計算科学・量子計算における情報圧縮

Data Compression in Computational Science and Quantum Computing

2022.11.10

#5:情報のエンタングルメントと行列積表現

Entanglement of information and matrix product states

理学系研究科　大久保　毅

Graduate School of Science, **Tsuyoshi Okubo**

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

Today’s topic

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

Outline

- Outline of tensor network decomposition
- Entanglement
  - Schmidt decomposition
  - Entanglement entropy and its area law
- Matrix product states
  - Matrix product states (MPS)
  - Canonical form
  - infinite MPS

Outline of tensor network decomposition

When we efficiently compress a vector?

$$\vec{v} = \sum_{i=1}^{M} C_i \vec{e}_i \qquad \vec{v} \in \mathbb{C}^M$$

If we can find a basis where the coefficients have a structure (correlation).

(1) Almost all C_i are zero (or very small).

We store only a few finite elements $\{(i, C_i)\}$

E.g.

Fourier transformation $\vec{v} = \sum_{k=1}^{M} D_k \vec{f}_k$

If we can neglect larger wave numbers, we can efficiently approximate the vector with smaller number of coefficients.

Classical state $|\Psi\rangle = |01011\ldots00\rangle$

In this case, we know that only a specific C_i is non-zero.

We need only an integer corresponding to the non-zero element.

When we efficiently compress a vector?

$$\vec{v} = \sum_{i=1}^{M} C_i \vec{e}_i \qquad \vec{v} \in \mathbb{C}^M$$

(2) All of C_i are not necessarily independent.

We store **"structure"** and **"independent elements"**.

$$\{(i, C_i)\}$$

E.g. Product state ("generalized" classical state)

A vector is decomposed into product of small vectors.

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

e.g.

$$|\phi_1\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\phi_1\rangle = |01\rangle - |10\rangle$$

(It is identical to the rank-1 CP decomposition.)

structure: **"product state"**

independent elements: **small vectors**

Tensor network decomposition of a vector

Target: Exponentially large Hilbert space

$\vec{v} \in \mathbb{C}^M$

with $M \sim a^N$

\+

Total Hilbert space is decomposed as a product of "local" Hilbert space.

$$\mathbb{C}^M = \mathbb{C}^a \otimes \mathbb{C}^a \otimes \cdots \mathbb{C}^a$$

*Local Hilbert space dimensions can be different.

Examples:

256=2^8

256=2^8

Picture image:

256 × *256* pixel image→ 2^{16} dimensional vector

→ *16*-leg tensor (with a = *2*)

Probability distribution:

e.g. Ising model $P(\{S_i\}) = \dfrac{e^{\beta J \sum_{\langle i,j \rangle} S_i S_j}}{Z}$

→ 2^N vector →N-leg tensor (with a = *2*)

Wave function:

$$|\Psi\rangle = \sum_{\{m_i=0,1\}} T_{m_1,m_2,\cdots,m_N} |m_1, m_2, \cdots, m_N\rangle \quad \rightarrow \quad T_{m_1,m_2,\cdots,m_N}$$

:*N-leg tensor*

Tensor network decomposition of a vector

Target:

Exponentially large Hilbert space

$$\vec{v} \in \mathbb{C}^M \qquad \mathbb{C}^M = \mathbb{C}^a \otimes \mathbb{C}^a \otimes \cdots \mathbb{C}^a$$

Tensor network decomposition

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

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

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

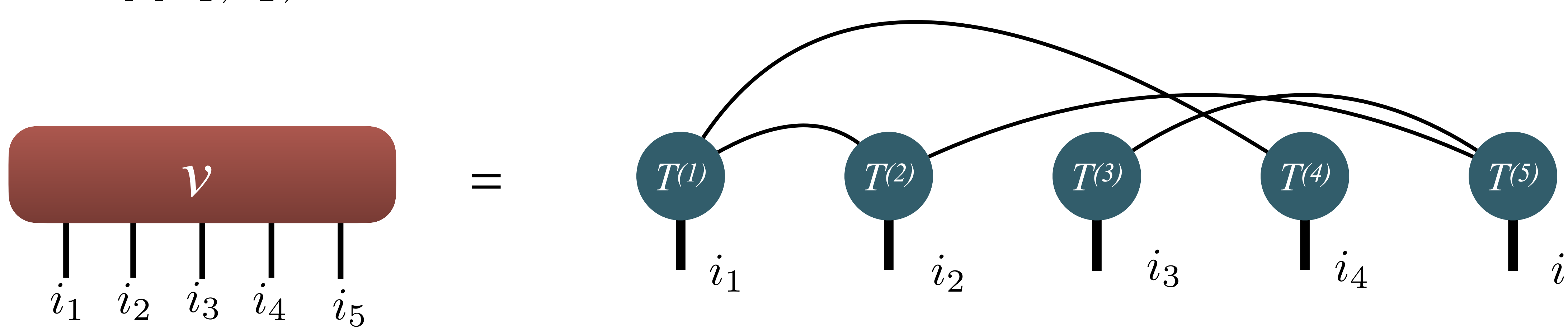
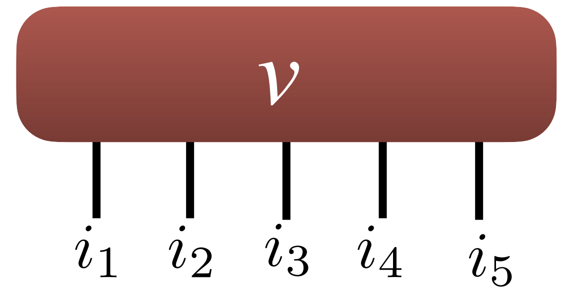


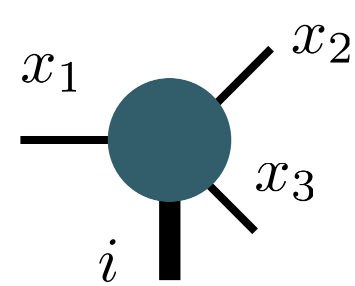
Diagram for a tensor network decomposition

- Vector v_{i_1,i_2,i_3,i_4,i_5}



*Vector looks like a tensor

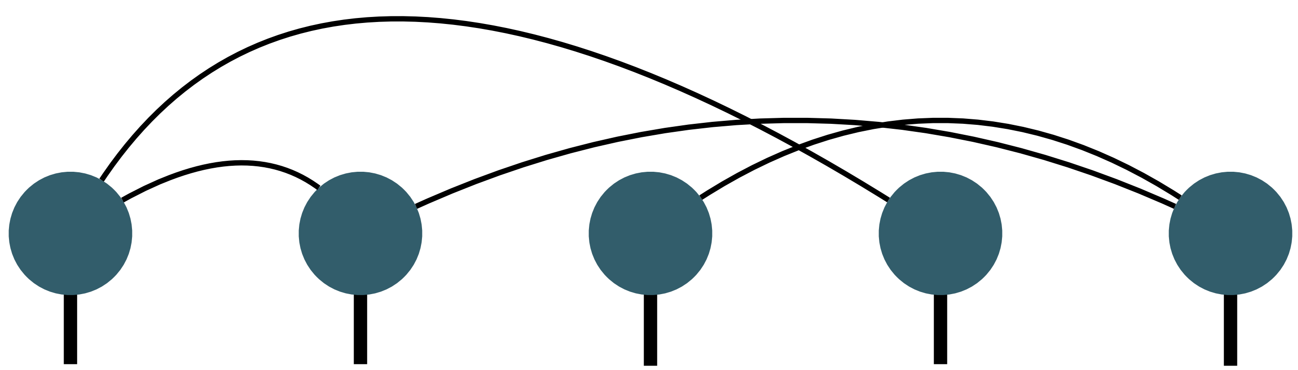
- Tensor $T[i]_{x_1,x_2,x_3}$



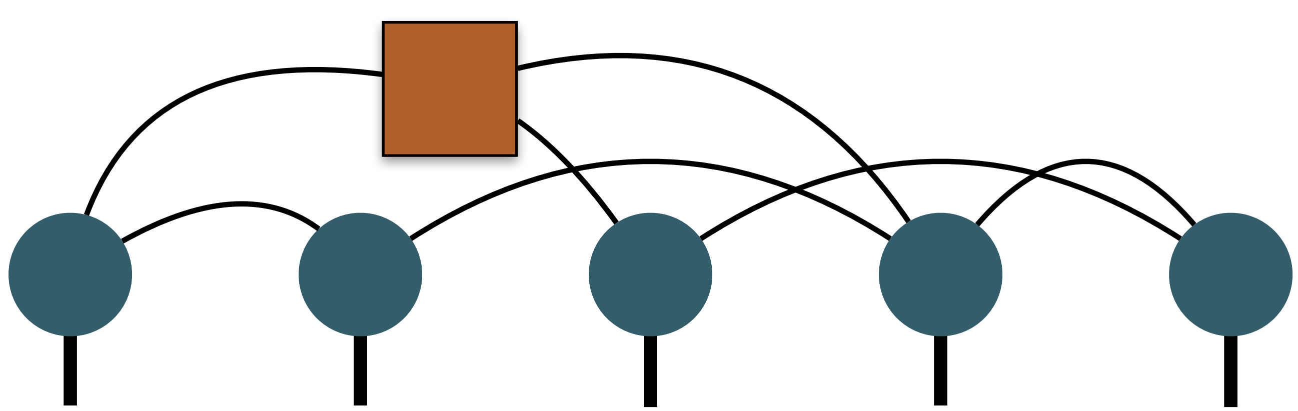
*We treat i as an index of the tensor.

Tensor network decomposition

$\vec{v} =$



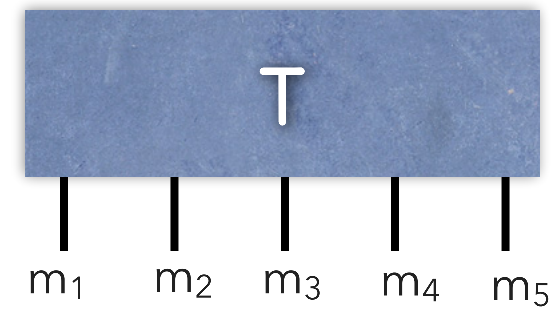
$\vec{w} =$



*We can consider tensors independent on i.

Another "generalization" of SVD to tensors.

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

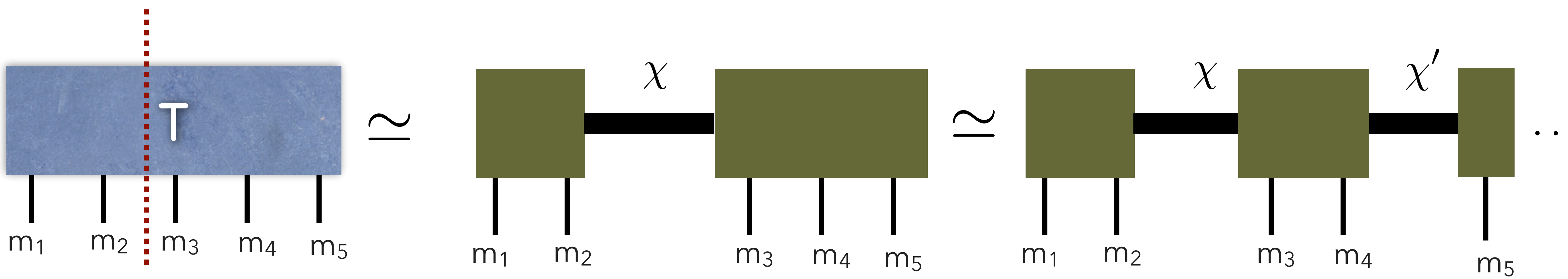


Cf. wave function: $|\Psi\rangle = \sum_{\{m_i=0,1\}} T_{m_1,m_2,\cdots,m_N}|m_1,m_2,\cdots,m_N\rangle$

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

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

We can perform the low rank approximation of T.



*obtained two objects are again tensors.

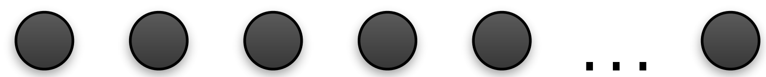
What does it mean?

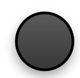
It is related to MPS

Entanglement（エンタングルメント）

N-qubit system (S=1/2 quantum spin system)

Example vector: Wave function of N-qubit systems



 takes two states $|0\rangle, |1\rangle$
$(|\uparrow\rangle, |\downarrow\rangle)$

$$
\begin{aligned}
|\Psi\rangle &= \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_N\rangle \\
&= \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle
\end{aligned}
$$

Coefficients = vector: $\vec{\Psi} \in \mathbb{C}^{2^N}$

* Inner product: $\langle \Phi | \Psi \rangle = \vec{\Phi}^* \cdot \vec{\Psi}$

Schmidt decomposition

General vector: $\vec{x} \in \mathbb{V}_1 \otimes \mathbb{V}_2$

$$\dim\, \mathbb{V}_1 = n_1, \dim\, \mathbb{V}_2 = n_2$$
$$(n_1 \geq n_2)$$

Schmidt decomposition

There exists special basis which satisfies

$$\vec{x} = \sum_{i=1}^{n_2} \lambda_i \vec{u}_i \otimes \vec{v}_i$$

No off-diagonal coupling!

Orthonormal basis

$$\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_{n_1}\} \in \mathbb{V}_1$$
$$\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{n_2}\} \in \mathbb{V}_2$$

Schmidt coefficient $\lambda_i \geq 0$

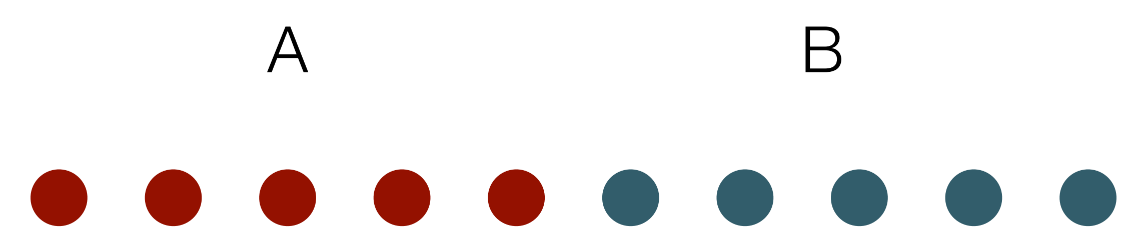
Schmidt decomposition is unique.

Schmidt decomposition of a quantum physics

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

Schmidt decomposition

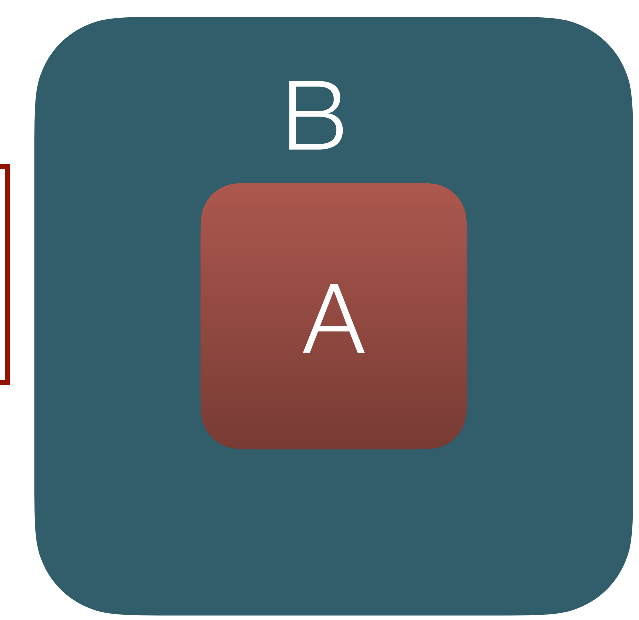
Divide a system into two parts, A and B:



General wave function can be represented by a superposition of orthonormal basis set.

$$|\Psi\rangle = \sum_{i,j} M_{i,j} |A_i\rangle \otimes |B_j\rangle = \sum_i \lambda_i |\alpha_i\rangle \otimes |\beta_i\rangle$$

$$\boxed{M_{i,j} \equiv \Psi_{\underbrace{(i_1,\dots)}_{\mathrm{A}},\underbrace{(\dots,i_N)}_{\mathrm{B}}} \quad |A_i\rangle = |i_1, i_2, \dots\rangle \quad |B_j\rangle = |\dots, i_{N-1}, i_N\rangle}$$



Orthonormal basis: $\langle A_i|A_j\rangle = \langle B_i|B_j\rangle = \delta_{i,j},$
$\langle \alpha_i|\alpha_j\rangle = \langle \beta_i|\beta_j\rangle = \delta_{i,j}$

Schmidt coefficient: $\lambda_i \geq 0$

Relation between SVD and Schmidt decomposition

$$|\Psi\rangle = \sum_{i,j} M_{i,j} |A_i\rangle \otimes |B_j\rangle$$

Singular values: $\lambda_m \geq 0$

SVD

$$M_{i,j} = \sum_m U_{i,m} \lambda_m V^\dagger_{m,j}$$

Singular vectors:

$$\sum_i U^\dagger_{m,i} U_{i,m'} = \delta_{m,m'}$$

$$\sum_j V^\dagger_{m,j} V_{j,m'} = \delta_{m,m'}$$

Relation to the Schmidt decomposition:

$$|\Psi\rangle = \sum_{i,j} M_{i,j} |A_i\rangle \otimes |B_j\rangle = \sum_m \lambda_m |\alpha_m\rangle \otimes |\beta_m\rangle$$

$$|\alpha_m\rangle = \sum_i U_{i,m} |A_i\rangle$$

$$|\beta_m\rangle = \sum_j V^\dagger_{m,j} |B_j\rangle$$

$$\Rightarrow \quad \langle \alpha_m | \alpha_{m'} \rangle = \langle \beta_m | \beta_{m'} \rangle = \delta_{m,m'}$$

SVD of the quantum state is directly related to the Schmidt decomposition.

$\Rightarrow$ **# of non-zero Schmidt coefficients is the rank of the matrix M**

Schmidt rank of a quantum state

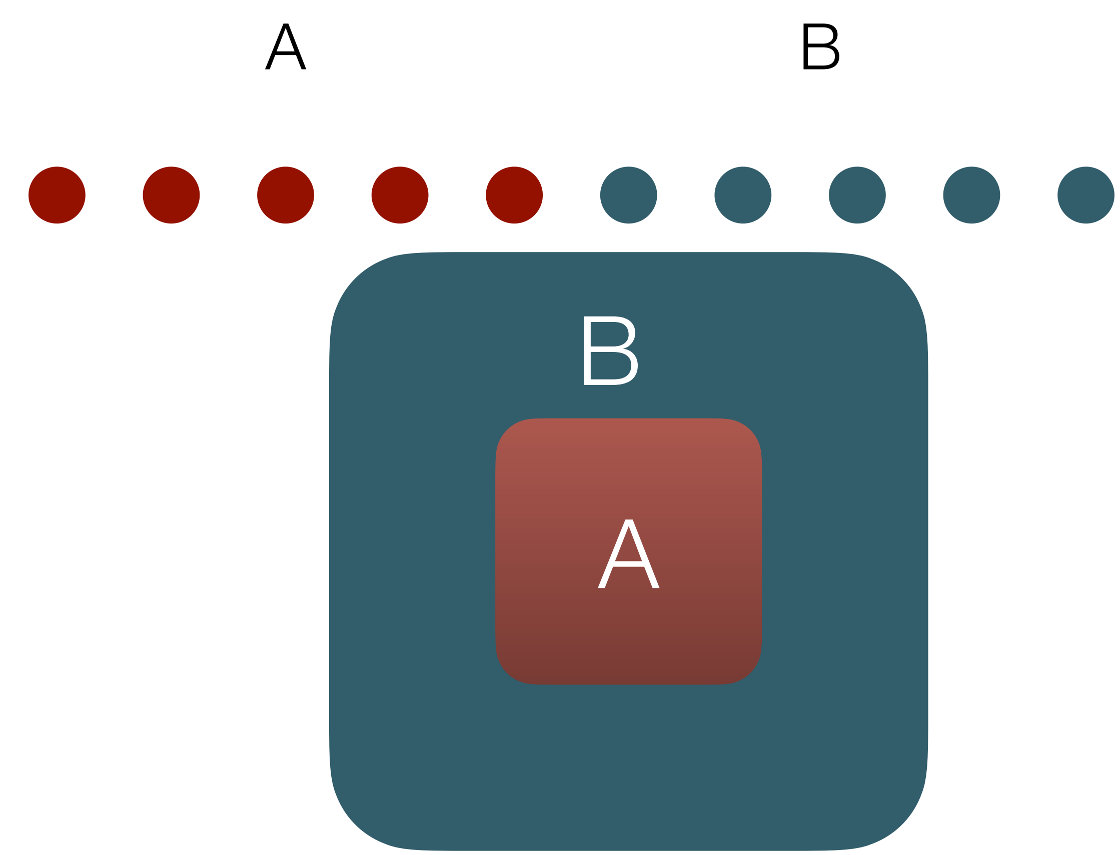
$$|\Psi\rangle = \sum_{i,j} M_{i,j} |A_i\rangle \otimes |B_j\rangle = \sum_m \lambda_m |\alpha_m\rangle \otimes |\beta_m\rangle$$

Schmidt rank:

# of non-zero Schmidt coefficients (= rank of the matrix *M*)

Schmidt rank characterizes the quantum correlation between *A* and *B*.

- Schmidt rank = 1 $|\Psi\rangle = |\alpha\rangle \otimes |\beta\rangle$



The quantum state is represented as a single product of two states.

Product state

- Schmidt rank > 1 $|\Psi\rangle = \sum_m \lambda_m |\alpha_m\rangle \otimes |\beta_m\rangle$

The quantum state is represented as a sum of product states.

(The Origine of the nonlocal correlation)

Cf. Nobel prize in physics 2022

Partial trace and reduced density matrix

For $\vec{x} \in \mathbb{V}_1 \otimes \mathbb{V}_2$ $\quad \dim\, \mathbb{V}_1 = n_1, \dim\, \mathbb{V}_2 = n_2$ $\quad |\vec{x}| = 1$

Density matrix: $\rho \equiv \vec{x}\vec{x}^\dagger$ $\quad (\rho_{ij} = x_i x_j^*)$

（密度行列） $(\rho = |x\rangle\langle x|)$ $\qquad$ *Note: $\mathrm{rank}\ \rho = 1$

Orthonormal basis: $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_{n_1}\} \in \mathbb{V}_1$ $\quad \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_{n_2}\} \in \mathbb{V}_2$

Basis for $\vec{x}$: $\vec{g}_{i_1,i_2} = \vec{e}_{i_1} \otimes \vec{f}_{i_2}$

Index: $i = (i_1, i_2)$

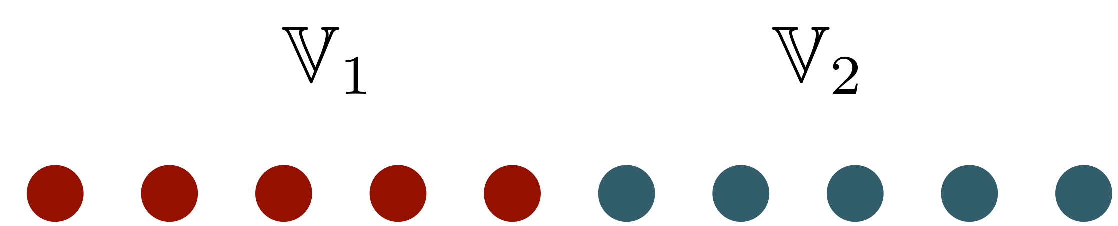
Reduced Density matrix:

（縮約密度行列）

$\rho_{\mathbb{V}_1} \equiv \mathrm{Tr}_{\mathbb{V}_2}\rho$: a positive-semidefinite square matrix in $\mathbb{V}_1$

*Note: generally, $\mathrm{rank}\ \rho_{\mathbb{V}_1} > 1$

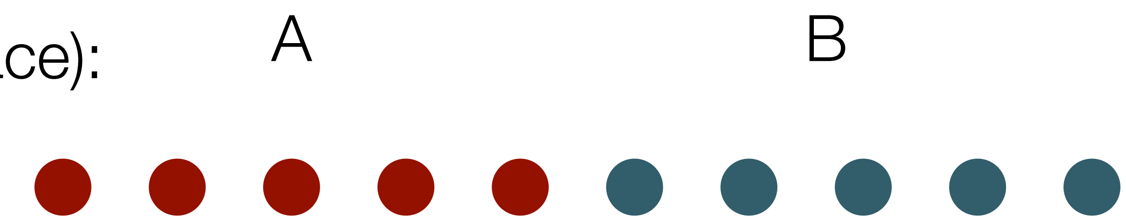
$$\boxed{(\rho_{\mathbb{V}_1})_{i_1,j_1} = \sum_{\underline{i_2}} \rho_{(i_1,\underline{i_2}),(j_1,\underline{i_2})}}$$



Entanglement entropy

Entanglement entropy:

Reduced density matrix of a sub system (sub space):

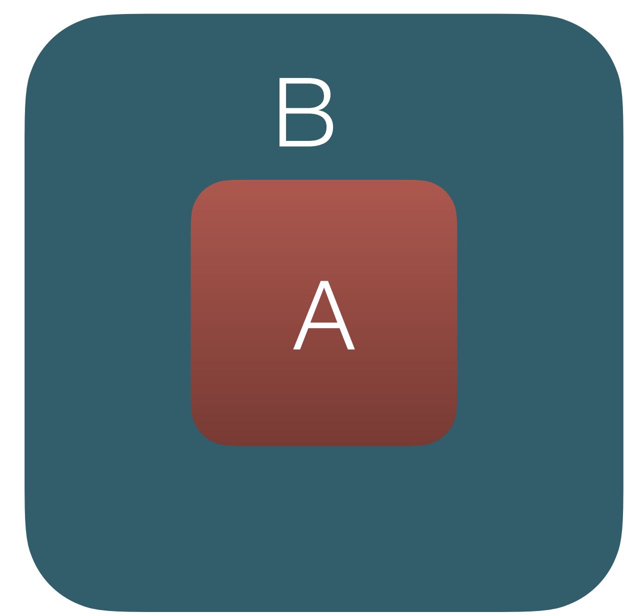


$$\rho_A = \mathrm{Tr}_B |\Psi\rangle\langle\Psi|$$

Entanglement entropy = von Neumann entropy of ρ_A

$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right)$$

Schmidt decomposition $|\Psi\rangle = \sum_i \lambda_i |\alpha_i\rangle \otimes |\beta_i\rangle$



$$\rho_A = \sum_i \lambda_i^2 |\alpha_i\rangle\langle\alpha_i|$$ (*Exercise)

$$S = -\sum_i \lambda_i^2 \log \lambda_i^2$$

$$(\sum_i \lambda_i^2 = 1)$$

Entanglement entropy is calculated through the spectrum of Schmidt coefficients.

(It also indicates $S = -\mathrm{Tr}(\rho_B \log \rho_B)$)

Intuition for EE

Entanglement entropy is related to spectrum of singular values.

$$S = -\mathrm{Tr}(\rho_A \log \rho_A) = -\sum_i \lambda_i^2 \log \lambda_i^2$$

- $\mathrm{rank}\rho_A = 1$

  $\lambda_1 = 1, \lambda_j = 0 \ (j \neq 1)$ ➡ $S = 0$

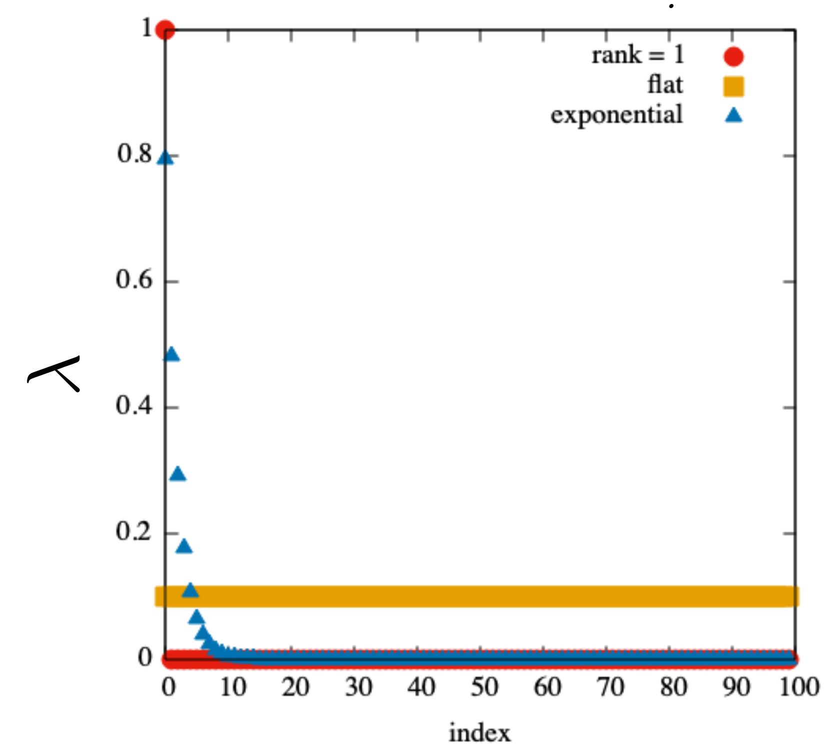
- Flat spectrum

  $\lambda_1 = \lambda_2 = \cdots = \lambda_n = \dfrac{1}{\sqrt{n}}$ ➡ $S = \log n$

- Exponential decay

  $\lambda_i \propto e^{-\alpha i}$ ➡ $S = 1 - \log 2\alpha \ (\alpha \ll 1, \alpha n \to \infty)$

Normalization: $(\sum \lambda_i^2 = 1)$



Smaller exponent gives larger entropy.

Intuition for EE: two S=1/2 spins

A B

1. $|\Psi\rangle = |\uparrow\rangle \otimes |\downarrow\rangle$

A product state ➡ $\lambda = 1, \; S = 0$

Product state : S=0

2. $|\Psi\rangle = \frac{1}{2}(|\uparrow\rangle - |\downarrow\rangle) \otimes (|\uparrow\rangle - |\downarrow\rangle)$

Another product state ➡ $\lambda = 1 \; , \; S = 0$

3. $|\Psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle \otimes |\downarrow\rangle - |\downarrow\rangle \otimes |\uparrow\rangle)$

Spin singlet ➡ $\lambda_1 = \lambda_2 = \frac{1}{\sqrt{2}} \; , \quad S = \log 2$ **Maximally entangled State**

4. $|\Psi\rangle = \left(x|\uparrow\rangle \otimes |\downarrow\rangle + \sqrt{1-x^2}|\downarrow\rangle \otimes |\uparrow\rangle\right)$

Complicated state ➡ $\lambda_1 = |x|, \lambda_2 = \sqrt{,1-x^2}$

$S = x^2 \log x^2 + \sqrt{1-x^2} \log(1-x^2)$

Larger entanglement entropy ~ Larger correlation between two parts

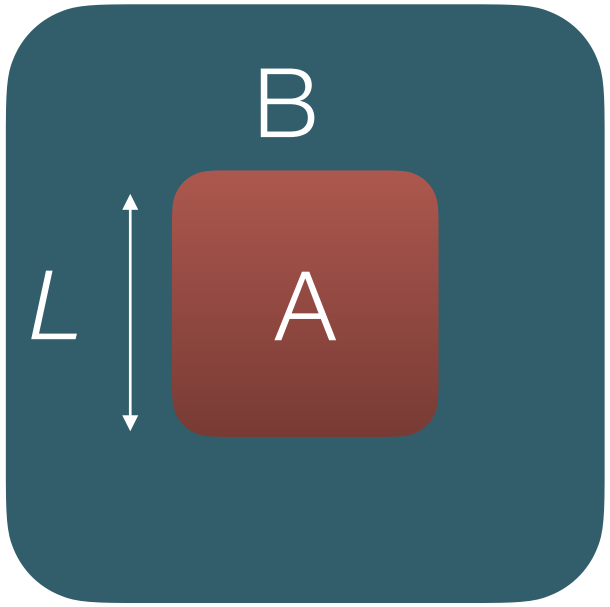
Area law of the entanglement entropy in physics

General wave functions (vector):

EE is proportional to its **volume (# of qubits)**.

$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right) \propto L^d$$

(c.f. random vector)



Ground state wave functions:

For a lot of ground states, EE is proportional to its area.

J. Eisert, M. Cramer, and M. B. Plenio, Rev. Mod. Phys, 277, **82** (2010)

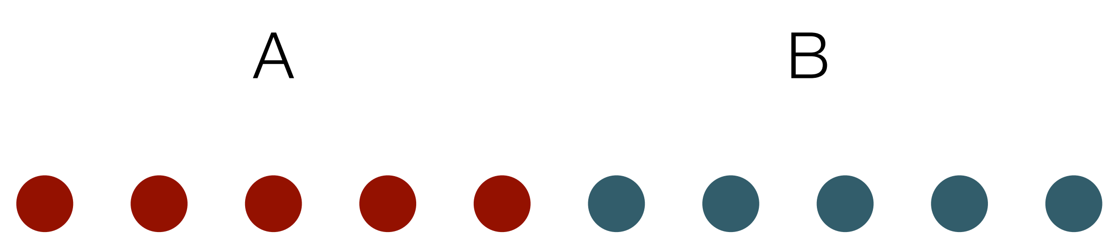
$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right) \propto L^{d-1}$$

In the case of one-dimensional system:

Gapped ground state for local Hamiltonian

M.B. Hastings, J. Stat. Mech.: Theory Exp. P08024 (2007)

$$S = O(1)$$



Ground state are in a small part of the huge Hilbert space

Expected entanglement scaling for spin systems

Table 1
Entanglement entropy scaling for various examples of states of matter, either disordered, ordered, or critical, with smooth boundaries (no corners).

Physical state	Entropy	Example
Gapped (brok. disc. sym.)	$aL^{d-1} + \ln(\mathrm{deg})$	Gapped XXZ [143]
$d = 1$ CFT	$\frac{c}{3} \ln L$	$s = \frac{1}{2}$ Heisenberg chain [21]
$d \geq 2$ QCP	$aL^{d-1} + \gamma_{\mathrm{QCP}}$	Wilson–Fisher O(N) [136]
Ordered (brok. cont. sym.)	$aL^{d-1} + \frac{n_G}{2} \ln L$	Superfluid, Néel order [147]
Topological order	$aL^{d-1} - \gamma_{\mathrm{top}}$	$\mathbb{Z}_2$ spin liquid [159]

(Nicolas Laflorencie, Physics Reports **646**, 1 (2016))

cf. free fermion

$$S \propto L^{d-1} \log L$$

For d ≥ 2, leading contribution satisfies area low
even for gapless (critical) systems.

Exercise: examples of Schmidt decomposition

1-1: Random wave function (Sample code: Ex1-1.ipynb)

- Make a random vector
- SVD it and see singular value spectrum and EE

1-2: Ground state of the transverse field Ising model

$$\mathcal{H} = -\sum_{i=1}^{L-1} S_{i,z}S_{i+1,z} - \Gamma\sum_{i=1}^{L} S_{i,x}$$

(Sample code: Ex1-2.ipynb)

- Calculate GS by diagonalizing Hamiltonian
- SVD it and see singular value spectrum and EE

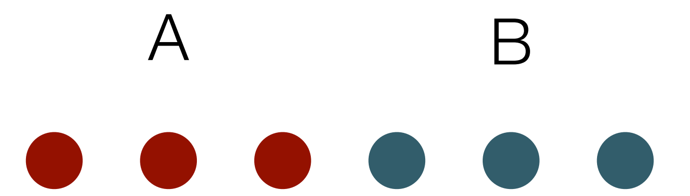
(Sample code: Ex1-3.ipynb)

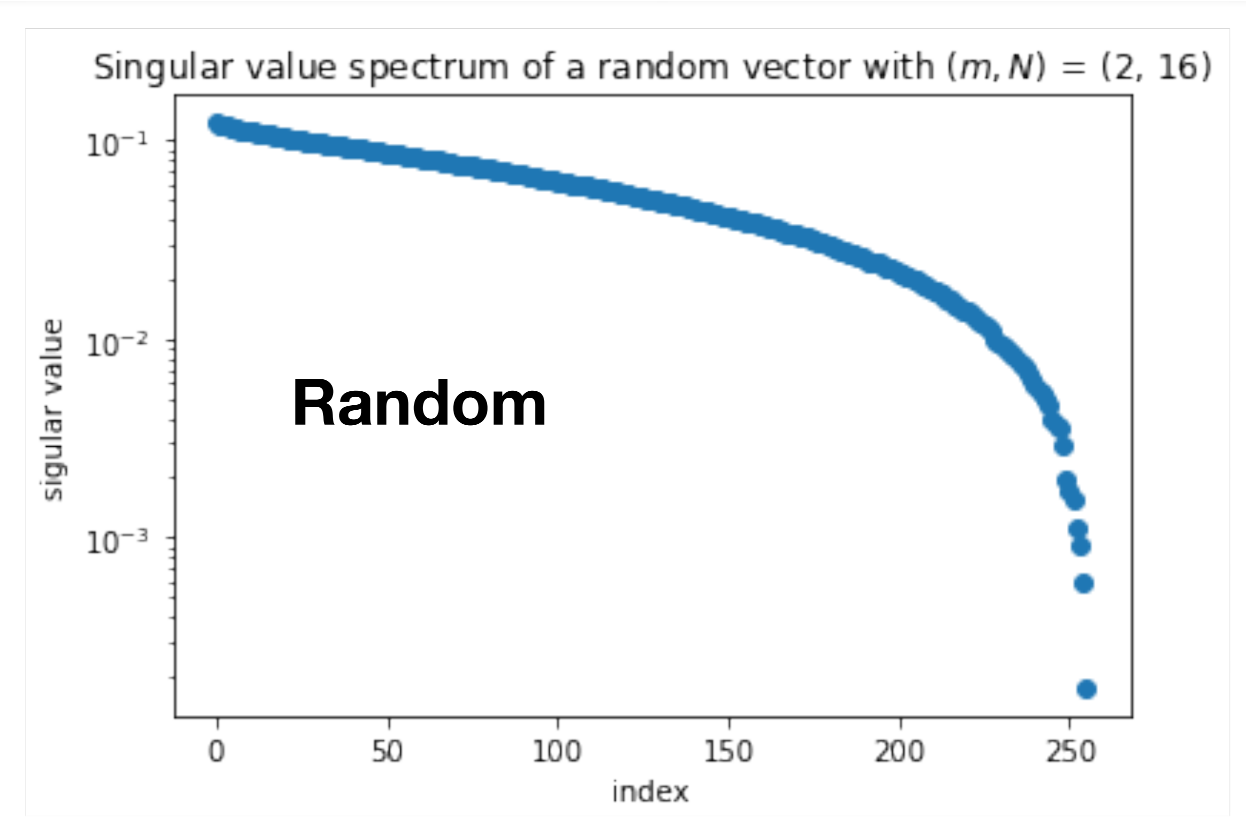
1-3: Picture image

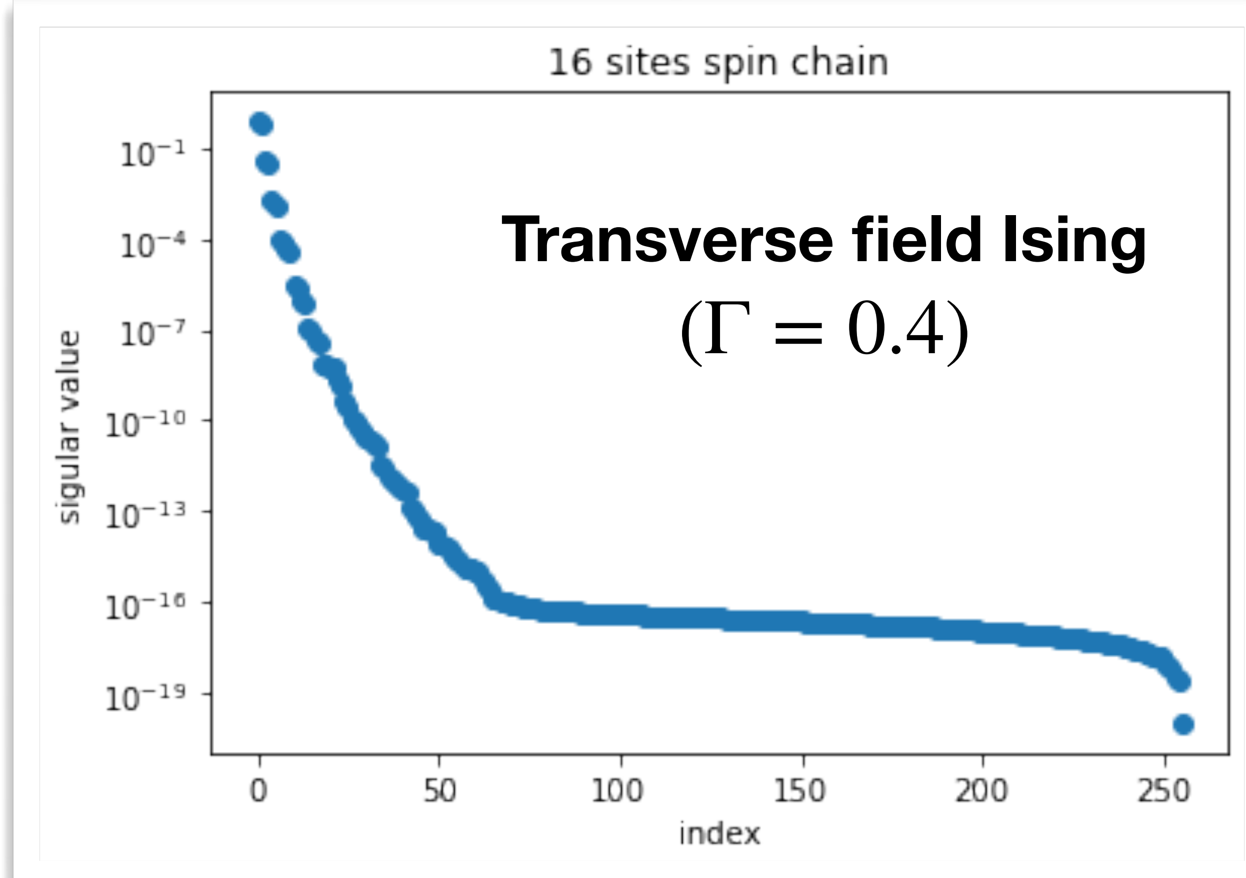
- Transform an image data to the vector in m^N dimension.
- SVD it and see singular value spectrum and EE

*** Try to simulate different system size "*N*"**
*** You can simulate other S by changing "*m*"**

Spectrum for N=16 $\vec{v} \in \mathbb{C}^{2^{16}}$



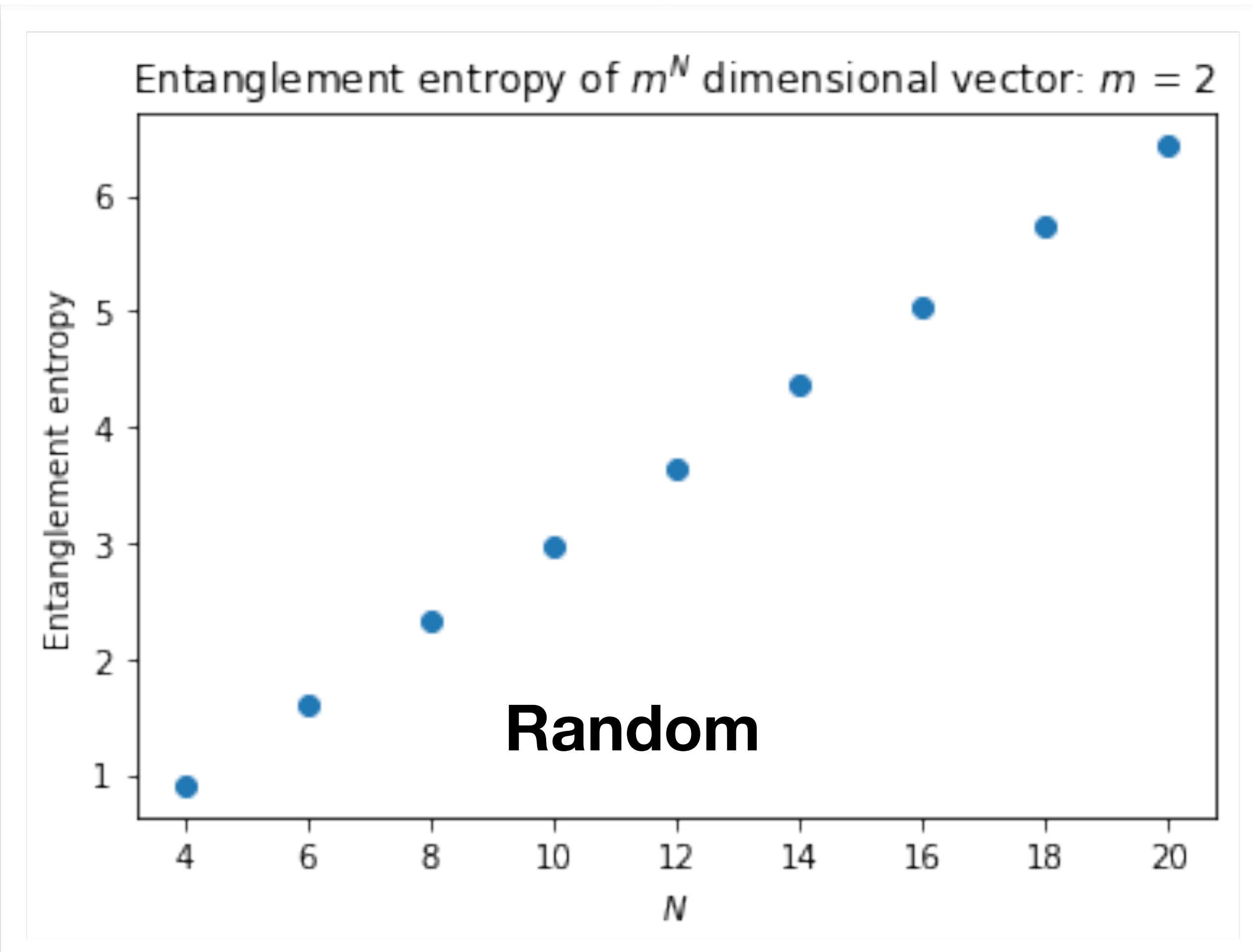


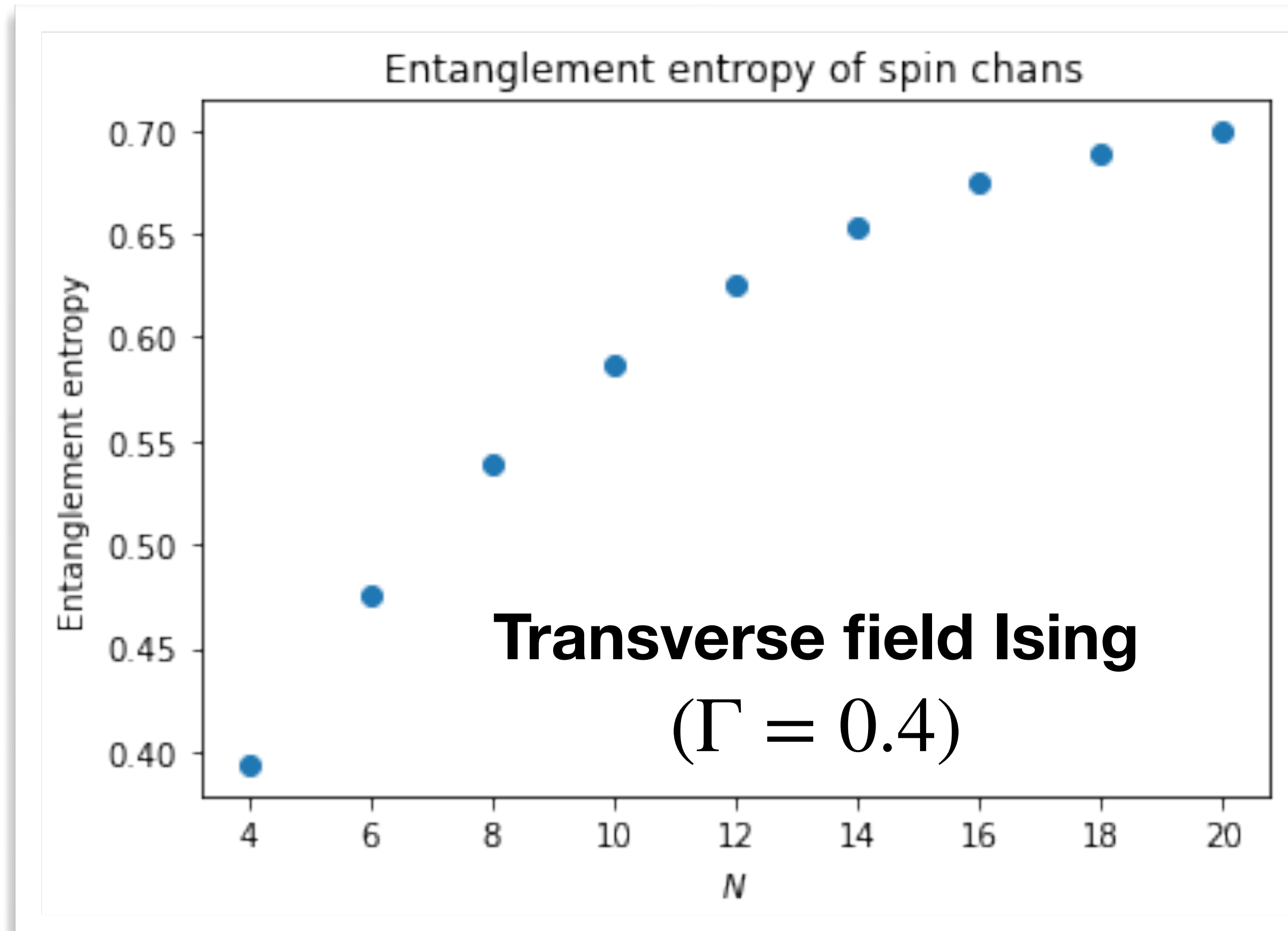


Ground state wave function has lower entanglement!

Scaling of the entanglement entropy

$\vec{v} \in \mathbb{C}^{2^N}$





Random vector: Volume low
Ground state: Area low

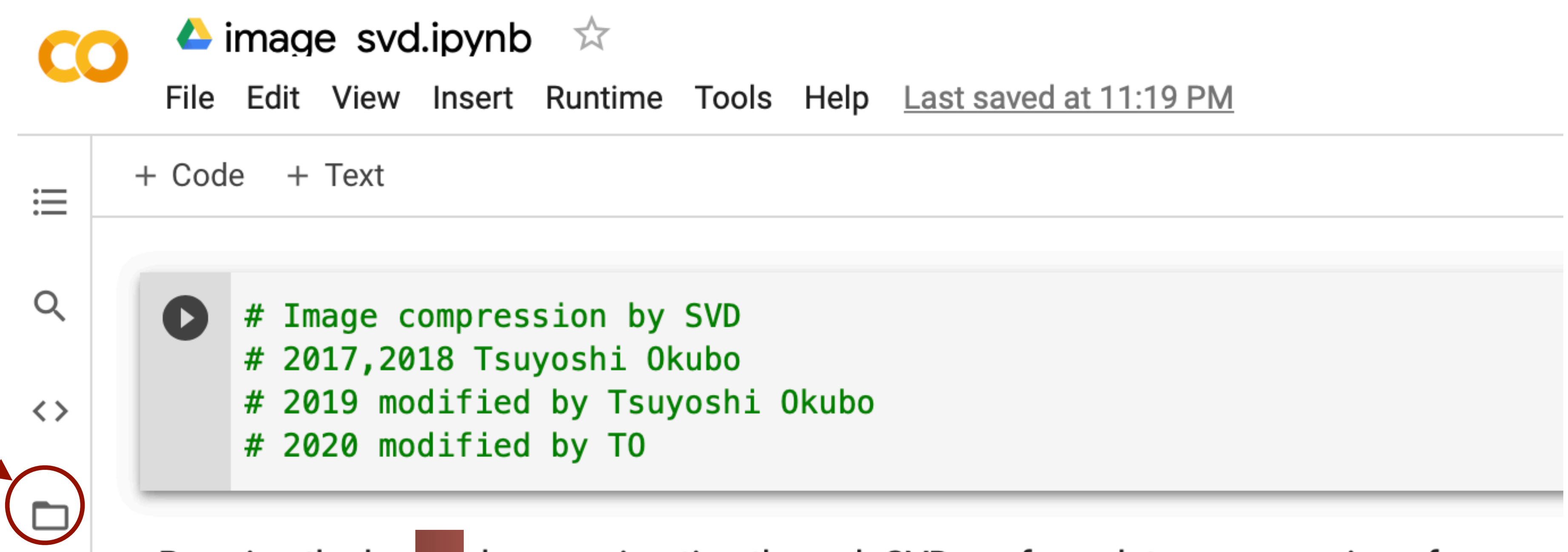
Exercises with Google Colab

I recommend you to use google colaboratory,
https://colab.research.google.com
where you can run .ipynb from your web browser.

When you use Google Colab, you need to also upload
"ED.py"
for the case of "Ex1-2.ipynb", and
your image file (sample.jpg),
for the case of "Ex1-3.ipynb".

How to use Google Colab

1. Open Ex1-3.ipynb in Google colab
   - Select "**File/upload notebook**" ("ファイル/ノートブックをアップロード") and upload Ex1-3.*ipynb*

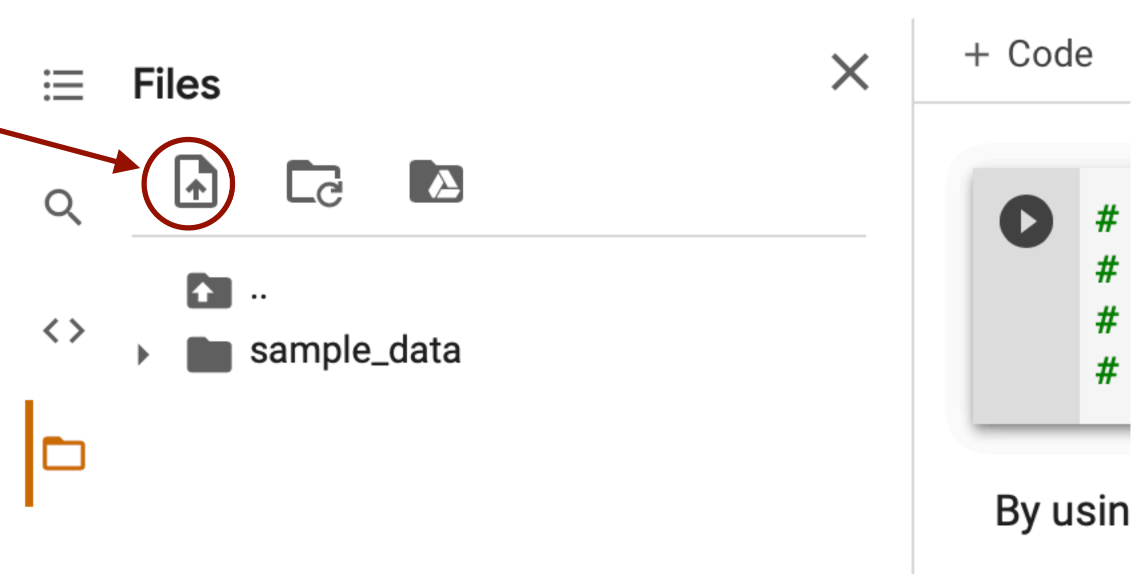


2. Click here

   (Wait a moment for the connection)

3. Click here and upload your image file (e.g. sample.jpg).

   (Uploaded file will be deleted after the session finishes.)



4. Select "**Runtime/Restart and Run all**".

Matrix product states（行列積状態）

(Tensor train decomposition)

Data compression of tensors (vectors)

Eg. General wave function:

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

Coefficient vector can represent any points in the Hilbert space.

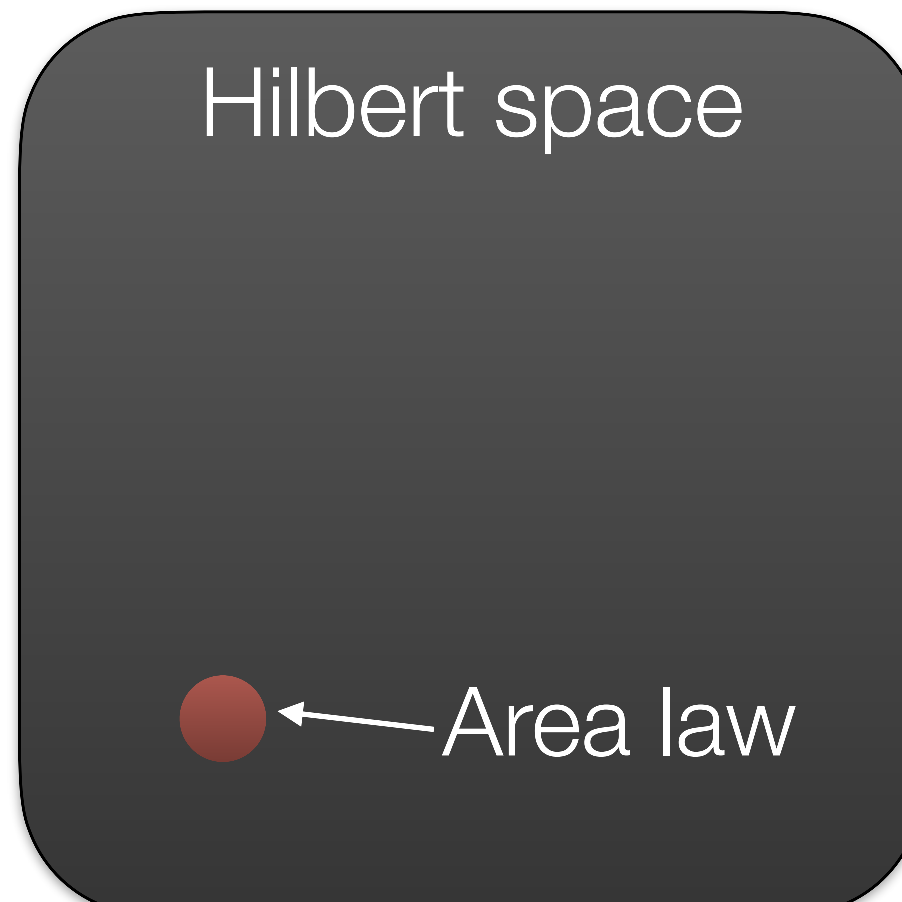
Ground states satisfy the area law.

In order to represent the ground state accurately, we might not need all of a^N elements.

Data compression by tensor decomposition:

Tensor network decomposition

***Same idea holds for any tensors.**



Matrix product state (MPS)

Good reviews:

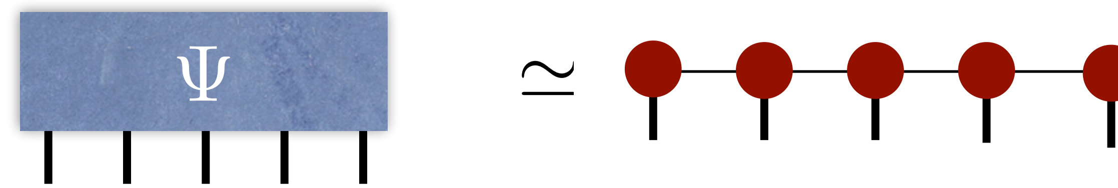
(U. Schollwöck, Annals. of Physics **326**, 96 (2011))
(R. Orús, Annals. of Physics **349**, 117 (2014))

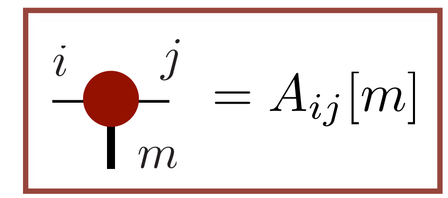
$$|\Psi\rangle = \sum_{\{i_1, i_2, \dots i_N\}} \Psi_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle$$

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

$A[i]$: Matrix for state i

MPS





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)

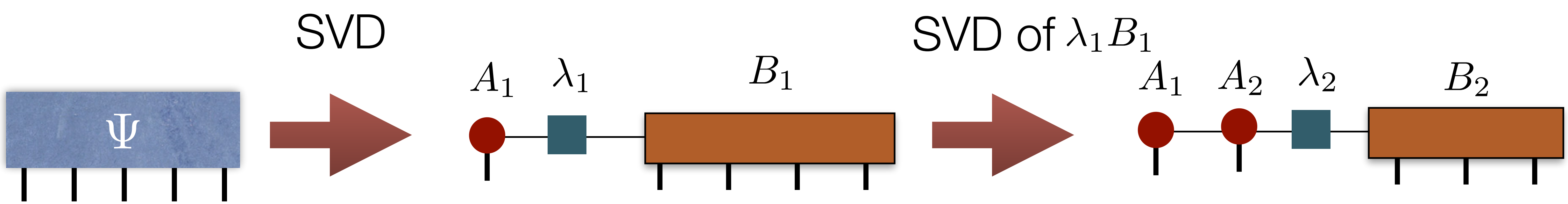
$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

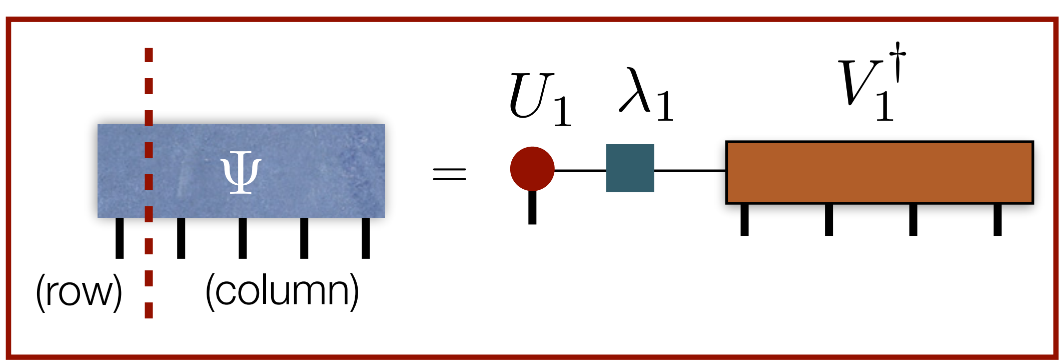
$$\Psi_{i_1 i_2 \dots i_N} = \phi_1[i_1] \phi_2[i_2] \cdots \phi_N[i_N]$$

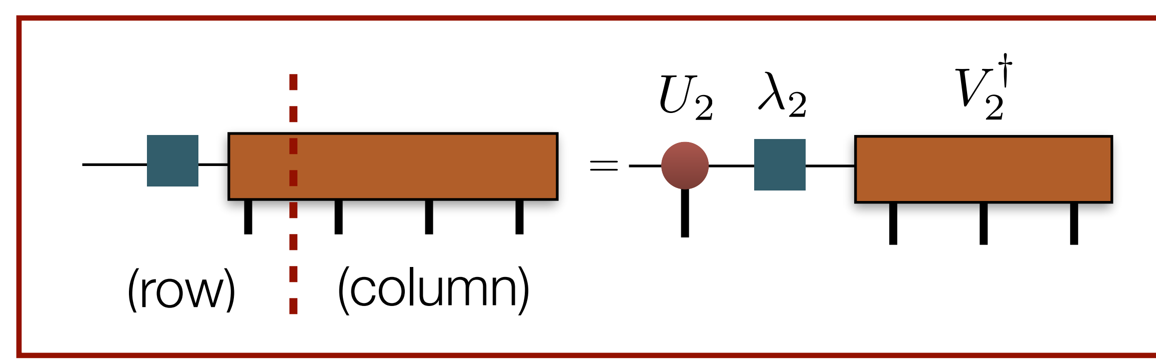
$$\phi_n[i] \equiv \langle i | \phi_i \rangle$$

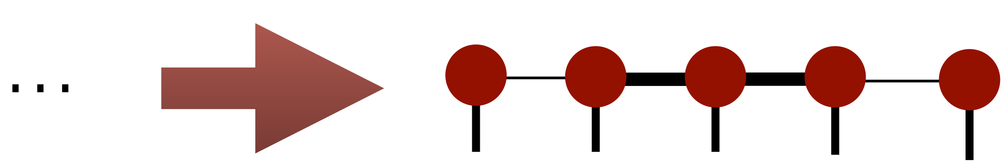
Matrix product state without approximation

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







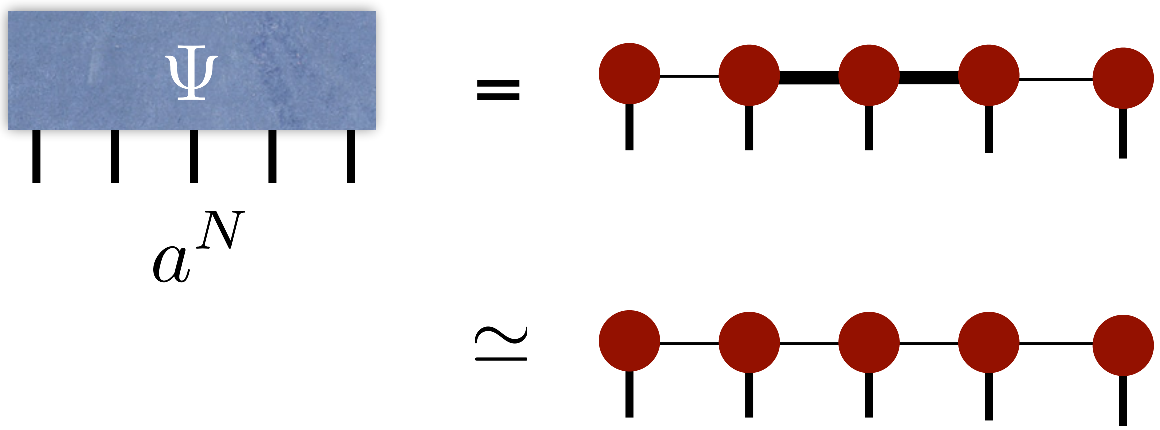


In this construction, the sizes of matrices depend on the position.

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

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

Matrix product state: Low rank approximation



$\boxed{\text{Maximum bond dimension} = a^{N/2}}$

a^N

$\boxed{\text{Maximum bond dimension} = \chi}$

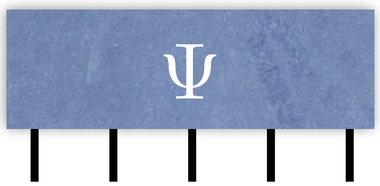
$\propto N a \chi^2$

If the entanglement entropy of the system is **O(1)** (independent of *N*), matrix size "χ" can be small for accurate approximation.

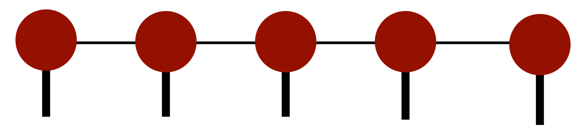
➡ MPS is good for gapped 1d systems.

On the other hand, if the EE increases as increase *N*, "χ" must be increased to keep the same accuracy.

Upper bound of Entanglement entropy

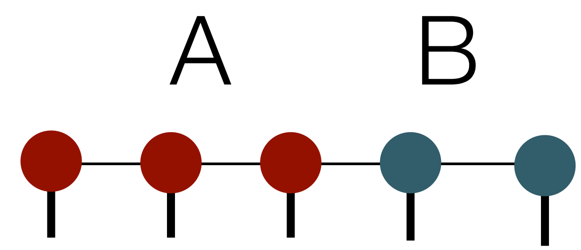


$\simeq$

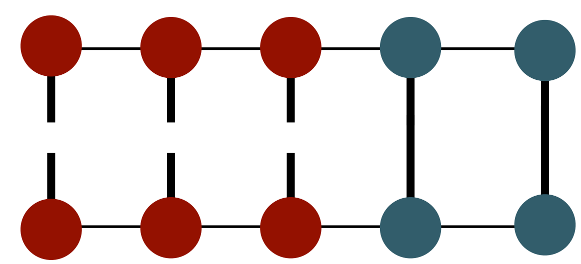


$\equiv |\tilde{\Psi}\rangle$:MPS with χ

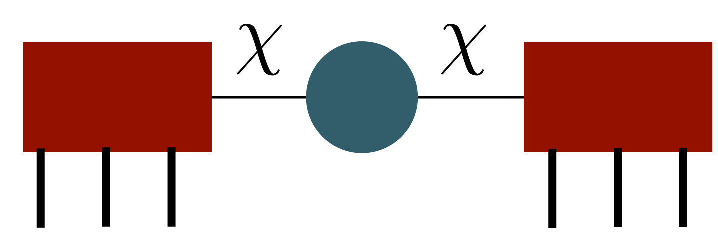
Reduced density matrix of region A:



$$\rho_A = \mathrm{Tr}_B |\tilde{\Psi}\rangle\langle\tilde{\Psi}| =$$



Structure of ρ_A:



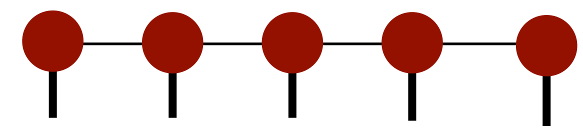
Here we used the properties of SVD 5: "multiplication" explained in #3.

$$\mathrm{rank}\ \rho_A \leq \chi$$

$$S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi$$

Required bond dimension in MPS representation

$$\boxed{S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi}$$



The upper bound is independent of the "length".

length of MPS ⇔ size of the problem

N $\qquad\qquad$ a^N

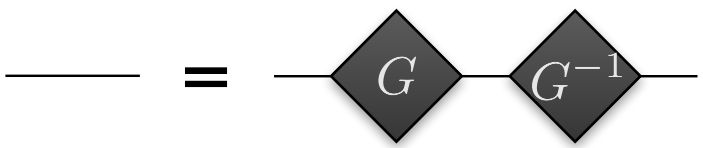
EE of the original vector	Required bond dimension in MPS representation
$S_A = O(1)$	$\chi = O(1)$
$S_A = O(\log N)$	$\chi = O(N^\alpha)$
$S_A = O(N^\alpha)$	$\chi = O(c^{N^\alpha})$

$(\alpha \leq 1)$

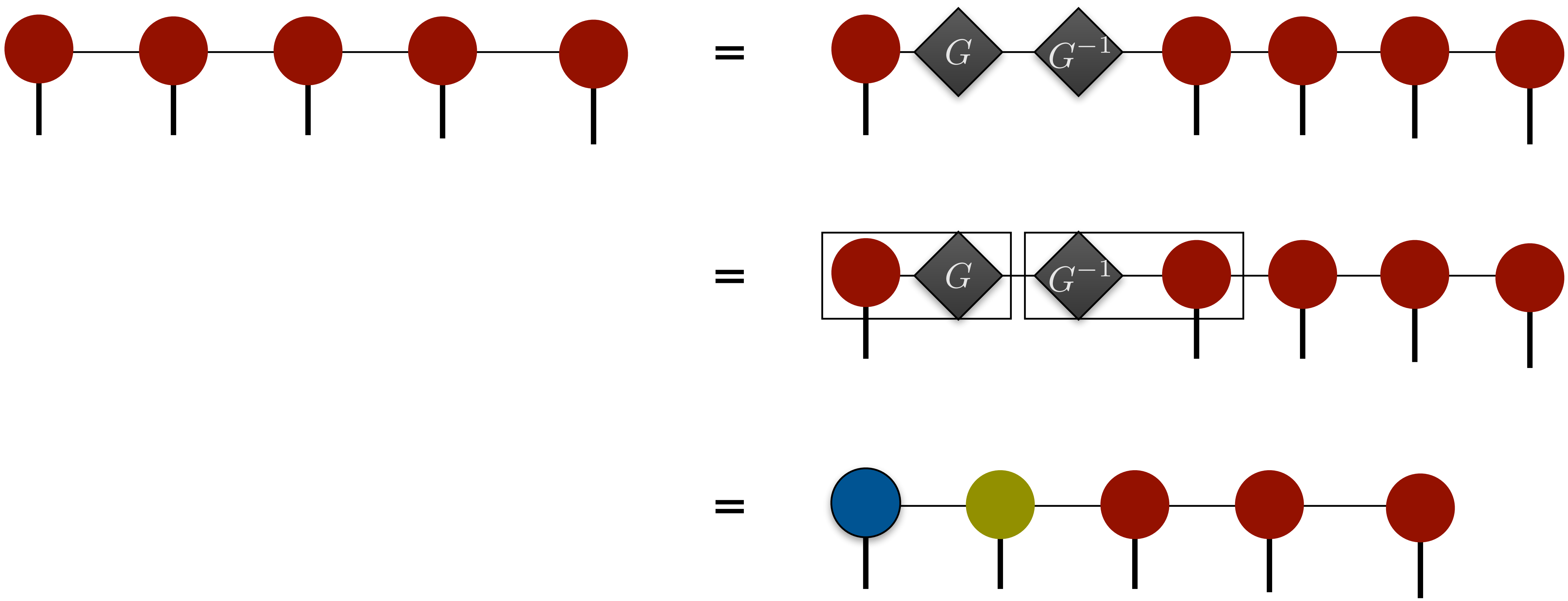
Matrix product states: Canonical form

Gauge redundancy of MPS

MPS is not unique: gauge degree of freedom $I = GG^{-1}$



We can insert a pair of matrices GG^{-1} to MPS

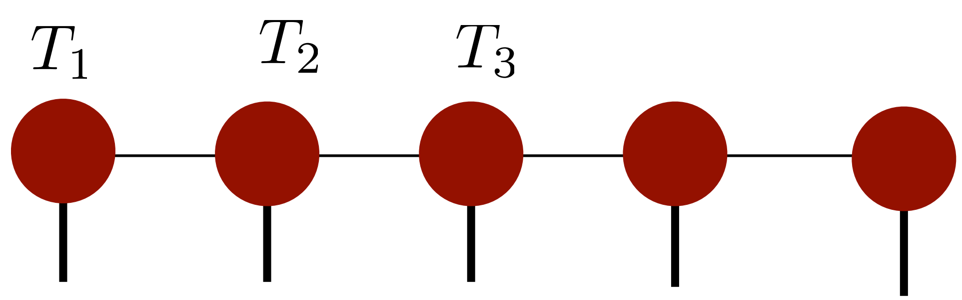


Canonical forms: Left and Right canonical forms

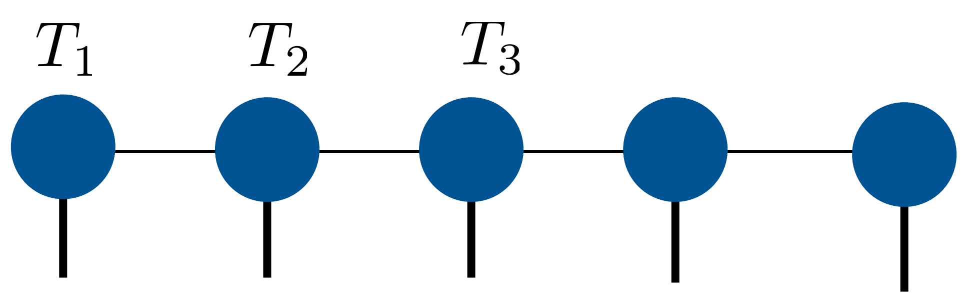
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

"canonical" forms of MPS

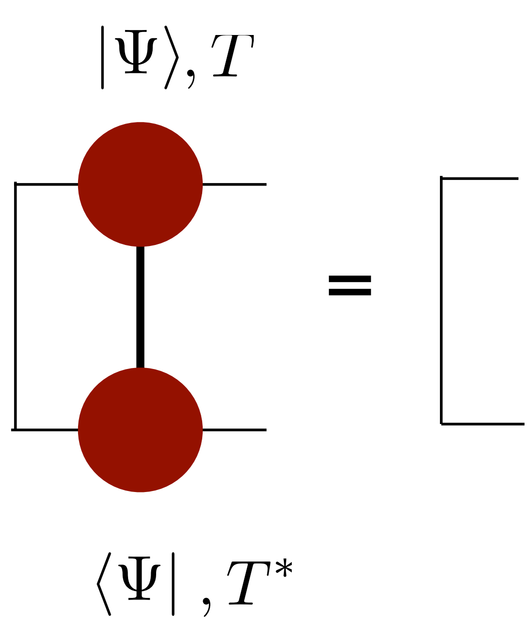
Left canonical form:

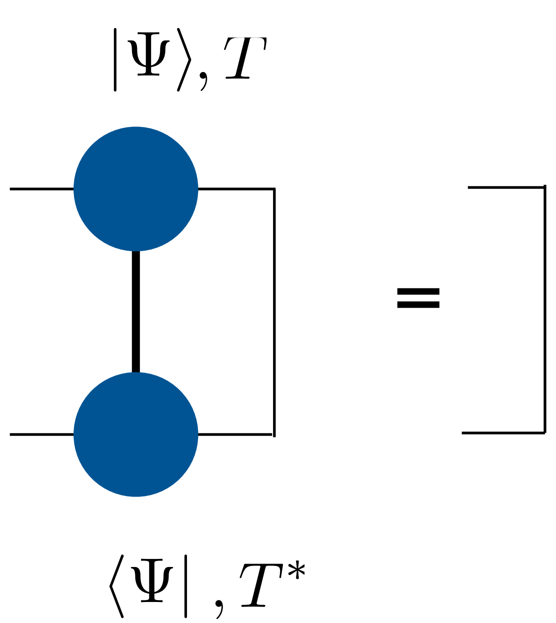


Right canonical form:



Satisfies (at least) left or canonical condition:



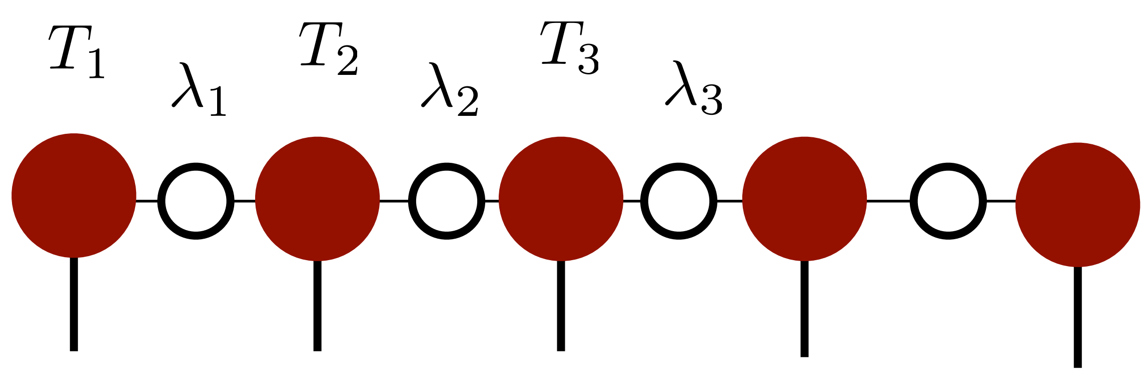


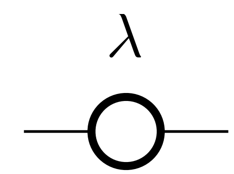
Gauge fix: Canonical form of MPS

Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

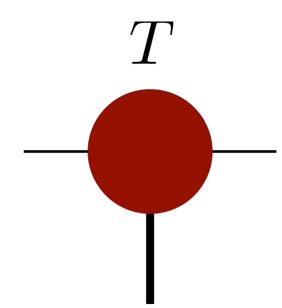
Another canonical form of MPS: (Vidal canonical form)

(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))



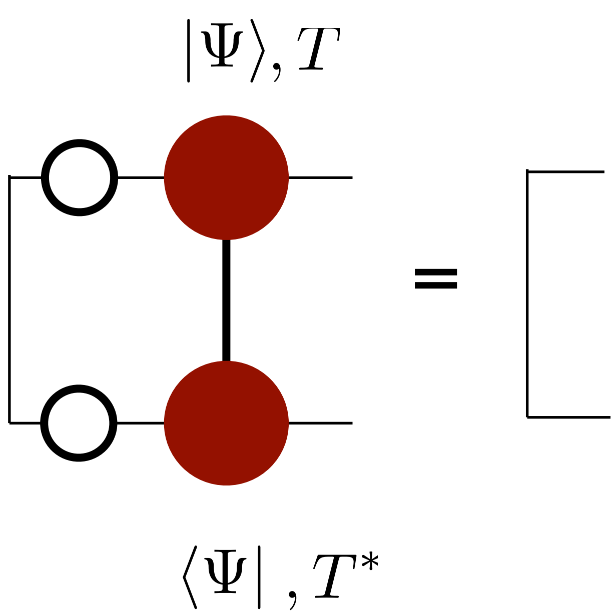


:Diagonal matrix corresponding to Schmidt coefficient

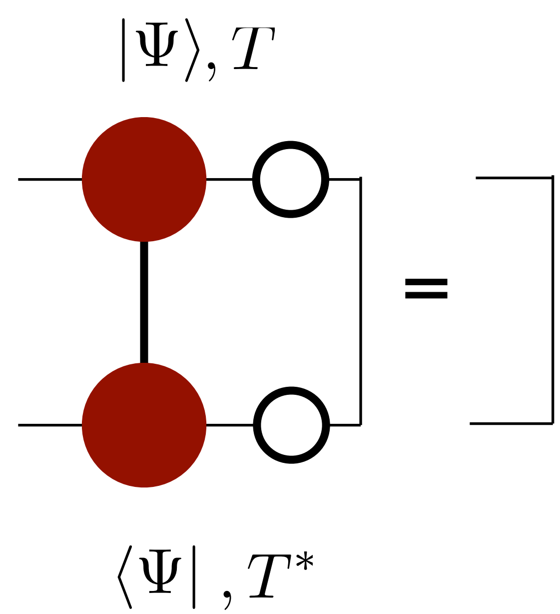


:Virtual indices corresponding to Schmidt basis

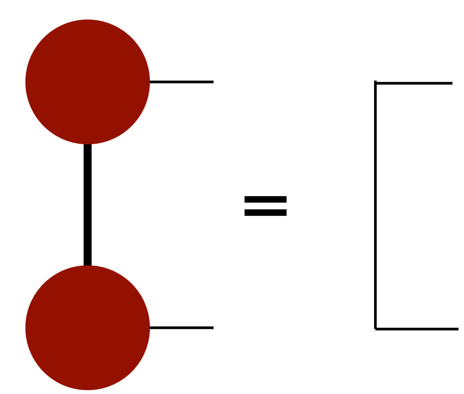
Left canonical condition:



Right canonical condition:



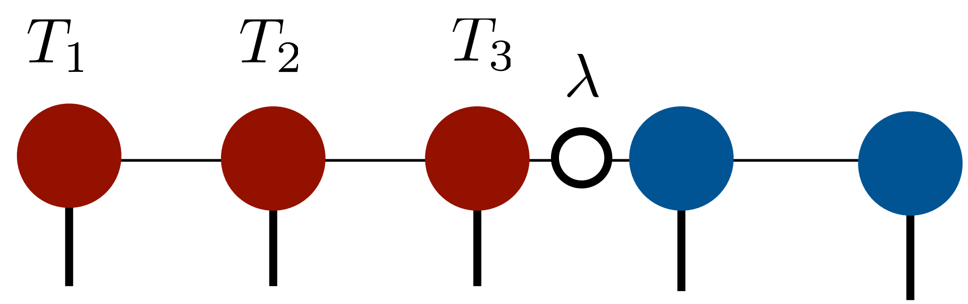
(Boundary)



Canonical forms: Mixed canonical forms

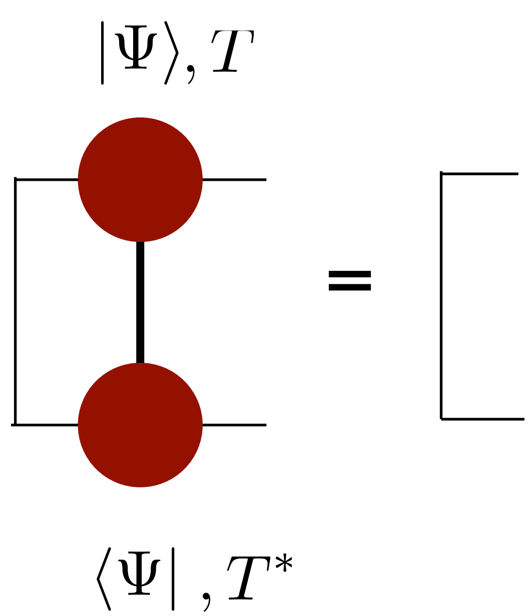
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

Mixed canonical form:

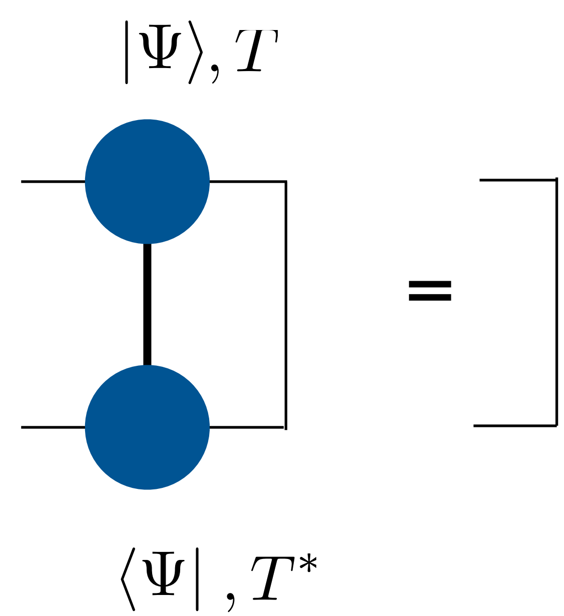


λ is identical with the Schmidt coefficient.

Left canonical condition:



Right canonical condition:

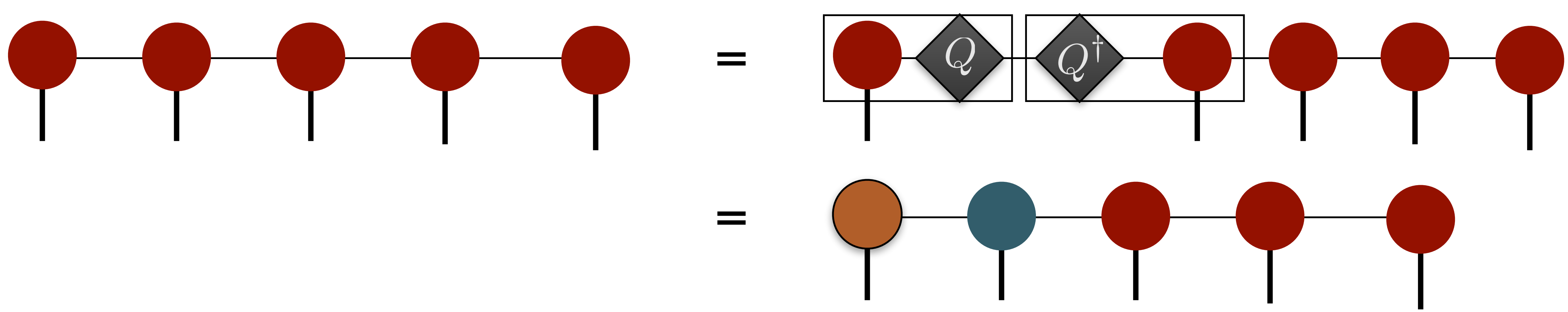


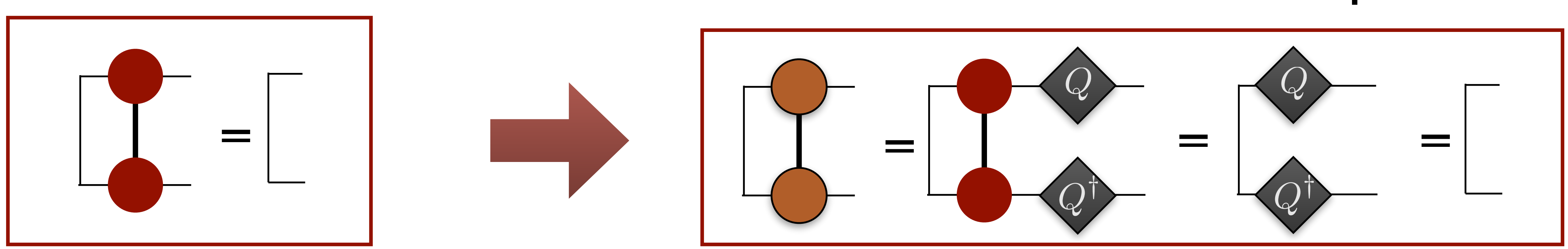
Canonical forms: Note

- Vidal canonical form is unique, up to trivial unitary transformation to virtual indices which keep the same diagonal matrix structure (Schmidt coefficients).

- Left, right and mixed canonical form is "not unique". Under general unitary transformation to virtual indices, it remains to satisfy the canonical condition

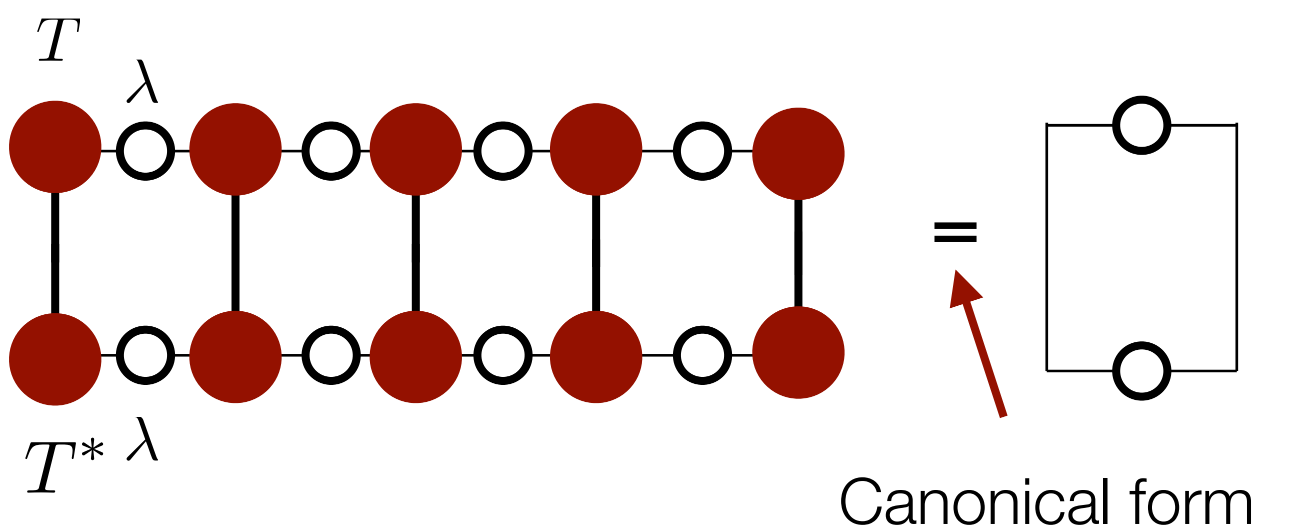
$$QQ^\dagger = Q^\dagger Q = I$$





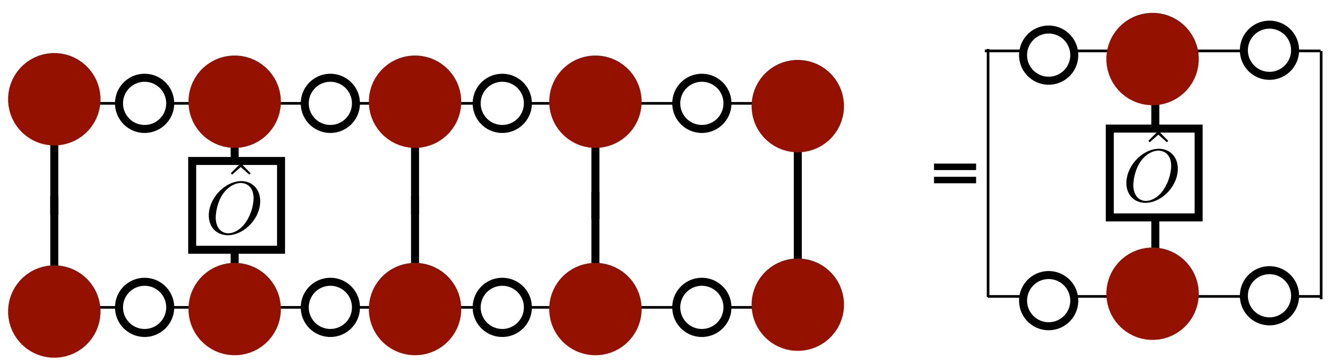
Expectation value with canonical forms

$$\langle \Psi | \Psi \rangle =$$



$$= \sum_i \lambda_i^2 = 1$$

$$\langle \Psi | \hat{O} | \Psi \rangle =$$

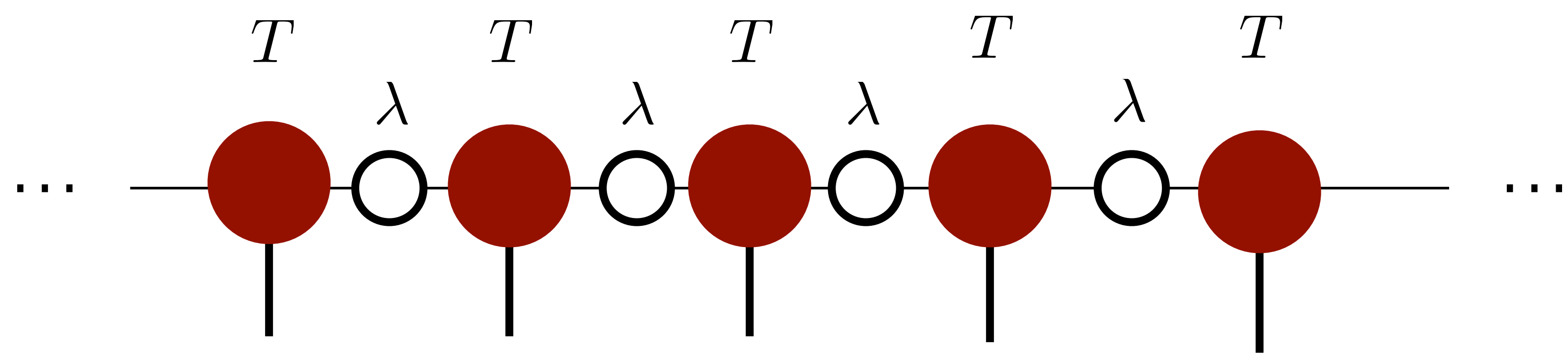


When we consider a mixed canonical form,
we also obtain similar simple diagram. (exercise)

Matrix product states: infinite MPS

MPS for infinite chains

By using MPS, we can write the wave function of a translationally invariant **infinite chain**



Infinite MPS (iMPS) is made by repeating T and λ infinitely.

Translationally invariant system → T and λ are independent of positions!

* Infinite MPS can be accurate when the EE satisfies the 1d area low (S~O(1)).

If the EE increases as increase the system size, we may need infinitely large χ for infinite system.

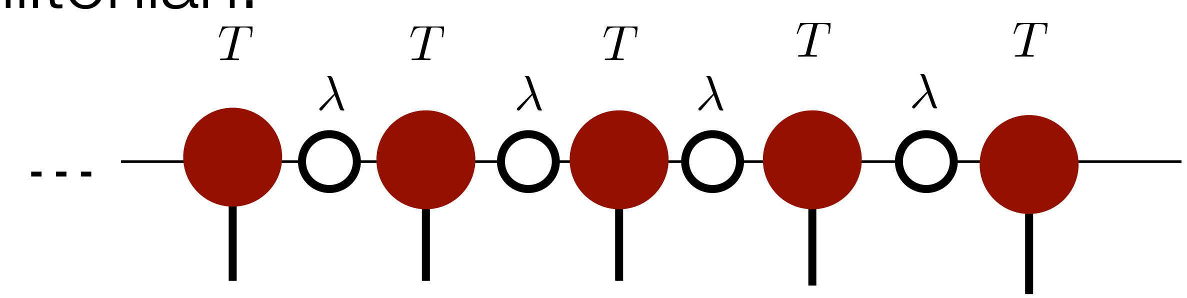
(In practice, we can obtain a reasonable approximation with finite χ.)

Example of iMPS: AKLT state (optional)

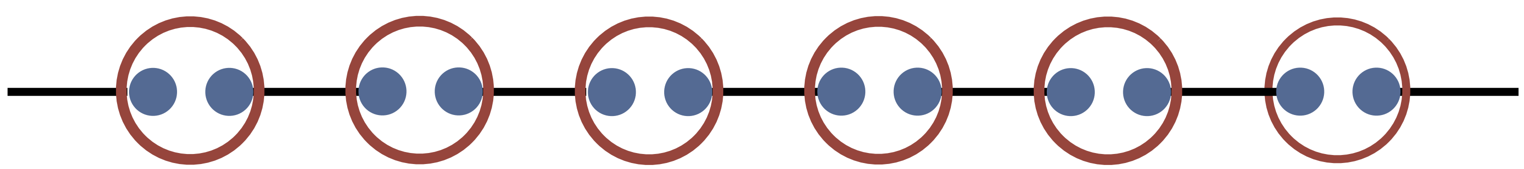
S=1 Affleck-Kennedy-Lieb-Tasaki (AKLT) Hamiltonian:

$$\mathcal{H} = J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + \frac{J}{3} \sum_{\langle i,j \rangle} \left(\vec{S}_i \cdot \vec{S}_j \right)^2$$

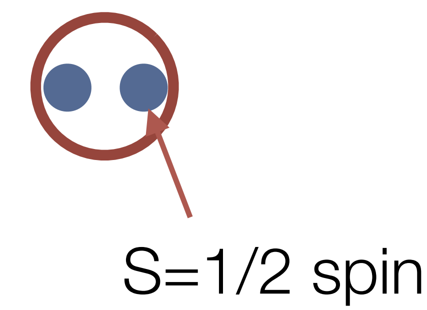
$$(J > 0)$$



The ground state of AKLT model:



S=1 spin:



Spin singlet

χ=2 iMPS: (U. Schollwock, Annals. of Physics **326**, 96 (2011))

$$T[S_z = 1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$T[S_z = 0] = \sqrt{\frac{2}{3}} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad , \lambda = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$T[S_z = -1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}$$

Exercise 2: Make MPS and approximate it

2: Make exact MPS and approximate it by truncating singular values

Try MPS approximation for a random vector, GS of spin model, or a picture image.
Let's see how the approximation efficiency depends on the bond dimensions and vectors.

Sample code: Ex2-1, Ex2-2, Ex2-3. ipynb, or .py

show help: *python Ex2-1.py -h*

These codes correspond to random vector, spin model and picture image, respectively.

I recommend *.ipynb because it contains an appendix part.

*If you run them at Goole Colab, please upload MPS.py in addition to the *.ipynb.

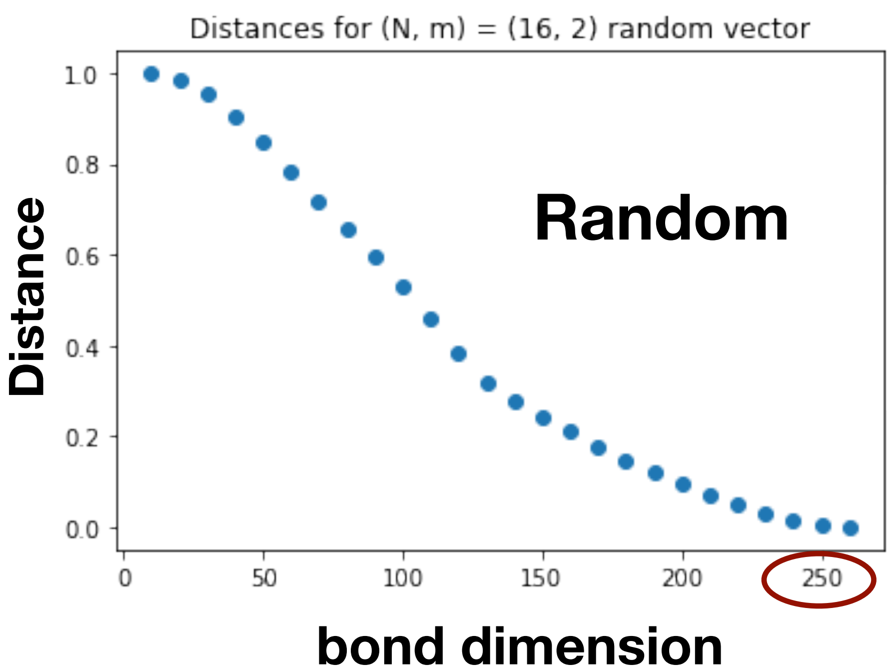
*In the case of Ex2-2 you also need ED.py.

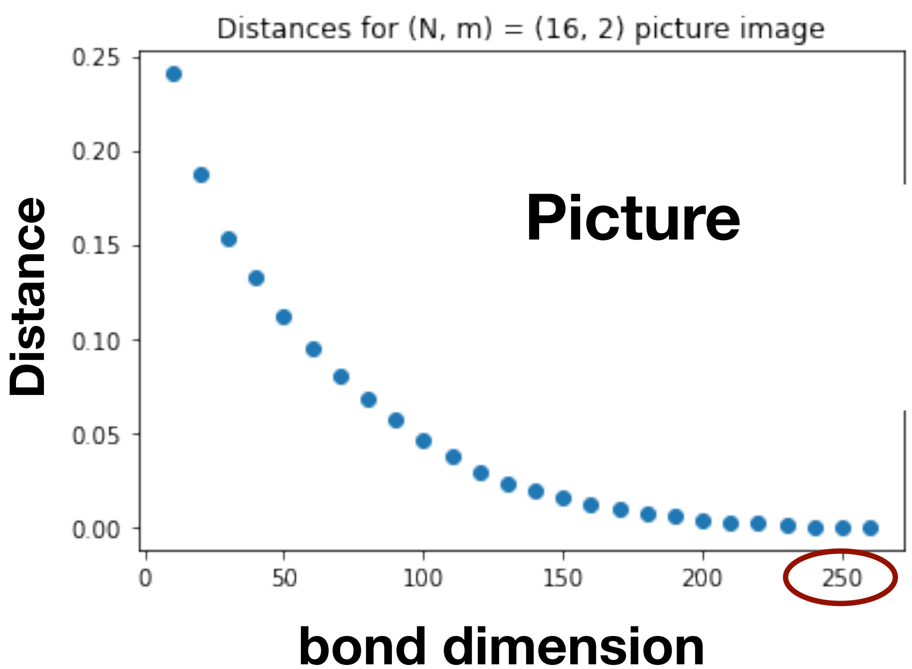
*In the case of Ex2-3 you also need picture file.

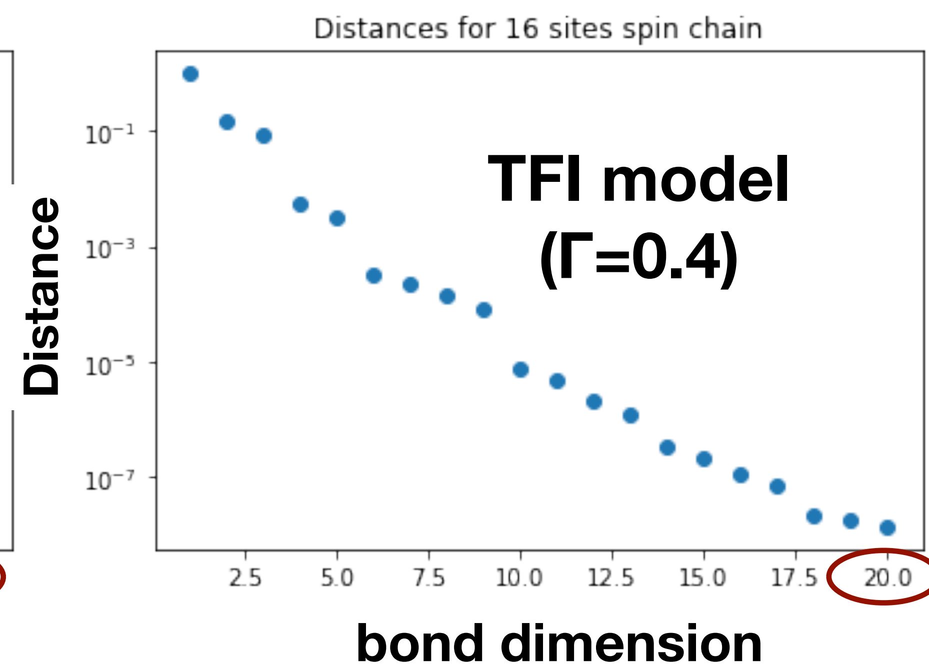
Exercise 2: Make MPS and approximate it

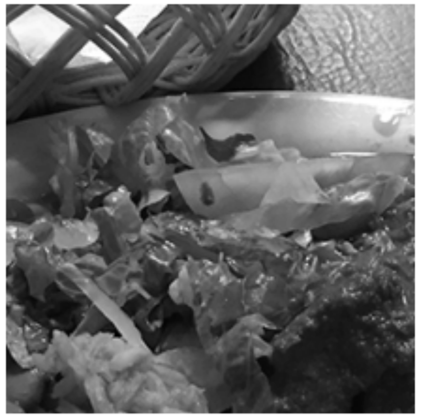
2^{16} dimensional vectors (=16-leg tensors)

Distance between the original and approximated vectors: $||\vec{v}_{ex} - \vec{v}_{ap}||$









$$\mathcal{H} = -\sum_{i=1}^{L-1} S_{i,z} S_{i+1,z} - \Gamma \sum_{i=1}^{L} S_{i,x}$$

Notice

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

Next (Nov. 24)

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