

Outline

— **Introduction**

- Overview on Tensor Network Applications
- Frustrated Quantum Spin Systems

— **Algorithms for Optimization**

- Exact Constructions
- Numerical Optimizations

— **Algorithms for Measurement**

- Corner Transfer Matrix Renormalization Group

Outline

— **Introduction**

- Overview on Tensor Network Applications
- Frustrated Quantum Spin Systems

— **Algorithms for Optimization**

- Exact Constructions
- Numerical Optimizations

— **Algorithms for Measurement**

- Corner Transfer Matrix Renormalization Group

Measurement

❖ Mapping to Classical Stat. Mech.

⇒ Wavefunction & Norm

$$\blacktriangleright |\psi\rangle = \begin{array}{c} \text{Diagram showing a state } |\psi\rangle \text{ represented by a grid of nodes connected by diagonal lines.} \\ \text{The grid has 4 horizontal rows and 5 vertical columns of nodes.} \end{array}$$

$$\boxed{\begin{array}{c} \text{Diagram showing a node } l \text{ with two outgoing lines.} \\ \text{A dashed red circle labeled } l_0 \text{ is associated with the top line.} \\ \text{A dashed red circle labeled } l_1 \text{ is associated with the bottom line.} \end{array}} = l$$

$$\blacktriangleright \langle\psi|\psi\rangle = \begin{array}{c} \text{Diagram showing the inner product } \langle\psi|\psi\rangle \text{ represented by a grid of nodes connected by diagonal lines.} \\ \text{The grid has 4 horizontal rows and 5 vertical columns of nodes, identical to the state diagram above.} \end{array}$$

$$= \begin{array}{c} \text{Diagram showing the result of the inner product calculation.} \\ \text{The grid has 4 horizontal rows and 5 vertical columns of nodes.} \\ \text{Each node is colored yellow, indicating it is a fully connected state.} \end{array}$$

Norm is Partition function of Classical Stat. Mech.!

Measurement

❖ Mapping to Classical Stat. Mech.

⇒ Wavefunction & Norm

$$\blacktriangleright |\psi\rangle = \begin{array}{c} \text{Diagram showing a grid of nodes connected by diagonal lines, representing a wavefunction state.} \end{array}$$

$$\boxed{l_0 \quad l_1} = l \quad \text{Diagram showing two nodes labeled } l_0 \text{ and } l_1 \text{ connected by a horizontal line, which is then connected to a yellow node labeled } l.$$

$$\blacktriangleright \langle\psi|\psi\rangle = \begin{array}{c} \text{Diagram showing a grid of nodes connected by diagonal lines, representing the wavefunction state.} \\ = \begin{array}{c} \text{Diagram showing a grid of nodes connected by horizontal lines, representing the norm calculation.} \end{array} \end{array}$$

$$= \begin{array}{c} \text{Diagram showing a grid of nodes connected by horizontal lines, all colored yellow.} \end{array}$$

⇒ Expectation value

$$\blacktriangleright \langle\psi|\hat{O}_i|\psi\rangle = \begin{array}{c} \text{Diagram showing a grid of nodes connected by diagonal lines, with one red node highlighted.} \end{array}$$

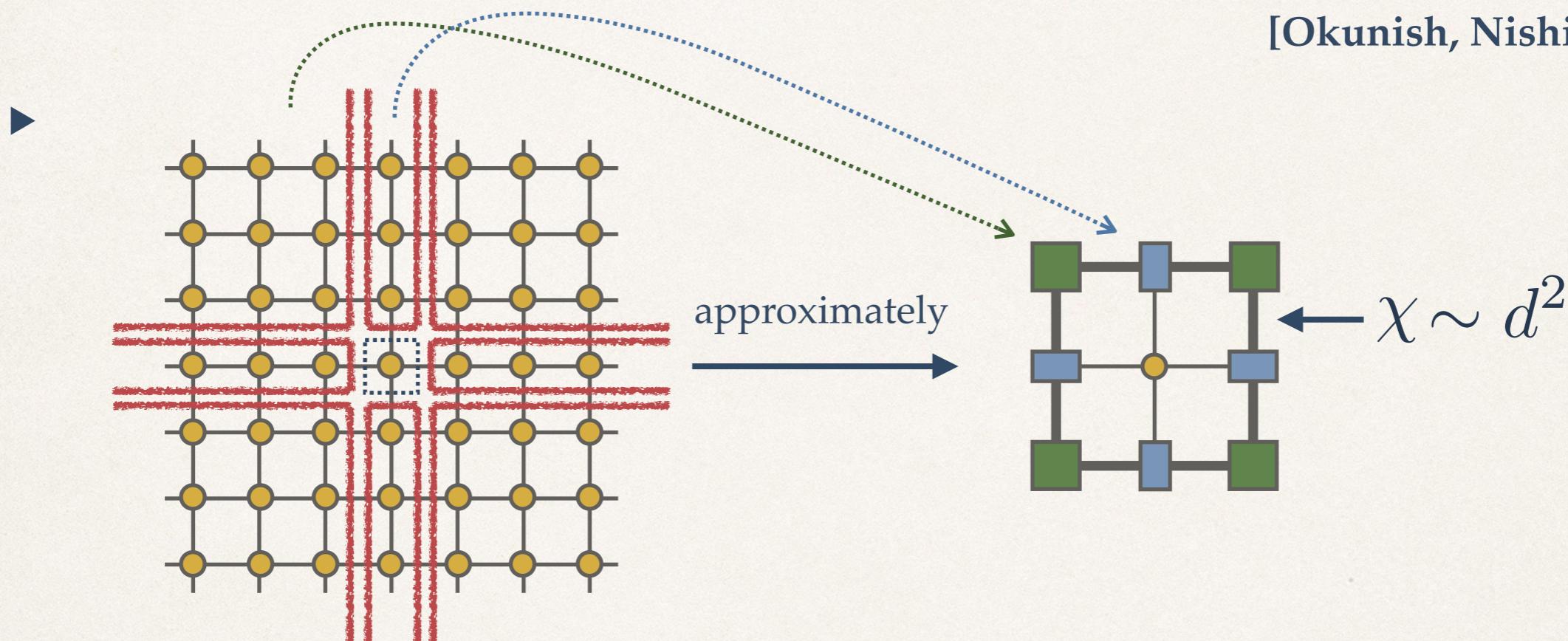
$$\boxed{\hat{O}} = \text{Diagram showing a red circle with a central dot, representing the expectation value of operator } \hat{O}.$$

Measurement

❖ Corner Transfer Matrix Renormalization Group

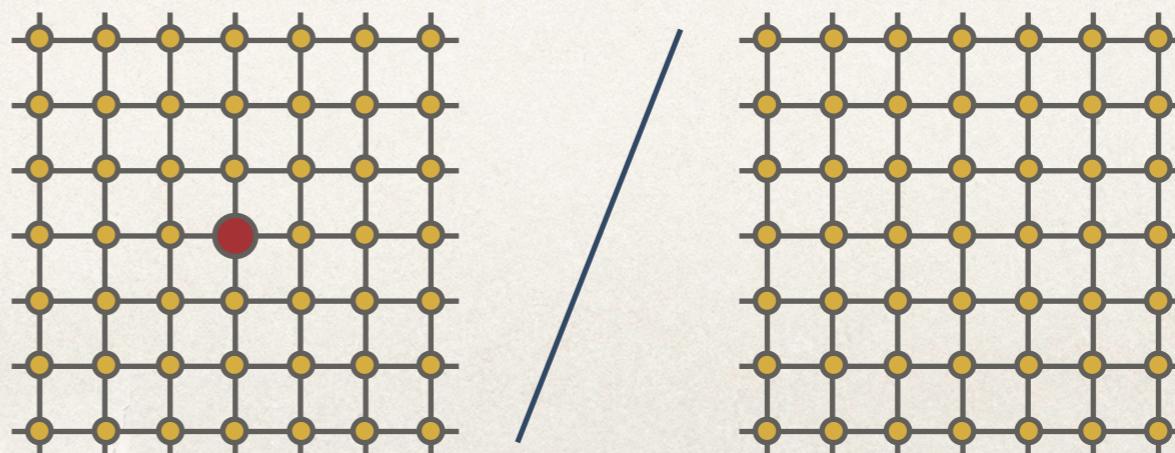
- ▷ Environment tensors effectively represent Infinite Tensor Network

[Okunish, Nishino (1996)]



- ▶ Useful for TPS measurement

$$\langle \hat{O}_i \rangle = \frac{\langle \psi | \hat{O}_i | \psi \rangle}{\langle \psi | \psi \rangle} =$$

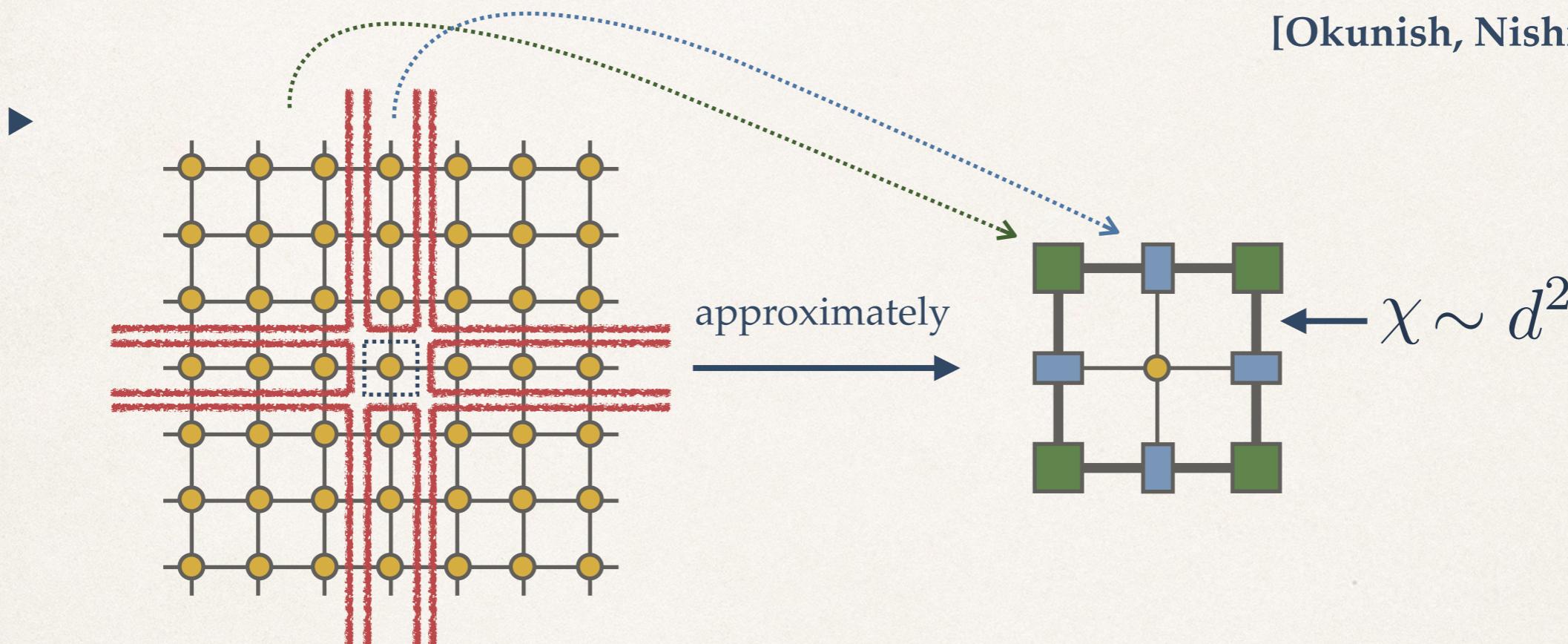


Measurement

❖ Corner Transfer Matrix Renormalization Group

- ▷ Environment tensors effectively represent Infinite Tensor Network

[Okunish, Nishino (1996)]



- ▶ Useful for TPS measurement

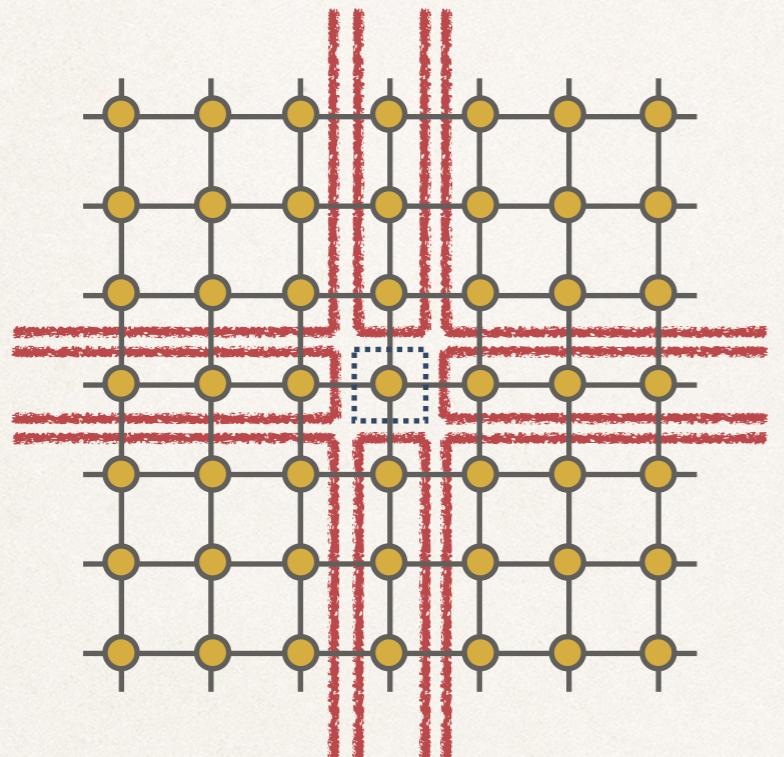
$$\langle \hat{O}_i \rangle = \frac{\langle \psi | \hat{O}_i | \psi \rangle}{\langle \psi | \psi \rangle} = \begin{array}{c} \text{Diagram of a 3x3 grid with a red circle at the center} \\ \diagup \quad \diagdown \end{array} \begin{array}{c} \text{Diagram of a 3x3 grid with a yellow circle at the center} \\ \diagup \quad \diagdown \end{array}$$

Measurement

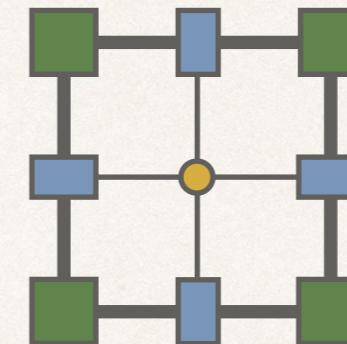
❖ Corner Transfer Matrix Renormalization Group

- ▷ Environment tensors effectively represent Infinite Tensor Network

[Okunish, Nishino (1996)]

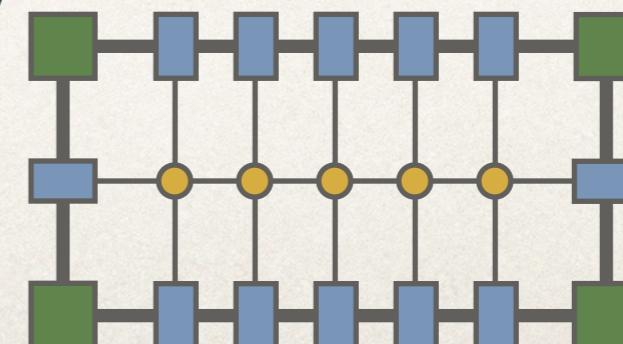
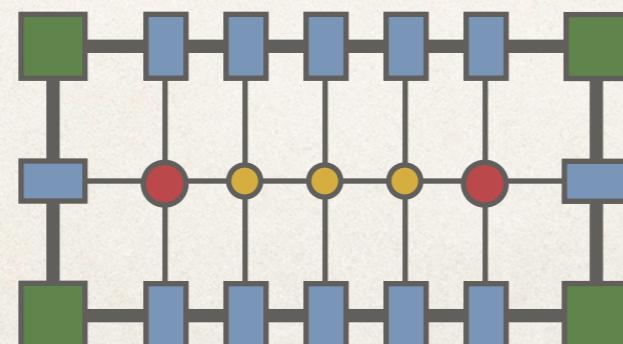


approximately



- ▶ Useful for TPS measurement

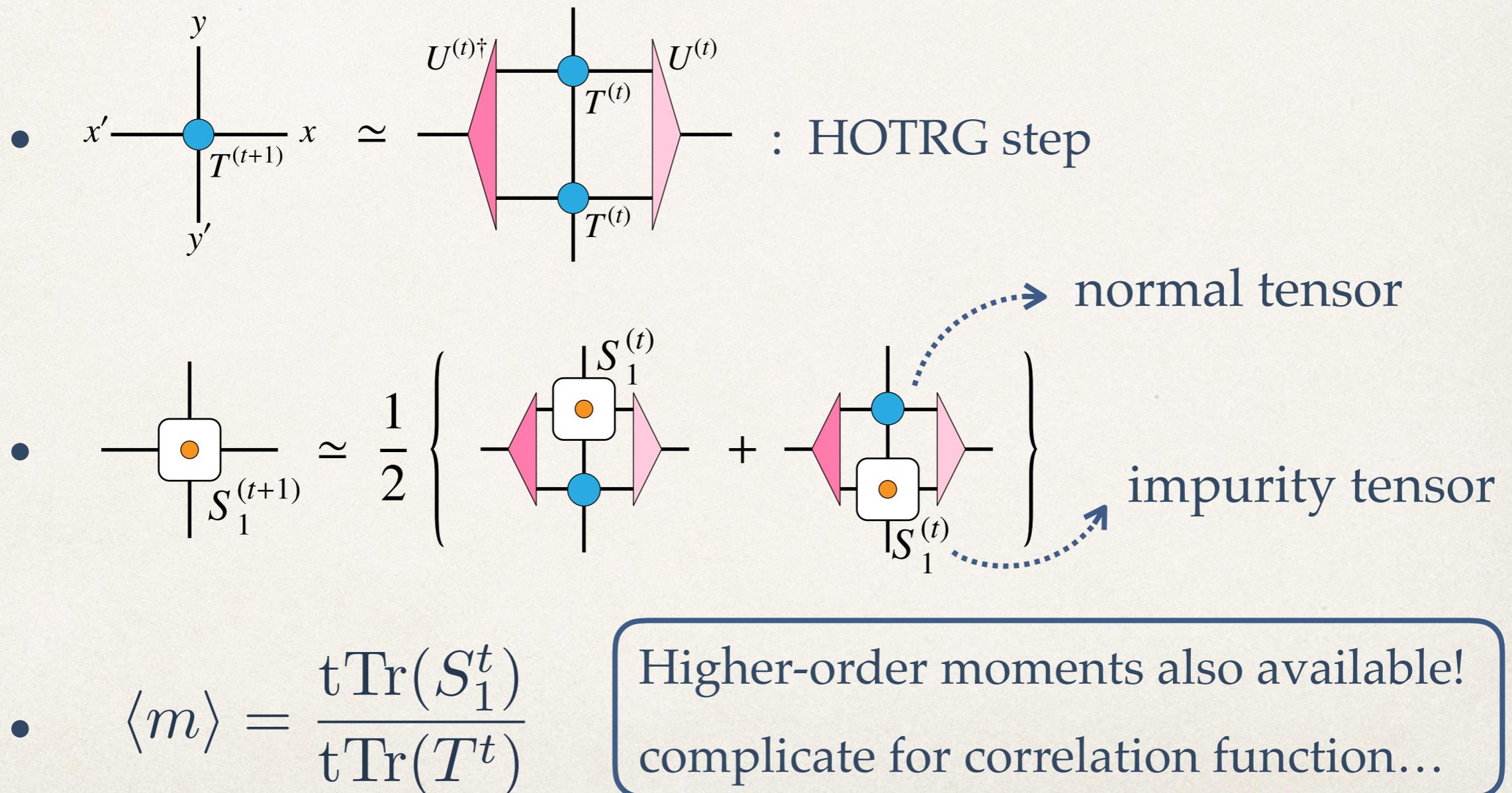
$$\langle \hat{O}_i \hat{O}_j \rangle =$$



Measurement

※ HOTRG Implementation [Satoshi, Naoki (2018)]

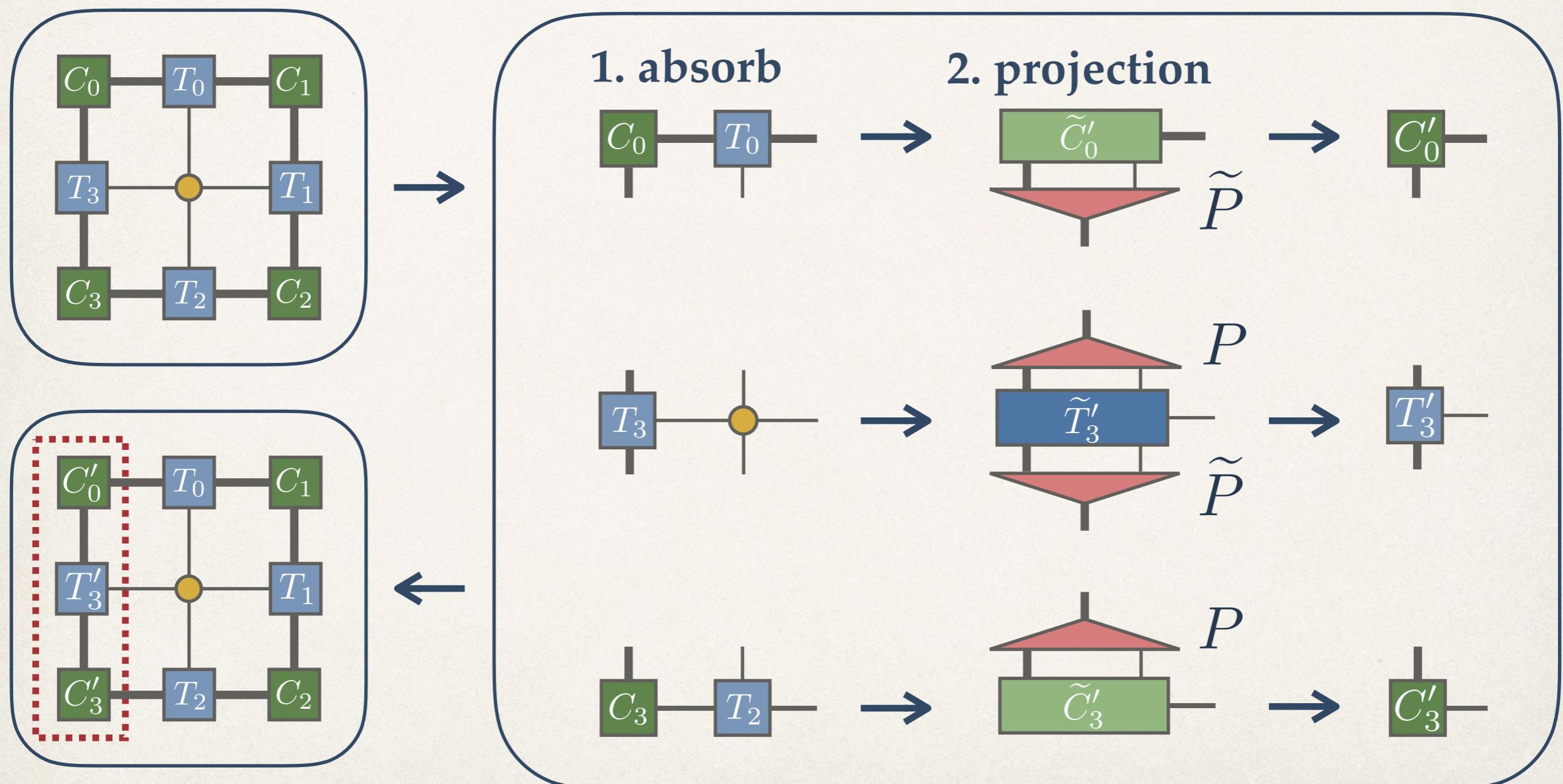
- ▷ “Impurity” tensor at which the operator is inserted
- ▶ Utilize the same isometry for both normal and impurity tensors



Corner Transfer Matrix Renormalization Group

❖ Algorithm in details

⇒ Left-direction RG [Corboz et al. Phys. Rev. Lett. 113, 046402 (2014)]



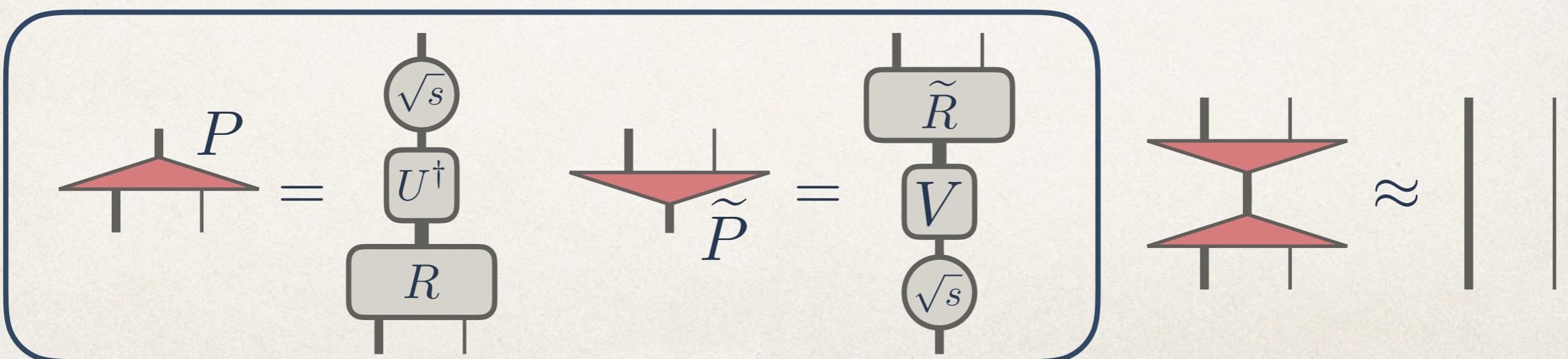
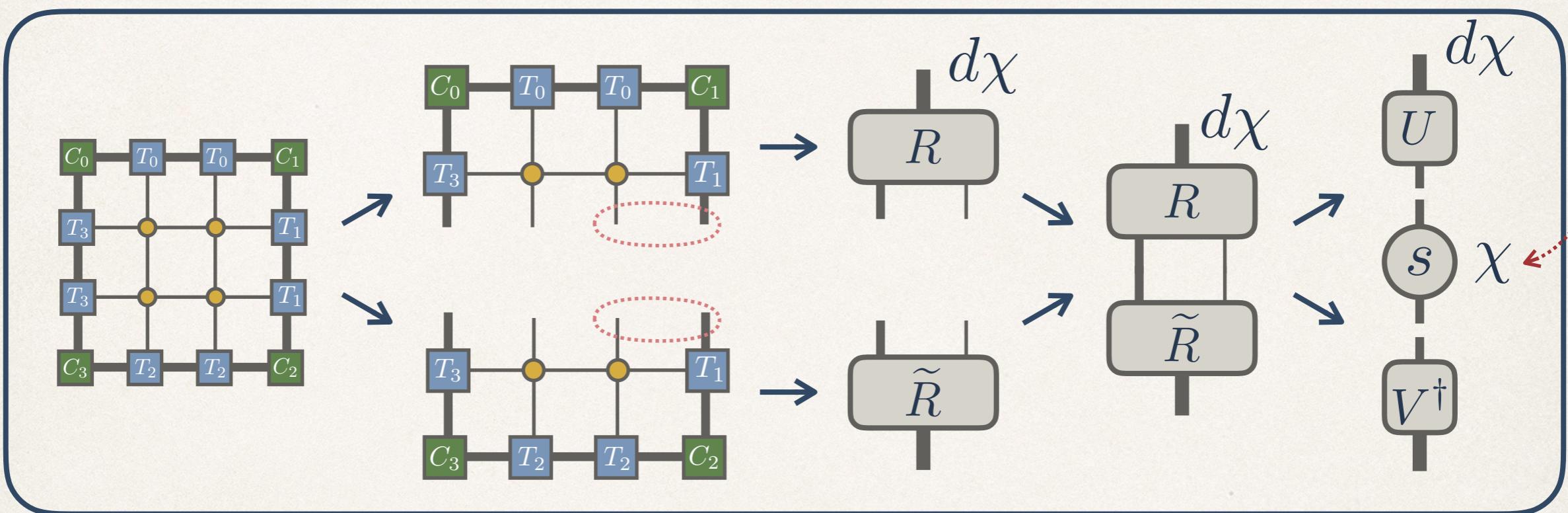
Left env. tensors are updated

Corner Transfer Matrix Renormalization Group

❖ Algorithm in details

⇒ How to obtain Projector for Left-RG

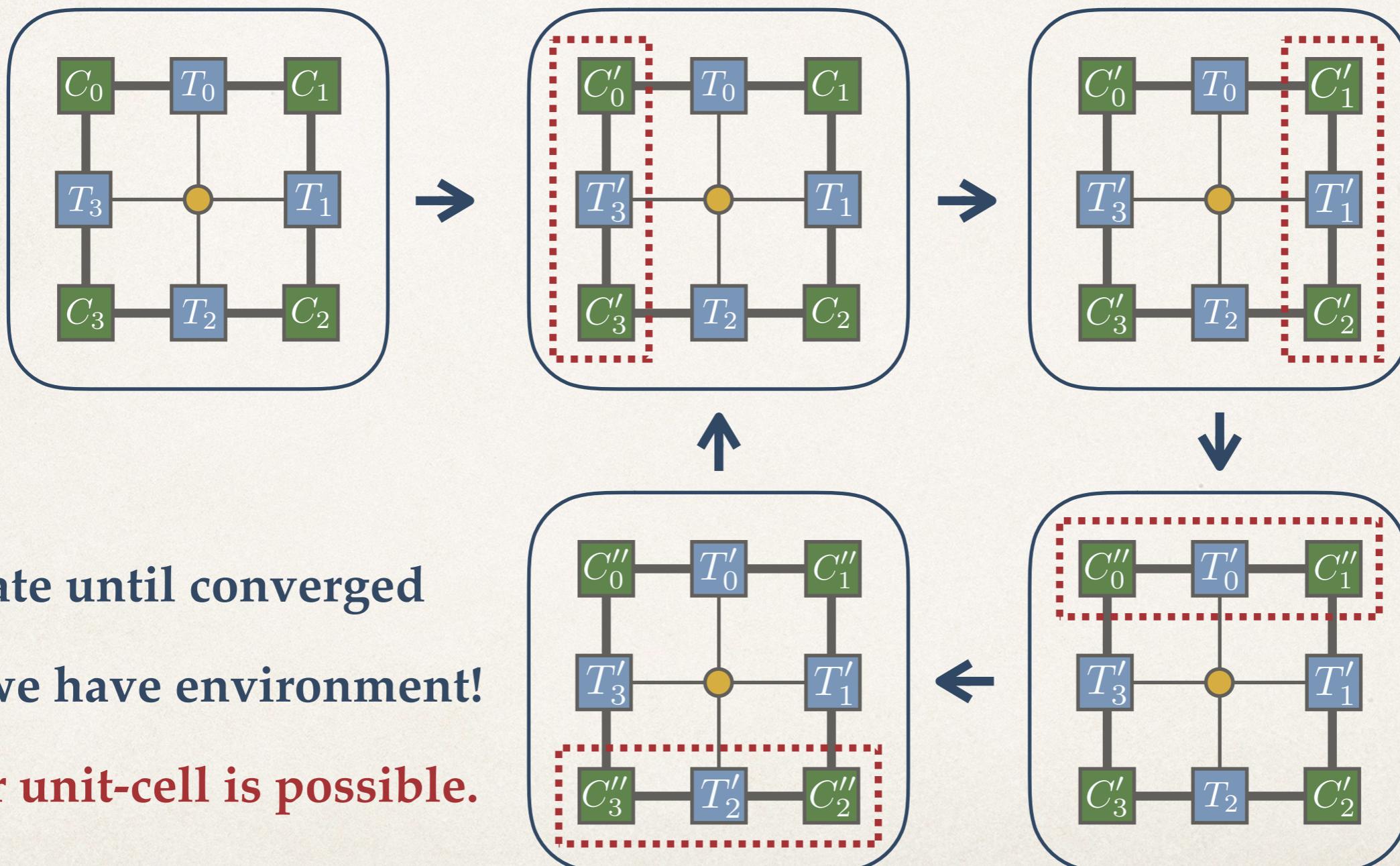
truncation!



Corner Transfer Matrix Renormalization Group

- ❖ Algorithm in details

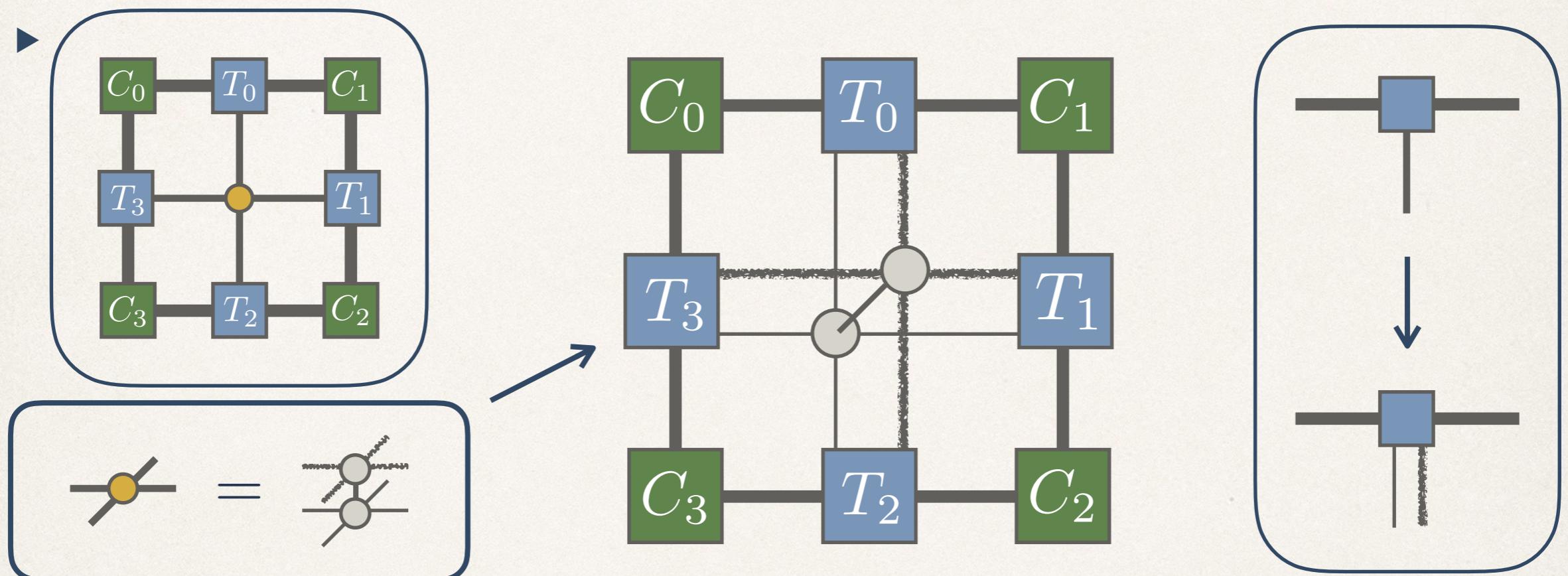
- ▷ Complete RG procedure



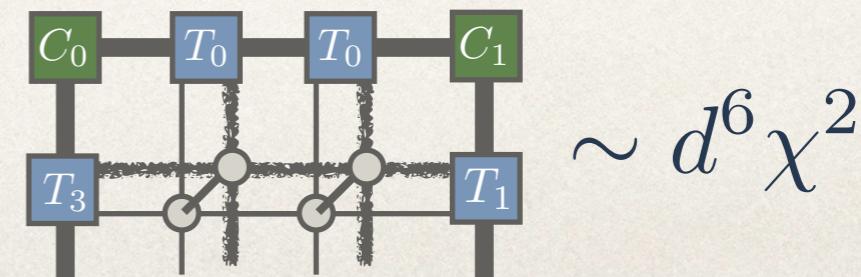
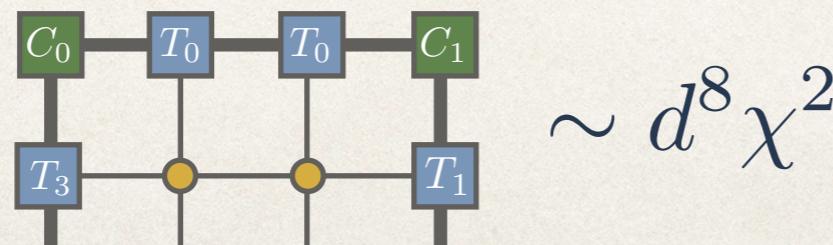
Corner Transfer Matrix Renormalization Group

❖ Algorithm in details

⇒ Tip to reduce complexity



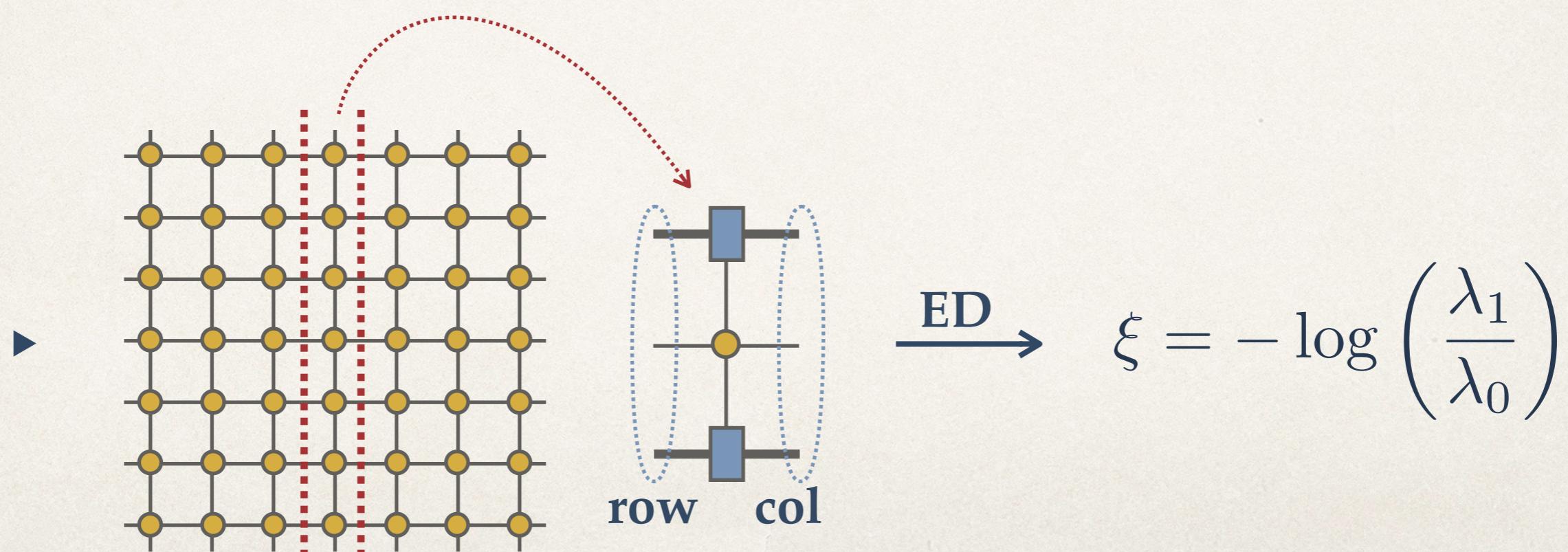
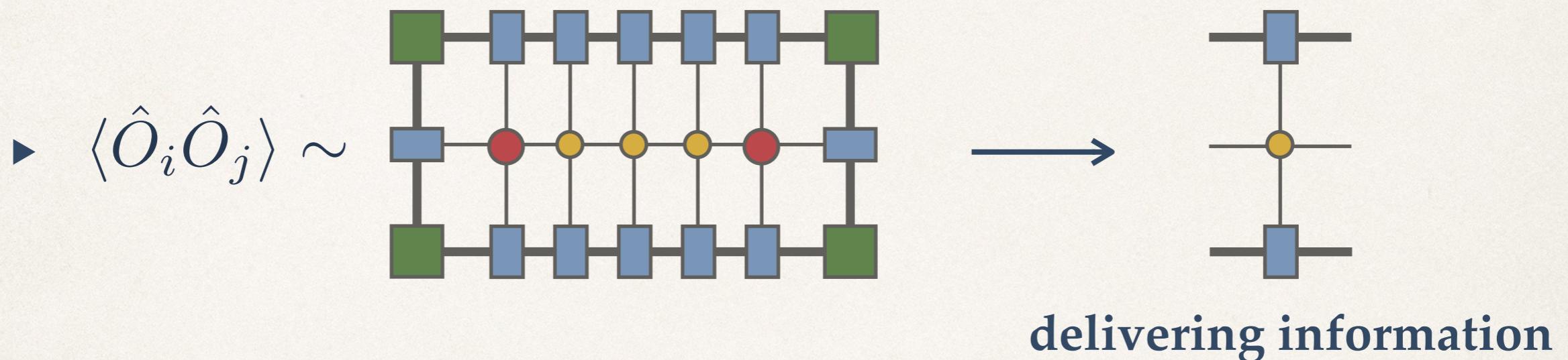
► Heaviest Part



Corner Transfer Matrix Renormalization Group

❖ Extracting Correlation Length

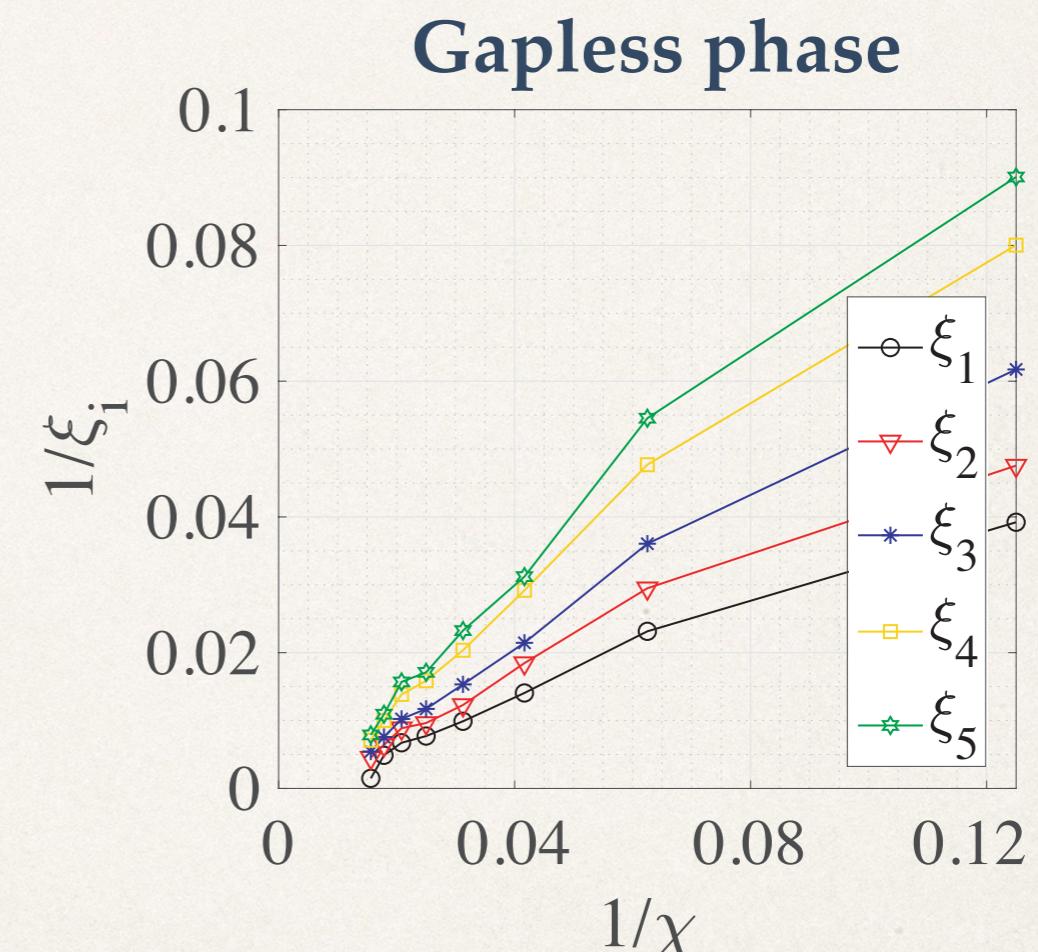
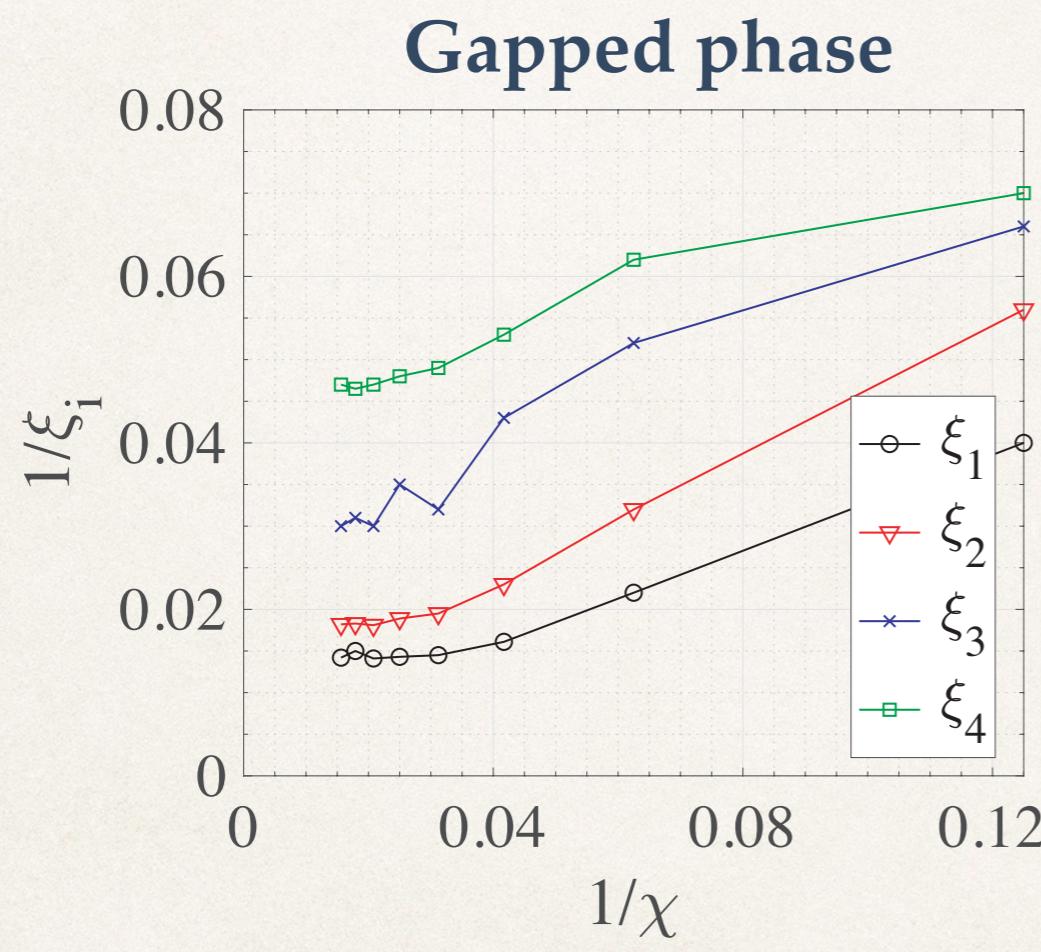
⇒ Who delivers the information(or correlation)?



Corner Transfer Matrix Renormalization Group

❖ Extracting Correlation Length

↳ Example: Gapless and Gapped Kitaev Spin Liquid



Gap nature can be extracted from Edge tensors in CTMRG

Outline

— **Introduction**

- Overview on Tensor Network Applications
- Frustrated Quantum Spin Systems

— **Algorithms for Optimization**

- Exact Constructions
- Numerical Optimizations

— **Algorithms for Measurement**

- Corner Transfer Matrix Renormalization Group

Numerical Optimization

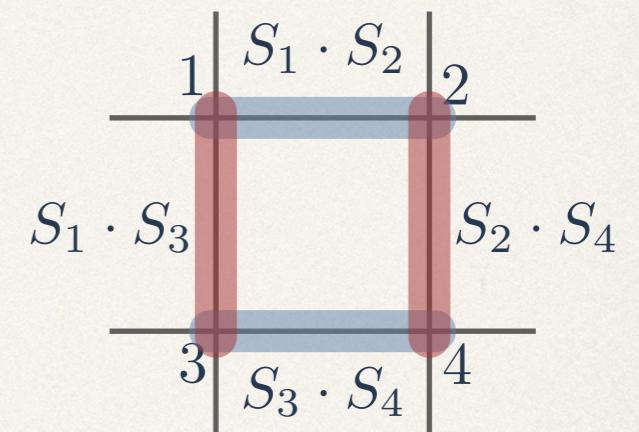
❖ Imaginary Time Evolution

⇒ Idea: $|\text{GS}\rangle = \lim_{N \rightarrow \infty} \left(e^{-\tau \hat{H}} \right)^N |\psi\rangle$

not commute

TN representation?

Generally, no...



⇒ Suzuki-Trotter decomposition

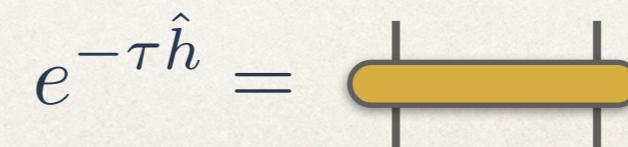
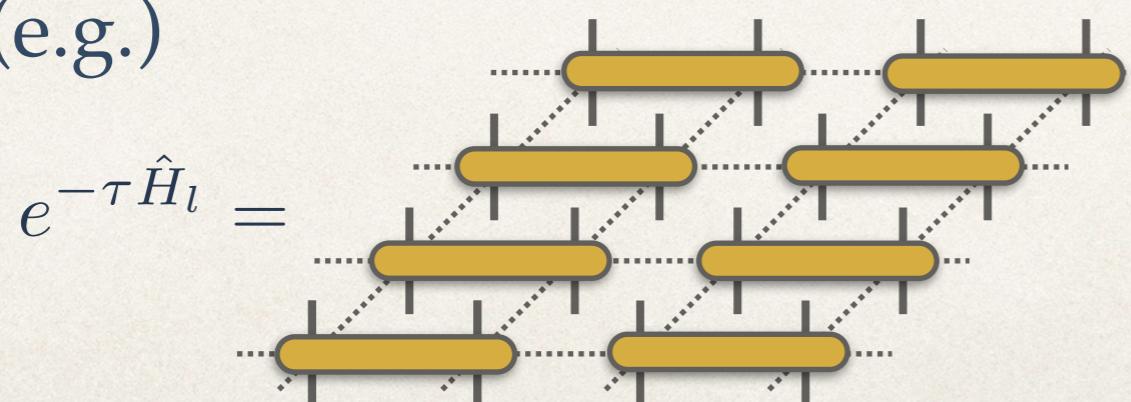
► group: $\hat{H} = \hat{H}_l + \hat{H}_u + \hat{H}_r + \hat{H}_d$

► decouple: $e^{-\tau \hat{H}} \approx [e^{-\tau \hat{H}_l}] e^{-\tau \hat{H}_u} e^{-\tau \hat{H}_r} e^{-\tau \hat{H}_d}$

\downarrow

$$e^{-\tau \hat{H}_l} = e^{-\tau \hat{h}_{ab}} \otimes e^{-\tau \hat{h}_{cd}} \otimes e^{-\tau \hat{h}_{cd}} \dots$$

(e.g.)



Numerical Optimization

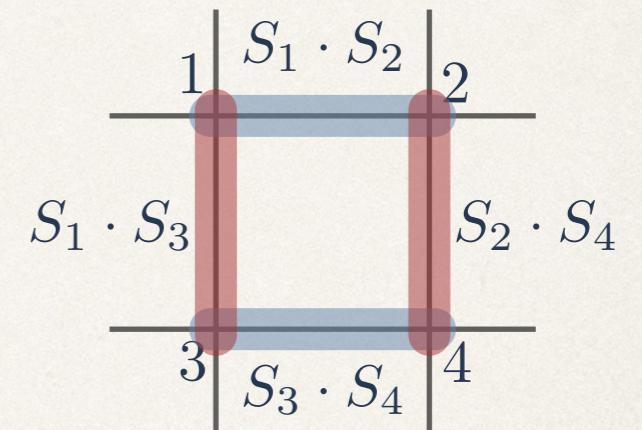
❖ Imaginary Time Evolution

⇒ Idea: $|\text{GS}\rangle = \lim_{N \rightarrow \infty} \left(e^{-\tau \hat{H}} \right)^N |\psi\rangle$

not commute

TN representation?

Generally, no...



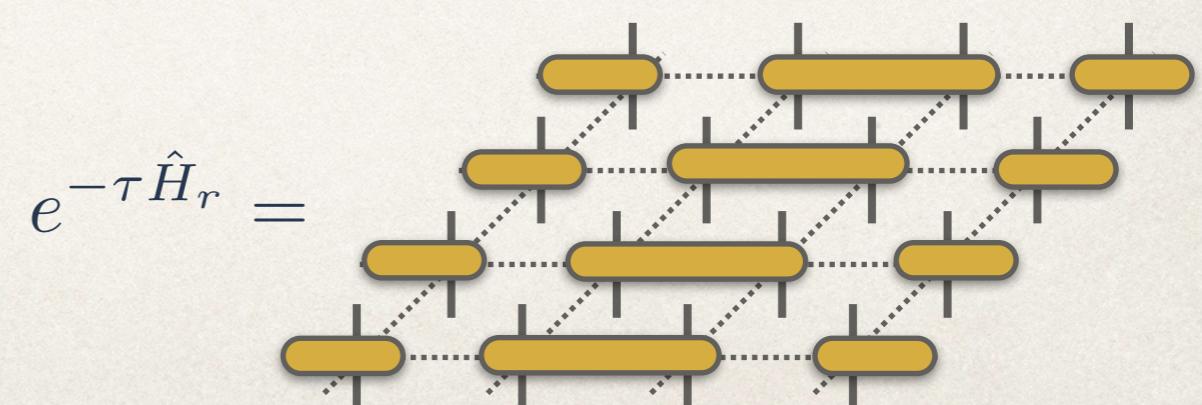
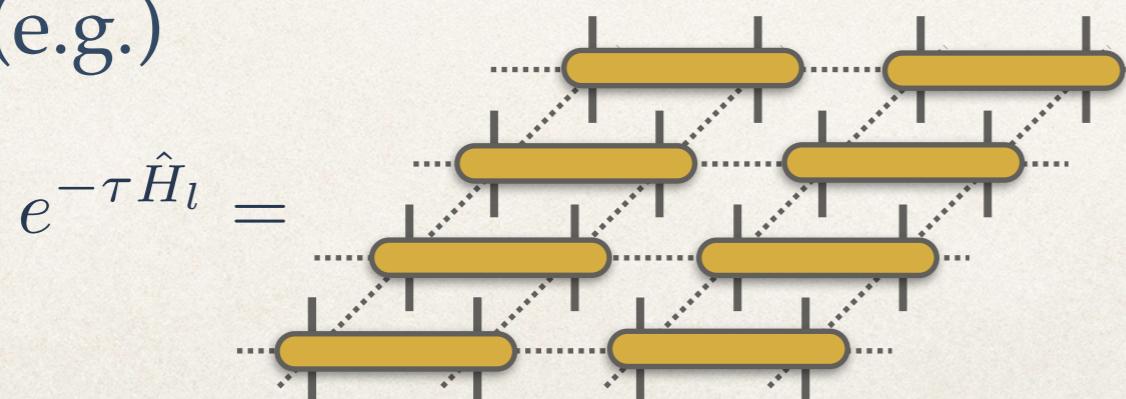
⇒ Suzuki-Trotter decomposition

► group: $\hat{H} = \hat{H}_l + \hat{H}_u + \hat{H}_r + \hat{H}_d$

► decouple: $e^{-\tau \hat{H}} \approx [e^{-\tau \hat{H}_l}] e^{-\tau \hat{H}_u} e^{-\tau \hat{H}_r} e^{-\tau \hat{H}_d}$

$$e^{-\tau \hat{H}_l} = e^{-\tau \hat{h}_{ab}} \otimes e^{-\tau \hat{h}_{cd}} \otimes e^{-\tau \hat{h}_{cd}} \dots$$

(e.g.)

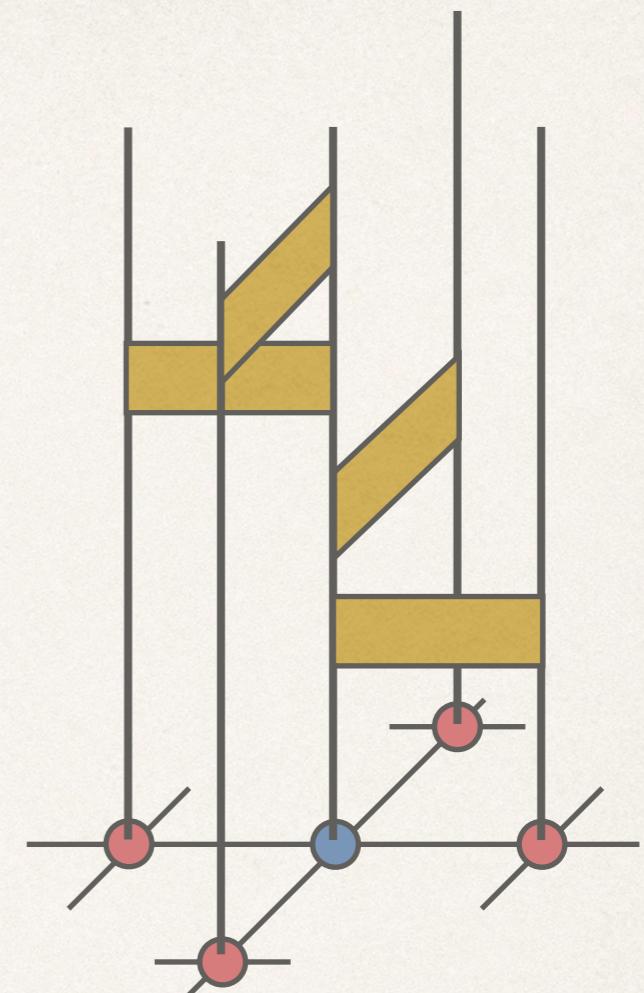
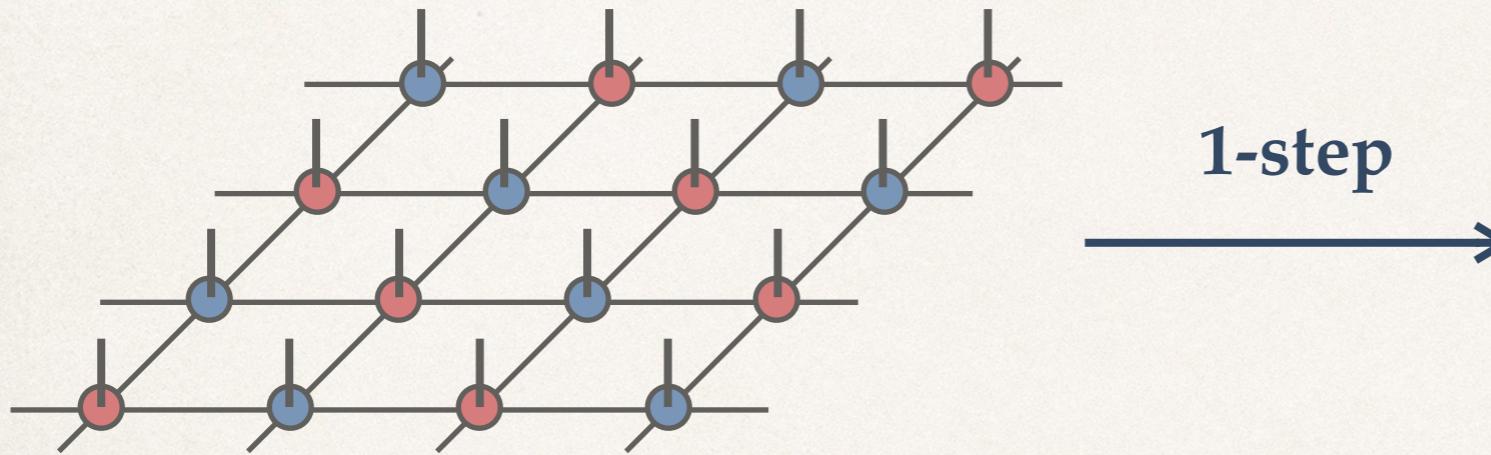


Numerical Optimization

❖ Imaginary Time Evolution

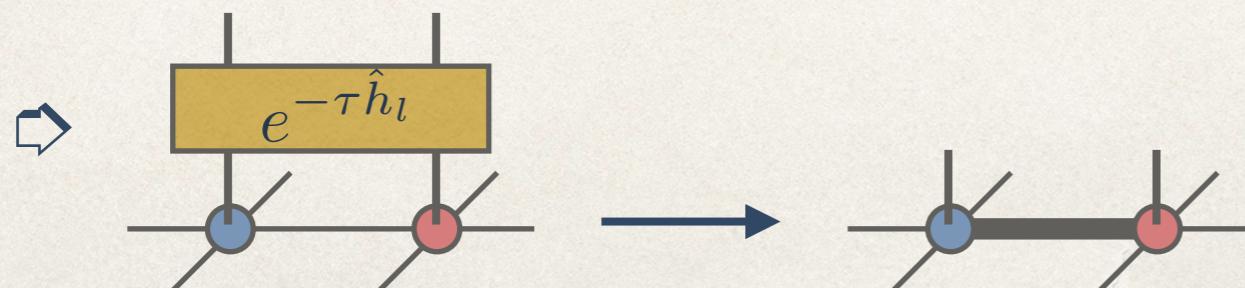
⇒ Applying ITE operator $e^{-\tau \hat{h}}$

(NOTE: ITE requires at least 2-site unitcell)



⇒ $e^{-\tau \hat{h}_l}$ = $U \Sigma V$

e.g.) Heisenberg $\tau = 0.001$ $\Sigma \approx \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}$



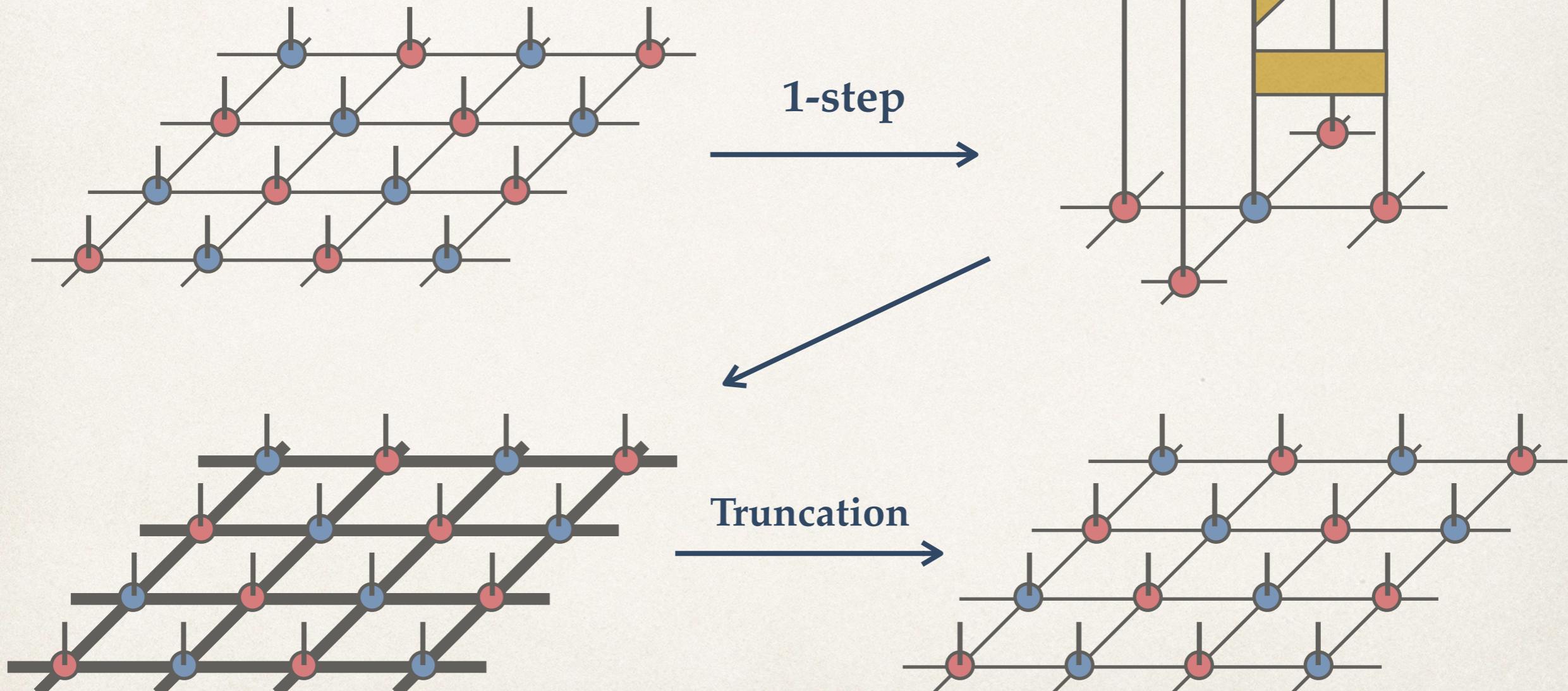
Bond dimension increases!

Numerical Optimization

❖ Imaginary Time Evolution

▷ Applying ITE operator $e^{-\tau \hat{h}}$

(NOTE: ITE requires at least 2-site unitcell)



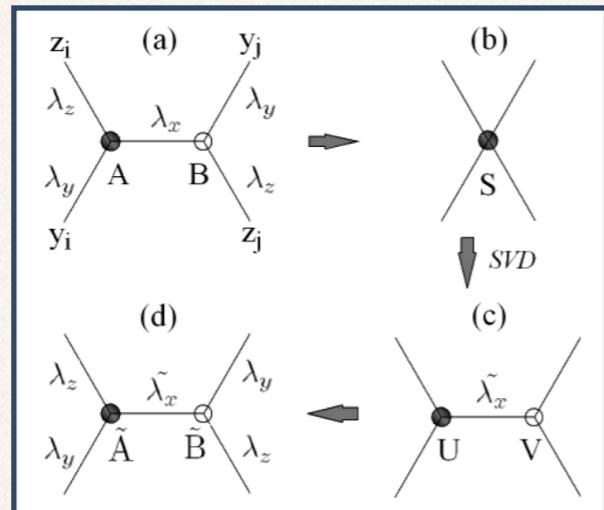
Truncation is required!

Numerical Optimization

❖ Imaginary Time Evolution

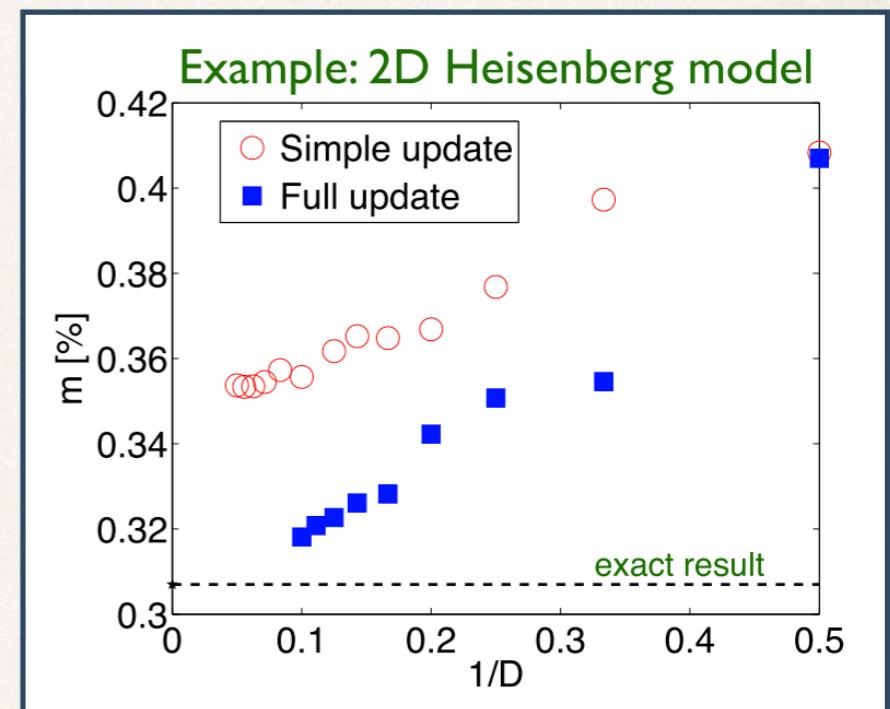
▷ How to truncate?

(a) Simple update - Easy and cheap $O(D^5)$



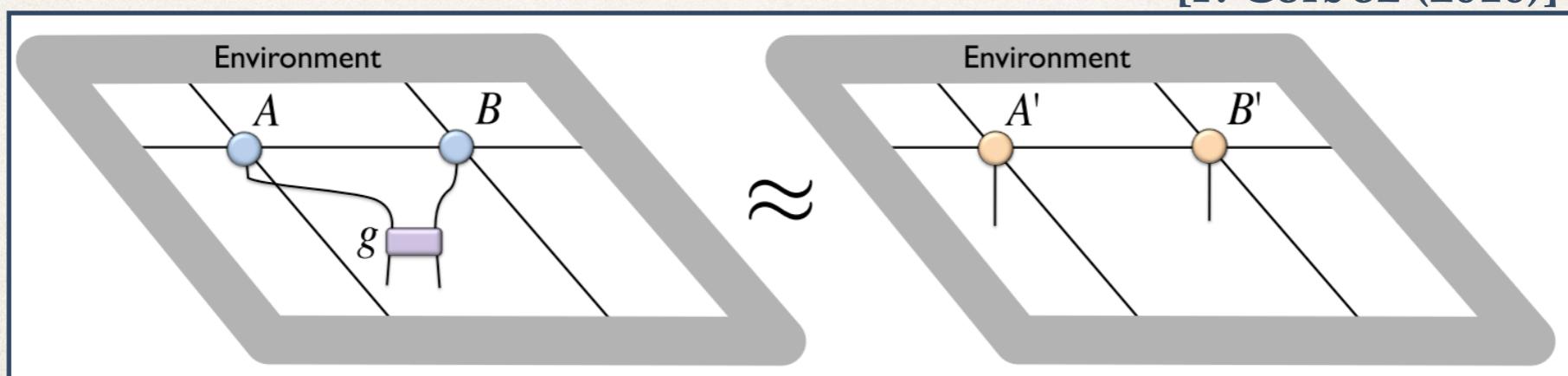
[Jiang at al. (2008)]

[P. Corboz (2016)]



(b) Full update - Better accurate but heavy $O(D^{10})$

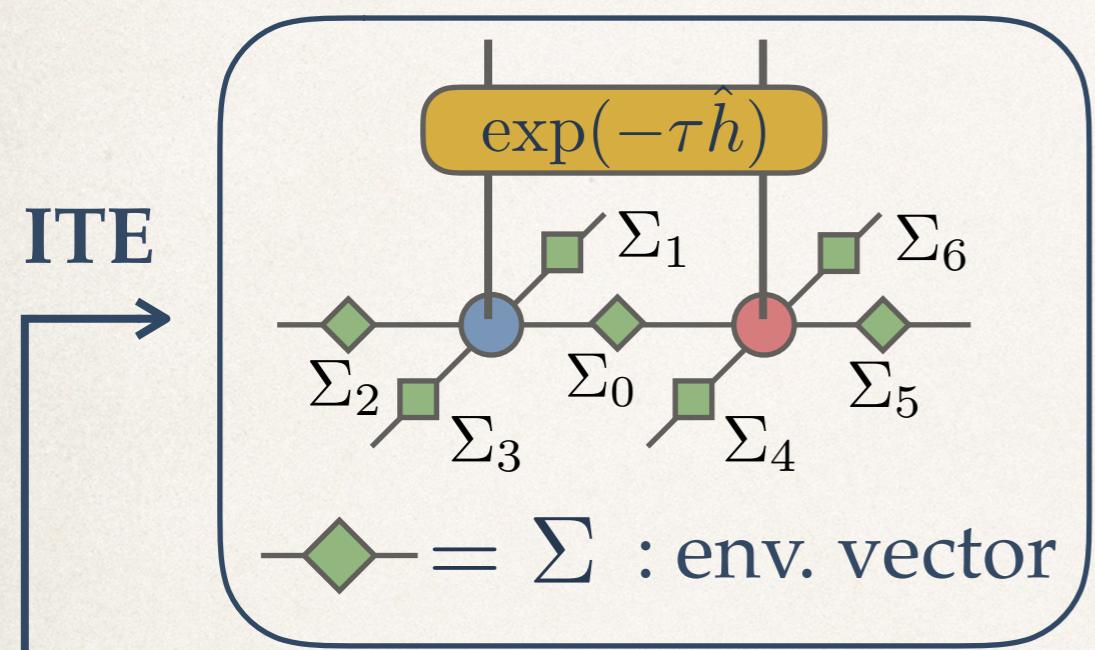
[P. Corboz (2016)]



Numerical Optimization

❖ Imaginary Time Evolution

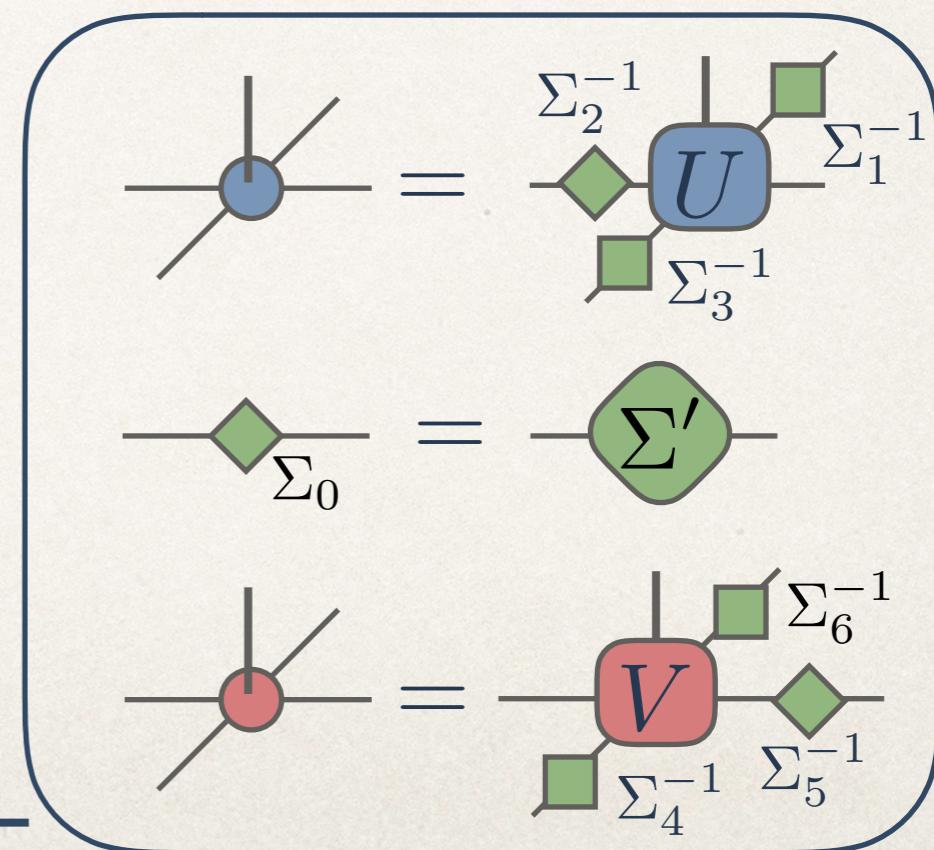
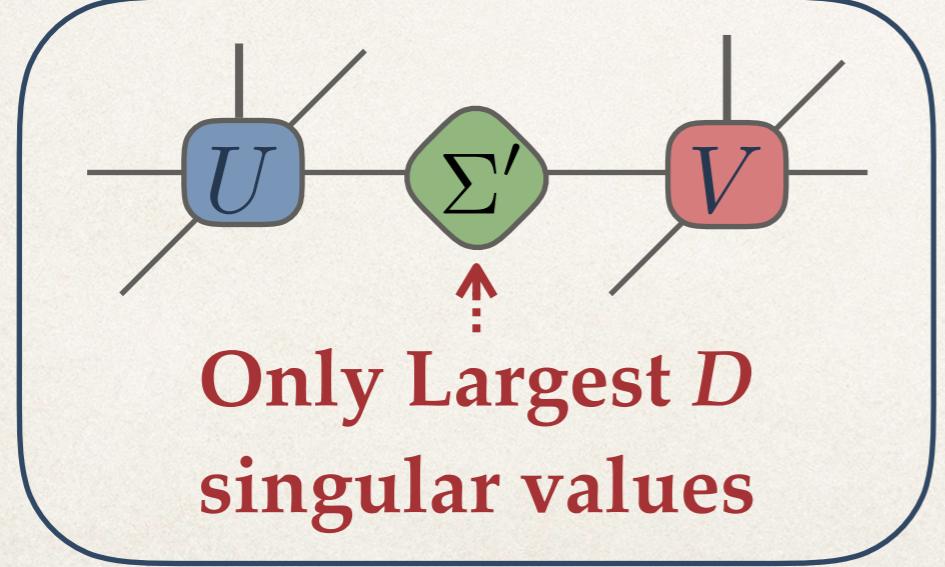
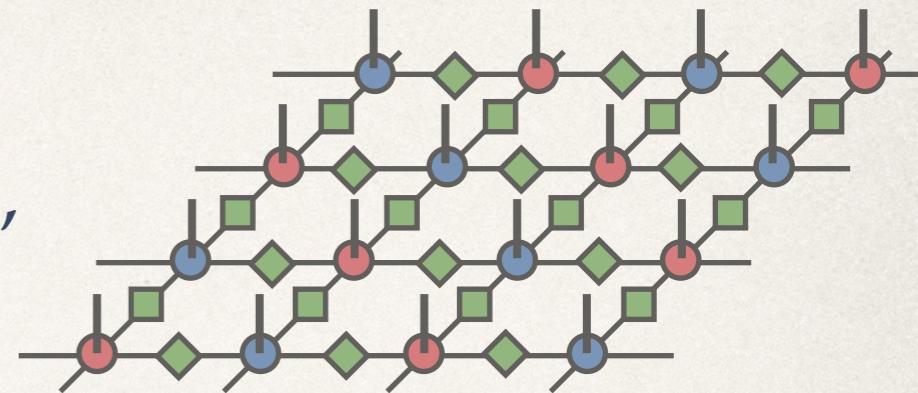
▷ Simple update: Averaged “Entanglement”



contract

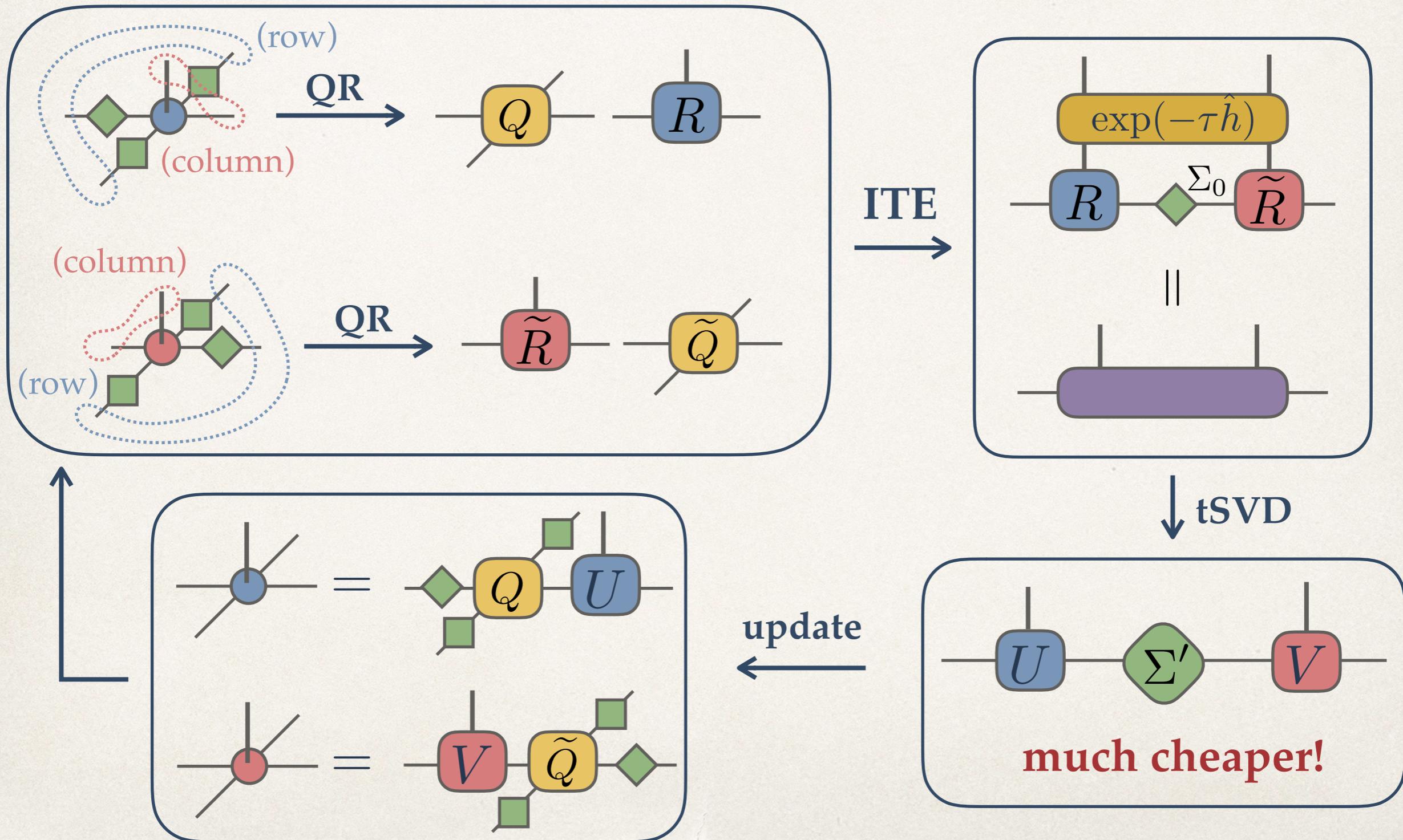
tSVD

update



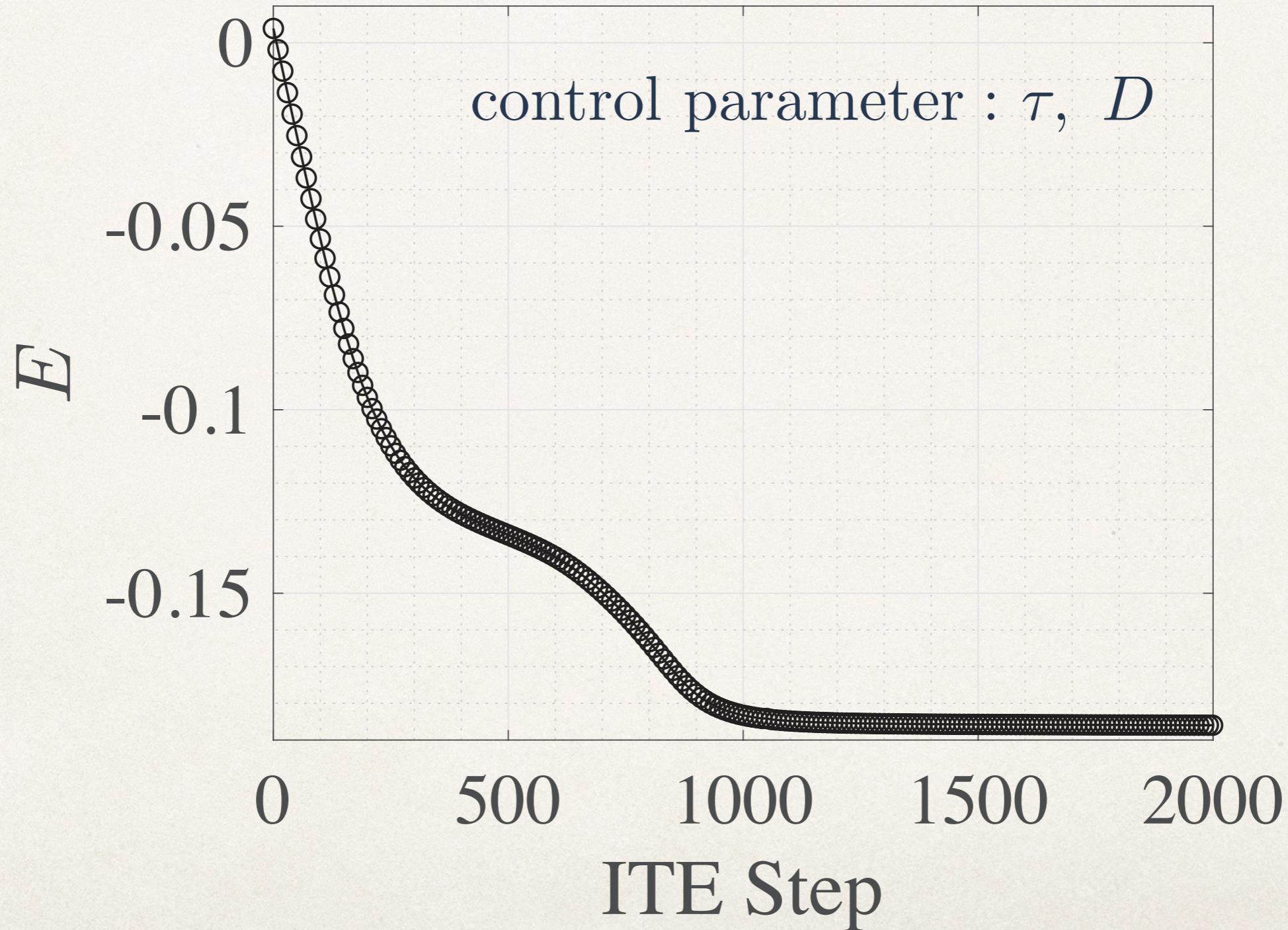
Numerical Optimization

- ❖ Imaginary Time Evolution
 - ▷ Tip to reduce complexity



Numerical Optimization

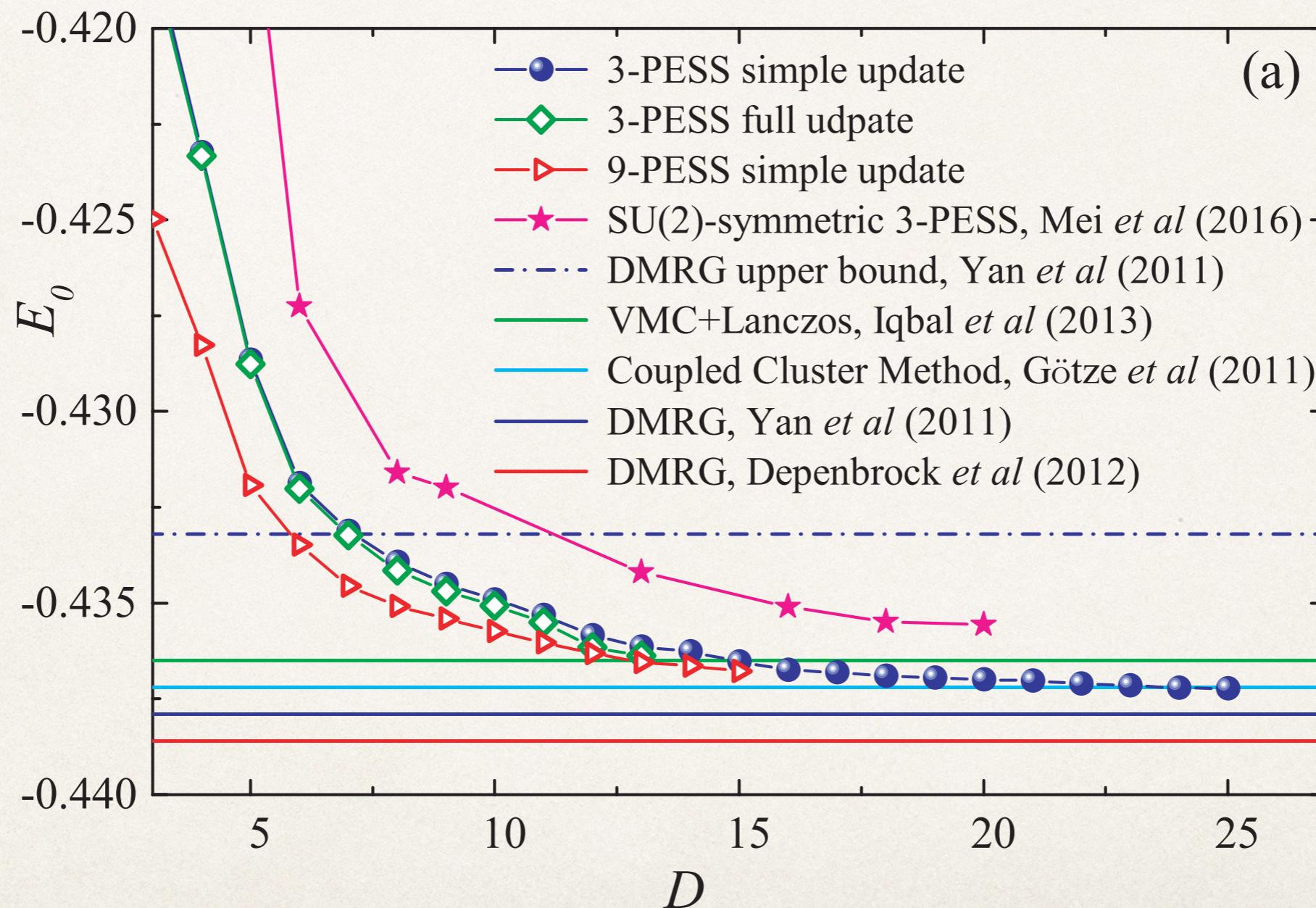
- ❖ Sample
 - ▷ ITE flow



Numerical Optimization

❖ Example1

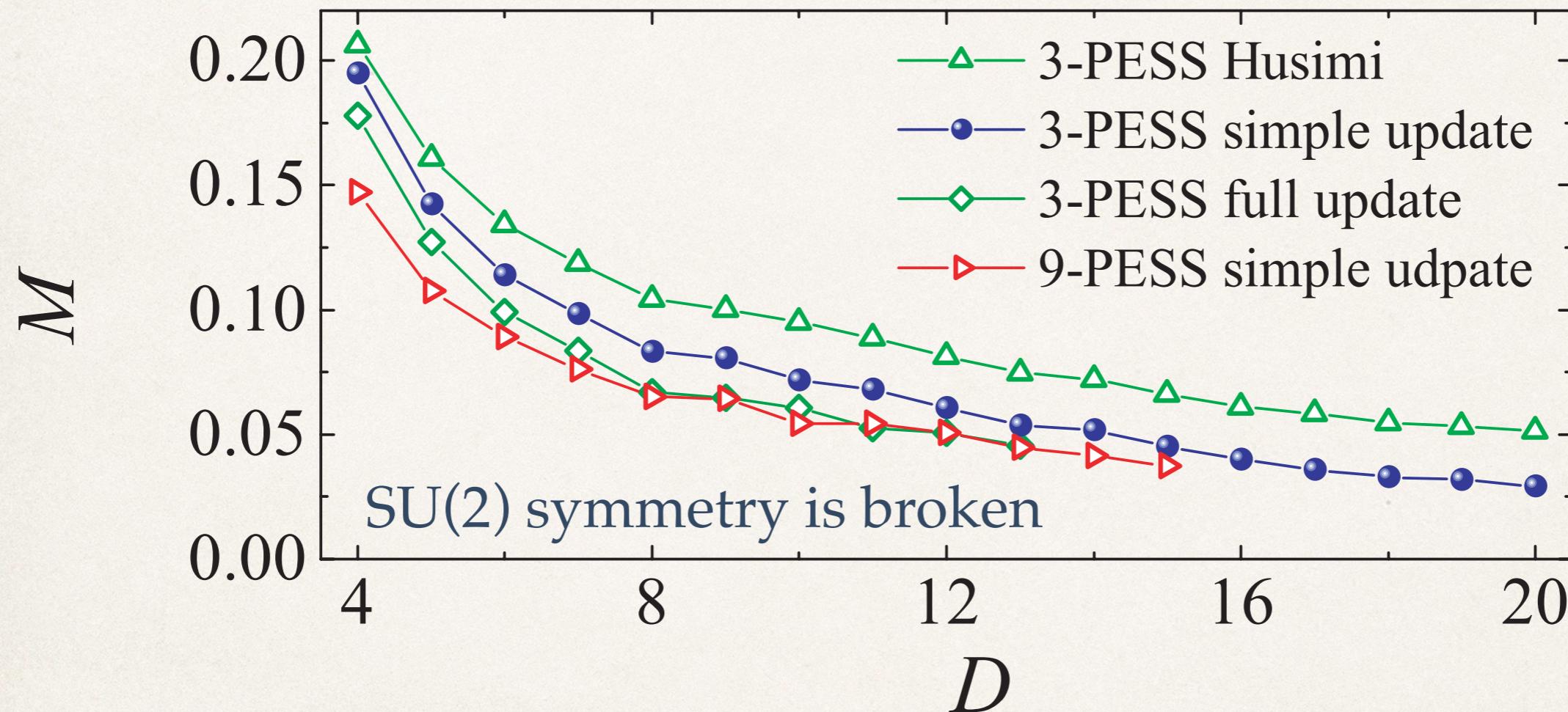
▷ Gapless Spin Liquid on Kagome [Liao et al. (2018)]



Numerical Optimization

❖ Example1

▷ Gapless Spin Liquid on Kagome [Liao et al. (2018)]

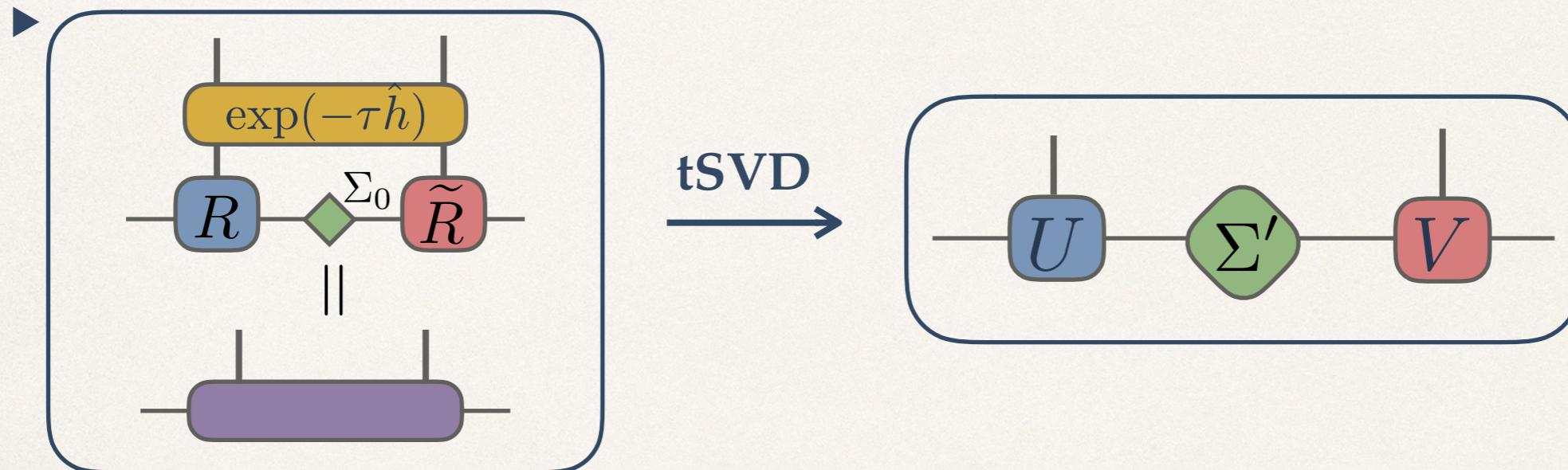


Magnetization is not exactly Zero, even though it is decreasing

Numerical Optimization

❖ Symmetric Simple Update

▷ Degeneracy in Singular Values



► if is symmetric, are degenerates

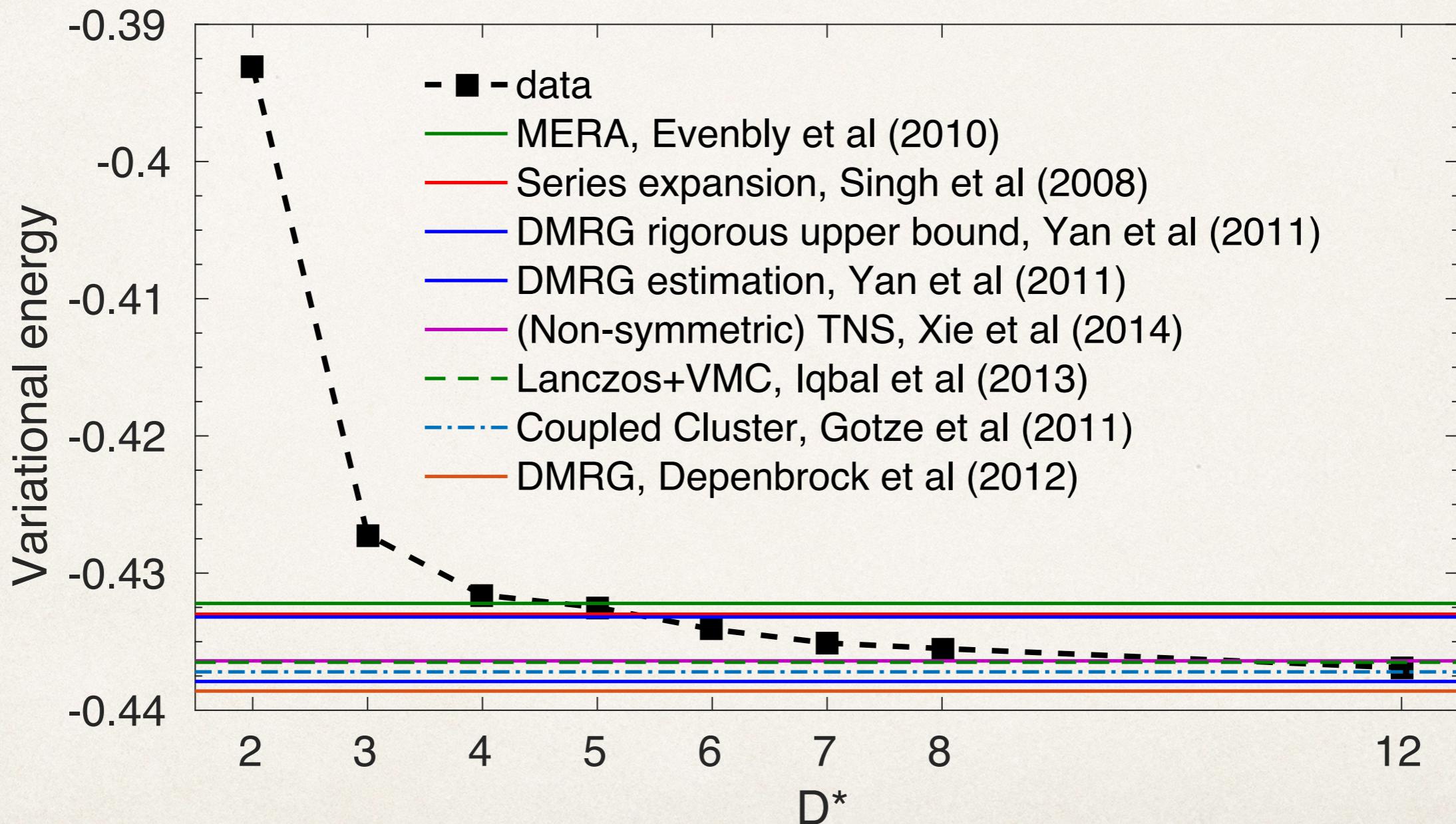
► Symmetric meaning

► **Keeping all degenerate SVs not breaks Symmetry**

Numerical Optimization

❖ Example2 - Symmetric SU

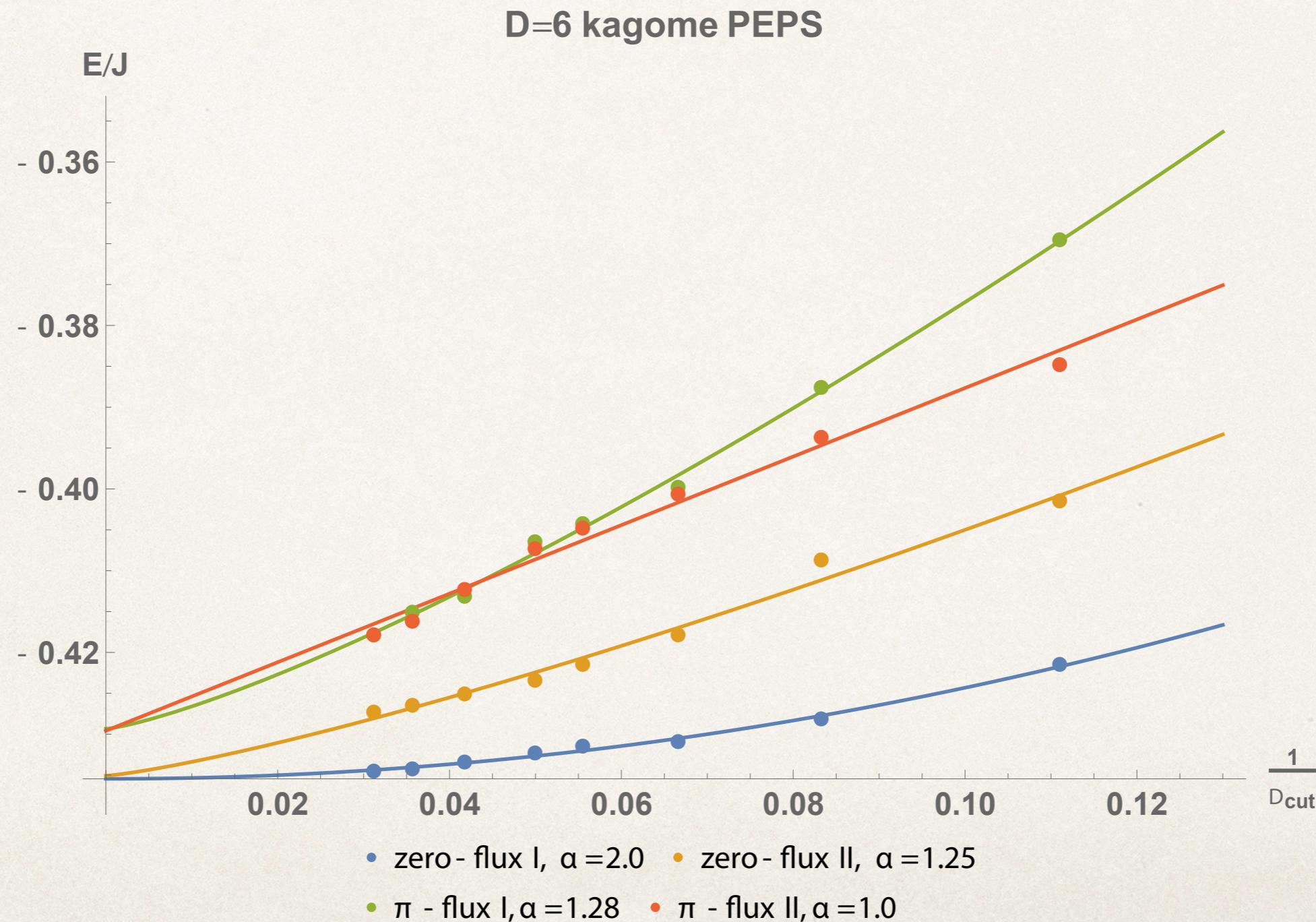
▷ Z2 Gapped Spin Liquid on Kagome [Mei et al. (2017)]



Numerical Optimization

❖ Example3 - Symmetric SU

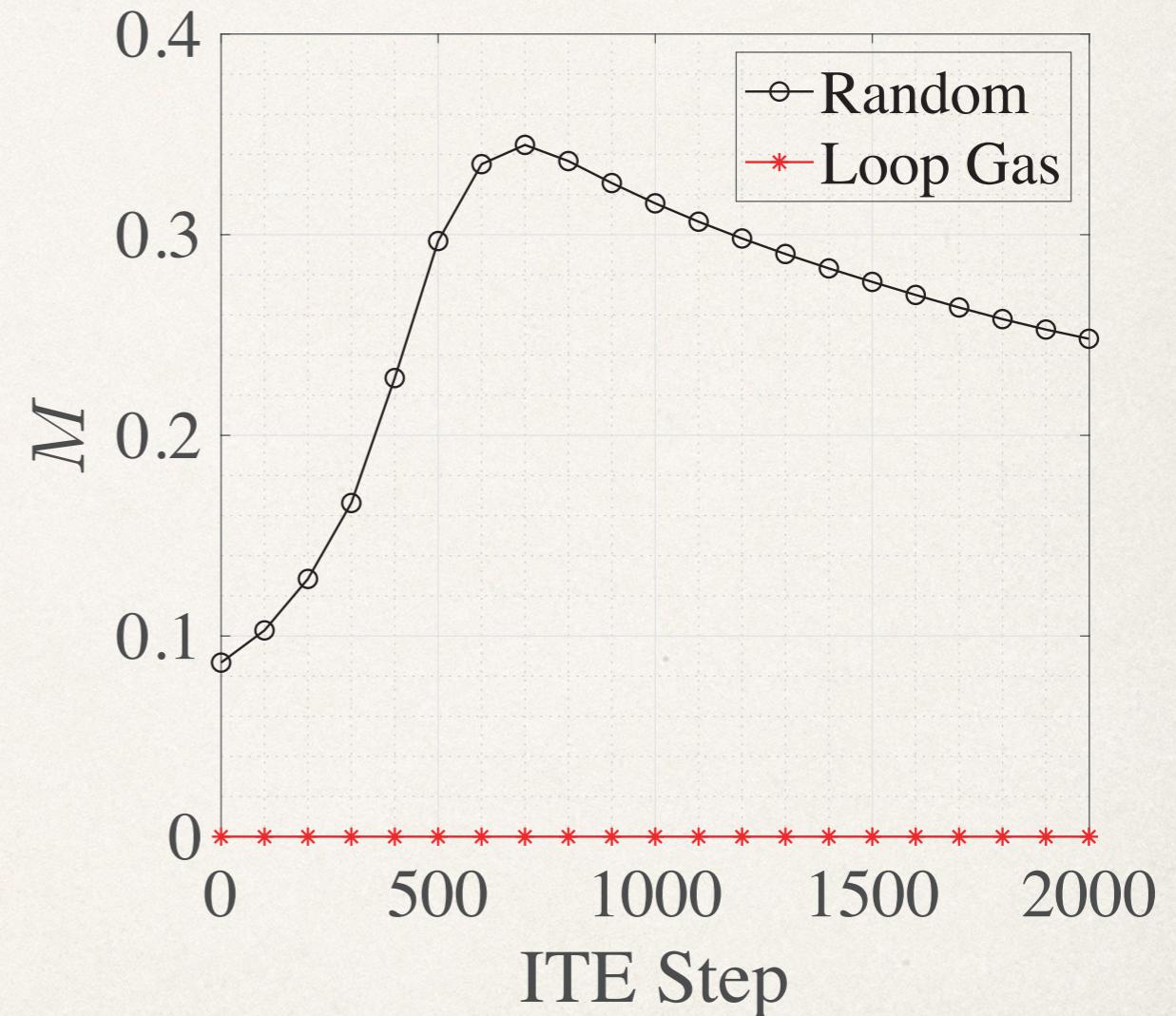
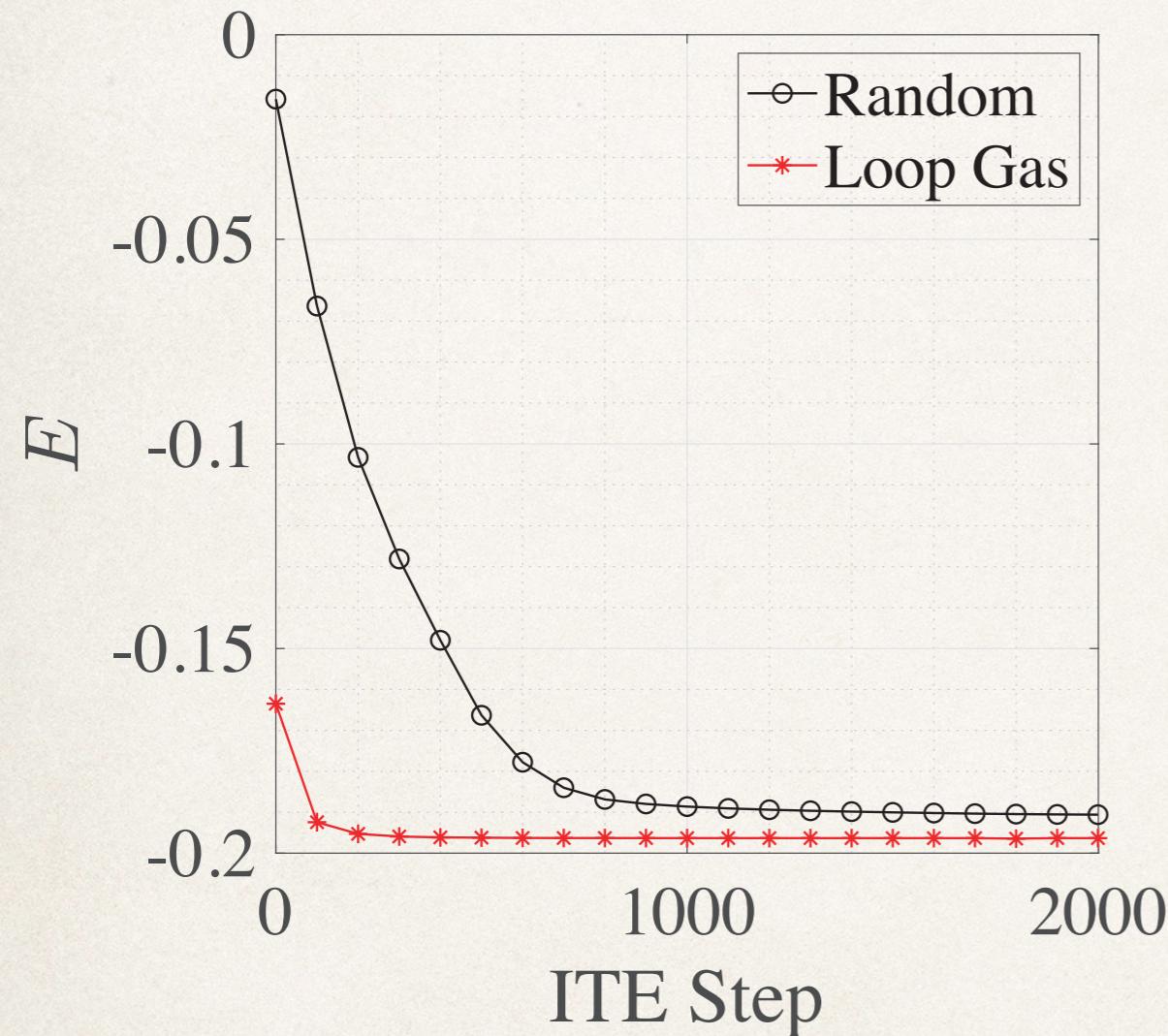
▷ Gapless Spin Liquid on Kagome [S. Jiang et al. (2019)]



Numerical Optimization

❖ Example4 - Symmetric SU

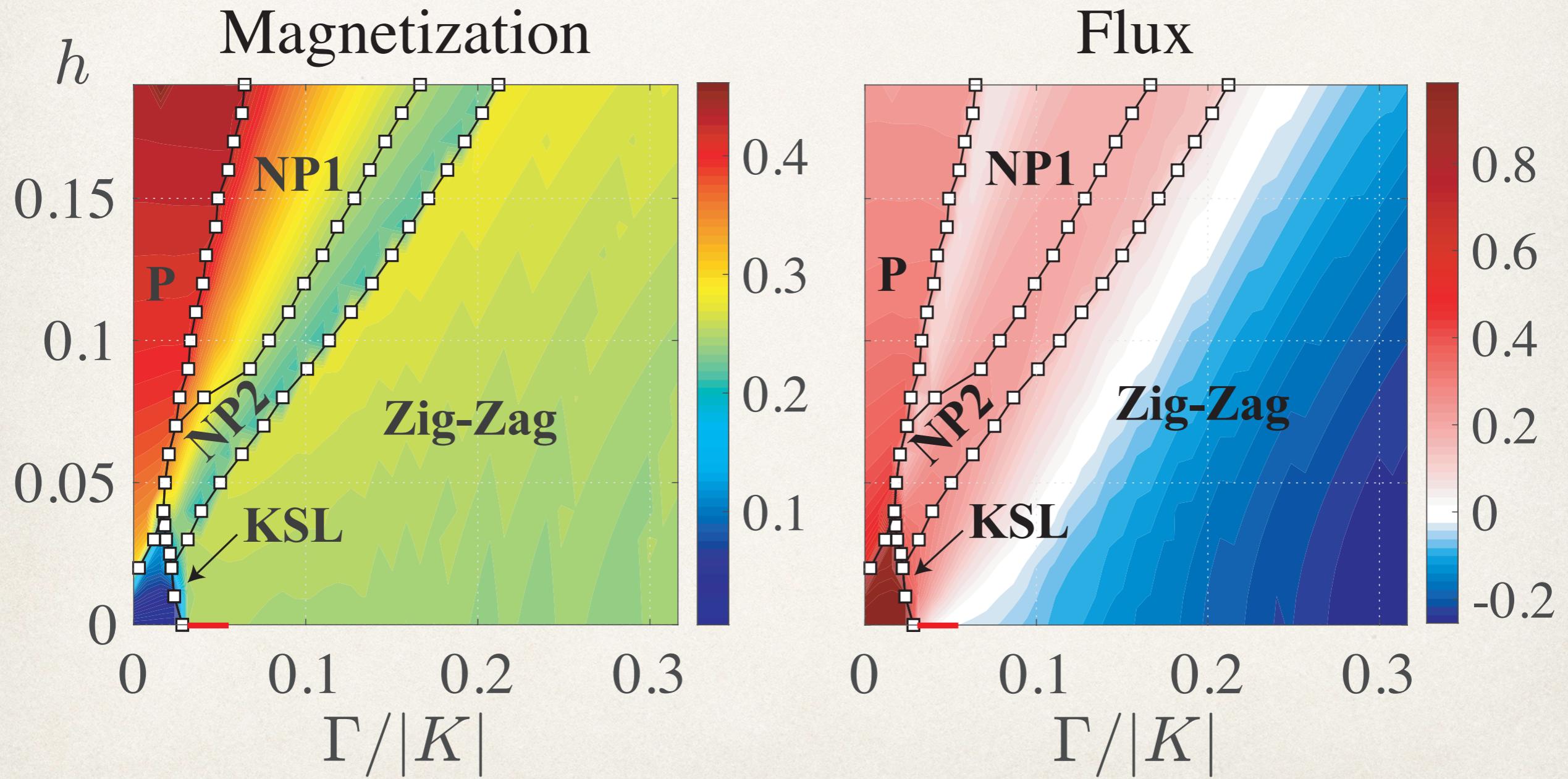
▷ Kitaev Spin Liquid **cannot** be obtained from Random state



Numerical Optimization

❖ Example5 - Symmetric SU

▷ Quantum Phase Diagram of K-G-G' model [H.-Y. Lee et al. (2019)]

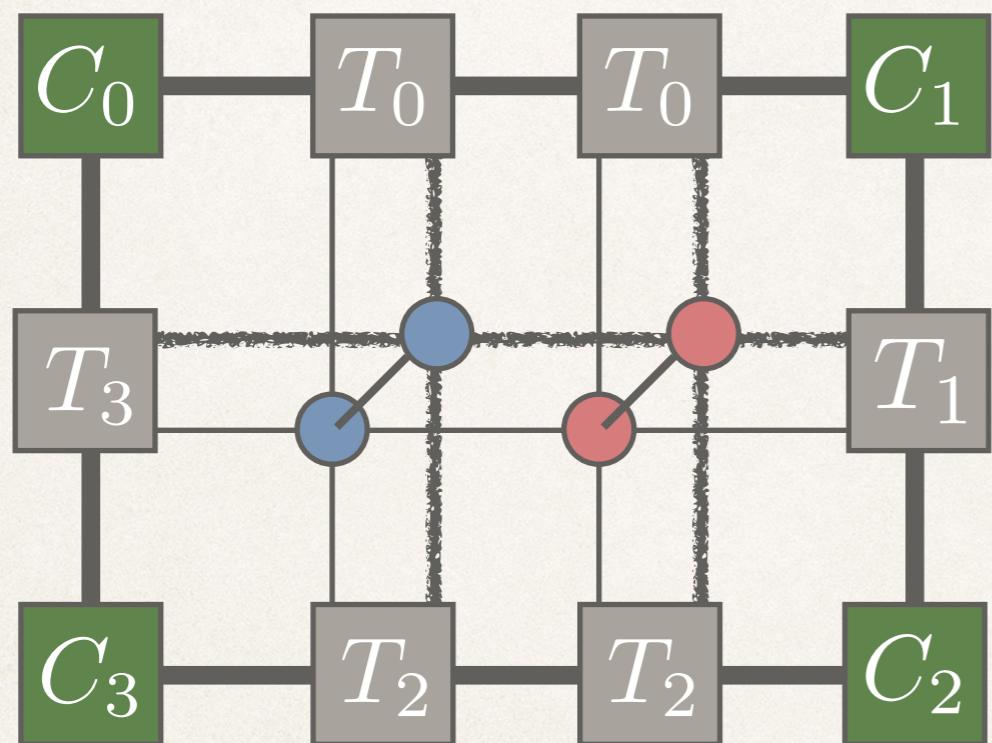


Numerical Optimization

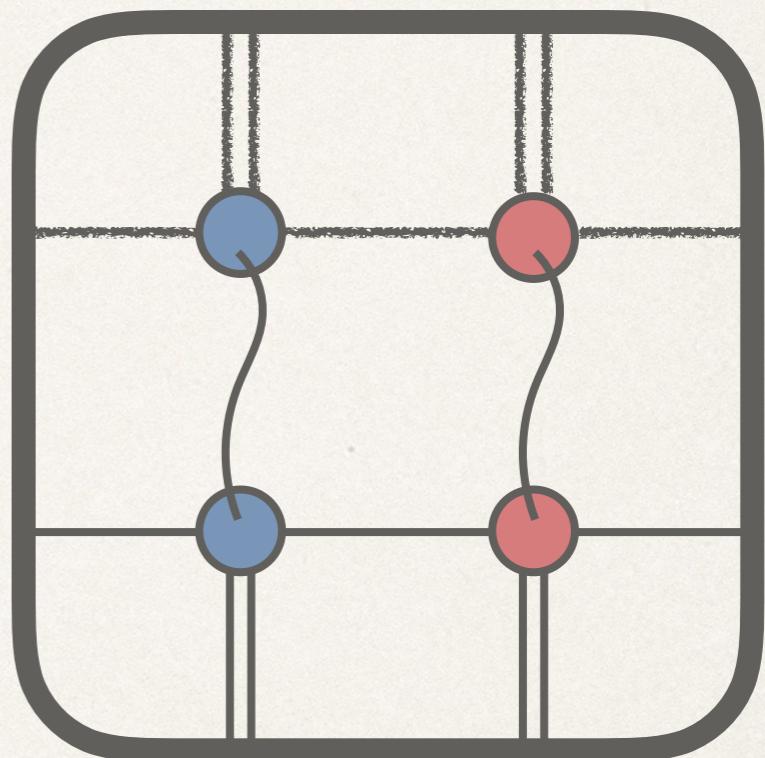
❖ Imaginary Time Evolution

▷ Full Update

- ▶ before started



=

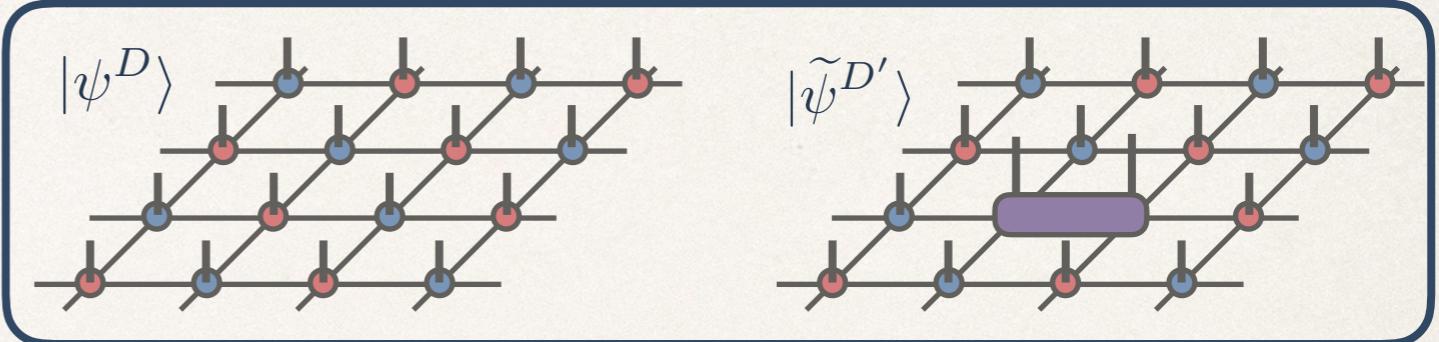


Numerical Optimization

❖ Imaginary Time Evolution

⇒ Full Update

$$\blacktriangleright |\tilde{\psi}^{D'}\rangle = e^{-\tau \hat{h}} |\psi^D\rangle$$

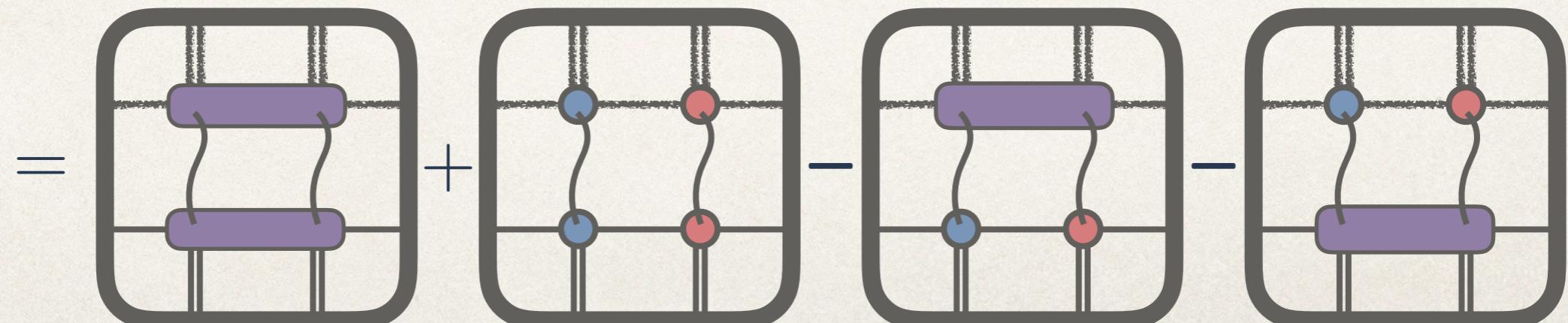


$$\blacktriangleright \text{Goal: } |\tilde{\psi}_n^{D'}\rangle \xrightarrow{\text{trunc}} |\psi_n^D\rangle \text{ such that } |\psi_n^D\rangle \simeq |\tilde{\psi}_n^{D'}\rangle$$

► Cost function:

$$\epsilon = || |\tilde{\psi}_n^{D'}\rangle - |\psi_n^D\rangle ||^2$$

$$= \langle \tilde{\psi}_n^{D'} | \tilde{\psi}_n^{D'} \rangle + \langle \psi_n^D | \psi_n^D \rangle - \langle \tilde{\psi}_n^{D'} | \psi_n^D \rangle - \langle \psi_n^D | \tilde{\psi}_n^{D'} \rangle$$



Numerical Optimization

❖ Imaginary Time Evolution

▷ Full Update

► Minimizing Cost function:

$$\frac{\partial \epsilon}{\partial T_{\text{red}}^*} = A T_{\text{red}} - B \xrightarrow{\text{want}} 0$$
$$= \frac{A_{(l_0, u_0, r_0, d_0, p_0)|(l_1, u_1, r_1, d_1, p_1)} [T_{\text{red}}]_{(l_1, u_1, r_1, d_1, p_1)}}{\text{Matrix}} - \frac{B_{(l_0, u_0, r_0, d_0, p_0)}}{\text{Vector}}$$

► Solve linear equation $A v = B$

→ $[T_{\text{red}}^{\text{new}}]_{lrud}^p = v_{(l, r, u, d, p)}$ → Replace T_{red} by $T_{\text{red}}^{\text{new}}$

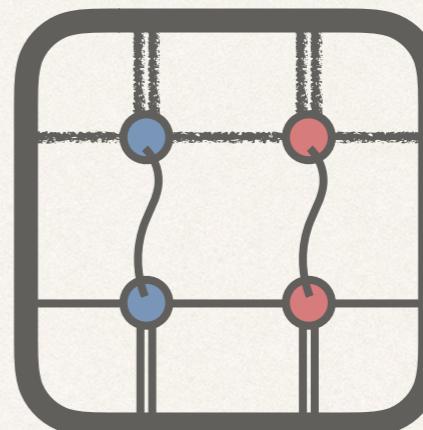
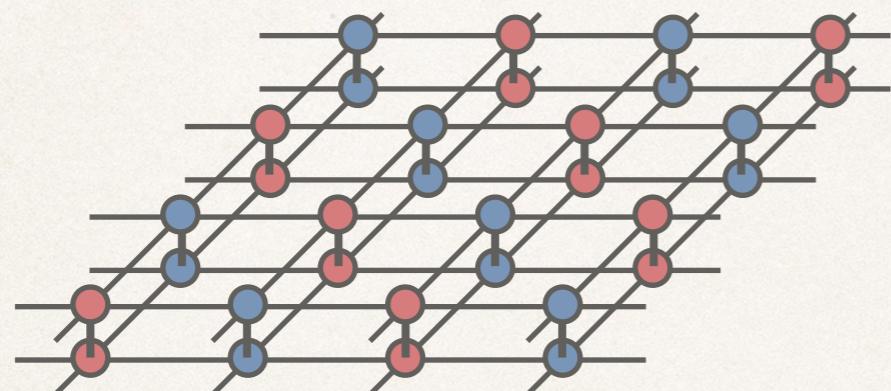
Env. should be computed at each step
→ expensive, but accurate
→ Fast FU algorithm [Orus et al. (2015)]

Numerical Optimization

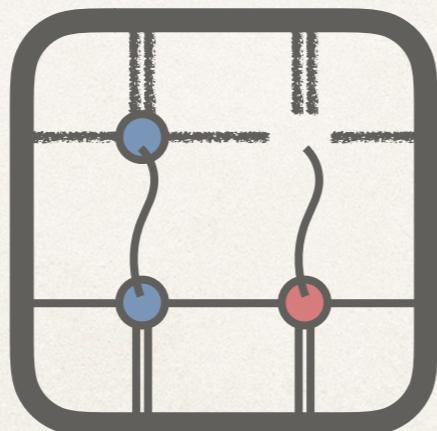
❖ Imaginary Time Evolution

⇒ Full Update

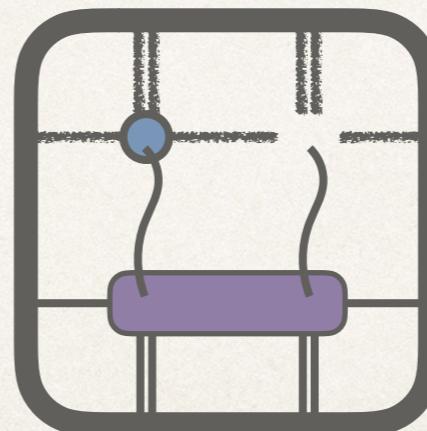
- • Step1: Calculating environment tensors



- Step2: Solving linear equation



=



- • Step3: Replace all tensors by solution tensor

Thank you very much!