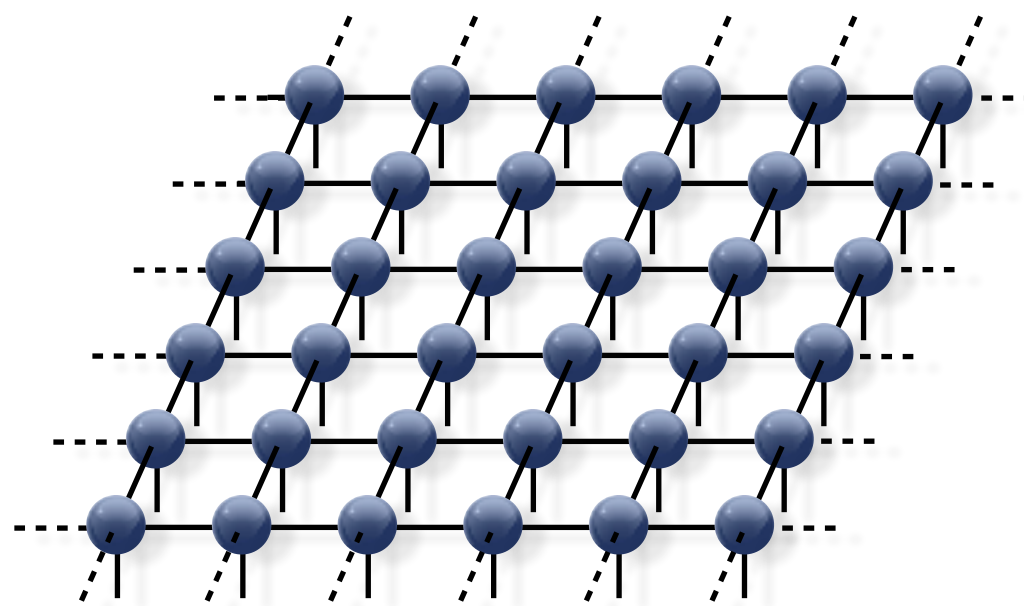
Lectures on

1: Introduction to iPEPS

2: Advanced iPEPS techniques

Philippe Corboz, Institute for Theoretical Physics, University of Amsterdam

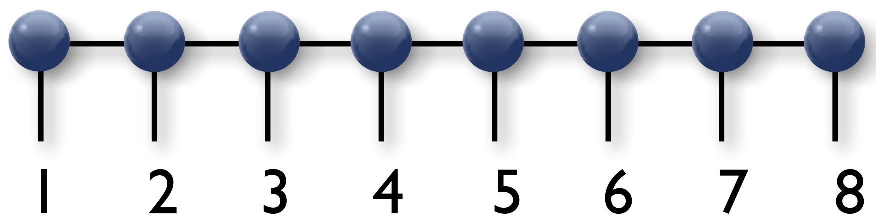


Overview: tensor networks in 1D and 2D

1D

MPS

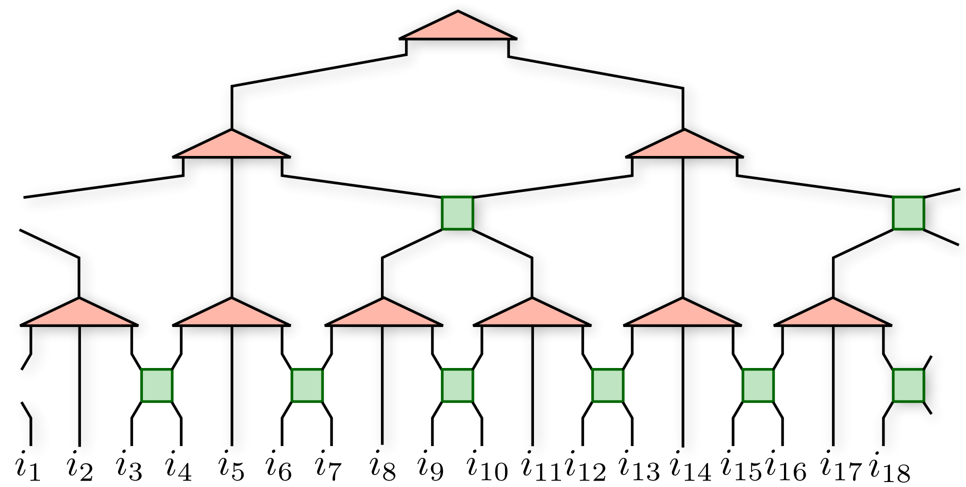
Matrix-product state



Underlying ansatz of the density-matrix renormalization group (**DMRG**) method

1D MERA

Multi-scale entanglement renormalization ansatz



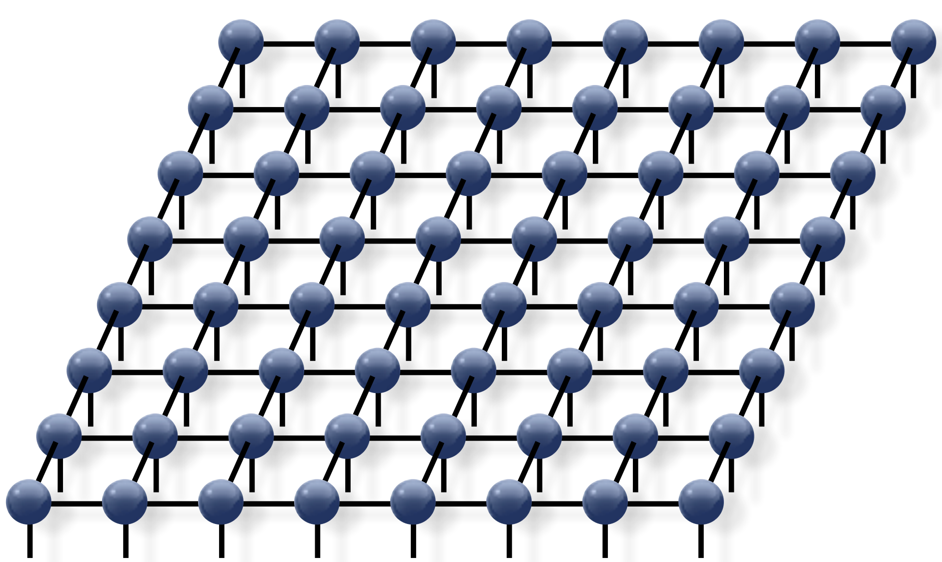
and more

- 1D tree tensor network
- ...

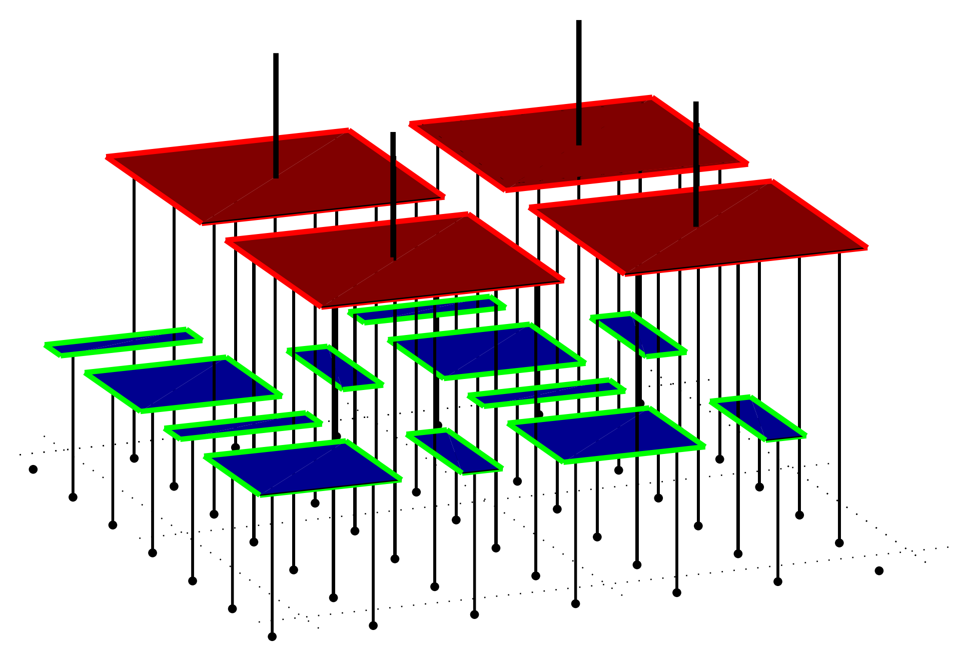
2D

PEPS

projected entangled-pair state



2D MERA



and more

- Entangled-plaquette states
- 2D tree tensor network
- String-bond states
- ...


finite T
layered systems
dynamics
open
systems
ground
states
excitations
spectral fcts
3D
Fermions
Tensor networks
... and more ...
spiral order
Top. order
long-range H
critical systems

Outline of today's lectures

- Tensor network ansatz
  - *Main idea of a tensor network ansatz & area law of the entanglement entropy*
  - *MPS, PEPS & iPEPS, (MERA)*
- Contraction of a 2D tensor network
  - *MPS-MPO approach, corner transfer matrix (CTM) method, Tensor Renormalization Group (TRG), Tensor network renormalization (TNR)*
- Optimization of iPEPS
  - *Imaginary time evolution: simple vs full optimization*
  - *Energy minimization: automatic differentiation*

Introduction to tensor networks

➡ **Aim**: Efficient representation of quantum many-body states

Lattice with N sites

1 2 3 4 5 .... N-1 N

$\mathbb{V}$: local Hilbert space e.g. $\{|\uparrow\rangle, |\downarrow\rangle\}$

Full Hilbert space $\mathbb{V} \otimes \mathbb{V} \otimes \mathbb{V} \otimes \mathbb{V} \otimes \mathbb{V} \otimes \; .... \; \otimes \mathbb{V} \otimes \mathbb{V}$

dimension 2^N grows exponentially with N

Hamiltonian $\hat{H} = \sum_{\langle ij \rangle} \hat{h}_{ij}$ sum of local terms

Represent the ground state

$$|\Psi\rangle = \sum_{\substack{i_1 i_2 \ldots i_N \\ i_k \in \{\uparrow, \downarrow\}}} \Psi_{i_1 i_2 \ldots i_N} |i_1 \otimes i_2 \otimes \cdots \otimes i_N\rangle$$

2^N coefficients

Complexity **~exp(N) many numbers ⟶ inefficient!**

Tensor network ansatz for a wave function

Lattice: ● 1 ● 2 ● 3 ● 4 ● 5 ● 6

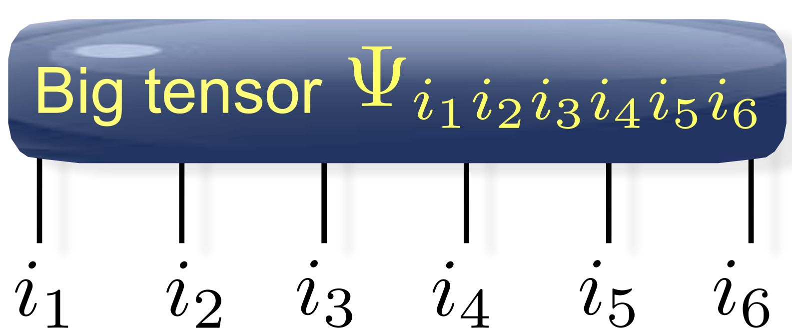
2 basis states per site: $\{|\uparrow\rangle, |\downarrow\rangle\}$

2^6 basis states

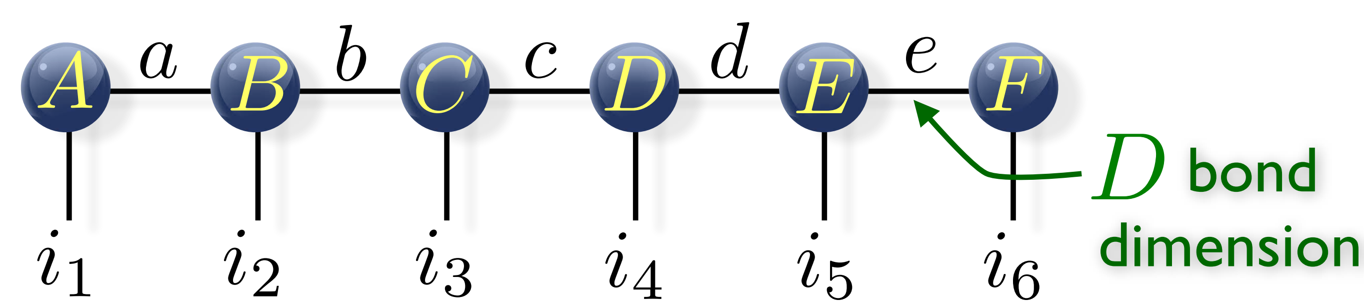
2^6 coefficients

State: $|\Psi\rangle = \sum_{i_1 i_2 i_3 i_4 i_5 i_6} \Psi_{i_1 i_2 i_3 i_4 i_5 i_6} |i_1 \otimes i_2 \otimes i_3 \otimes i_4 \otimes i_5 \otimes i_6\rangle$

Tensor/multidimensional array



Tensor network: matrix product state (**MPS**)

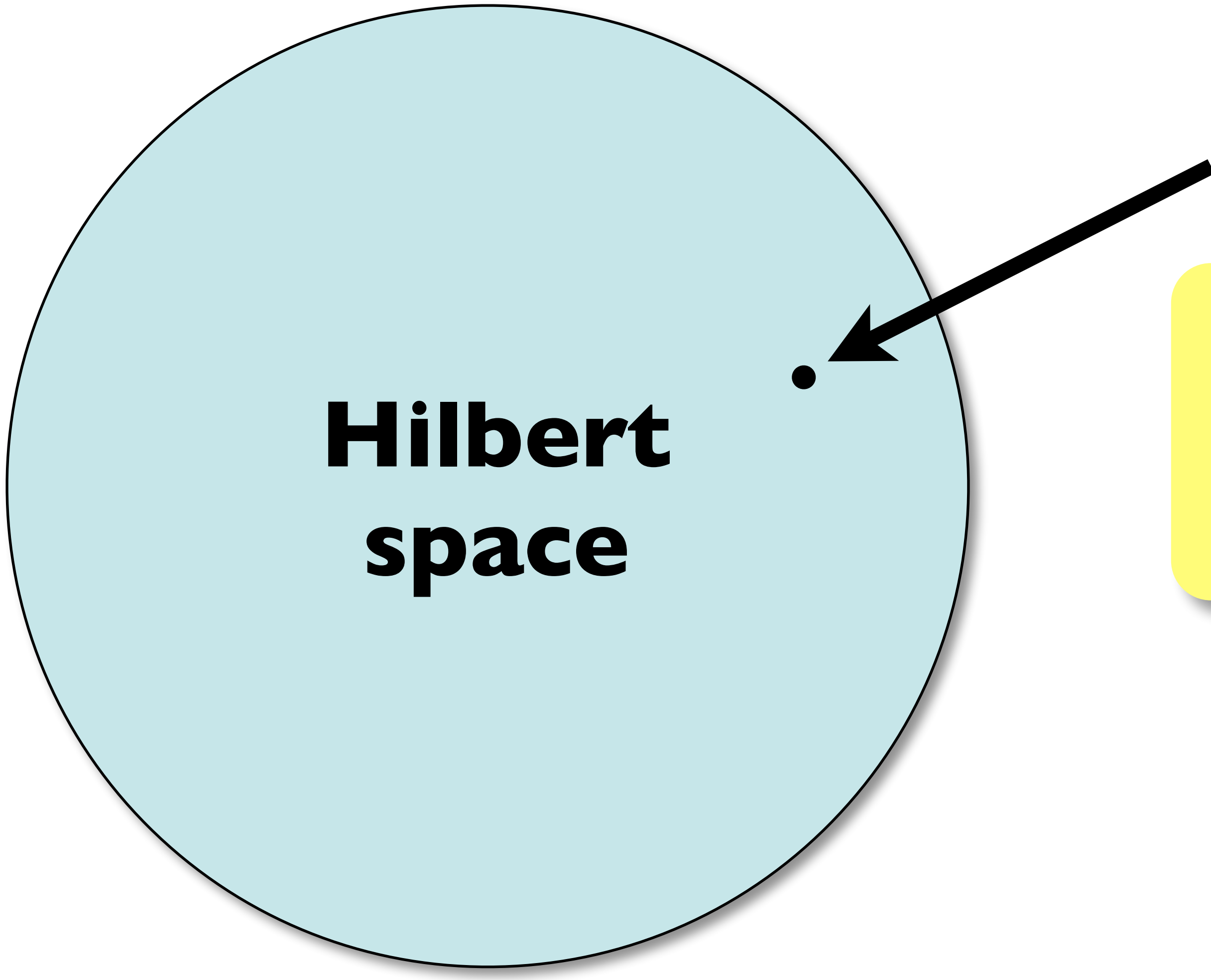


$$\Psi_{i_1 i_2 i_3 i_4 i_5 i_6} \approx \sum_{abcde} A^{a}_{i_1} B^{ab}_{i_2} C^{bc}_{i_3} D^{cd}_{i_4} E^{de}_{i_5} F^{e}_{i_6} = \tilde{\Psi}_{i_1 i_2 i_3 i_4 i_5 i_6}$$

exp(N) many numbers **vs** poly(D,N) numbers

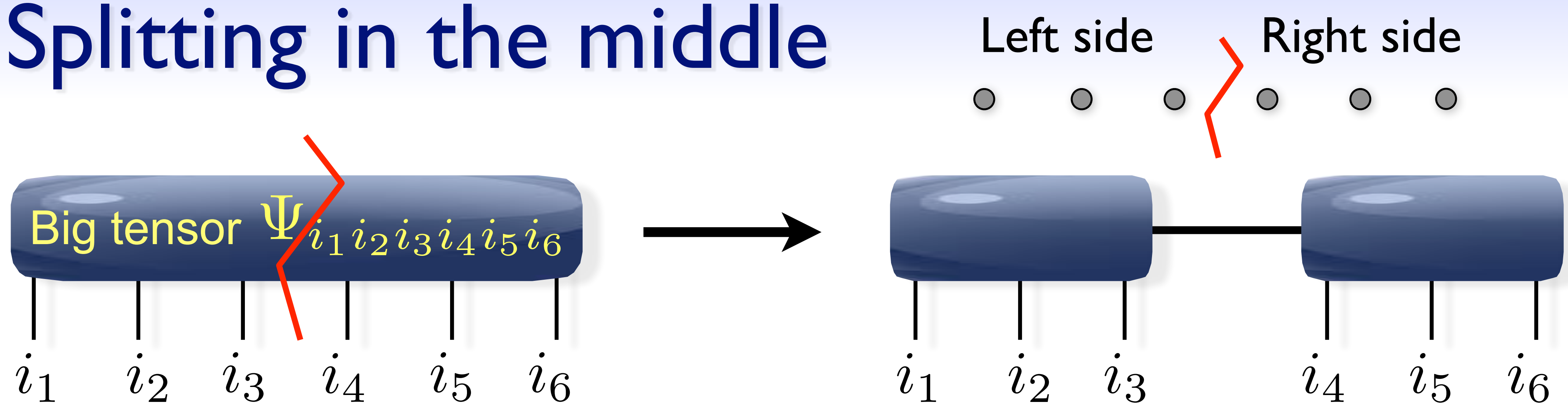
Efficient representation!

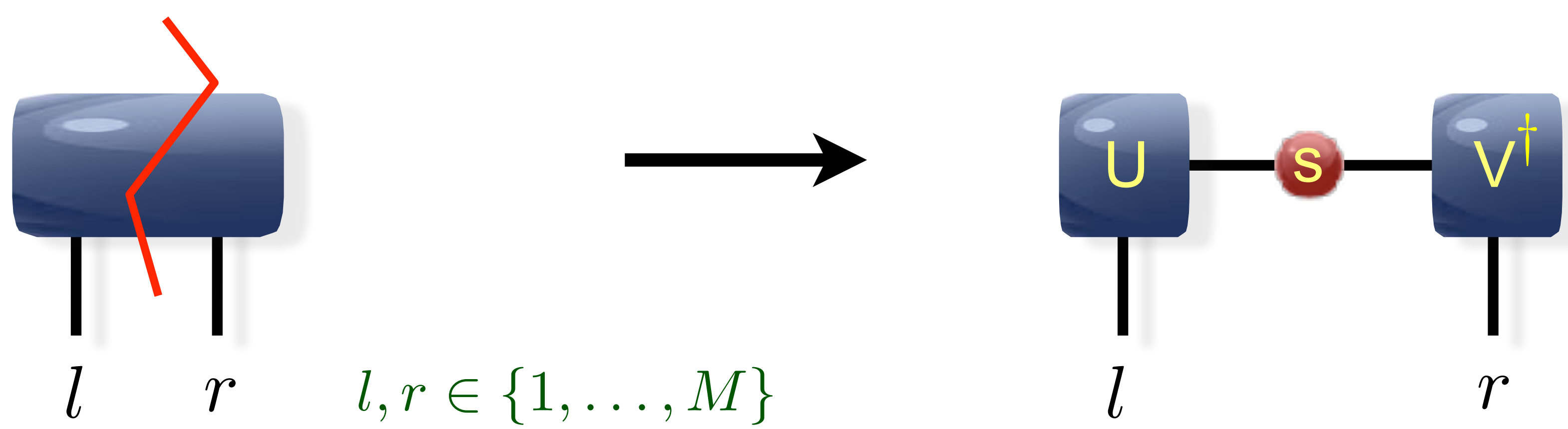
“Corner” of the Hilbert space



- ★ GS of local H’s are less entangled than a random state in the Hilbert space
- ★ *Area law of the entanglement entropy*

Splitting in the middle





$l, r \in \{1, \ldots, M\}$

$M = 2^{N/2}$

Singular value decomposition

$$\Psi = U s V^\dagger$$

$$s_{kk} \geq 0$$

diagonal matrix!

$$\Psi_{lr} = \sum_k U_{lk} s_{kk} V^*_{rk}$$

$$|\Psi\rangle = \sum_{lr} \Psi_{lr} |l\rangle |r\rangle = \sum_{lr} \sum_k U_{lk} s_{kk} V^*_{rk} |l\rangle |r\rangle$$

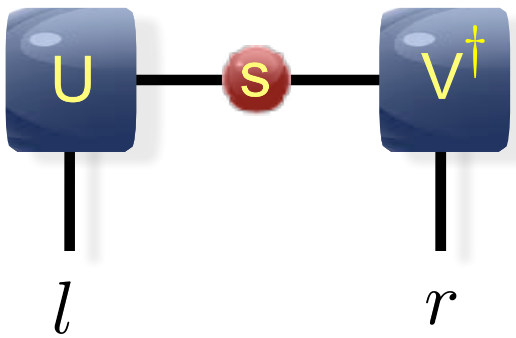
$$= \sum_k s_{kk} |u_k\rangle |v_k\rangle$$

Schmidt decomposition

How many relevant singular values?

$$|\Psi\rangle = \sum_{k}^{M} s_{kk} |u_k\rangle |v_k\rangle$$

how many non-zero singular values?



★ Special cases:

$$s_{11} = 1, \quad s_{kk} = 0 \quad \text{for} \quad k > 1$$

$$|\Psi\rangle = 1|u_1\rangle|v_1\rangle$$

Product state

$$s_{11} = \frac{1}{\sqrt{2}}, \quad s_{22} = \frac{1}{\sqrt{2}}, \quad s_{kk} = 0 \quad \text{for} \quad k > 2$$

$$|\Psi\rangle = \frac{1}{\sqrt{2}}|u_1\rangle|v_1\rangle + \frac{1}{\sqrt{2}}|u_2\rangle|v_2\rangle$$

Entangled state

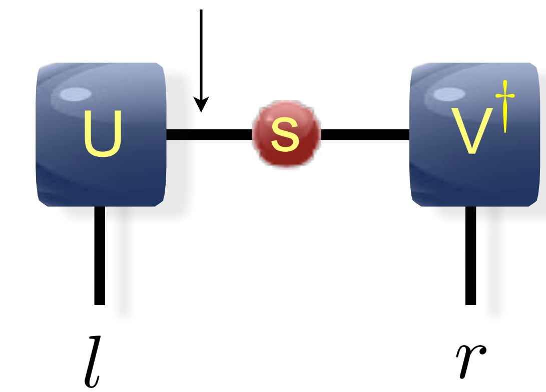
$$s_{kk} = \frac{1}{\sqrt{M}}, \quad \text{for all } k$$

Maximally entangled state

How many relevant singular values?

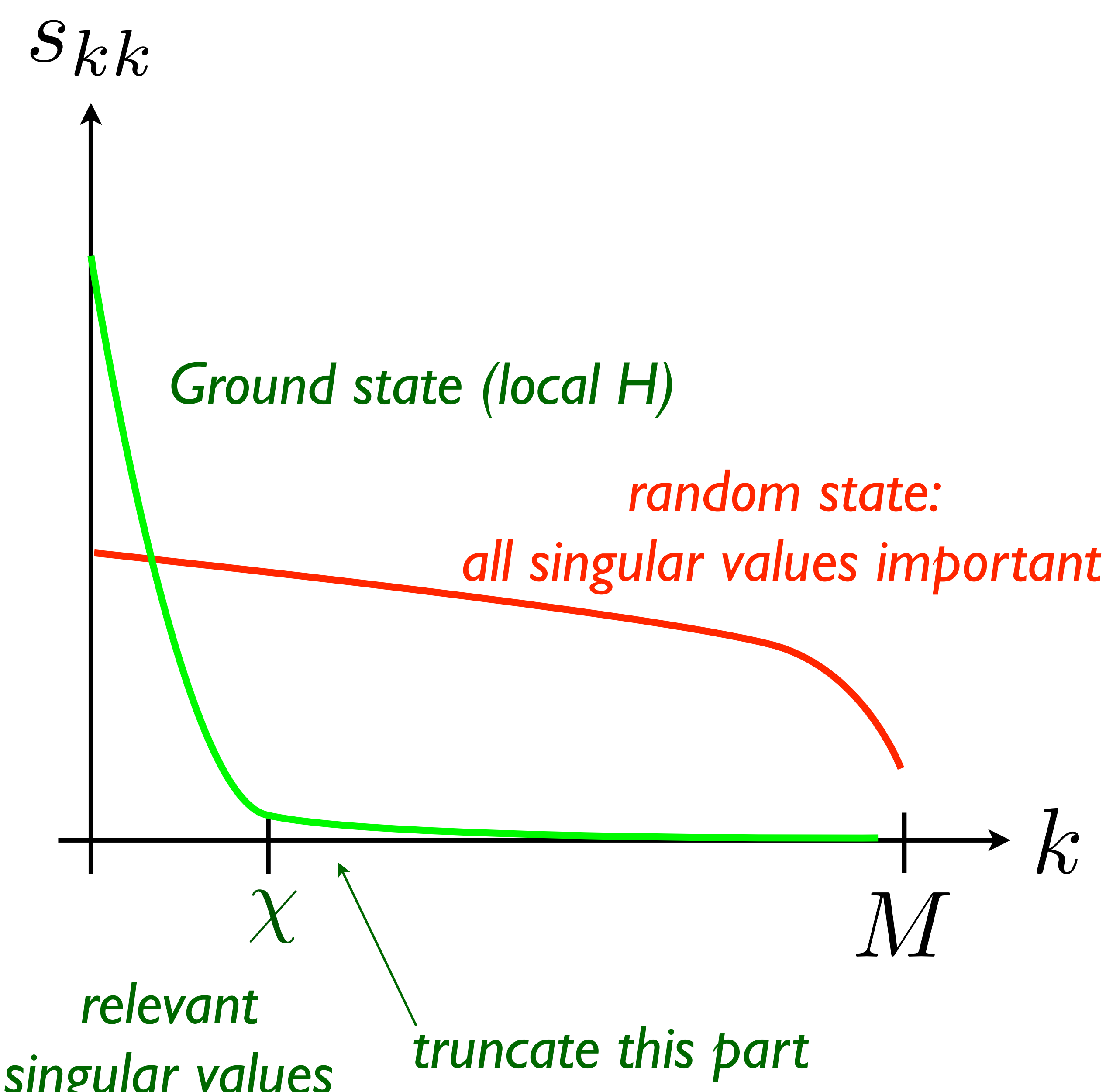
bond dimension

$D = \chi$



$$|\Psi\rangle = \sum_{k}^{M} s_{kk} |u_k\rangle |v_k\rangle$$

*how many **relevant** singular values?*



$$|\Psi\rangle \approx |\tilde{\Psi}\rangle = \sum_{k}^{\chi} s_{kk} |u_k\rangle |v_k\rangle$$

keeping the χ largest singular values minimizes the error

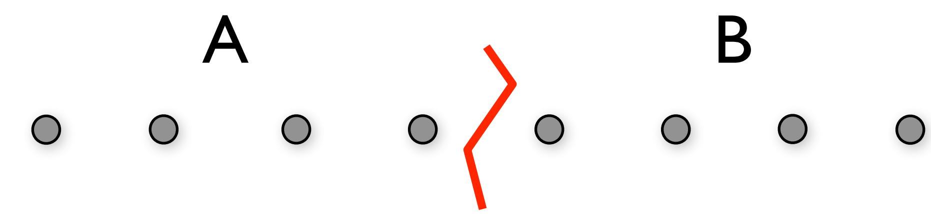
$$|||\Psi\rangle - |\tilde{\Psi}\rangle||$$

KEY IDEA OF DMRG!

Steven R. White

Reduced density matrix

$$|\Psi\rangle = \sum_k^M s_{kk} |u_k\rangle |v_k\rangle$$



★ Reduced density matrix of left side: *describes system on the left side*

$$\rho_A = \mathrm{tr}_B[\rho] = \mathrm{tr}_B[|\Psi\rangle\langle\Psi|] = \sum_k \lambda_k |u_k\rangle\langle u_k| \qquad \lambda_k = s_{kk}^2 \quad \textit{probability}$$

★ **Entanglement entropy:** $S(A) = -\mathrm{tr}[\rho_A \log \rho_A] = -\sum_k \lambda_k \log \lambda_k$

- Product state: $S(A) = -1 \log 1 = 0$

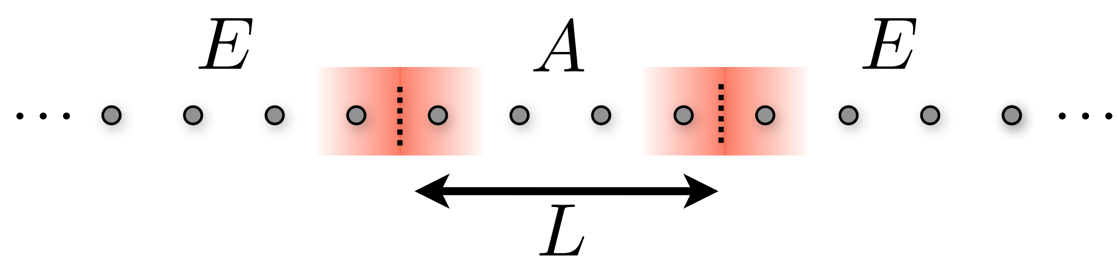
⋮

- Maximally entangled state: $S(A) = -\sum_k \frac{1}{M} \log \frac{1}{M} = logM$

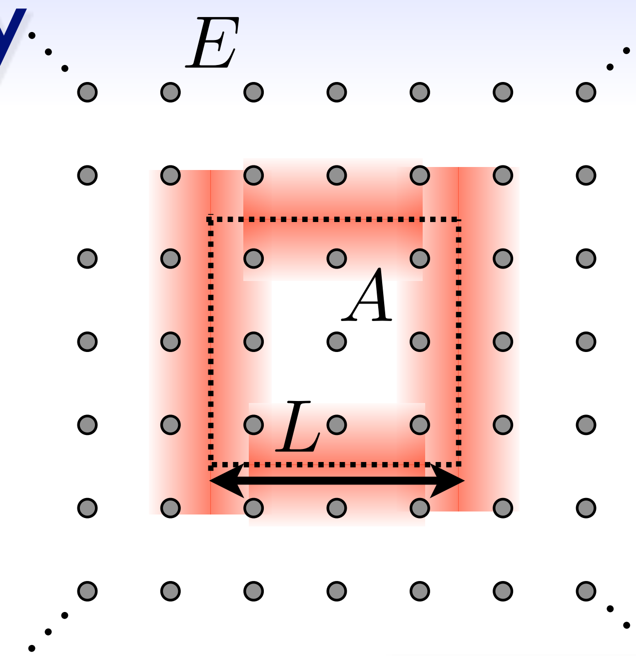
How large is S in a ground state? How does it ***scale*** *with system size?*

Area law of the entanglement entropy

1D



2D



Entanglement entropy $S(A) = -\mathrm{tr}[\rho_A \log \rho_A] = -\sum_i \lambda_i \log \lambda_i$

relevant states
$\chi \sim \exp(S)$

General (random) state

$S(L) \sim L^d$ **(volume)**

Ground state (local, gapped Hamiltonian)

$S(L) \sim L^{d-1}$ **(area law)**

Critical ground states:
(all in 1D but not all in 2D)

1D $S(L) \sim \log(L)$

2D $S(L) \sim L \log(L)$

1D $S(L) = const$ $\chi = const$

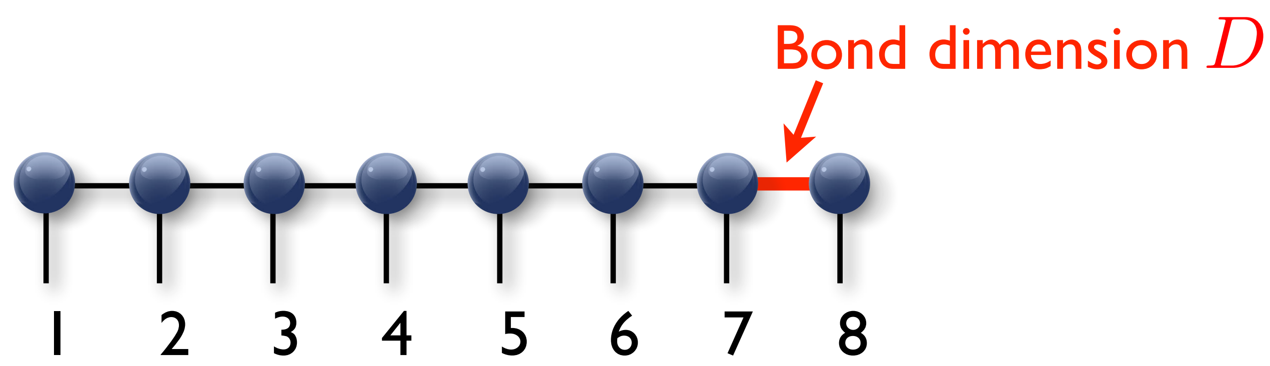
2D $S(L) \sim \alpha L$ $\chi \sim \exp(\alpha L)$

MPS & PEPS

1D

MPS

Matrix-product state



Physical indices (lattices sites)

S. R. White, PRL 69, 2863 (1992)

Fannes et al., CMP 144, 443 (1992)

Östlund, Rommer, PRL 75, 3537 (1995)

✓ Reproduces area-law in 1D

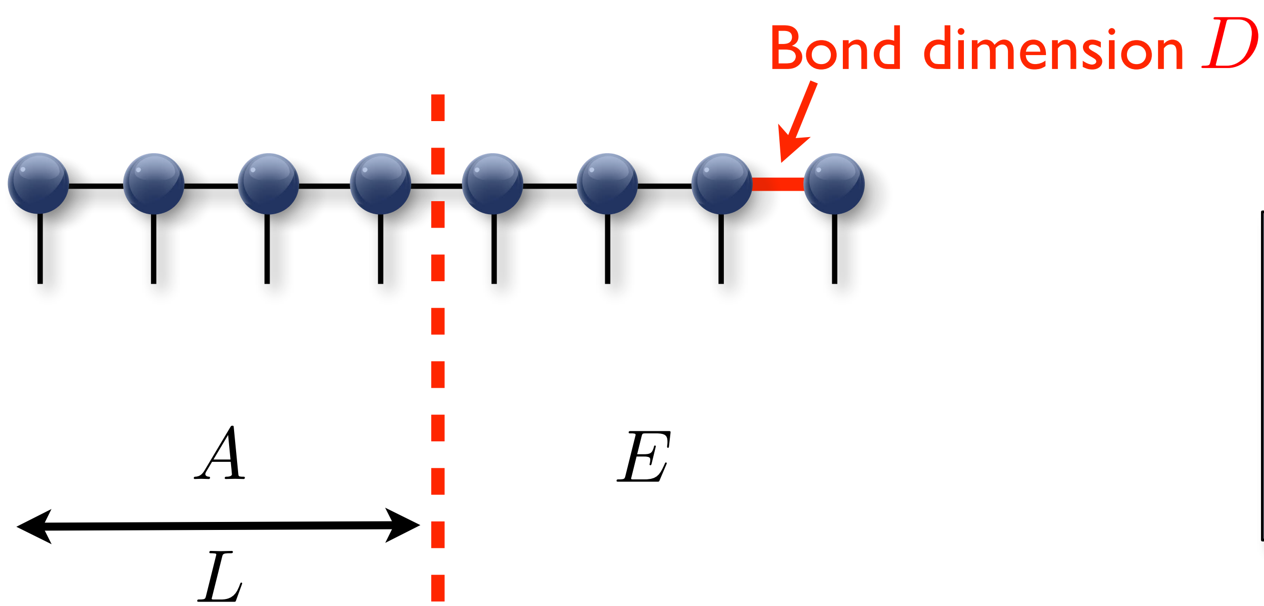
$$S(L) = const$$

MPS & PEPS

1D

MPS

Matrix-product state



➡ One bond can contribute at most log(D) to the entanglement entropy

$$rank(\rho_A) \leq D \longrightarrow S(A) \leq log(D) = const$$

$$\uparrow$$

$$S(A) = -\mathrm{tr}[\rho_A \log \rho_A] = -\sum_i \lambda_i \log \lambda_i$$

$$\lambda_i = \frac{1}{D} \rightarrow S(A) = -\sum_i^D \frac{1}{D} \log \frac{1}{D} = \log(D)$$

✓ Reproduces area-law in 1D

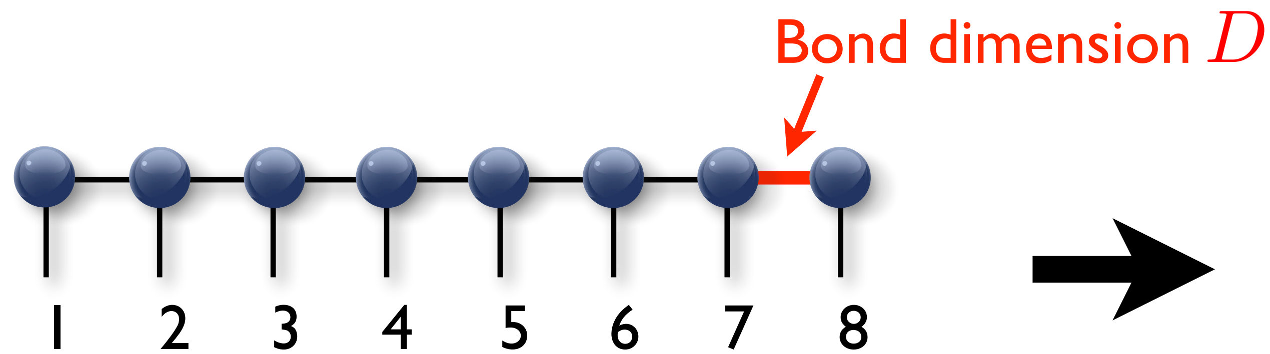
$$S(L) = const$$

MPS & PEPS

1D

MPS

Matrix-product state



Physical indices (lattices sites)

S. R. White, PRL 69, 2863 (1992)

Fannes et al., CMP 144, 443 (1992)

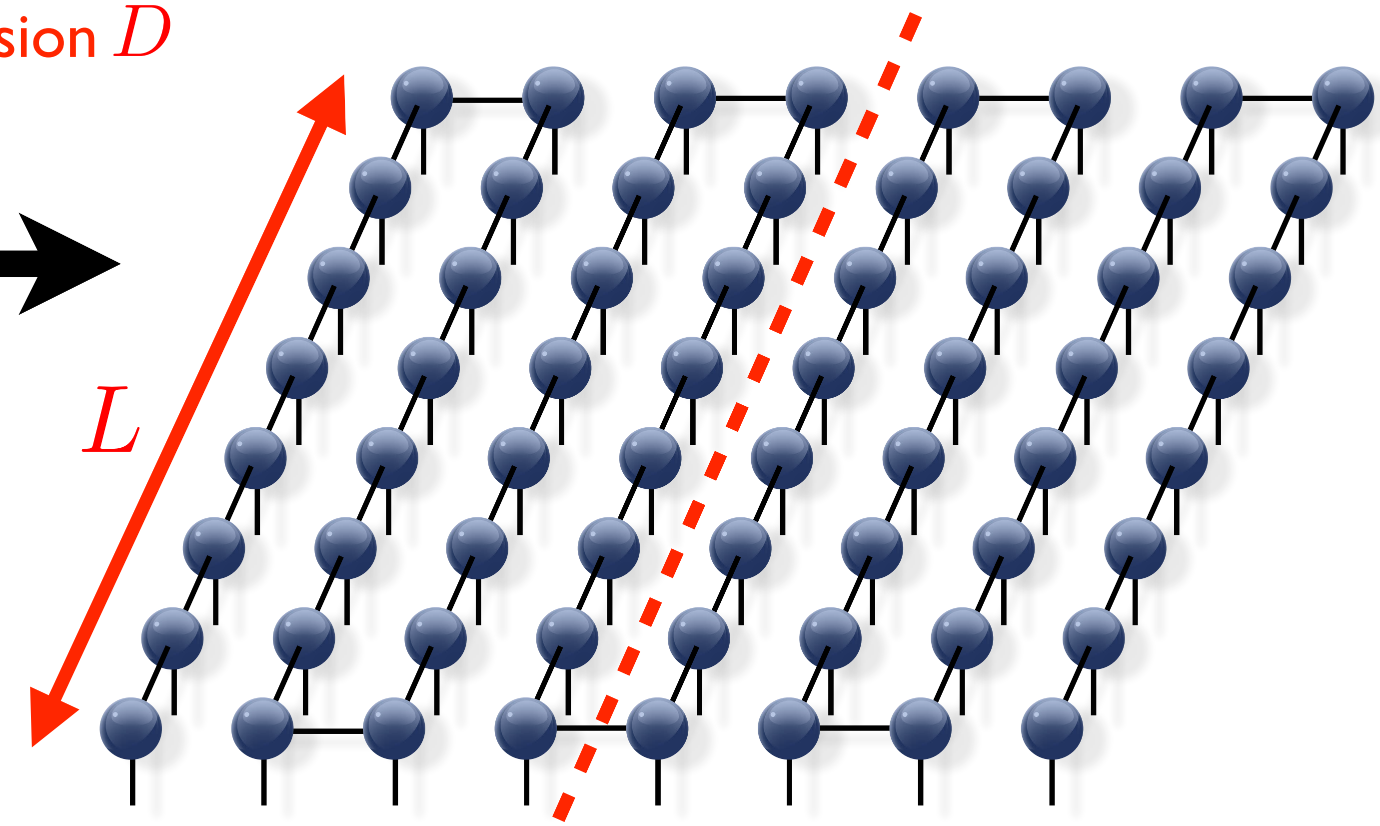
Östlund, Rommer, PRL 75, 3537 (1995)

✓ Reproduces area-law in 1D

$$S(L) = const$$

2D

can we use an MPS?



!!! Area-law in 2D !!!

$$S(L) \sim L$$

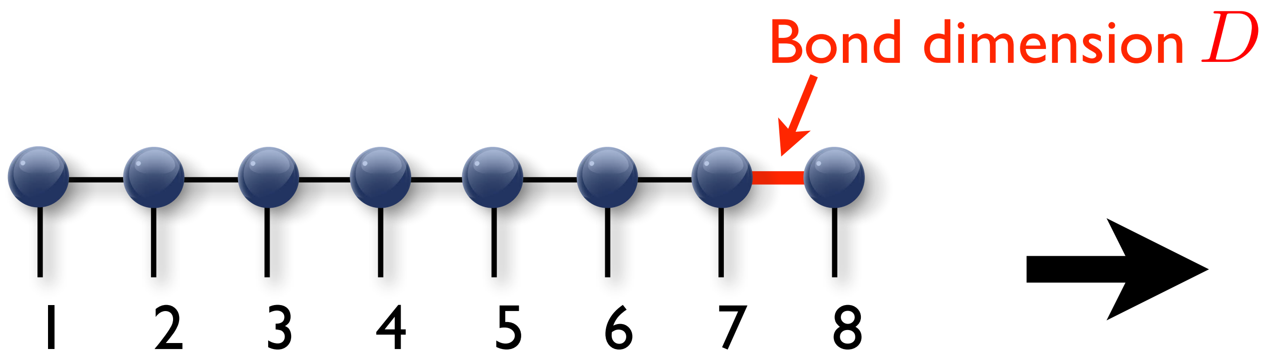
→ $D \sim exp(L)$

MPS & PEPS

1D

MPS

Matrix-product state



Physical indices (lattices sites)

S. R. White, PRL 69, 2863 (1992)

Fannes et al., CMP 144, 443 (1992)

Östlund, Rommer, PRL 75, 3537 (1995)

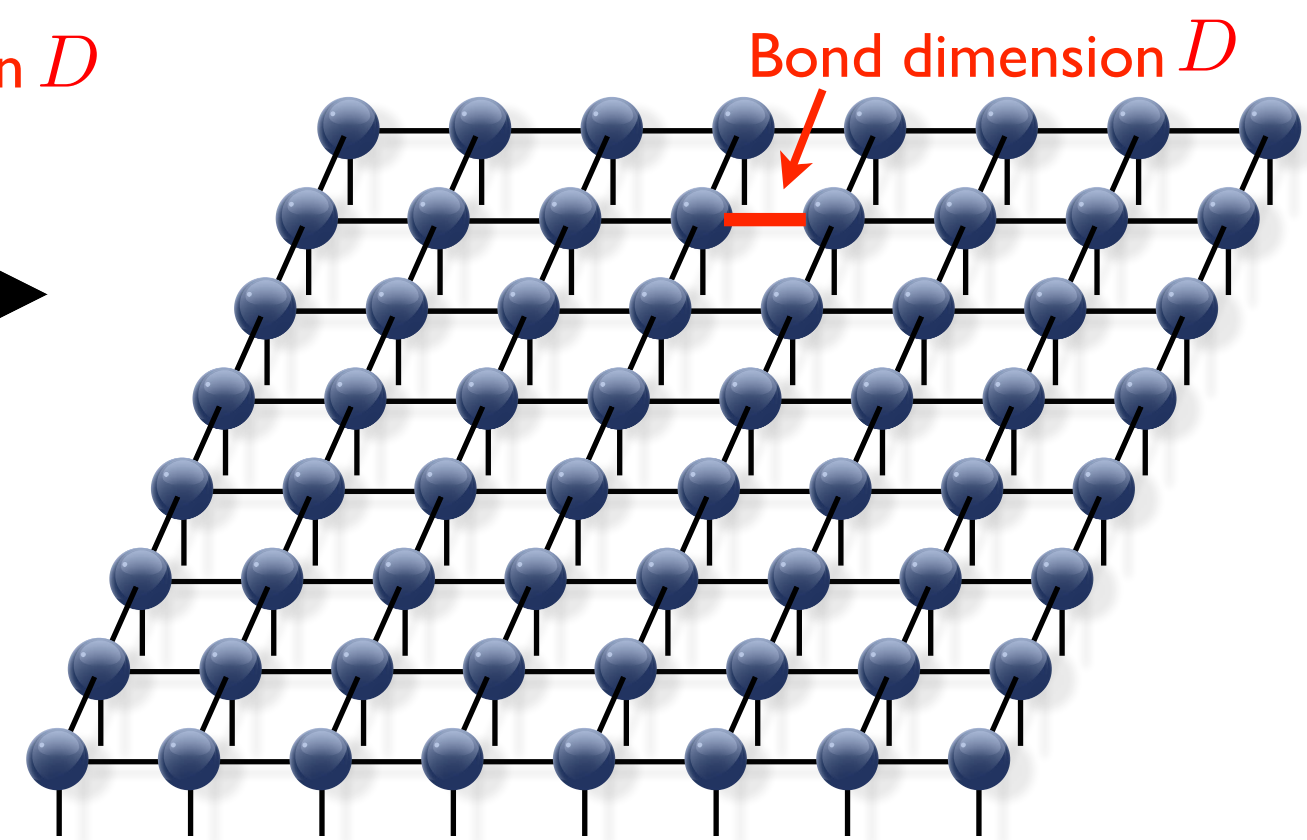
✓ Reproduces area-law in 1D

$$S(L) = const$$

2D

PEPS (TPS)

projected entangled-pair state
(tensor product state)

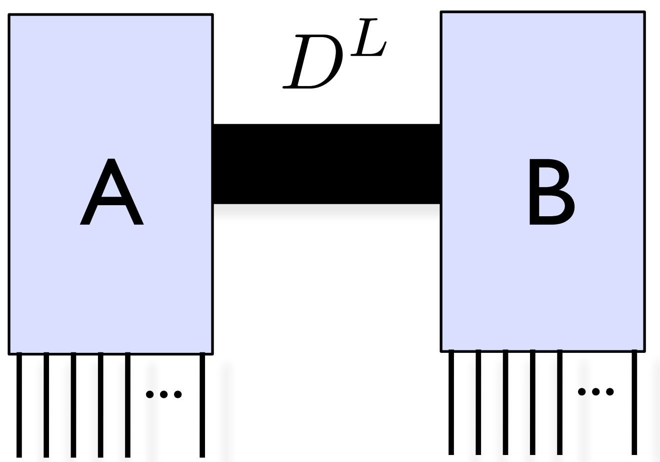


F. Verstraete, J. I. Cirac, cond-mat/0407066
Nishio, Maeshima, Gendiar, Nishino, cond-mat/0401115

✓ Reproduces area-law in 2D

$$S(L) \sim L$$

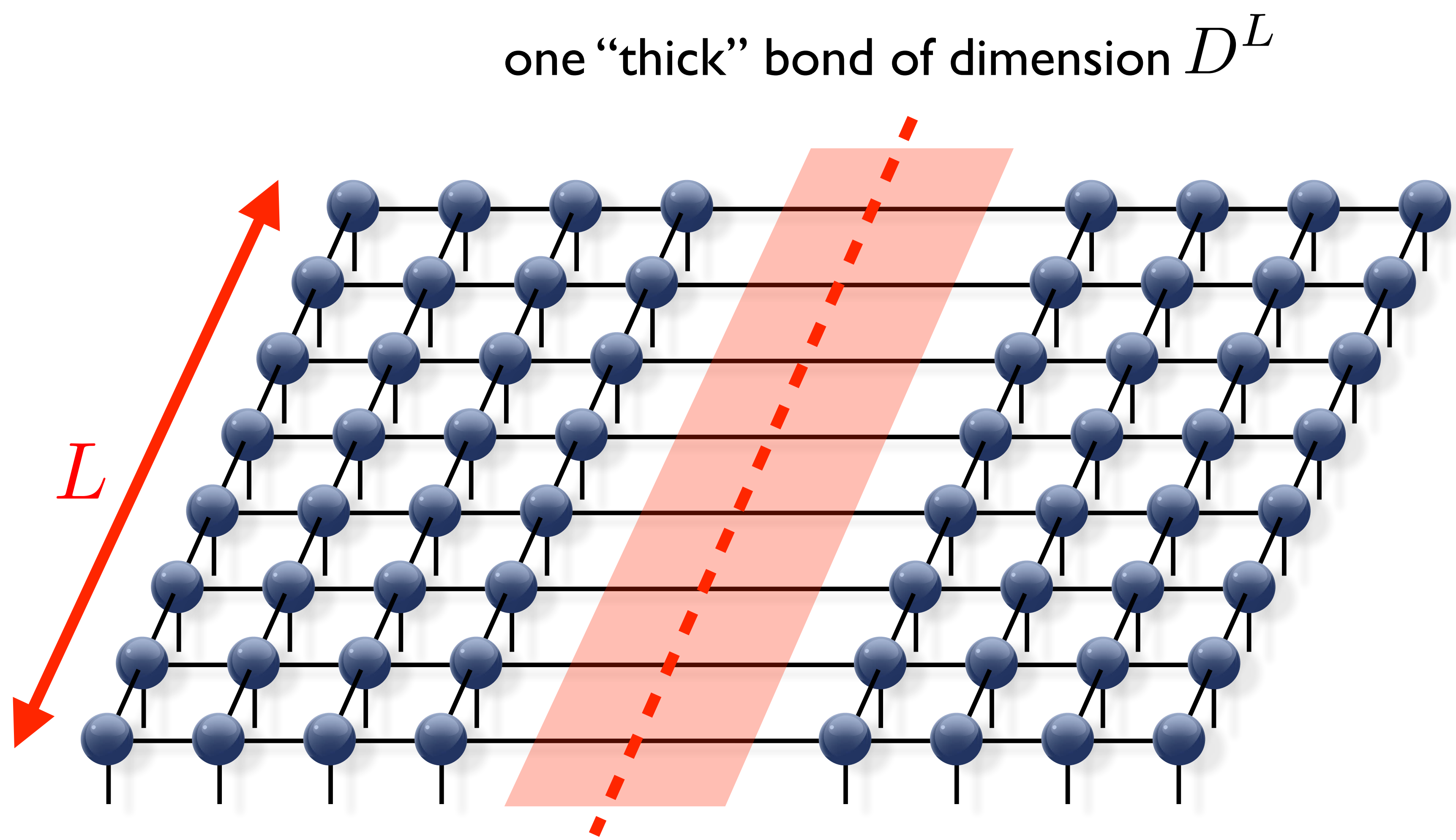
PEPS: Area law



$$S(A) \leq L \log D \sim L$$

each cut auxiliary bond can contribute (at most) log D to the entanglement entropy

The number of cuts scales with the cut length



✓ Reproduces area-law in 2D

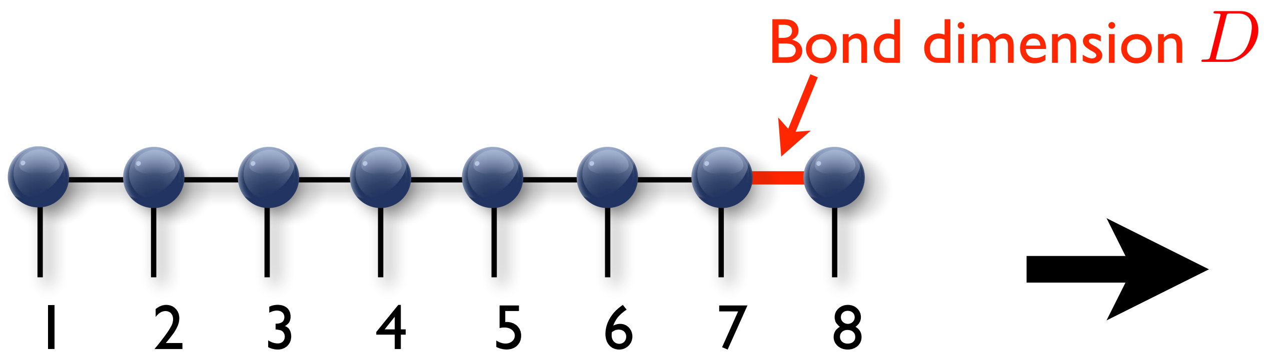
$$S(L) \sim L$$

MPS & PEPS

1D

MPS

Matrix-product state



Physical indices (lattices sites)

S. R. White, PRL 69, 2863 (1992)

Fannes et al., CMP 144, 443 (1992)

Östlund, Rommer, PRL 75, 3537 (1995)

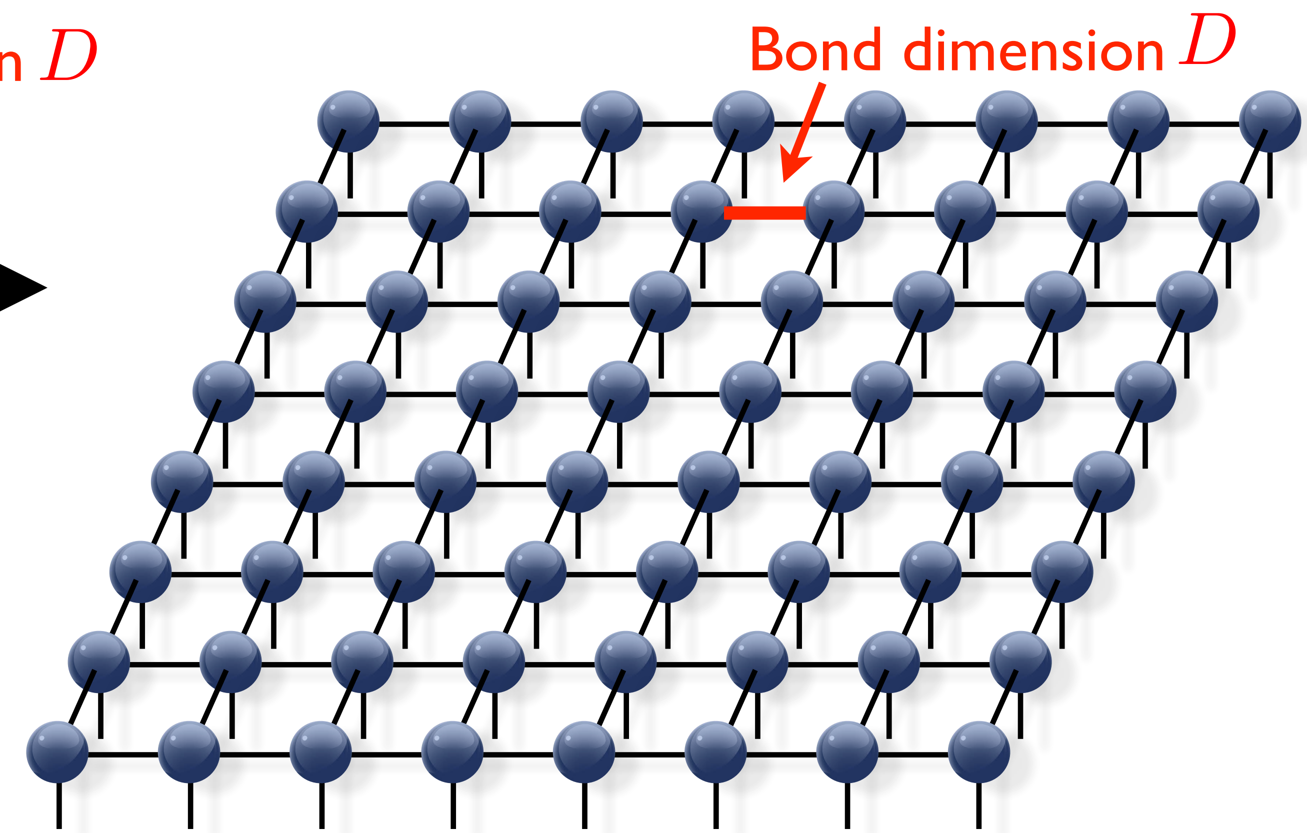
✓ Reproduces area-law in 1D

$$S(L) = const$$

2D

PEPS (TPS)

projected entangled-pair state
(tensor product state)



F. Verstraete, J. I. Cirac, cond-mat/0407066
Nishio, Maeshima, Gendiar, Nishino, cond-mat/0401115

✓ Reproduces area-law in 2D

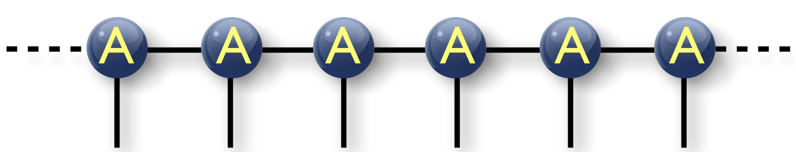
$$S(L) \sim L$$

Infinite PEPS (iPEPS)

1D

iMPS

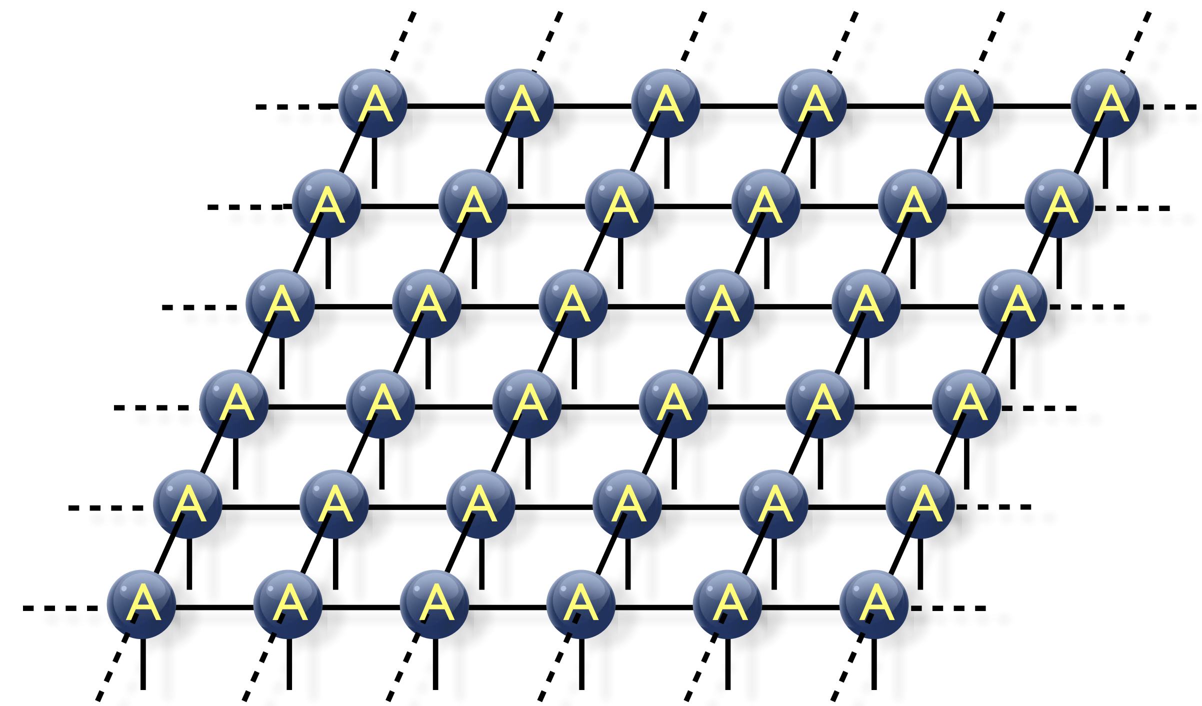
infinite matrix-product state



2D

iPEPS

infinite projected entangled-pair state



Jordan, Orus, Vidal, Verstraete, Cirac, PRL (2008)

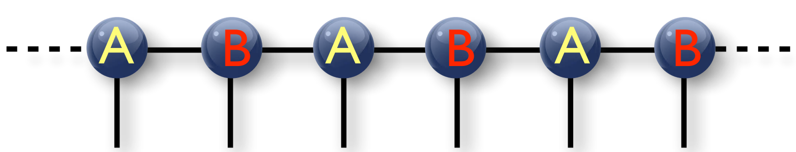
★ Work directly in the thermodynamic limit:
No finite size and boundary effects!

Infinite PEPS (iPEPS)

1D

iMPS

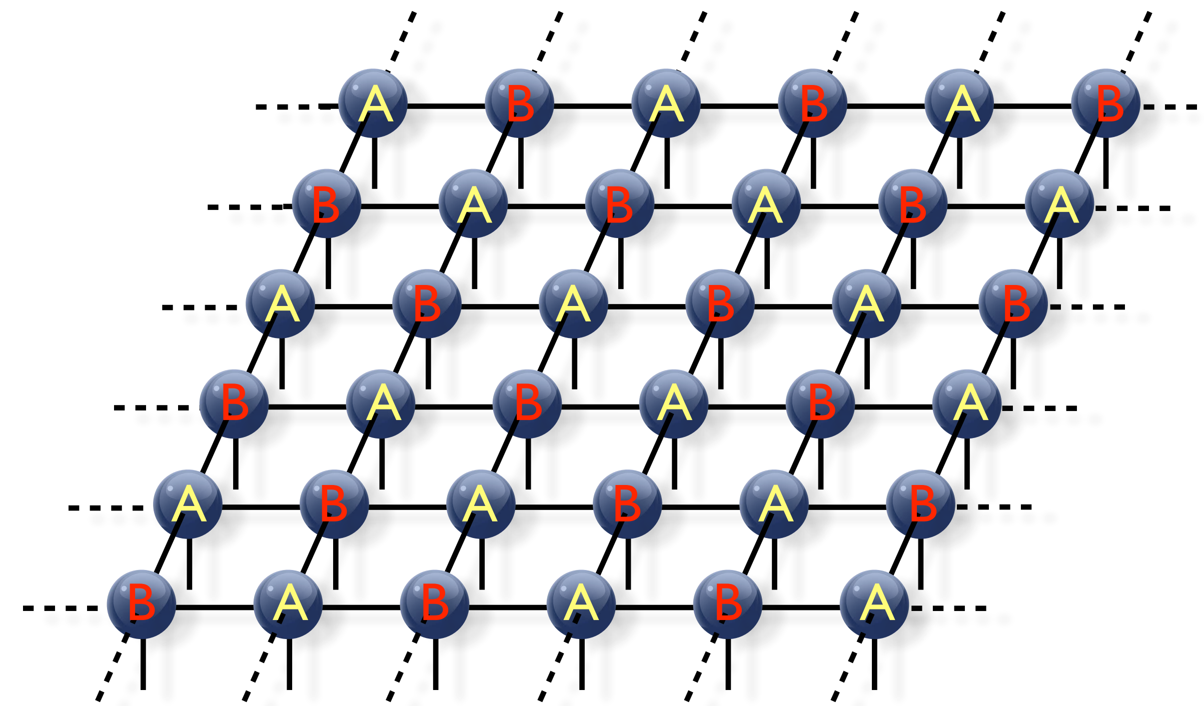
infinite matrix-product state



2D

iPEPS

infinite projected entangled-pair state



Jordan, Orus, Vidal, Verstraete, Cirac, PRL (2008)

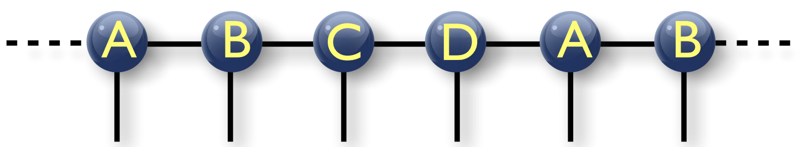
★ Work directly in the thermodynamic limit:
No finite size and boundary effects!

iPEPS with arbitrary unit cells

1D

iMPS

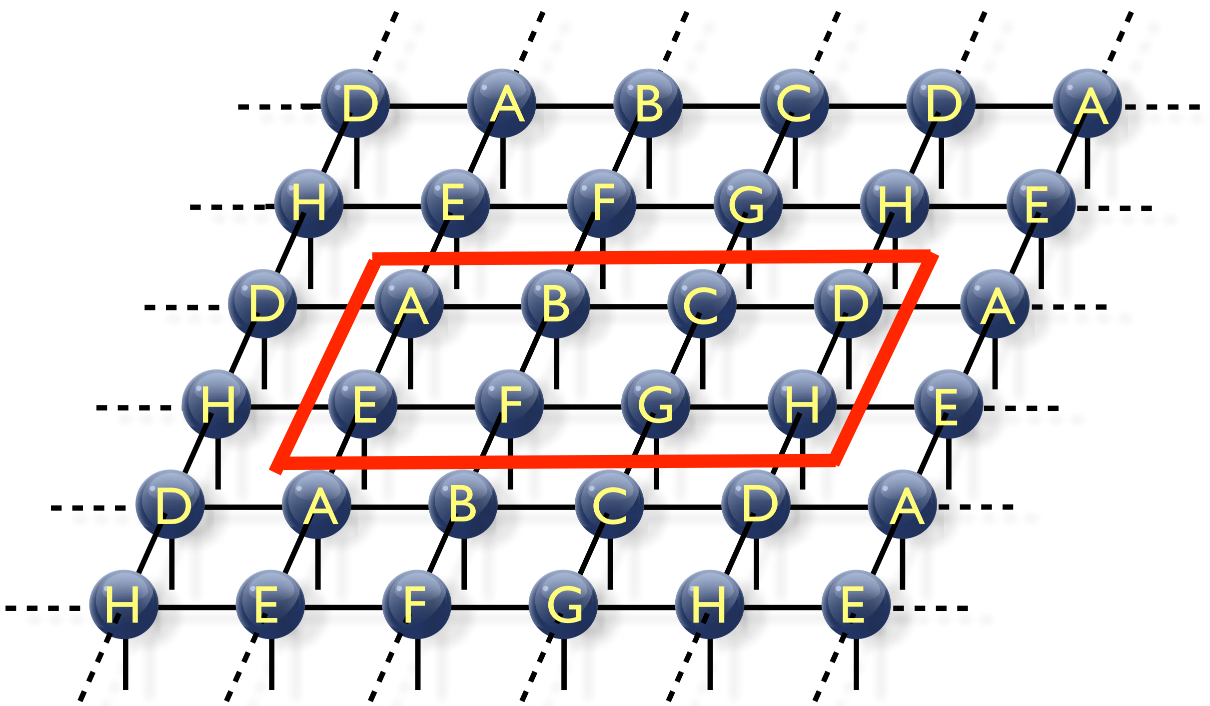
infinite matrix-product state



2D

iPEPS

with arbitrary unit cell of tensors



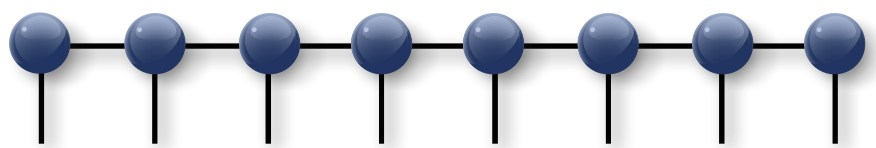
here: 4x2 unit cell

PC, White, Vidal, Troyer, PRB **84** (2011)

★ Run simulations with different unit cell sizes and compare variational energies

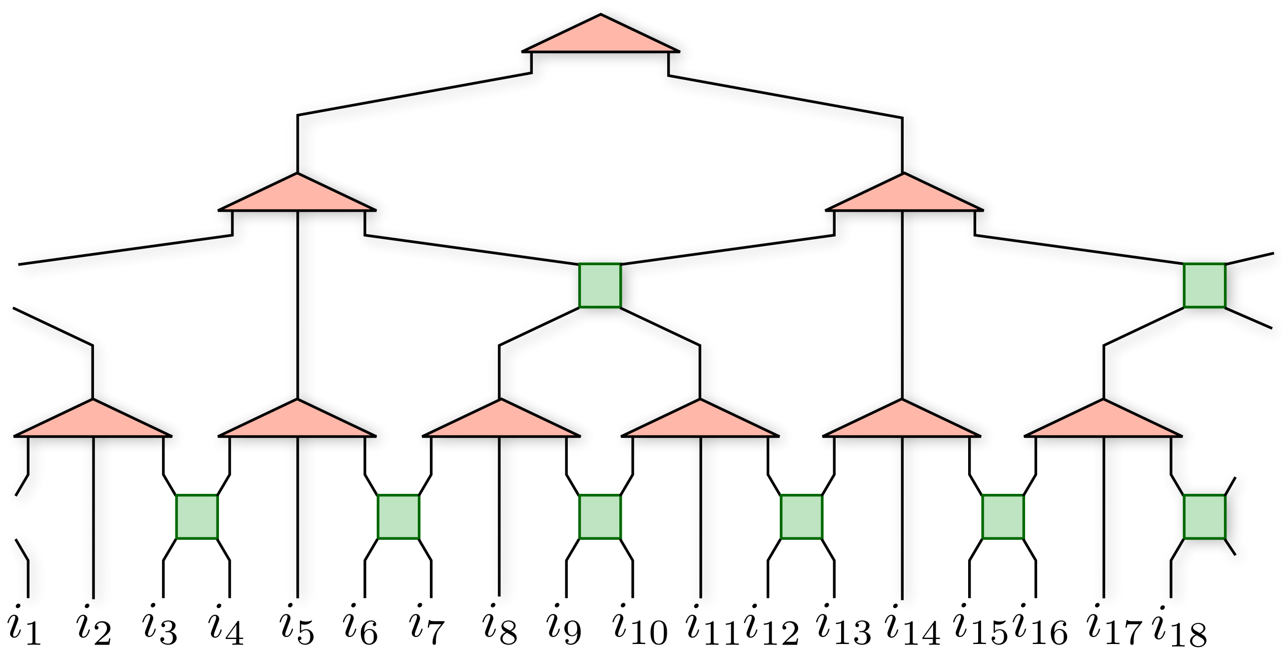
Hierarchical tensor networks (tree TN/MERA)

MPS



“flat”

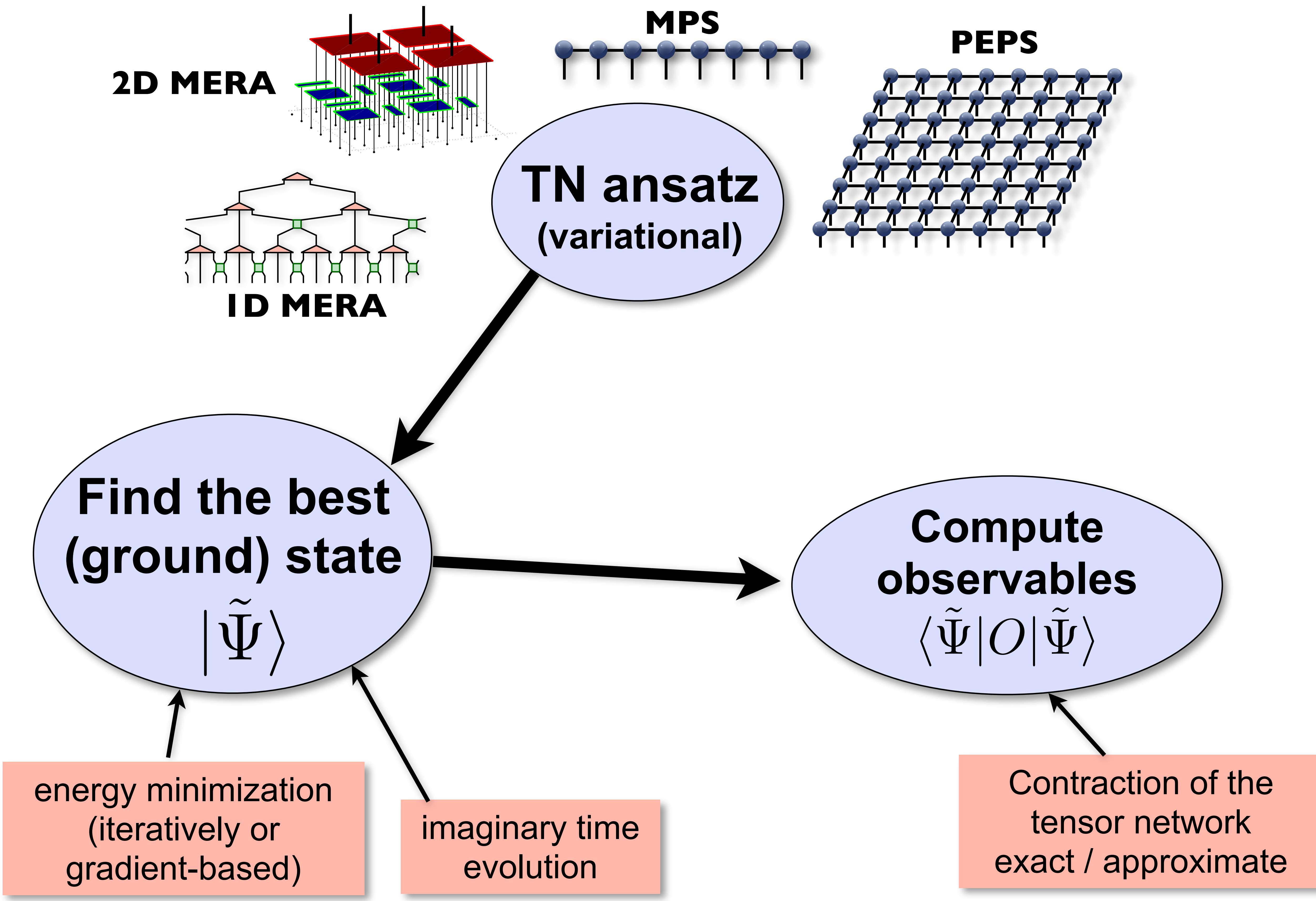
MERA



tensors at different length scales

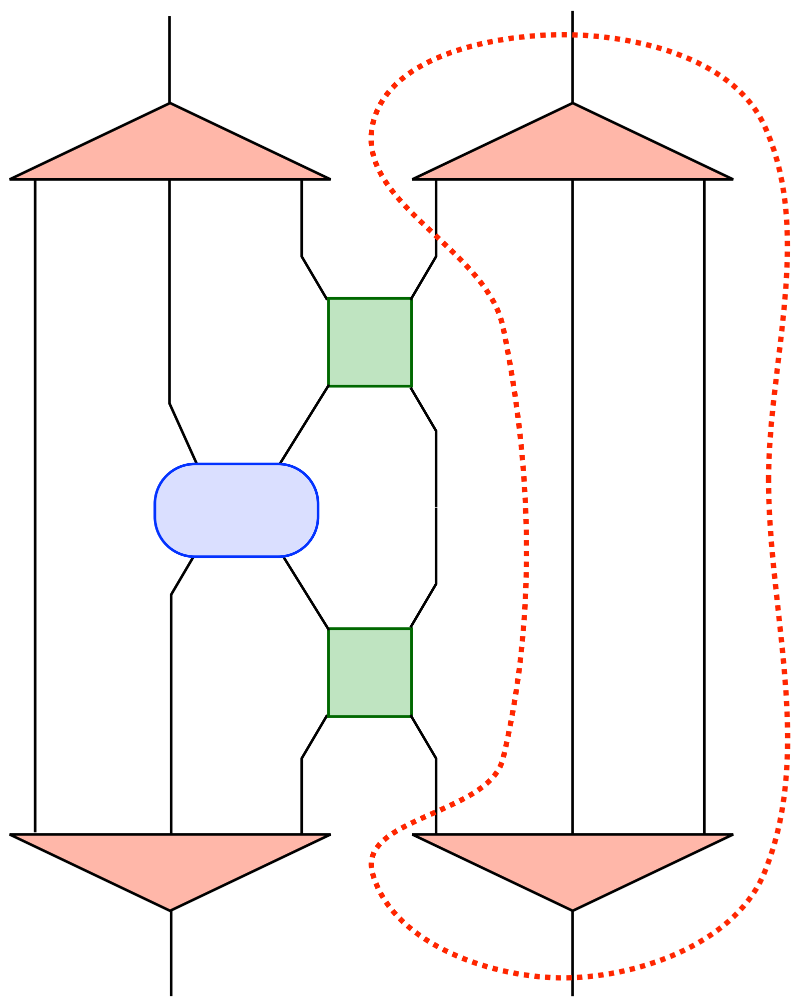
★ Powerful ansatz for critical systems!
(reproduces *S(L) ~ log L* scaling)

Overview: Tensor network algorithms (ground state)

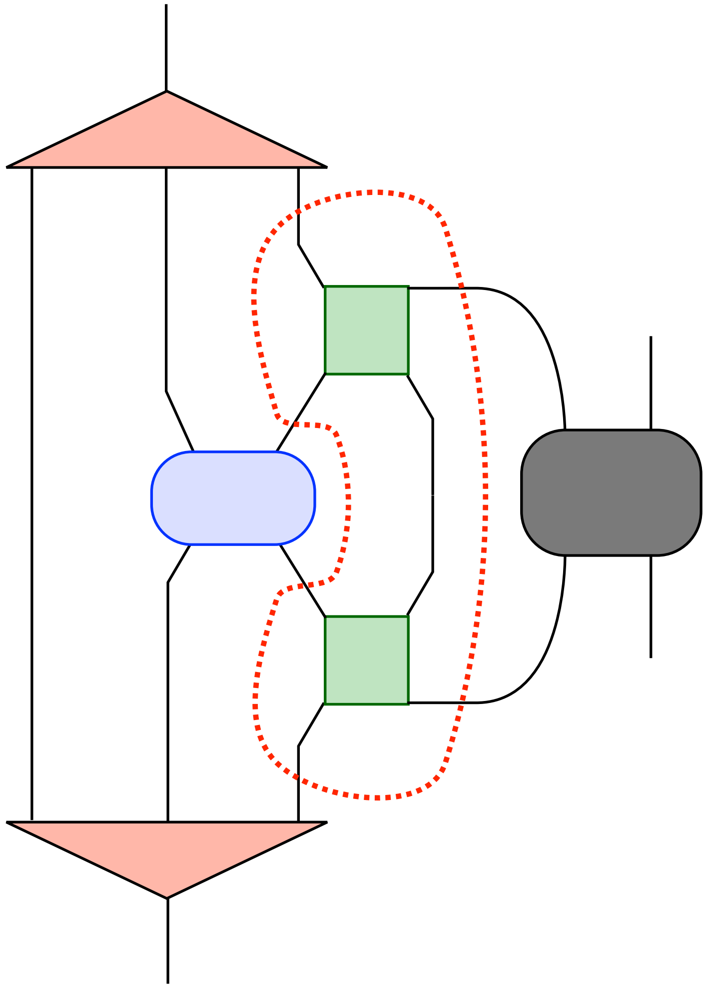


Contraction

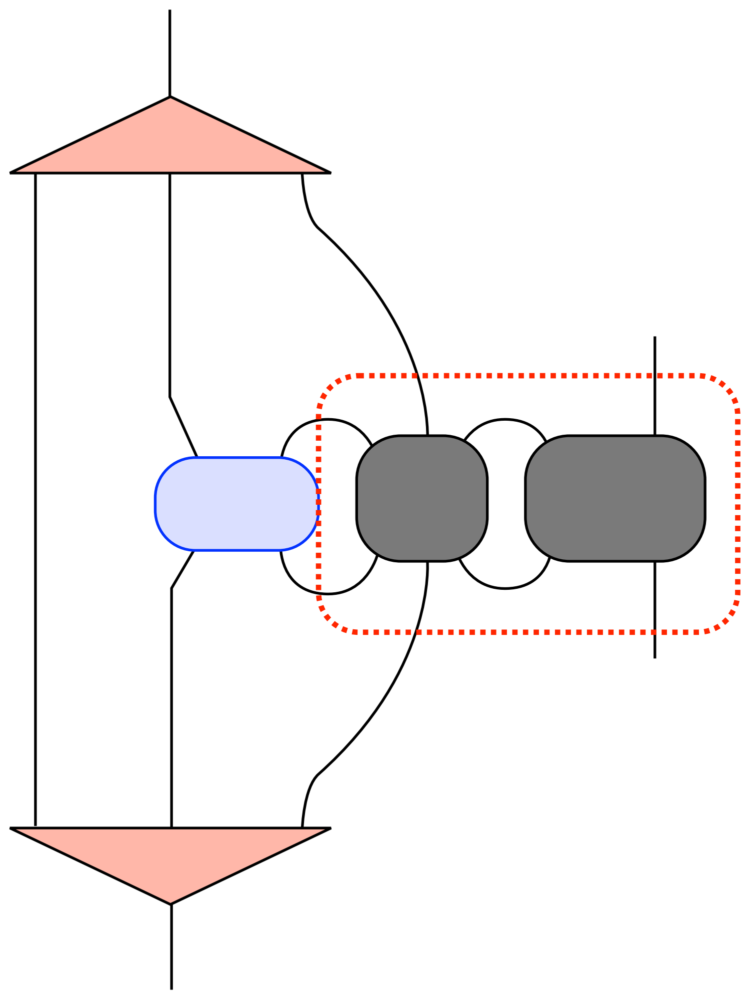
Contracting a tensor network



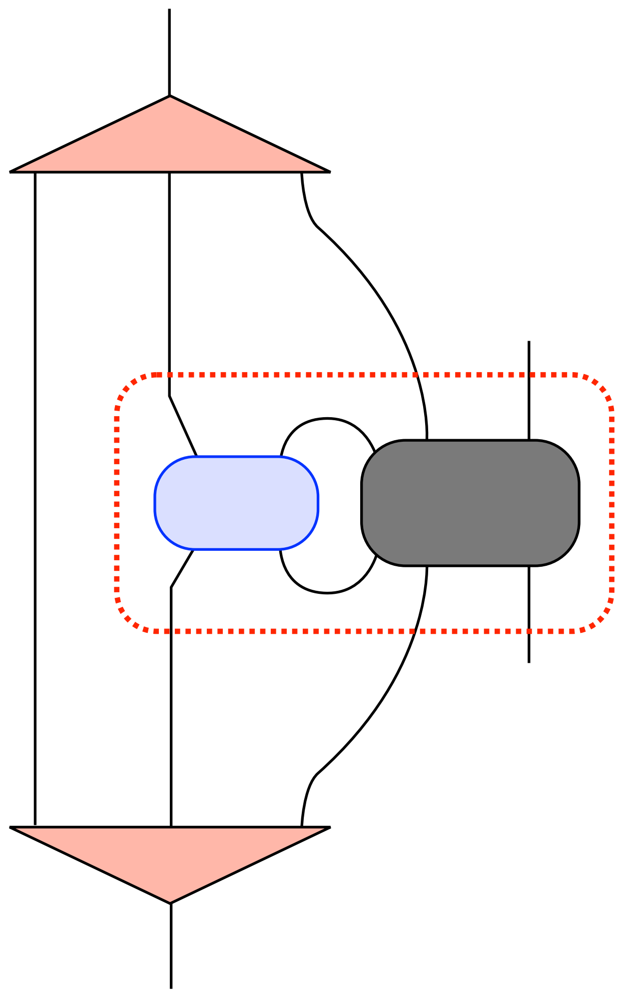
Pairwise contractions...



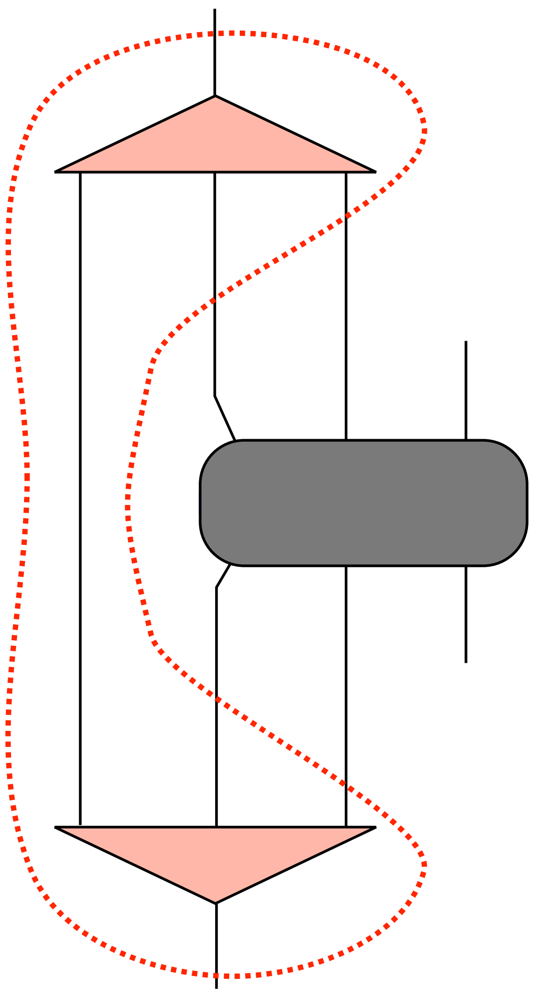
Pairwise contractions...



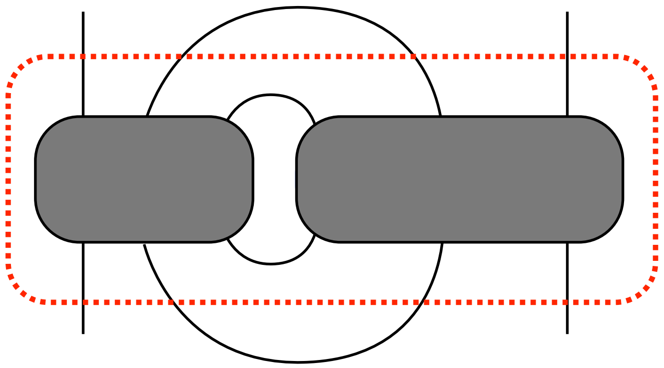
Pairwise contractions...



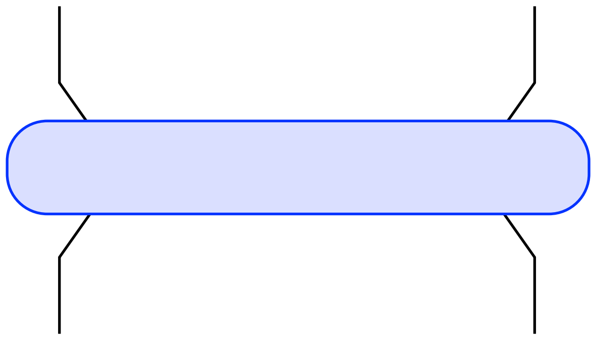
Pairwise contractions...



Pairwise contractions...



Pairwise contractions...

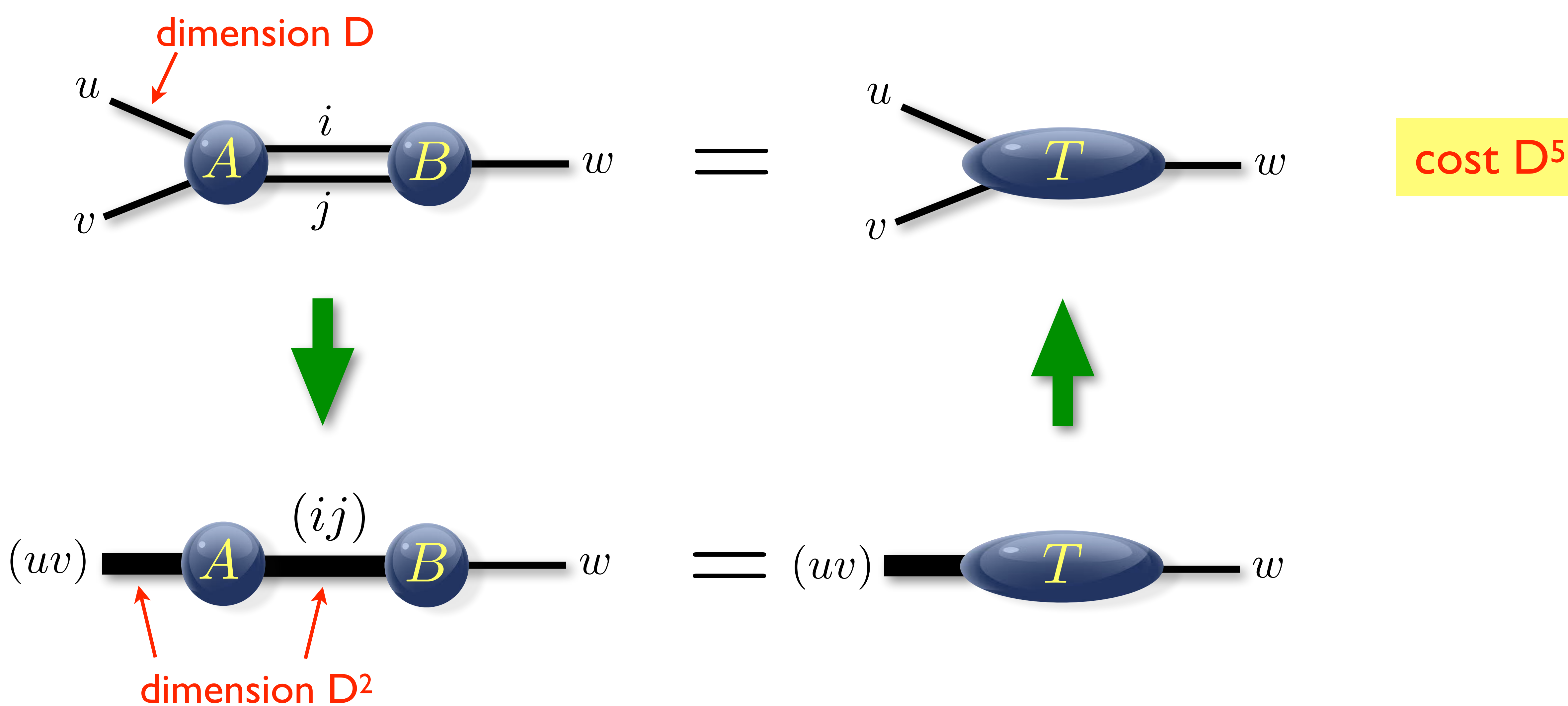


done!

the order of contraction matters for the computational cost!!!

Contracting a tensor network

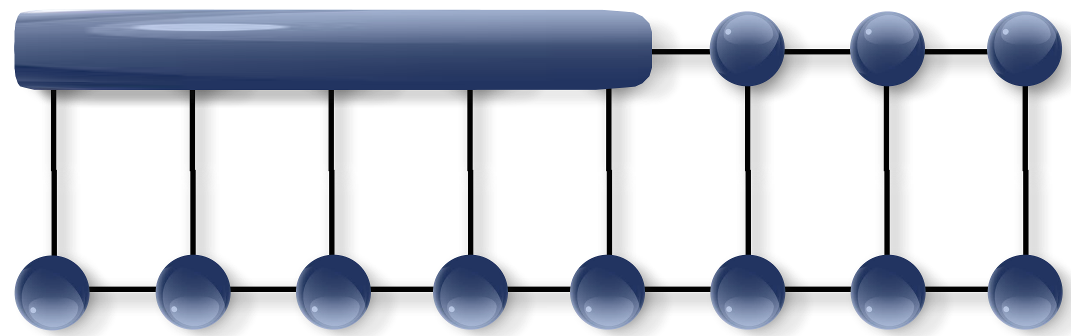
★ Reshape tensors into matrices and multiply them with optimized routines (BLAS)



★ Computational cost: multiply the dimensions of all legs (connected legs only once)

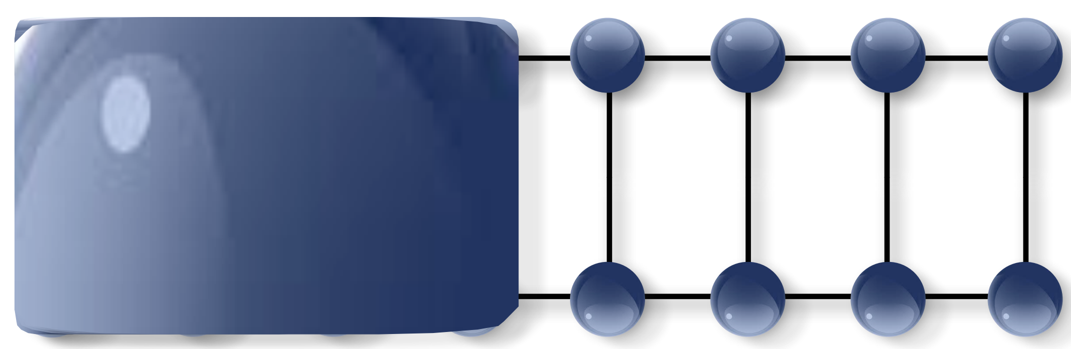
Contracting an MPS

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



BAD!

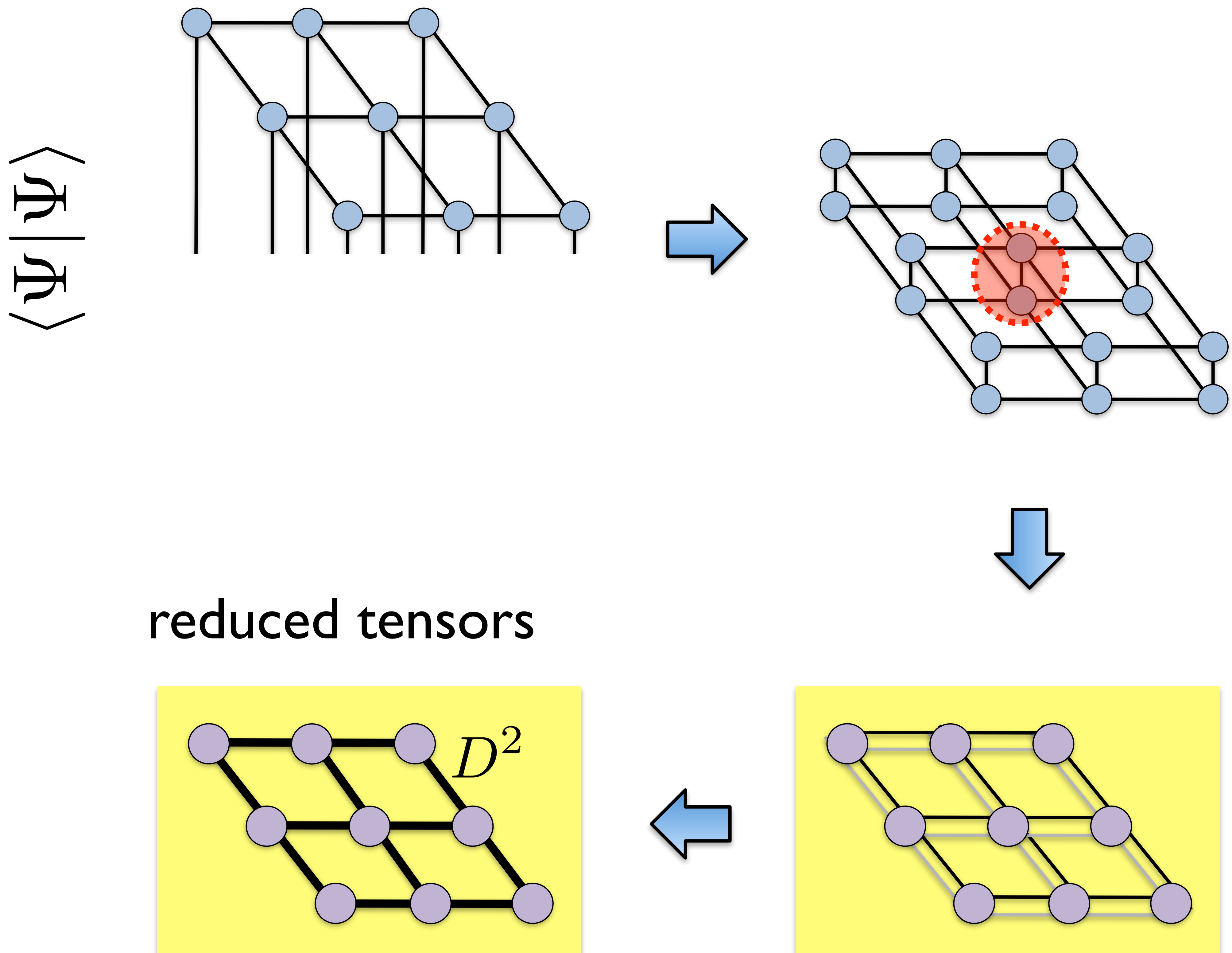
$\langle \Psi | \Psi \rangle =$



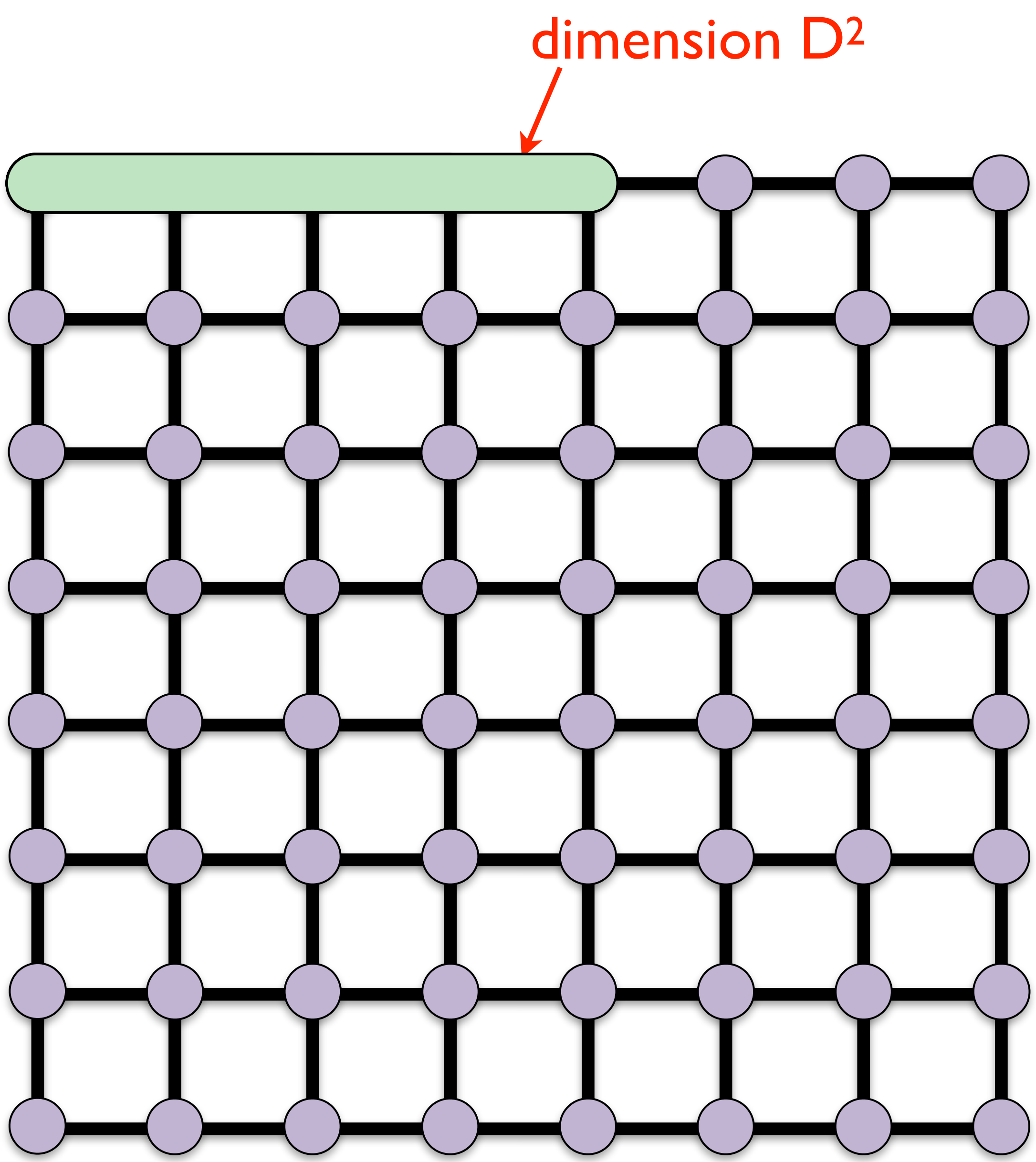
Good!

cost: $O(D^3)$

Contracting the PEPS



Contracting the PEPS



Problem:

exact contraction ~ D^{2L}

O(exp(L))

NOT EFFICIENT!

Contracting the PEPS

★ Exact contraction of an PEPS is exponentially hard!

→ *use controlled approximate contraction scheme*

MPS-MPO-based / VUMPS

Murg, Verstraete, Cirac, PRA75 '07
Jordan, et al. PRL79 (2008)
Haegeman & Verstraete (2017)
…

Corner transfer matrix method

Nishino, Okunishi, JPSJ65 (1996)
Orus, Vidal, PRB 80 (2009)
Fishman et al, PRB 98 (2018)
…

TRG

Tensor Renormalization Group
(variants: HOTRG, SRG, HOSRG)

Levin, Nave, PRL99 (2007)
Xie et al. PRL 103 (2009)
Xie et al. PRB 86 (2012), …

↓

TNR

Tensor Network Renormalization
Evenbly & Vidal, PRL 115 (2015)

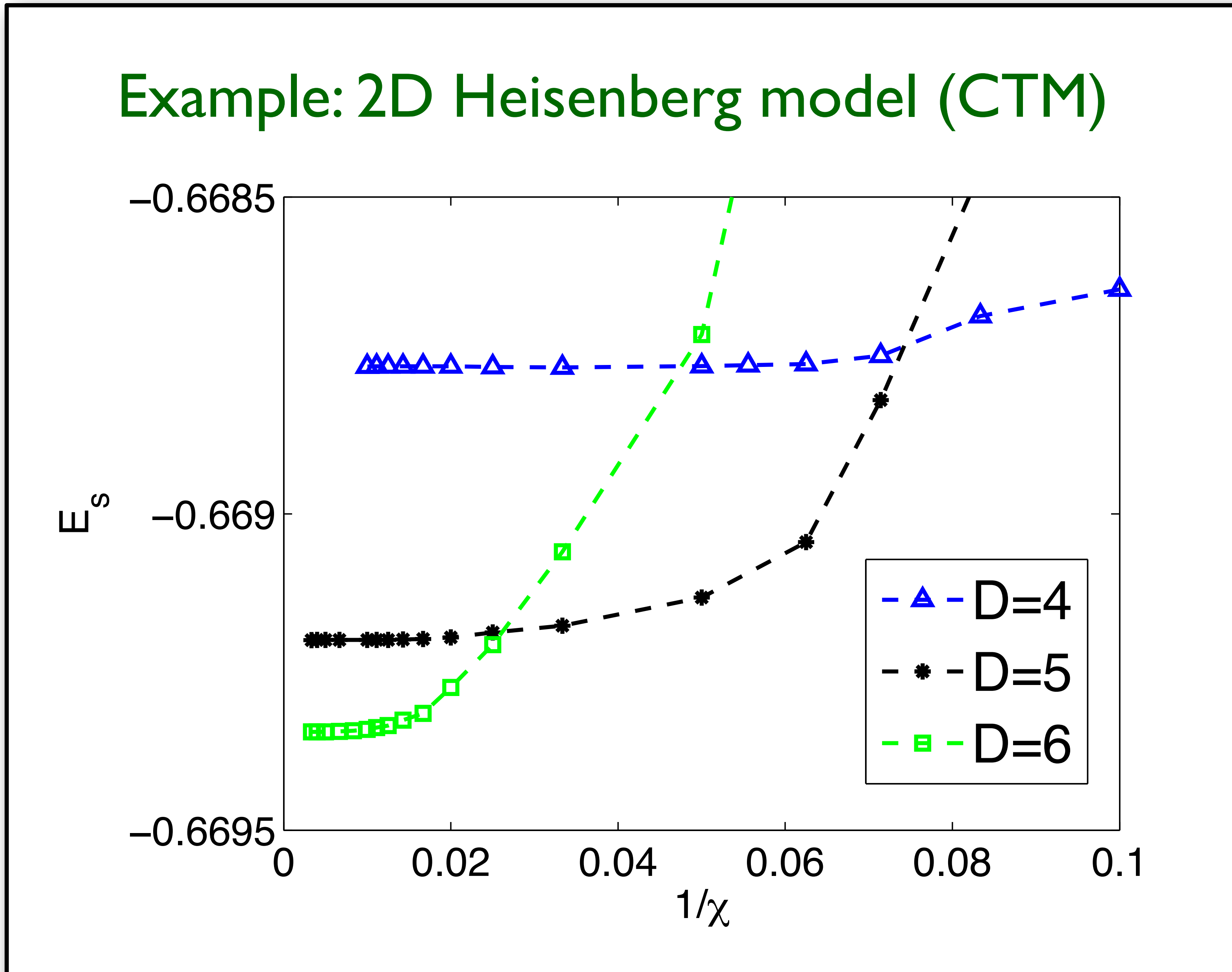
Loop-TNR:
Yang, Gu & Wen, PRL 118 (2017)

★ Accuracy of the approximate contraction is controlled by "boundary dimension" χ

★ Convergence in χ needs to be carefully checked

★ Overall cost: $\mathcal{O}(D^{10\ldots14})$ with $\chi \sim D^2$

Contracting the PEPS



★ Fast convergence

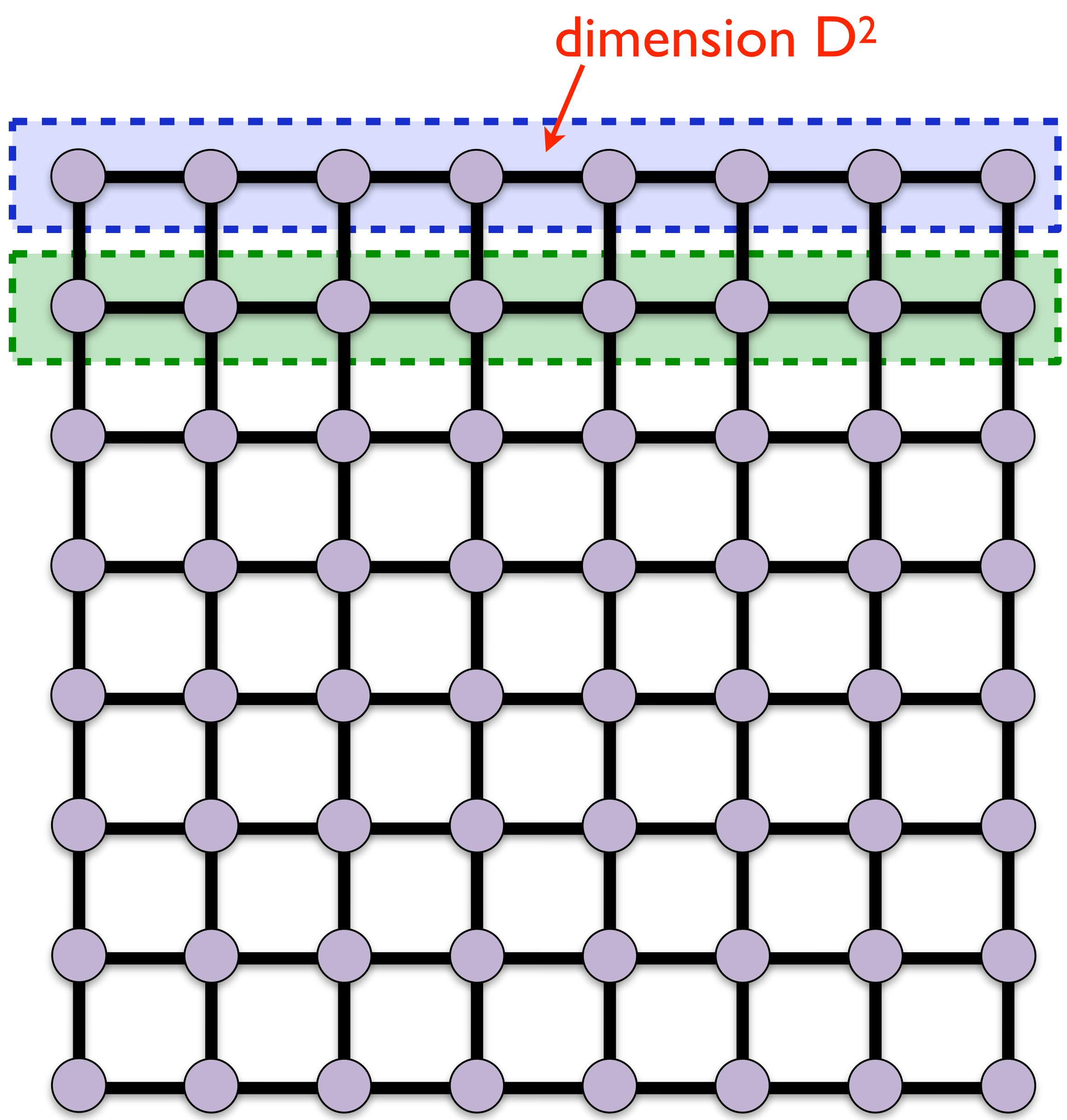
★ Effect of finite D is much larger!

★ Be careful with "variational" energy!!!

★ Convergence needs to be checked!

Contracting the PEPS using an MPS

Verstraete, Murg, Cirac, Adv. in Phys. 57, 143 (2008)

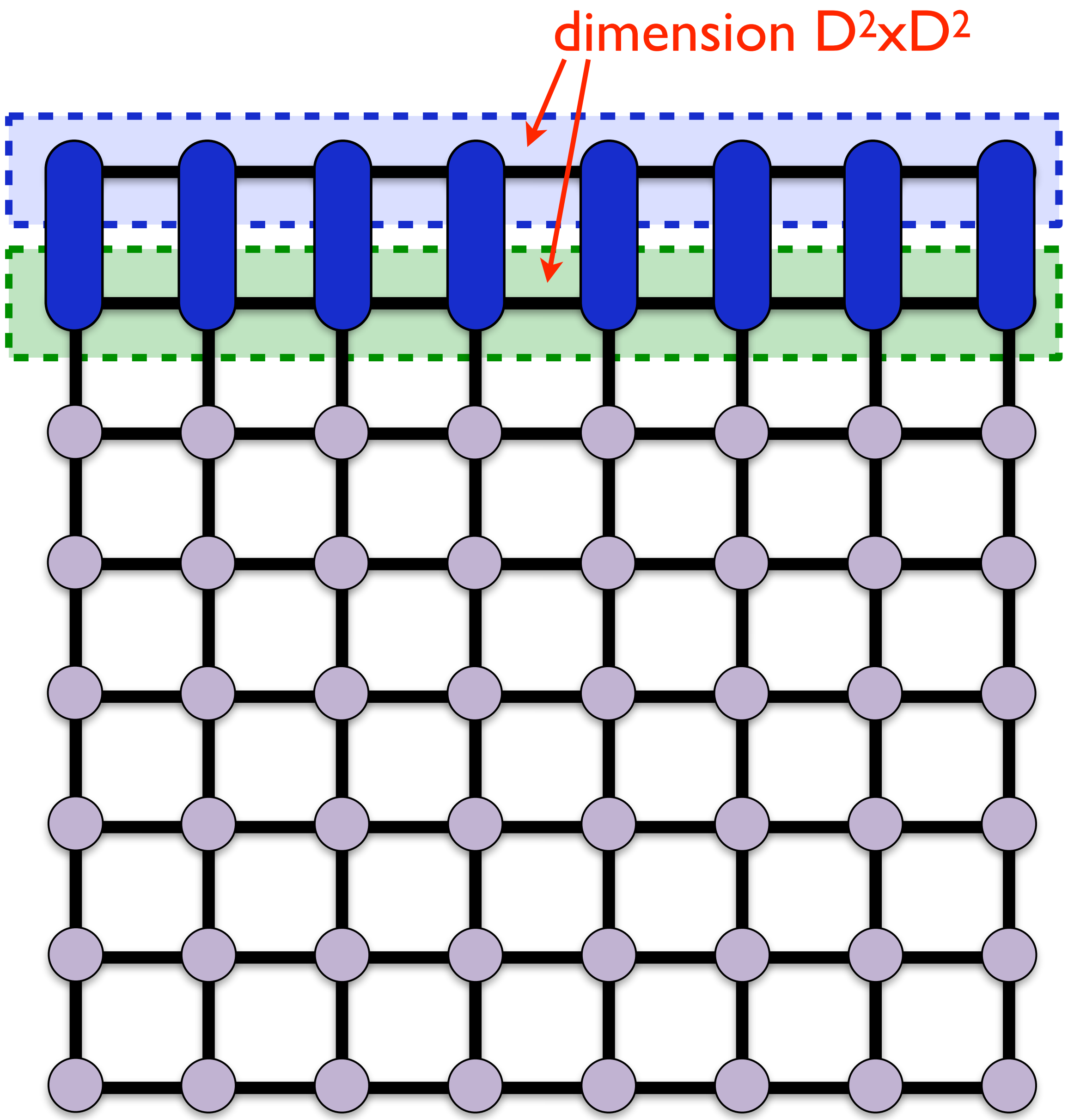


this is an MPS

this is an MPO (matrix product operator)

Contracting the PEPS using an MPS

Verstraete, Murg, Cirac, Adv. in Phys. 57, 143 (2008)



this is an MPS with bond dimension $D^2 \times D^2$

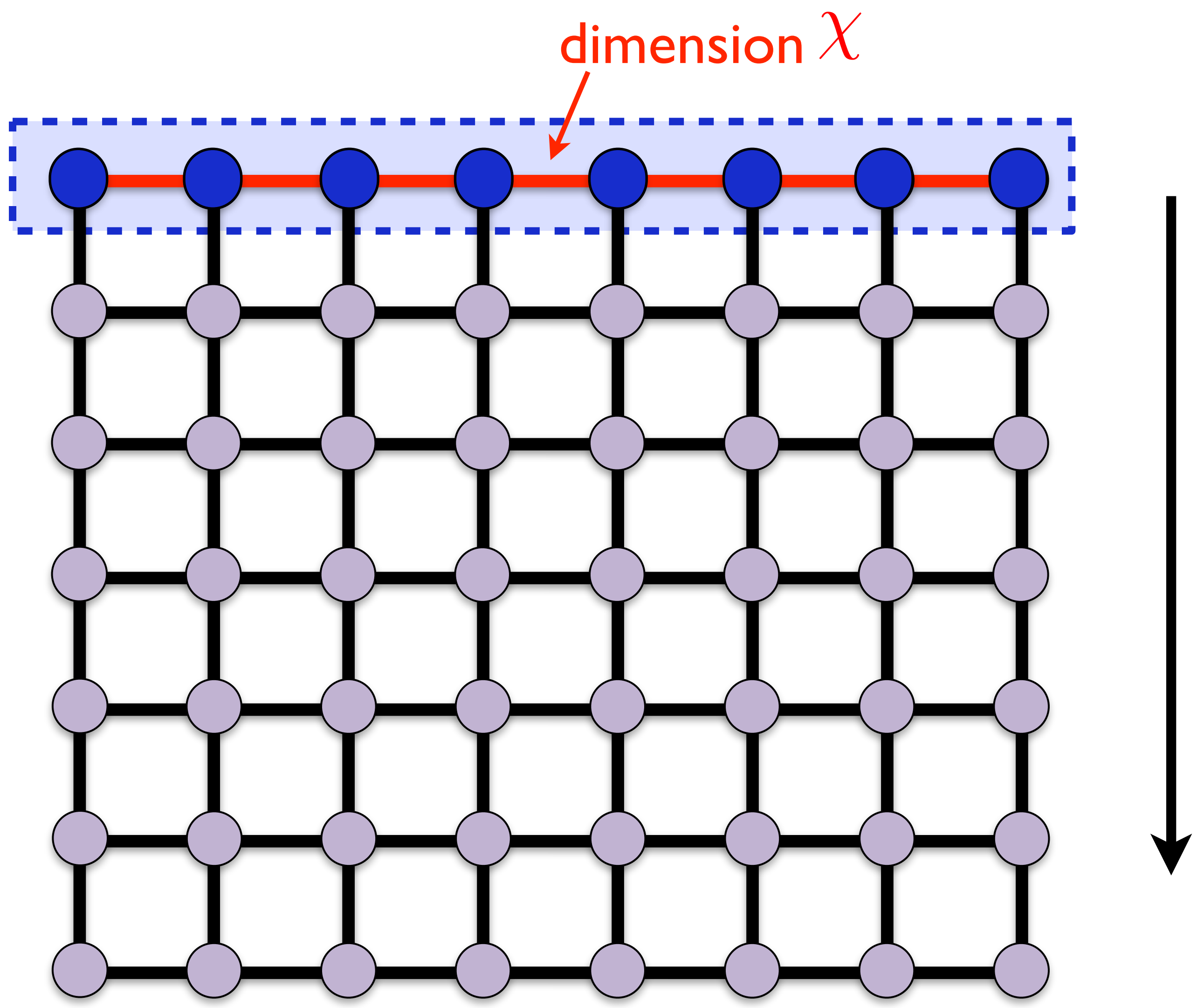
truncate the bonds to χ

there are different techniques for the efficient MPO-MPS multiplication (SVD, variational optimization, zip-up algorithm...)

Schollwöck, Annals of Physics 326, 96 (2011)
Stoudenmire, White, New J. of Phys. 12, 055026 (2010).

Contracting the PEPS using an MPS

Verstraete, Murg, Cirac, Adv. in Phys. 57, 143 (2008)

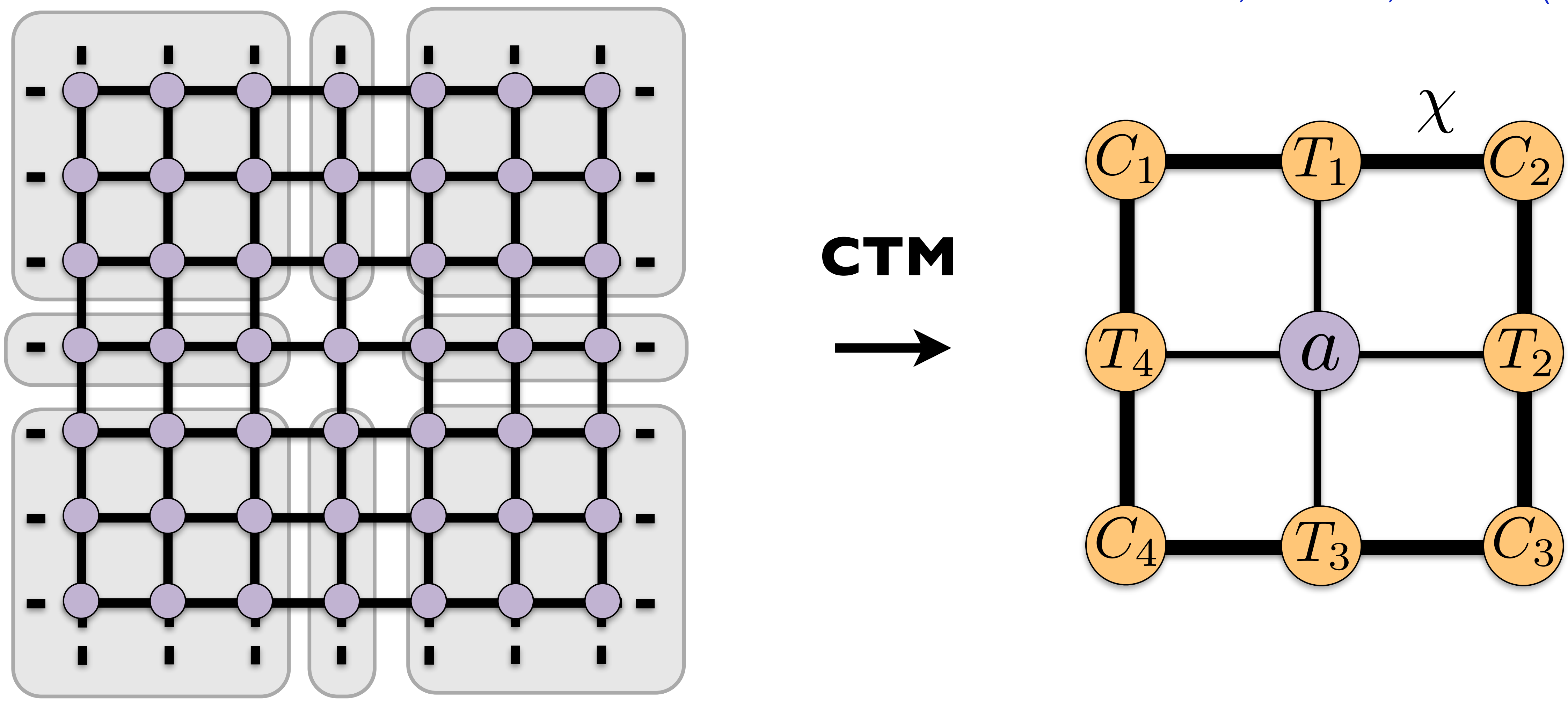


proceed...

- ★ We can do this from several directions
- ★ Similar procedure when computing an expectation value

Contracting the iPEPS using the corner transfer matrix method

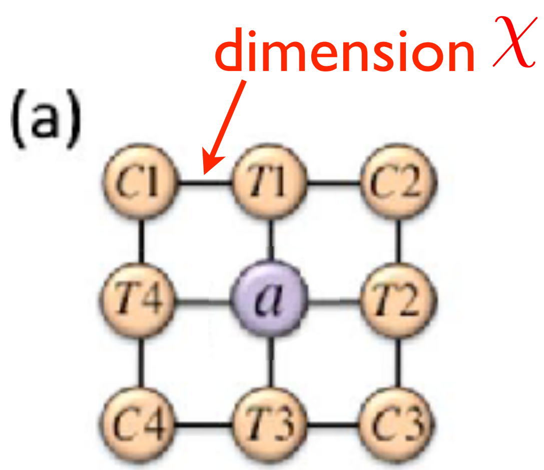
Baxter, J. Math. Phys. 9 (1968)
Nishino, Okunishi, JPSJ 65 (1996)



- Environment tensors account for infinite system around a bulk site
- CTM: Compute environment in an iterative way
- Accuracy can be systematically controlled with χ

Contracting the iPEPS using the corner transfer matrix method

Baxter, J. Math. Phys. 9 (1968)
Nishino, Okunishi, JPSJ65 (1996)
Orus, Vidal, PRB 80 (2009)

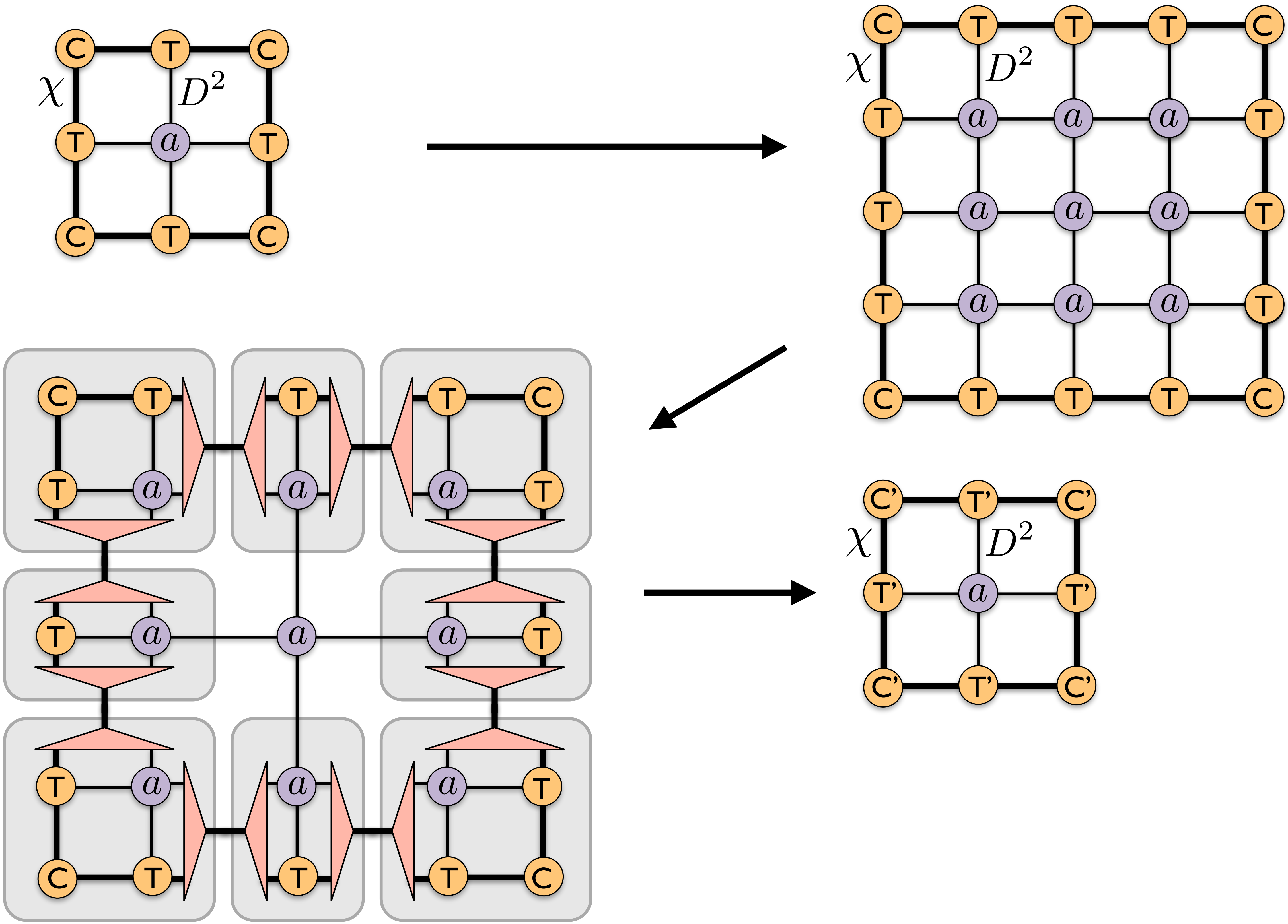


- ★ Let the system grow in all directions.
- ★ Repeat until convergence is reached
- ★ The boundary tensors form the **environment**
- ★ Can be generalized to arbitrary unit cell sizes

Corboz, et al., PRB 84 (2011)

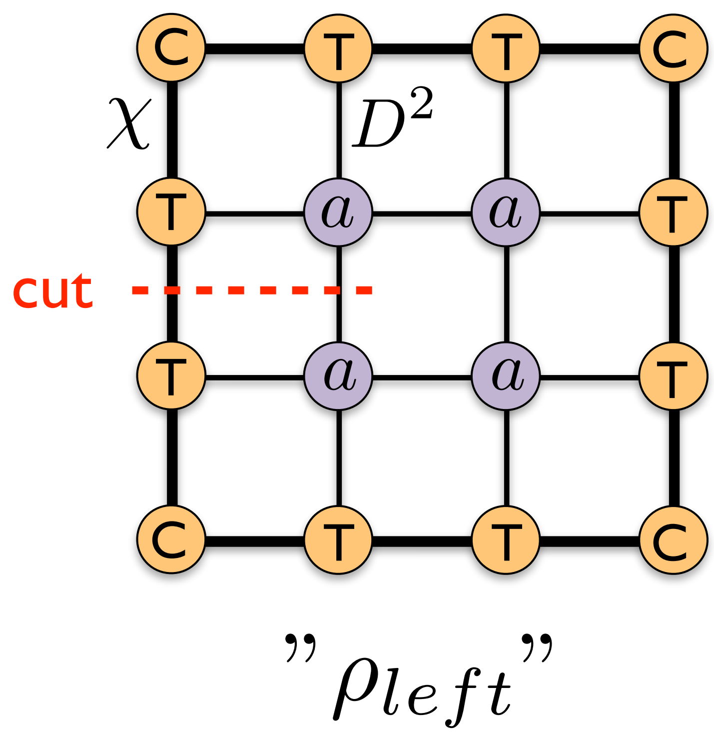
Simplest case: rotational & mirror symmetric tensors

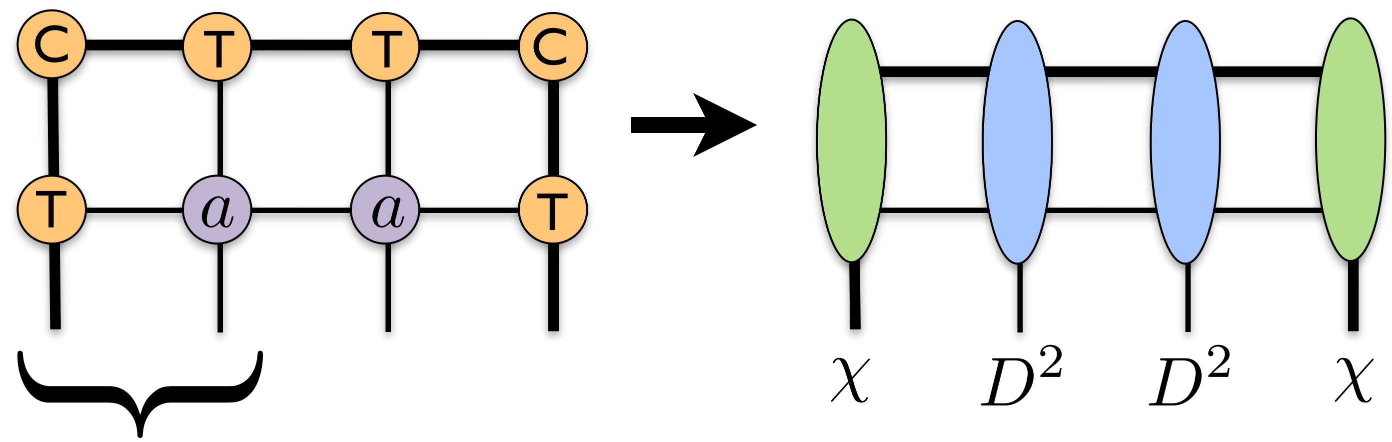
Nishino, Okunishi, JPSJ65 (1996)



Simplest case: rotational&mirror symmetric tensors

Nishino, Okunishi, JPSJ65 (1996)

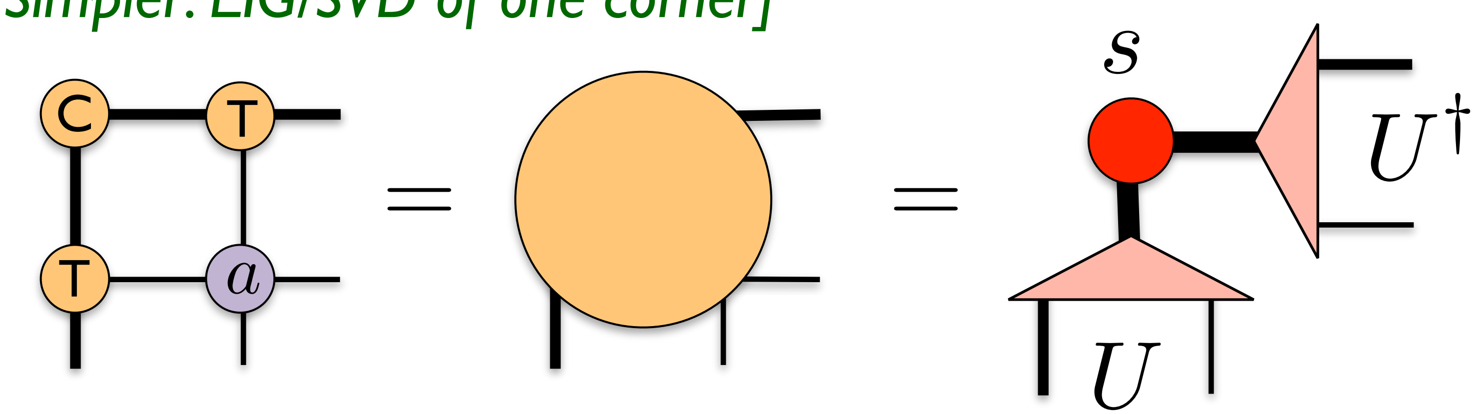




Relevant subspace?

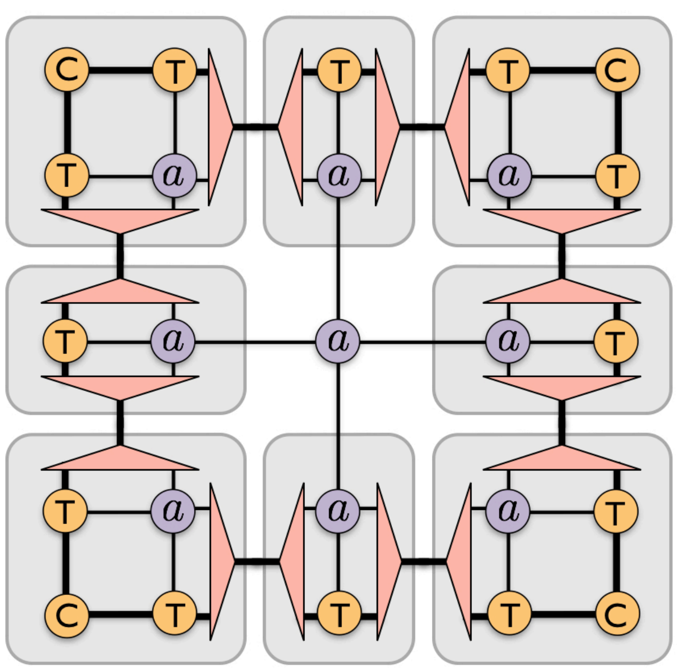
DMRG: Eigenvectors with largest eigenvalues of ρ_{left}

[Simpler: EIG/SVD of one corner]

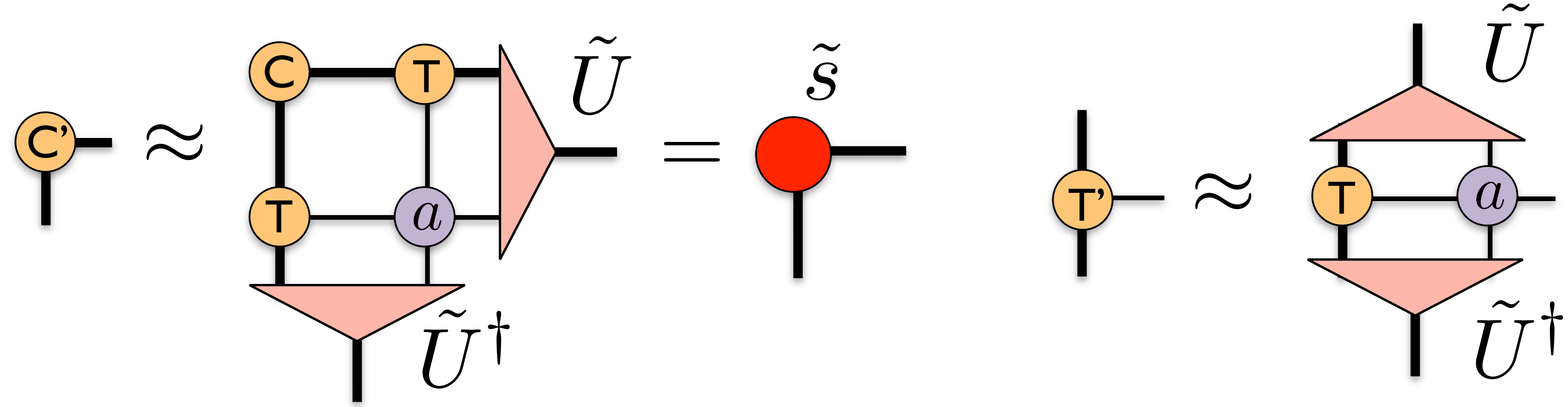


How can we best truncate from

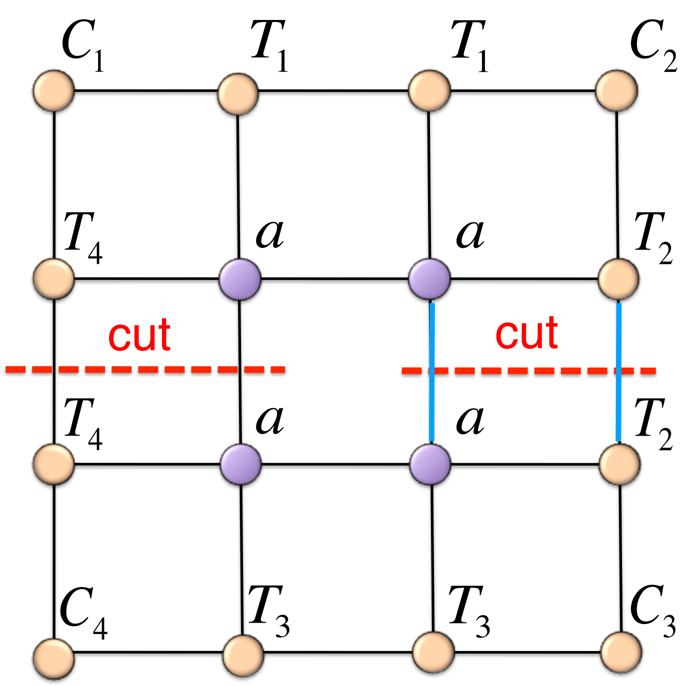
$$\chi D^2 \to \chi$$

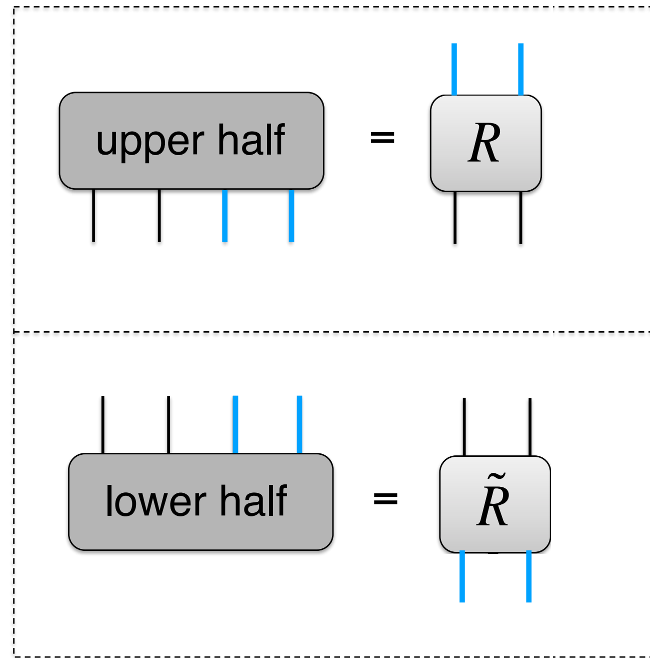


Renormalized tensors: keep only χ states with largest weight

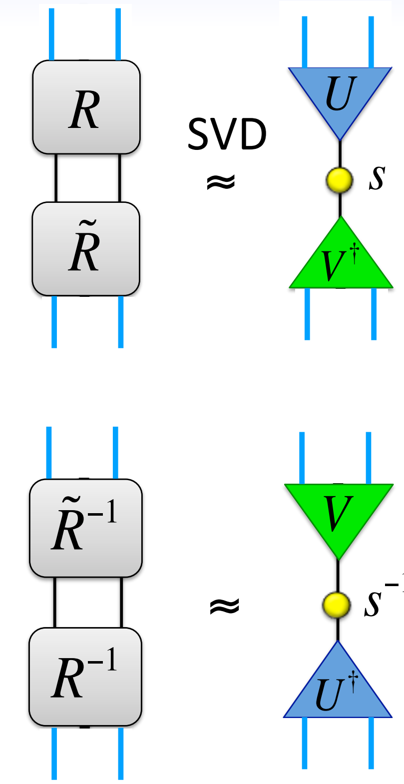


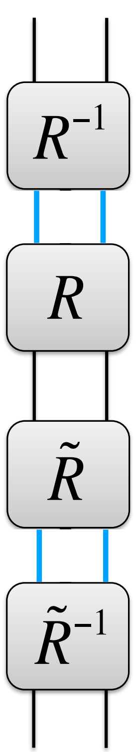
General case: Renormalization step (left move)



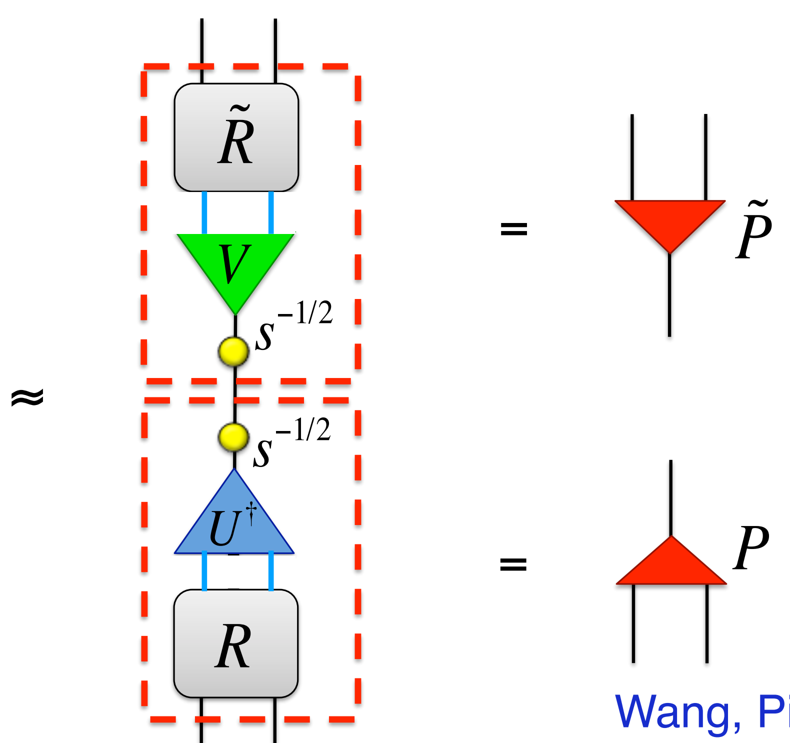


alternatively: only use upper left and lower left corners



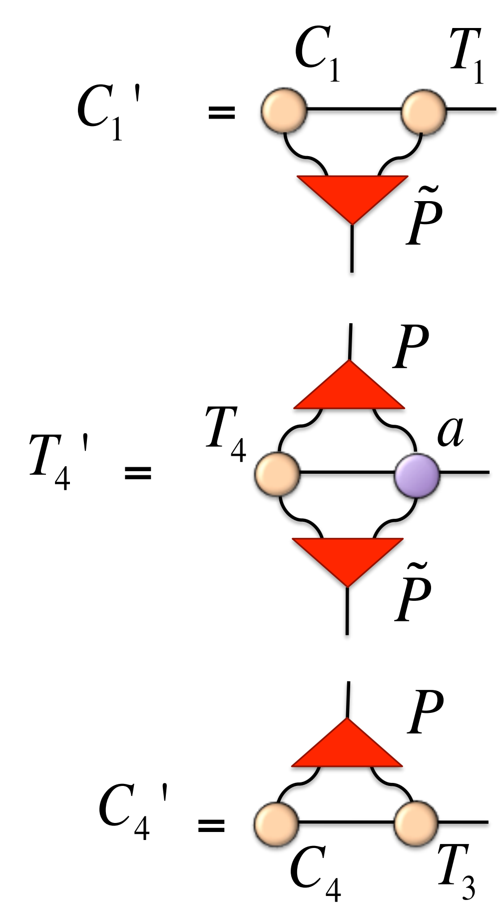


identity



approx. identity

projectors onto relevant subspace



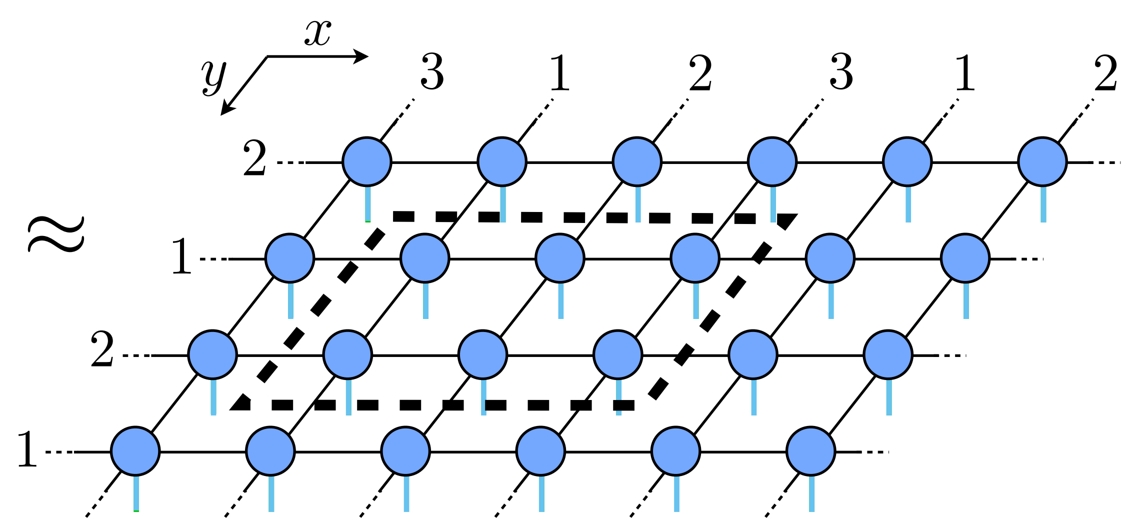
Wang, Pižorn & Verstraete, PRB 83 (2011)
Huang, Chen & Kao, PRB 86 (2012)
PC, Rice, Troyer, PRL 113 (2014)
T. Okubo, private comm.

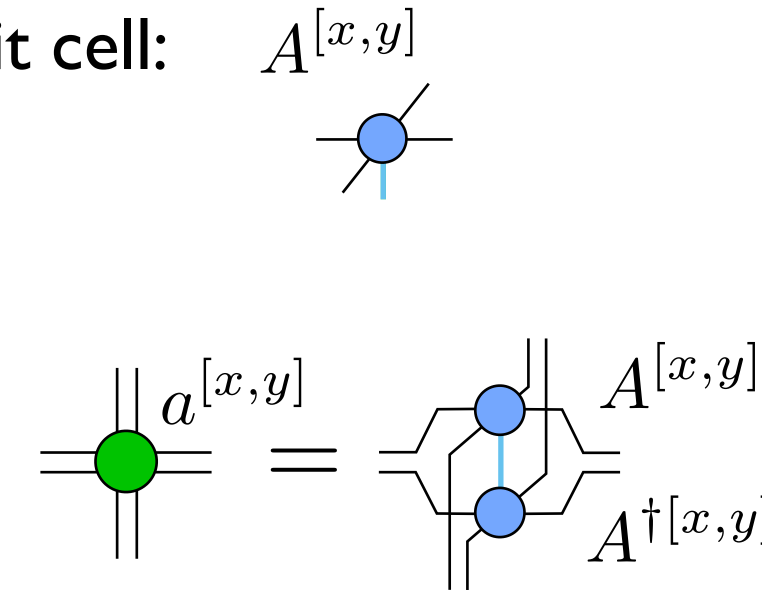
CTM with larger unit cells

PC, White, Vidal, Troyer, PRB **84** (2011)

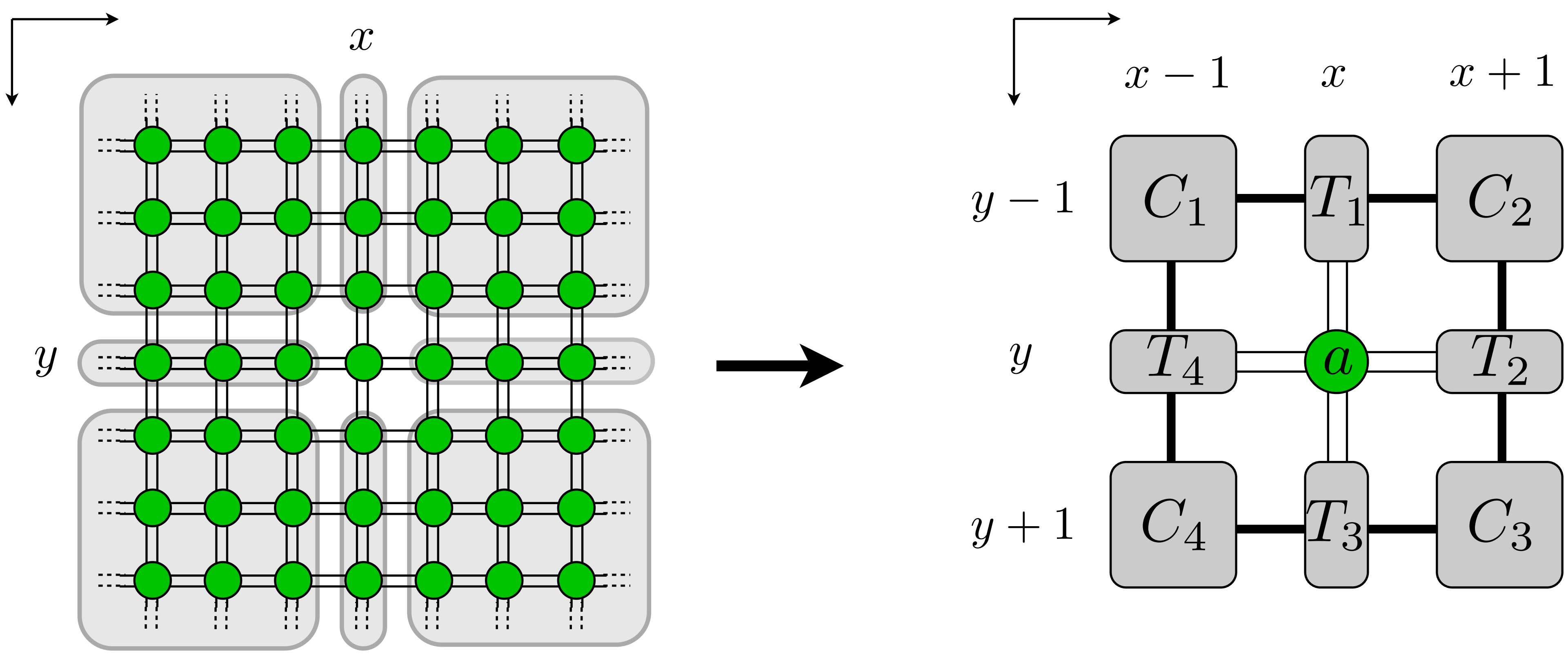
★ Each tensor has coordinates with respect to the unit cell:

$$|\Psi\rangle \approx$$



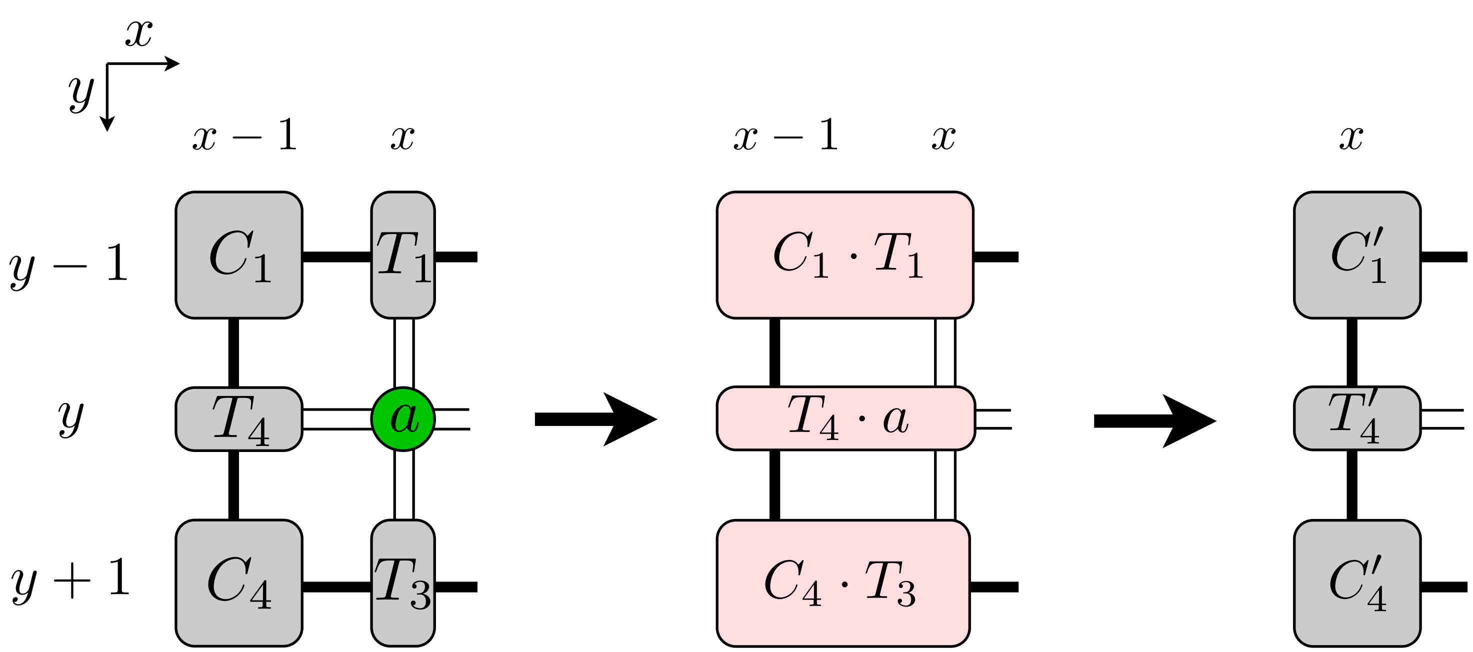


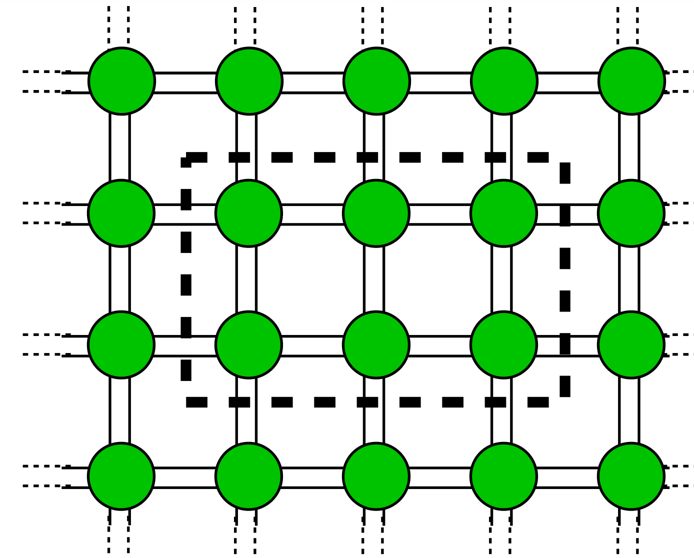
★ Keep a copy of every environment tensors C_1, ... C_4, T_1, ..., T_4 for each coordinate



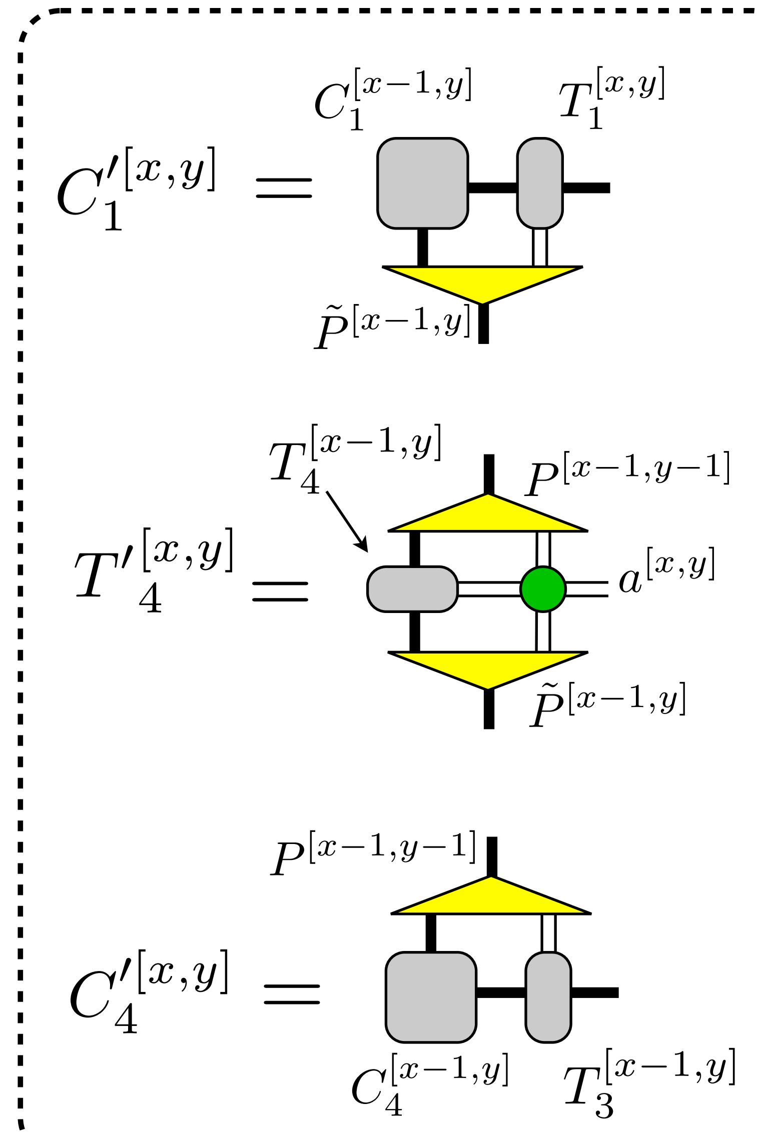
CTM with larger unit cells

Left move for $L_x \times L_y$ cell: *do for all x and y!*





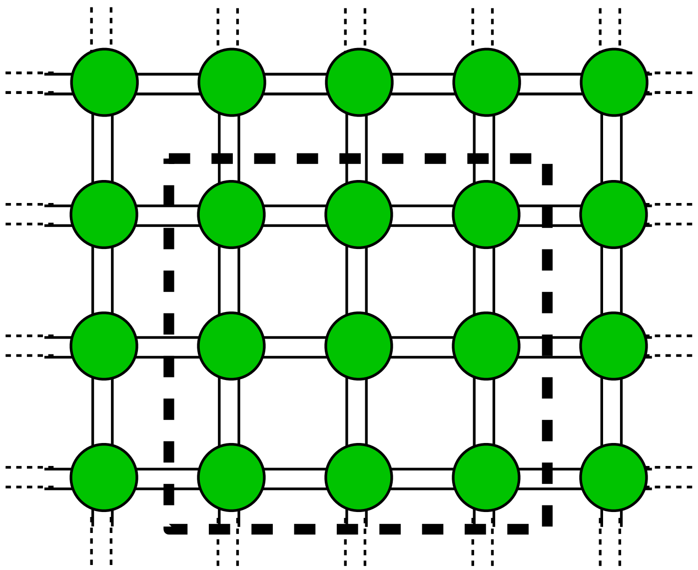
- Do for all $x \in [1, L_x]$
  - Do for all $y \in [1, L_y]$
    - Compute projectors $P^{[x-1,y]}$, $\tilde{P}^{[x-1,y]}$
  - Do for all $y \in [1, L_y]$
    - Compute updated environment tensors: $C_1'^{[x,y]}$, $C_4'^{[x,y]}$, $T_4'^{[x,y]}$



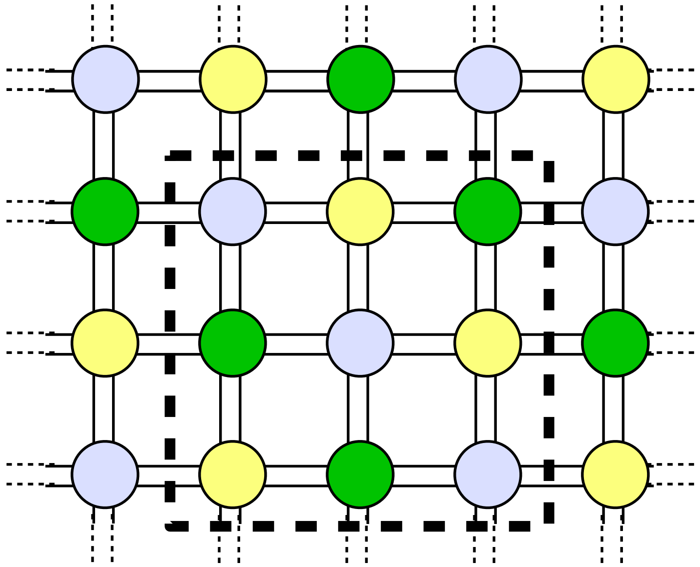
CTM with larger unit cells

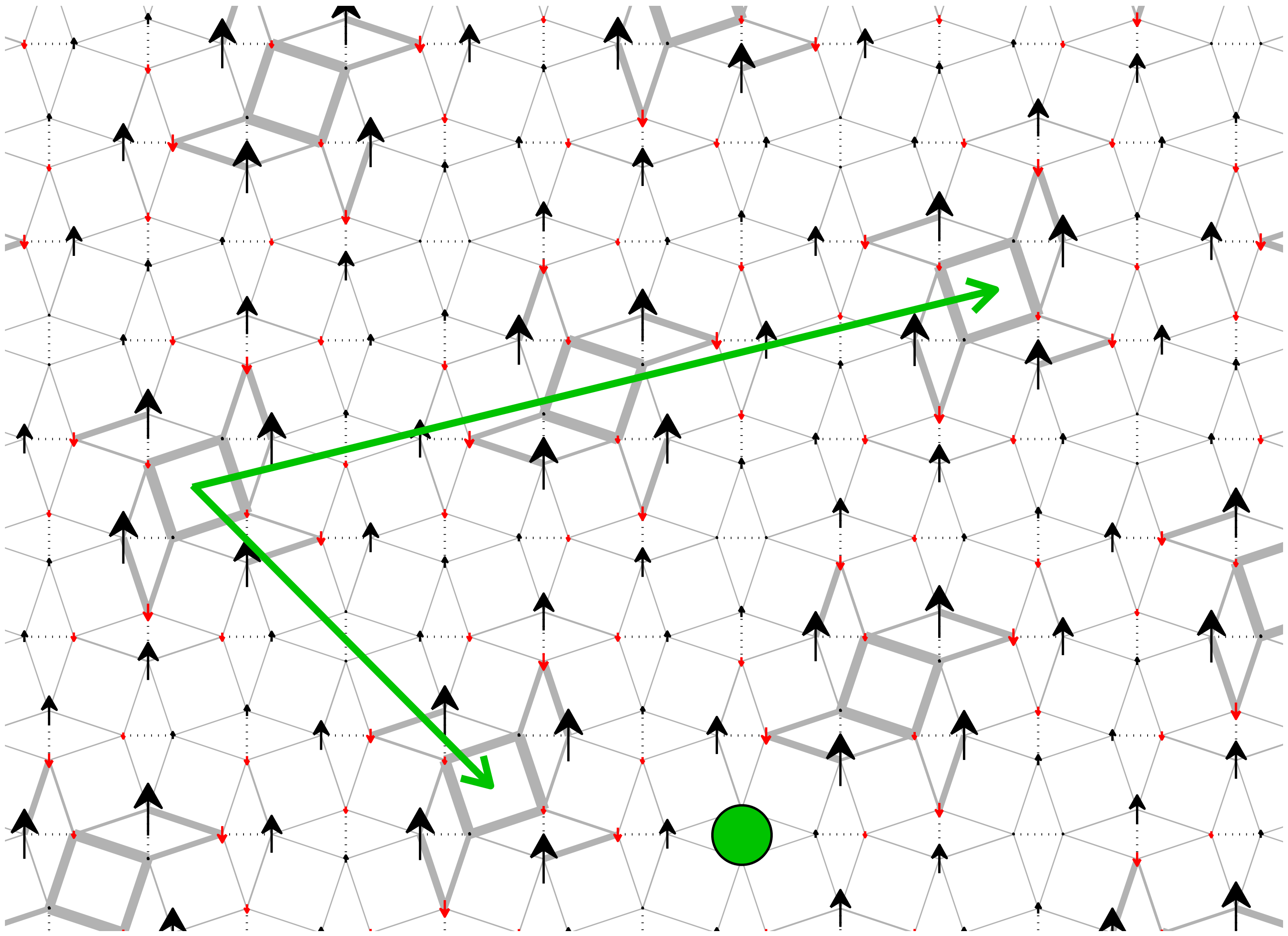
Other shapes than rectangular cell possible:

All 9 tensors different:



Only 3 different tensors:





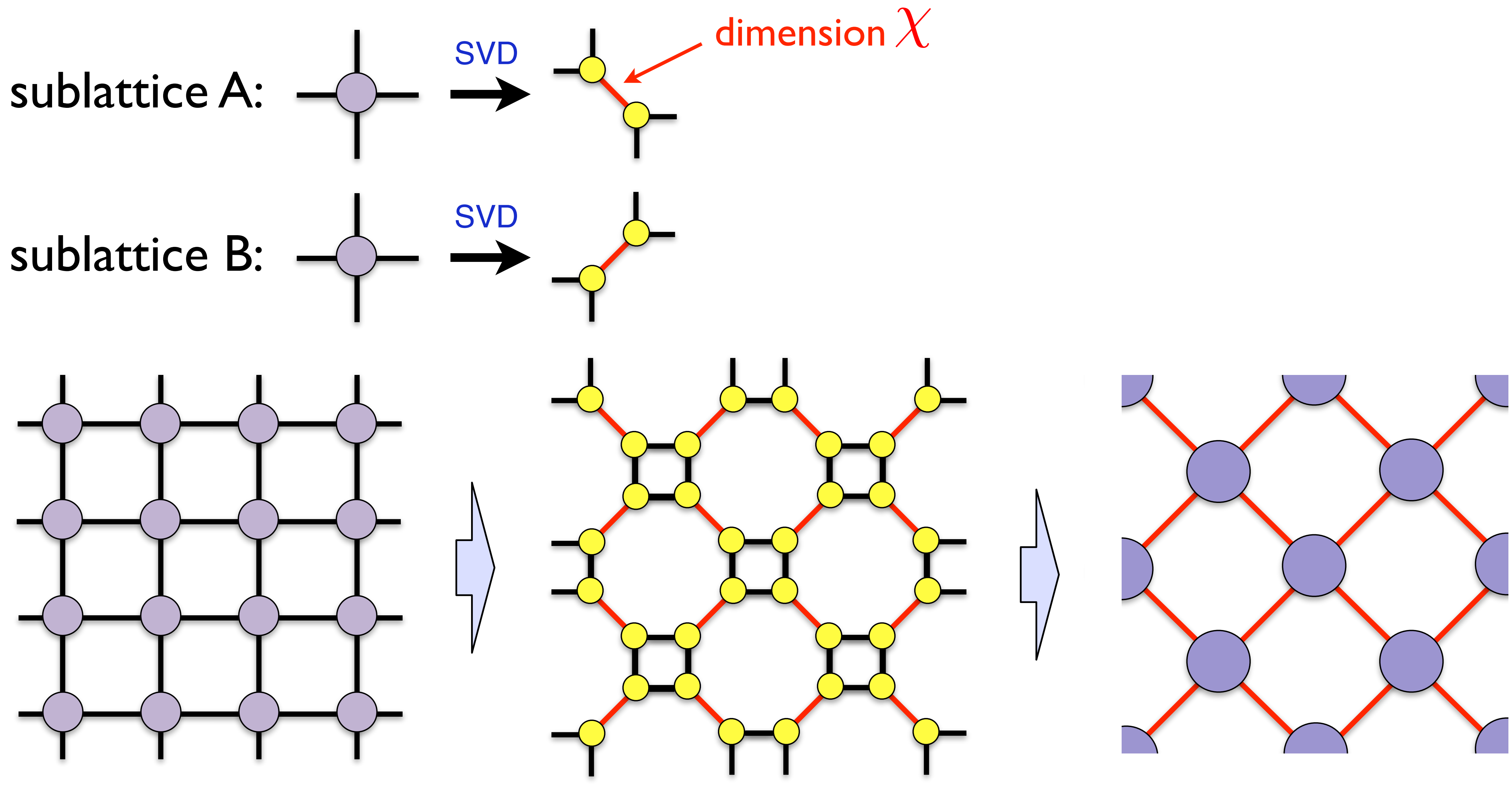
Unit cell with 30 tensors (60 sites)

(example: Shastry-Sutherland model)

Contracting the PEPS/iPEPS using TRG

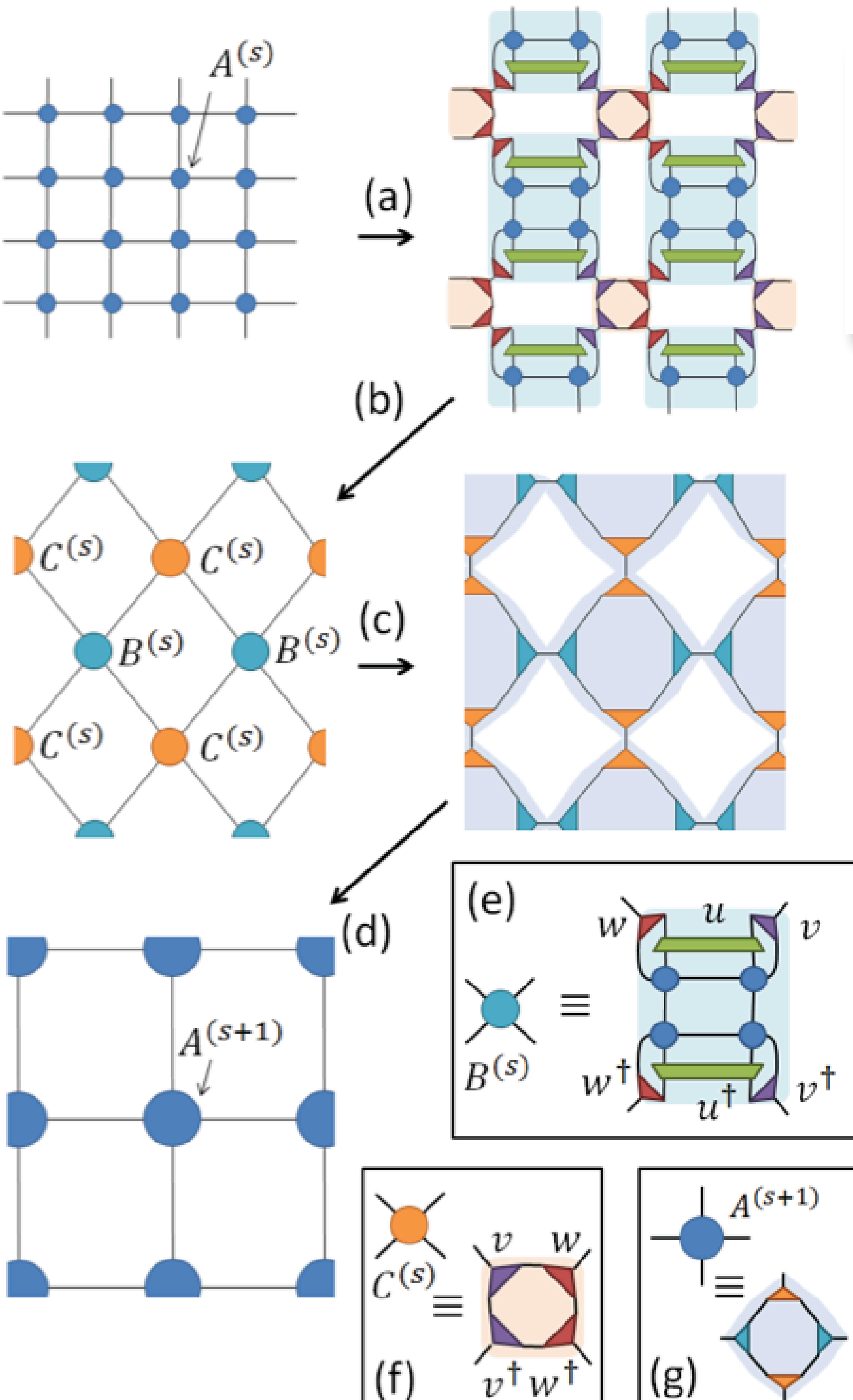
Tensor **R**enormalization **G**roup

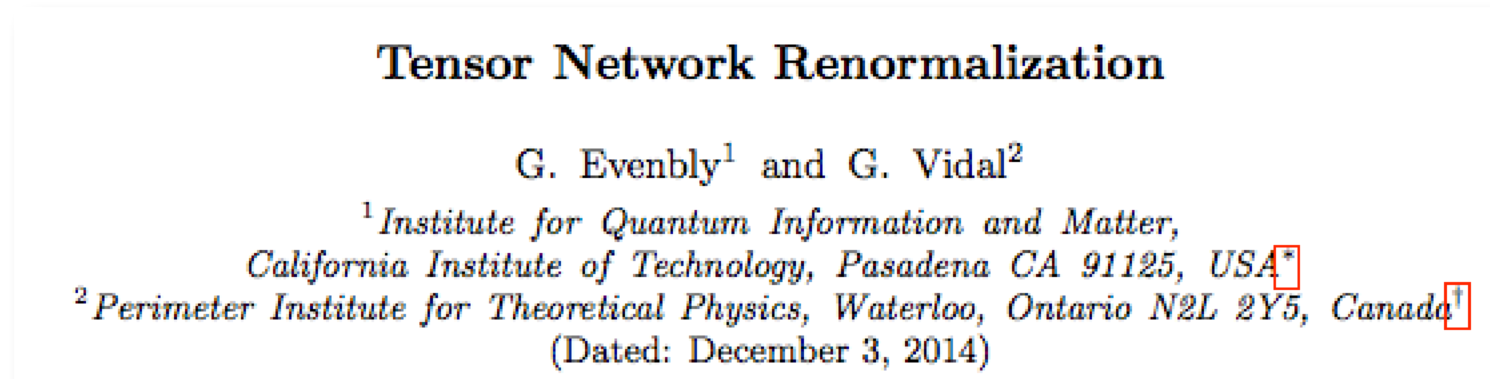
Gu, Levin, Wen, B78, (2008)
Levin, Nave, PRL99 (2007)
Xie et al. PRL 103, (2009)



- ★ Contract PEPS with periodic boundary conditions
- ★ Finite or infinite systems
- ★ Related schemes: SRG, HOTRG, HOSRG, ...

More advanced: Tensor network renormalization




Tensor Network Renormalization

G. Evenbly[1] and G. Vidal[2]

[1] *Institute for Quantum Information and Matter, California Institute of Technology, Pasadena CA 91125, USA*[*]

[2] *Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada*[†]

(Dated: December 3, 2014)

Evenbly & Vidal, PRL 115 (2015)

- ★ Additional ingredient: **Disentanglers**
- ★ Remove short-range entanglement at each coarse-graining step (key idea of the **MERA)**
- ★ Faster convergence with chi
- ★ Especially important for **critical** systems
- ★ Another variant: Loop-TNR:
  Yang, Gu & Wen, PRL 118 (2017)

Contracting the PEPS

★ Exact contraction of an PEPS is exponentially hard!

→ *use controlled approximate contraction scheme*

MPS-MPO-based / VUMPS

Murg, Verstraete, Cirac, PRA75 '07
Jordan, et al. PRL79 (2008)
Haegeman & Verstraete (2017)
…

Corner transfer matrix method

Nishino, Okunishi, JPSJ65 (1996)
Orus, Vidal, PRB 80 (2009)
Fishman et al, PRB 98 (2018)
…

TRG

Tensor Renormalization Group
(variants: HOTRG, SRG, HOSRG)

Levin, Nave, PRL99 (2007)
Xie et al. PRL 103 (2009)
Xie et al. PRB 86 (2012), …

↓

TNR

Tensor Network Renormalization
Evenbly & Vidal, PRL 115 (2015)

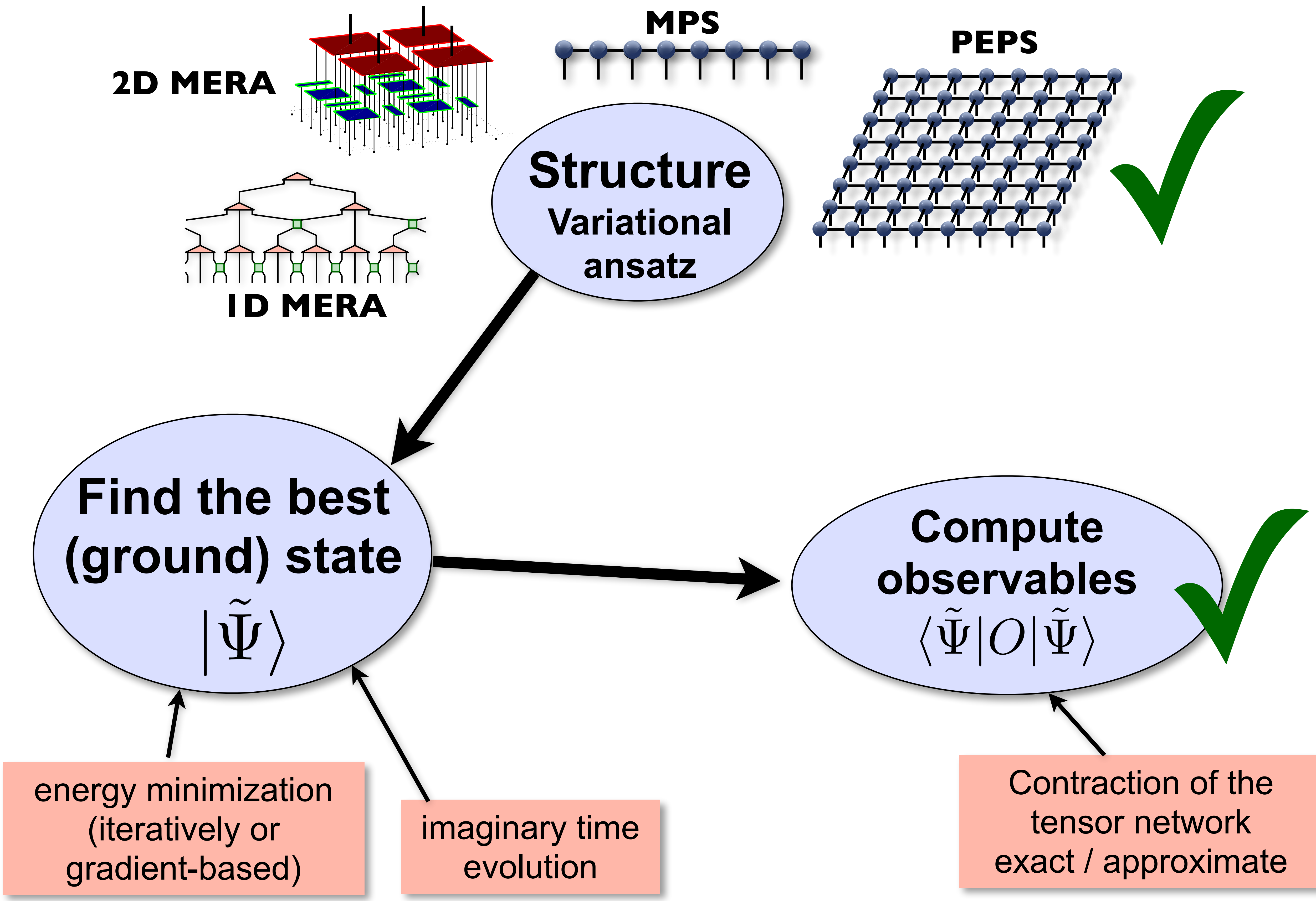
Loop-TNR:
Yang, Gu & Wen, PRL 118 (2017)

★ Accuracy of the approximate contraction is controlled by "boundary dimension" χ

★ Convergence in χ needs to be carefully checked

★ Overall cost: $\mathcal{O}(D^{10\dots 14})$ with $\chi \sim D^2$

Overview: Tensor network algorithms (ground state)



Optimization

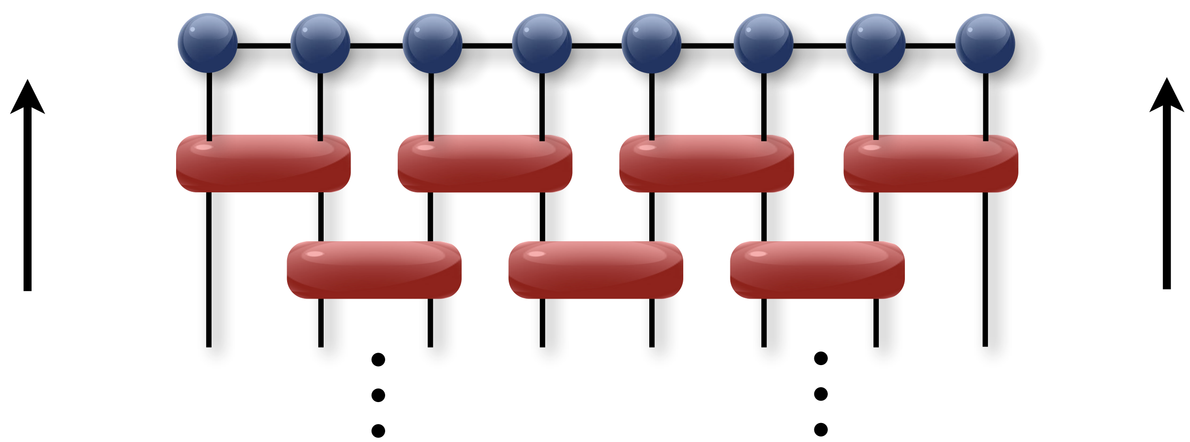
Optimization via imaginary time evolution

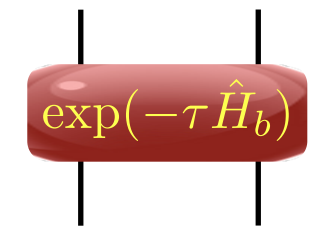
- Idea: $\exp(-\beta\hat{H})|\Psi_i\rangle \xrightarrow{\beta \to \infty} |\Psi_{GS}\rangle$

Trotter-Suzuki decomposition:

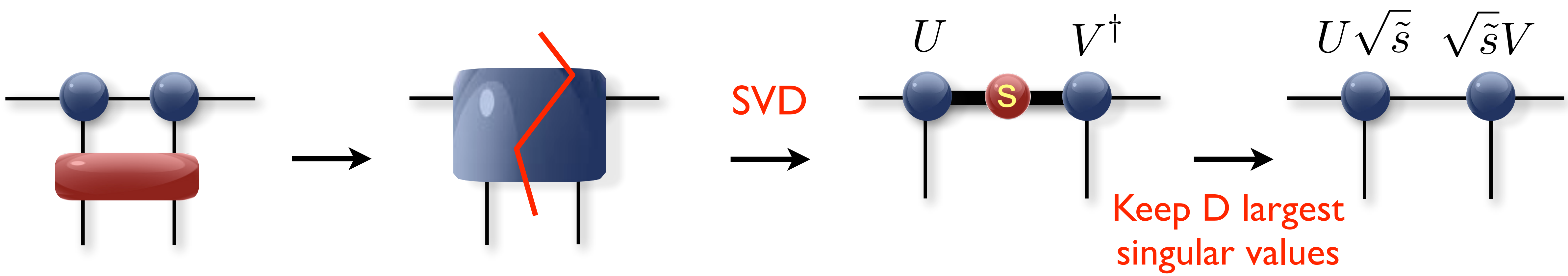
$$\exp(-\beta\hat{H}) = \exp(-\beta\sum_b \hat{H}_b) = \left(\exp(-\tau\sum_b \hat{H}_b)\right)^n \approx \left(\prod_b \exp(-\tau\hat{H}_b)\right)^n, \quad \tau = \beta/n$$

- 1D:





- At each step: apply a two-site operator to a bond and truncate bond back to *D*



Time **E**volving **B**lock **D**ecimation (TEBD) algorithm

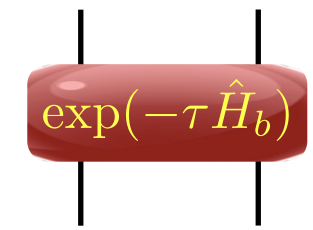
Note: MPS needs to be in canonical form

Optimization via imaginary time evolution

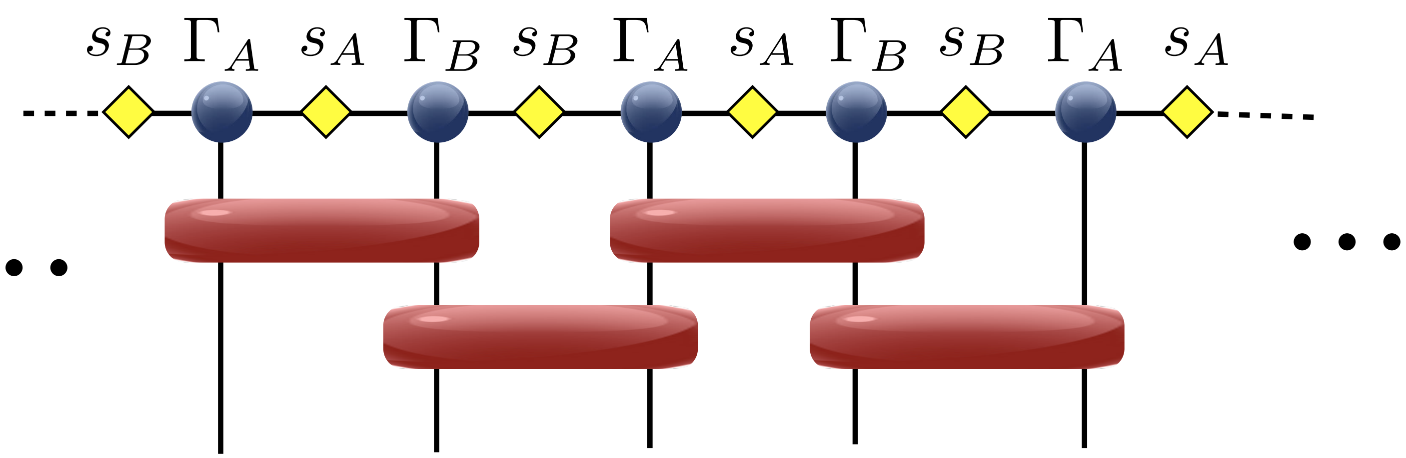
- Idea: $\exp(-\beta\hat{H})|\Psi_i\rangle \xrightarrow{\beta\to\infty} |\Psi_{GS}\rangle$

Trotter-Suzuki decomposition:

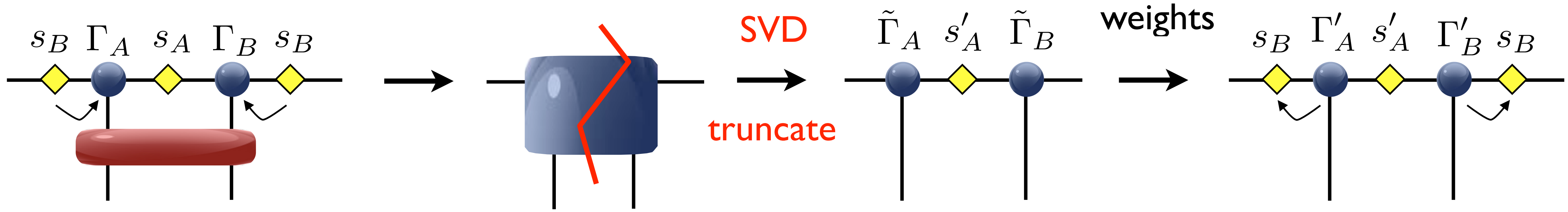
$$\exp(-\beta\hat{H}) = \exp(-\beta\sum_b \hat{H}_b) = \left(\exp(-\tau\sum_b \hat{H}_b)\right)^n \approx \left(\prod_b \exp(-\tau\hat{H}_b)\right)^n, \quad \tau = \beta/n$$



- 1D:

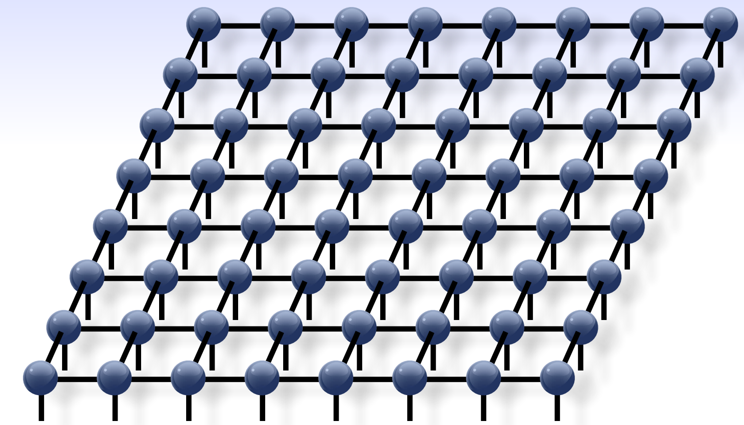


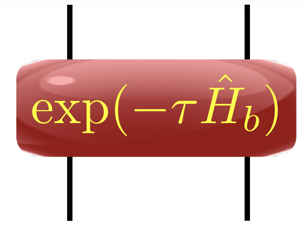
- At each step: apply a two-site operator to a bond and truncate bond back to *D*



infinite **T**ime **E**volving **B**lock **D**ecimation (iTEBD)

Optimization via imaginary time evolution



- **2D: same idea:** apply 

to a bond and truncate bond back to D

- **However**, SVD update is not optimal (because of loops in PEPS)!

simple update (SU)

Jiang et al, PRL 101 (2008)

★ "local" update like in TEBD

★ Cheap, but not optimal (e.g. overestimates magnetization in S=1/2 Heisenberg model)

full update (FU)

Jordan et al, PRL 101 (2008)

★ Take the full wave function into account for truncation

★ optimal, but more expensive

★ Fast-full update [Phien et al, PRB 92 (2015)]

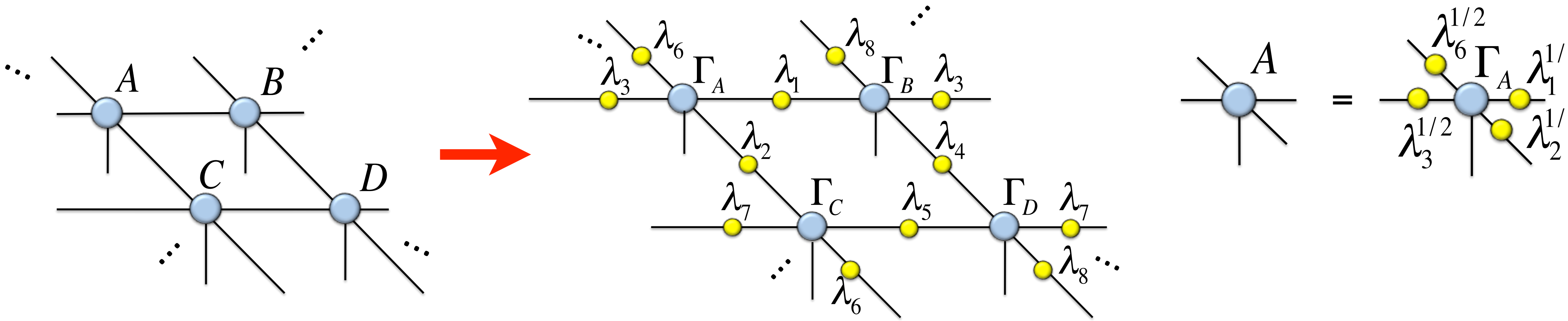
Cluster update

Wang, Verstraete, arXiv:1110.4362 (2011)

Optimization: simple update

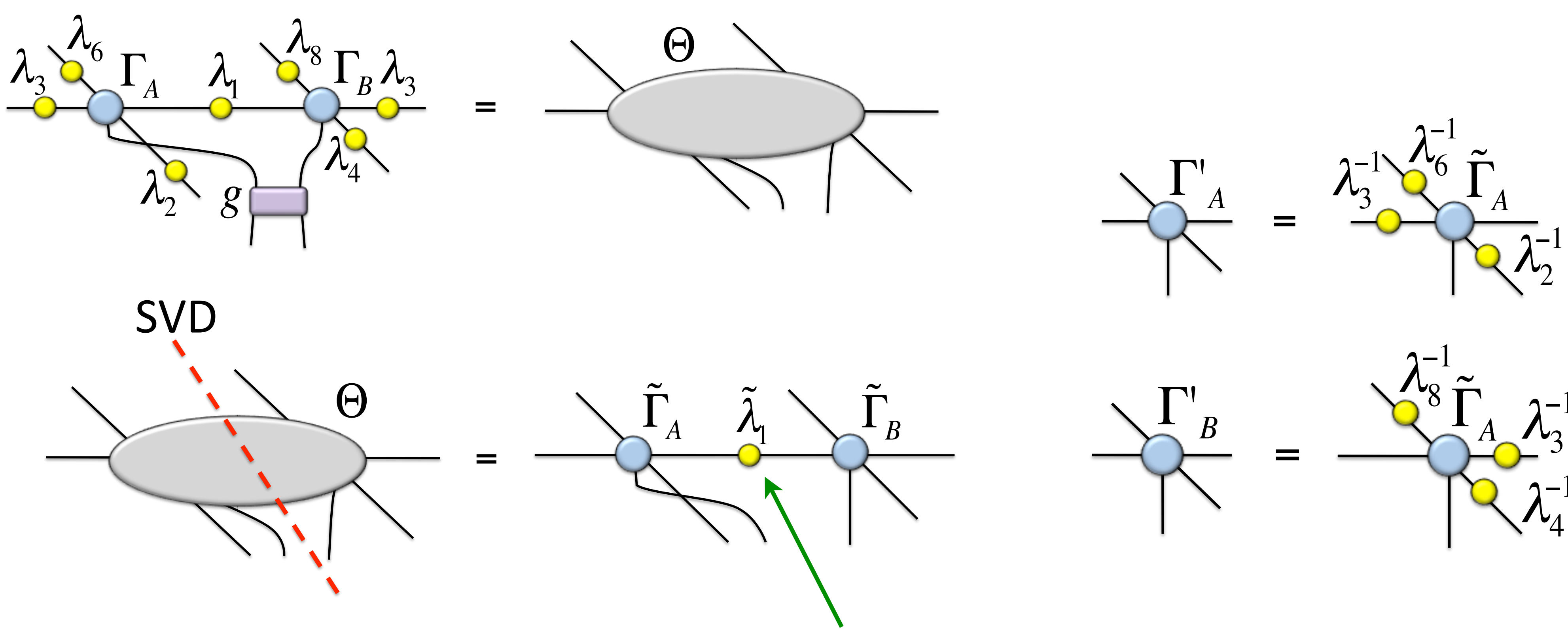
Jiang, et al., PRL 101, 090603 (2008)

- iPEPS with "weights" on the bonds (takes environment effectively into account)



- Update works like in 1D with iTEBD (infinite time-evolving block decimation)

G. Vidal, PRL 91, 147902 (2003)

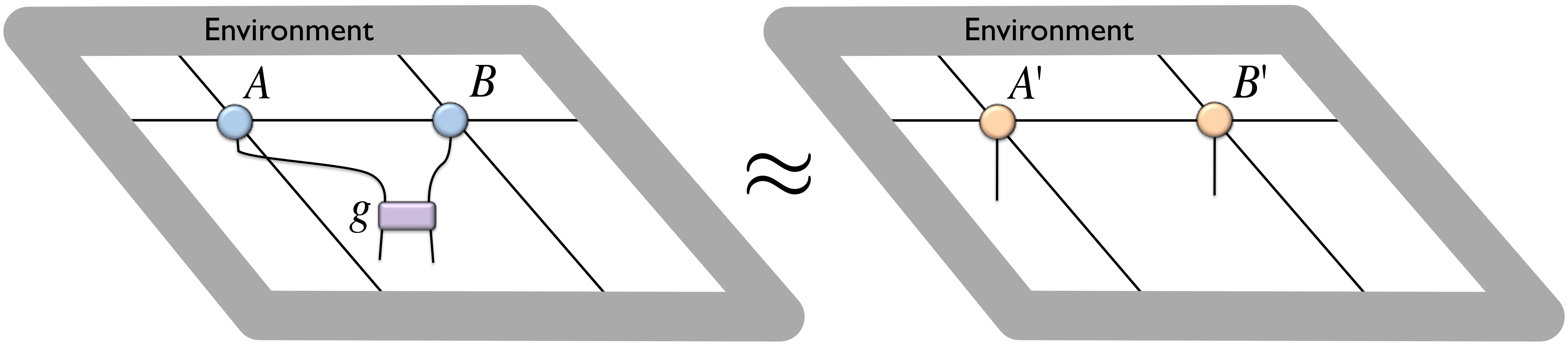


keep only D largest singular values

Optimization: full update

Jordan, Orus, Vidal, Verstraete, Cirac, PRL (2008)
Corboz, Orus, Bauer, Vidal, PRB 81, 165104 (2010)

- Approximate old PEPS + gate with a new PEPS with bond dimension D



$$|\tilde{\Psi}\rangle = g|\Psi\rangle \qquad \approx \qquad |\Psi'\rangle$$

- Minimize $\quad || \, |\tilde{\Psi}\rangle - |\Psi'\rangle \, ||^2 = \langle\tilde{\Psi}|\tilde{\Psi}\rangle + \langle\Psi'|\Psi'\rangle - \langle\tilde{\Psi}|\Psi'\rangle - \langle\Psi'|\tilde{\Psi}\rangle$
- Iteratively / CG / Newton / ...
- The full wave function is taken into account for the truncation!
- At each step the environment has to be computed! expensive... but optimal!

Optimization: simple vs full update

simple update

★ "local" update like in TEBD

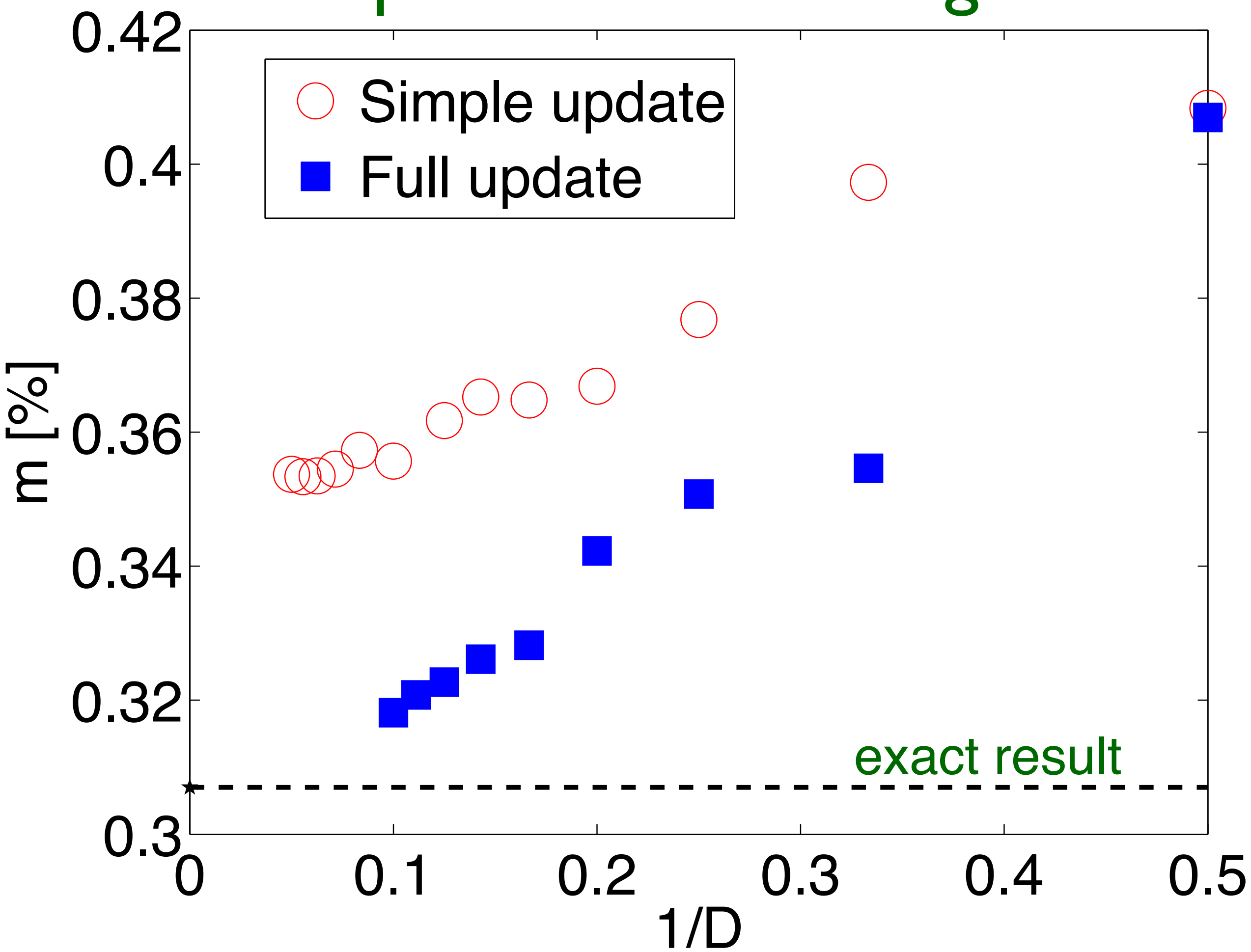
★ Cheap, but not optimal (e.g. overestimates magnetization in S=1/2 Heisenberg model)

full update

★ Take the full wave function into account for truncation

★ optimal, but computationally more expensive

Example: 2D Heisenberg model



- Combine the two: Use simple update to get an initial state for the full update

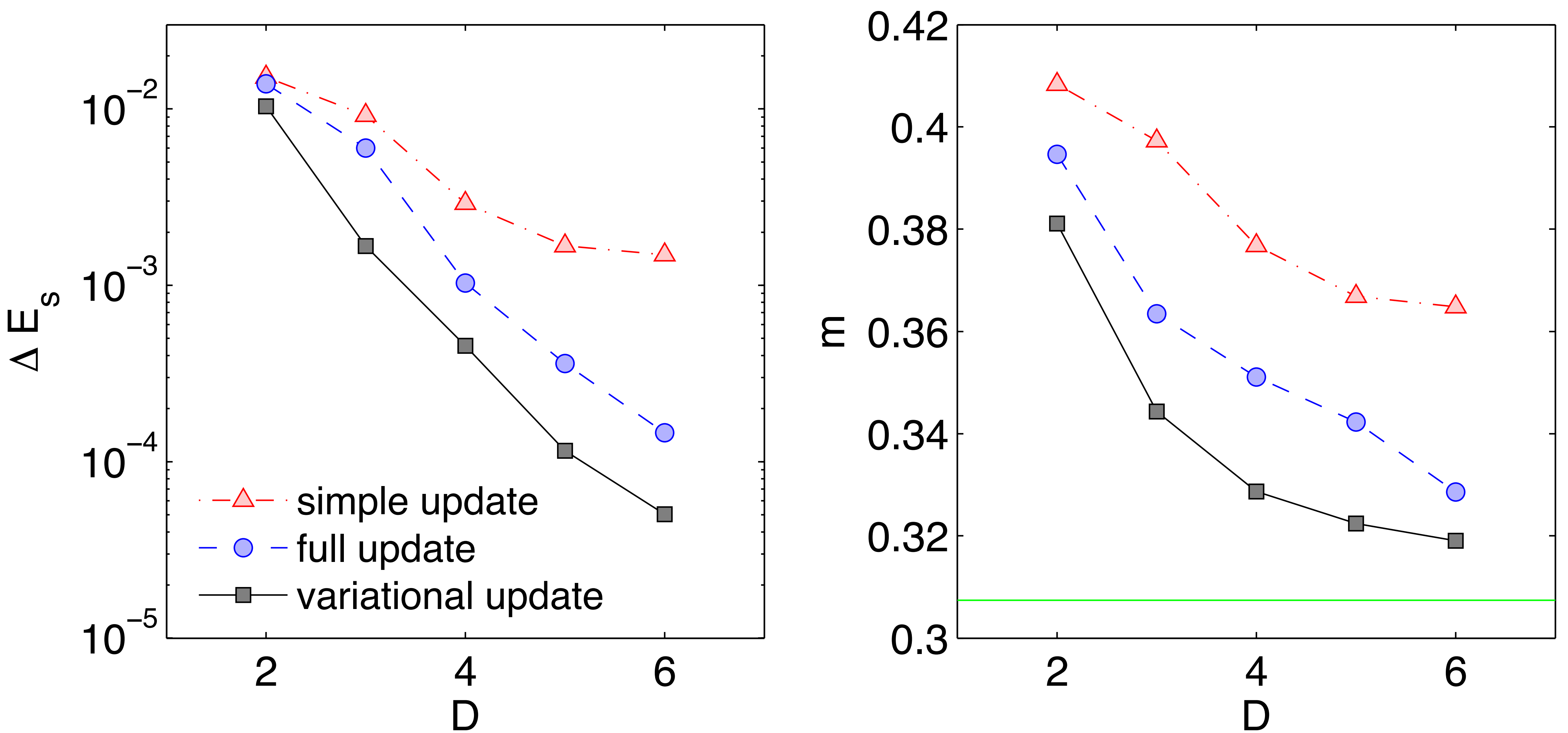
- Don't compute environment from scratch but recycle previous one
  → **fast full update** Phien, Bengua, Tuan, PC, Orus, PRB 92 (2015)

Summary: optimization in iPEPS

- Imaginary time evolution
  - ✦ Simple update: cheap and simple, but not accurate
    Jiang et al, PRL 101 (2008)
  - ✦ Cluster update: improved accuracy
    Wang et al, arXiv:1110.4362
  - ✦ Full update: high accuracy, more expensive
    Jordan et al, PRL 101 (2008)
  - ✦ Fast-full update: high accuracy, cheaper than FU
    Phien et al, PRB 92 (2015)

- Energy minimization
  - ✦ DMRG-like sweeping: higher accuracy, similar cost as FFU
    PC, PRB 94 (2016)
  - ✦ Channel environments: higher accuracy, similar cost as FFU
    Vanderstraeten, Haegeman, PC, and Verstraete, PRB 94 (2016)
  - ✦ Automatic differentiation: higher accuracy, similar cost and **simpler!**
    Liao, Liu, Wang, Xiang, PRX (2019)
  - ✦ … and more …

Energy minimization: 2D Heisenberg model



- Variational energy minimization algorithm is more accurate than imaginary time evolution
- Variational update (D=6): -0.66941
- Extrapolated QMC result: -0.66944 [Sandvik&Evertz 2010]

Summary: optimization in iPEPS

- Imaginary time evolution
  - ✦ Simple update: cheap and simple, but not accurate
    Jiang et al, PRL 101 (2008)
  - ✦ Cluster update: improved accuracy
    Wang et al, arXiv:1110.4362
  - ✦ Full update: high accuracy, more expensive
    Jordan et al, PRL 101 (2008)
  - ✦ Fast-full update: high accuracy, cheaper than FU
    Phien et al, PRB 92 (2015)

- Energy minimization
  - ✦ DMRG-like sweeping: higher accuracy, similar cost as FFU
    PC, PRB 94 (2016)
  - ✦ Channel environments: higher accuracy, similar cost as FFU
    Vanderstraeten, Haegeman, PC, and Verstraete, PRB 94 (2016)
  - ✦ Automatic differentiation: higher accuracy, similar cost and **simpler!**
    Liao, Liu, Wang, Xiang, PRX (2019)
  - ✦ … and more …

Automatic differentiation

PHYSICAL REVIEW X **9**, 031041 (2019)

Differentiable Programming Tensor Networks

Hai-Jun Liao,[1,2] Jin-Guo Liu,[1] Lei Wang,[1,2,3,*] and Tao Xiang[1,4,5,†]

[1]*Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*
[2]*CAS Center for Excellence in Topological Quantum Computation, University of Chinese Academy of Sciences, Beijing 100190, China*
[3]*Songshan Lake Materials Laboratory, Dongguan, Guangdong 523808, China*
[4]*University of Chinese Academy of Sciences, Beijing 100049, China*
[5]*Collaborative Innovation Center of Quantum Matter, Beijing 100190, China*

(Received 2 April 2019; published 5 September 2019)

Differentiable programming is a fresh programming paradigm which composes parameterized algorithmic components and optimizes them using gradient search. The concept emerges from deep learning but is ... rogramming
tensor netwo ... orithm as a
computation ... l efficiently
using automa ... or networks
contraction a ... nd efficient
backpropaga ... heat of the
Ising model ... the tensor
renormalizati ... of infinite
projected ent ... ain state-of-
the-art variati ... ing removes
laborious human efforts in deriving and implementing analytical gradients for tensor network programs, which opens the door to more innovations in tensor network algorithms and applications.

Computing gradients in an automatized fashion!
Simplifies codes substantially!
Implemented in machine learning frameworks (TensorFlow, PyTorch, …)

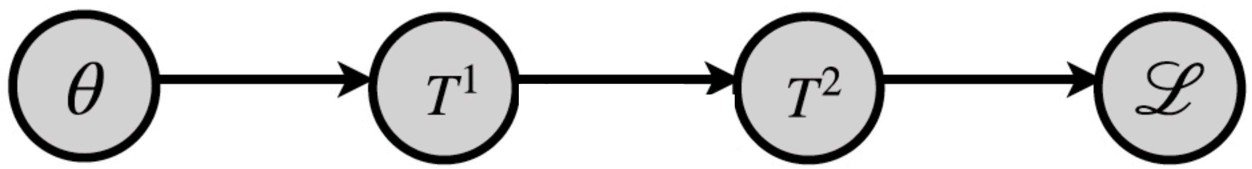
DOI: 10.1103/PhysRevX.9.031041

Subject Areas: Computational Physics, Condensed Matter Physics

Automatic differentiation

Liao, Liu, Wang, Xiang, PRX (2019)

computation graph:



input (tensors)

output (e.g. energy)

Compute the gradient via chain rule:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial T^n} \frac{\partial T^n}{\partial T^{n-1}} \cdots \frac{\partial T^2}{\partial T^1} \frac{\partial T^1}{\partial \theta}$$

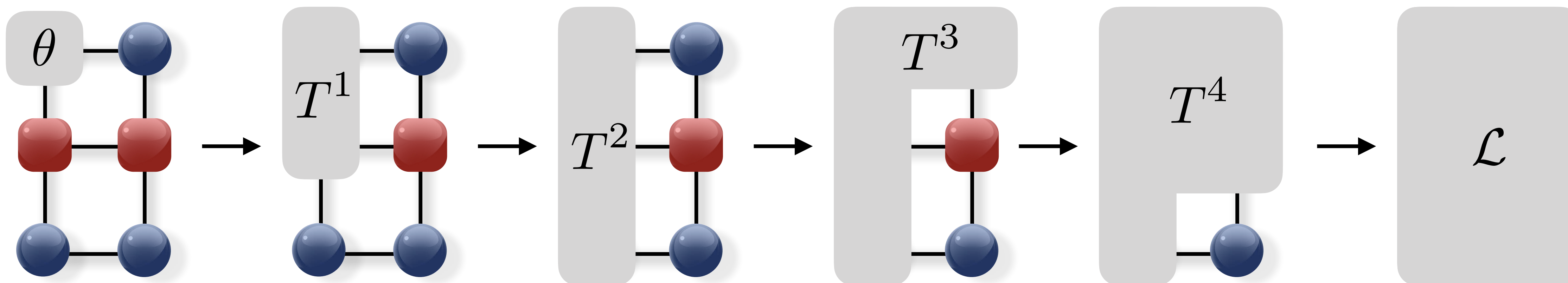
from left to right
(back propagation algorithm)

Define forward and backward function of each elementary operation (primitives), e.g. addition, multiplication, math functions, matrix-matrix multiplications, eigenvalue decompositions, etc.

→ Gradient can be computed in an automatized fashion

Simple example

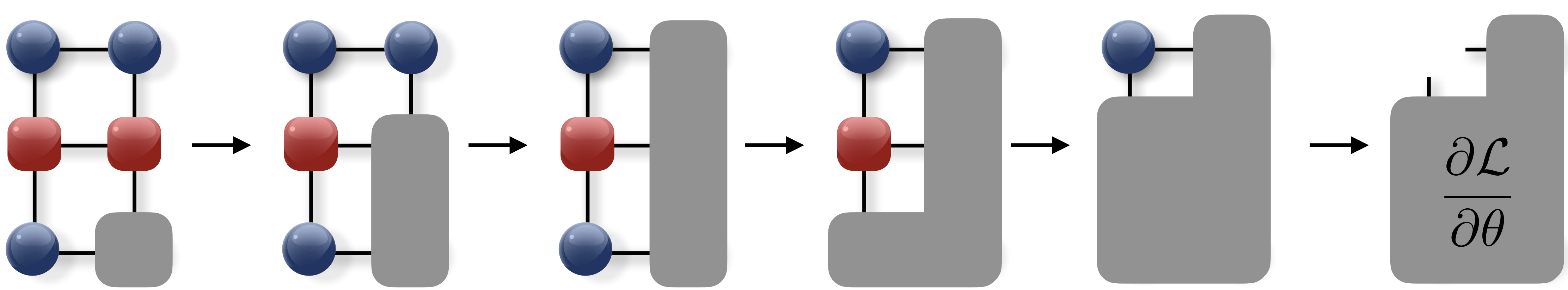
(tensor)  (energy)



Compute the gradient via chain rule:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial T^4}\frac{\partial T^4}{\partial T^3}\frac{\partial T^3}{\partial T^2}\frac{\partial T^2}{\partial T^1}\frac{\partial T^1}{\partial \theta}$$

from left to right
(back propagation algorithm)



Summary: optimization in iPEPS

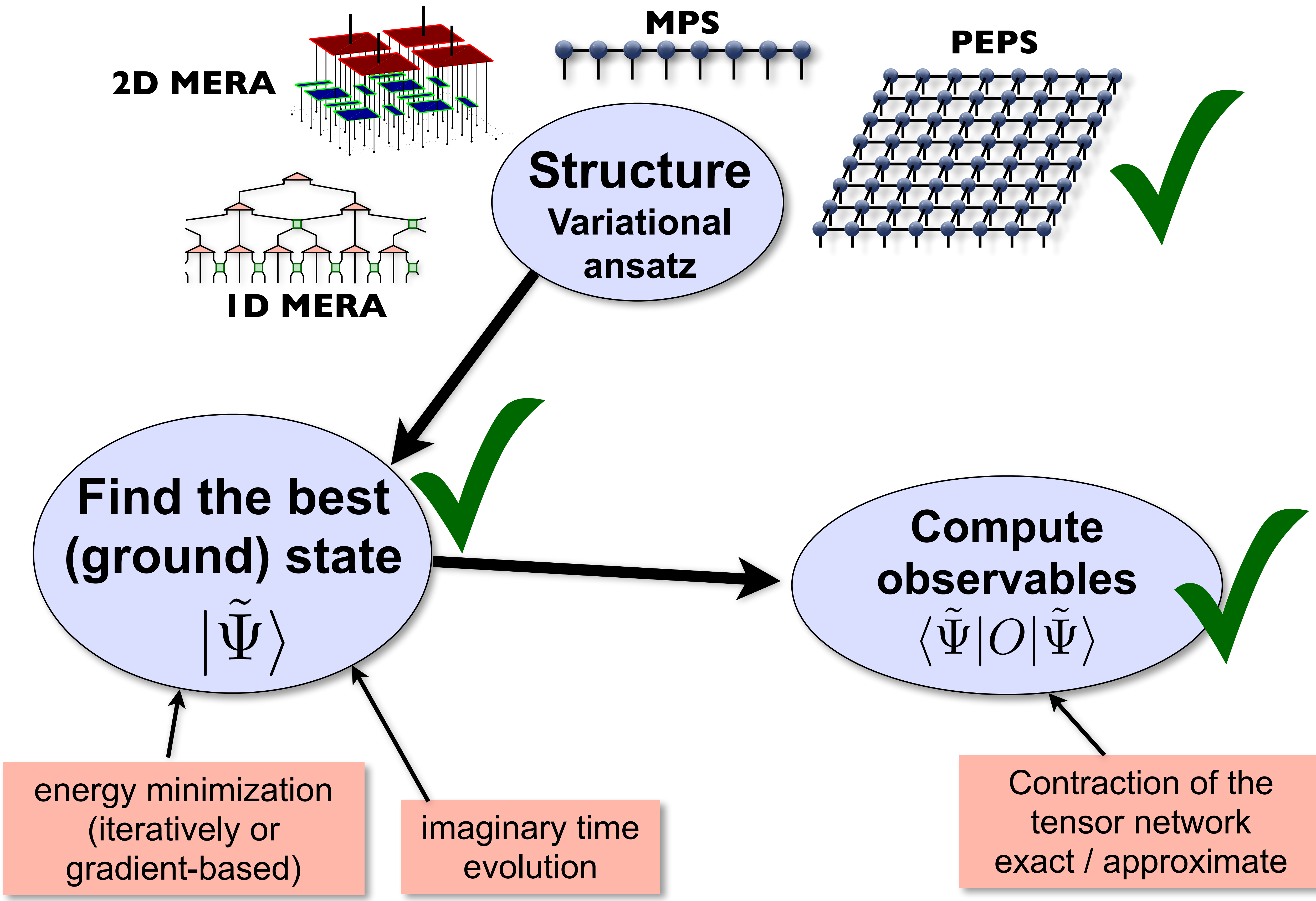
- Imaginary time evolution
  - ✦ Simple update: cheap and simple, but not accurate
    Jiang et al, PRL 101 (2008)
  - ✦ Cluster update: improved accuracy
    Wang et al, arXiv:1110.4362
  - ✦ Full update: high accuracy, more expensive
    Jordan et al, PRL 101 (2008)
  - ✦ Fast-full update: high accuracy, cheaper than FU
    Phien et al, PRB 92 (2015)

+ COMBINATIONS!

- Energy minimization
  - ✦ DMRG-like sweeping: higher accuracy, similar cost as FFU
    PC, PRB 94 (2016)
  - ✦ Channel environments: higher accuracy, similar cost as FFU
    Vanderstraeten, Haegeman, PC, and Verstraete, PRB 94 (2016)
  - ✦ Automatic differentiation: higher accuracy, similar cost and **simpler!**
    Liao, Liu, Wang, Xiang, PRX (2019)
  - ✦ … and more …

Still room for improvement!

Summary: Tensor network algorithms (ground state)



iPEPS ground state simulations in 2D

- Many applications to challenging problems, including frustrated spin, SU(N), bosonic systems, *t-J* / Hubbard models, and more, see e.g.:

- Dusuel, Kamfor, Orús, Schmidt, Vidal, PRL 106 (2011)
- Corboz, Läuchli, Penc, Troyer, Mila, PRL 107 (2011)
- Zhao, Xu, Chen, Wei, et al., PRB 85 (2012)
- Corboz, Lajkó, Läuchli, Penc, Mila, PRX 2 (2012)
- Corboz, Mila, PRB 87 (2013); PRL 112 (2014)
- Gu, Jiang, Sheng, Yao, Balents, Wen, PRB 88 (2013)
- Osorio Iregui, Corboz, Troyer, PRB 90 (2014)
- Corboz, Rice, Troyer, PRL 113 (2014)
- Picot, Poilblanc, PRB 91 (2015)
- Picot, Ziegler, Orús, Poilblanc, PRB 93 (2016)
- Nataf, Lajkó, et al., PRB 93 (2016)
- Liao, Xie, Chen, Liu, Xie, et al., PRL 118 (2017)
- Zheng, et al., Science 358, 1155 (2017)
- Niesen, Corboz, PRB 95 (2017); SciPost Physics 3 (2017); Rev. B 97 (2018)
- Haghshenas, Lan, Gong, Sheng, PRB 97 (2018)
- Chen, Vanderstraeten, et al, PRB 98 (2018)
- Jahromi, Orús, PRB 98 (2018)
- Lee, Kawashima, PRB 97 (2018)
- Yamaguchi, Sasaki, Okubo, et al.,PRB 98 (2018)
- Haghshenas, Gong, Sheng, PRB 99, 174423 (2019)
- Chung, Corboz, PRB 100 (2019)
- Ponsioen, Chung, Corboz, PRB 100 (2019)
- Boos, Crone, Niesen, et al, PRB 100 (2019)

⋮ ⋮

- Shi, et al, Nature Communications 10 (2019)
- Kshetrimayum, Balz, Lake, Eisert, arXiv:1904.00028
- Lee, Kaneko, Okubo, Kawashima, PRL 123 (2019)
- Gauthé, Capponi, Mambrini, Poilblanc, PRB 101 (2020)
- Lee, Kaneko, Chern, Okubo, Yamaji, Kawashima, Kim, Nat Commun 11 (2020)
- Chen, Capponi, Wietek, Mambrini, Schuch, Poilblanc, PRL 125, (2020)
- Hasik, Poilblanc, Becca, SciPost Physics 10 (2021)
- Liu, Gong, Li, Poilblanc, Chen, Gu, SB 67 (2022)
- Shi, et al., Nat Commun 13 (2022)
- Liu, Hasik, Gong, Poilblanc, Chen, Gu, PRX 12 (2022)
- Peschke, Ponsioen, Corboz, PRB 106 (2022)
- Hasik, Van Damme, Poilblanc, Vanderstraeten, PRL 129 (2022)
- Ponsioen, Chung, Corboz, PRB 108 (2023)
- Weerda, Rizzi, arXiv:2309.12811 (2023)
- Xu, Capponi, Chen, et al.,PRB 108 (2023)
- Xi, Chen, Xie, Yu, PRB 107 (2023)
- Nomura et al., Nat Commun 14 (2023)
- Hasik, Corboz, PRL 133 (2024).
- Schmoll, Naumann, Eisert, Iqbal, arXiv:2407.07145

⋮ ⋮

Summary of today's lecture

- ✓ Main idea of a tensor network ansatz & area law of entanglement entropy
- ✓ iPEPS ansatz to represent 2D ground states in the thermodynamic limit
- ✓ Contraction of the 2D tensor network
  - ★ Accuracy systematically controlled by χ
  - ★ Corner transfer matrix (CTM) method, MPS-MPO based contraction, Tensor Renormalization Group, Tensor Network Renormalization
- ✓ Optimization of iPEPS using
  - ★ imaginary time evolution: simple vs full updated
  - ★ energy minimization: automatic differentiation

➡ Exercise: CTM method for the 2D classical Ising model

➡ slides / codes: https://tinyurl.com/4rdyh7ex

Thank you for your attention!