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

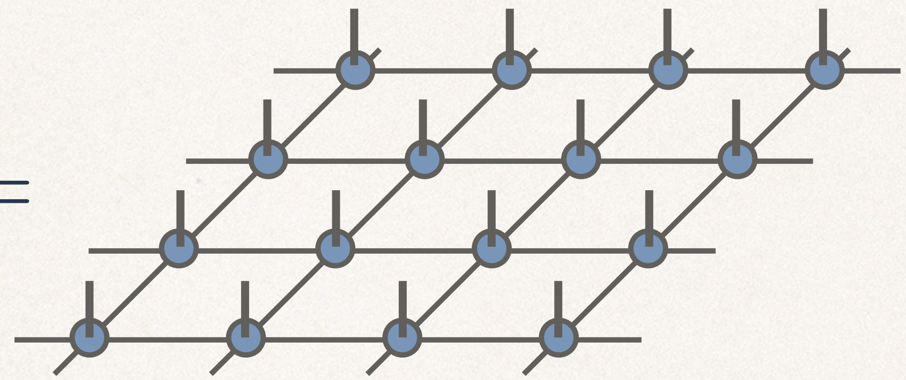
Algorithms for Optimization

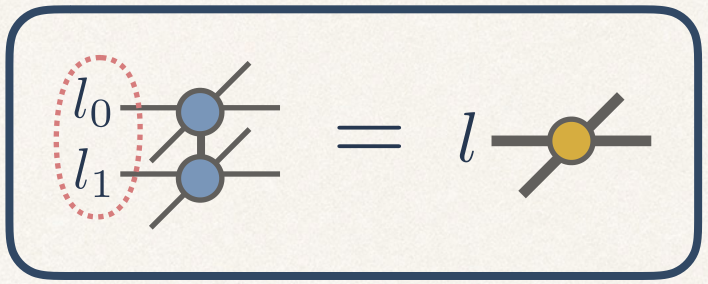
- Exact Constructions
- Numerical Optimizations

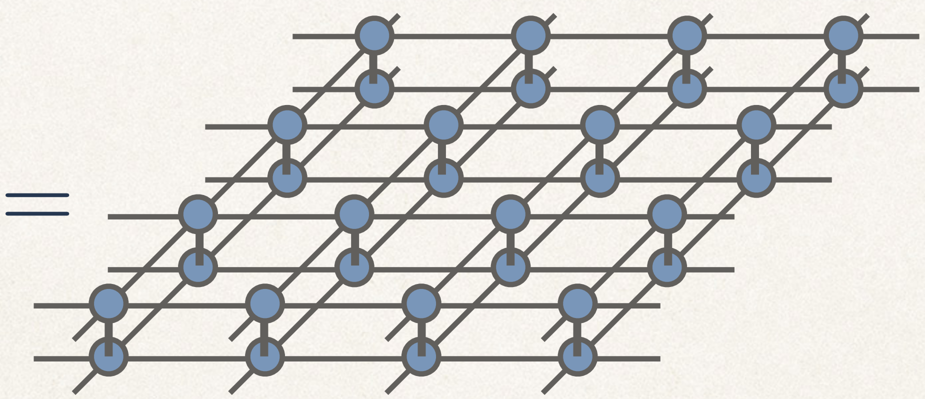
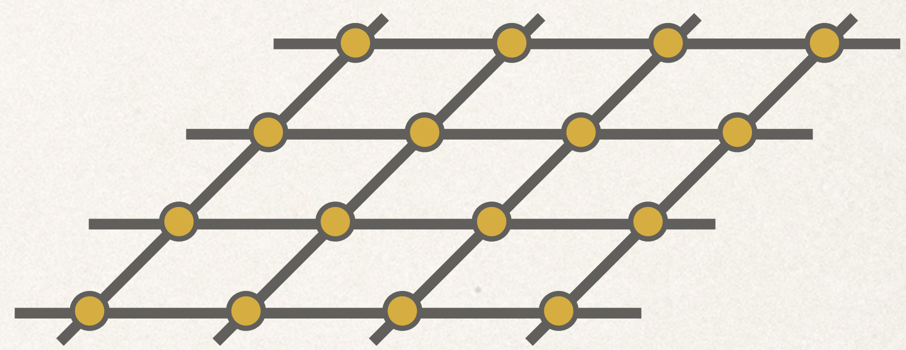
Algorithms for Measurement

- Corner Transfer Matrix Renormalization Group

Measurement

- Mapping to Classical Stat. Mech.
  - Wavefunction & Norm
    - $|\psi\rangle =$ 

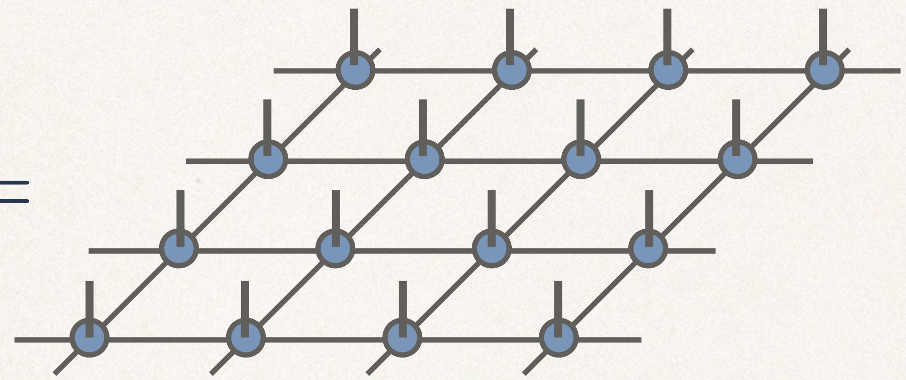


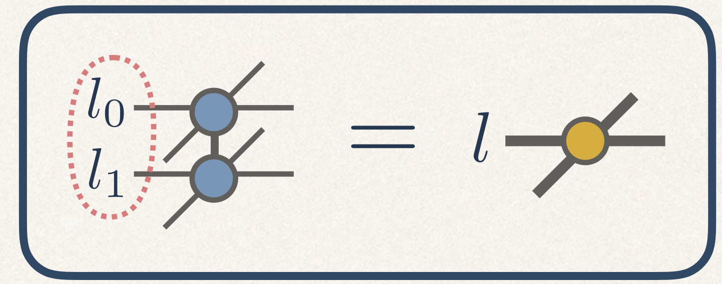
- $\langle\psi|\psi\rangle =$  $=$ 

Norm is Partition function of Classical Stat. Mech.!

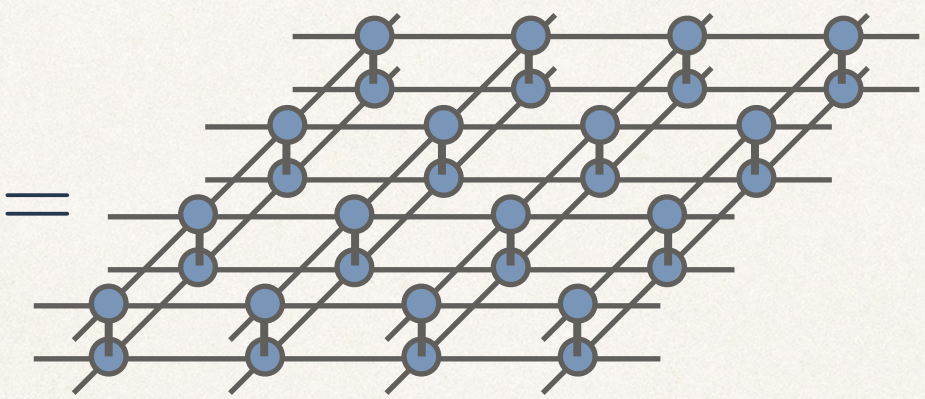
Measurement

- Mapping to Classical Stat. Mech.
  - Wavefunction & Norm
    - $|\psi\rangle =$

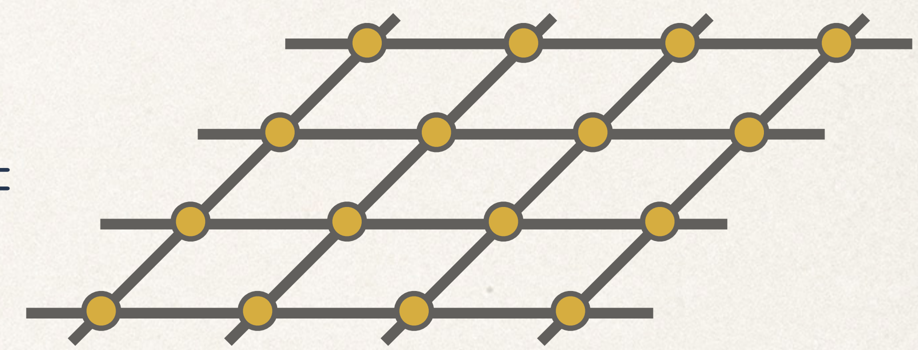




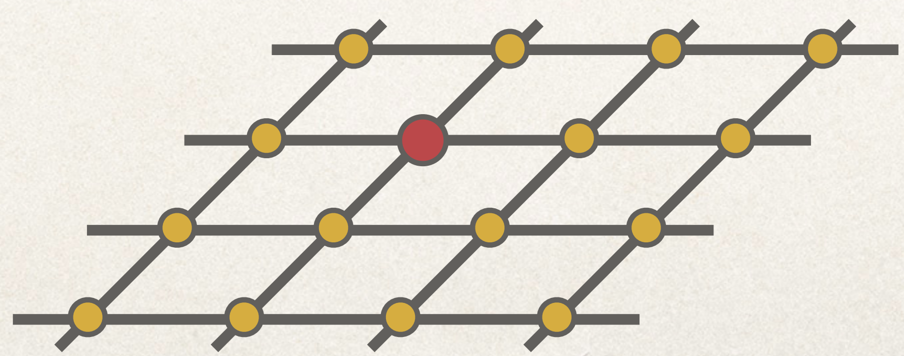
    - $\langle\psi|\psi\rangle =$

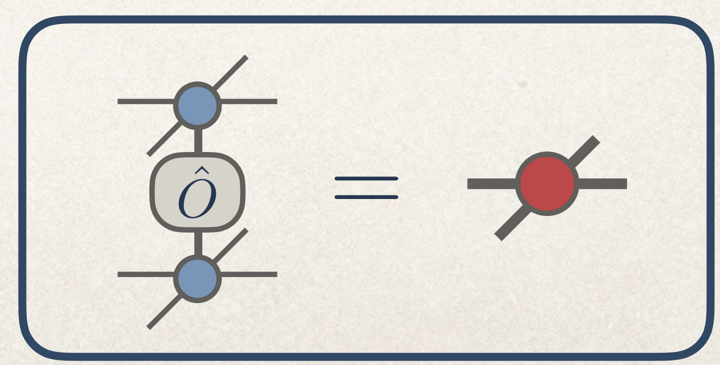


$=$



  - Expectation value
    - $\langle\psi|\hat{O}_i|\psi\rangle =$





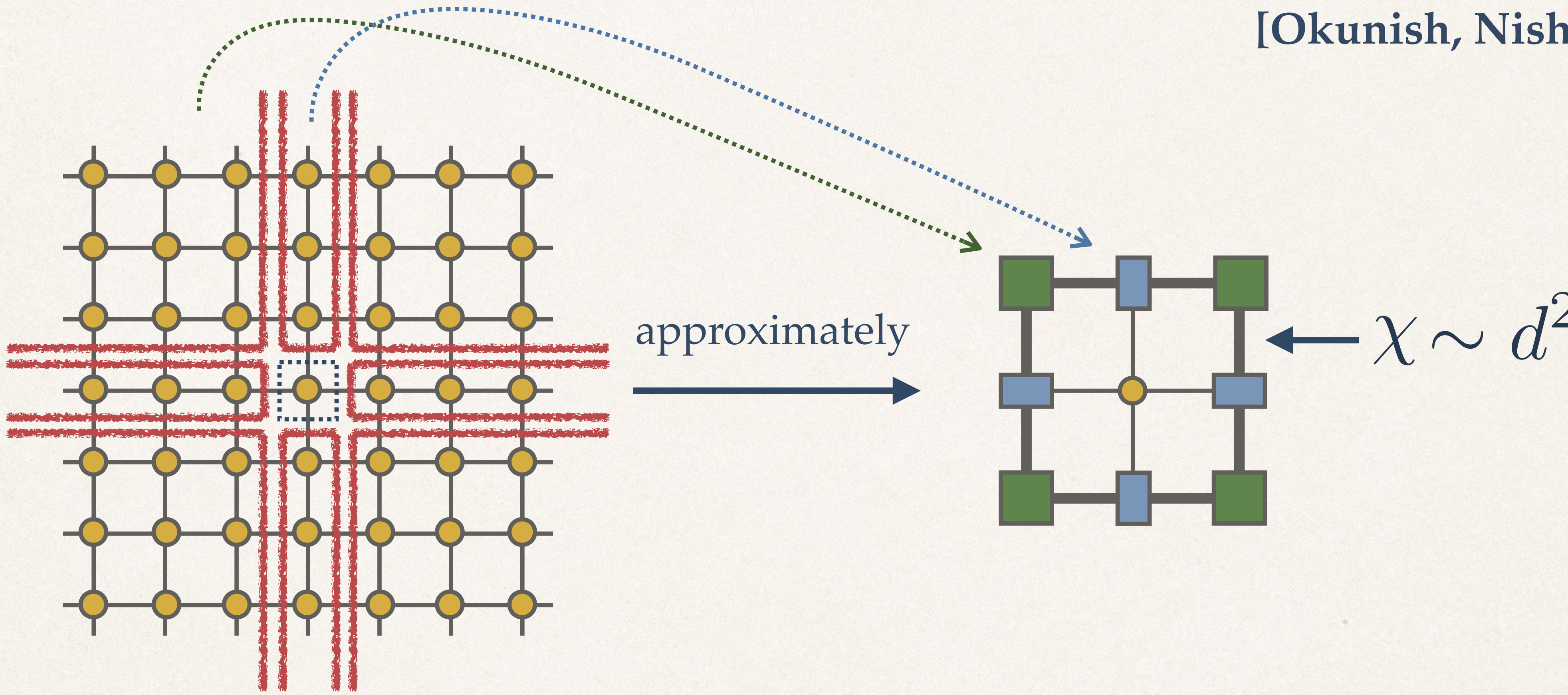
Measurement

❖ Corner Transfer Matrix Renormalization Group

➮ Environment tensors effectively represent Infinite Tensor Network

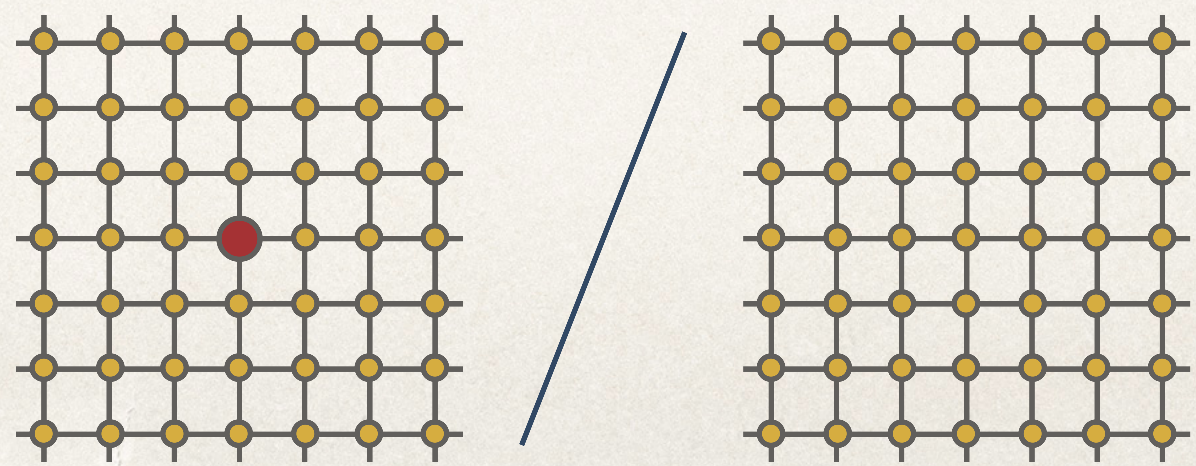
[Okunish, Nishino (1996)]

▸



▸ Useful for TPS measurement

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



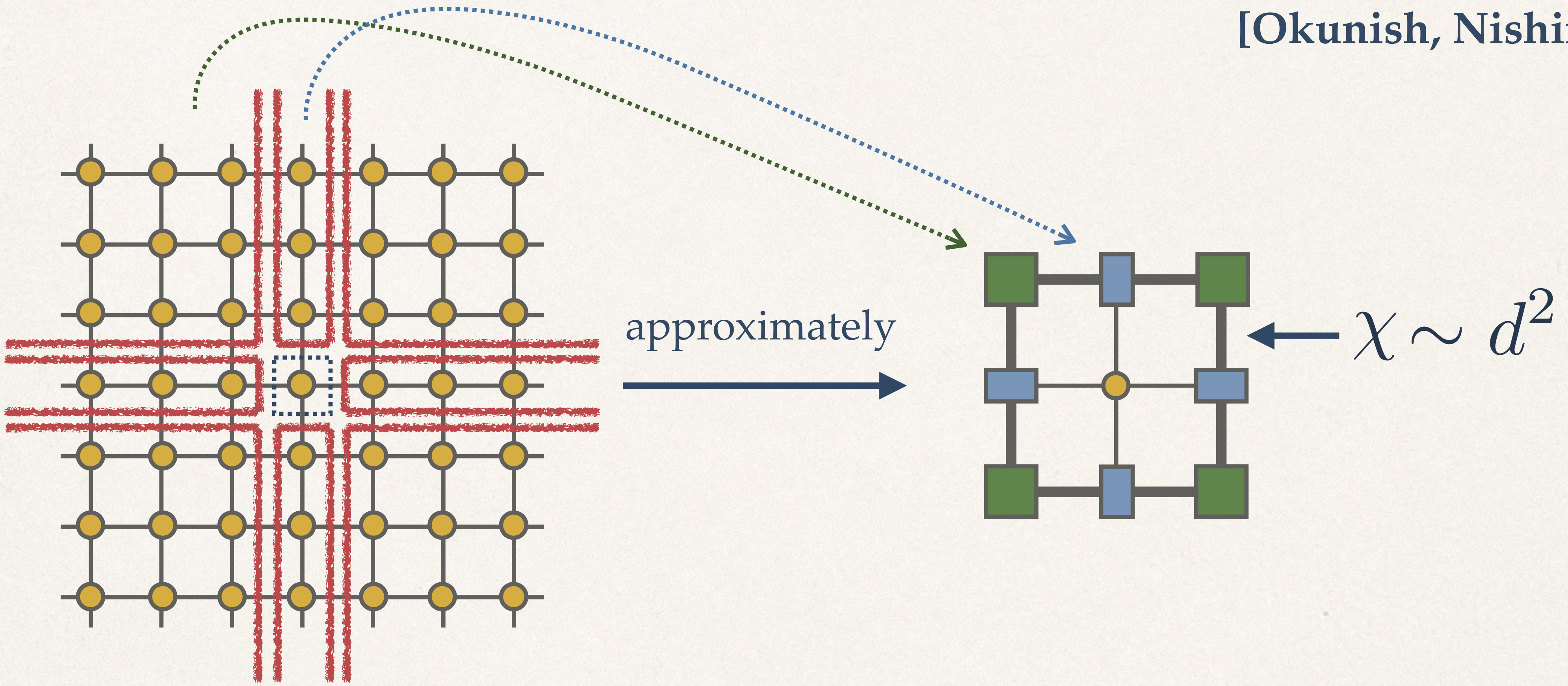
Measurement

❖ Corner Transfer Matrix Renormalization Group

➪ Environment tensors effectively represent Infinite Tensor Network

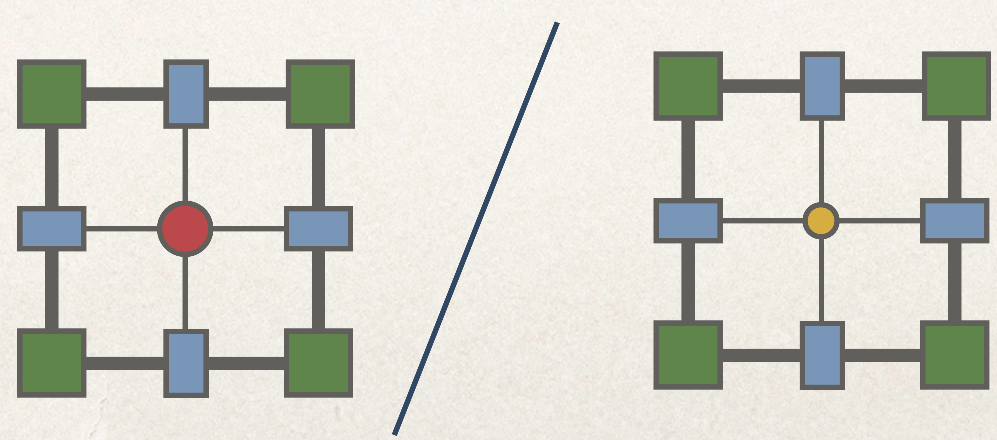
[Okunish, Nishino (1996)]

▸



▸ Useful for TPS measurement

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



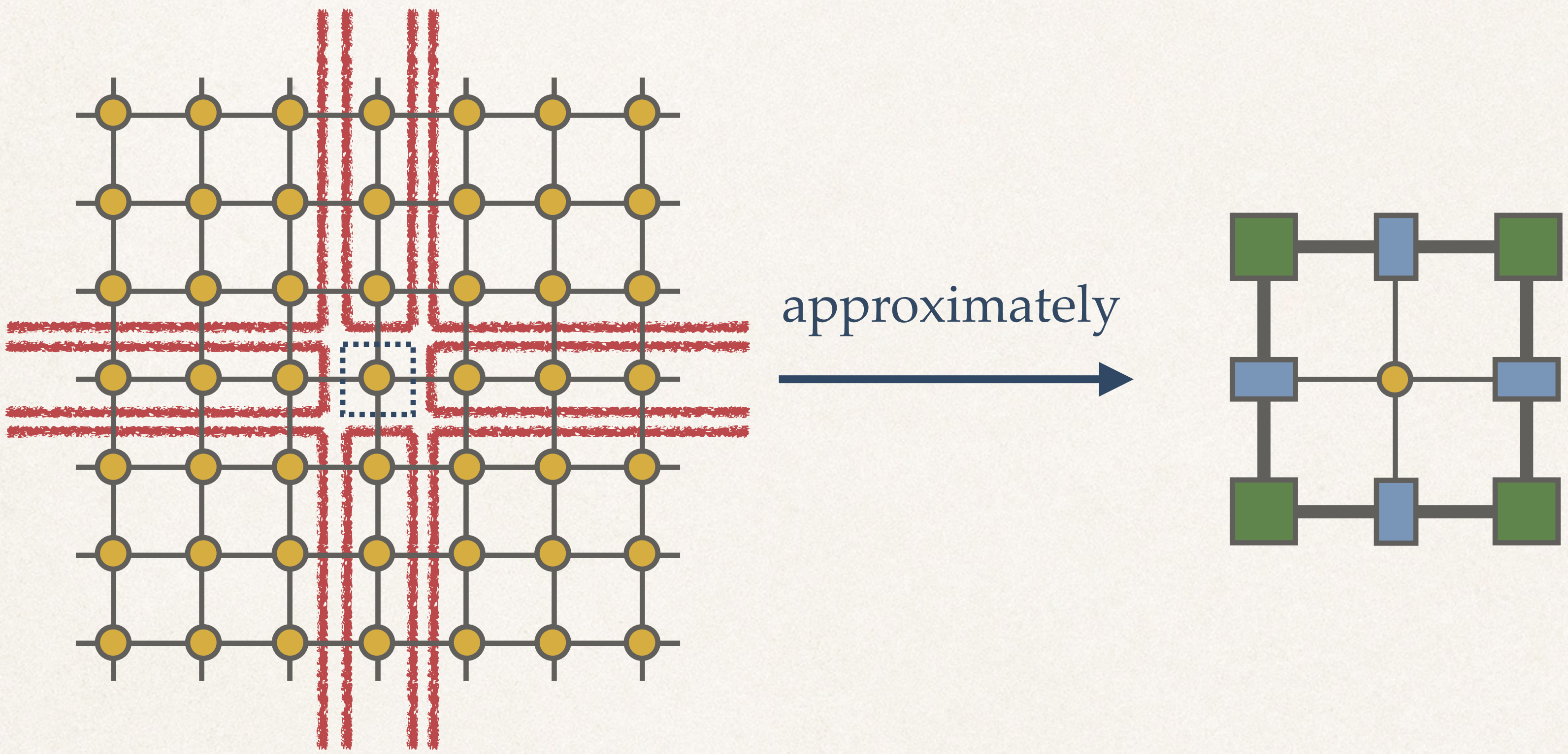
Measurement

❖ Corner Transfer Matrix Renormalization Group

➪ Environment tensors effectively represent Infinite Tensor Network

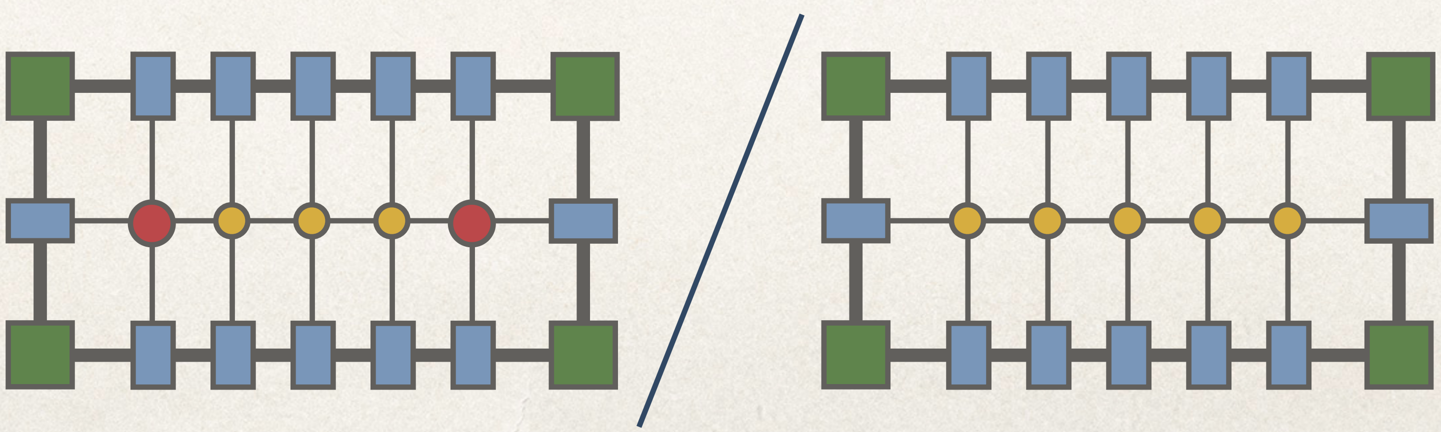
[Okunish, Nishino (1996)]

▸



▸ Useful for TPS measurement

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

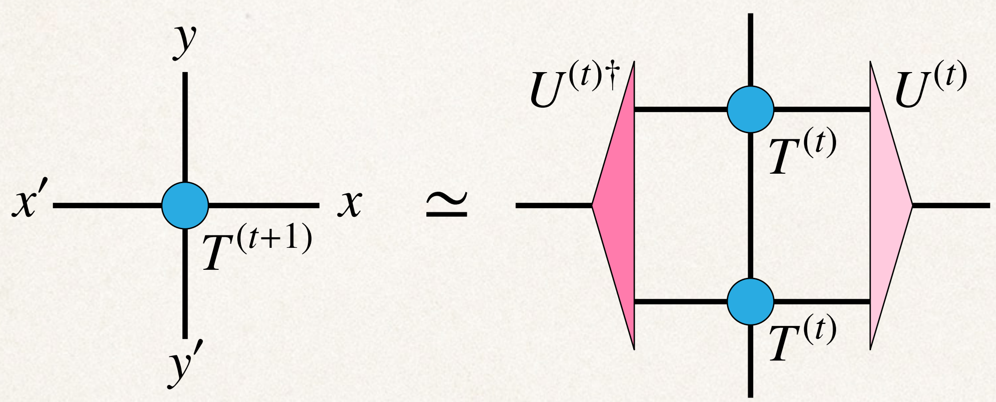


Measurement

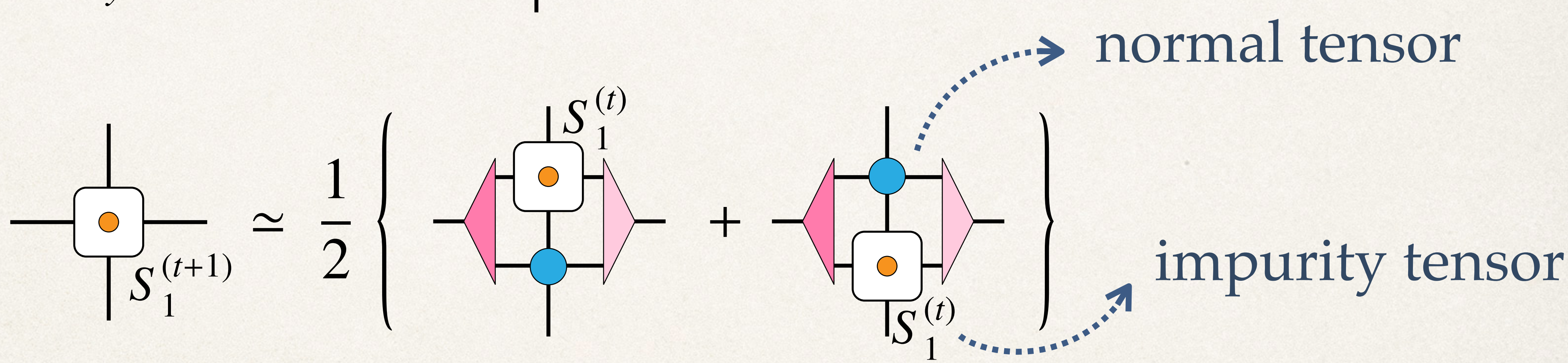
※ HOTRG Implementation **[Satoshi, Naoki (2018)]**

➯ "Impurity" tensor at which the operator is inserted

- Utilize the same isometry for both normal and impurity tensors

- 

: HOTRG step

- 

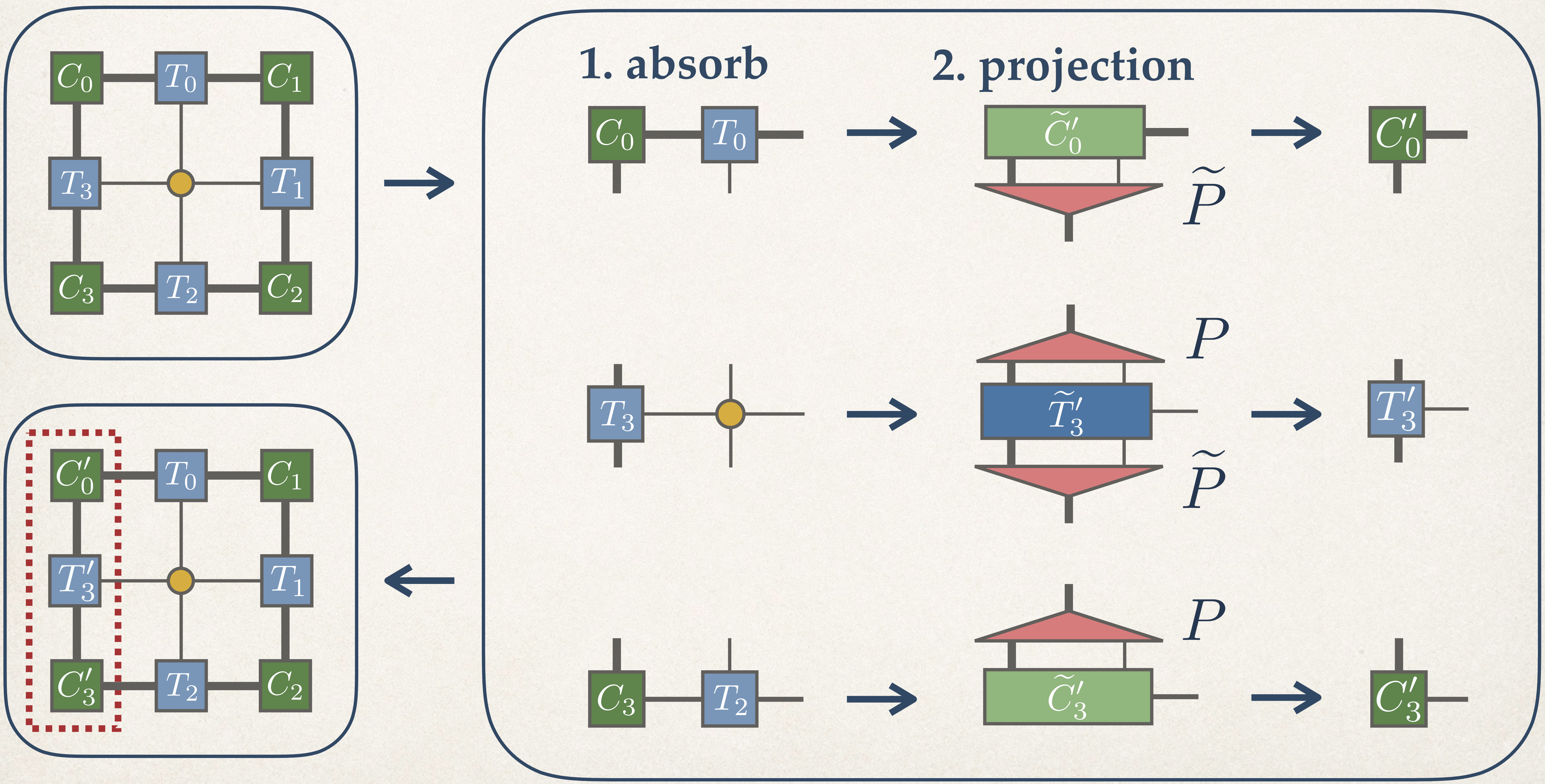
- $$\langle m \rangle = \frac{\mathrm{tTr}(S_1^t)}{\mathrm{tTr}(T^t)}$$

Higher-order moments also available!
complicate for correlation function…

Corner Transfer Matrix Renormalization Group

❖ Algorithm in details

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



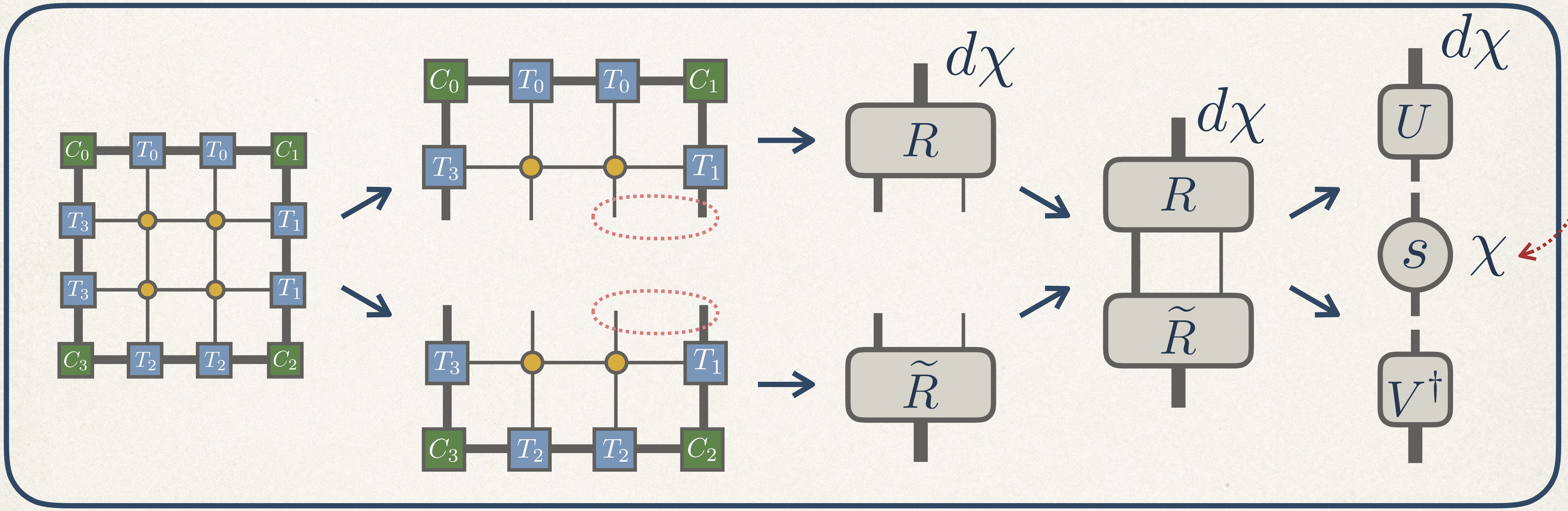
Left env. tensors are updated

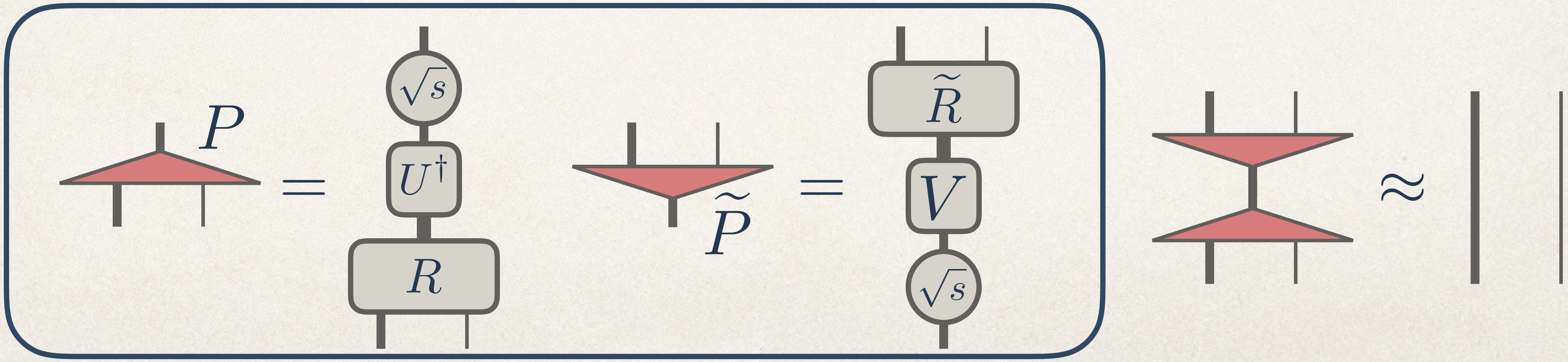
Corner Transfer Matrix Renormalization Group

❖ Algorithm in details

- How to obtain **Projector for Left-RG**

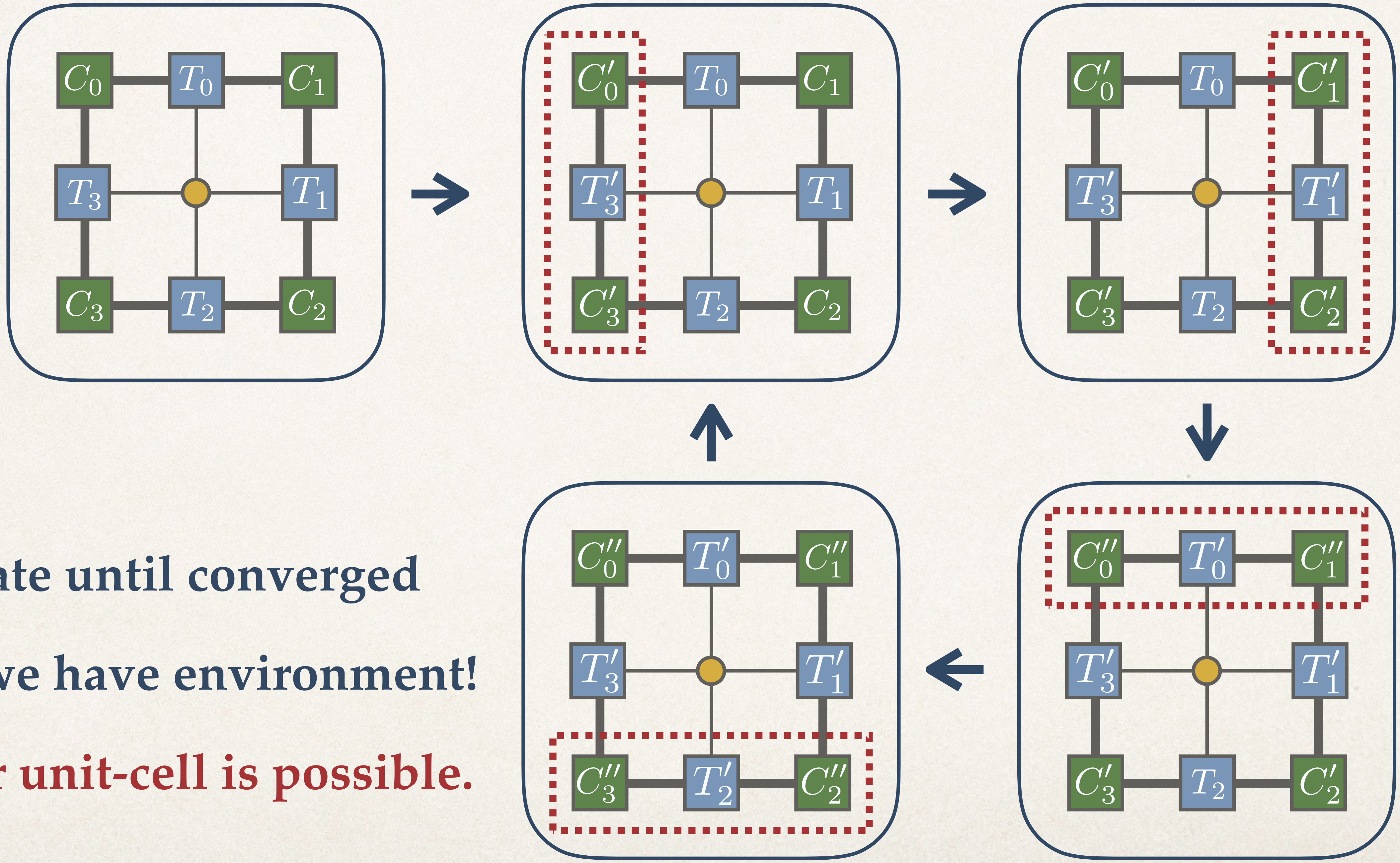
truncation!





Corner Transfer Matrix Renormalization Group

- ❖ Algorithm in details
  - Complete RG procedure



Iterate until converged

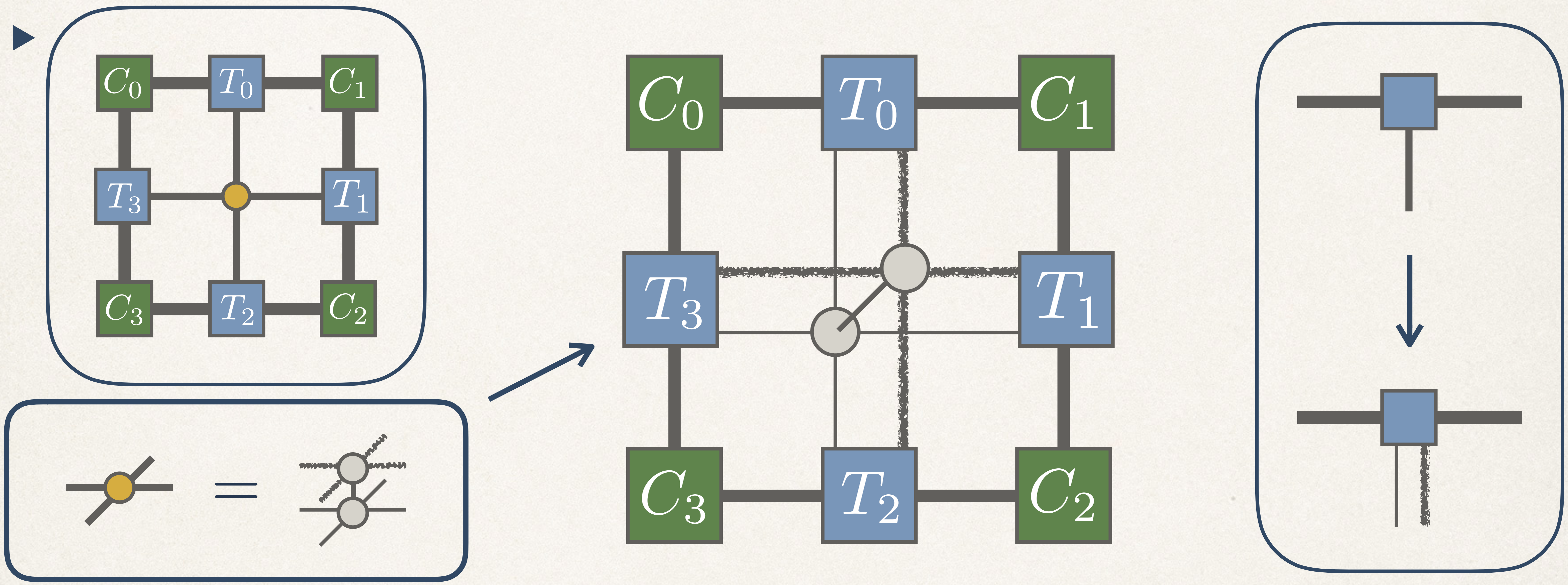
Now, we have environment!

Larger unit-cell is possible.

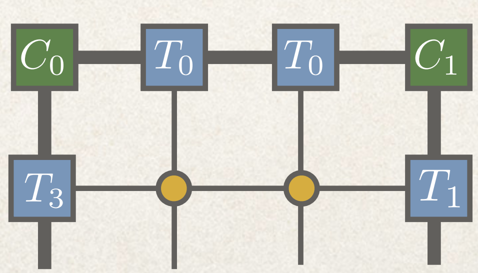
Corner Transfer Matrix Renormalization Group

❖ Algorithm in details

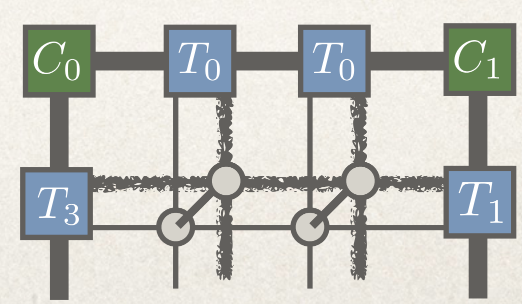
- Tip to reduce complexity

  - 

  - Heavy Part



$\sim d^8 \chi^2$

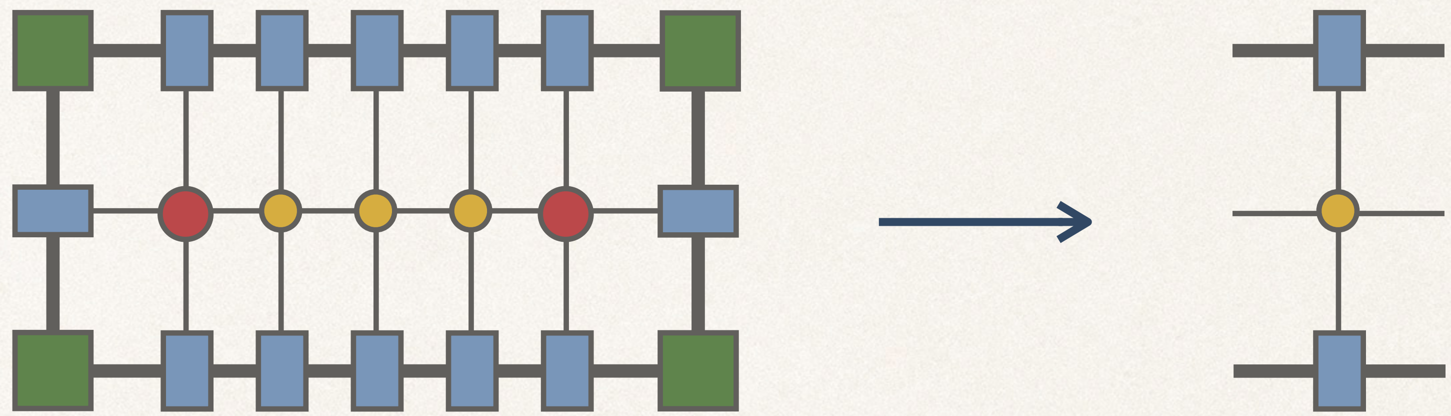


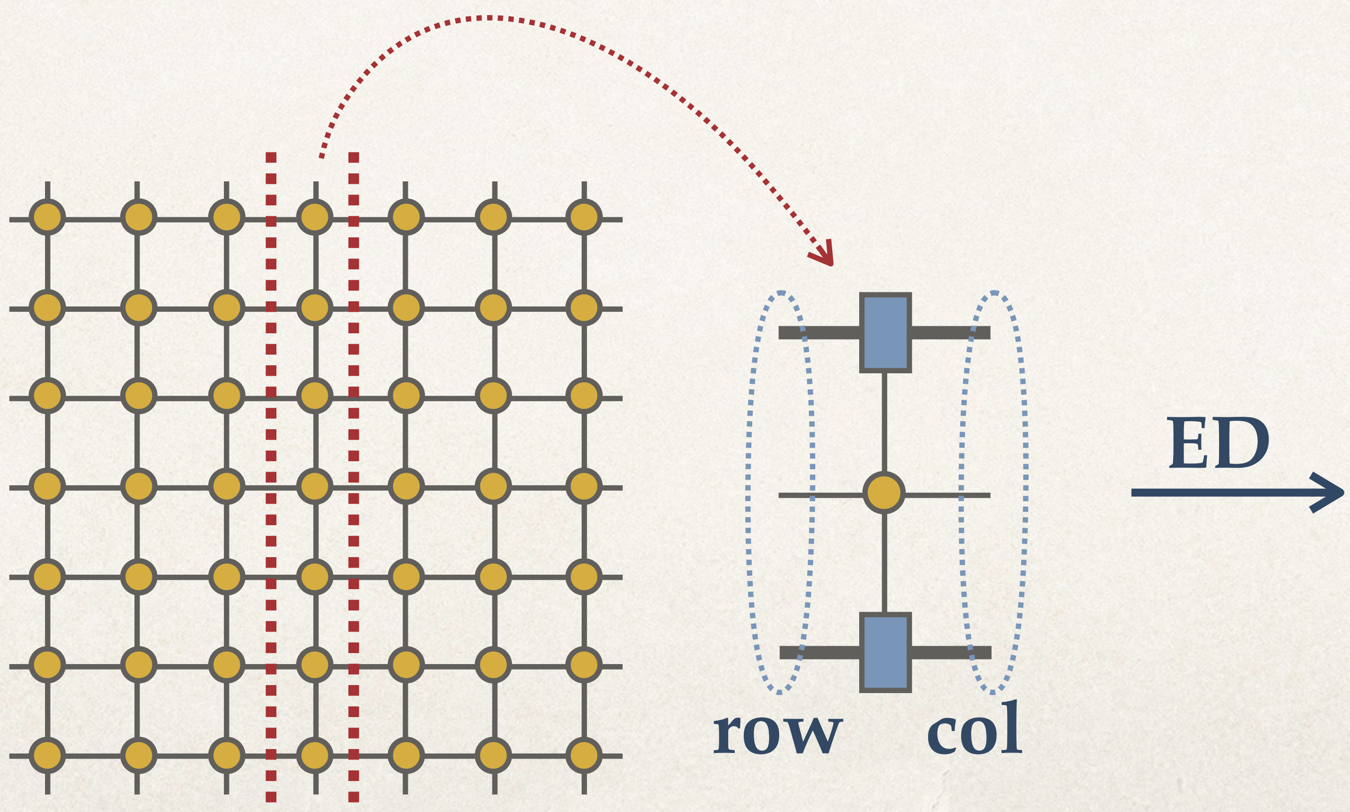
$\sim d^6 \chi^2$

❖ Extracting Correlation Length

- Who delivers the information(or correlation)?

- $\langle \hat{O}_i \hat{O}_j \rangle \sim$





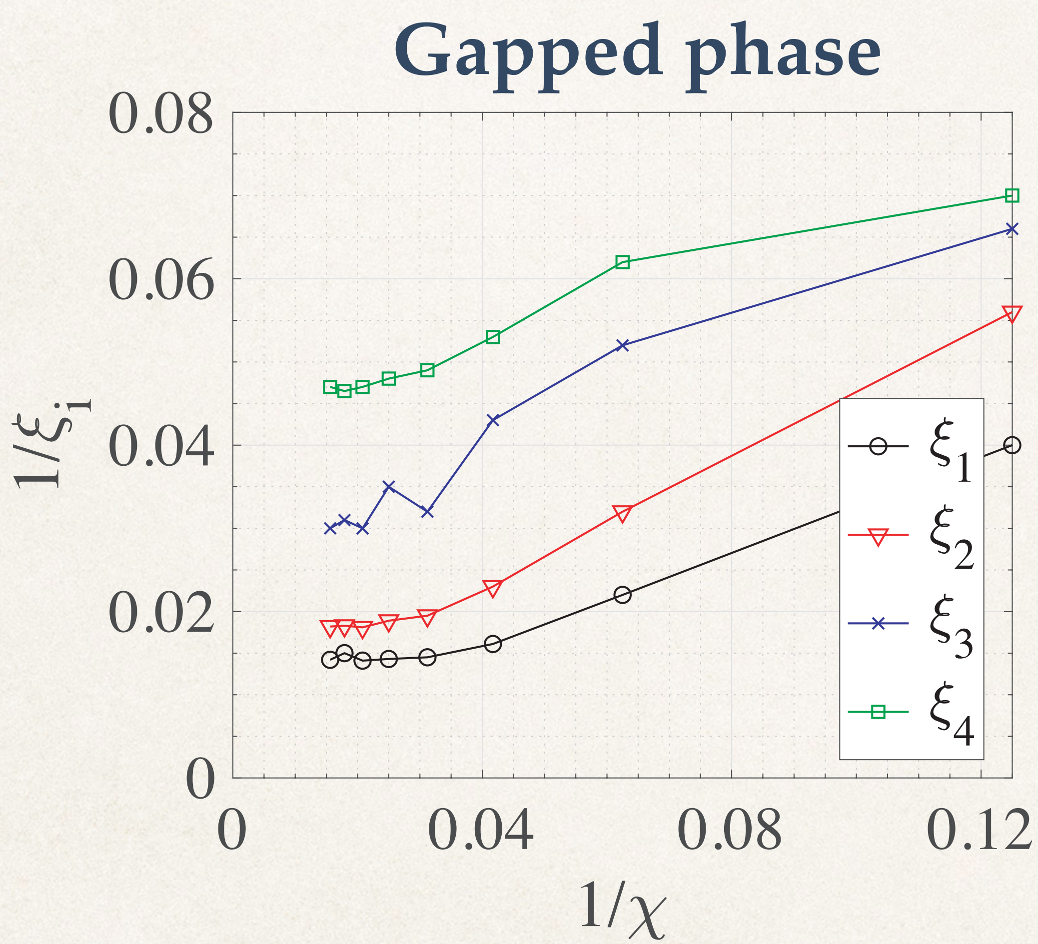
$$\xi = -\log\left(\frac{\lambda_1}{\lambda_0}\right)$$

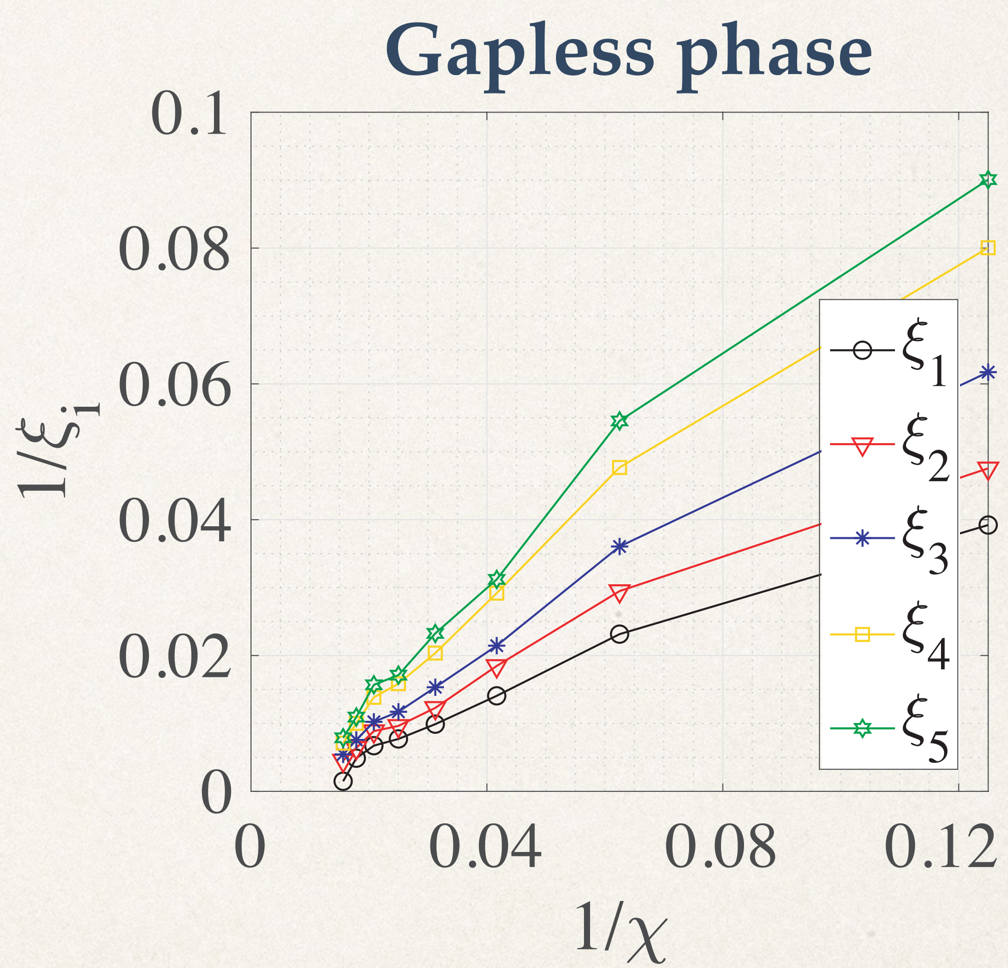
Corner Transfer Matrix Renormalization Group

❖ 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

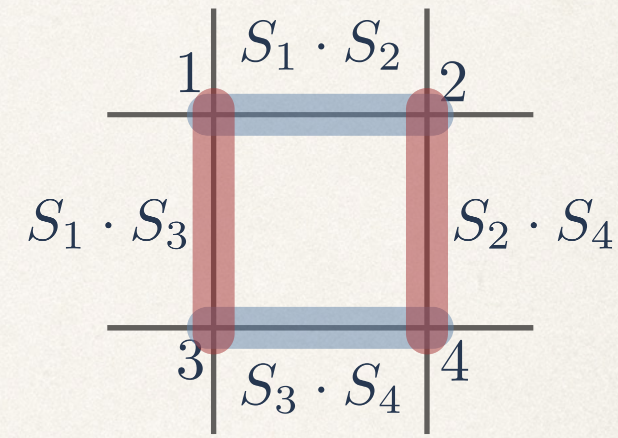
Numerical Optimization

❖ Imaginary Time Evolution

TN representation?

Generally, no…

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



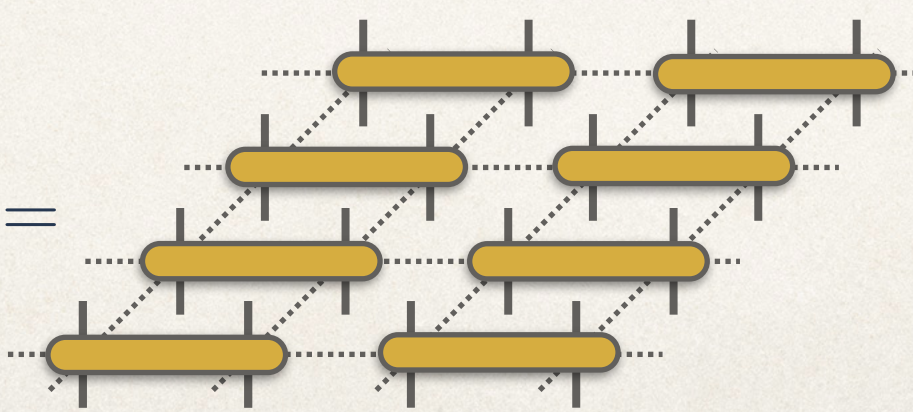
➩ Suzuki-Trotter decomposition

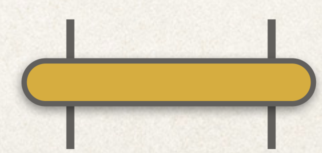
not commute

- 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.)

$e^{-\tau \hat{H}_l} =$ 

$e^{-\tau \hat{h}} =$ 

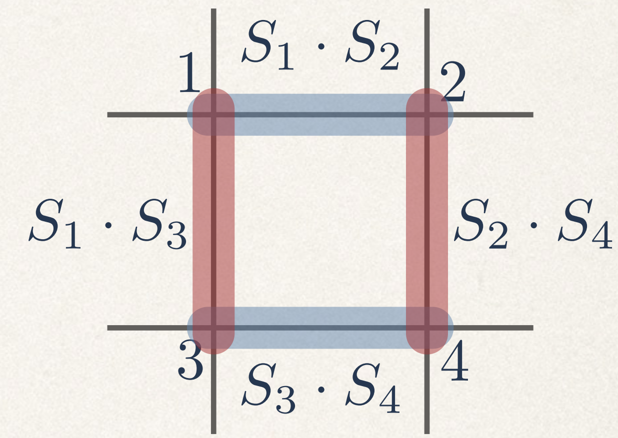
Numerical Optimization

❖ Imaginary Time Evolution

TN representation?

Generally, no…

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



- Suzuki-Trotter decomposition

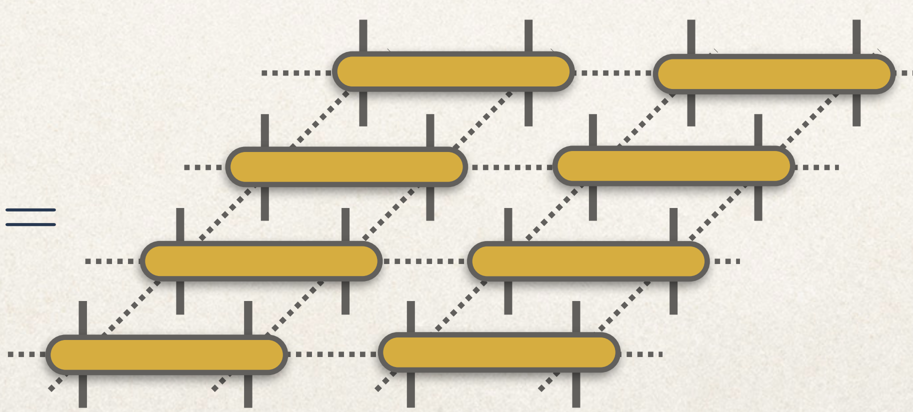
not commute

  - group: $\hat{H} = \hat{H}_l + \hat{H}_u + \hat{H}_r + \hat{H}_d$
  - decouple: $e^{-\tau \hat{H}} \approx e^{-\tau \hat{H}_l} e^{-\tau \hat{H}_u} e^{-\tau \hat{H}_r} e^{-\tau \hat{H}_d}$

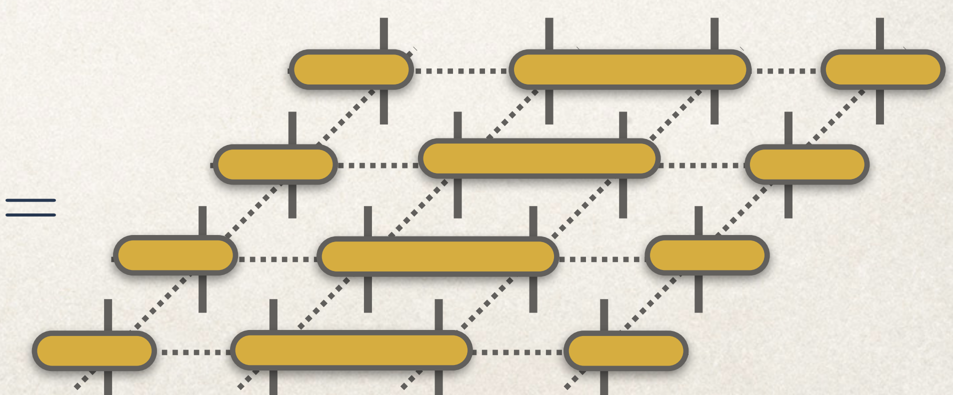
$$e^{-\tau \hat{H}_l} = e^{-\tau \hat{h}_{ab}} \otimes e^{-\tau \hat{h}_{cd}} \otimes e^{-\tau \hat{h}_{cd}} \dots$$

(e.g.)

$e^{-\tau \hat{H}_l} =$



$e^{-\tau \hat{H}_r} =$

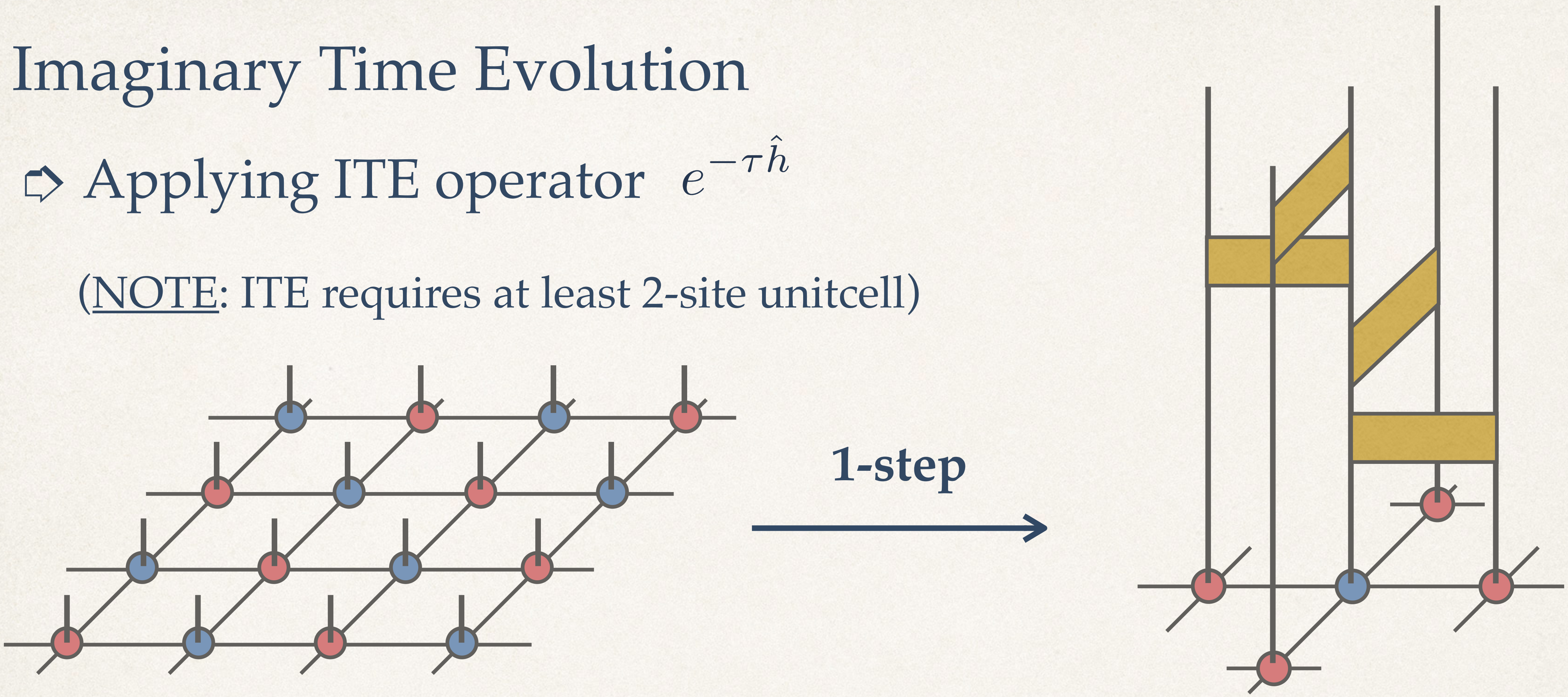


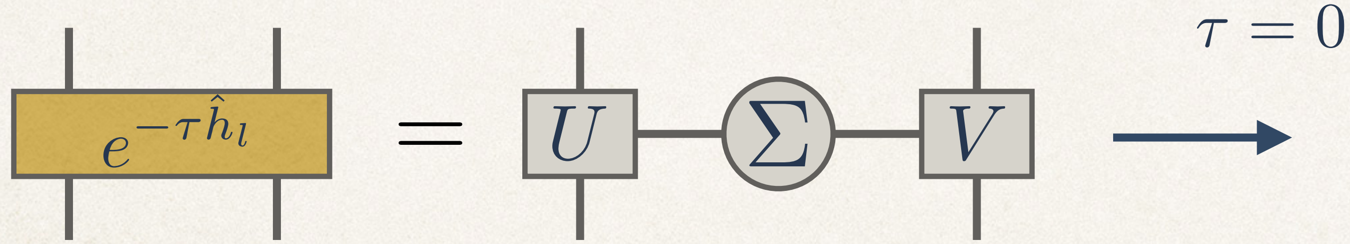
Numerical Optimization

❖ Imaginary Time Evolution

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

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



➯ 

e.g.) Heisenberg $\tau = 0.001$

$$\Sigma \approx \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}$$

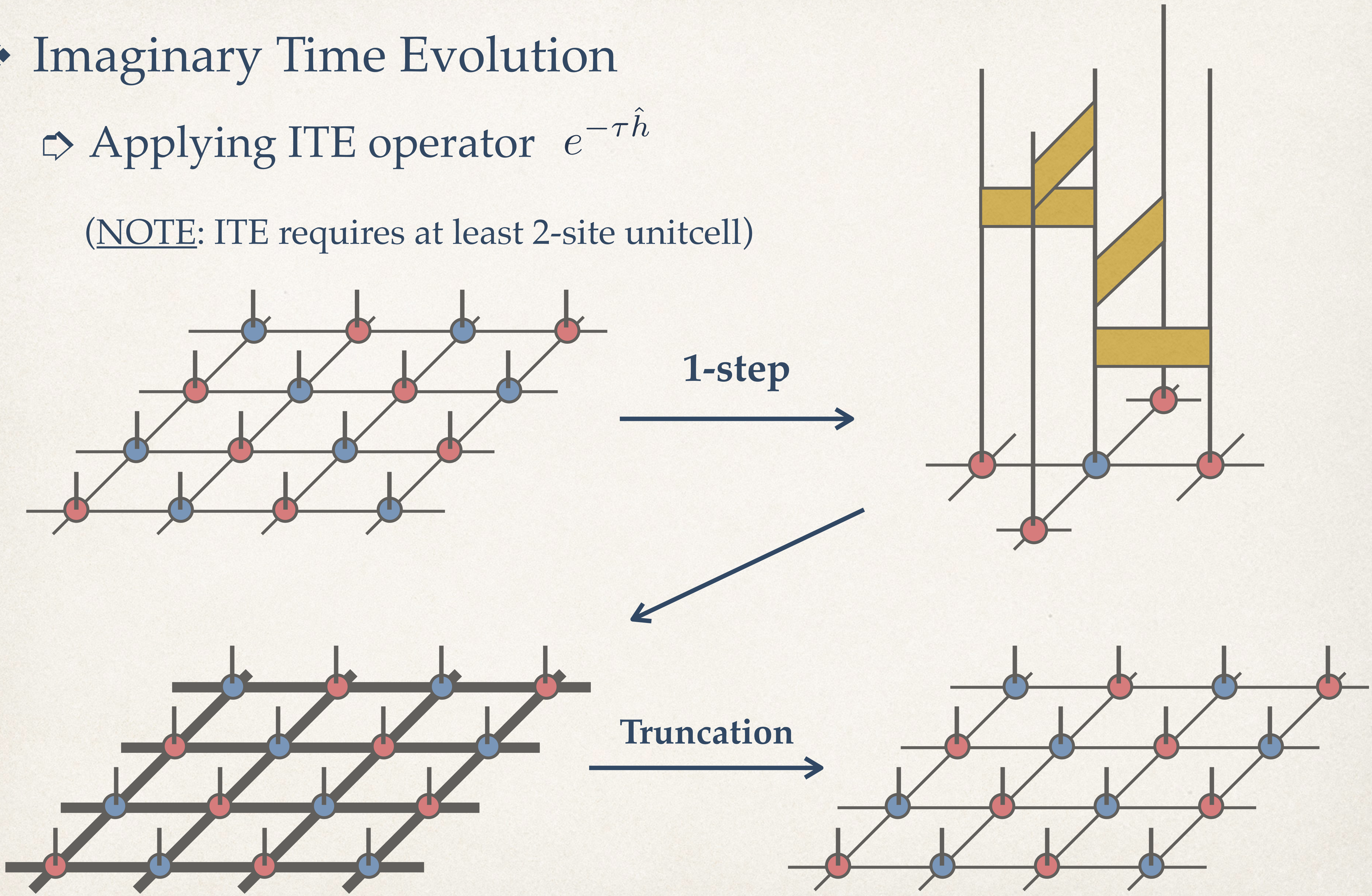
➯ 

Bond dimension increases!

Numerical Optimization

- Imaginary Time Evolution
  - Applying ITE operator $e^{-\tau \hat{h}}$

    (<u>NOTE</u>: ITE requires at least 2-site unitcell)



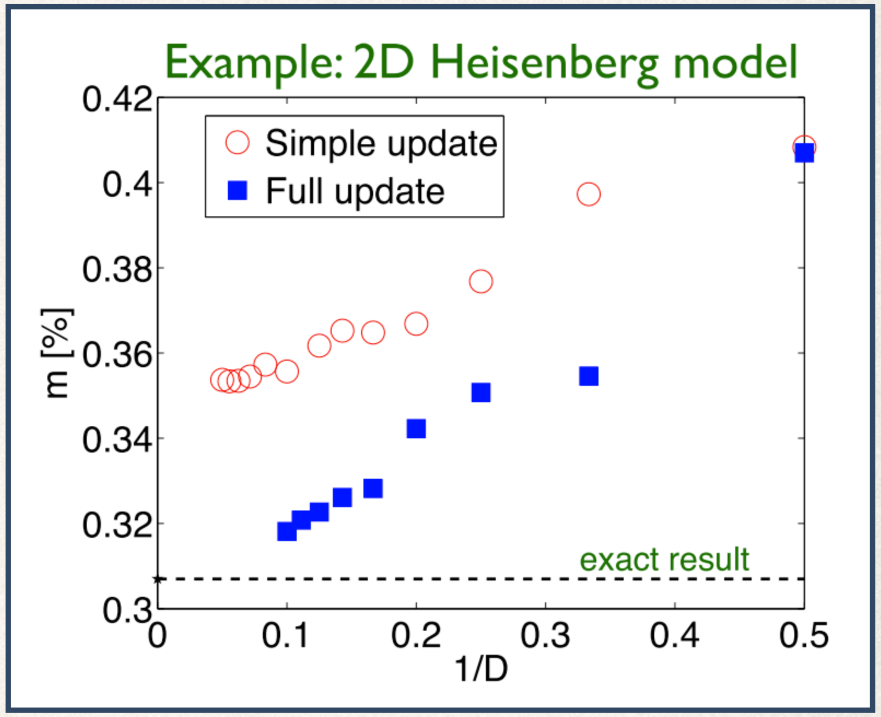
Truncation is required!

Numerical Optimization

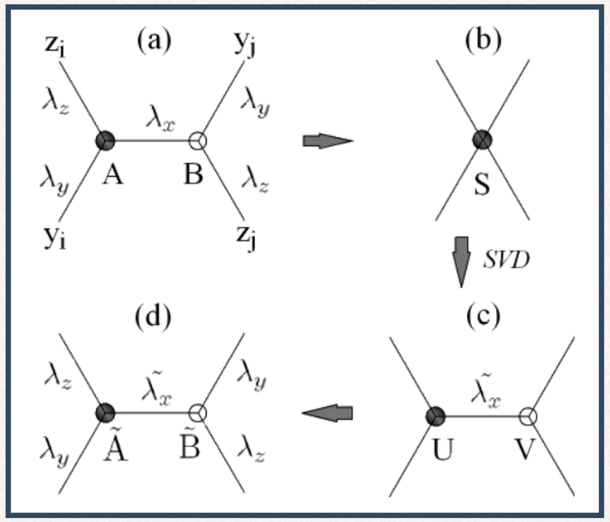
❖ Imaginary Time Evolution

➯ How to truncate?

[P. Corboz (2016)]



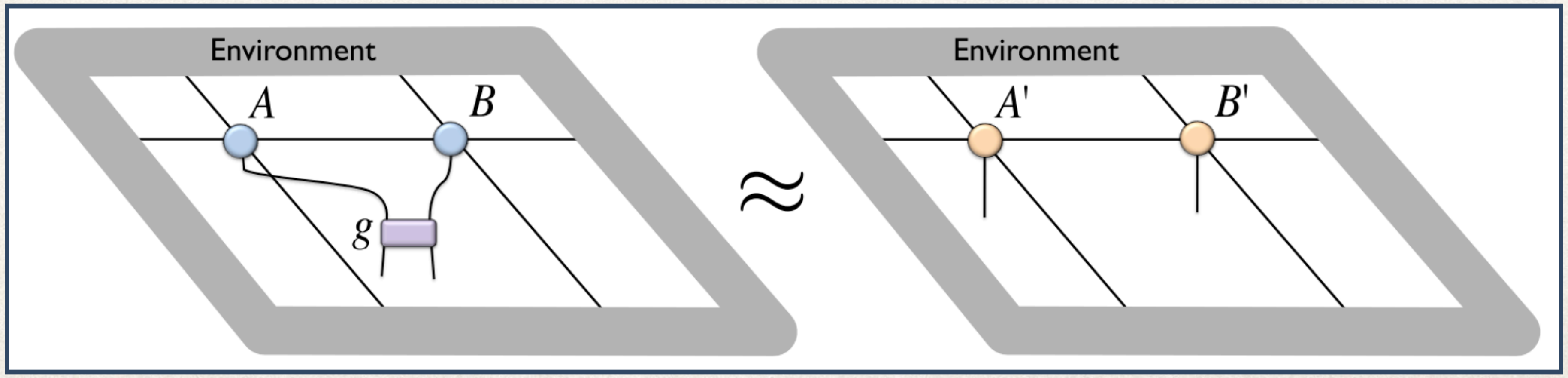
(a) Simple update - Easy and cheap $O(D^5)$



[Jiang at al. (2008)]

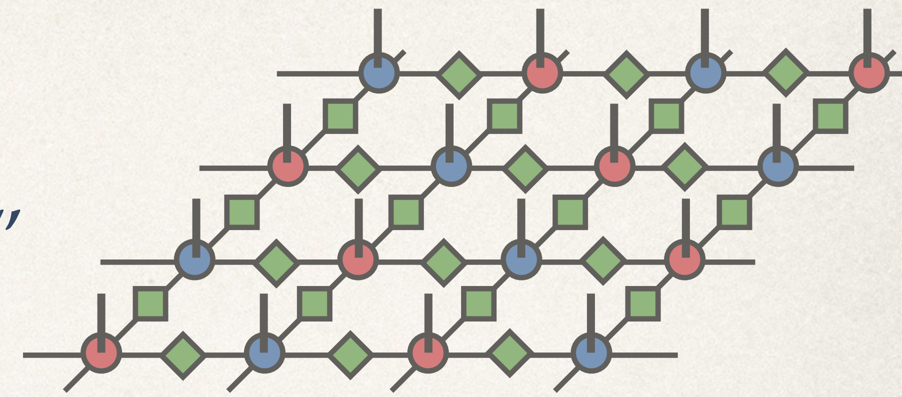
(b) Full update - Better accurate but heavy $O(D^{10})$

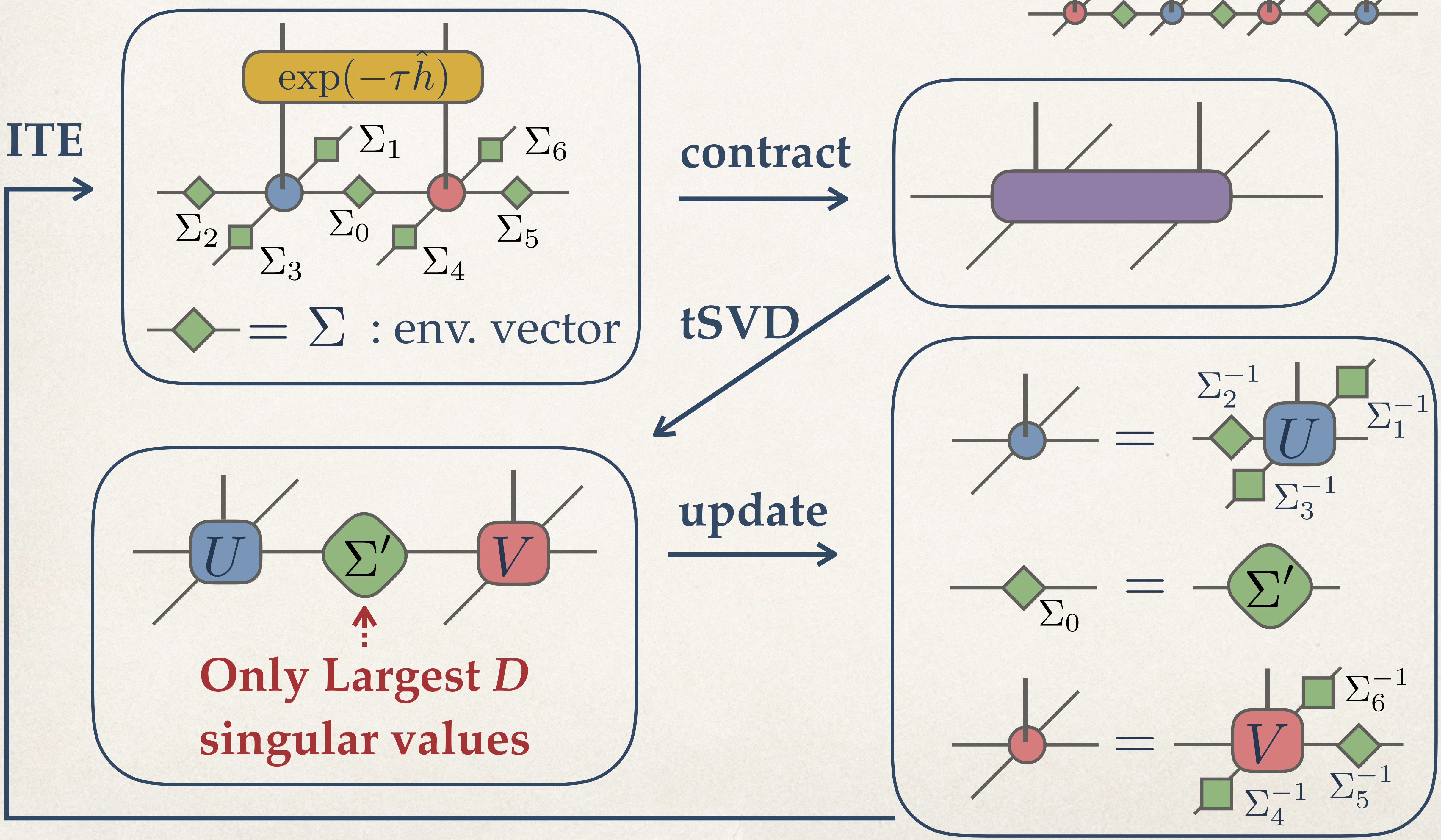
[P. Corboz (2016)]



Numerical Optimization

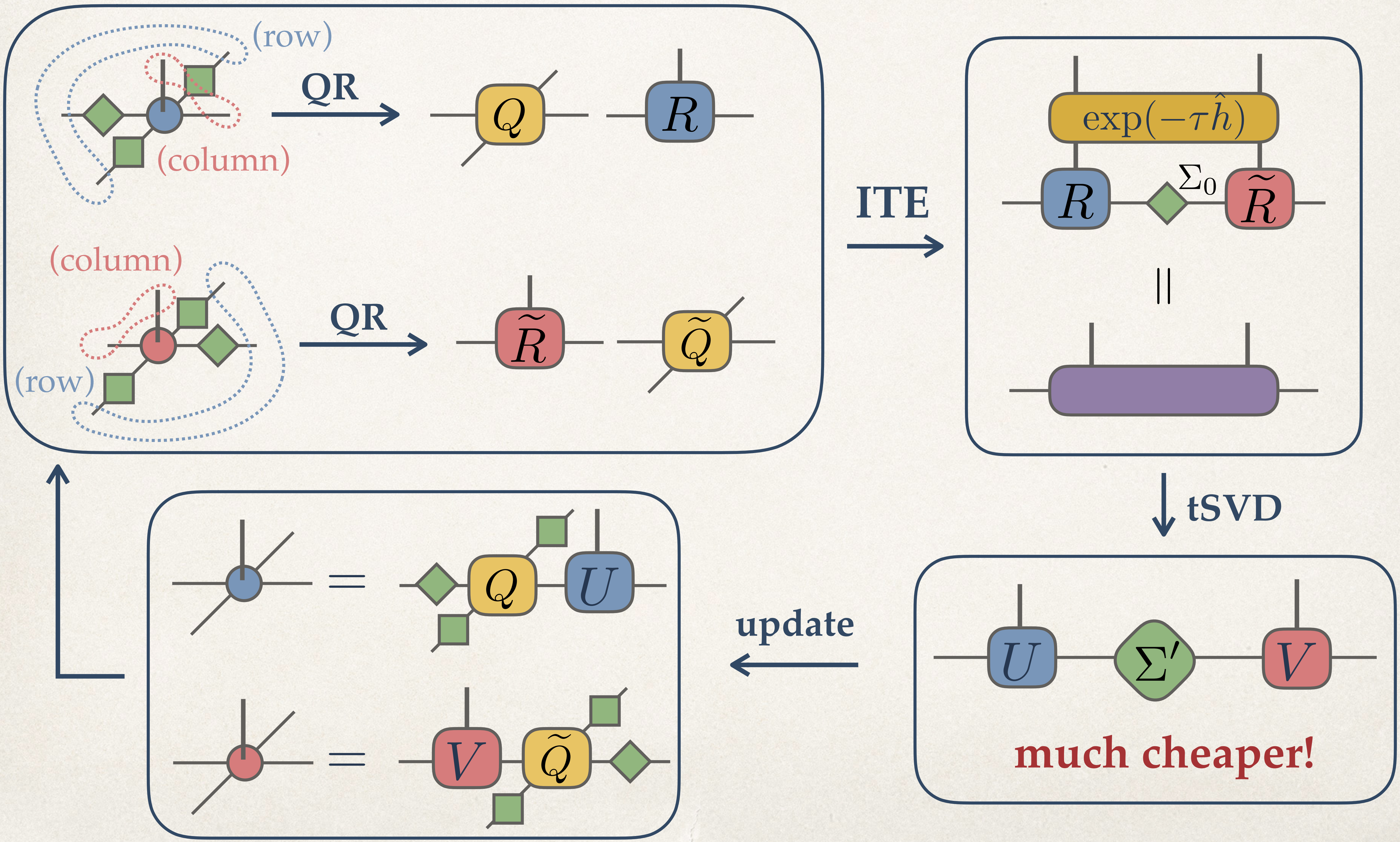
- Imaginary Time Evolution
  - **Simple update**: Averaged "Entanglement"





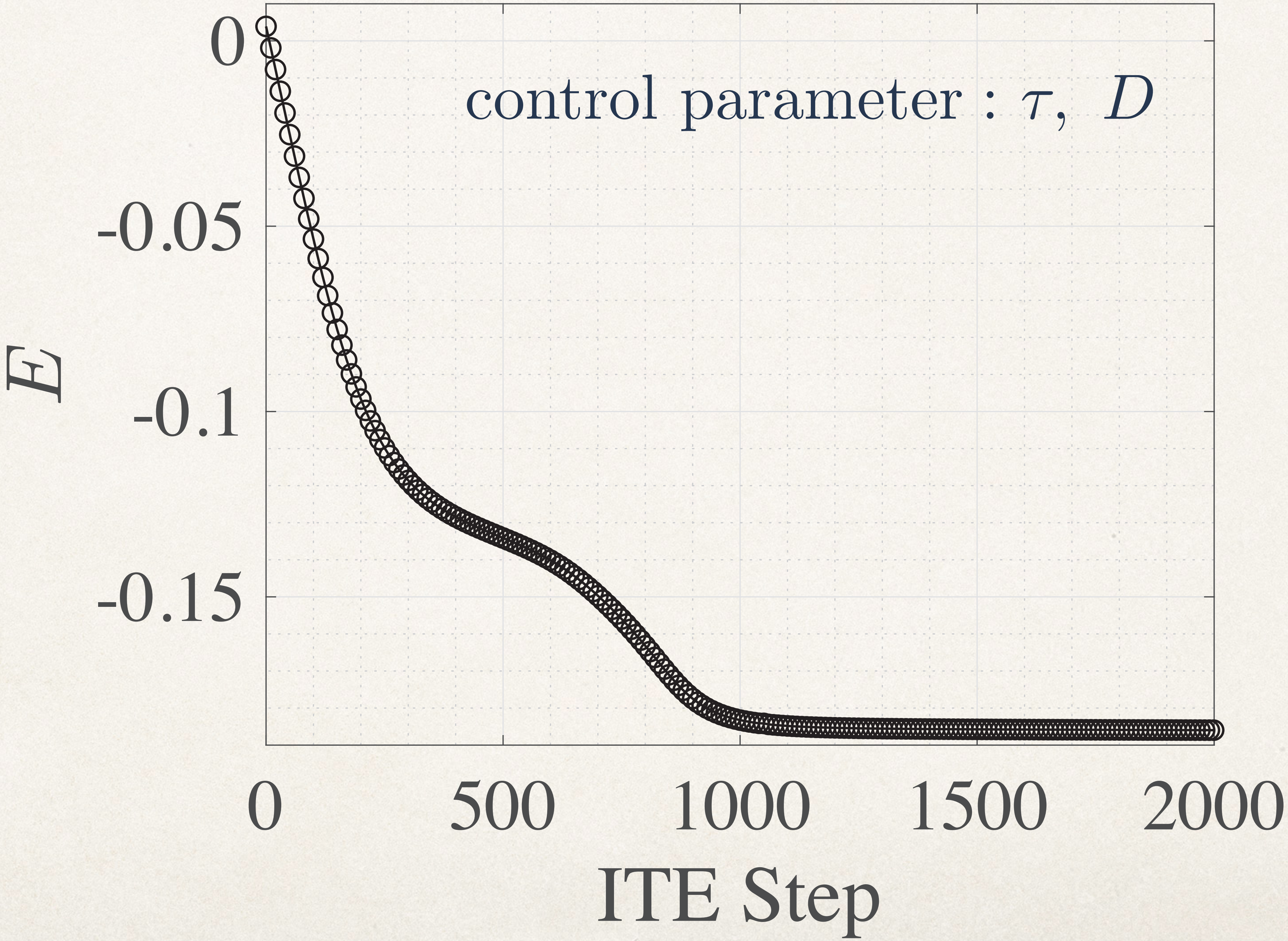
Numerical Optimization

- Imaginary Time Evolution
  - Tip to reduce complexity



Numerical Optimization

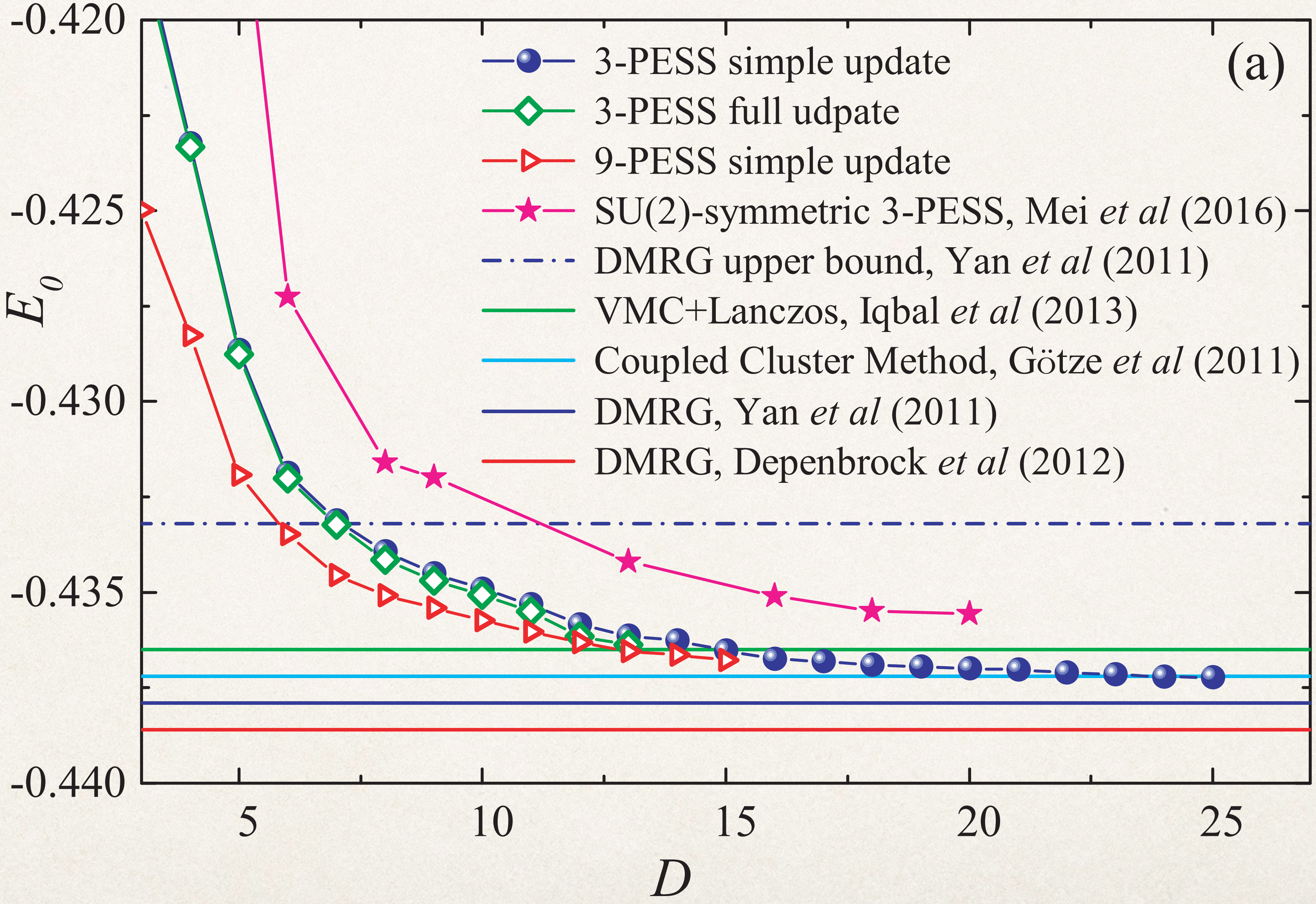
- Sample
  - ITE flow



Numerical Optimization

❖ Example1

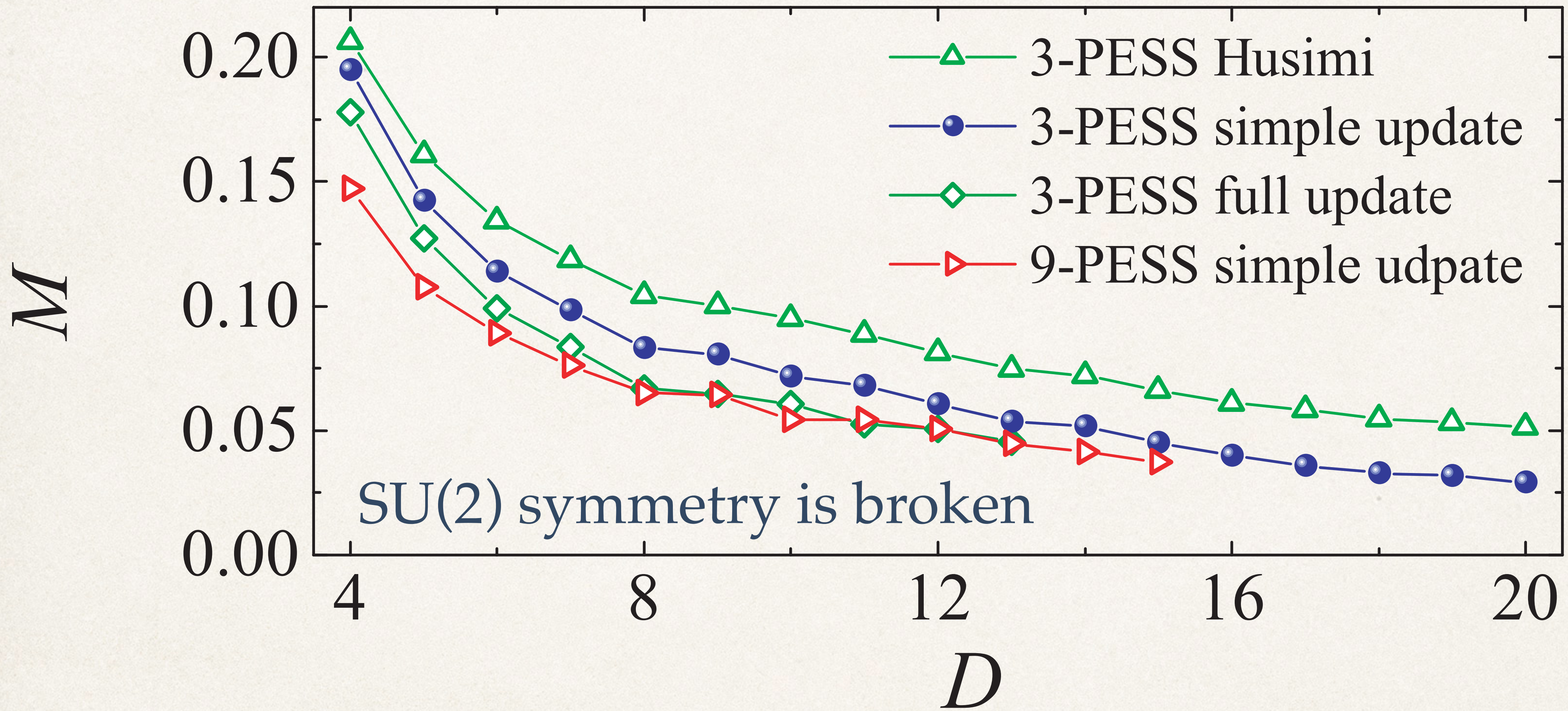
➪ Gapless Spin Liquid on Kagome **[Liao et al. (2018)]**



Numerical Optimization

❖ Example1

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

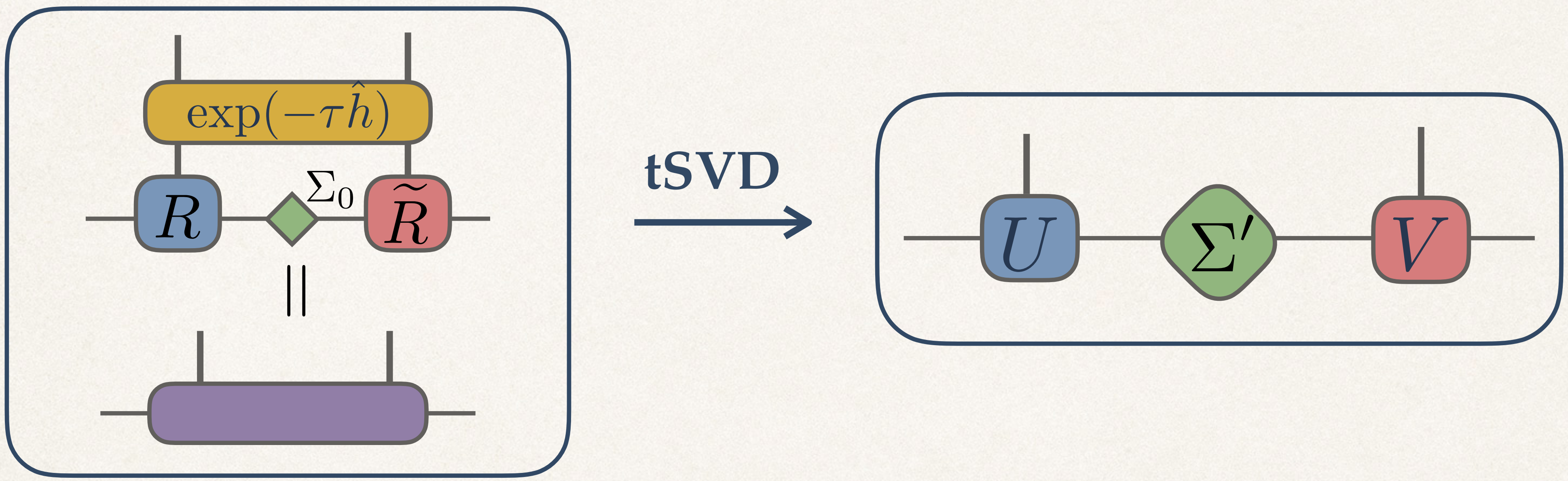


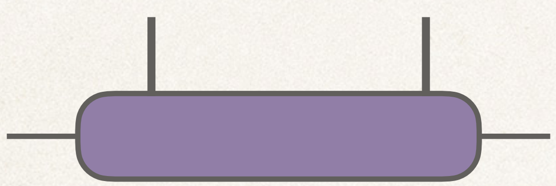
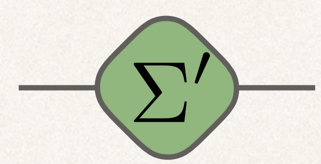
Magnetization is not exactly Zero, even though it is decreasing

Numerical Optimization

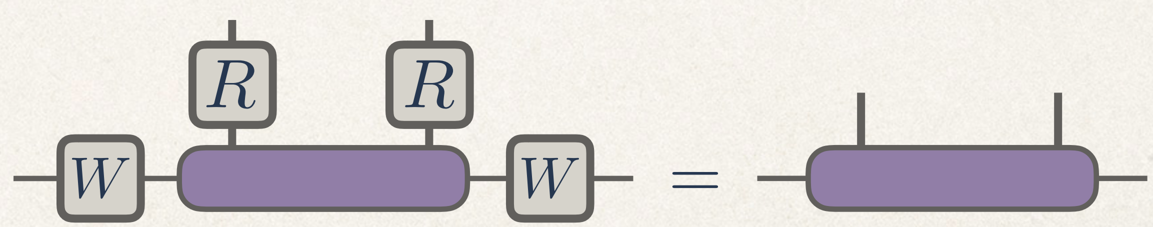
❖ **Symmetric** Simple Update

➯ Degeneracy in Singular Values

- 

- if  is symmetric, 

 are degenerates

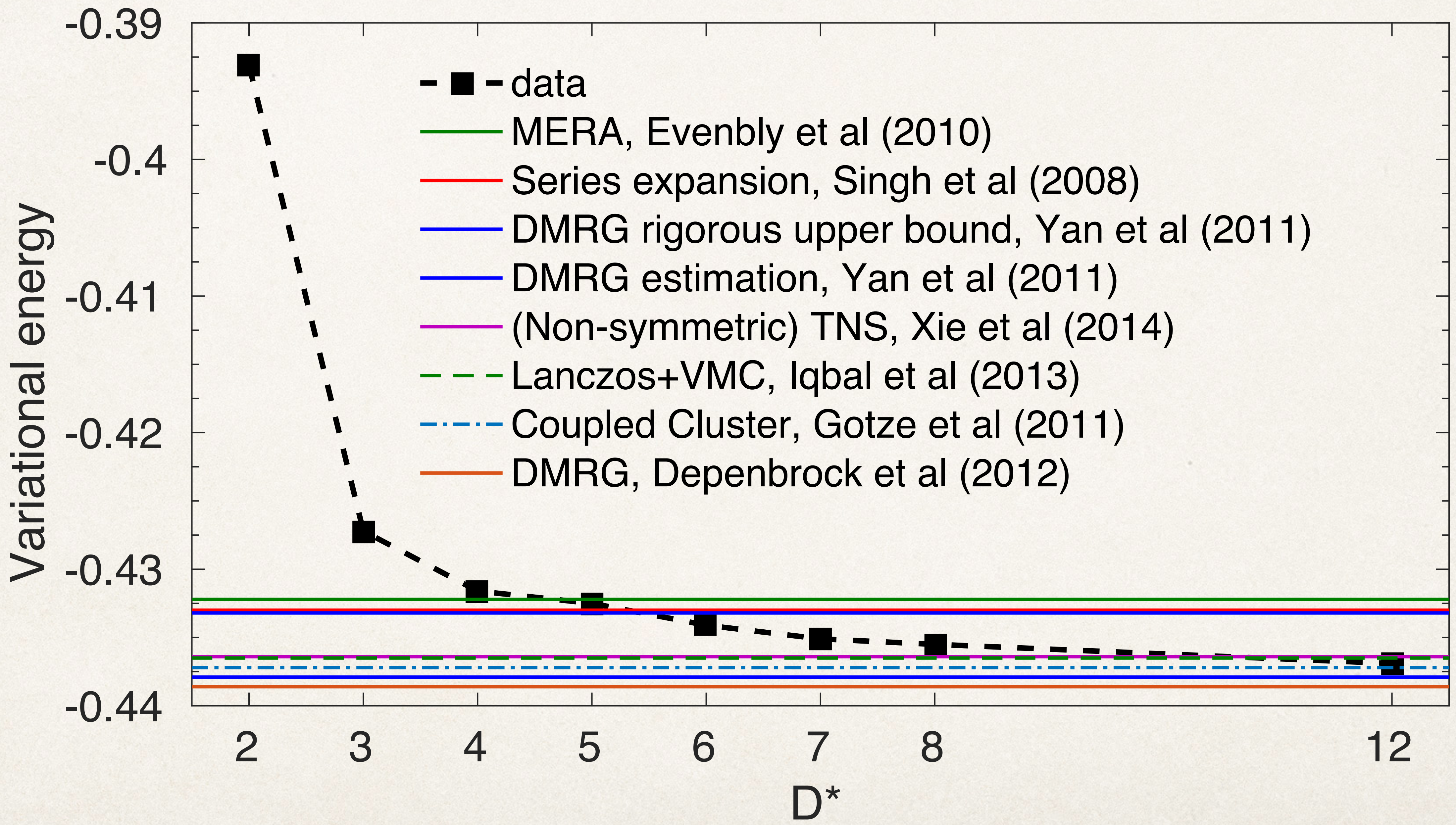
- Symmetric meaning 

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

Numerical Optimization

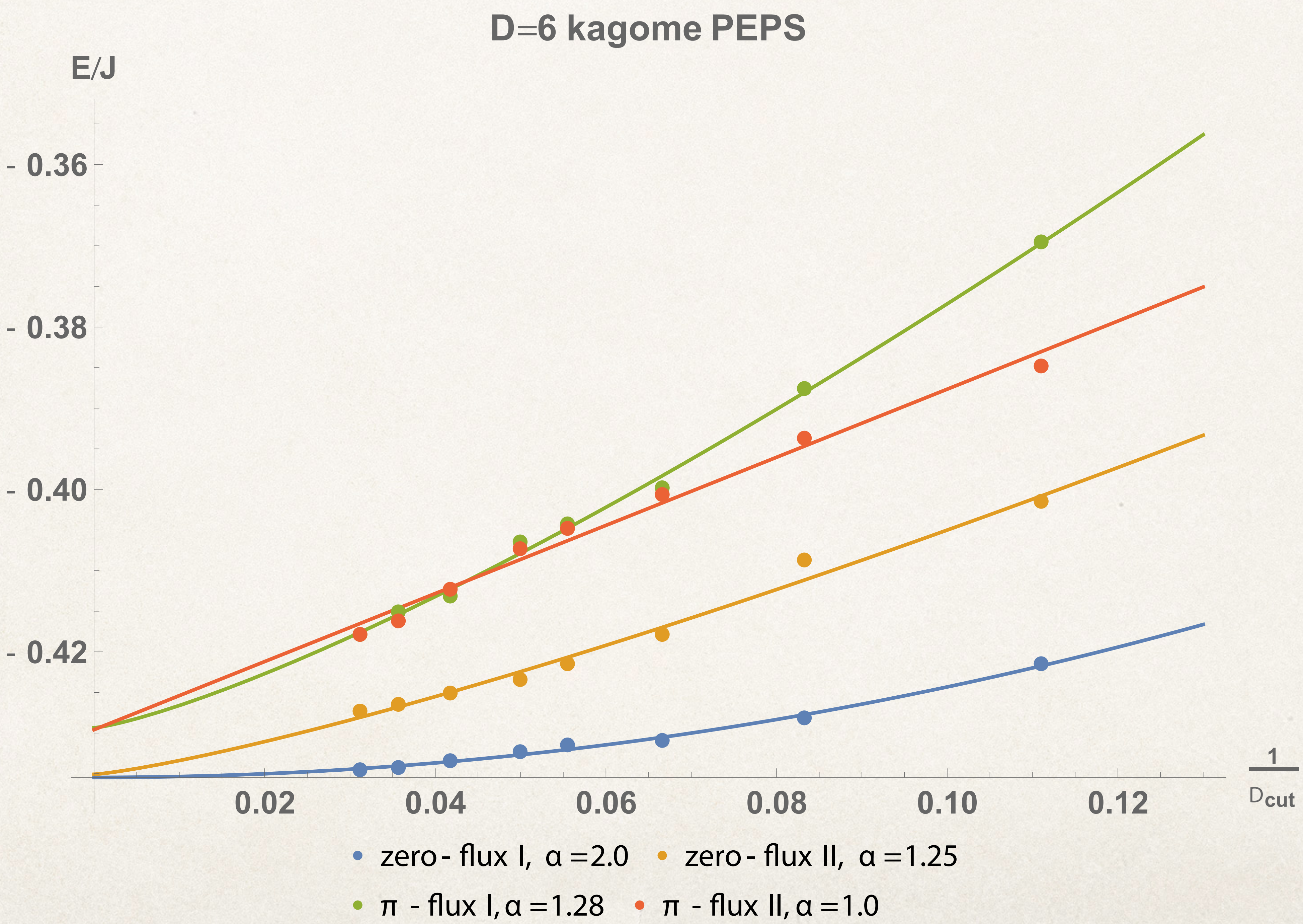
❖ Example2 - Symmetric SU

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



Numerical Optimization

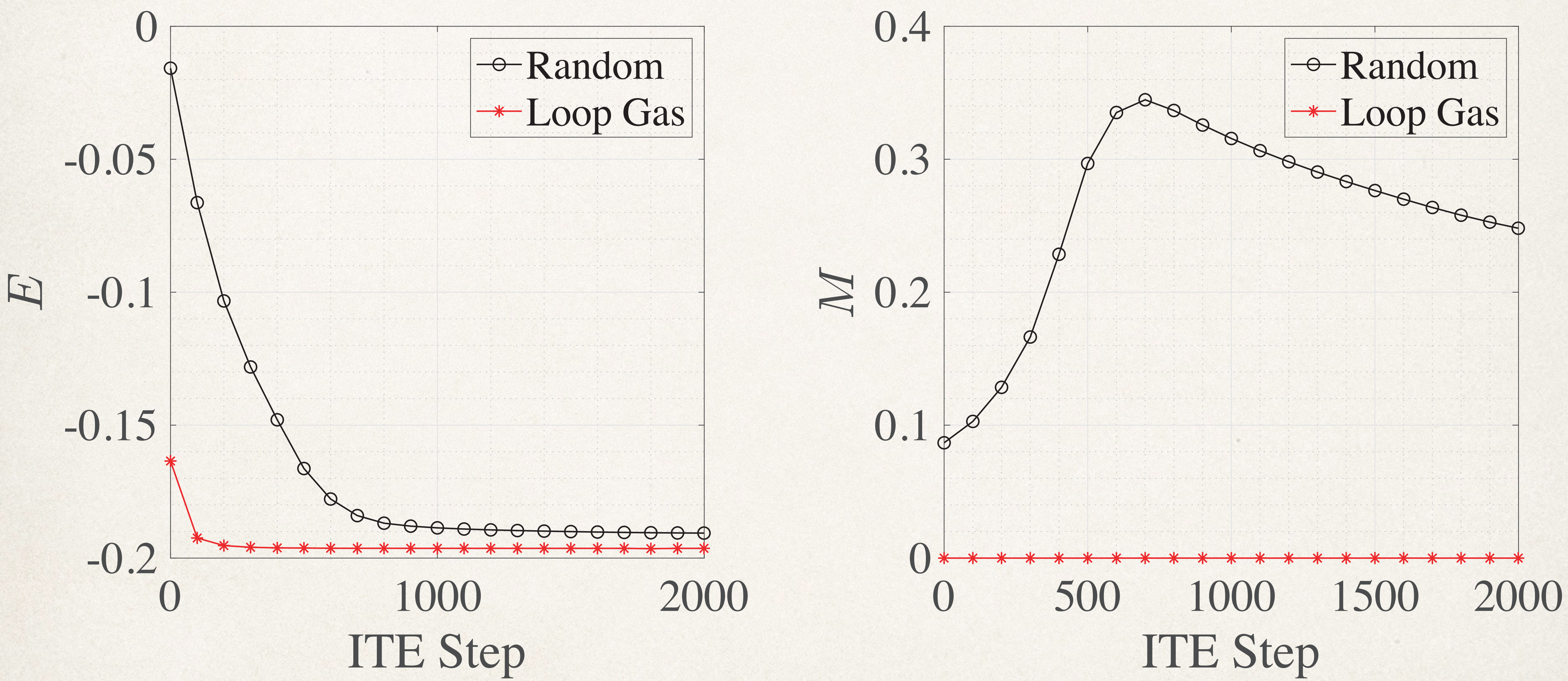
- Example3 - Symmetric SU
  - Gapless Spin Liquid on Kagome **[S. Jiang et al. (2019)]**



Numerical Optimization

❖ Example4 - Symmetric SU

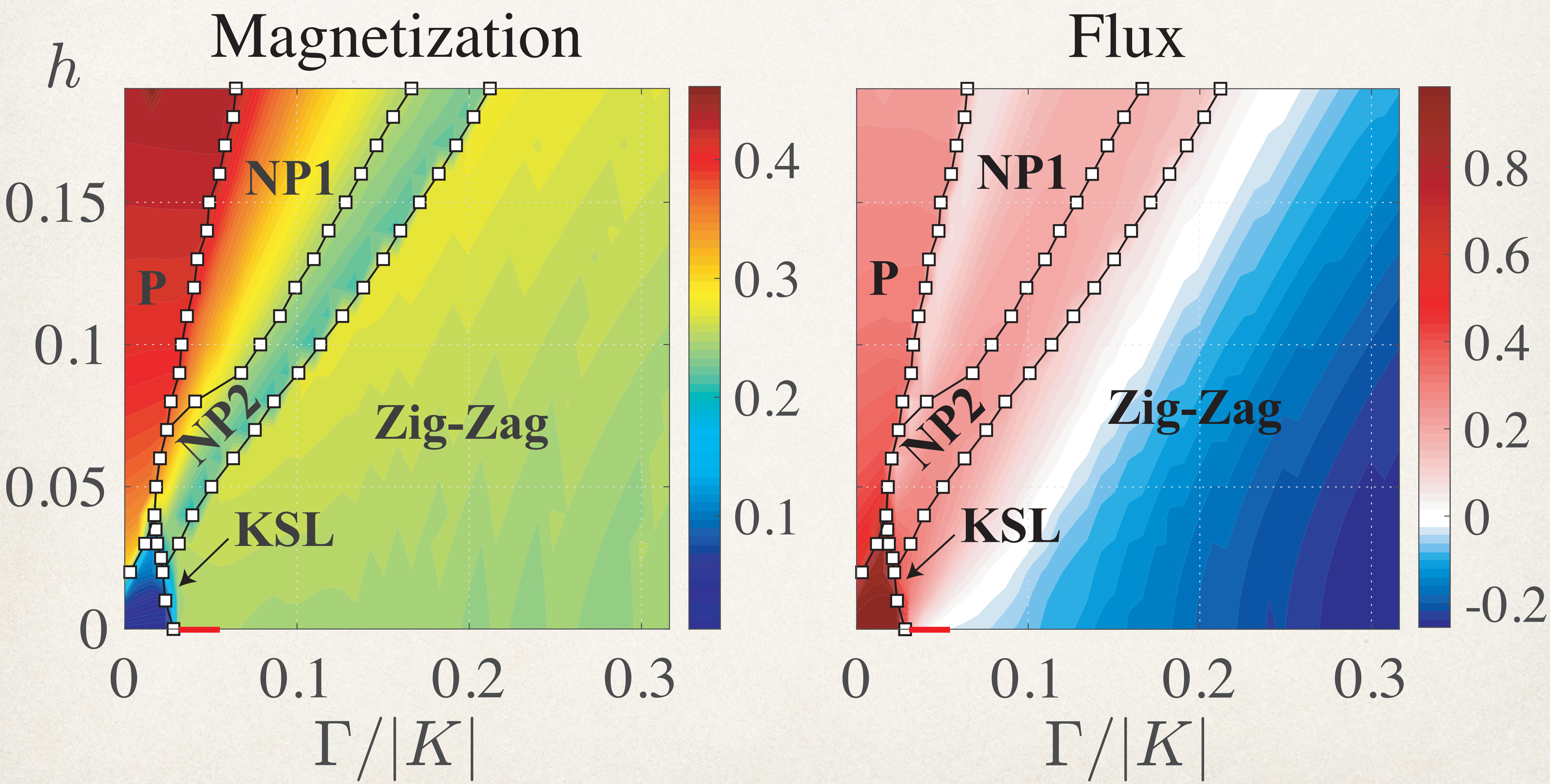
➪ Kitaev Spin Liquid **cannot** be obtained from Random state



Numerical Optimization

❖ Example5 - Symmetric SU

➯ Quantum Phase Diagram of K-G-G′ model **[H.-Y. Lee et al. (2019)]**

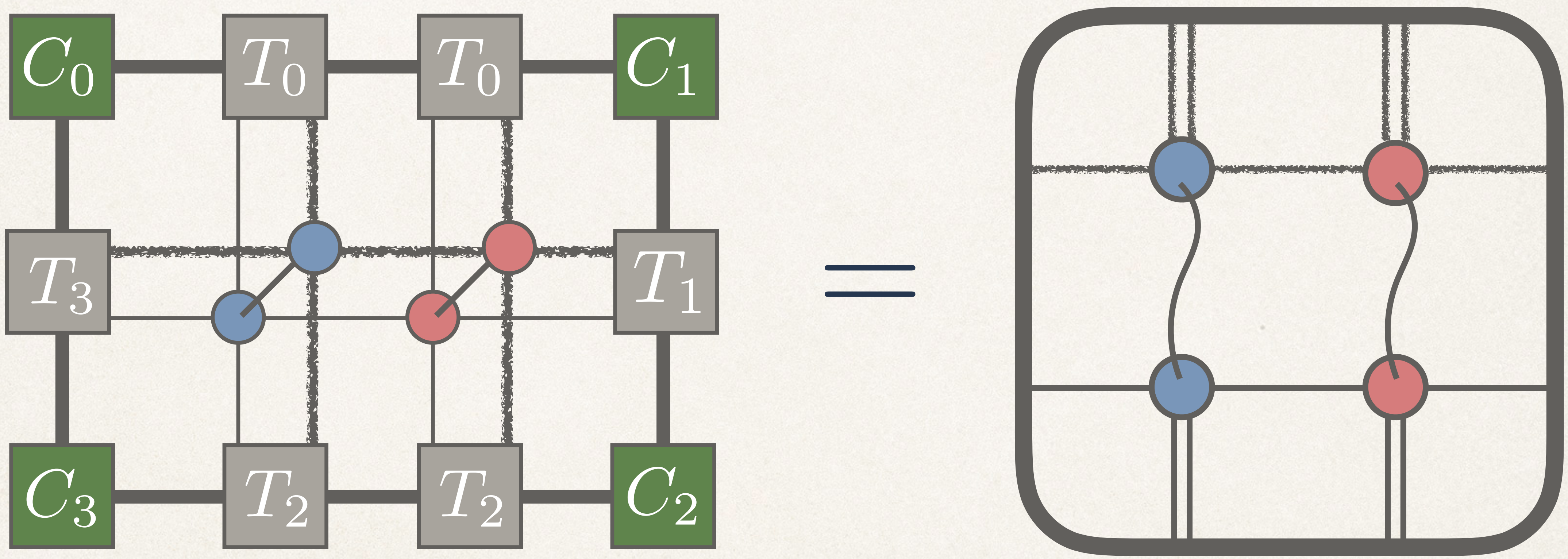


Numerical Optimization

❖ Imaginary Time Evolution

➪ **Full Update**

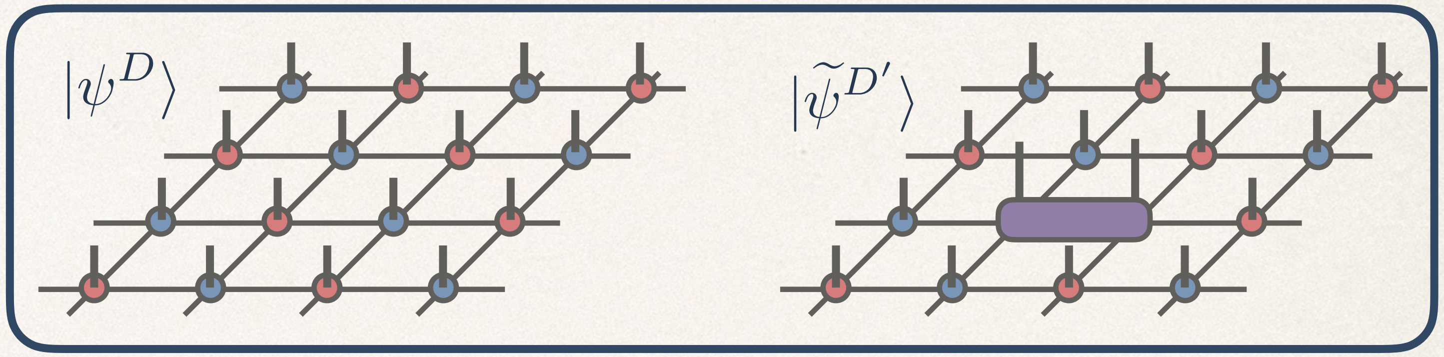
▸ before started



Numerical Optimization

❖ Imaginary Time Evolution

➯ **Full Update**



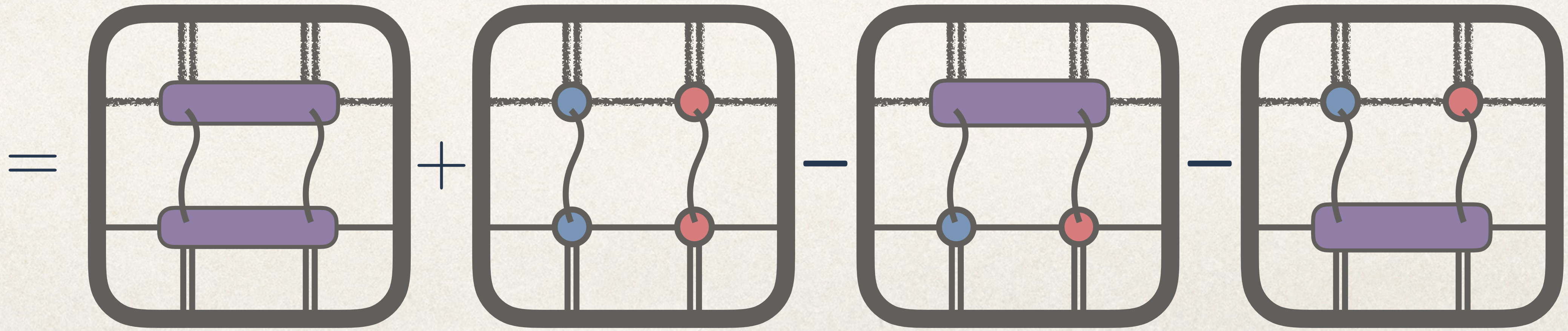
- $|\widetilde{\psi}^{D'}\rangle = e^{-\tau\hat{h}}|\psi^D\rangle$

- Goal: $|\widetilde{\psi}_n^{D'}\rangle \xrightarrow{\textbf{trunc}} |\psi_n^D\rangle$ **such that** $|\psi_n^D\rangle \simeq |\widetilde{\psi}_n^{D'}\rangle$

- Cost function:

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

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



Numerical Optimization

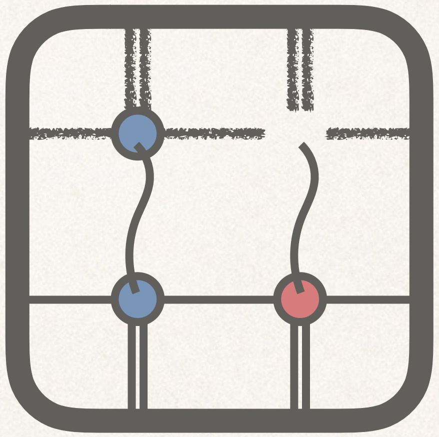
❖ Imaginary Time Evolution

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

➪ **Full Update**

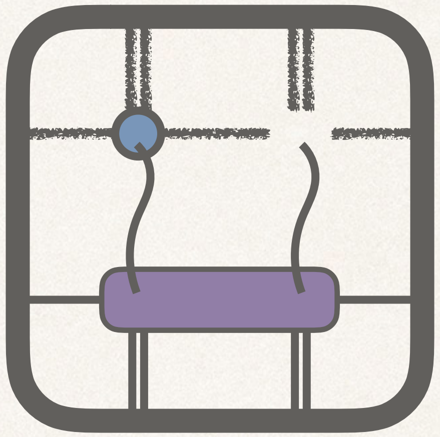
- Minimizing Cost function:

$$\frac{\partial \epsilon}{\partial T^*_{\text{red}}} =$$



$A\,T_{\text{red}}$

–



B

$\xrightarrow{\textbf{want}} 0$

$$= \underbrace{A_{(l_0,u_0,r_0,d_0,p_0)|(l_1,u_1,r_1,d_1,p_1)}}_{\textbf{Matrix}} \underbrace{[T_{\text{red}}]_{(l_1,u_1,r_1,d_1,p_1)}}_{\textbf{Vector}} - \underbrace{B_{(l_0,u_0,r_0,d_0,p_0)}}_{\textbf{Vector}}$$

- Solve linear equation $A\,v = B$

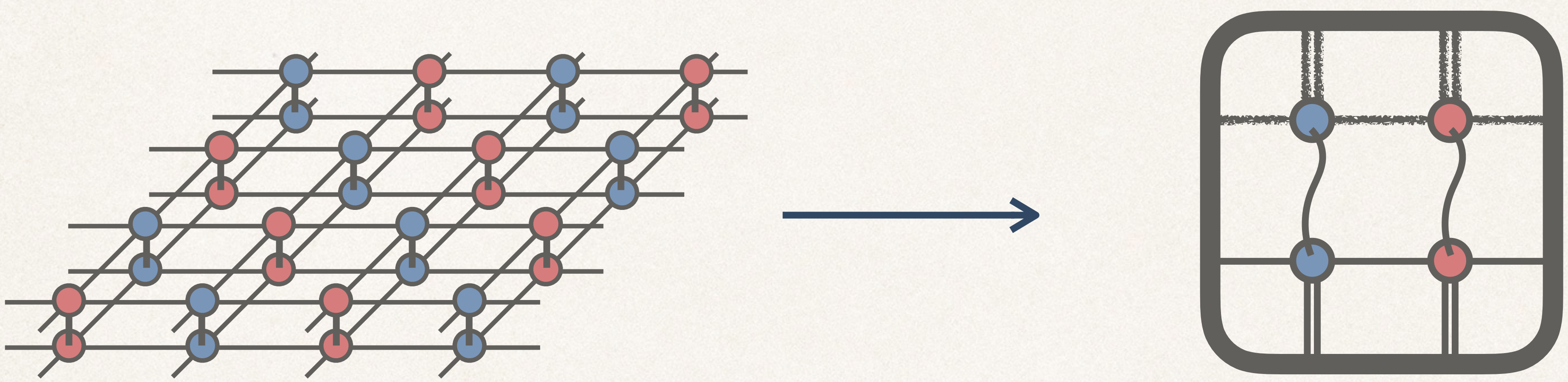
$\longrightarrow$ $[T^{\text{new}}_{\text{red}}]^p_{lrud} = v_{(l,r,u,d,p)}$ $\longrightarrow$ Replace T_{red} by $T^{\text{new}}_{\text{red}}$

Numerical Optimization

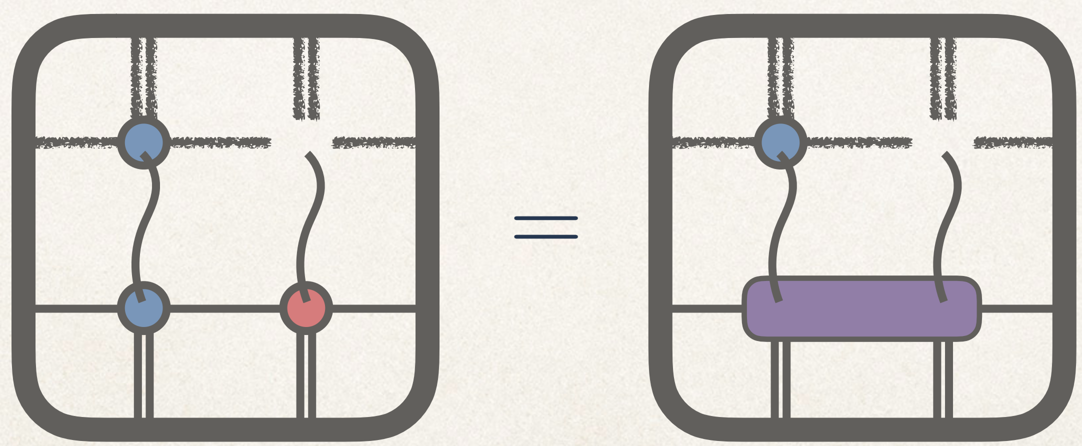
❖ Imaginary Time Evolution

➩ **Full Update**

- Step1: Calculating environment tensors



- Step2: Solving linear equation



- Step3: Replace all tensors by solution tensor

Thank you very much!