

計算科学・量子計算における情報圧縮：パイロット講義

Data Compression in Computational Science and
Quantum Computing

2021.12.11

#1: Tensor network and Tensor Renormalization Group

理学系研究科 量子ソフトウェア寄付講座 大久保 毅

Graduate School of Science, **Tsuyoshi Okubo**

Lecture materials are available at

<https://github.com/utokyo-qsw/data-compression>

Quantum Software Endowed Chair
(量子ソフトウェア寄付講座)

<https://qsw.phys.s.u-tokyo.ac.jp>

Schedule of the pilot lecture

- 12/14: Tensor network and tensor renormalization group **[Okubo]**
(テンソルネットワークとテンソル繰り込み群)
- 12/21: Quantum computers and simulations **[Todo]**
(量子コンピュータ・シミュレーション)
- 1/11: Quantum error corrections and tensor network **[Okubo]**
(量子誤り訂正とテンソルネットワーク)
- 1/25: Quantum-classical hybrid algorithms and tensor network **[Todo]**
(量子古典ハイブリッドアルゴリズムとテンソルネットワーク)

Outline

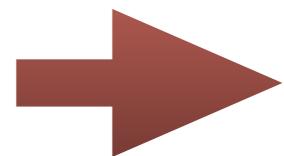
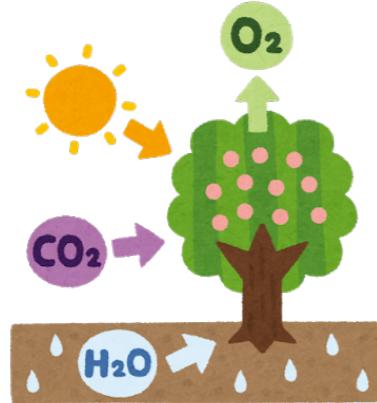
- Motivation: Quantum many-body problems
- Data compression and Tensor Network
 - SVD and its generalization to tensors
 - Matrix product states (MPS)
 - Application: approximated "gate" operation by MPS
- Contraction of tensor networks
 - Tensor network representation of a scalar
 - Tensor network renormalization

Motivations: Quantum many-body problems

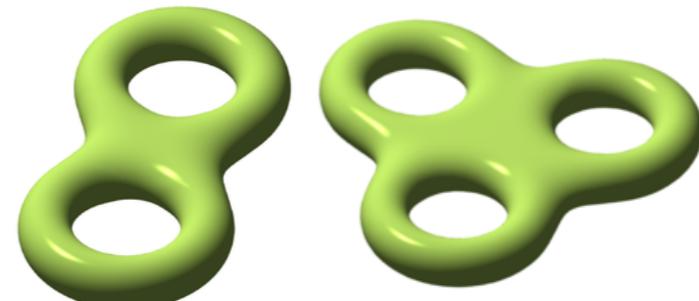
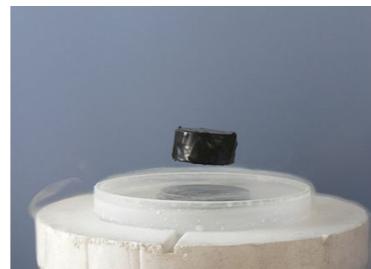
Motivation: Quantum many-body problems

A variety of phenomena

- Chemical reaction
- Superconductivity
- Topological states
- ...



Quantum many-body problems



Cited from wikipedia: "Meisner effect", "Torus"

Schrödinger equation

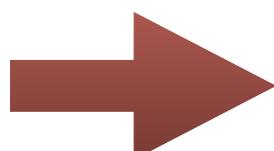
$$i\hbar \frac{\partial}{\partial t} |\Psi\rangle = \mathcal{H} |\Psi\rangle$$

\mathcal{H} :Hamiltonian

$|\Psi\rangle$:Wave function (state vector)

$$\mathcal{H} |\Psi\rangle = E |\Psi\rangle$$

E :Energy



To solve the problem numerically by (classical) computer,
we need **huge memory** and **huge computation time**.

Quantum systems

Example of quantum system: Array of quantum bits

1 bit

- A quantum bit is represented by two basis vectors.

$$|0\rangle, |1\rangle \text{ or } (|\uparrow\rangle, |\downarrow\rangle)$$

2 bits



The vector space is spanned by four basis vectors.

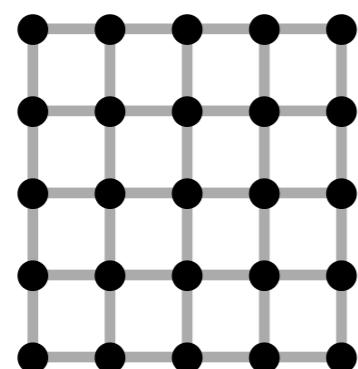
$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

$$\text{Simple notation: } |00\rangle, |01\rangle, |10\rangle, |11\rangle$$

$$\rightarrow |\Psi\rangle = \sum_{\alpha, \beta=0,1} C_{\alpha, \beta} |\alpha\beta\rangle$$

$C_{\alpha, \beta}$:complex number

N bits



Dimension of the vector space $= 2^N$

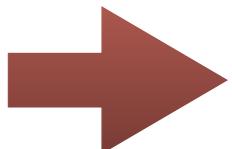
Exponentially large!

$$|\Psi\rangle = \sum_{\{i_1, i_2, \dots, i_N\}} \Psi_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle$$

Difficulty in quantum many-body problems

Schrödinger equation: $i\hbar \frac{\partial}{\partial t} |\Psi\rangle = \mathcal{H}|\Psi\rangle$

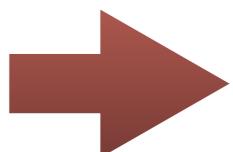
- Dimension of the vector space is **exponentially large**



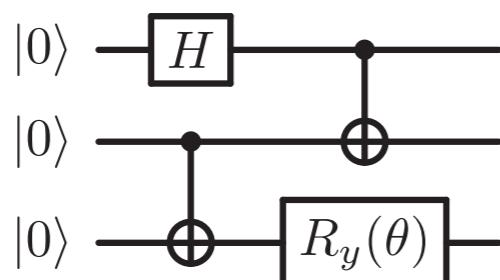
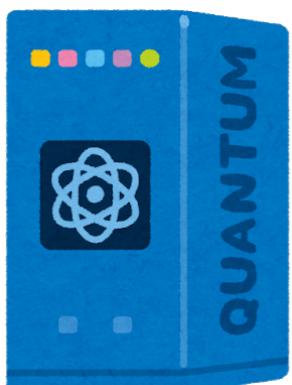
To solve the problem **exactly** by (classical) computer, we need **huge memory** and **huge computation time**.

e.g., We can simulate only ~50 qubits in classical supercomputer.

Quantum computer



It can treat **quantum state directly**, and then (ideally) there is no problems originated from the exponentially large vector space.



Classical computer?

- There are several techniques to treat quantum many body problems.
 - One of them is **a data compression based on tensor networks**.
- We may use them to simulate quantum computers.

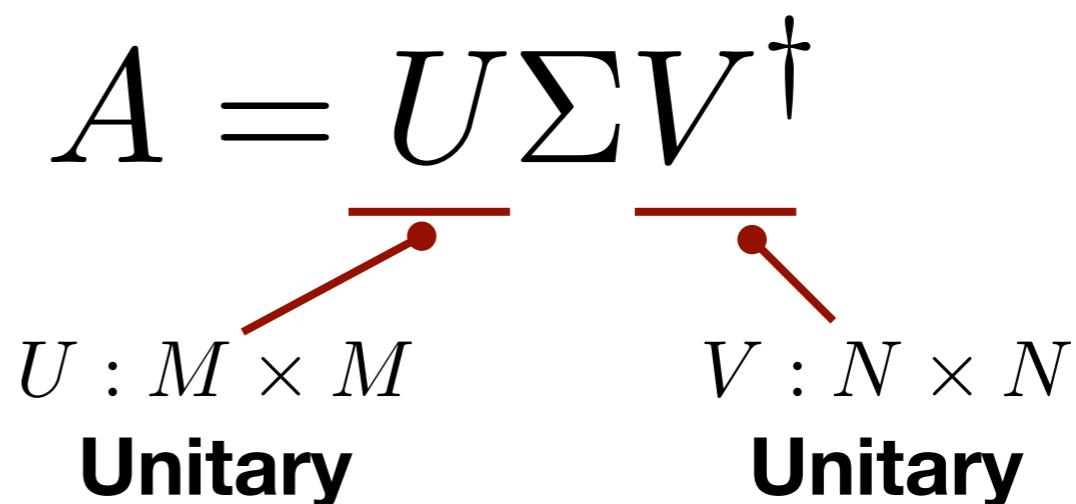
Data compression and tensor networks:
SVD and its generalization to tensors

Singular value decomposition (SVD)

Singular value decomposition (特異値分解)

$$A : M \times N$$

$$A_{ij} \in \mathbb{C}$$

$$A = U \Sigma V^\dagger$$


$U : M \times M$ $V : N \times N$

Unitary **Unitary**

$$\Sigma = \begin{pmatrix} \Sigma_{r \times r} & 0_{r \times (N-r)} \\ 0_{(M-r) \times r} & 0_{(M-r) \times N-r} \end{pmatrix}$$

$$\Sigma_{r \times r} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix}$$

Diagonal matrix with
non-negative real elements

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

Singular values

Amount of data in SVD representation

$$A : M \times N$$

$$A = U\Sigma V^\dagger = U \begin{pmatrix} \Sigma_{r \times r} & 0_{r \times (N-r)} \\ 0_{(M-r) \times r} & 0_{(M-r) \times N-r} \end{pmatrix} V^\dagger$$

**neglect zero
singular values**

$$\rightarrow = \bar{U} \Sigma_{r \times r} \bar{V}^\dagger$$

$$\bar{U} : M \times r, \bar{V}^\dagger : r \times N$$

If $\text{rank}(A)$ is much smaller than M and N ,

$$r \ll M, N$$

we can reduce the data to represent A .

(At this stage, no data loss)

$$\boxed{\begin{aligned} U &= (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_M) \\ V &= (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) \end{aligned}}$$



$$\boxed{\begin{aligned} \bar{U} &= (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r) \\ \bar{V} &= (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r) \end{aligned}}$$

Low rank approximation by SVD

Consider a matrix obtained by **neglecting smaller singular values**

$$A = \bar{U} \Sigma_{r \times r} \bar{V}^\dagger \rightarrow \tilde{A} = \tilde{U} \Sigma_{k \times k} \tilde{V}^\dagger \quad (k < r)$$

$$\begin{aligned}\Sigma_{r \times r} &= \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \\ \bar{U} &= (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r) \\ \bar{V} &= (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r)\end{aligned}$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$$

$$\text{rank}(A) = r$$

$$\begin{aligned}\Sigma_{k \times k} &= \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k) \\ \tilde{U} &= (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k) \\ \tilde{V} &= (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k)\end{aligned}$$

Keep **the largest k singular values** (and corresponding singular vectors).

$$\text{rank}(\tilde{A}) = k < r$$

This approximation is one of optimal low rank approximations of a matrix.

It minimizes distance defined by, e.g., Frobenius norm.

Approximation of tensors?

Quantum state: $|\Psi\rangle = \sum_{\{i_1, i_2, \dots, i_N\}} \underline{\Psi_{i_1 i_2 \dots i_N}} |i_1 i_2 \dots i_N\rangle$

The coefficient is a tensor in $\mathbb{C}^M = \mathbb{C}^a \otimes \mathbb{C}^a \otimes \dots \otimes \mathbb{C}^a$

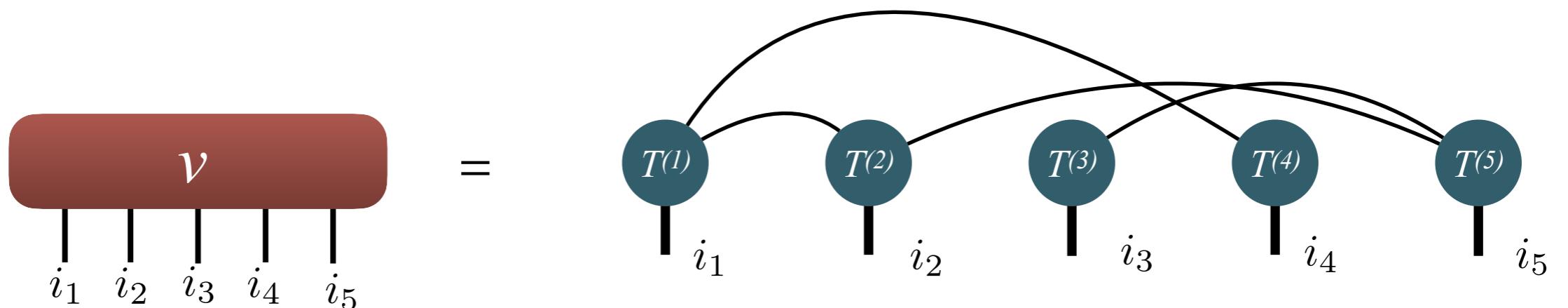
*Exponentially large Hilbert space: $M = a^N$

→ Tensor network decomposition

$$v_i = v_{i_1, i_2, \dots, i_N} = \sum_{\{x\}} T^{(1)}[i_1]_{x_1, x_2, \dots} T^{(2)}[i_2]_{x_1, x_3, \dots} \dots T^{(N)}[i_N]_{x_3, x_{100}, \dots}$$

$i_n = 0, 1, \dots, a - 1$: index of local Hilbert space

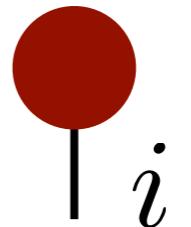
$T[i]_{x_1, x_2, \dots}$: local tensor for "state" i



Graphical representations for tensor network

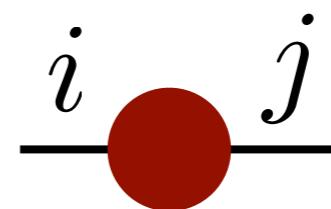
- Vector

$$\vec{v} : v_i$$



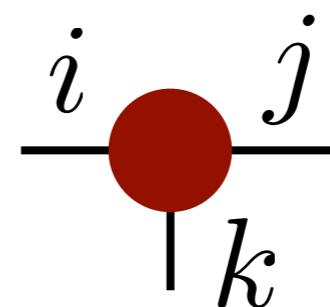
- Matrix

$$M : M_{i,j}$$



- Tensor

$$T : T_{i,j,k}$$



* **n-rank tensor = n-leg object**

When indices are not presented in a graph, it represent a tensor itself.

$$\vec{v} = \text{---} \bullet$$

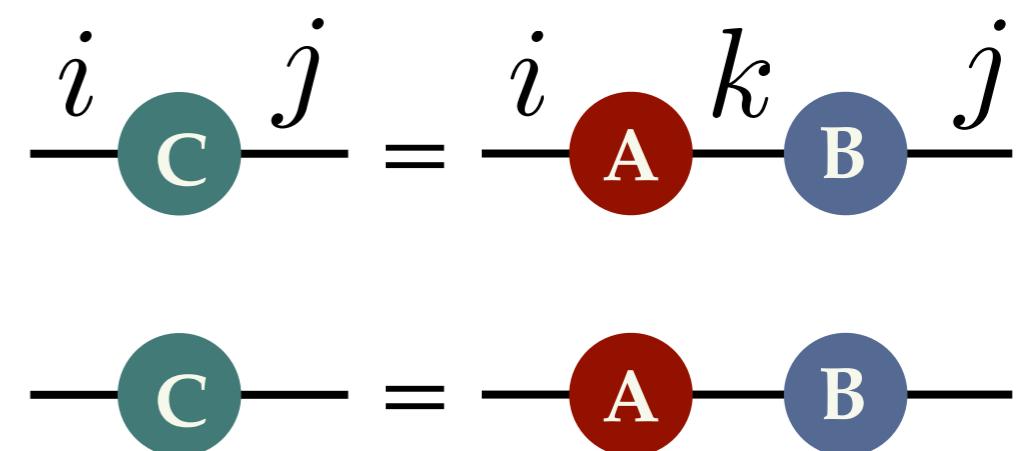
$$T = \text{---} \bullet \text{---}$$

Graphical representations for tensor network

Matrix product

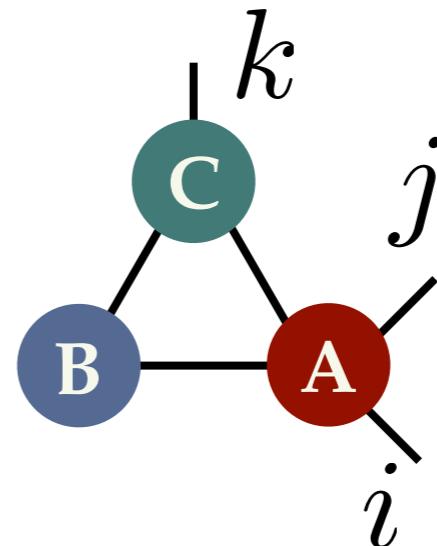
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$

$$C = AB$$



Generalization to tensors

$$\sum_{\alpha, \beta, \gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$

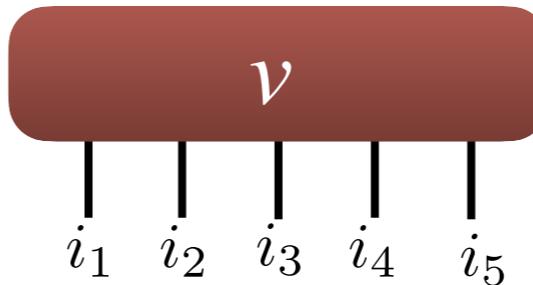


Contraction of a network = Calculation of a lot of multiplications
(縮約)

Diagram for a tensor network decomposition

- Vector

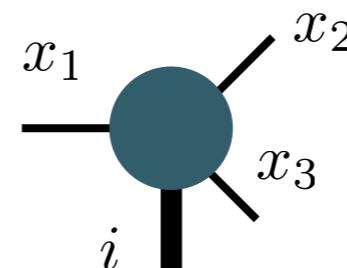
$$v_{i_1, i_2, i_3, i_4, i_5}$$



*Vector looks like a tensor

- Tensor

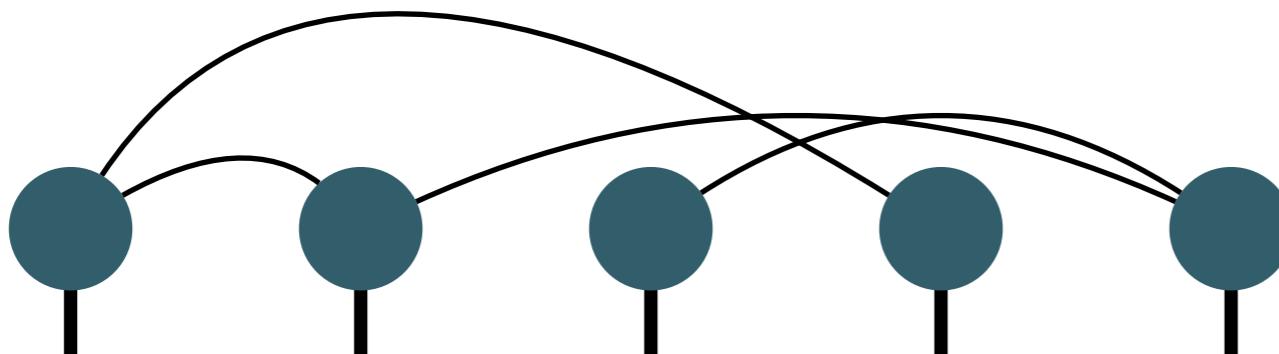
$$T[i]_{x_1, x_2, x_3}$$



*We treat i as an index
of the tensor.

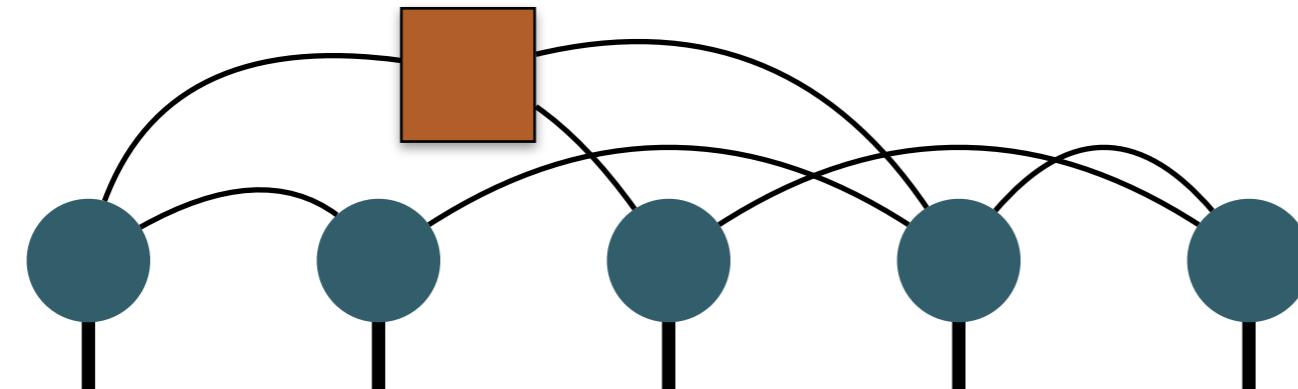
Tensor network decomposition

$$\vec{v} =$$



*We can consider tensors
independent on i .

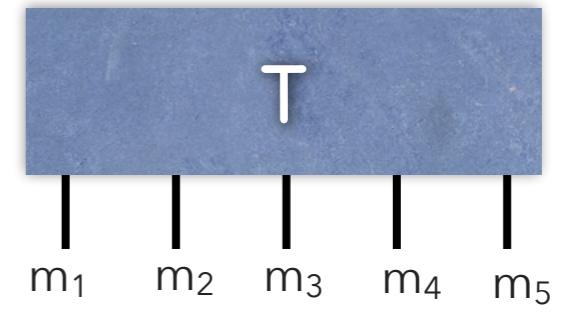
$$\vec{w} =$$



"generalization" of SVD to tensors.

T_{m_1, m_2, \dots, m_N} : N-leg tensor (or Vector)

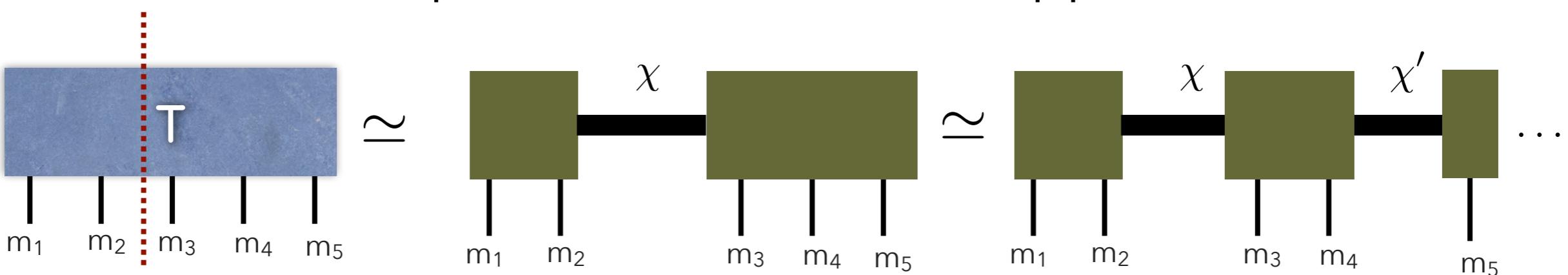
Cf. wave function: $|\Psi\rangle = \sum_{\{m_i=0,1\}} T_{m_1, m_2, \dots, m_N} |m_1, m_2, \dots, m_N\rangle$



We can consider it as a matrix by making two groups:

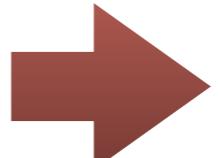
$T_{\{m_1, m_2, \dots, m_M\}, \{m_{M+1}, \dots, m_N\}}$

→ We can perform the low rank approximation of T .



*obtained two objects
are again tensors.

What does it mean?



It is related to MPS

Matrix product states (MPS)

Good reviews:

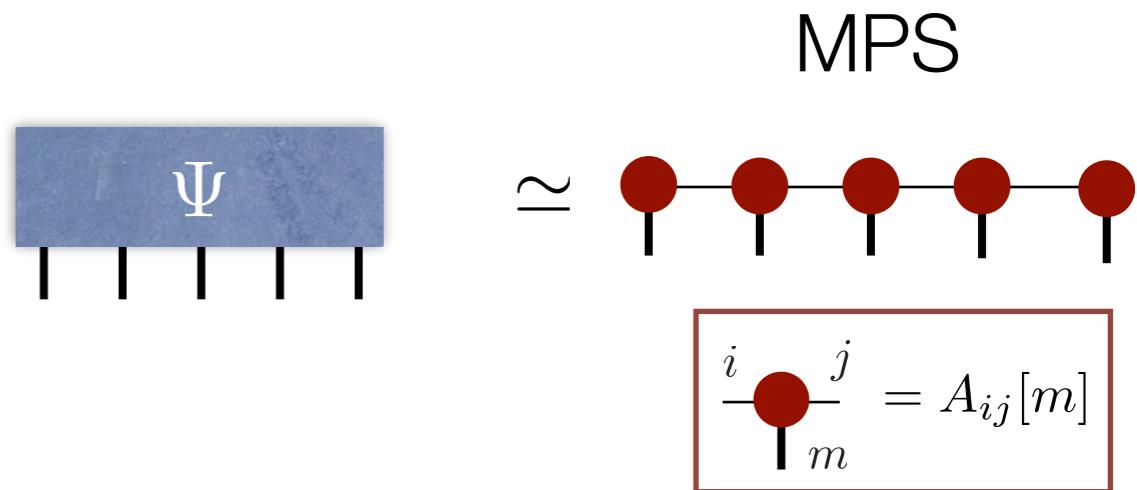
Matrix product state (MPS)

(U. Schollwöck, Annals. of Physics **326**, 96 (2011))
(R. Orús, Annals. of Physics **349**, 117 (2014))

$$|\Psi\rangle = \sum_{\{i_1, i_2, \dots, i_N\}} \Psi_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle$$

$$\Psi_{i_1 i_2 \dots i_N} \simeq A_1[i_1] A_2[i_2] \cdots A_N[i_N]$$

$A[i]$: Matrix for state i



Note:

- MPS is called "tensor train decomposition" in applied mathematics
(I. V. Oseledets, SIAM J. Sci. Comput. **33**, 2295 (2011))
- A product state is represented by MPS with 1×1 "Matrix" (scalar)

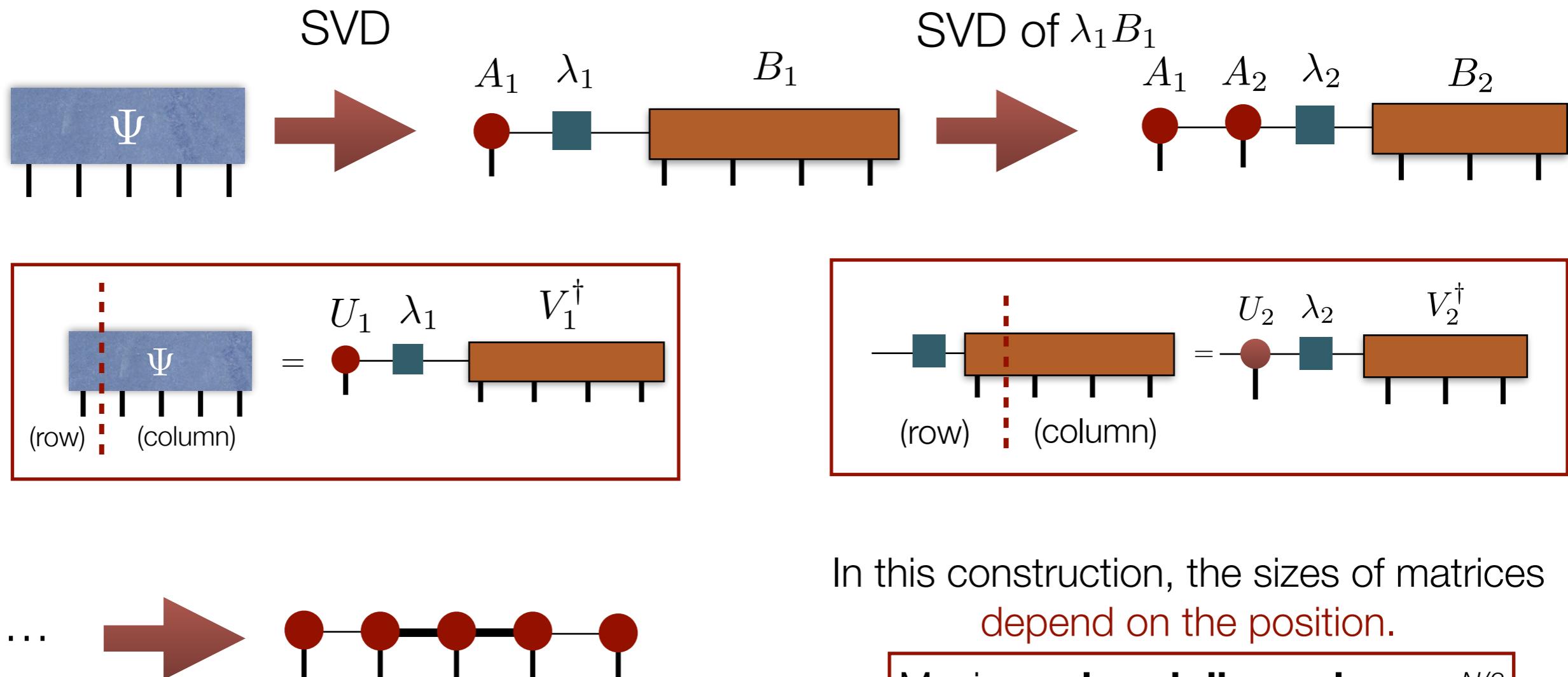
$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

$$\Psi_{i_1 i_2 \dots i_N} = \phi_1[i_1] \phi_2[i_2] \cdots \phi_N[i_N]$$

$$\phi_n[i] \equiv \langle i | \phi_i \rangle$$

Matrix product state without approximation

General vectors can be represented by MPS exactly through successive Schmidt decompositions

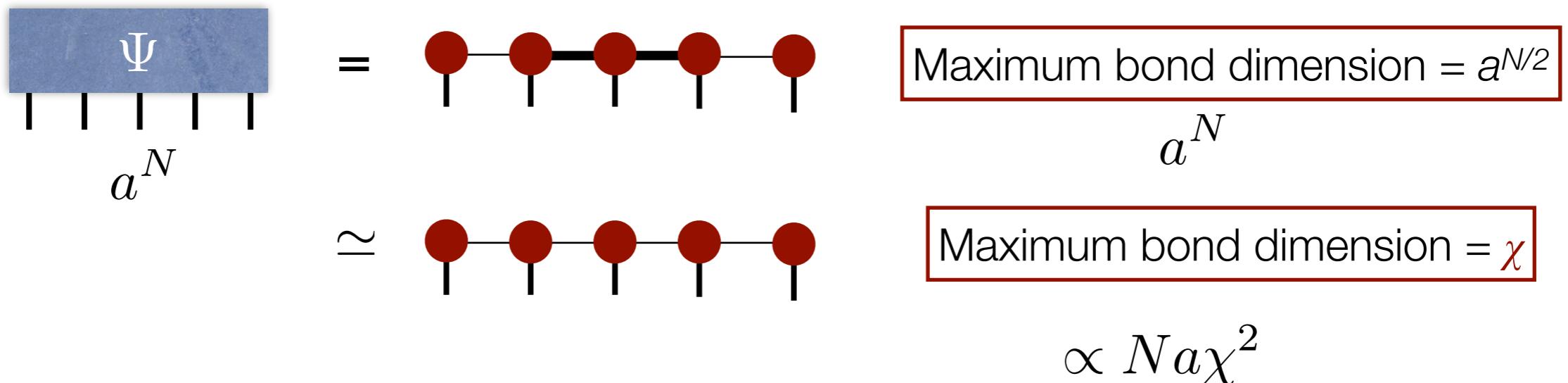


In this construction, the sizes of matrices depend on the position.

Maximum **bond dimension** = $a^{N/2}$

At this stage, **no data compression**.

Matrix product state: Low rank approximation



If we can approximate the original tensor accurately by MPS with bond dimension χ , we obtain data compression from exponential to polynomial.

- Here we assume, χ is independent of N .
 - This is true, e.g., one-dimensional gapped quantum system.
 - For general cases, we can consider the entanglement entropy to check the property.
- Even if χ increases as we increase N , we can use MPS to approximate a tensor in practice.

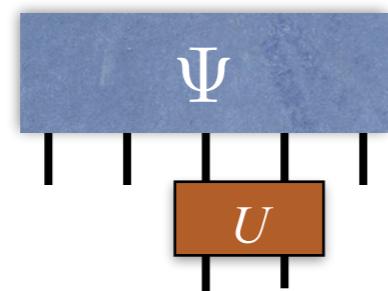
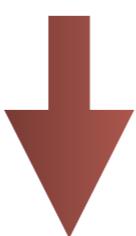
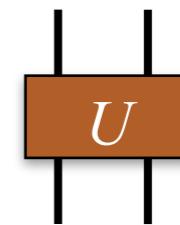
Application: approximated "gate" operation by MPS

Gate operation to a quantum state

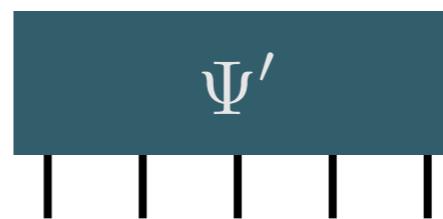
Quantum state:



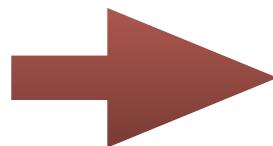
+ local two-body operation:



=



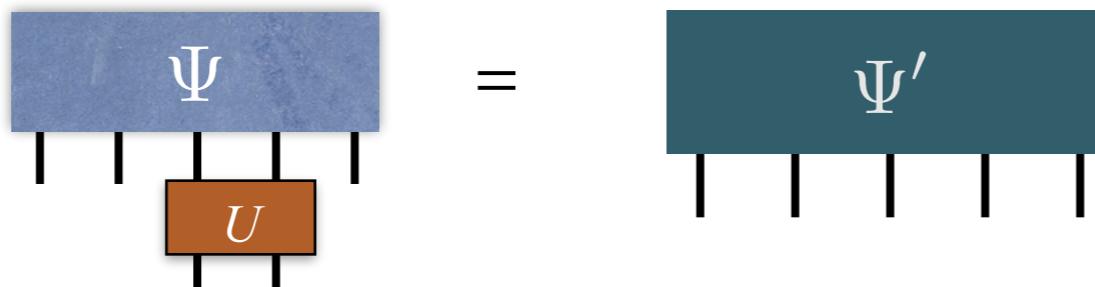
If # of "qubits" increases, a (classical) simulation of this operation needs **exponentially** large resources.



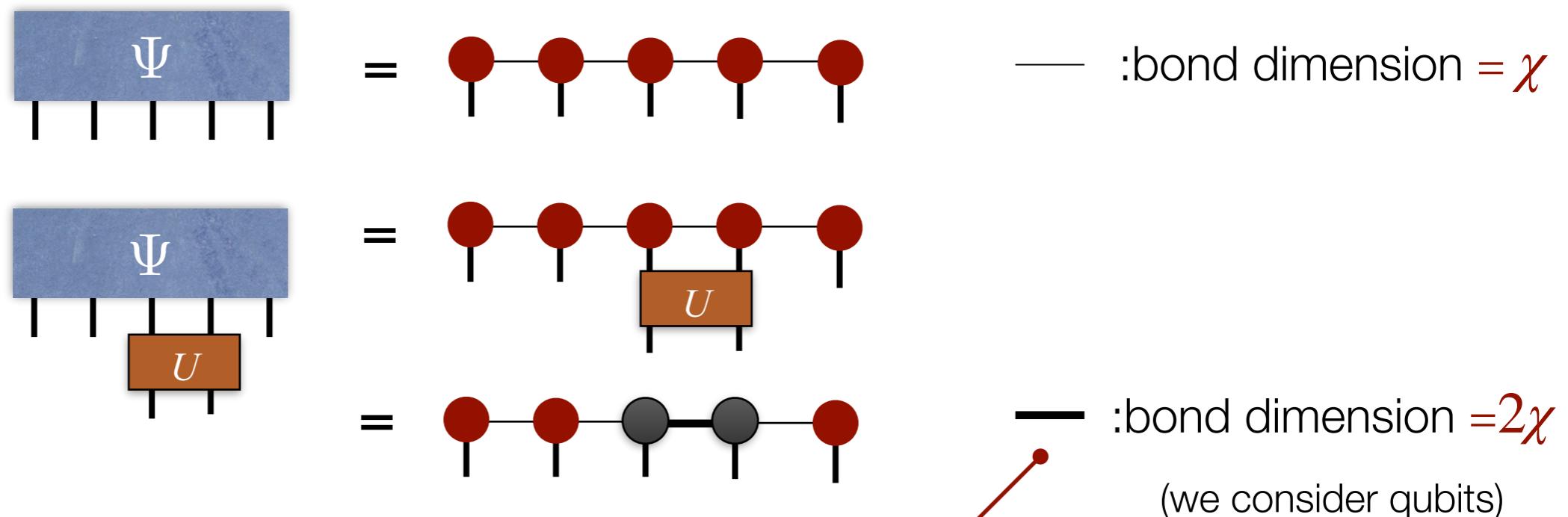
Approximated simulation by MPS?

(We can simulate this by polynomial cost.)

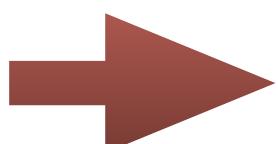
Gate operation in MPS representation



MPS representation:



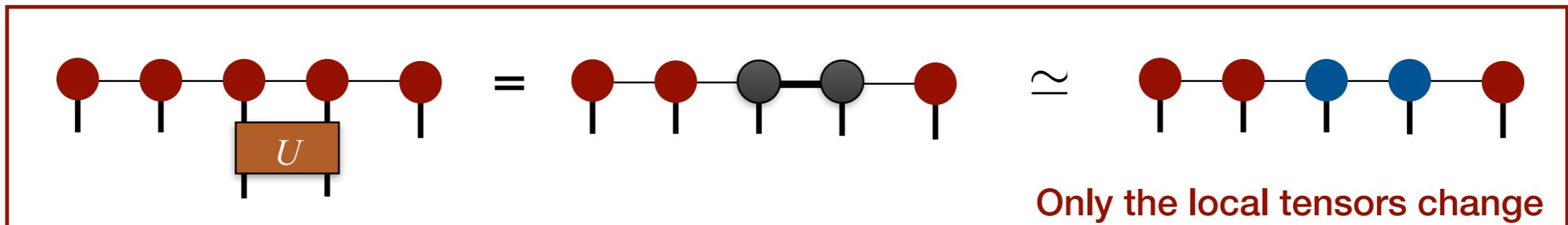
* Even if the initial state is an MPS, the bond dimension of the new state can be increased due to the gate operation.



If we want to repeat gate operations, we need to reduce the bond dimension **by an approximation**.

Approximated gate operation in MPS

Consider an approximation by



- This approximation may not be the best approximation.
 - Generally, local gate operation can affect all tensors in MPS.
 - However, if U is a unitary operator, this approximation is almost optimal.

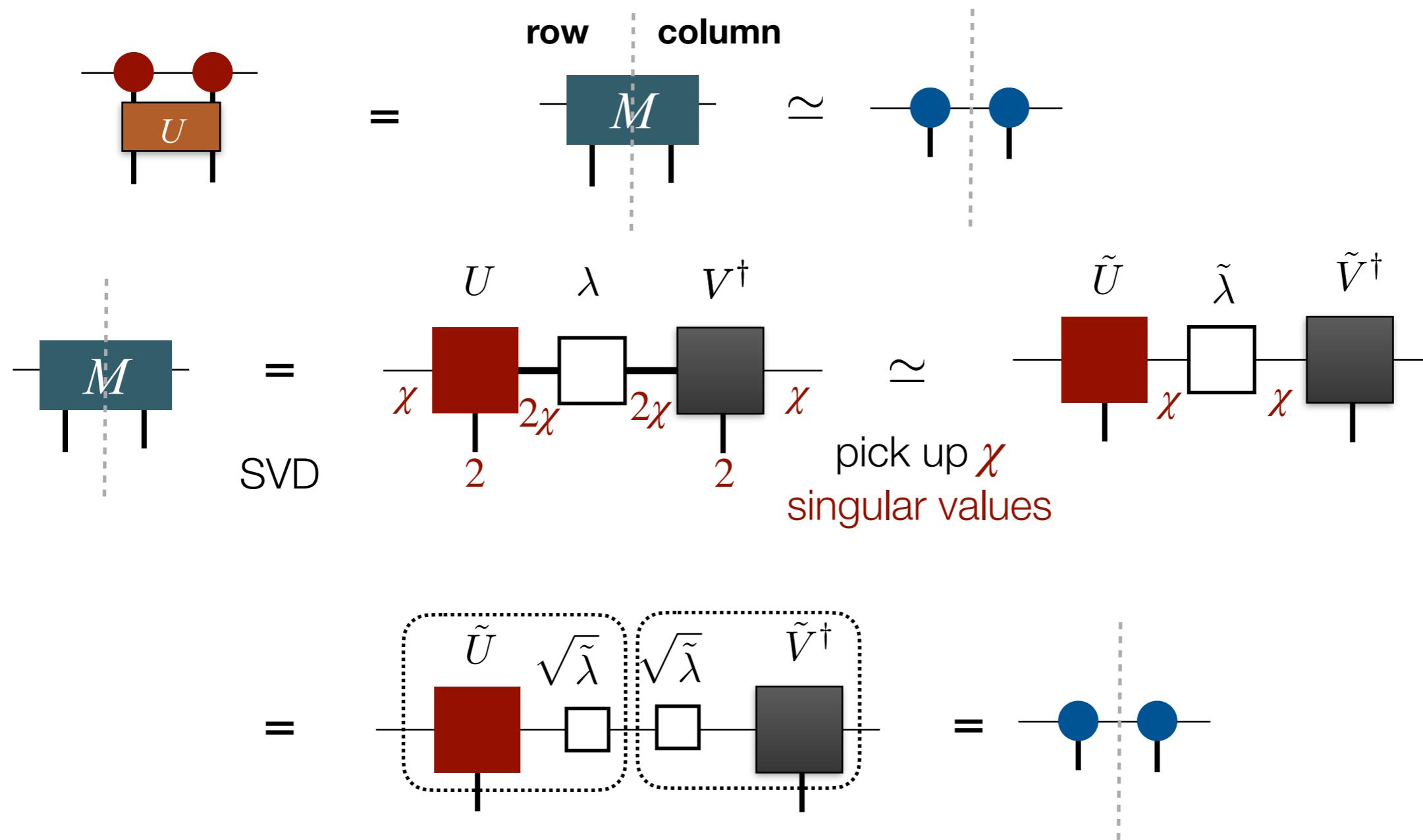
(Rough) approximation neglecting "environment".



Optimal approximation is achieved by SVD.

Approximation by SVD

Approximation by SVD.



Remarks on approximation by MPS

Note that this approximation consider **only** the local structure.

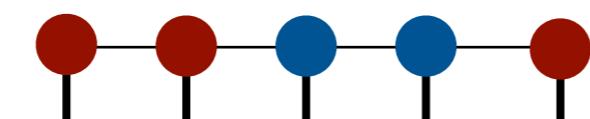
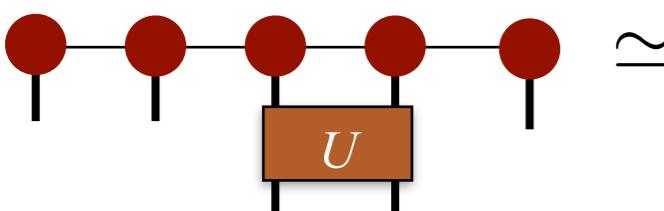
→ In general, the approximation is **not necessarily the best global approximation.**

To include the effect of whole structure, we can use, e.g.,
TEBD algorithm.

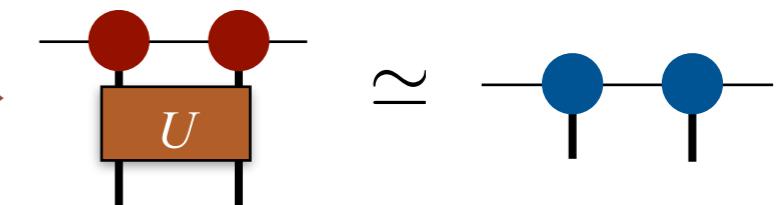
Time evolving block decimation (G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))

Keywords: Schmidt coefficient, canonical form, entanglement, ...

Global approximation



Local approximation

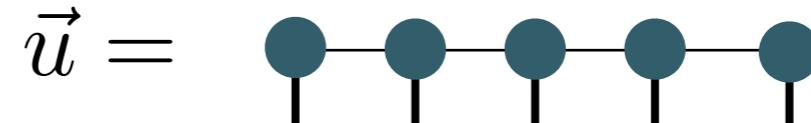
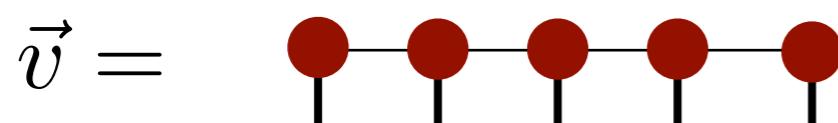


Contraction of tensor networks:
Tensor network representation of **a scalar**

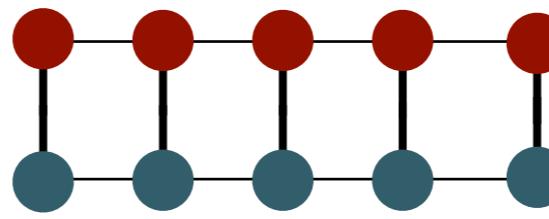
Tensor network representation of a scalar

Example: inner product of two TNSs

MPS



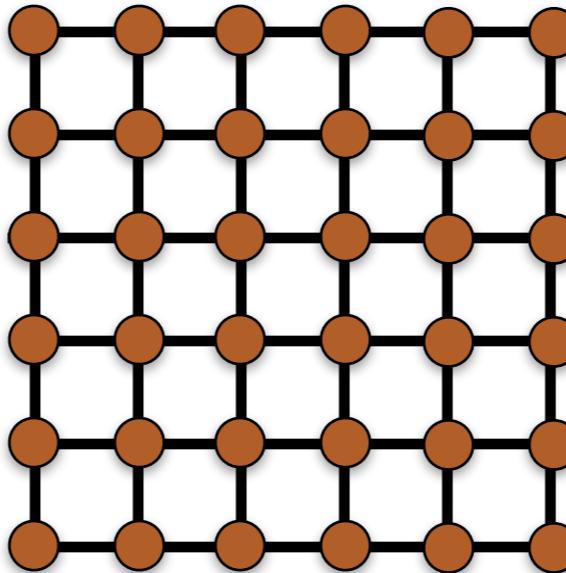
$$\vec{v} \cdot \vec{u}^* =$$



$$= \text{---|---|---|---|---}$$

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

$\beta = \frac{1}{k_B T}$: 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

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

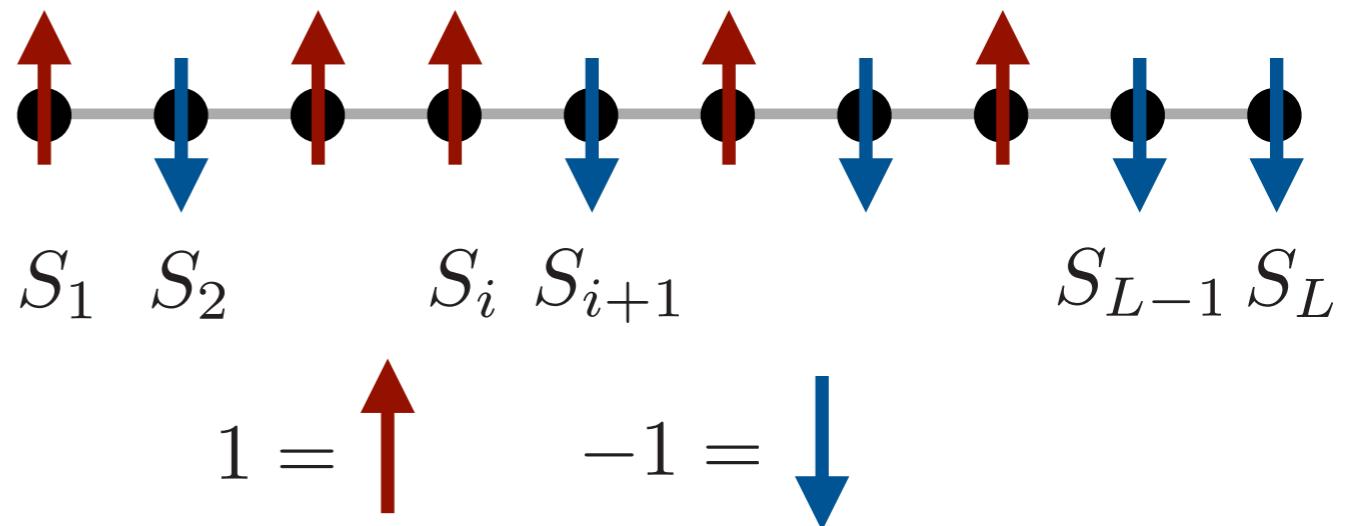
log of the partition function = Free energy

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:

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

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

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

Transfer matrix
(転送行列)

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

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

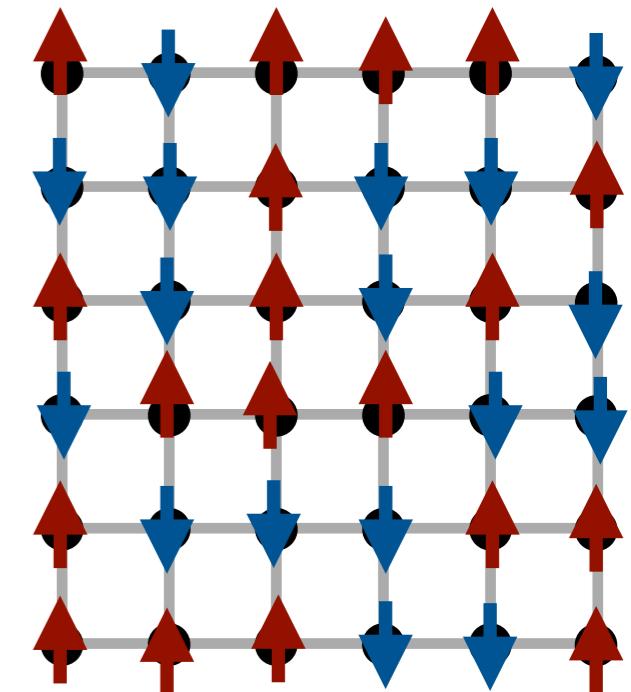
$$\sum_{S_1=\pm 1, S_L=\pm 1} S_1 \quad \text{---} \quad S_L$$

Tensor network representation in two dimension

Classical Ising model on the square lattice

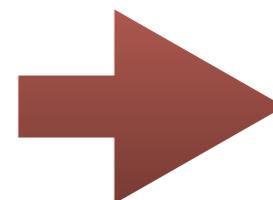
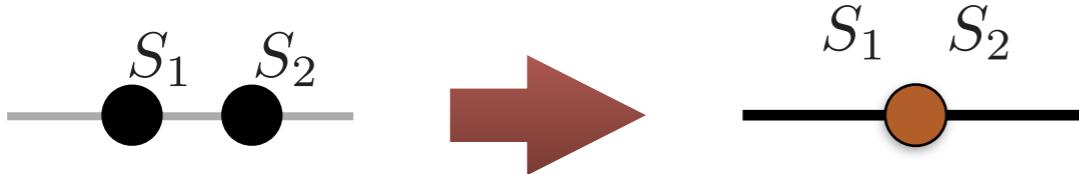
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j \quad (S_i = \pm 1 = \uparrow, \downarrow)$$

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



We can use a tensor instead of the transfer matrix.

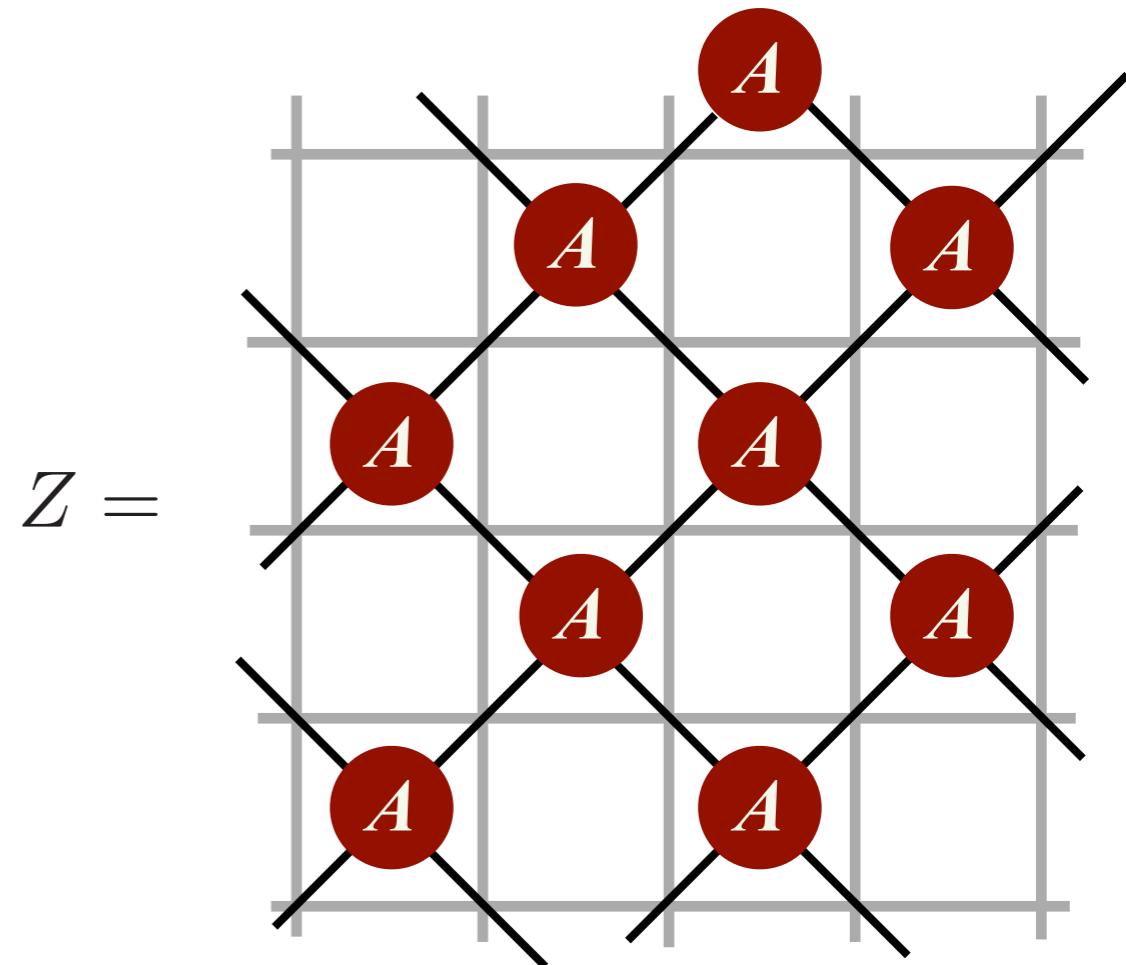
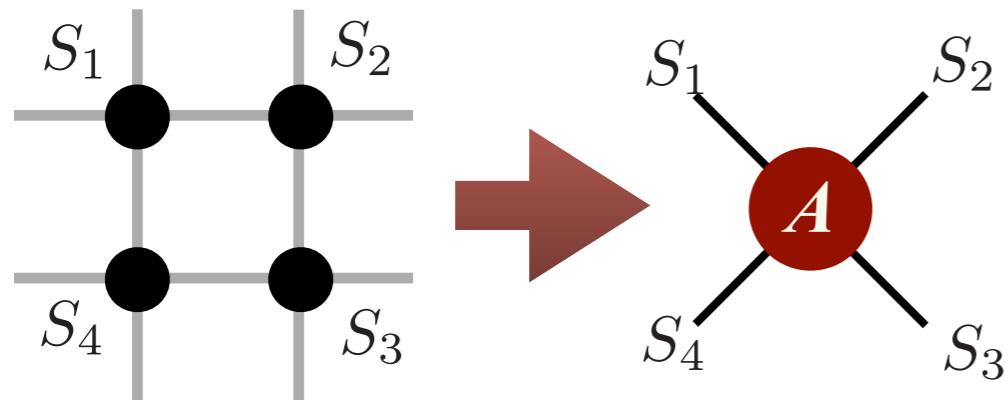
$$e^{\beta J S_1 S_2} = T_{S_1 S_2}$$



Tensor?

Tensor network representation in two dimension

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



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.

Cost of the contraction of a tensor network

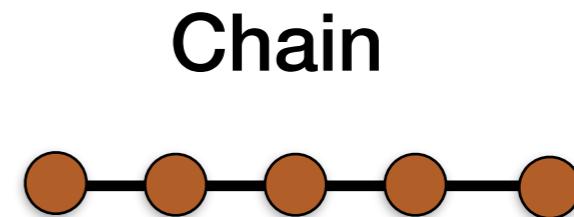
Cost of tensor network contraction:

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

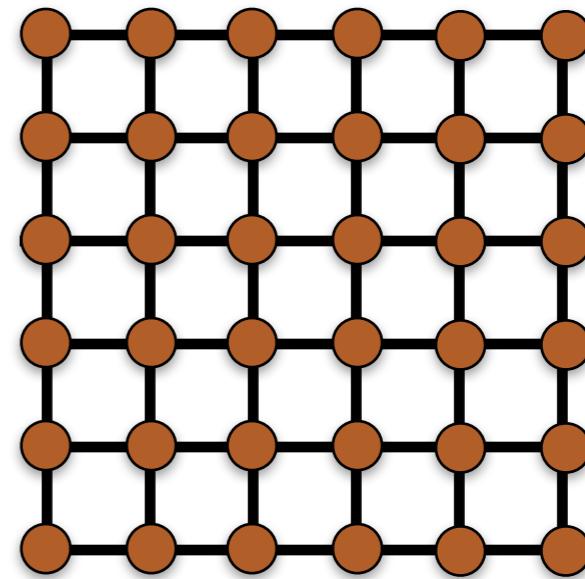
Chain: $O(ND^2)$ (Open)
 $O(ND^3)$ (Periodic)

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

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



Square lattice



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



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

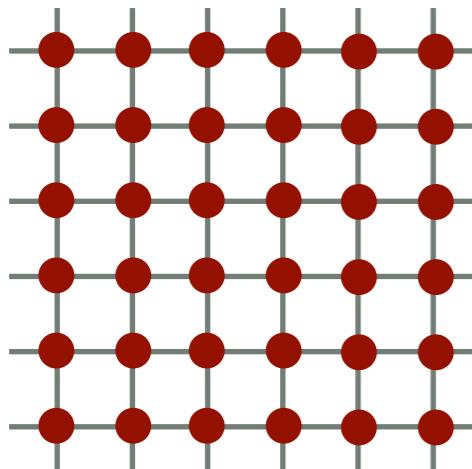
Tensor renormalization group

Tensor renormalization group (テンソル繰り込み群)

- 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

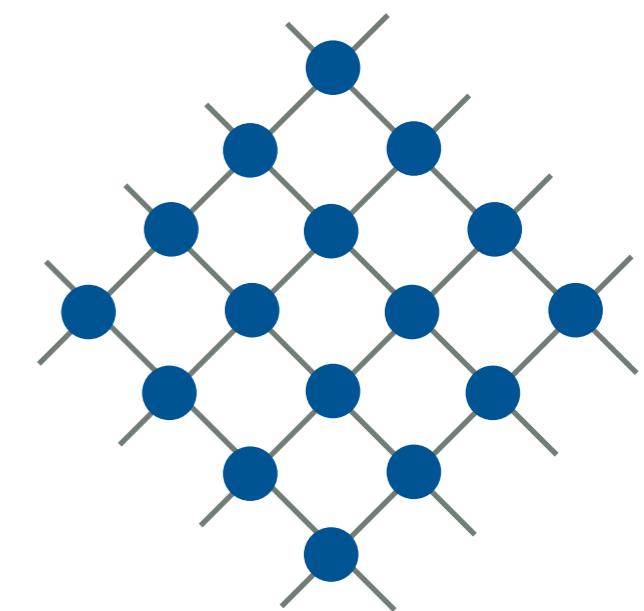
Scalar represented
by $L \times L$ tensors



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

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

$(L \times L)/2$ tensors



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

$$A = U \Lambda V^\dagger \approx \tilde{U} \tilde{\Lambda} \tilde{V}^\dagger$$

$A : M \times N$

$(M \leq N)$

$\Lambda : M \times M$

(Diagonal matrix)

$U, V : (M, N) \times M$

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

(Keeping the R largest singular values)

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

In addition,

$$= \tilde{U} \sqrt{\tilde{\Lambda}} \sqrt{\tilde{\Lambda}} \tilde{V}^\dagger = X Y$$

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

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

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

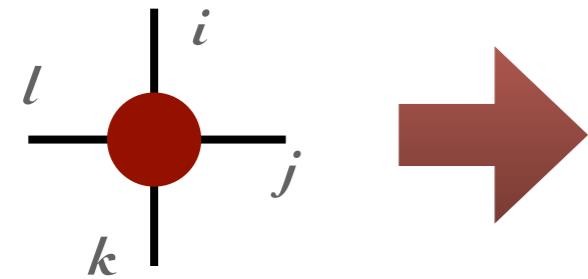
Recipe of Tensor Renormalization Group (TRG)

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

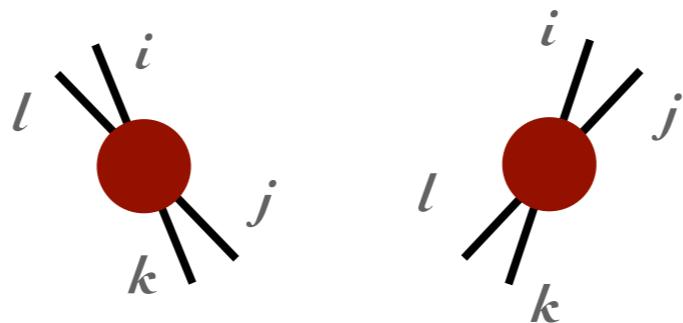
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

1. Decomposition

Regard a tensor as a matrix

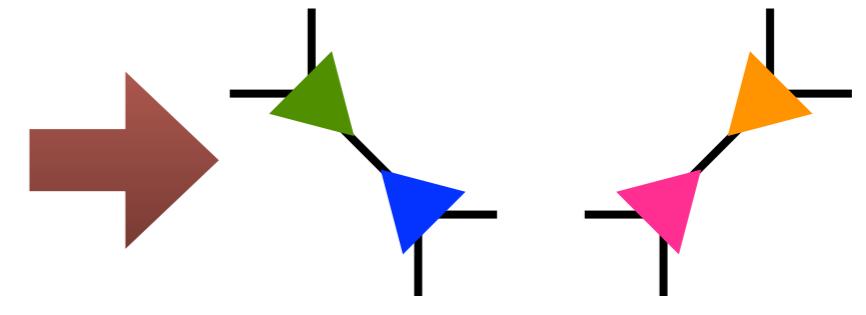


$$A_{i,j,k,l}$$

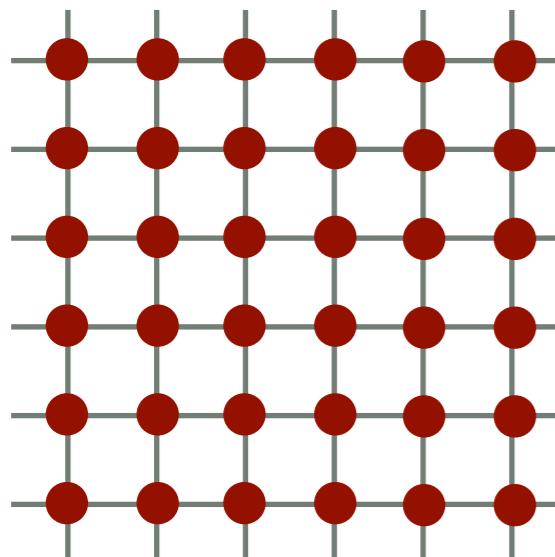


$$A_{(i,l),(j,k)} \quad A_{(i,j),(k,l)}$$

D-rank approximation
by SVD



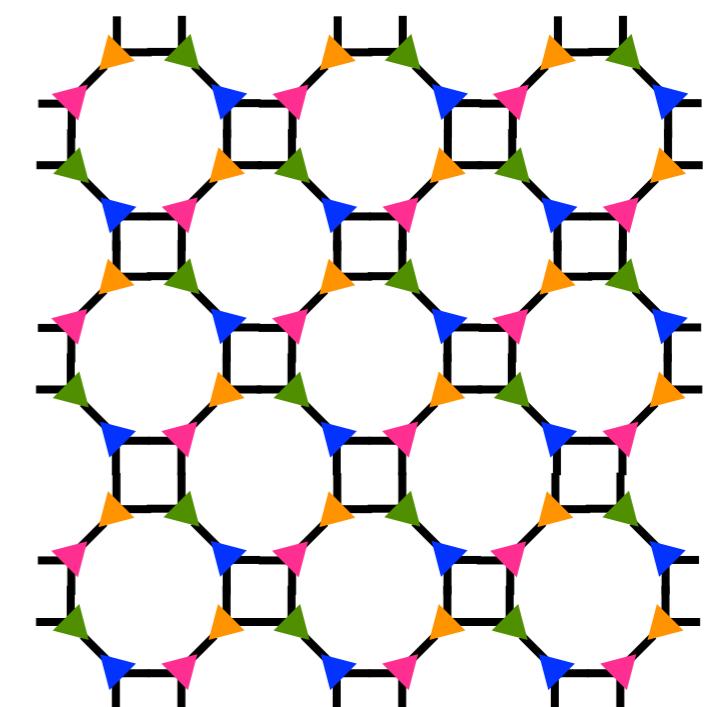
$$A: D \times D \times D \times D$$



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



Approximation

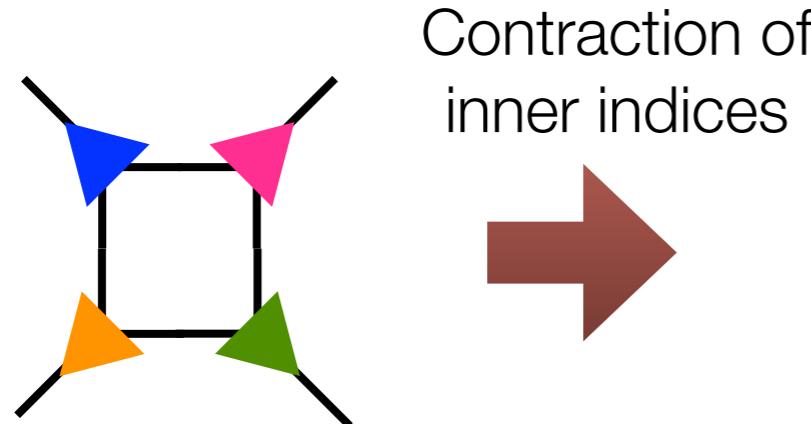


Recipe of Tensor Renormalization Group (TRG)

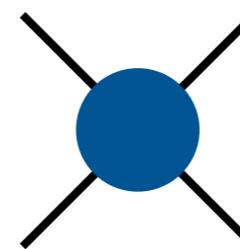
M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)

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

2. Coarse graining

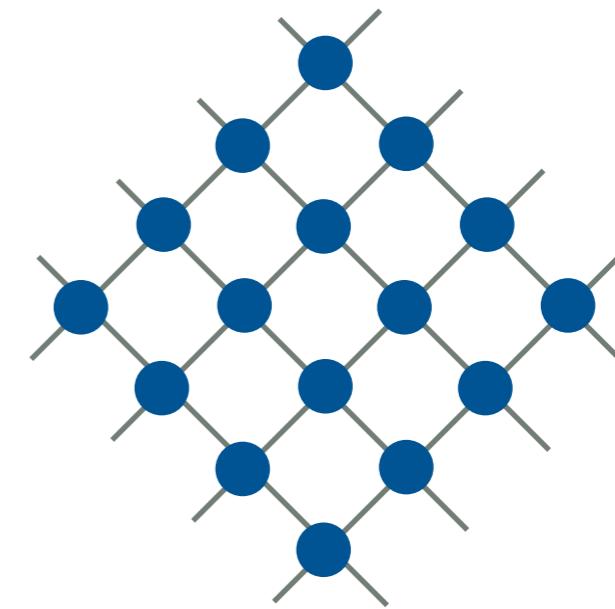
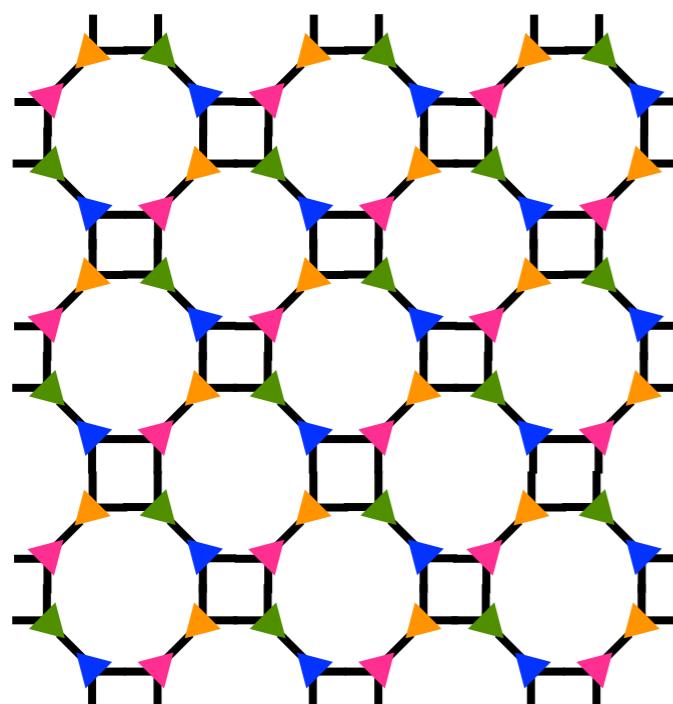


Contraction of
inner indices



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

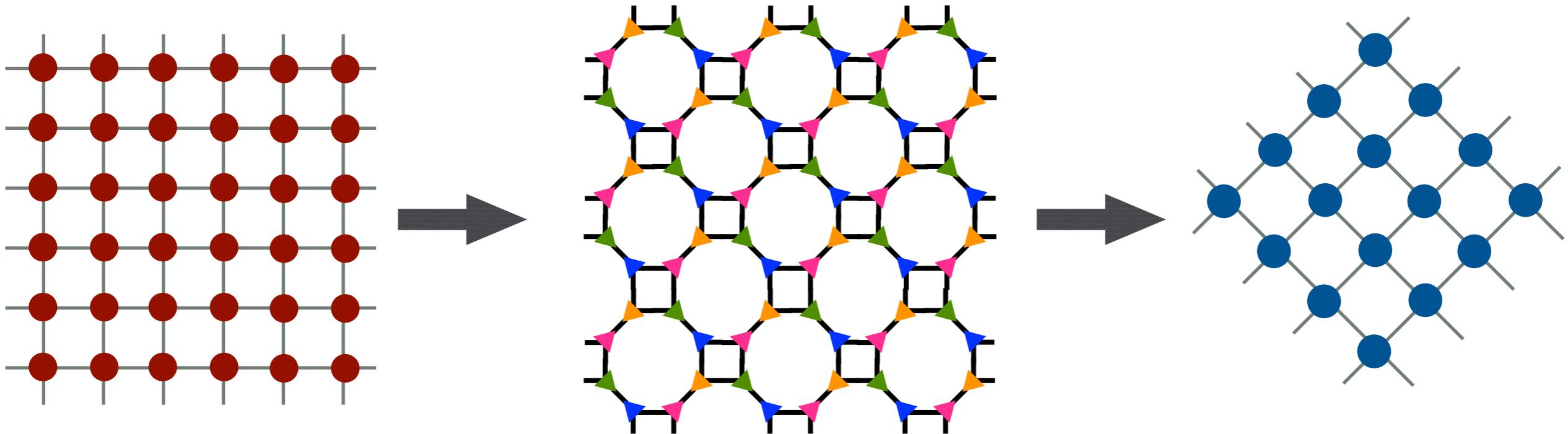
$$\tilde{A} : D \times D \times D \times D$$



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)



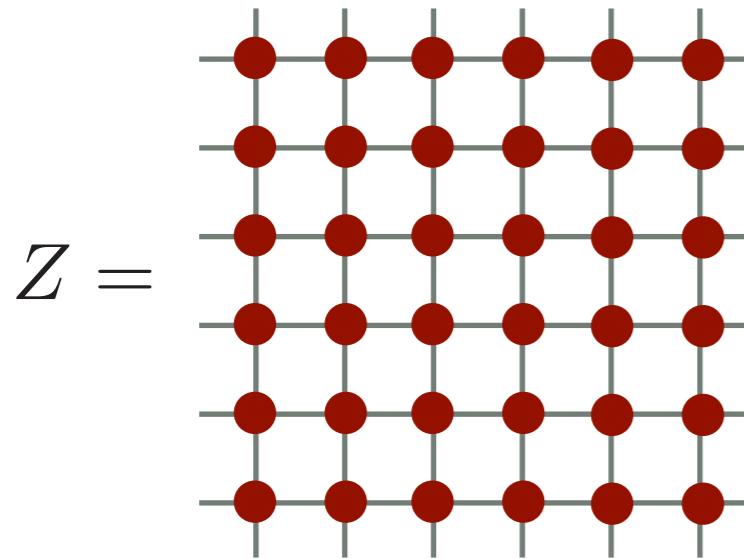
Calculation cost: $\text{SVD} = O(D^6)$ (per tensor)
 $\text{Contraction} = O(D^6)$

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

We can calculate the contraction in polynomial cost!

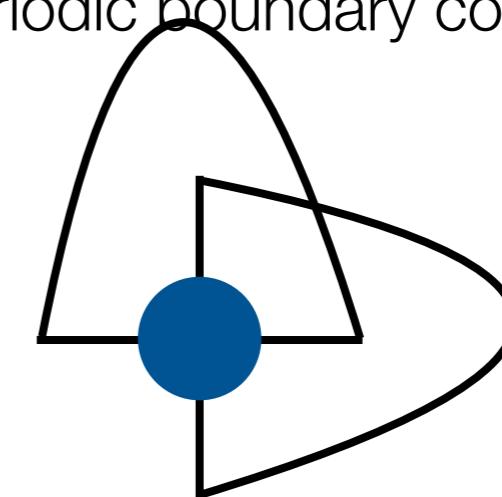
Application to a classical partition function

Partition function

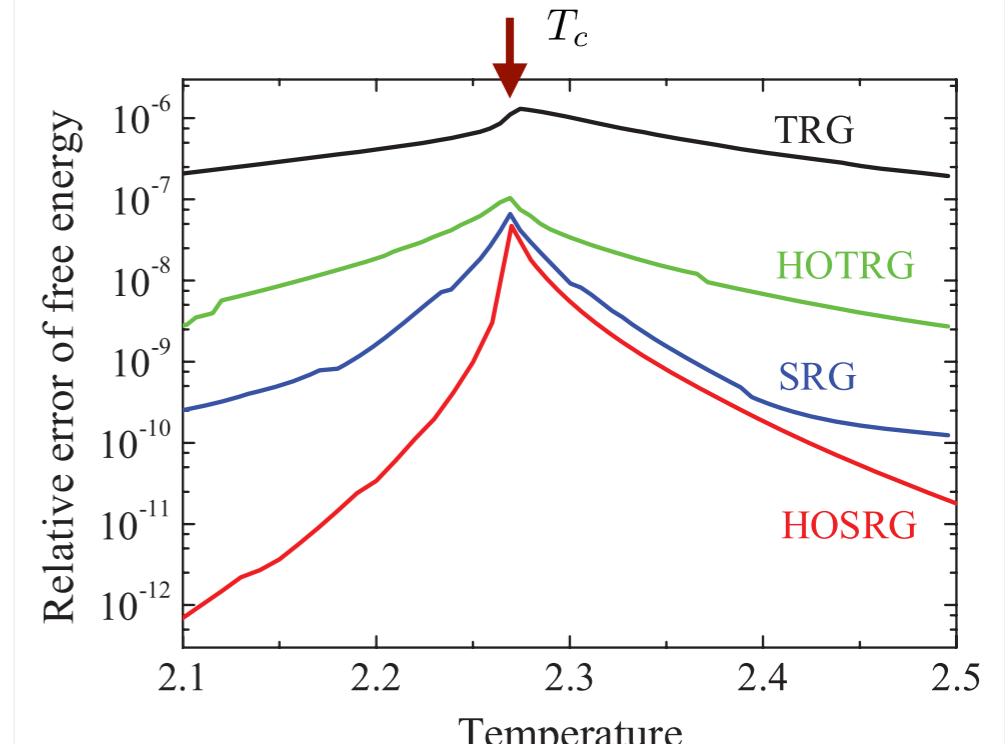


Repeat TRG step
until **only a few**
tensors remain.

(Periodic boundary condition)



Error of free energy for 2D Ising model



We can calculate physical quantities from Z .

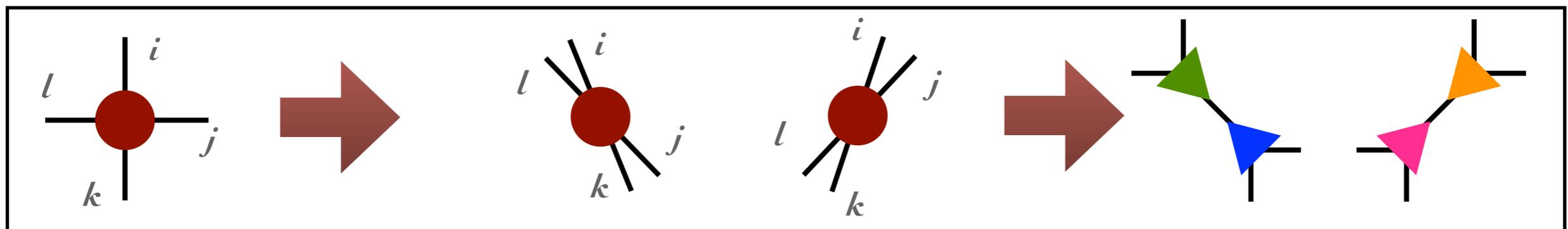
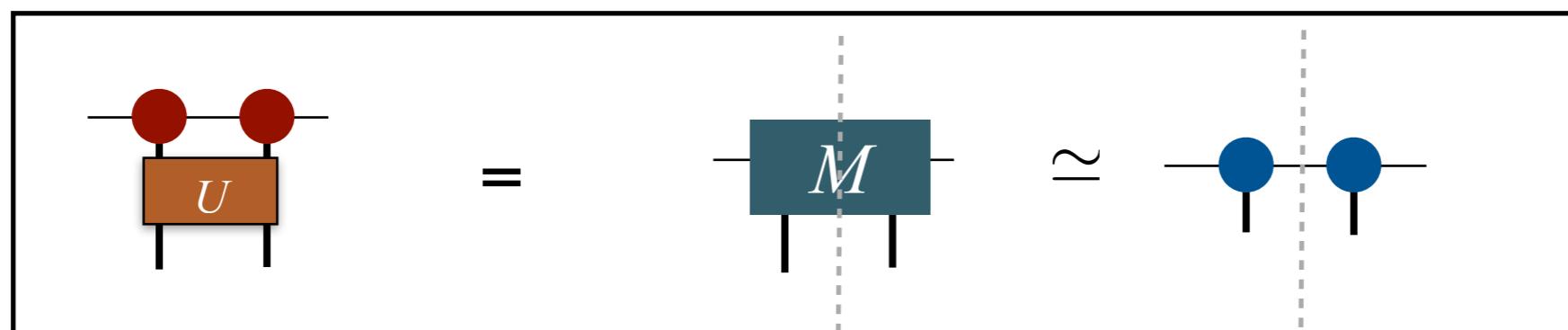
Free energy: $F = -k_B T \ln Z$

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

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

Remarks on TRG

- TRG like idea can be applicable to general contractions of tensor networks.
 - You might find **similarity to the MPS approximation** of gate operation.
- In the application of TRG to critical phenomena, there are lots of interesting topics.
- Keywords: short-range entanglement, renormalization operator, TNR...



Next week (12/21) will be given by Prof. Todo

- 12/14: Tensor network and tensor renormalization group **[Okubo]**
(テンソルネットワークとテンソル繰り込み群)
- **12/21: Quantum computers and simulations [Todo]**
(量子コンピュータ・シミュレーション)
- 1/11: Quantum error corrections and tensor network **[Okubo]**
(量子誤り訂正とテンソルネットワーク)
- 1/25: Quantum-classical hybrid algorithms and tensor network **[Todo]**
(量子古典ハイブリッドアルゴリズムとテンソルネットワーク)