計算科学・量子計算における情報圧縮

Data compression in computational science and quantum computing

2023.01.05

#11: 量子コンピュータのシミュレーション

Simulation of Quantum Computers

Synge Todo / 藤堂眞治

Department of Phys., UTokyo / 東京大学大学院理学系研究科

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

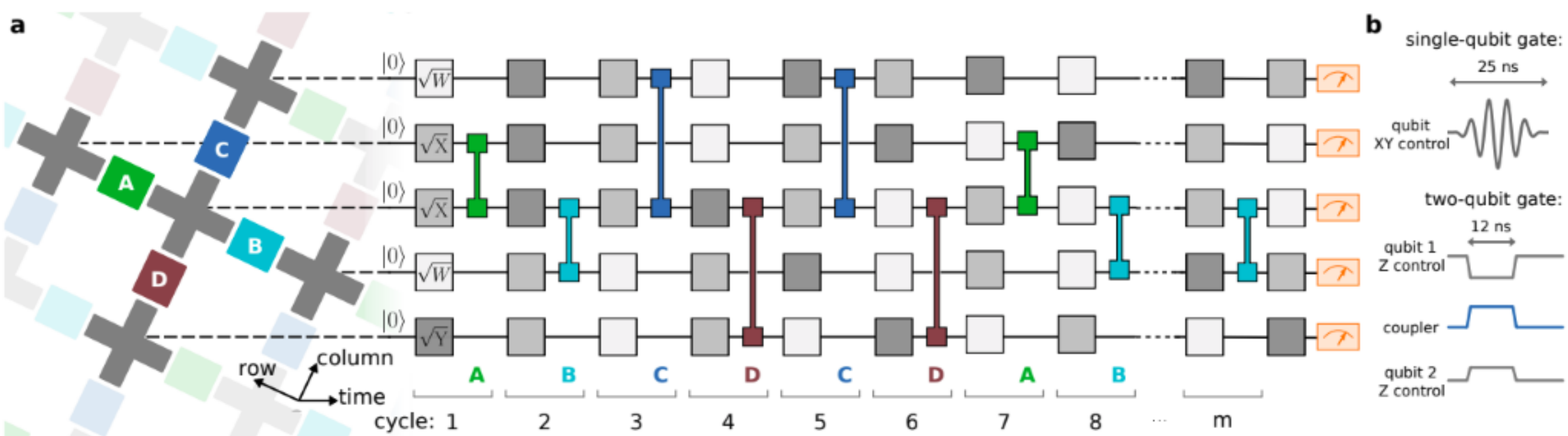
Outline

- Quantum Computation
  - Quantum circuits
  - Quantum algorithms
  - Grover’s algorithm
- Quantum Circuits and Tensor Network
- Simulation Based on TEBD Algorithm
  - Vidal canonical form
  - TEBD algorithm
  - Long-range interaction
  - Sampling states from MPS wave function
- Tensor Decomposition of Quantum Gates
- Another Approach (CATN)

Quantum Computation

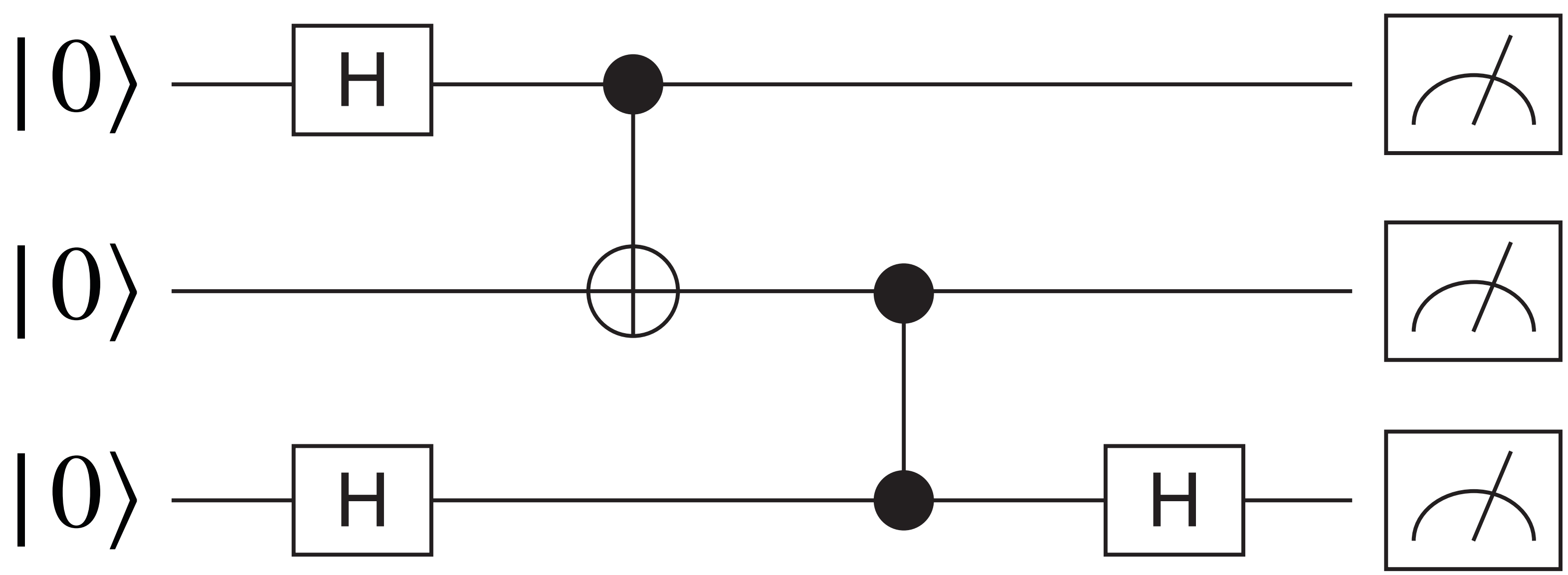
Quantum circuits

- Prepare a set of quantum bits (qubits)
- A number of quantum gates (typically 1-qubit or 2-qubit gates) are applied to qubits in order
  - quantum gates are unitary operations
  - combination of quantum gates are also unitary operation
- Finally, perform measurements to extract information



Arute, F., Arya, K., Babbush, R. et al. Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510 (2019). https://doi.org/10.1038/s41586-019-1666-5

State bifurcation and interference



- $|000\rangle \Rightarrow \frac{1}{2}(|0\rangle + |1\rangle)|0\rangle(|0\rangle + |1\rangle) = \frac{1}{2}(|000\rangle + |001\rangle + |100\rangle + |101\rangle)$

$$\Rightarrow \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle + |111\rangle) \Rightarrow \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle - |111\rangle)$$

$$\Rightarrow \frac{1}{2\sqrt{2}}(|00\rangle + |11\rangle)(|0\rangle + |1\rangle) + (|00\rangle - |11\rangle)(|0\rangle - |1\rangle)$$

$$= \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |110\rangle + |111\rangle + |000\rangle - |001\rangle - |110\rangle + |111\rangle)$$

$$= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Quantum operations in quantum circuits

- Parallelism
  - inputting a state of superposition produces a superposition of the corresponding outputs.
- Bifurcation
  - states bifurcate (in terms of computational basis) when H-gates, etc. are applied
- Interference
  - superposition coefficients of the states are complex, and they may cancel each other out and vanish
- Collapse
  - collapse to one of the states in the basis used for the measurement

Essence of quantum algorithm

- Prepare a superposition of many states, using Hadamard gates, etc.
  - if non-selective projective measurements is performed at this stage, the entropy is extensive (proportional to the number of qubits)
- Manipulate and interfere the states so that desired answer have large amplitude
  - entropy of the quantum state remains zero
    - interference reduces the entropy after non-selective projective measurement
    - a kind of data compression?
- Selective projective measurement (measure and see the result) to get the correct answer with high probability
  - sampling from the state with entropy as small as possible
    - a kind of information extraction?

Grover’s Algorithm

- Quantum algorithm for unstructured search
  - finding an entry that obey a certain condition from N unsorted data sequence
  - assume that there is only one entry that obeys the condition
  - classical algorithm
    - average cost: $N/2$
  - Grover’s algotithm
    - average cost: $O(\sqrt{N})$
- Problem setup
  - n-bit binary numbers from 0 to $2^n - 1 \quad (n = \log_2 N)$
  - suppose a function $f(x)$ returns 1 only when $x = \omega$, 0 for otherwise
  - we want to find ω

Grover's Algorithm (1996)

- Prepare a unitary operator (or "oracle")

$$\begin{cases} U_\omega|x\rangle = -|x\rangle & \text{for } x = \omega \\ U_\omega|x\rangle = |x\rangle & \text{otherwise} \end{cases}$$

- Grover's algorithm
  - prepare uniform superposition of all states

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = H^{\otimes n}|0\rangle_n$$

  - apply oracle U_ω and Grover diffusion operator U_s $r(N)$ times

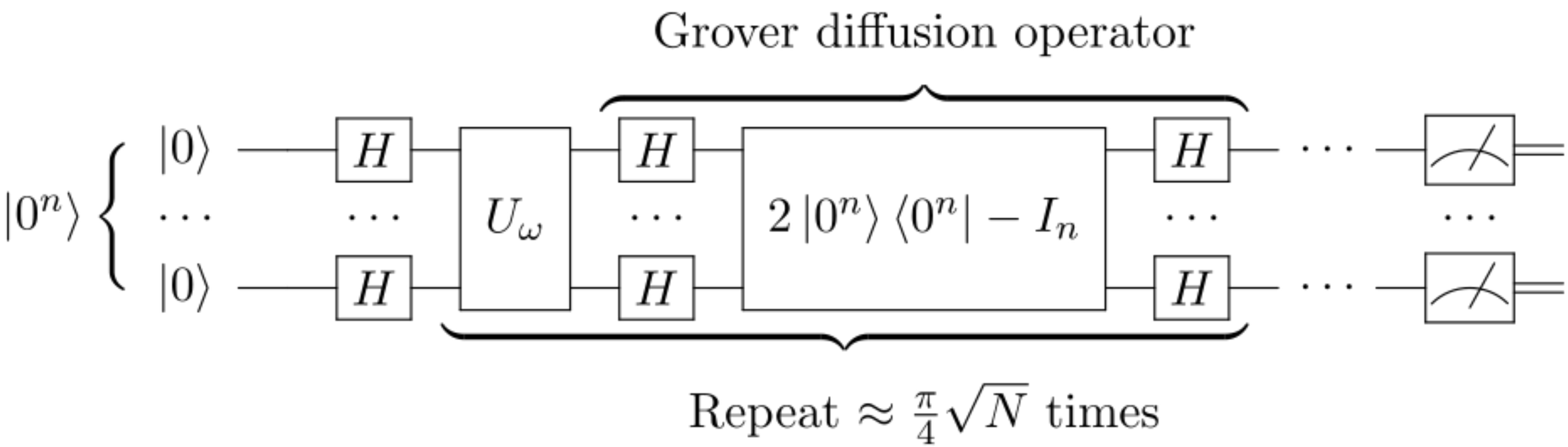
$$(U_s U_\omega)^{r(N)}|s\rangle$$

$$U_s = 2|s\rangle\langle s| - 1$$

$$r(N) = \frac{\pi}{4}\sqrt{N}$$

  - measure the resulting state in the computational basis
  - measurement gives ω with high probability O(1)

Grover’s Algorithm



https://en.wikipedia.org/wiki/Grover%27s_algorithm

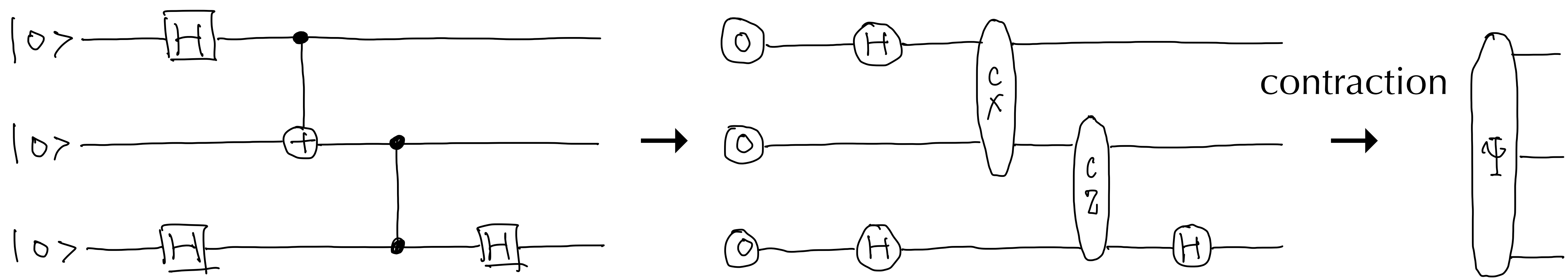
Quantum Circuits and Tensor Network

Tensor Network Representation of Quantum Circuits

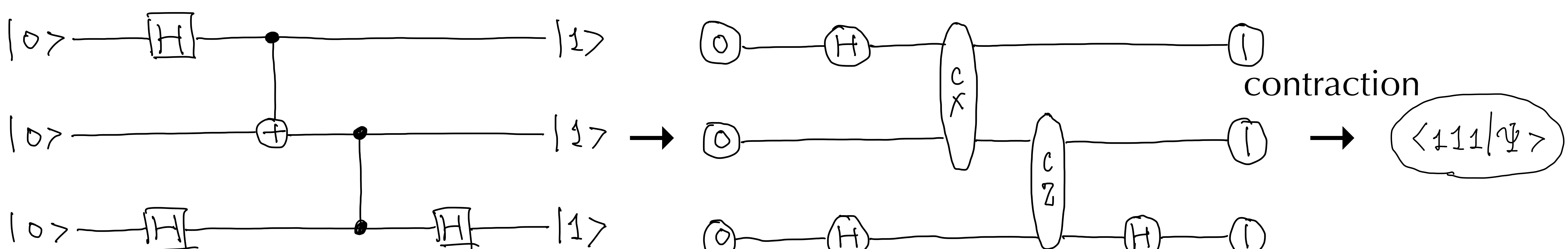
- Tensor representation of quantum gates and states
  - 1-qubit gate:
    - 2×2 matrix → 2-leg tensor
  - 2-qubit gate:
    - 4×4 matrix → $2 \times 2 \times 2 \times 2$ tensor → 4-leg tensor
  - r-qubit gate:
    - $2^r \times 2^r$ matrix → $2 \times 2 \times \cdots \times 2$ tensor → $2r$-leg tensor
  - initial quantum states
    - product state, e.g., $|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle$
      → a set of n 1-leg tensors (vectors)

Tensor Network Representation

- Tensor network representation of a quantum circuit
  - all bond dimensions are 2

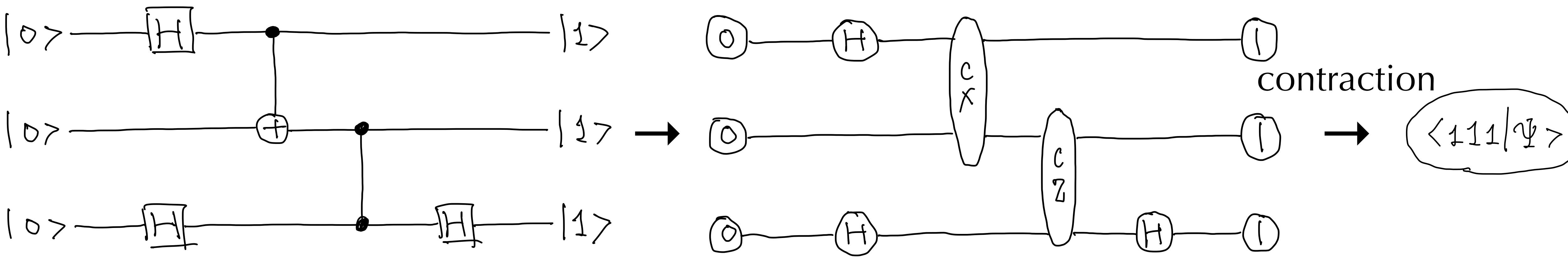


- Taking contraction from the initial state (left to right)
  - equivalent to Schrödinger simulation



Evaluating an Amplitude

- Instead of obtaining wave function itself, let's evaluate an amplitude (a coefficient) in the wave function: e.g., $\langle 111|\Psi\rangle$

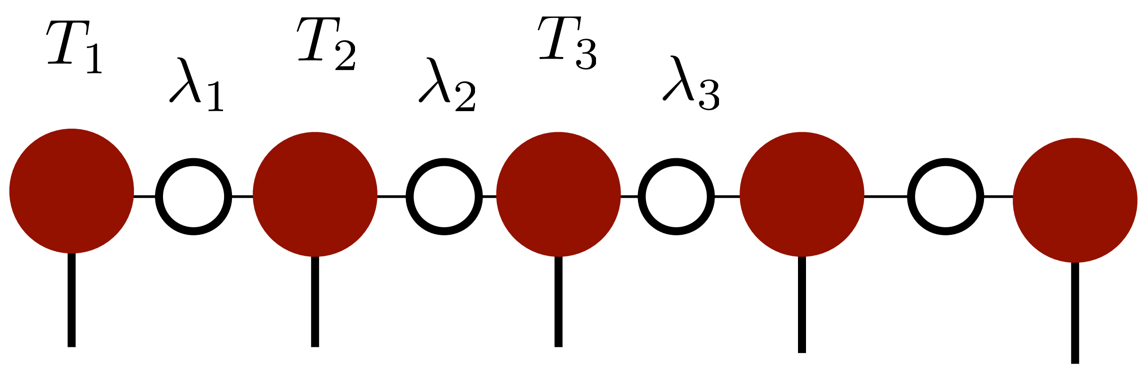


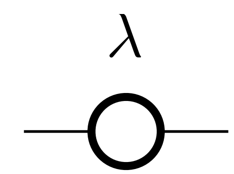
- Order of contraction does not change the final result
- Freedom in contraction order → possibility to reduce the cost

Simulation Based on TEBD Algorithm

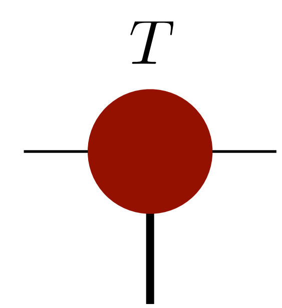
Vidal canonical form

- Another canonical form of MPS (Vidal canonical form)



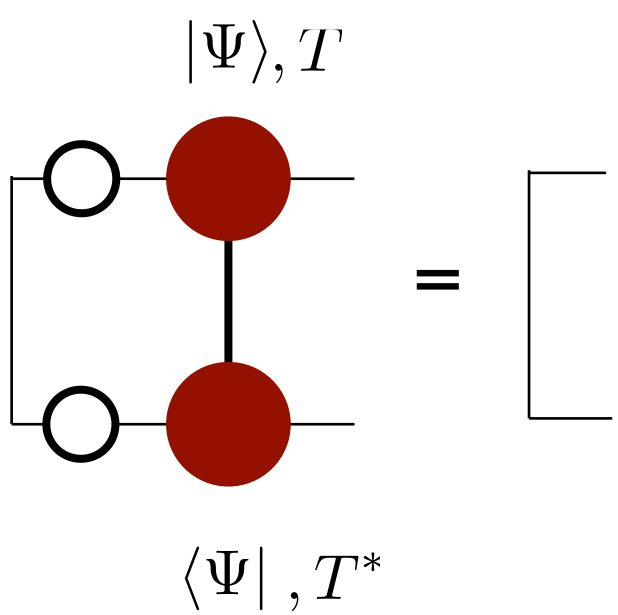


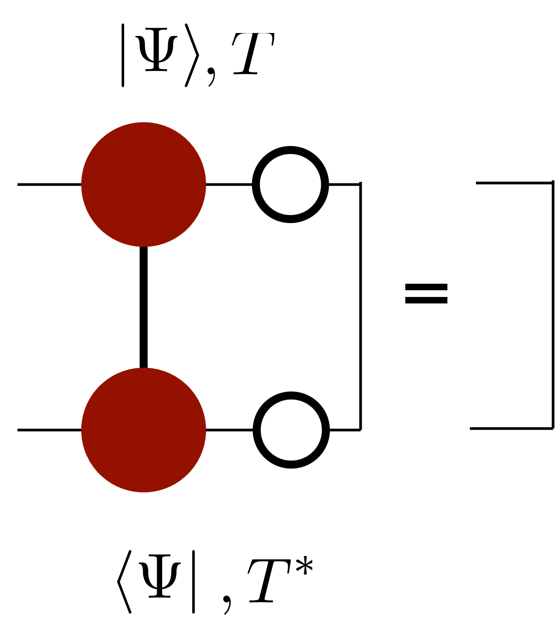
diagonal matrix corresponding to Schmidt coefficient

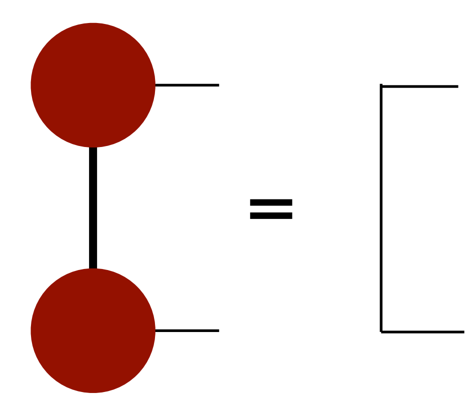


Virtual indices corresponding to Schmidt basis

- Left / right / boundary canonical conditions





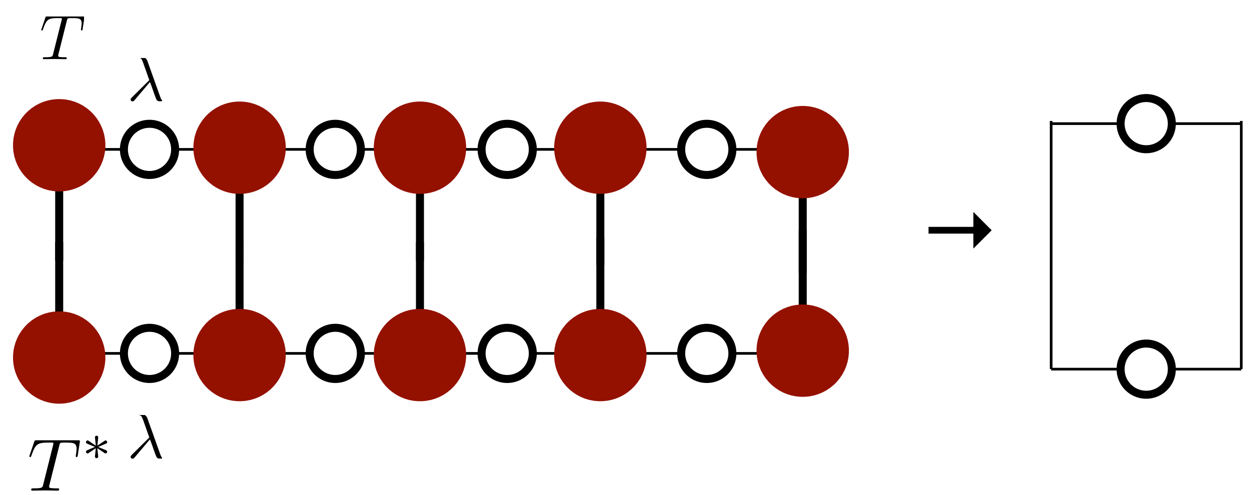


G. Vidal, Phys. Rev. Lett. 91, 147902 (2003)

Expectation values of MPS with canonical form

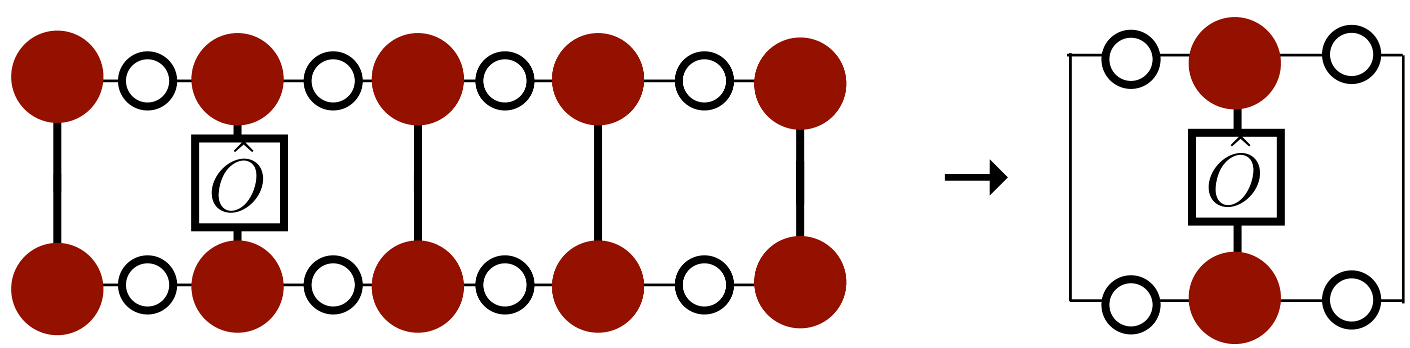
- Evaluation of expectation values becomes extremely simple under canonical condition

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



$= \sum_i \lambda_i^2 = 1$

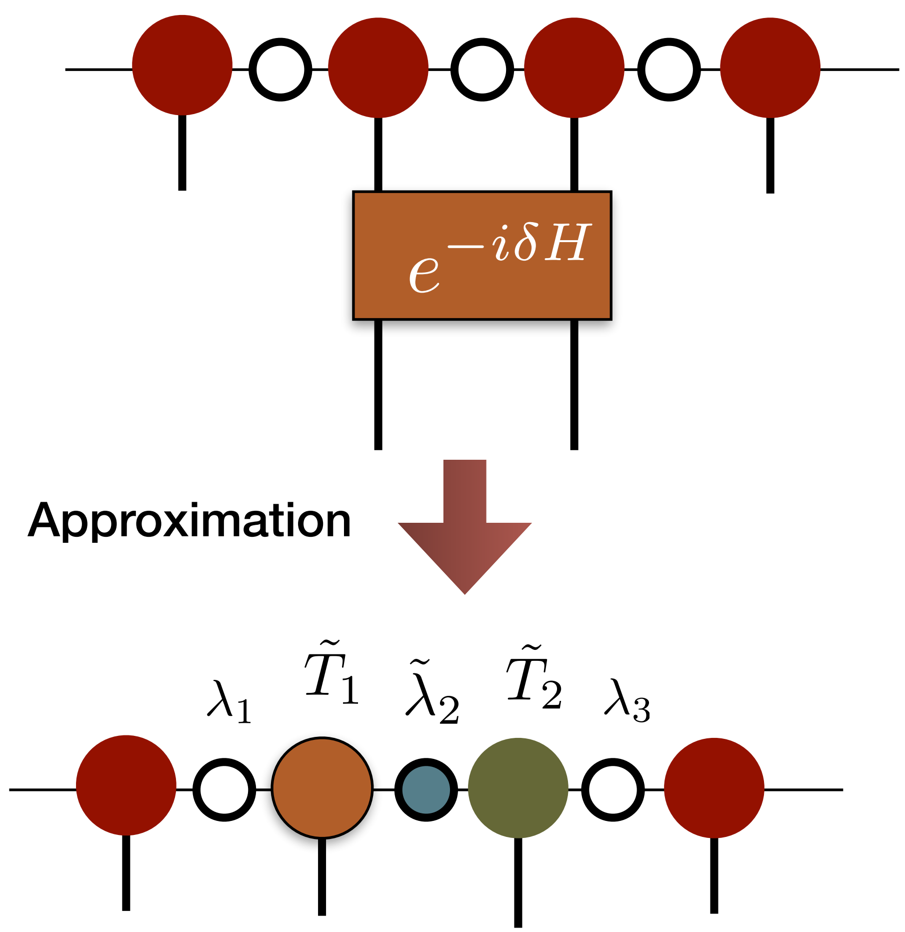
$\langle\Psi|\hat{O}|\Psi\rangle =$



- similar simple diagram can also be obtained with mixed canonical form

TEBD algorithm

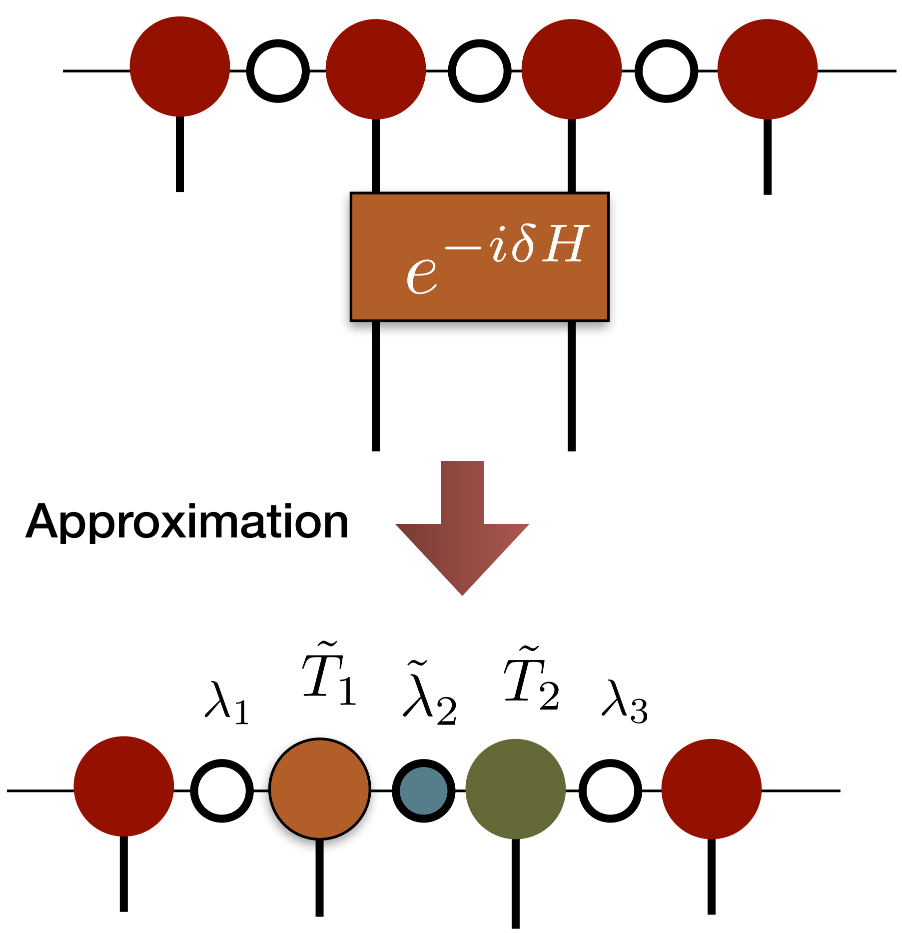
- Time Evolving Block Decimation (TEBD)
  - perform accurate transformation locally by using canonical MPS

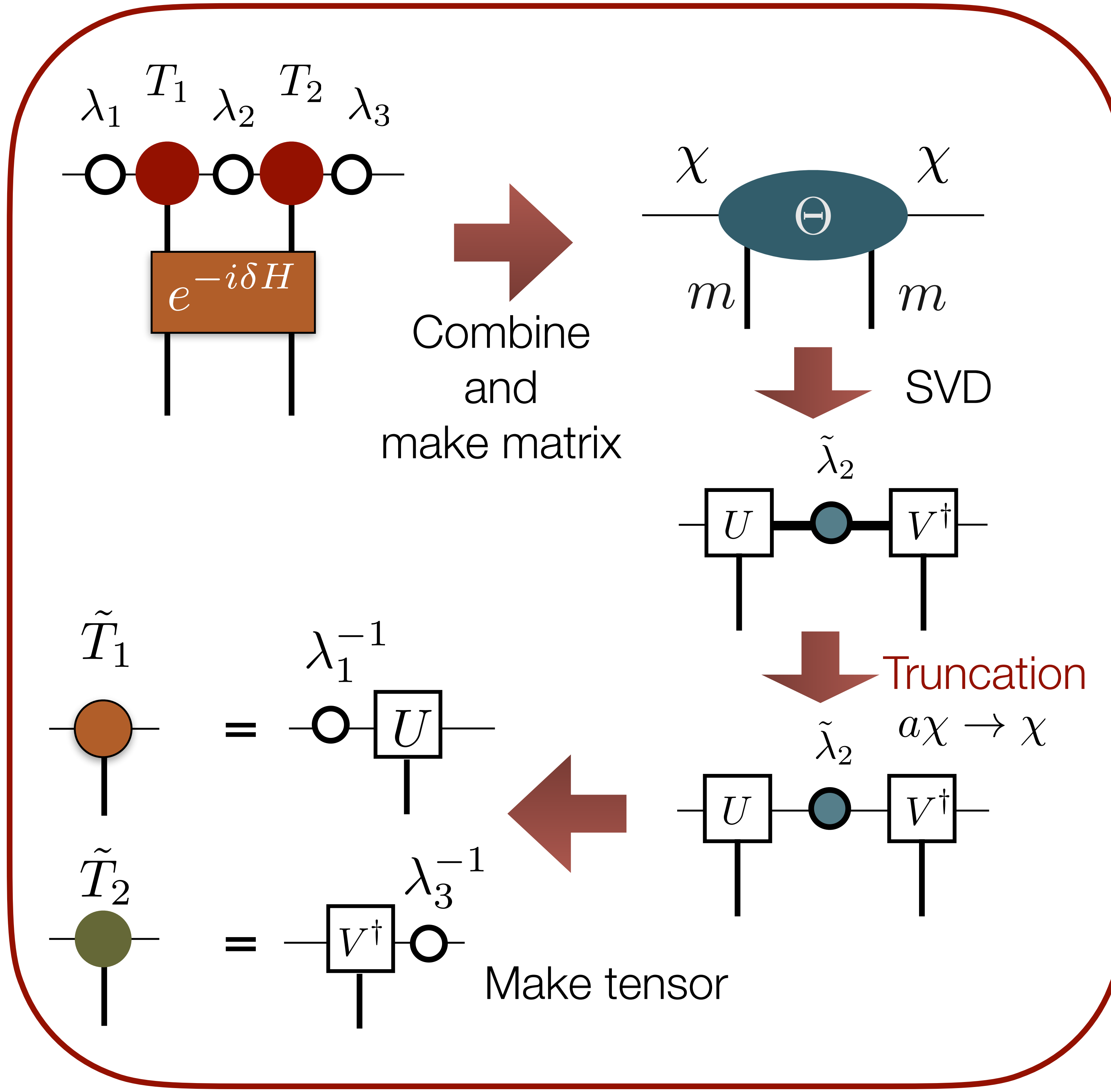


  - only two matrices that are directly applied TE operator change

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

TEBD algorithm

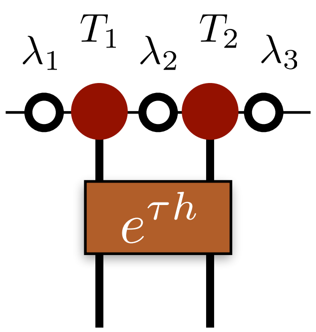




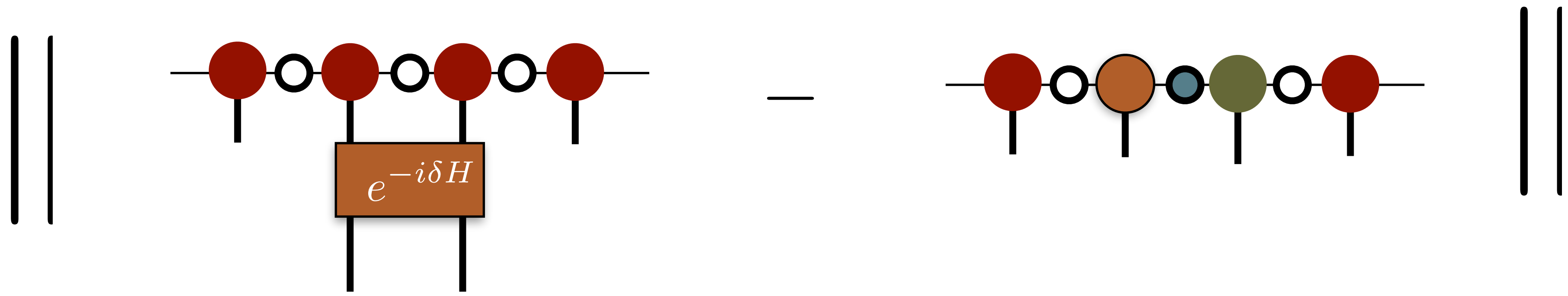
G. Vidal, Phys. Rev. Lett. 91, 147902 (2003)

Why TEBD is accurate?

- For accurate calculation, the canonical form is important
  - If λ is equal to the Schmidt coefficient, it contains all information of the remaining part of the system

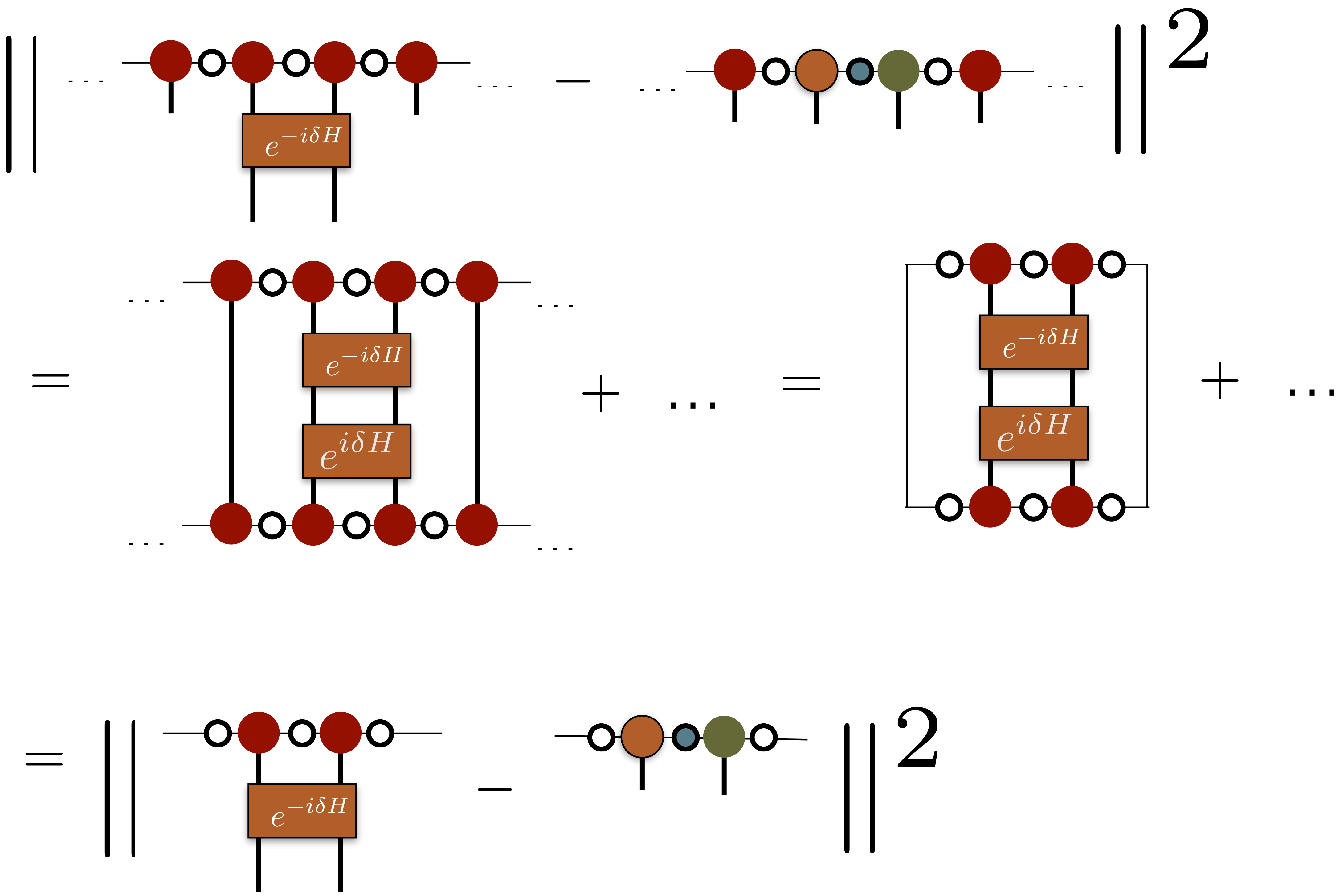


  - due to the canonical form, we can prove that TEBD algorithm minimize the distance of two quantum states:



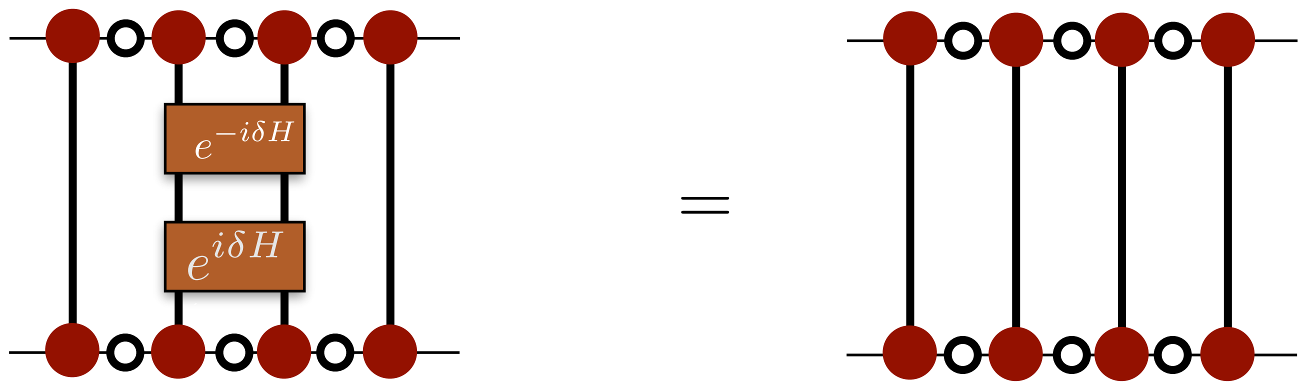
  - truncation based on local SVD can be globally optimal, even if we look at a part of the MPS

Distance between MPS's



Why TEBD is accurate?

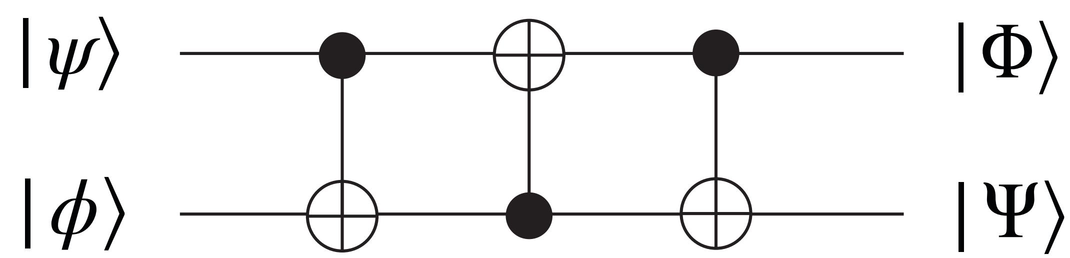
- If the operator is unitary, MPS keeps canonical form even after approximation



  - Unitary operator does not affect to the other Schmidt coefficients
  - If we choose the initial MPS as the canonical form, TEBD algorithm almost keeps it
  - (So, TEBD is almost “globally optimal”)

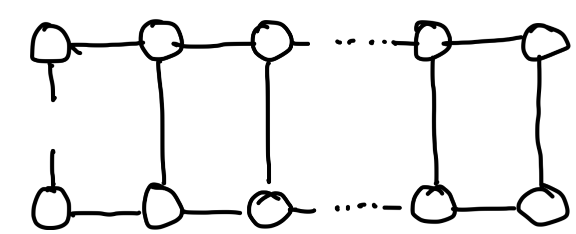
Long-range interaction

- How to apply operations between distant qubits
  - converted into nearest-neighbor operation by using swap gates

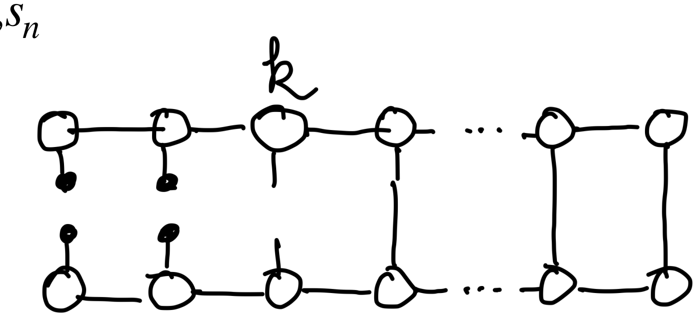


Sampling states from MPS wave function

- Sampling in 2^n-dimensional space is difficult
  - $P(s_1, s_2, \cdots, s_n) = |\langle s_1 s_2 \cdots s_n | \Psi \rangle|^2 / \langle \Psi | \Psi \rangle$
  - tower sampling: memory cost $\sim O(2^n)$, computation cost $\sim O(2^n)$
- Once (approximate) MPS representation of wave function is obtained
  - it is straightforward to sample a state by using
  - marginal distribution: $P(s_1) = \sum_{s_2, \cdots, s_n} P(s_1, s_2, \cdots, s_n)$



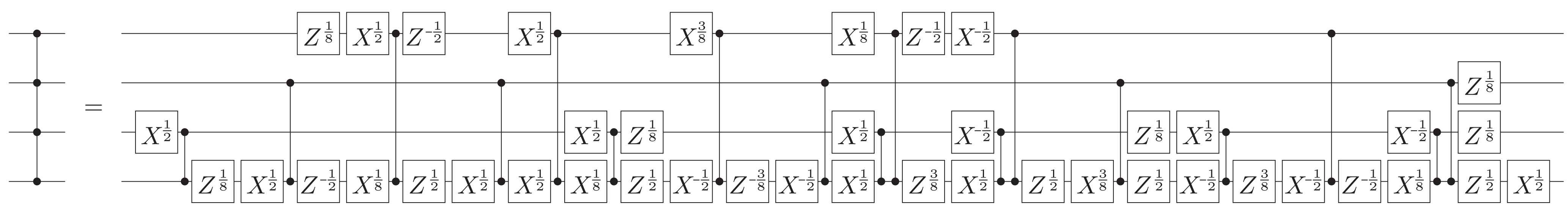
  - conditional distribution: $P(s_k | s_1, \cdots, s_{k-1}) = \sum_{s_{k+1}, \cdots, s_n} P(s_1, s_2, \cdots, s_n)$



Tensor Decomposition of Quantum Gates

Efficient representation of quantum circuit

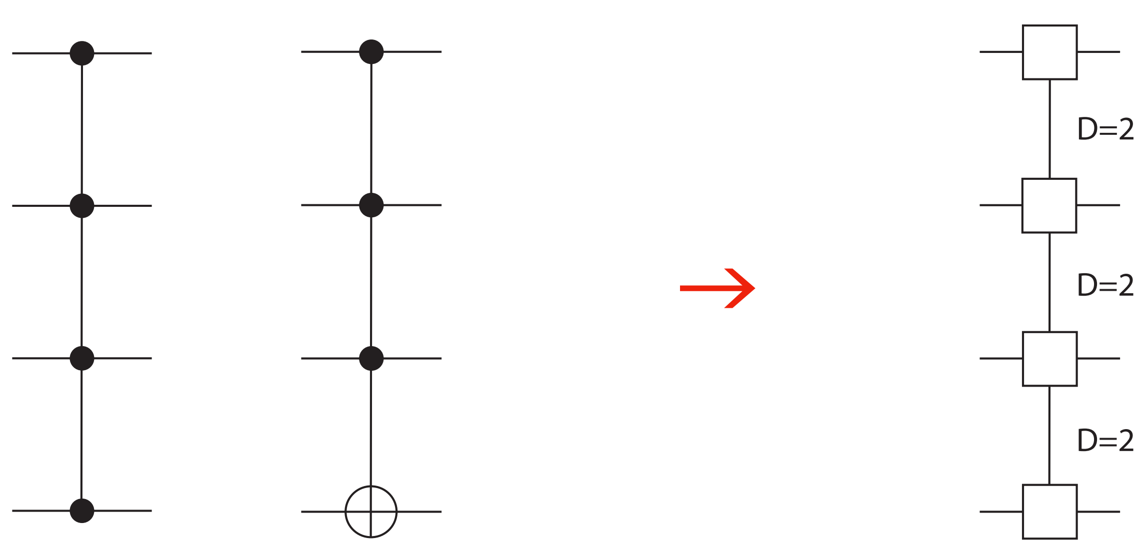
- Operation with multiple control bits (CCZ, CCCZ, ...)



K. M. Nakanishi, T. Satoh, S. Todo, arXiv:2109.13223

  - finding efficient decomposition (into 1-bit, 2-bit operations) is highly nontrivial

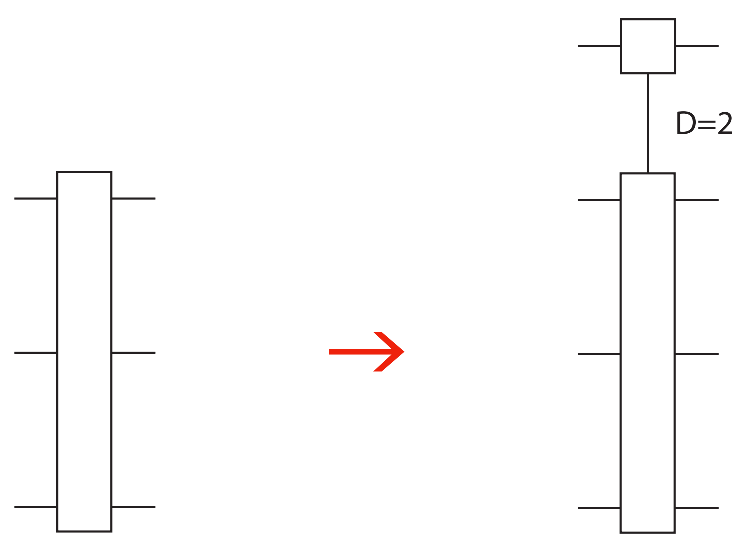
- Tensor decomposition of controlled-operations is trivial
  - operation with n-bit control bits → matrix-product operator with $D = 2$



"If" clause in quantum computation

$$U \quad \rightarrow \quad |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

- No-go theorem
  - universal controllization of arbitrary (unknown) unitary operation can not be implemented by quantum circuits Araujo et al (2014)
- Controllization of tensor operators
  - can be achieved just by adding a 2x2x2 tensor to the network



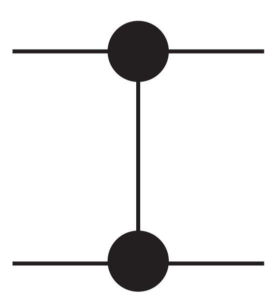
- (Multi-controlled operations are not a good building block of quantum algorithms?)

Decomposition to MPO

- Decomposition is not unique due to gauge degrees of freedom

$$AB = A(MM^{-1})B = (AM)(M^{-1}B) = A'B'$$

- Rank of the matrix depends on how the tensor is represented in the matrix form
  - two different matrix representations of CZ gate



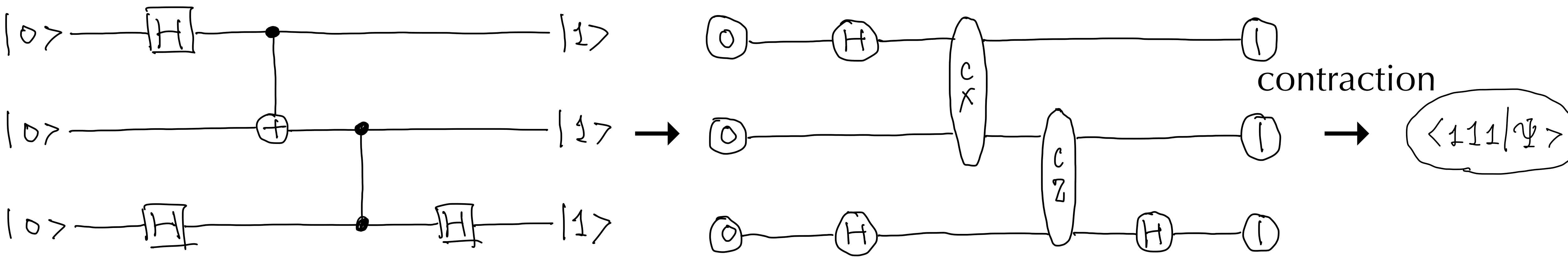
$$U^{\mathrm{CZ}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$M^{\mathrm{CZ}} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

Another Approach (CATN)

Evaluating an Amplitude

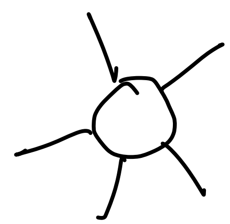
- Instead of obtaining wave function itself, let’s evaluate an amplitude (a coefficient) in the wave function: e.g., $\langle 111 | \Psi \rangle$



- Order of contraction does not change the final result
- Freedom in contraction order → possibility to reduce the cost

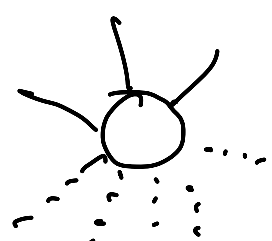
Memory cost of tensors

- Assuming all bond dimensions are χ
  - 5-leg tensor



T_{ijkln} $O(\chi^5)$

  - n-leg tensor

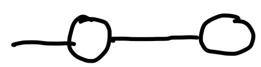


$O(\chi^n)$

- Memory cost grows exponentially as the number of legs increases

Computational Cost of Tensor Contraction

- Matrix-vector multiplication

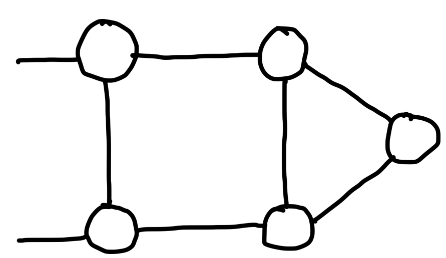


$$w_i = \sum_j T_{ij} v_j \qquad O(\chi^2)$$

- Matrix-matrix multiplication

$$A_{ij} = \sum_k T_{ik} R_{kj} \qquad O(\chi^3)$$

- Contraction of tensors



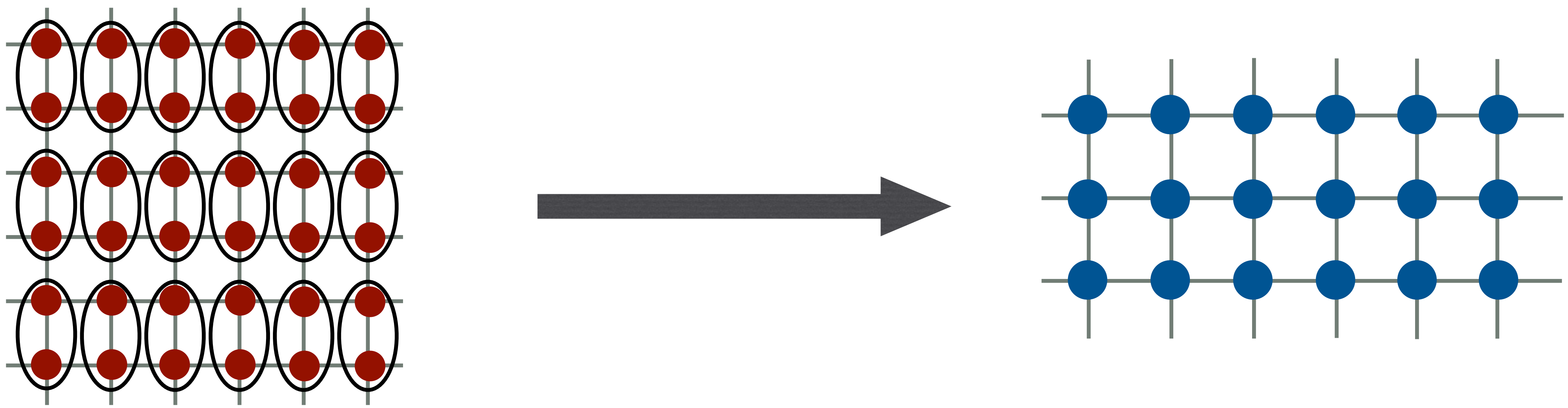
$$O(\chi^?)$$

- Cost of tensor contraction strongly depends of contraction order
- In general, computational cost increases exponentially as the number of legs increases

slide from Lecture #9

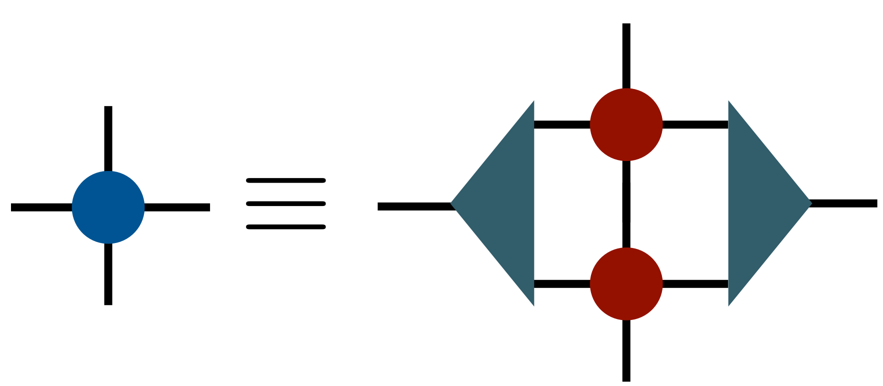
Central idea of Anisotropic TRG

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

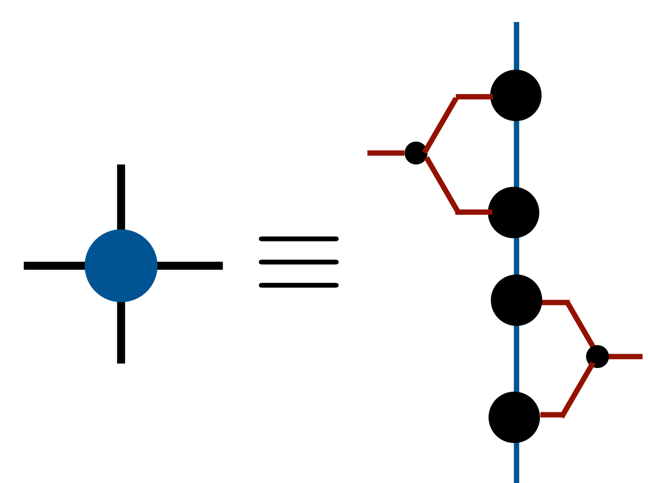


In order to reduce the computation cost, we decompose the local tensor into small pieces before performing coarse-graining.

HOTRG $O(D^7)$



ATRG $O(D^5)$

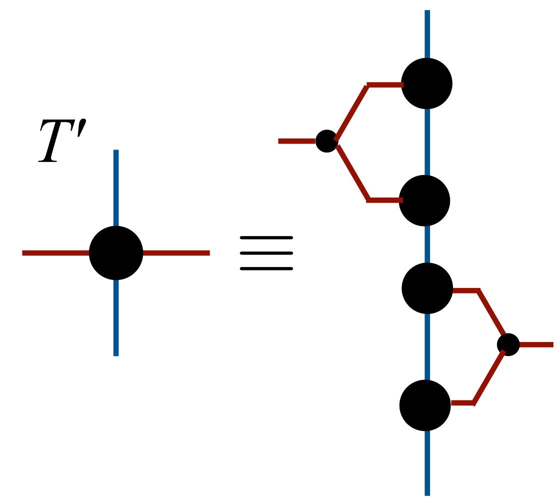


slide from Lecture #9

Summary of 2D ATRG

Memory storage: $O(D^3)$

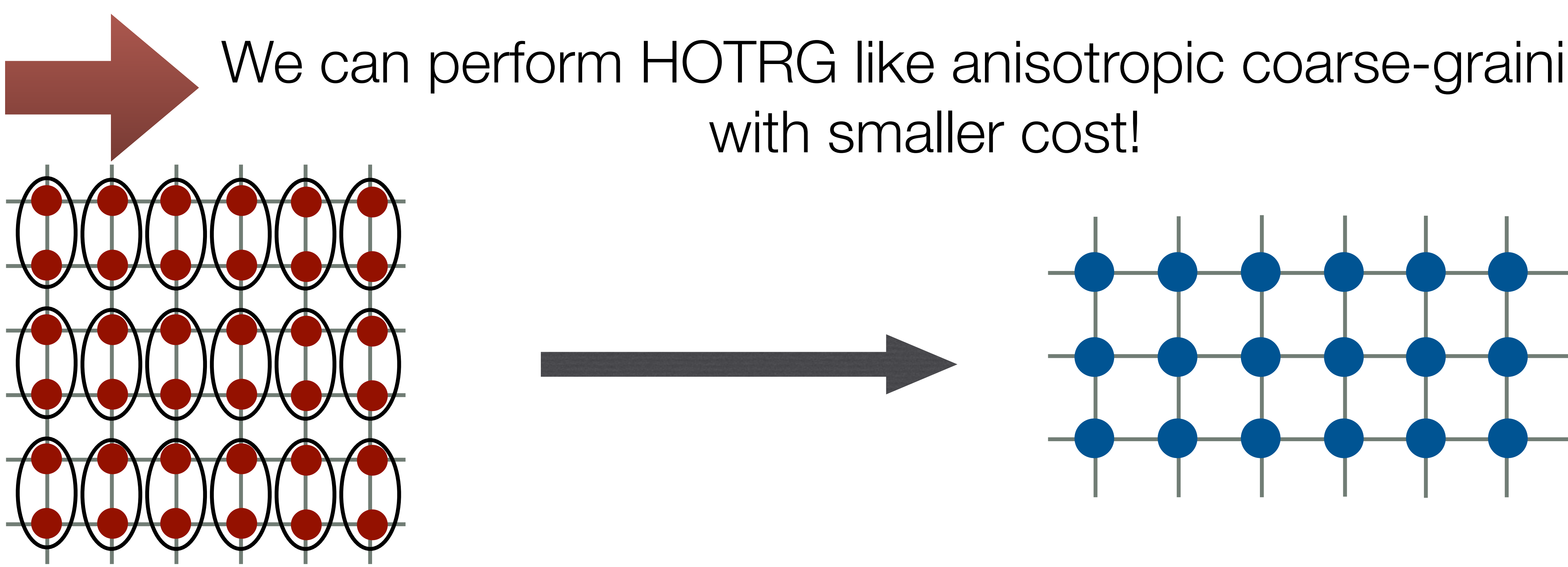
* We do not explicitly create 4-leg tensor.
(We need only 3-leg tensors!)



Computation time: $O(D^5)$

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

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



slide from Lecture #9

Benchmarks

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

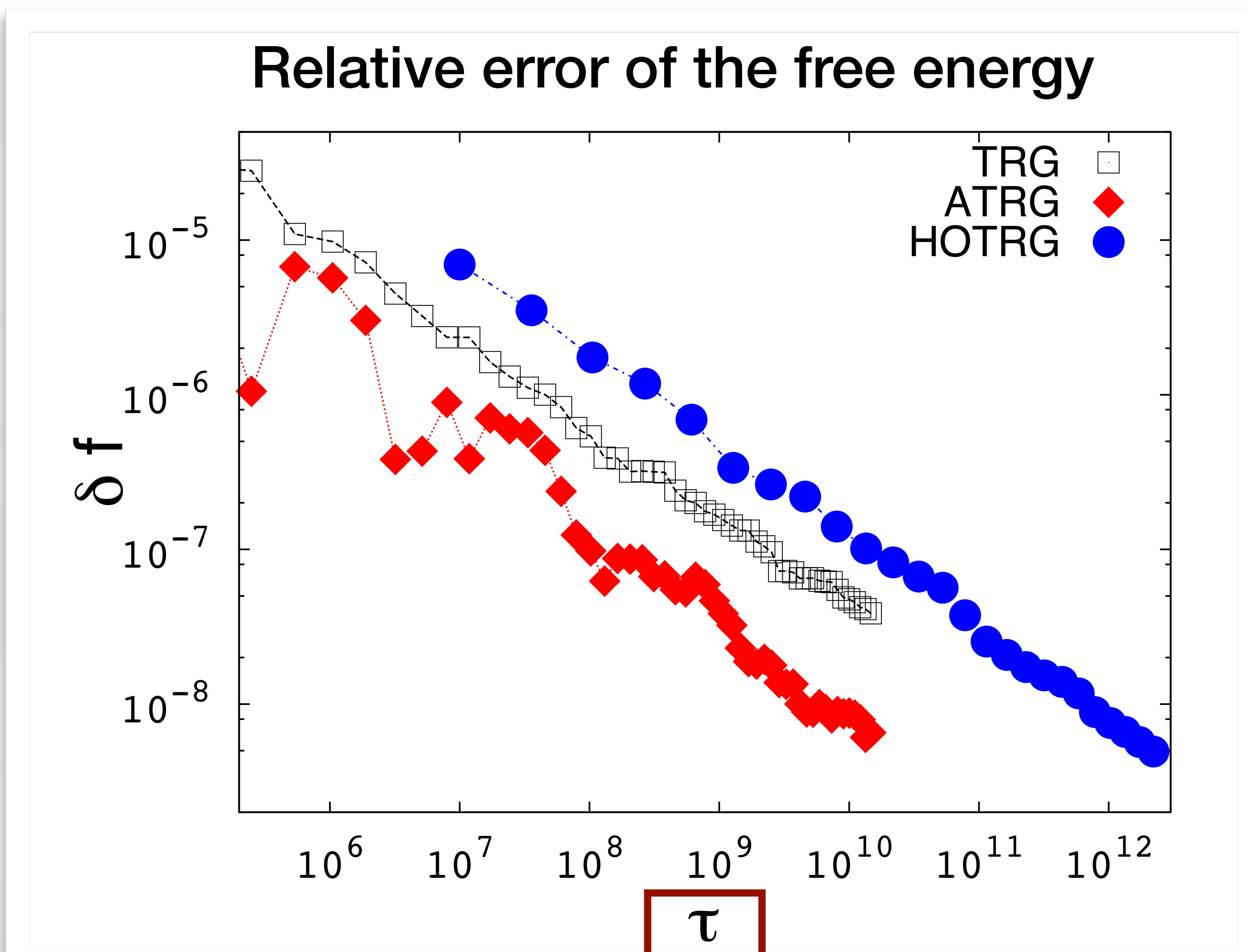
The computation costs are different between ATRG and HOTRG.

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

HOTRG: $O(D^7)$

Leading order computation time:

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

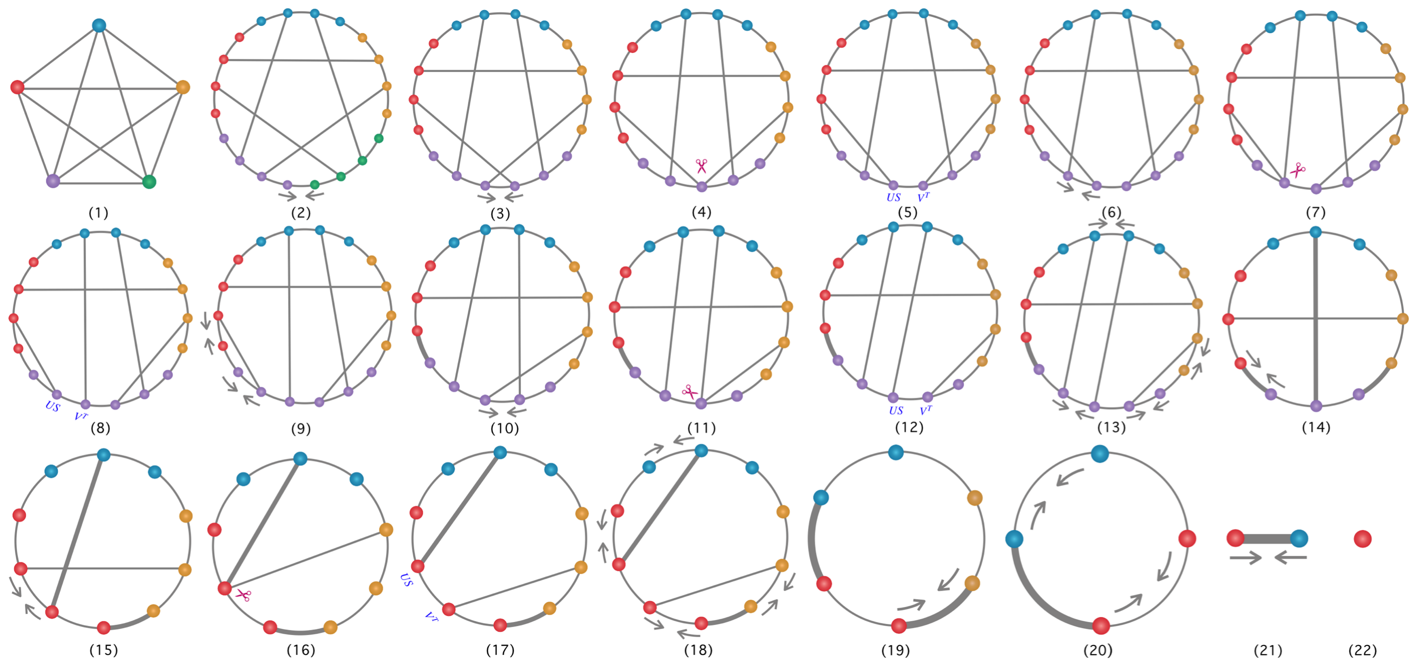


ATRG is the best!

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

Approximate Arbitrary-order Contraction

- Combination of
  - contraction of tensor network with optimized order
  - low-rank approximation using SVD (if bond dimension exceeds χ)
- e.g.)



- Pan, F., Zhou, P., Li, S., & Zhang, P. Contracting Arbitrary Tensor Networks: General Approximate Algorithm and Applications in Graphical Models and Quantum Circuit Simulations. Physical Review Letters, 125(6), 60503 (2020). https://doi.org/10.1103/PhysRevLett.125.060503

Approximate Arbitrary-order Contraction

- Swap, contract & merge

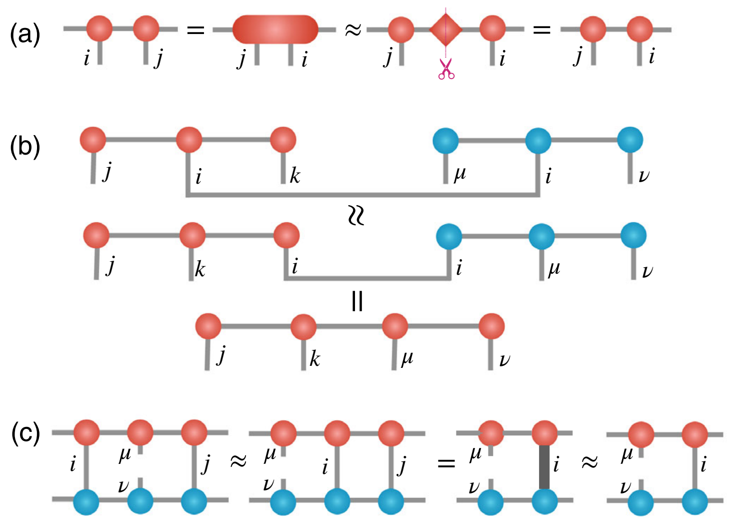
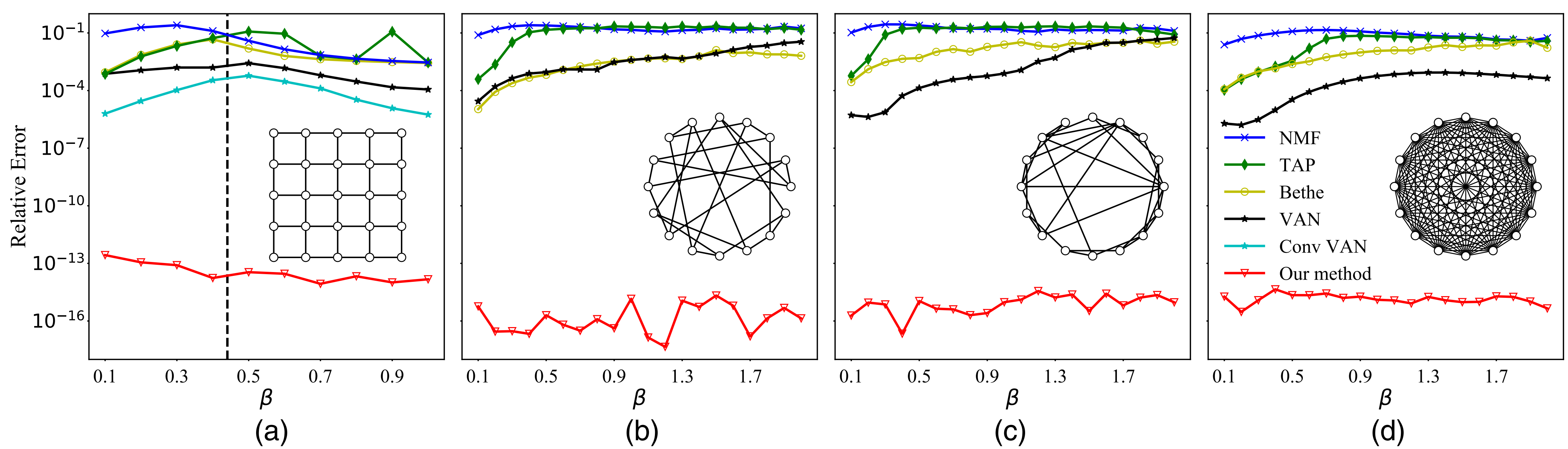


FIG. 3. Illustration of the (a) swap, (b) contract, and (c) merge operations. The scissor symbol indicates truncation of the singular values.

Pan et al, Phys. Rev. Lett. 125, 60503 (2020)

- Benchmark results



Next (Jan. 12)

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