計算科学における情報圧縮

Information Compression in Computational Science

2019.1.10

#13:テンソルネットワーク繰り込みによる情報圧縮

Information compression by 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

(Share problems)

- Tensor network renormalization
  - Tensor Renormalization Group (TRG) in two dimension
  - Generalization to higher dimensions

(Understand idea of TRG)

- Tensor network renormalization around critical point
  - Fixed point of TRG: Corner double line tensors

(Understand "entanglement structure")

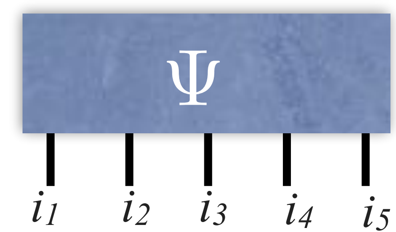
- Report problems

Tensor network representation of a scalar

Tensor network state: approximation for a vector

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

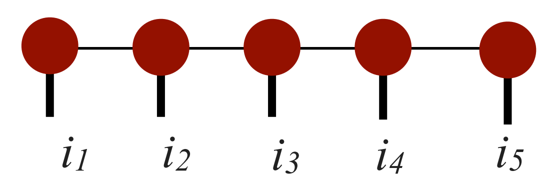
Vector (or N-rank tensor): $\Psi_{i_1 i_2 \ldots i_N}$ =



$\boxed{\text{\# of Elements} = a^N}$

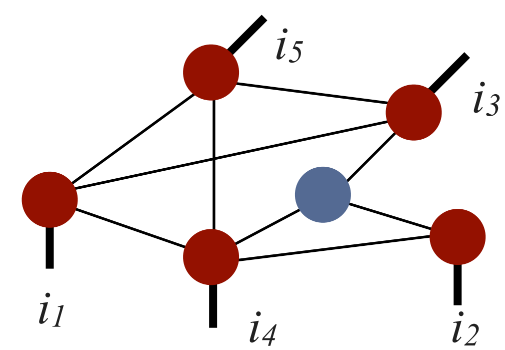
"Tensor network" decomposition

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



$A[m]$: Matrix for state m

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



X,Y : Tensors

Tr : Tensor network contraction

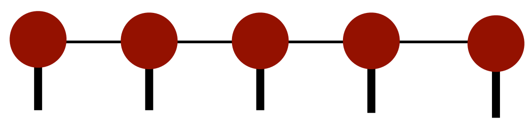
By choosing a "good" network, we can express G.S. wave function efficiently.

ex. $\boxed{\text{MPS: \# of elements} = 2ND^2}$ D: dimension of the matrix A

Exponential→ Linear *If D does not depend on N…

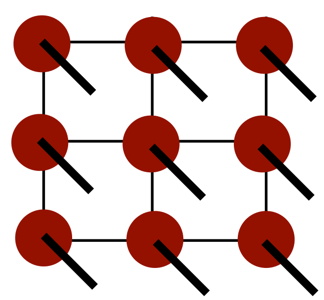
Examples of TNS

MPS:



Good for 1-d gapped systems

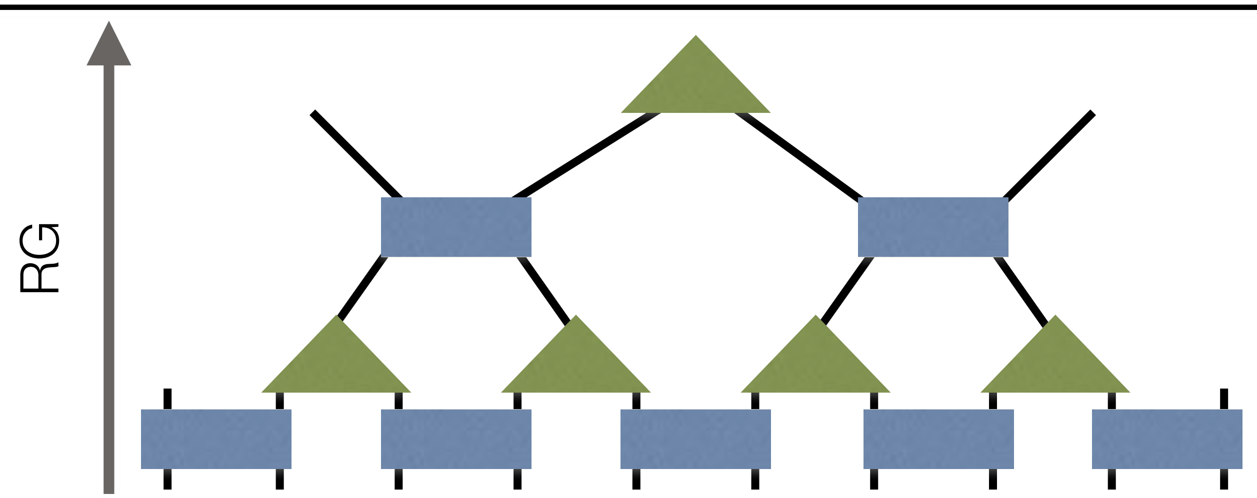
PEPS, TPS:



For higher dimensional systems

Extension of MPS

MERA:

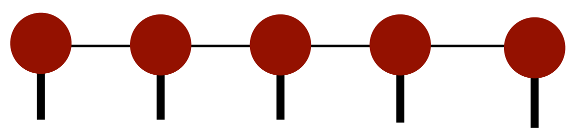


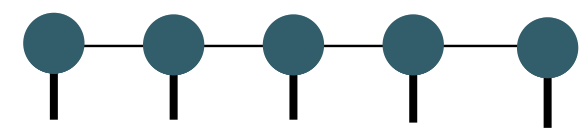
Scale invariant systems

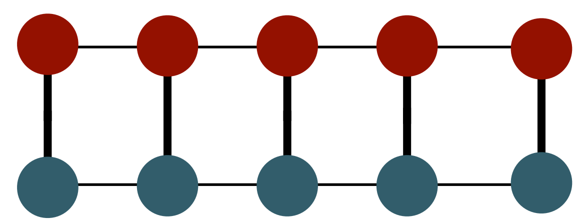
Tensor network representation of a scalar

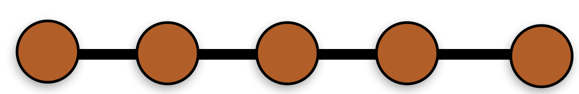
Example: inner product of two TNSs

MPS

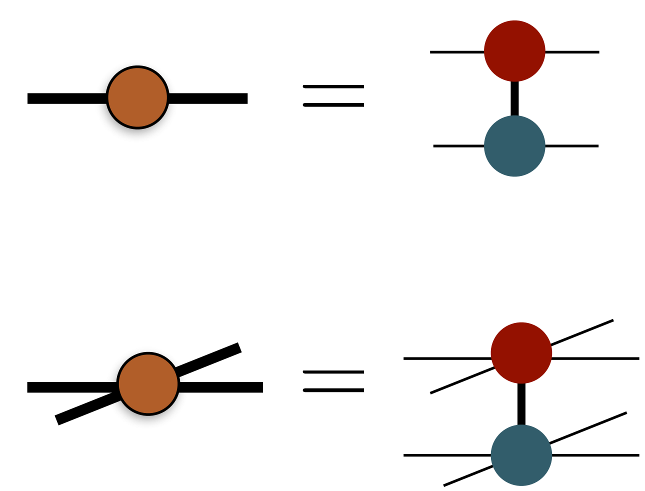
$\vec{v} =$ 

$\vec{u} =$ 

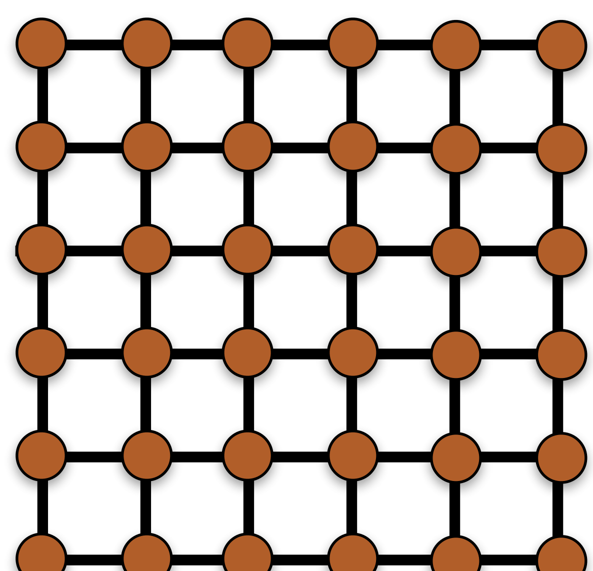
$\vec{v} \cdot \vec{u}^* =$ 

$=$ 

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

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

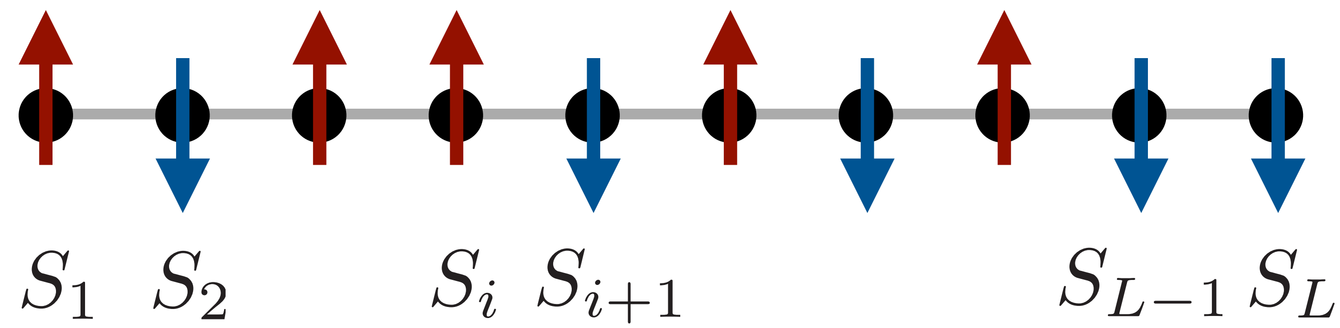
log of the partition function＝Free energy

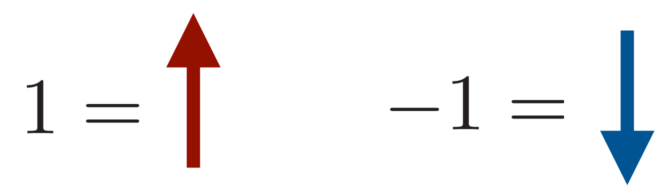
Tensor network representation of partition function

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:

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

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

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

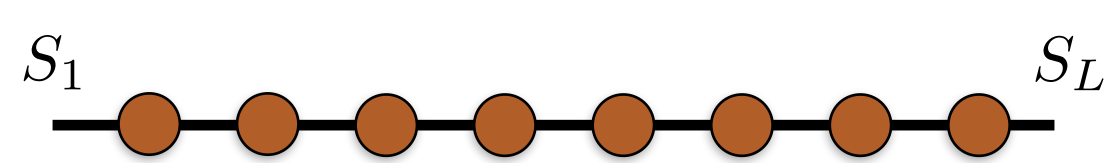
Transfer matrix
(転送行列)

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

$$T = \begin{array}{cc} +1 & -1 \\ \end{array}$$

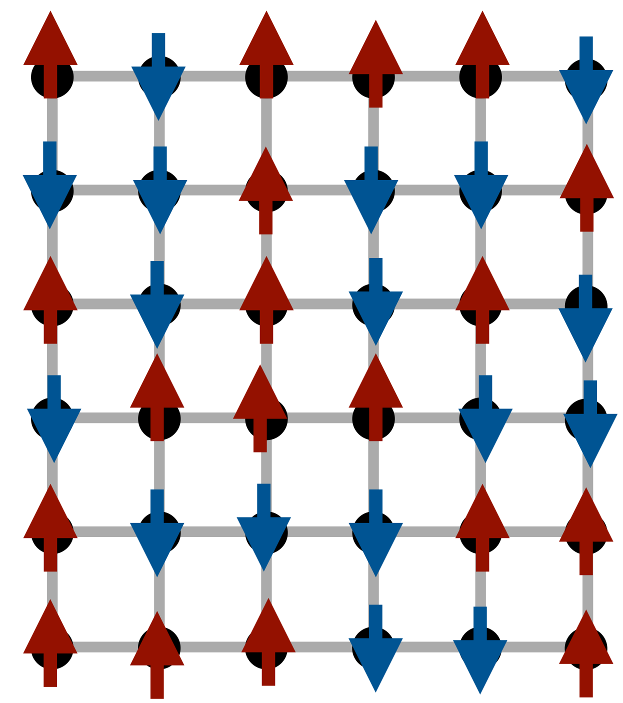
$$T = \begin{array}{c} \begin{array}{cc} \quad +1 & \quad -1 \end{array} \\ \begin{pmatrix} e^{\beta J} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J} \end{pmatrix} \end{array} \begin{array}{c} \\ +1 \\ -1 \end{array}$$

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



Tensor network representation in two dimension

Classical Ising model on the square lattice



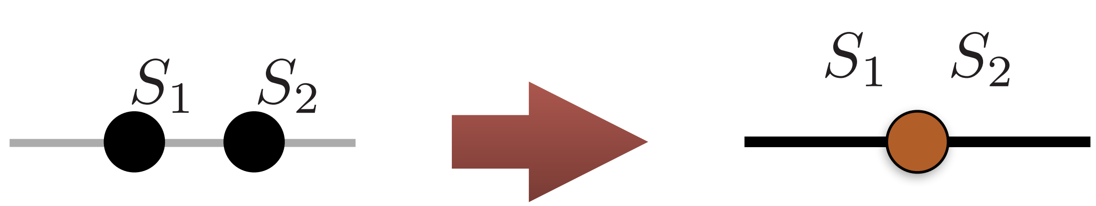
$$\mathcal{H} = -J\sum_{\langle i,j\rangle} S_i S_j \qquad (S_i = \pm 1 = \uparrow, \downarrow)$$

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

We can use a tensor instead of the transfer matrix.

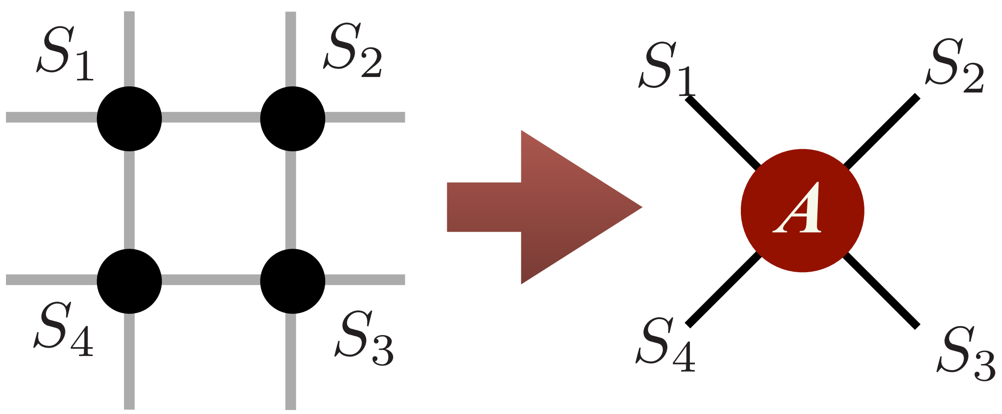
$$e^{\beta J S_1 S_2} = T_{S_1 S_2}$$

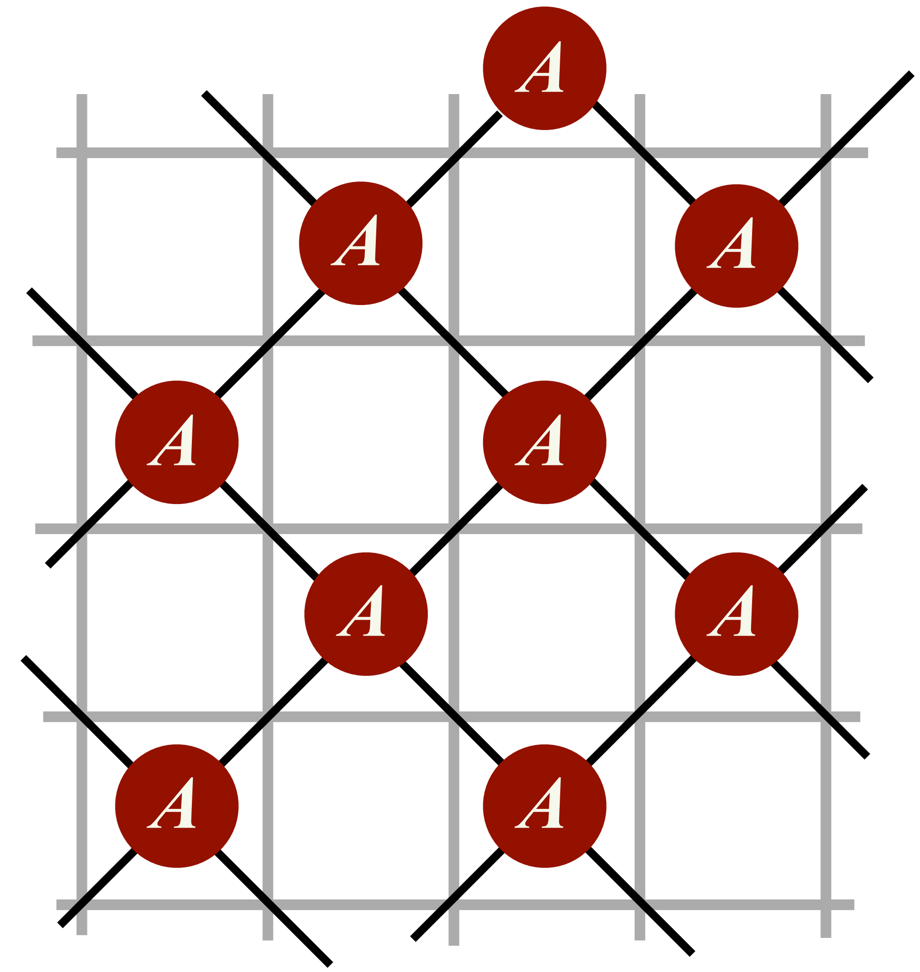
$\Rightarrow$ Tensor?



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)

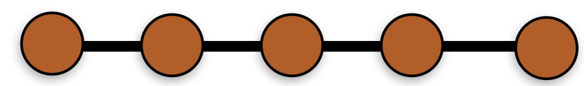
$O(ND^3)$ (Periodic)

Square: $O(D^L)$ (Open)

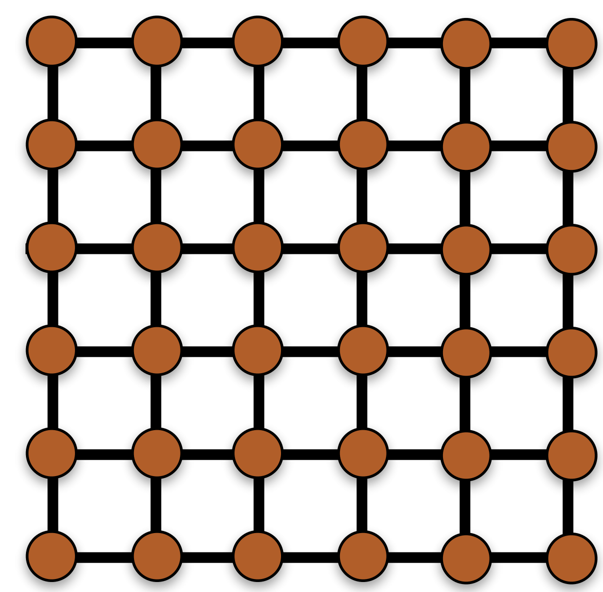
$O(D^{2L})$ (Periodic)

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

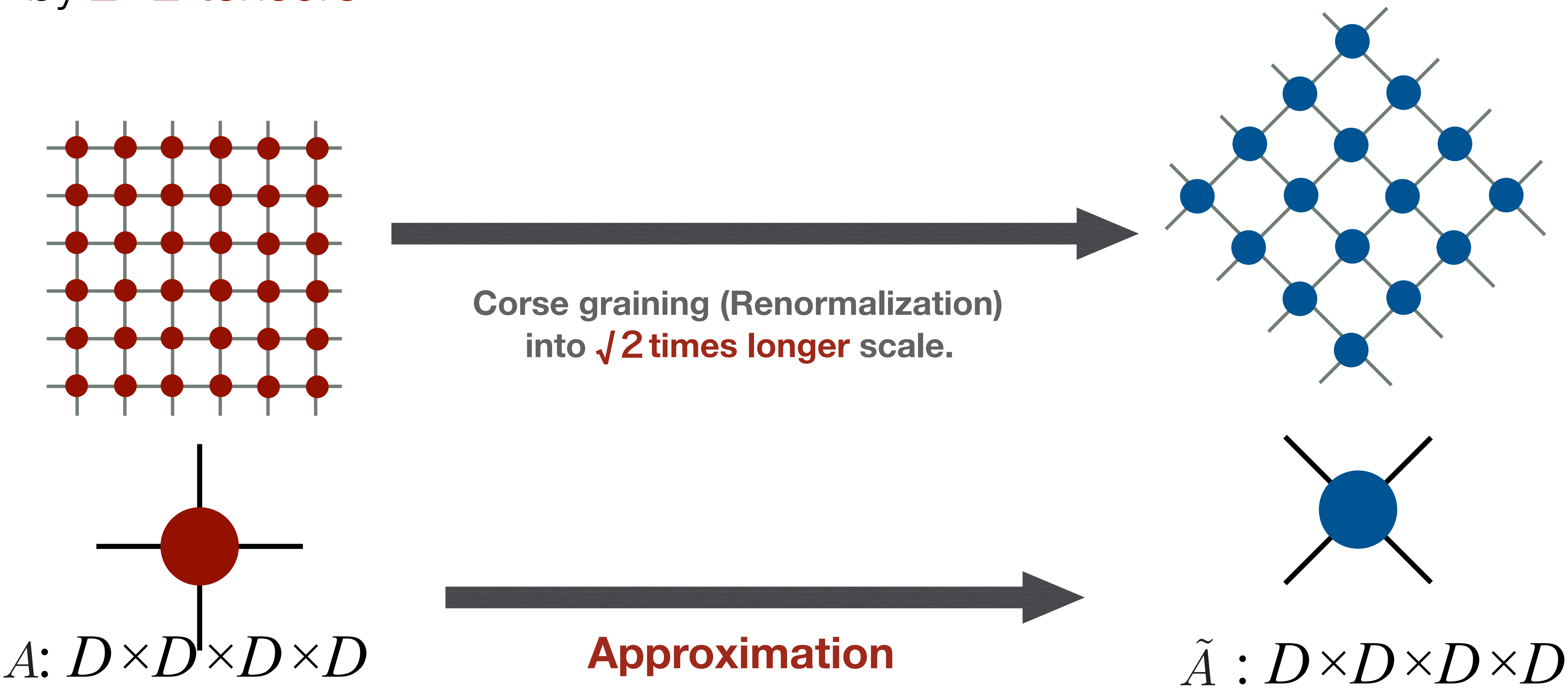
Tensor network renormalization
（テンソルネットワーク繰り込み）

- 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

Scaler represented by $L \times L$ tensors

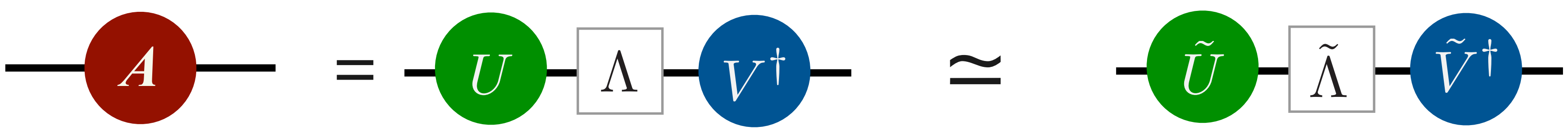
$(L \times L)/2$ tensors



Reduce the number of tensors keeping their size constant

Key technique: low rank approximation by SVD

$\boxed{\text{Best low-rank approximation of a matrix = SVD}}$



$A : M \times N$

(M ≦ N)

$\Lambda : M \times M$

(Diagonal matrix)

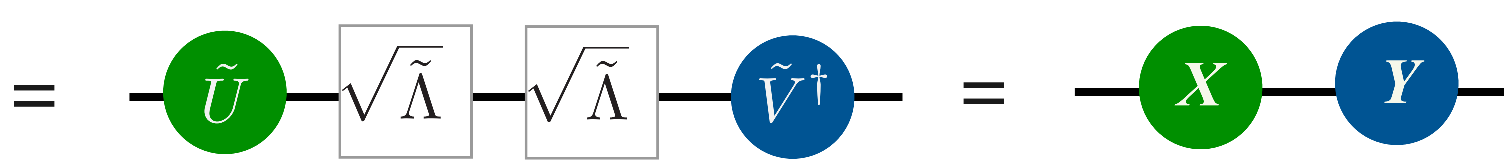
$U, V : (\mathrm{M,N}) \times M$

$\tilde{\Lambda} : R \times R$

(Keeping the R largest singular values)

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

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.

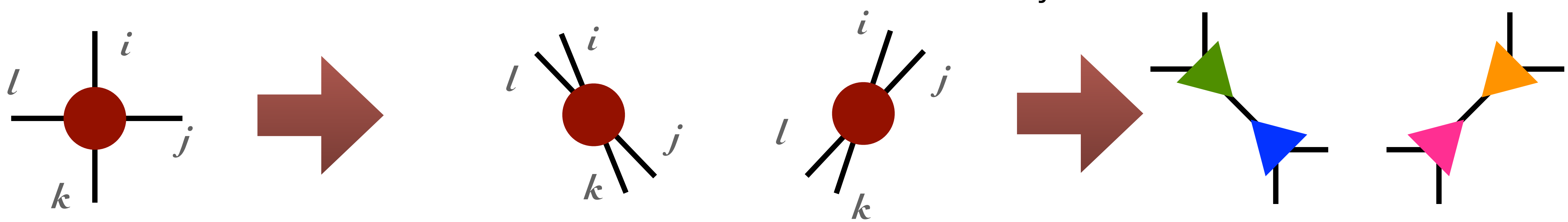
Recipe of Tensor Renormalization Group (TRG)

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

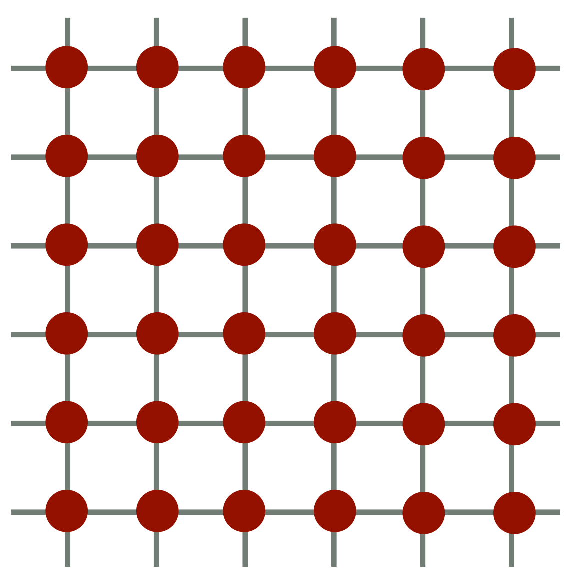
D-rank approximation by SVD



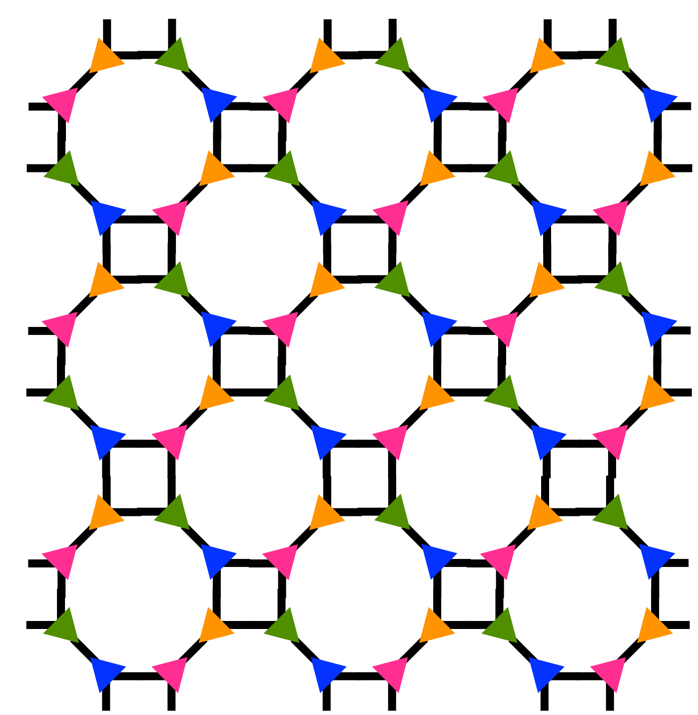
$A_{i,j,k,l}$ $\qquad$ $A_{(i,l),(j,k)}$ $\qquad$ $A_{(i,j),(k,l)}$

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

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



Approximation



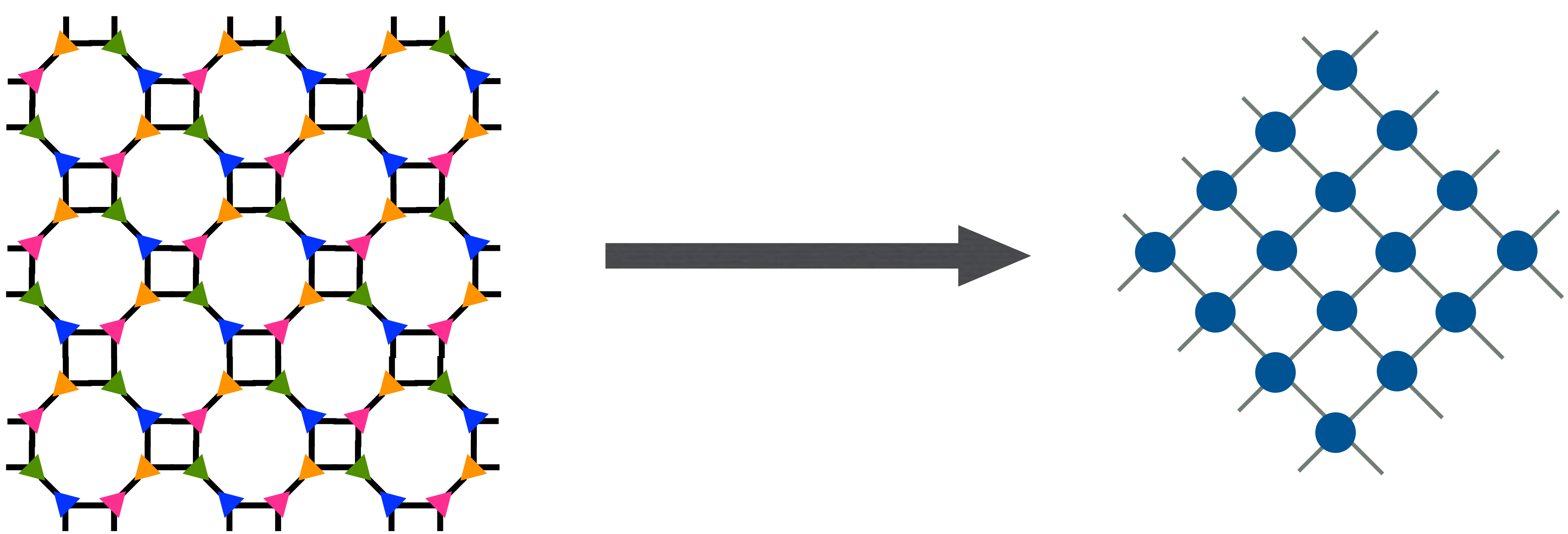
Recipe of Tensor Renormalization Group (TRG)

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

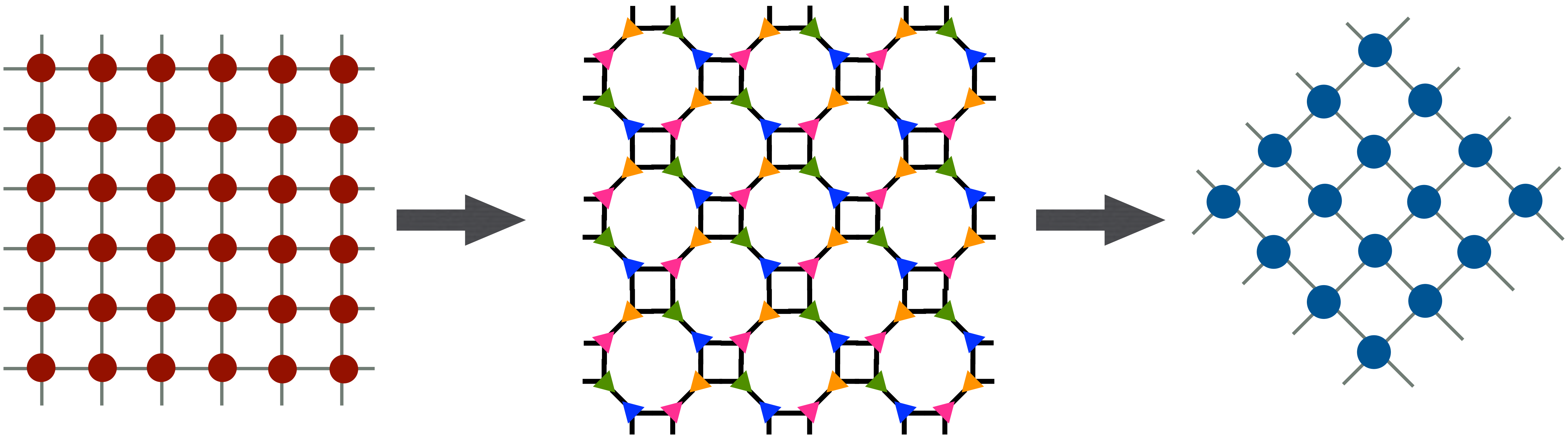
$$\tilde{A} : D \times D \times D \times D$$

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: SVD= $O(D^6)$

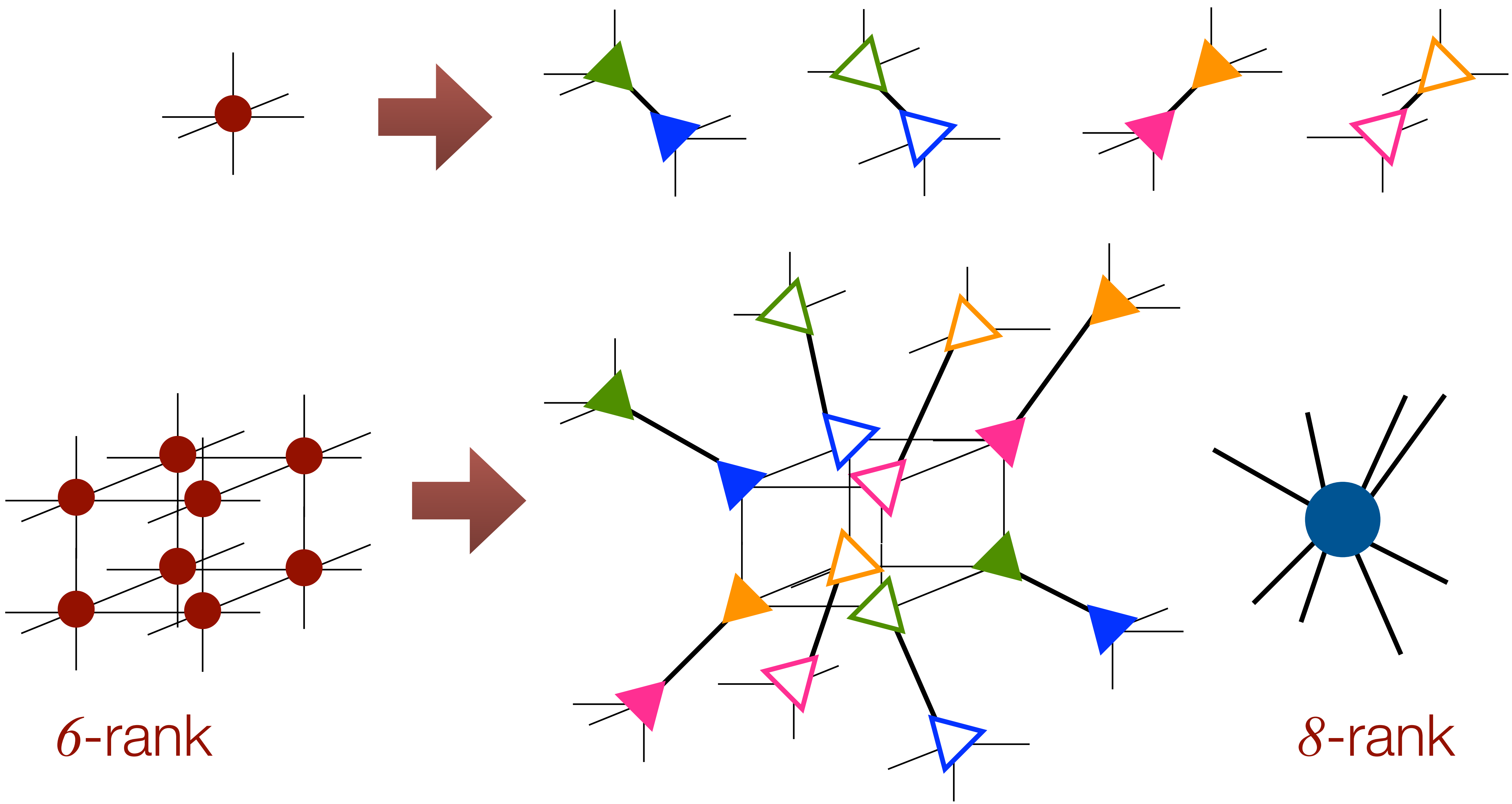
Contraction= $O(D^6)$ (per tensor)

*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

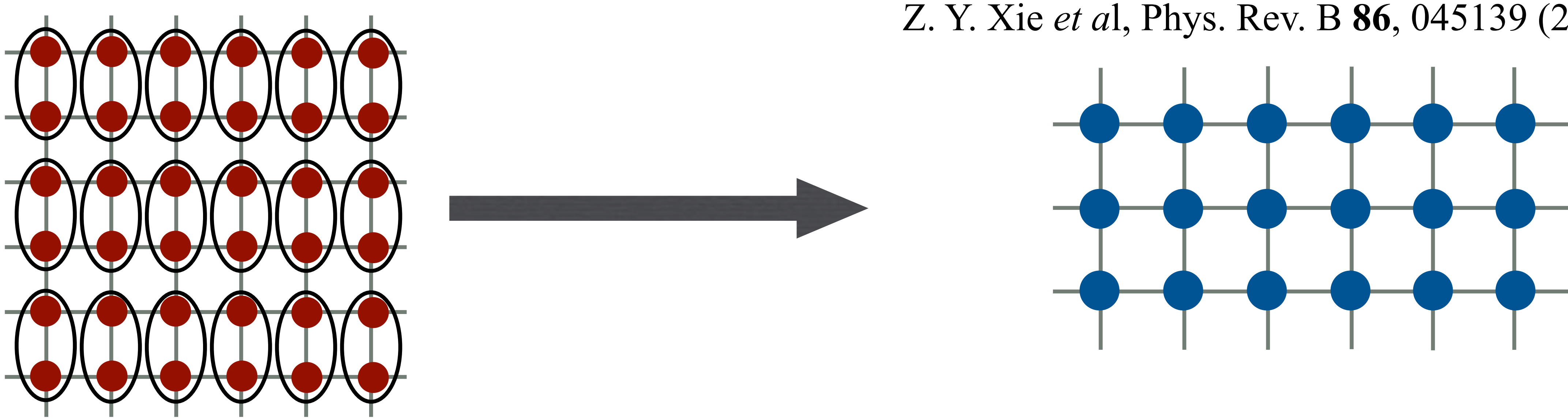
Simple generalization of TRG to cubic lattice (three dimension)



Tensor renormalization group by using HOSVD

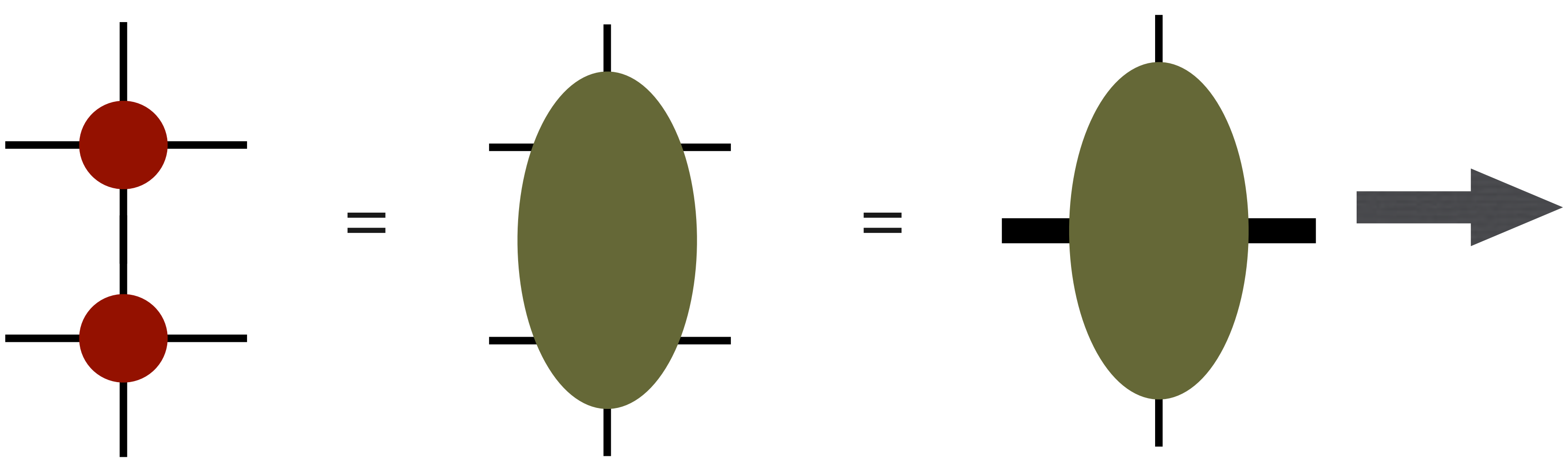
Anisotropic coarse graining by using HOSVD instead of SVD

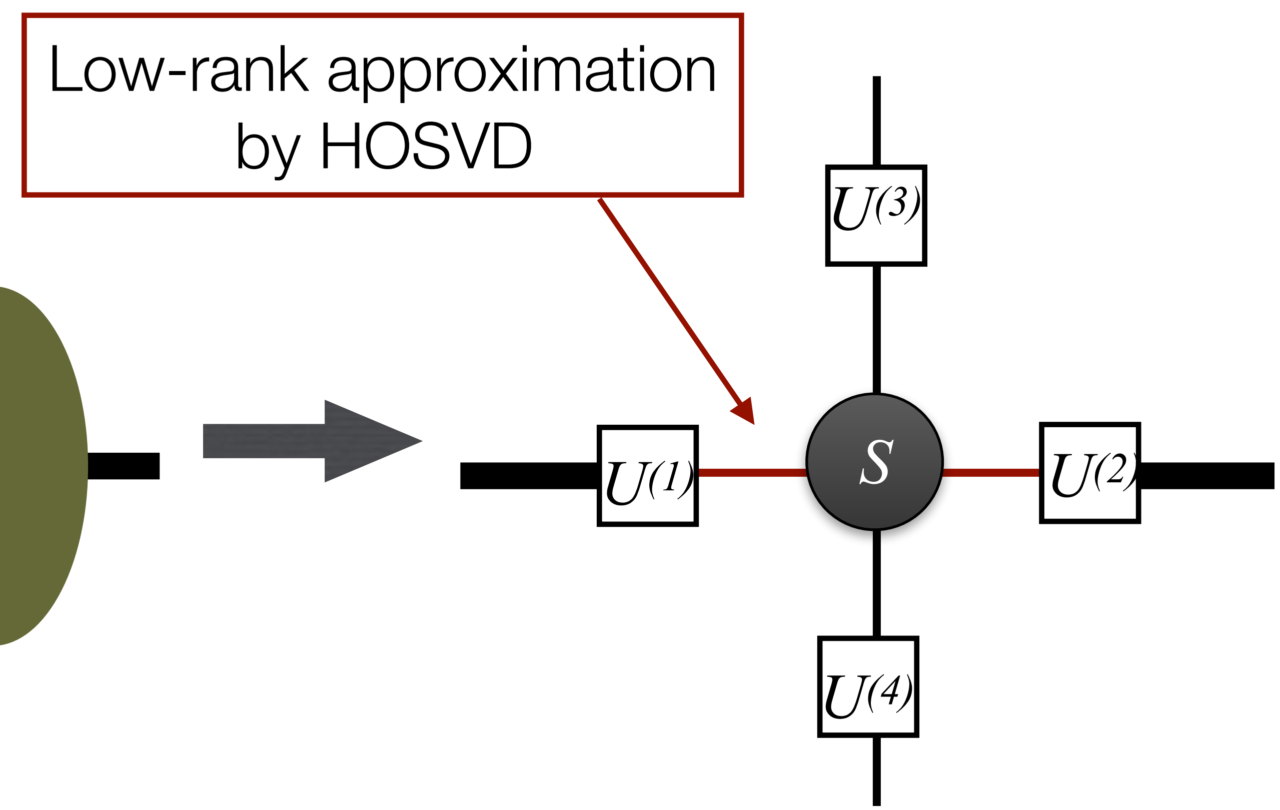
Z. Y. Xie *et al*, Phys. Rev. B **86**, 045139 (2012)



Basic idea of HOTRG algorithm:

(For details, see the original paper.)





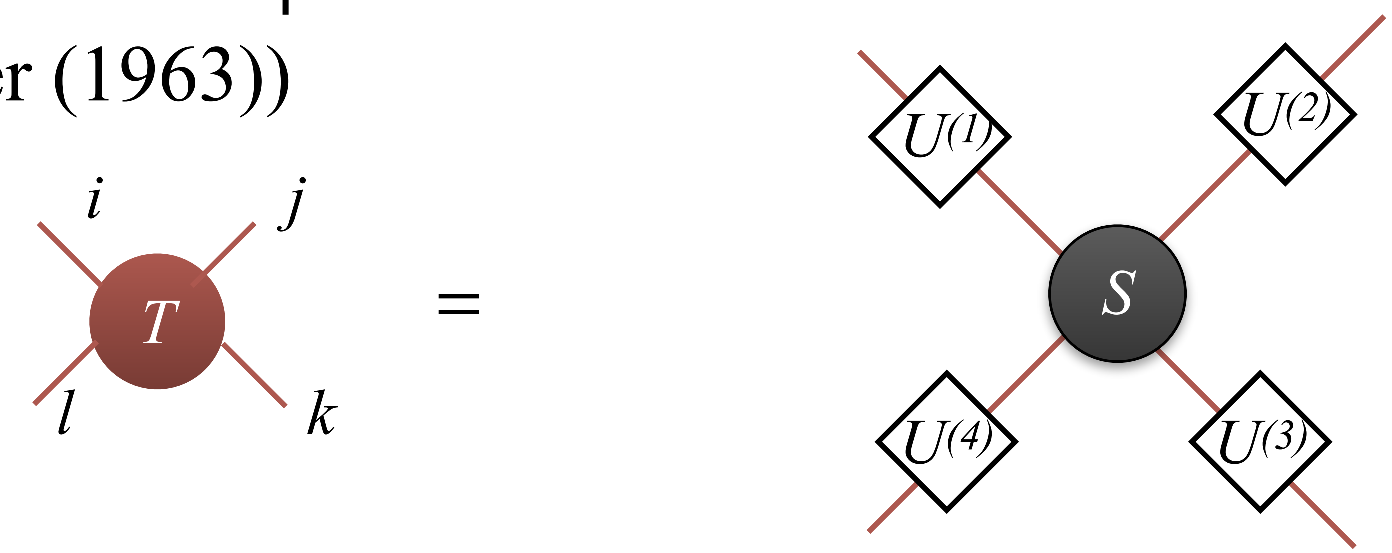
(Reference of HOSVD)

Tucker decomposition: generalization of SVD

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

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

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

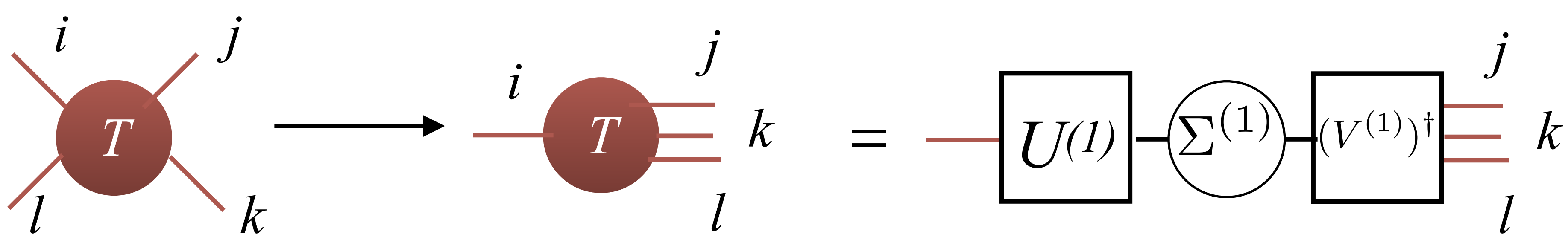
rank-(I',J', K', L') approximation

(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.

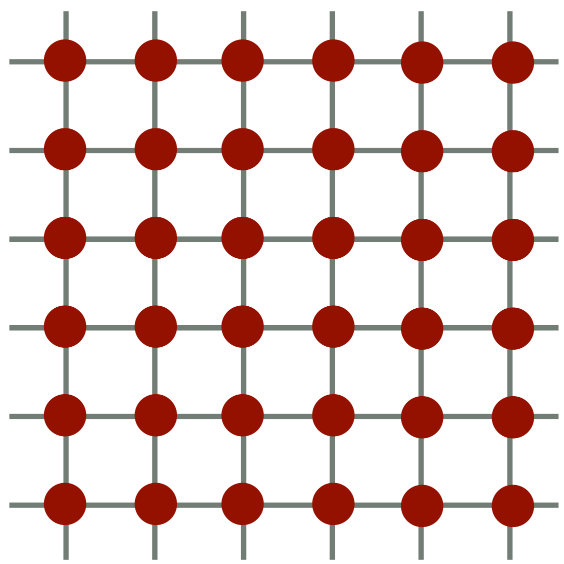
Disadvantage:

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

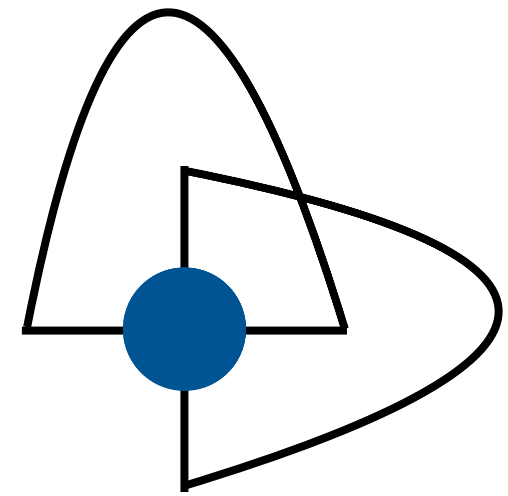
Application to a classical partition function

Partition function

(Periodic boundary condition)

$Z =$ 

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: $E = -\dfrac{\partial \ln Z}{\partial \beta}$ (Use difference approximation)

Specific heat: $C = \dfrac{1}{k_B T^2} \dfrac{\partial^2 \ln Z}{\partial \beta^2}$ (Use difference approximation)

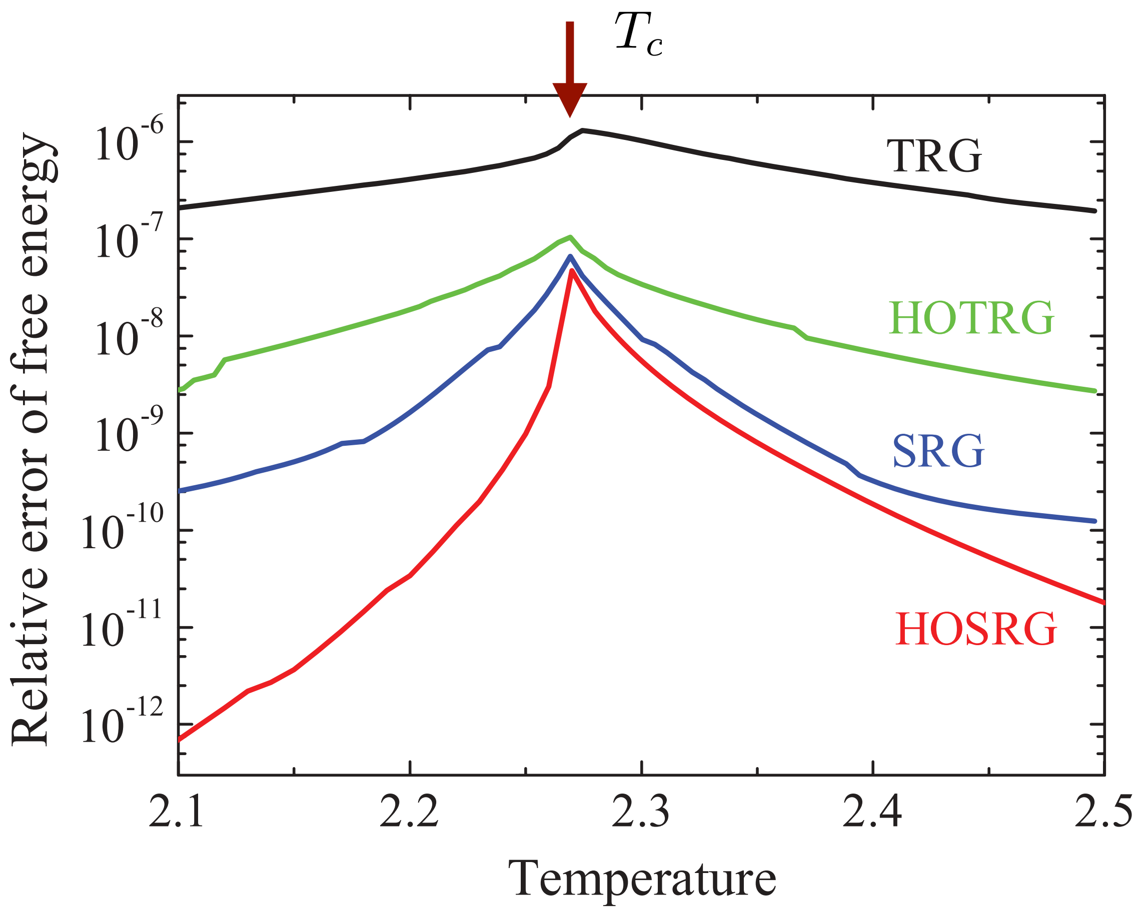
Example of calculation

Ising model in infinite size

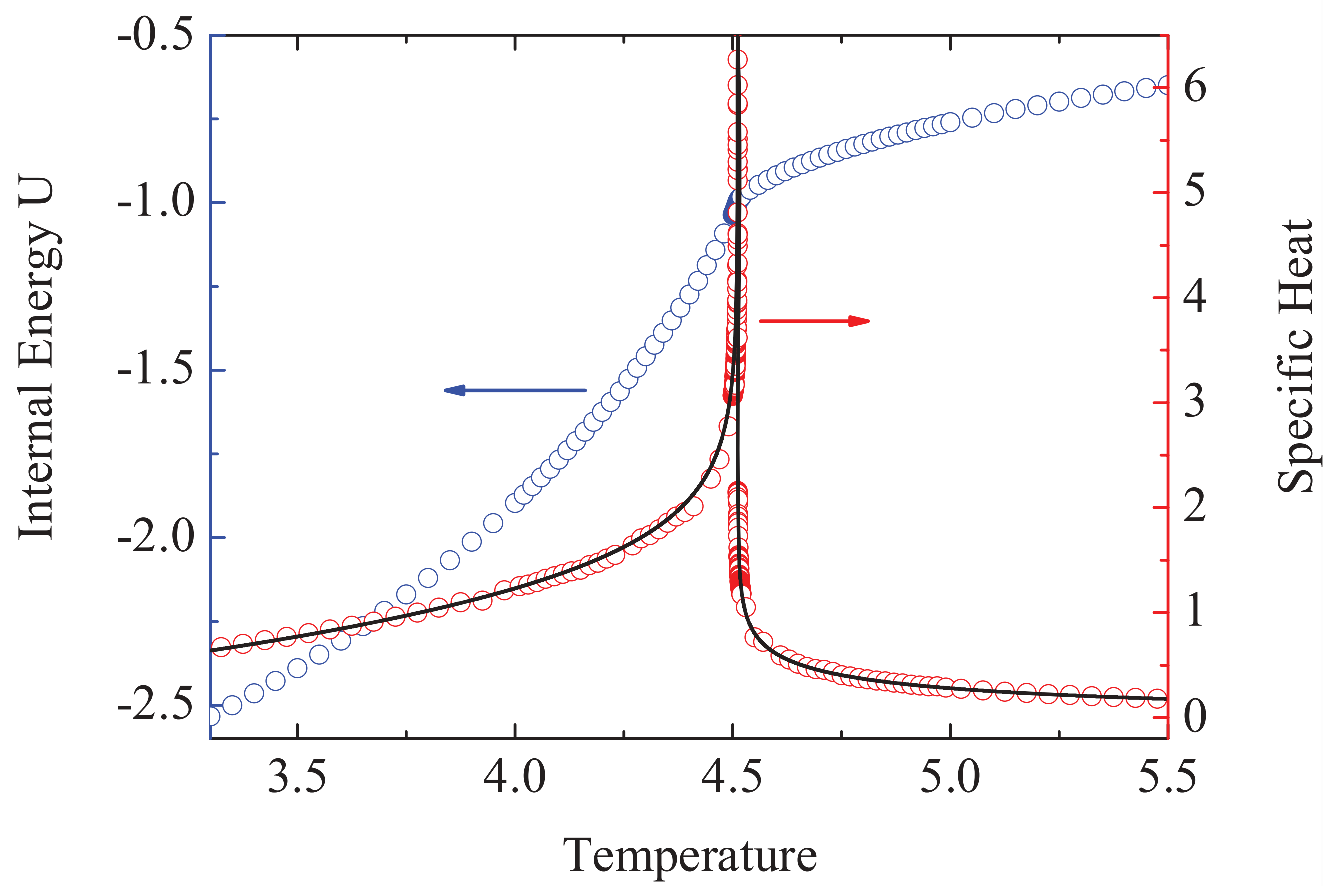
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

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



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

Tensor network renormalization at the critical point
— When the accuracy of TRG becomes worse?

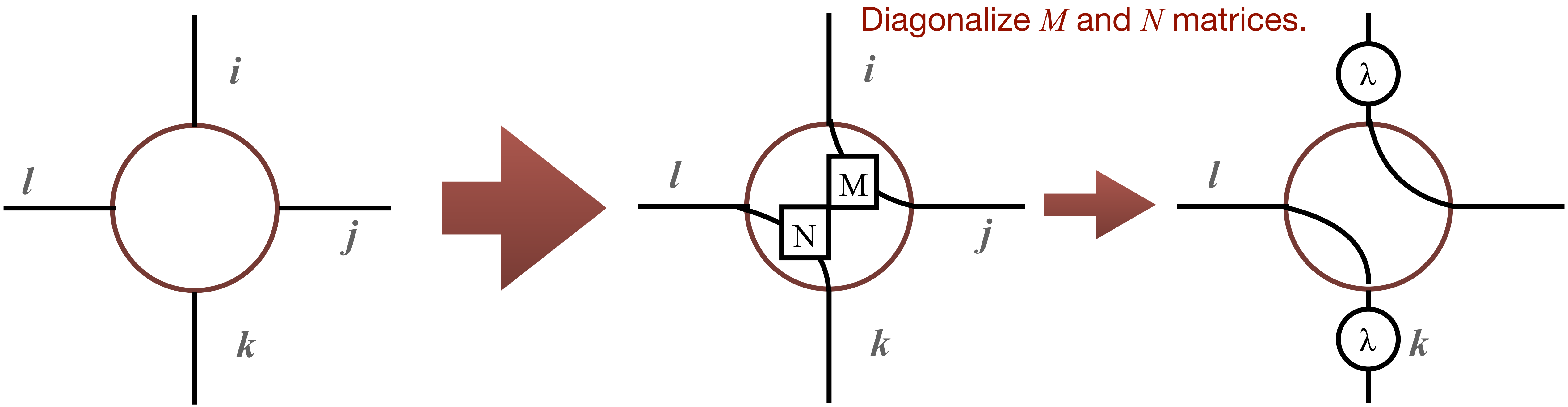
Correlation (entanglement) within a tensor

General tensor

A_{ijkl}

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

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



New rule for representation of the correlation:

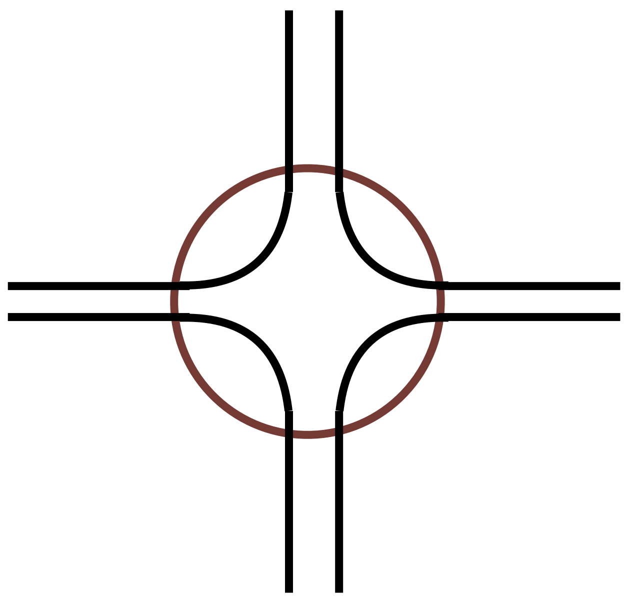
$$i \;\text{———}\; j = \delta_{ij}$$

(+ we neglect eigenvalues in the graph.)

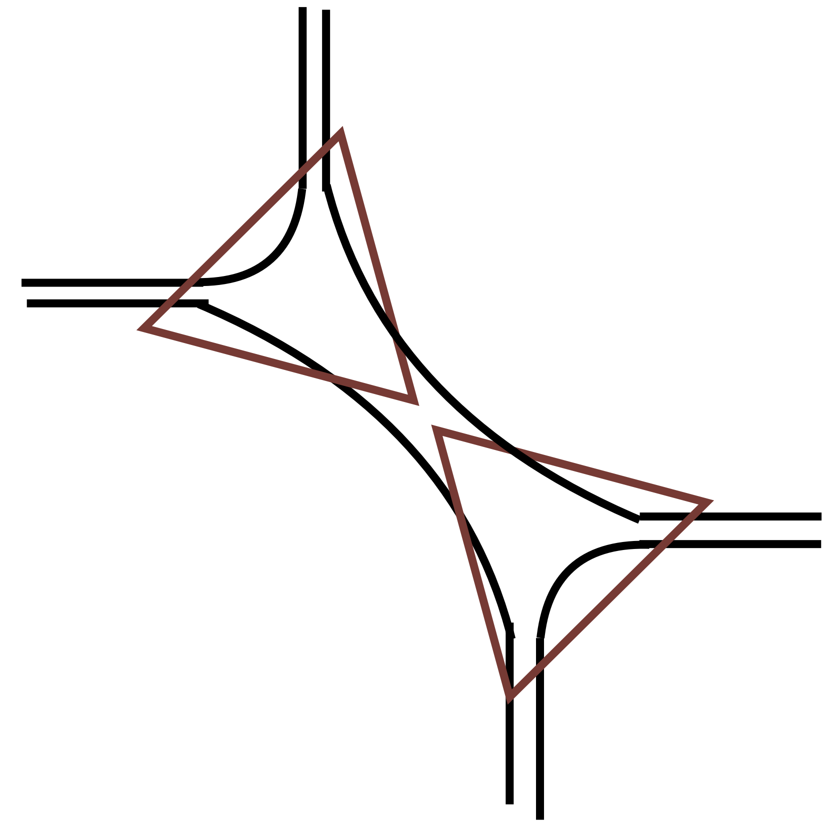
Fixed point of TRG: Corner Double Line tensor (固定点)

Corner Double Line (CDL) tensor:

There are correlations among the nearest legs.



SVD



Original bond dimension = D

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

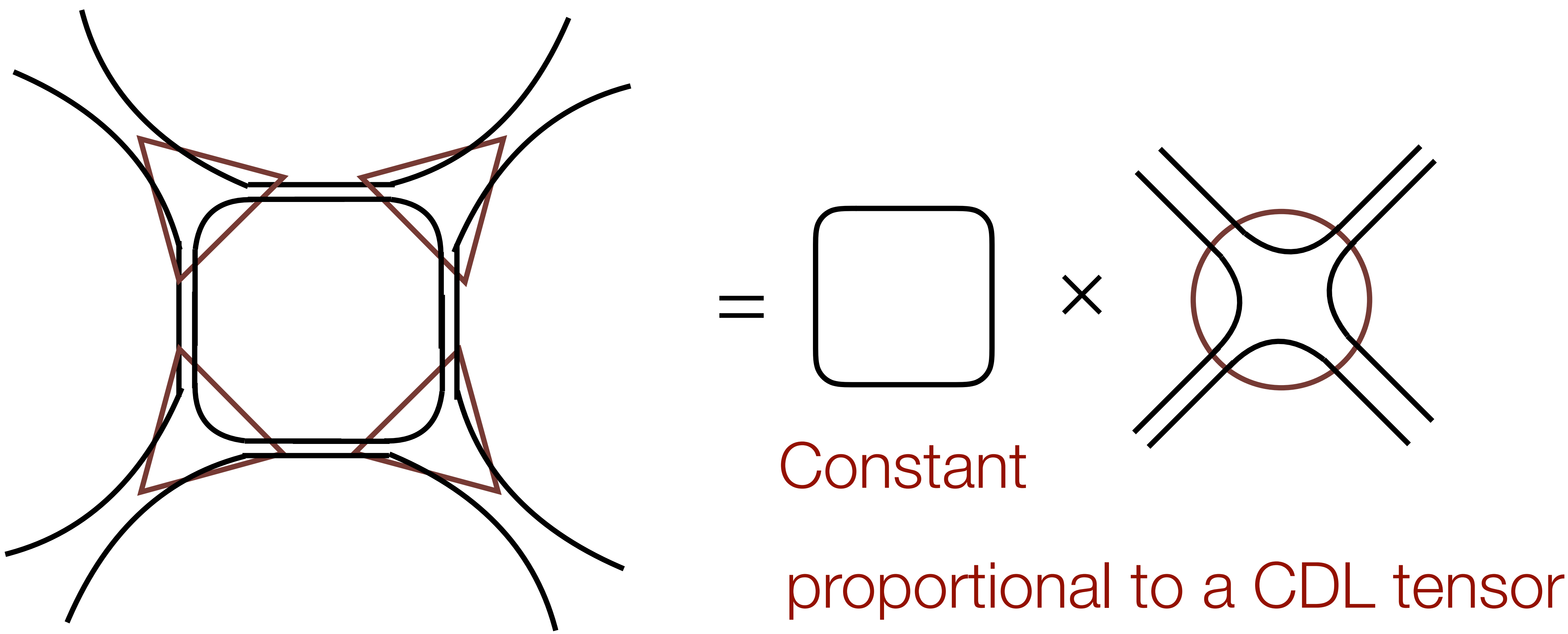
Degree of freedoms connecting two tensors.

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:



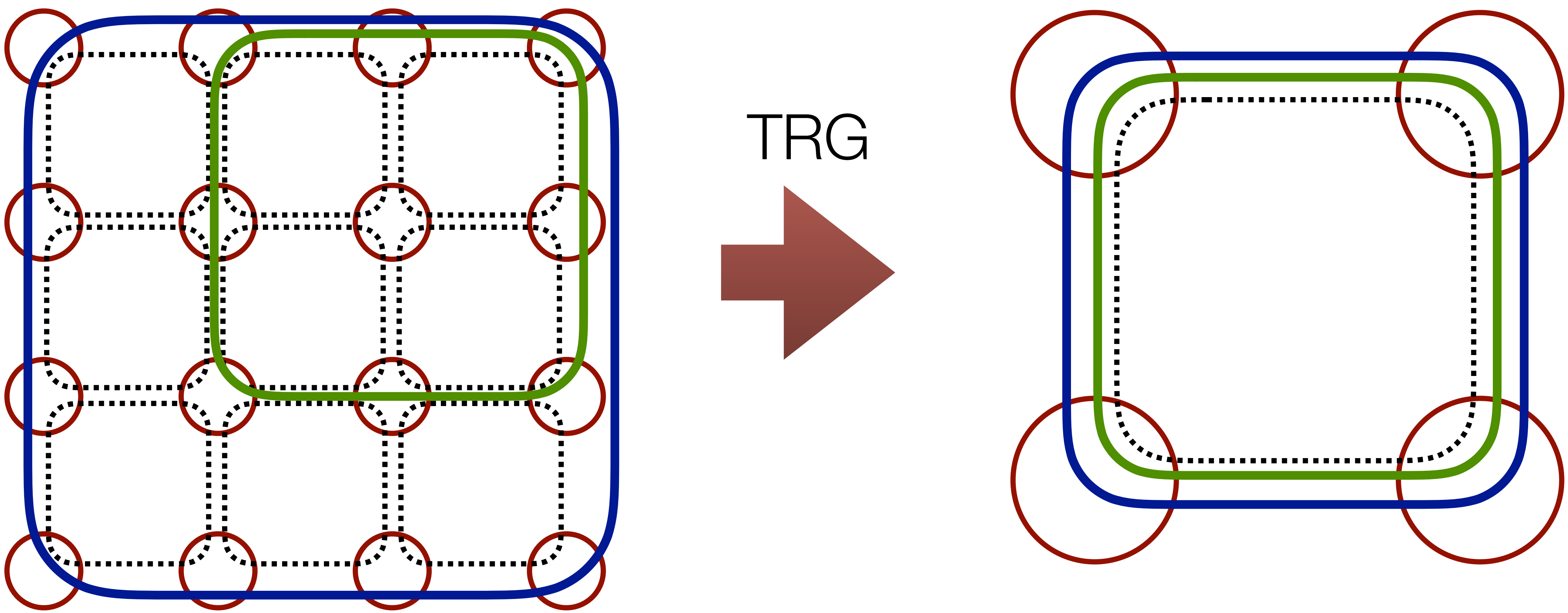
CDL tensor is a fixed point of TRG (and also HOTRG).

$\boxed{\text{CDL tensor remains as CDL tensor along TRG.}}$

Problems in TRG: accumulation of correlations

Correlation in several scales

Correlations remains after TRG.



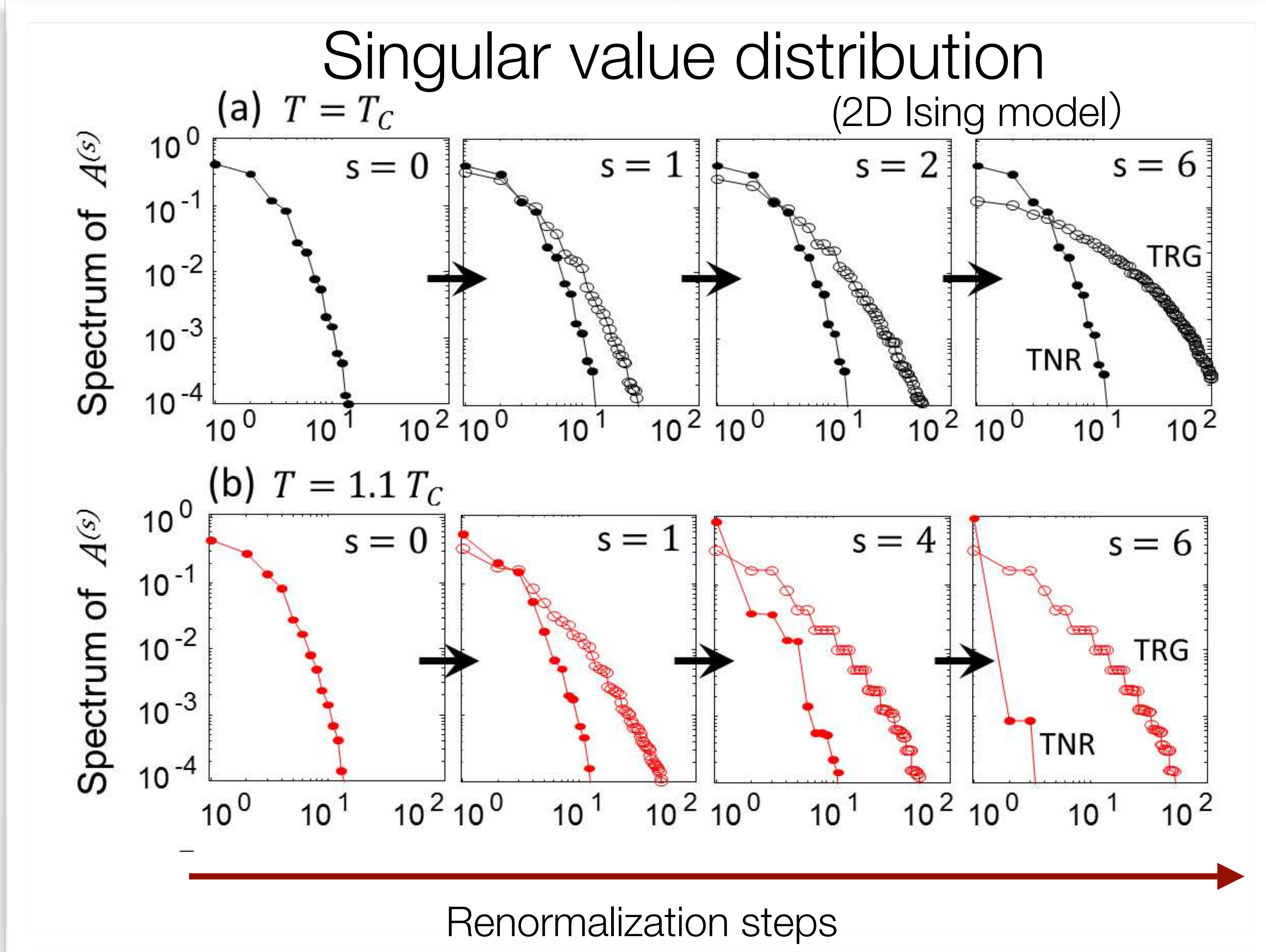
Ideal renormalization:

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.

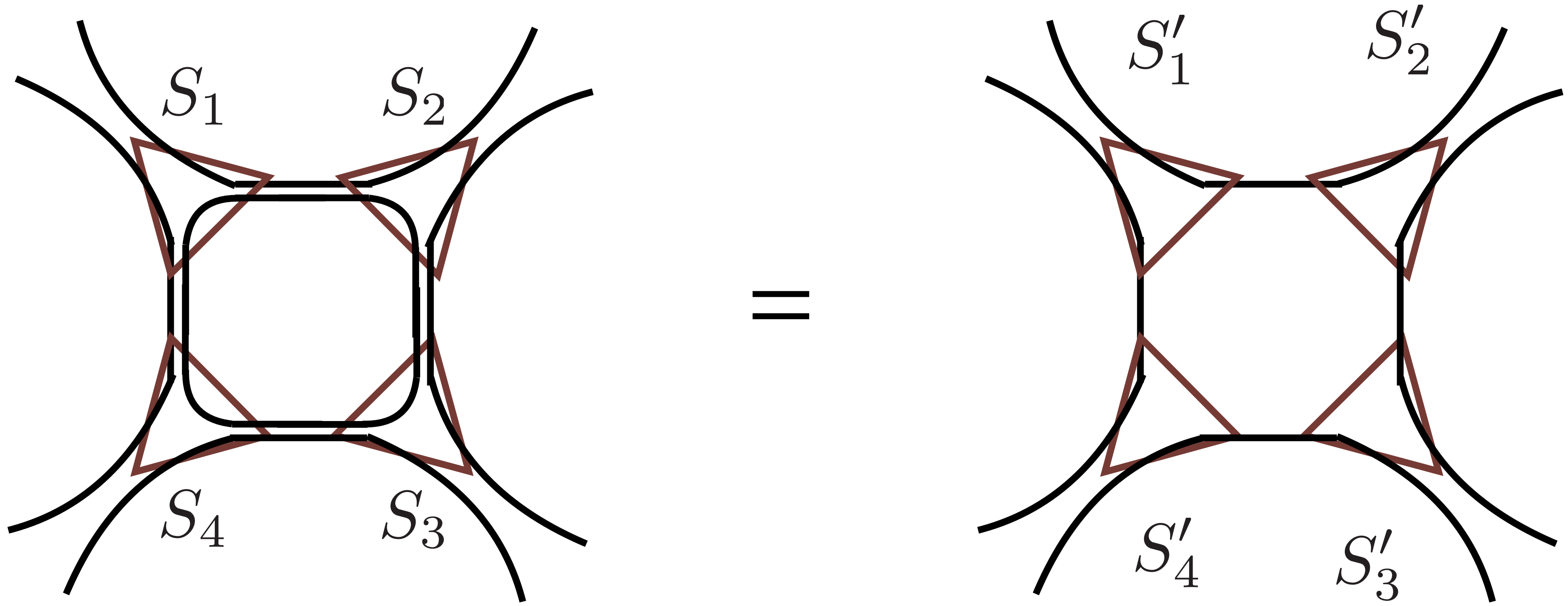
→ Increase of truncation error (decrease of accuracy)

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

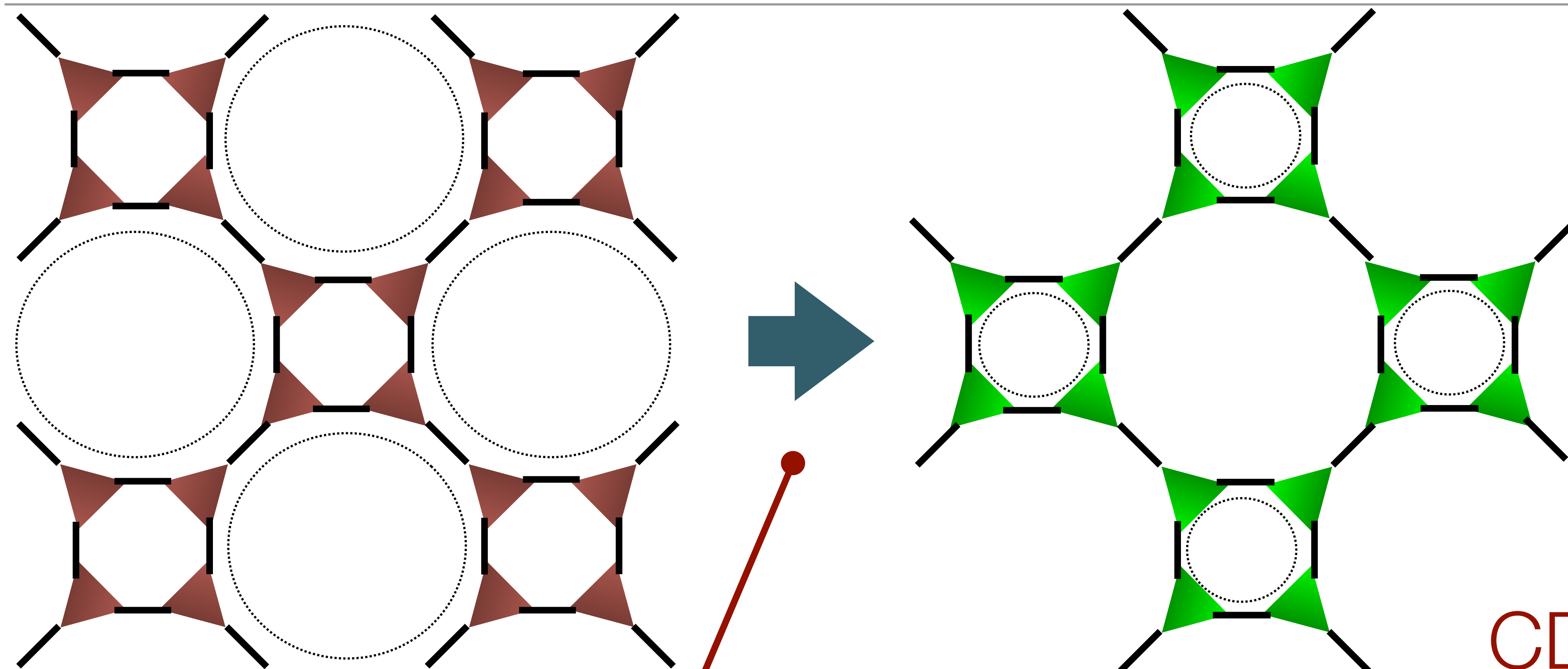
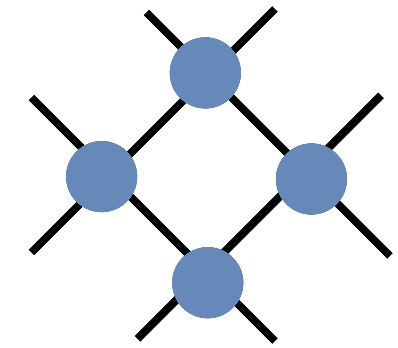
$$\mathbf{S'} = \square^{1/4}\, \mathbf{S}$$

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

Insert this "approximation" into the TRG algorithm.

Tensor Entanglement Filtering Renormalization

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

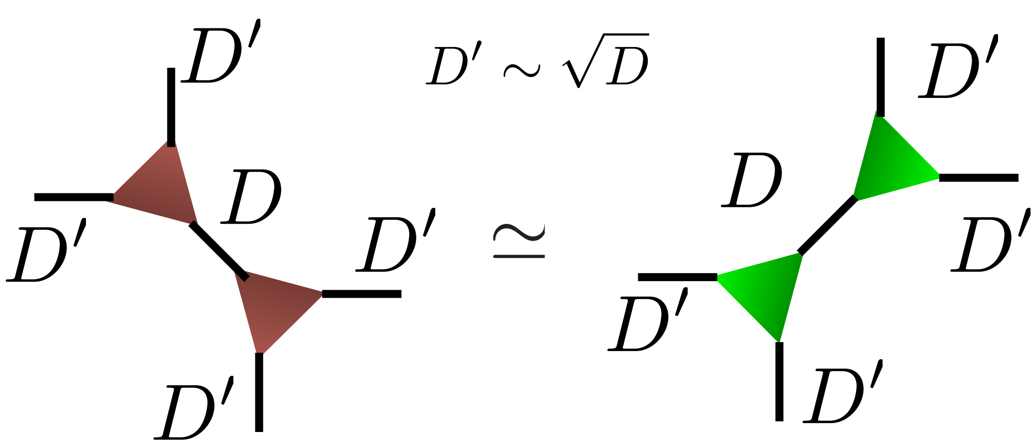
S' (after approximation)

CDL structure is perfectly removed!

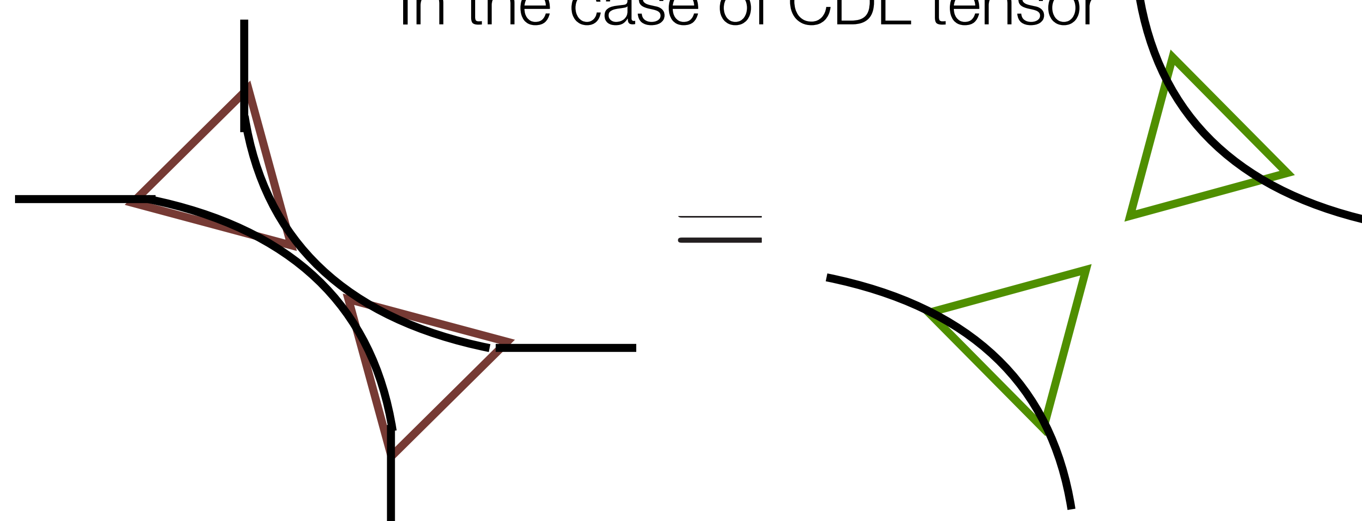
Change of SVD:

$D' < D$

$D' \sim \sqrt{D}$



In the case of CDL tensor

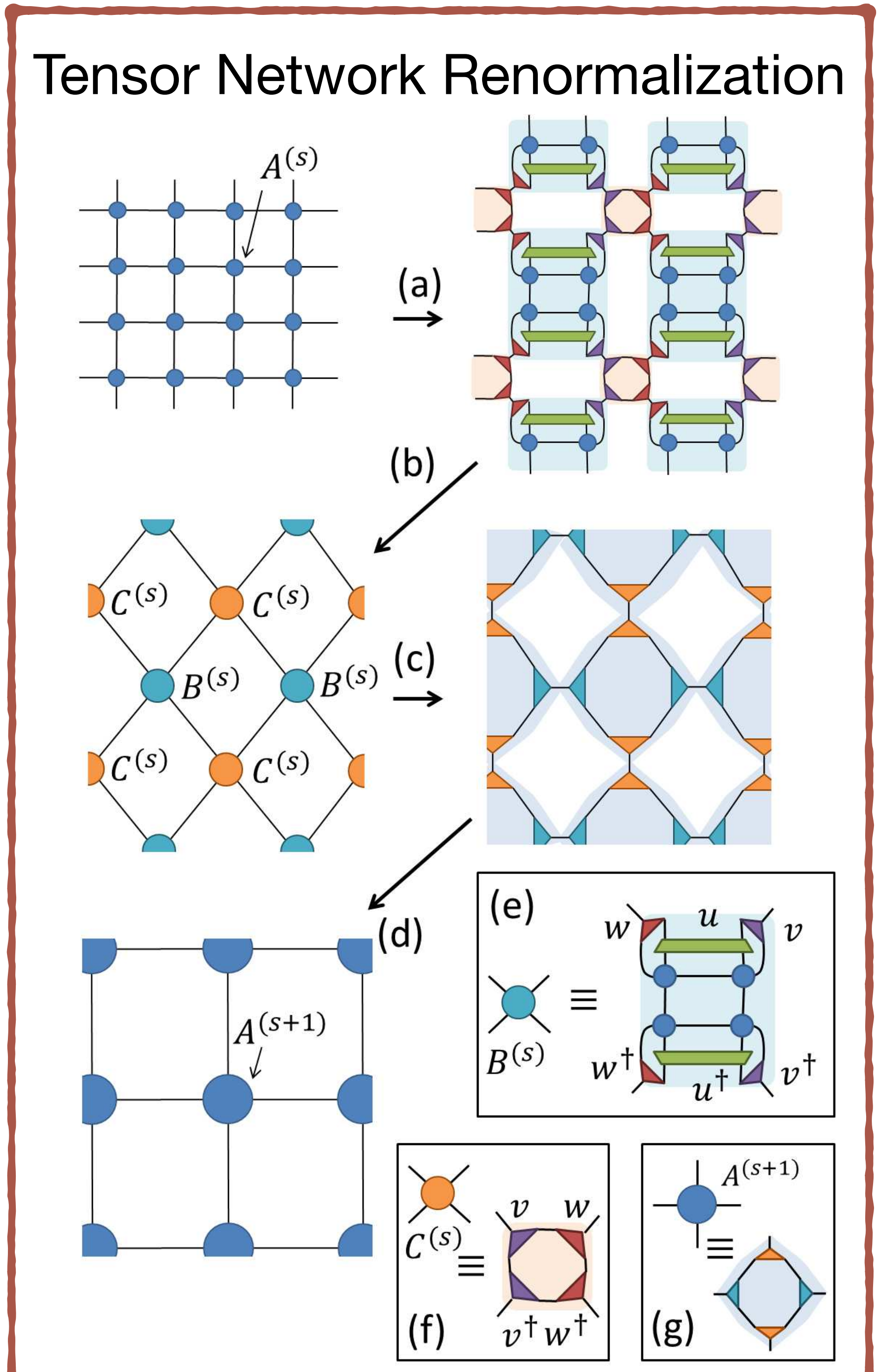


Remaining problem in TEFR

- TEFR works well 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.

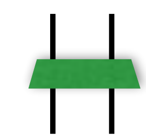
Recent progress: Tensor Network Renormalization

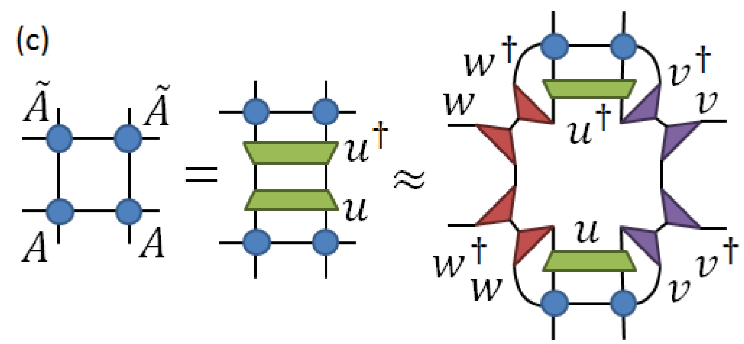
G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405(2015).
arXiv:1412.0732.



Point of TNR

Use of a disentangler(Unitary tensor)

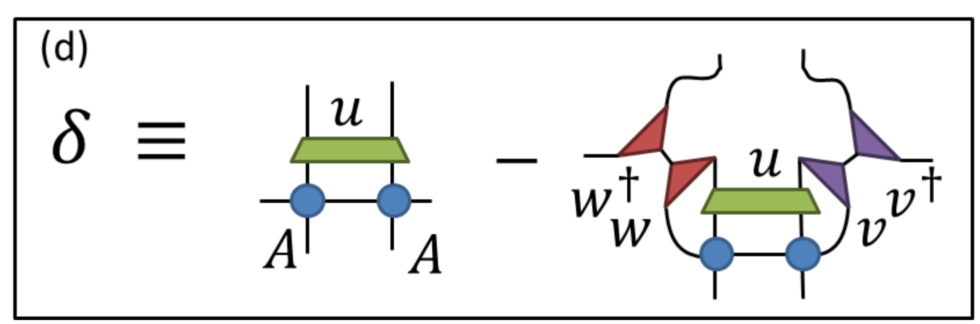




It can remove short range entanglement efficiently.

(Not only CDL)

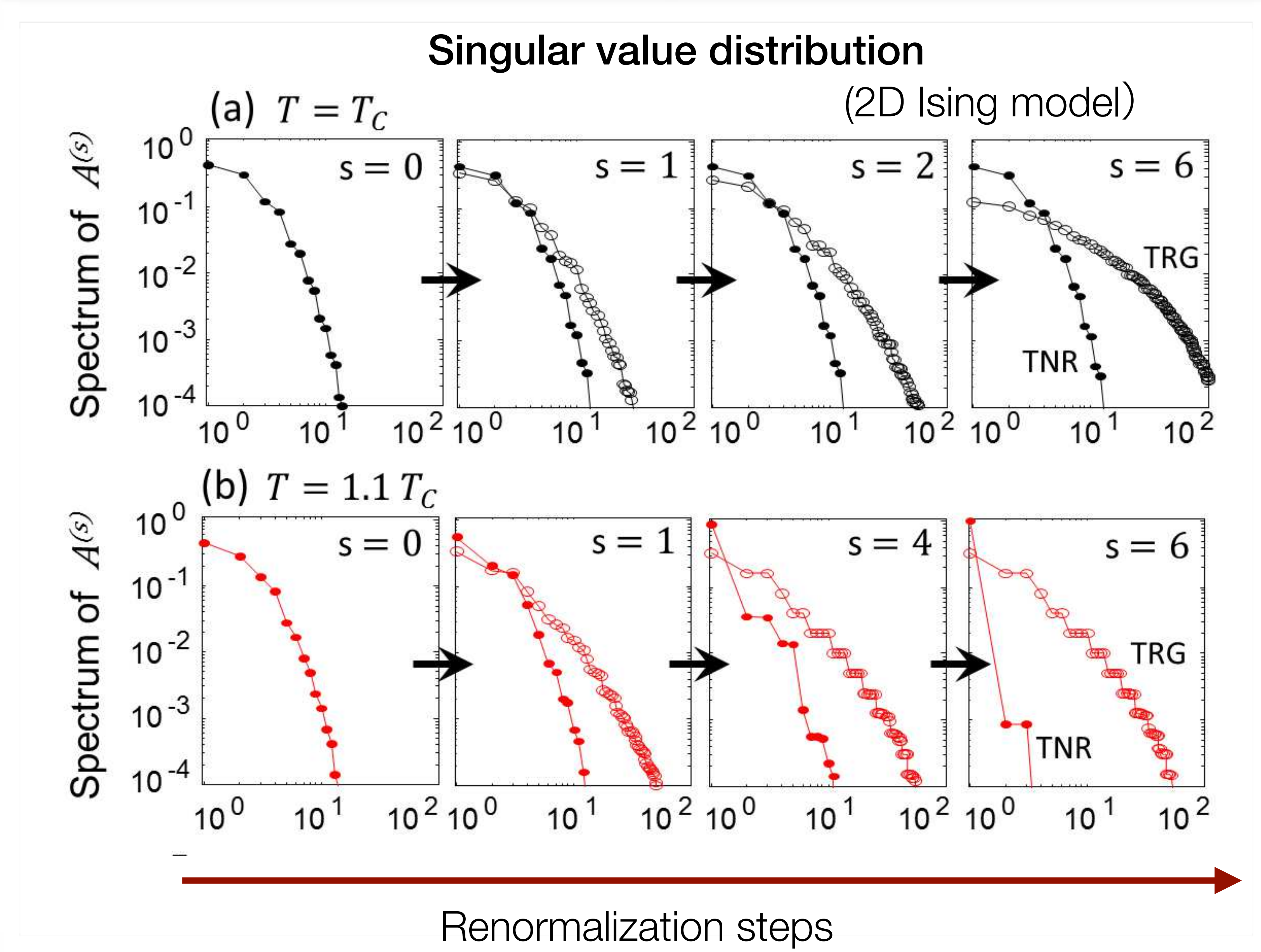
Approximation by using two-tensor cluster:

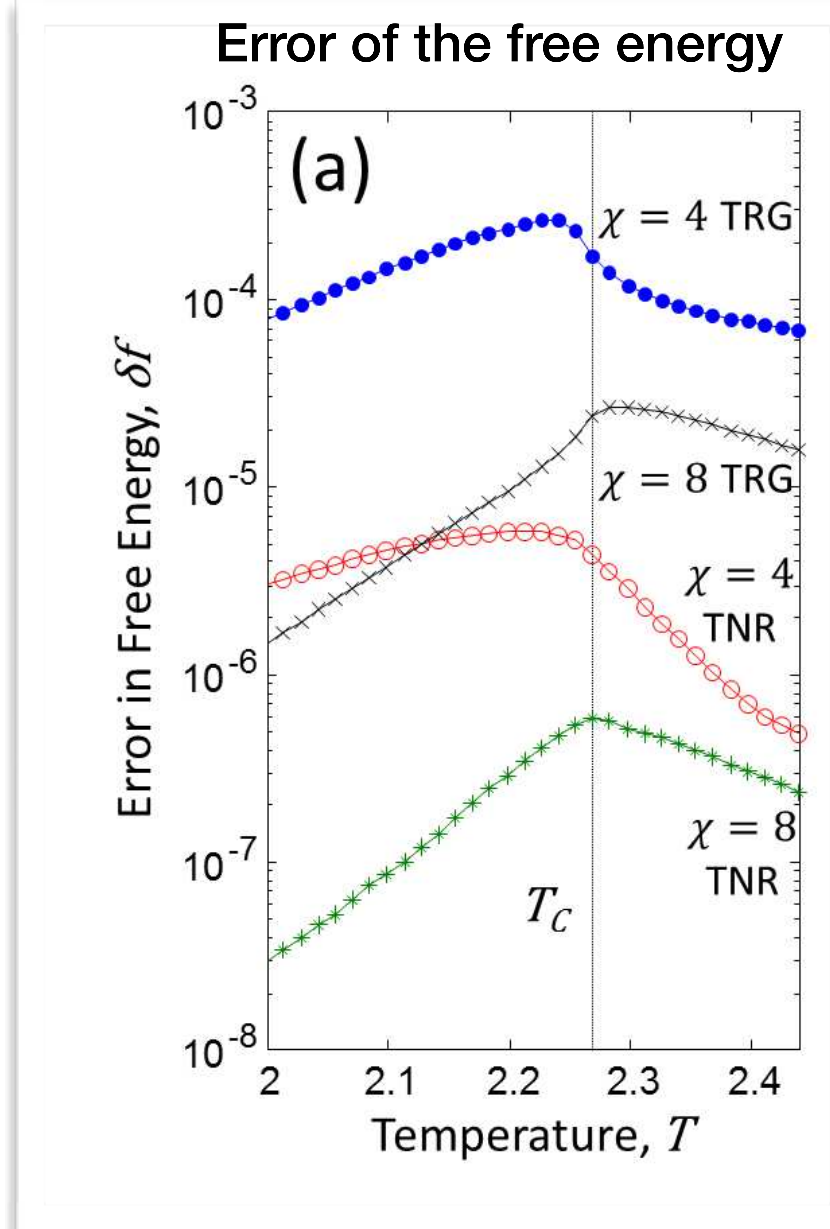


Better accuracy than the simple SVD of single tensor

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)

Report problems 3

- Pleas solve both of the problems:

  (A) Report problem (MPS)

  (B) Report problem (EE in TPS)

- 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: 2019/1/31**

(A) Report problem (MPS)

Try MPS approximation of a vector

1. Prepare a vector in m^N dimension. (It should be normalized.)
2. Make exact MPS from it.
3. By using low rank approximation based on SVD, make approximate MPS with bond dimension chi_max.
   - Note: After this approximation, the norm of vector becomes smaller.
4. Calculate distance between exact and approximate MPSs

$$||\vec{v}_{ex} - \vec{v}_{ap}||$$

5. Vary chi_max and investigate (and discuss) behavior of the distance.

6. (optional) by varying m, N, and type of vectors, discuss behavior of the distance.

(A) Report problem (MPS)

This can be done by sample python code Report_Random.py for a random vector.

Usage: python Report_Random.py -N N -m m -chi chi_max -s $seed$

output:

```
okubo$ python Report_Random.py
Parameters: N, m, chi_max = 10, 2, 20
Random seed: = None
Truncation: chi_max = 20
Distance between exact and truncated MPS = 0.22331729137806558
```

You can also consider a spin model by Report_Model.py.

Usage: python Report_Model.py -N N -m m -chi chi_max -Jz Jz -Jxy Jx -hx hx

```
okubo$ python Report_Model.py
Model parameters: Jz, Jxy, hx = -1.0, 0.0, 0.5
Parameters: N, m, chi_max = 16, 2, 2
Ground state energy per bond = -0.33360646500808594
Energy of Exact MPS = -0.33360646500808566
Truncation: chi_max = 2
Energy of MPS with truncation = -0.33270456639893253
Distance between exact and truncated MPS = 0.10640004821107511
```

(A) Report problem (MPS)

Usage: python Report_Model.py -N N -m m -chi chi_max -Jz Jz -Jxy Jxy -hx hx

$$\mathcal{H} = J_z \sum_{i=1}^{N-1} S_i^z S_{i+1}^z + J_{xy} \sum_{i=1}^{N-1} (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y) - h_x \sum_{i=1}^{N} S_i^x$$

When we set

$J_z = -1, J_{xy} = 0, h_x \neq 0$ → Transverse filed Ising model with $S = (m-1)/2$

$J_z = 1, J_{xy} = 1, h_x = 0$ → Heisenberg model with $S = (m-1)/2$

In sample codes, you can see help message by

python Report_Random -h
python Report_Model.py -h

(B) Report problem (EE in TPS)

1. Explain the upper bound of entanglement entropy of TPS (compulsory)

    1. Consider enough large vector with dimension m^N and suppose it is represented by $L \times L$ square lattice TPS with bond dimension D. ($N = L^2$, and the vector is normalized. We assume open boundary TPS.)
    2. Divide the system into two part: $l \times l$ small square at the center of the lattice and the other part. Then, represent the reduced density matrix by tensors of TPS.
    3. Represent the maximum rank of the reduced density matrix by D and l. (Please include "derivation" of the result.)

(B) Report problem (EE in TPS) (cont.)

1. Explain the upper bound of entanglement entropy of TPS (compulsory)

4. Explain the upper bound of the entanglement entropy calculated from the reduced density matrix.
   You can use the following fact:

$$-\mathrm{Tr}\ \rho \log \rho \leq \log(\mathrm{dim}\rho)$$

ρ : (reduced) density matrix

In the report, it is enough to explain the upper bound roughly.
(I don't require mathematically rigorous proof.)

It might be useful to use tensor network diagrams.

(B) Report problem (EE in TPS) (cont.)

2.Explain the upper bound of entanglement entropy of TPS in general dimension d (optional).

1. By using similar argument to the case of square lattice TPS, explain the upper bound of EE of TPS on d-dimensional cubic lattice.

Keywords

第１回：　現代物理学における巨大なデータ
第２回：　現代物理学と情報圧縮
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：　データ解析の高速化：スパース・モデリングの物質科学への応用
第９回：　データ空間の圧縮：クリロフ部分空間法の物質科学への応用
第１０回：　高度なデータ圧縮：情報のエンタングルメントと行列積表現
第１１回：　行列積表現の固有値問題への応用
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による情報圧縮

就活・インターンシップに役立つ！

プレゼンスキル&ビジネスマナー講習

*** 受講者募集***

当講習会は、プロの講師をお招きし、自身の研究について専門外の方にも平易に伝え、アピールするためのプレゼンテーションスキルおよび、社会人として必要不可欠な基本ルール・ビジネスマナー（Eメール、名刺交換、電話応対など）を講義、実習を交えながら習得します。また、特別講座として、スティーブン・G・ブランク著『アントレプレナーの教科書』を共訳、出版された渡邊哲氏をお招きし、アントレプレナーシップの基本的な理念、そよびその必要性についてご講演いただきます。

2019年 2月4日 月

開催時間

9:30～17:40

場　所

東京大学物性研究所（柏キャンパス）
*TX柏の葉キャンパス駅よりシャトルバスあり

参加対象

東京大学に所属する博士課程・修士課程の大学院生・博士研究員（PD）等で、主に物理/化学/情報科学分野の研究に携わっている方
＊ 研究科や専攻は問いません。

定　員

20名程度（申込先着順）
※定員に達し次第、申込を締め切ります。

事前準備

プレゼンテーション講習用のスライド（ppt）を準備のこと。詳細はwebを参照。

服　装

スーツまたはそれに準じる服装

持ち物

ノートPC・スマートフォン、名刺入れ

プログラム（予定）

9:30～9:40	ガイダンス
9:40～14:00	ビジネスマナー講習
14:00～16:30	プレゼンテーション講習
16:30～17:30	特別講座
17:30～17:40	総括

※途中休憩、昼食を挟みます。

担当講師

【ビジネスマナー/プレゼンテーション講習】
メイン講師：下田　令雄成　氏
株式会社シャイニング　代表取締役
サブ講師：小川　雅則　氏
株式会社シャイニング　認定プロフェッショナル

【特別講座】
企業イノベーション＆アントレプレナーシップ講座
～イノベーションをビジネスとして成功させる方法～
講師：渡邊　哲 氏
株式会社マキシマイズ　代表取締役

参加申込・詳細

下記サイトより事前申込要

申込期限：
1月29日（火）正午

http://pcoms.issp.u-tokyo.ac.jp/events/eventsfolder/skillup2019

- 期日：2/4@柏キャンパス
- 締切：1/29 （先着順）
  - 残席10名程度
- 今回はプレゼン演習に重点