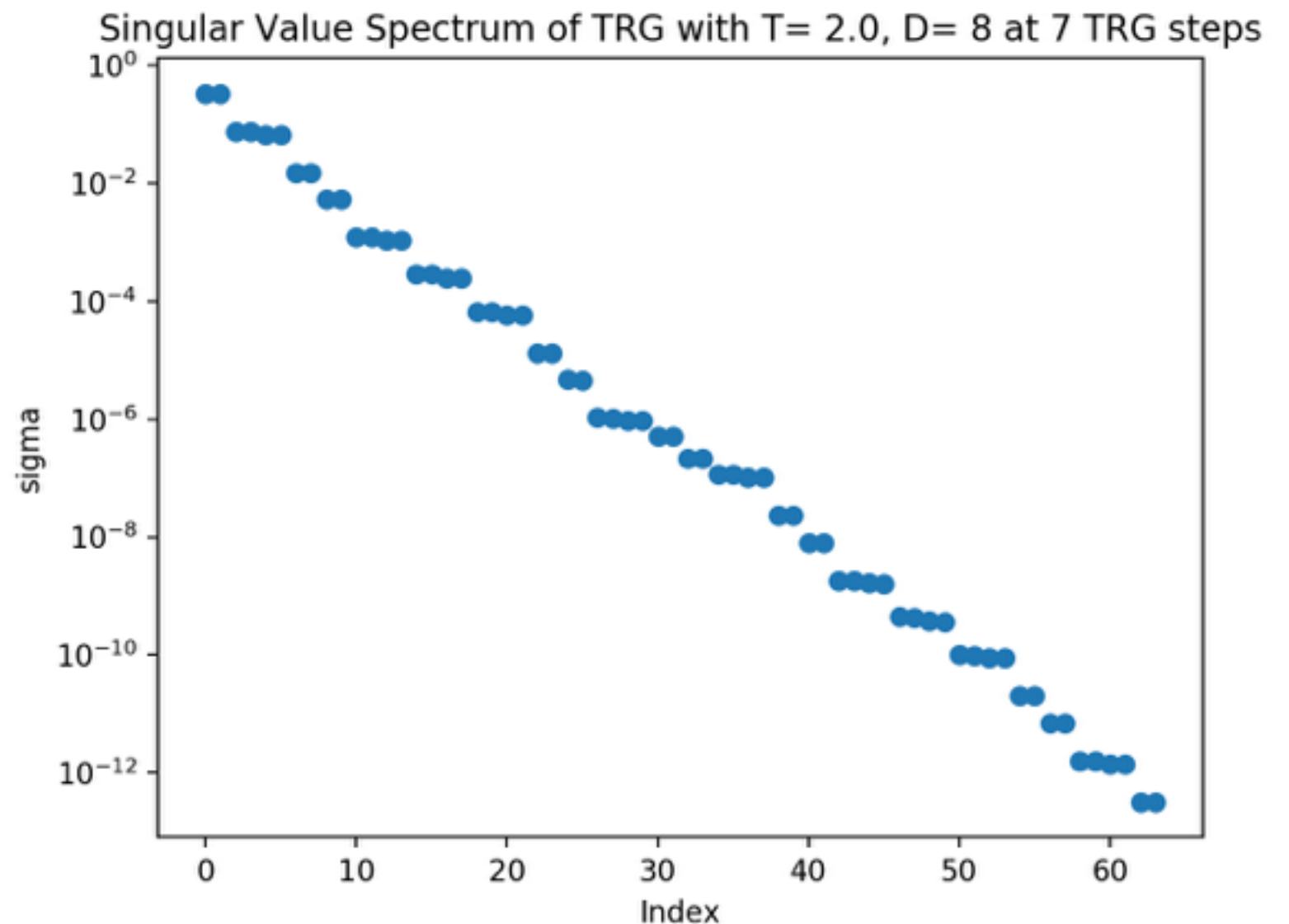
T : temperature

n : related to TRG steps
through the relation

$$\text{TRG step} = 2n - 1$$

The standard output is
free energy density of
 $L=2^n$

Ising model ($N=L \times L$)



Keywords

- 第1回： 現代物理学における巨大なデータ
- 第2回： 情報圧縮と繰り込み
- 第3回： 情報圧縮の数理 1 (線形代数の復習)
- 第4回： 情報圧縮の数理 2 (特異値分解と低ランク近似)
- 第5回： 情報圧縮の数理 3 (スパース・モデリングの基礎)
- 第6回： 情報圧縮の数理 4 (クリロフ部分空間法の基礎)
- 第7回： 物質科学における情報圧縮
- 第8回： スパース・モデリングの物質科学への応用
- 第9回： クリロフ部分空間法の物質科学への応用
- 第10回： 行列積表現の基礎
- 第11回： 行列積表現の応用
- 第12回： テンソルネットワーク表現への発展
- 第13回： テンソルネットワーク繰り込みと低ランク近似の応用