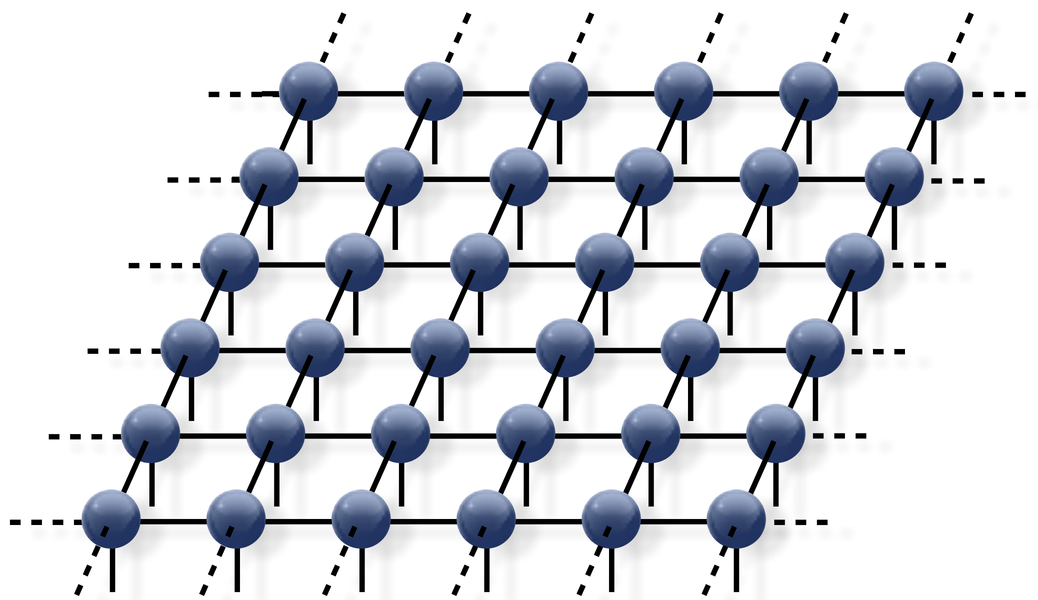
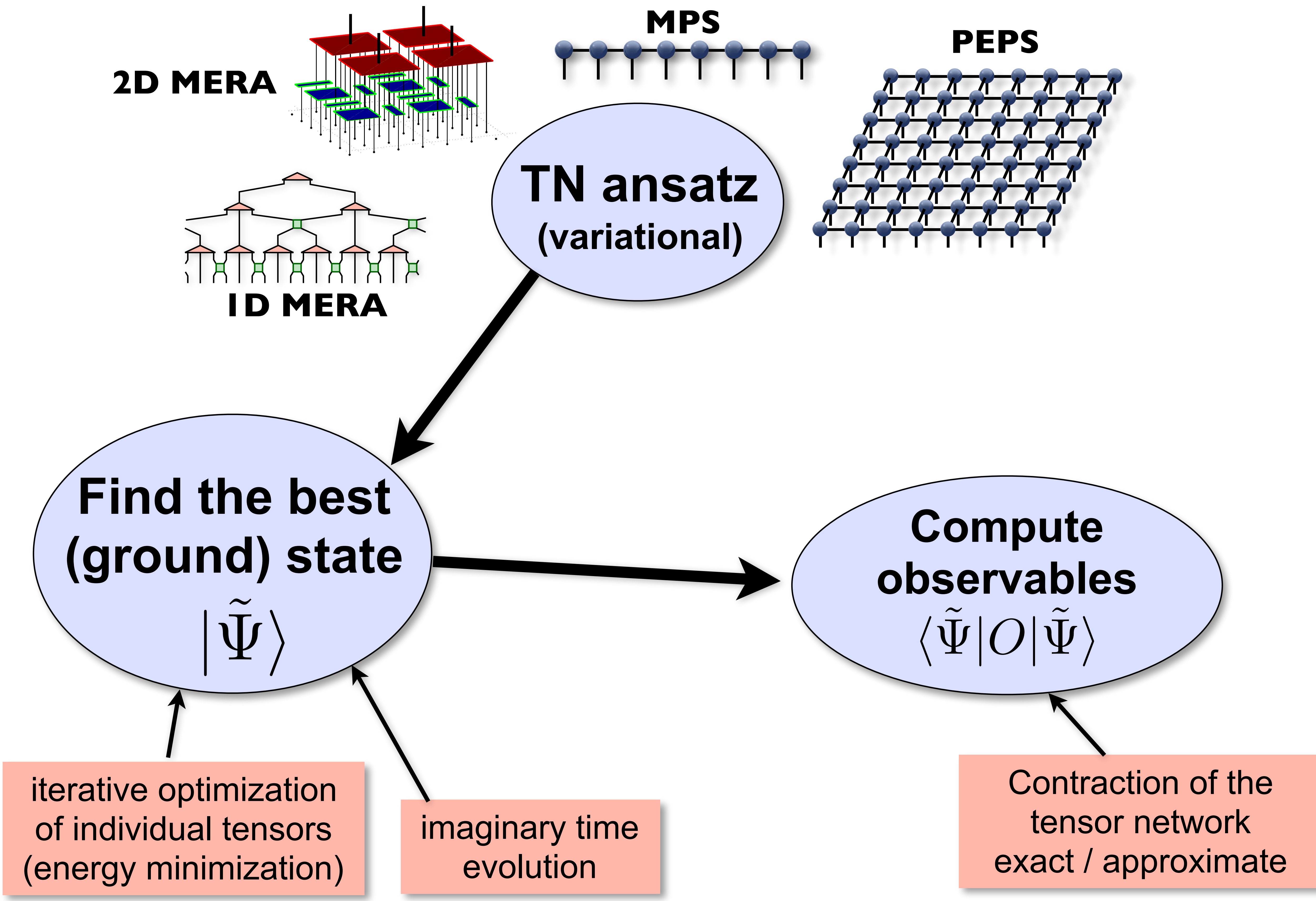
Lecture 11: tensor network algorithms (iPEPS)

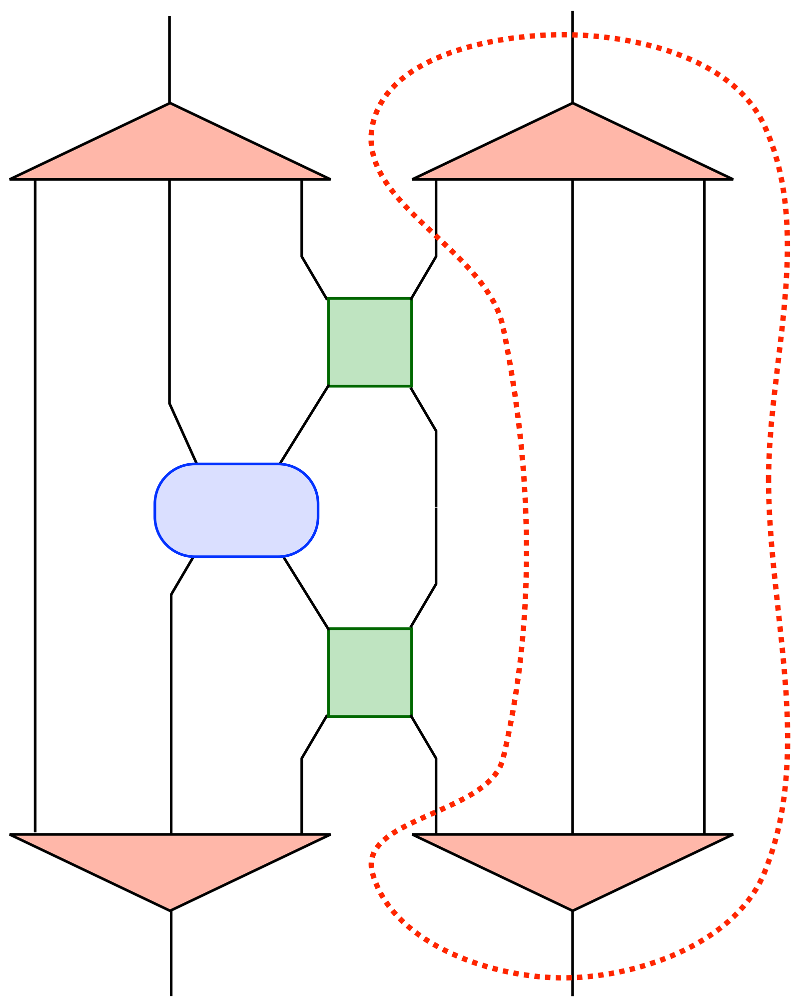
Philippe Corboz, Institute for Theoretical Physics, University of Amsterdam



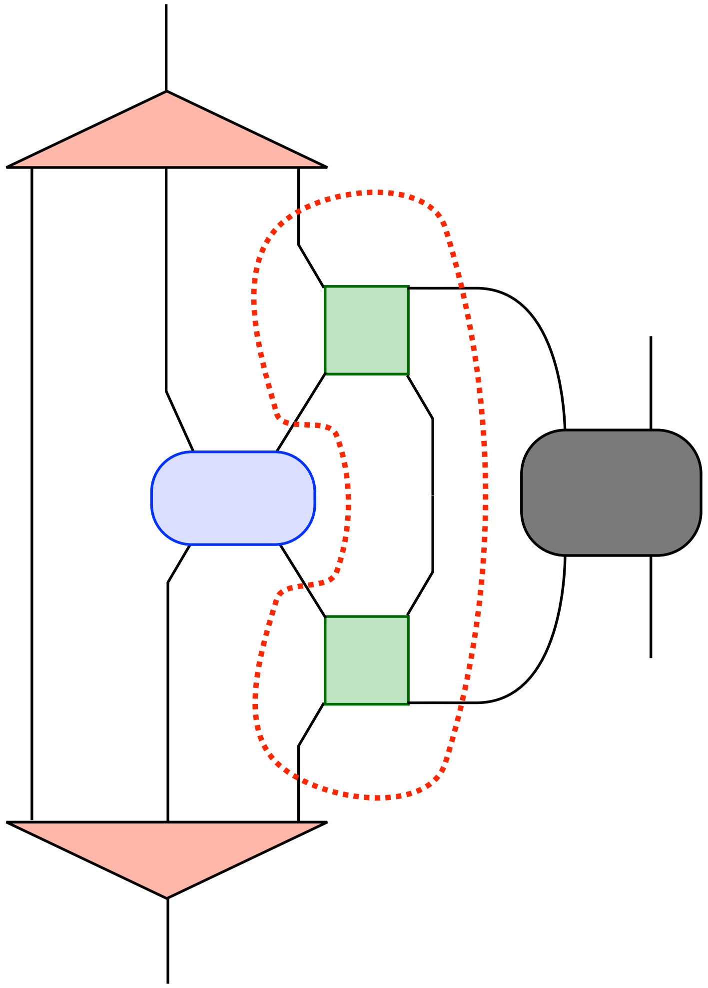
Overview: Tensor network algorithms (ground state)



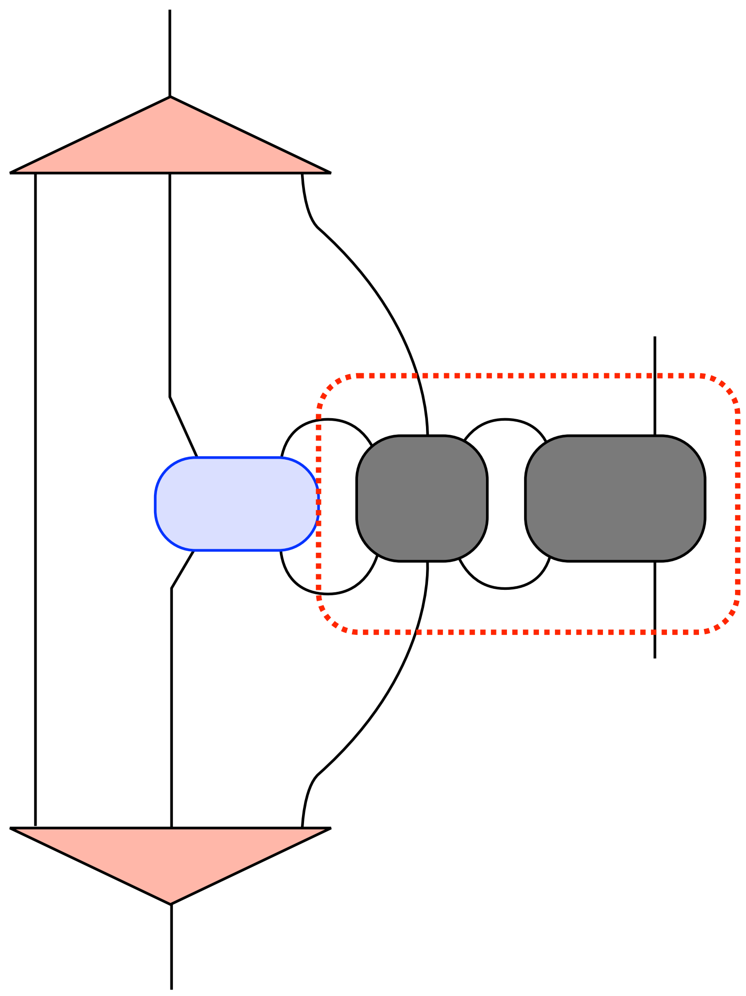
Contracting a tensor network



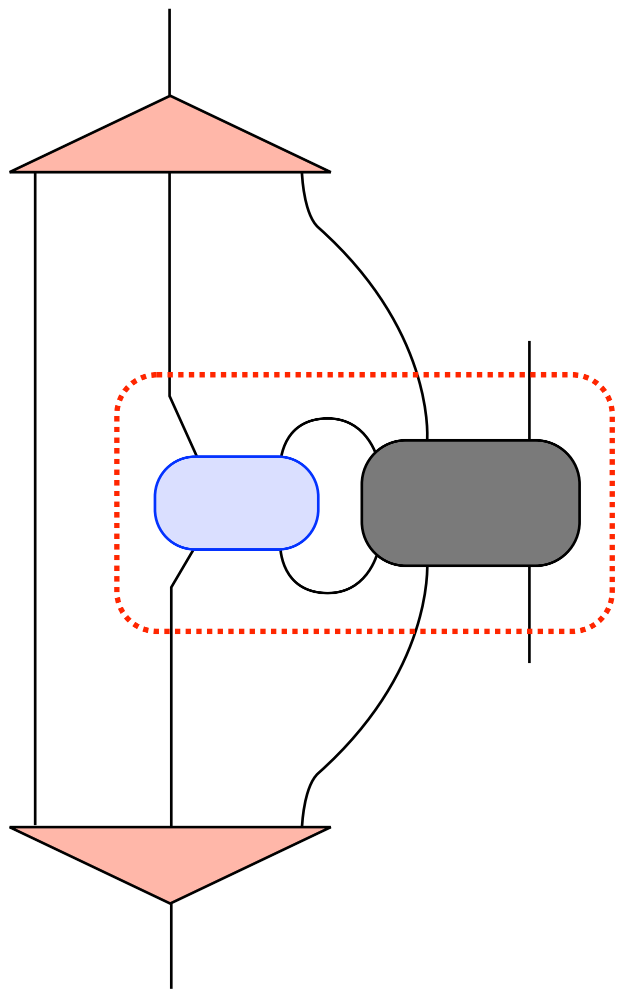
Pairwise contractions...



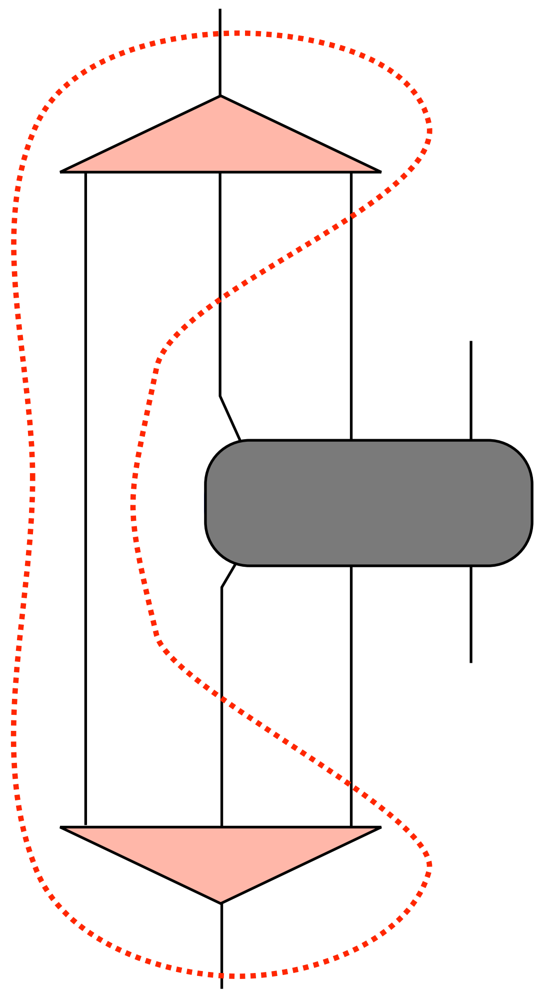
Pairwise contractions...



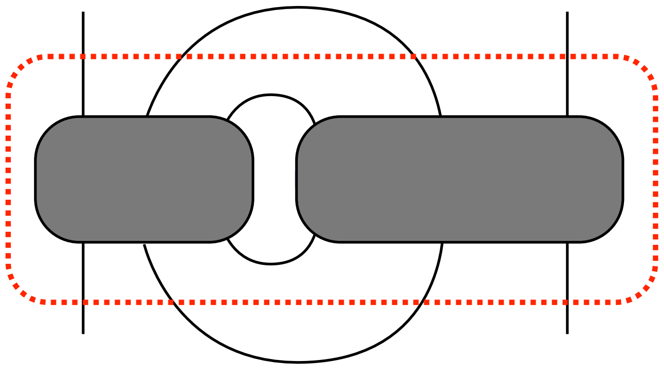
Pairwise contractions...



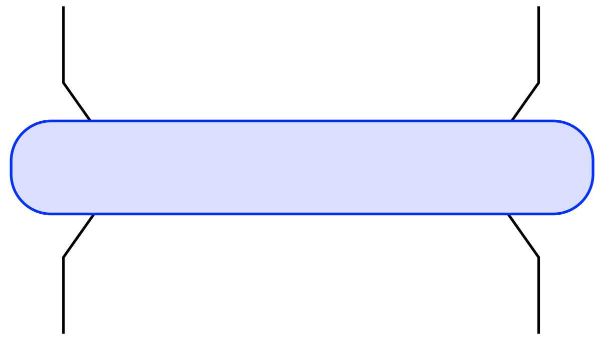
Pairwise contractions...



Pairwise contractions...



Pairwise contractions...

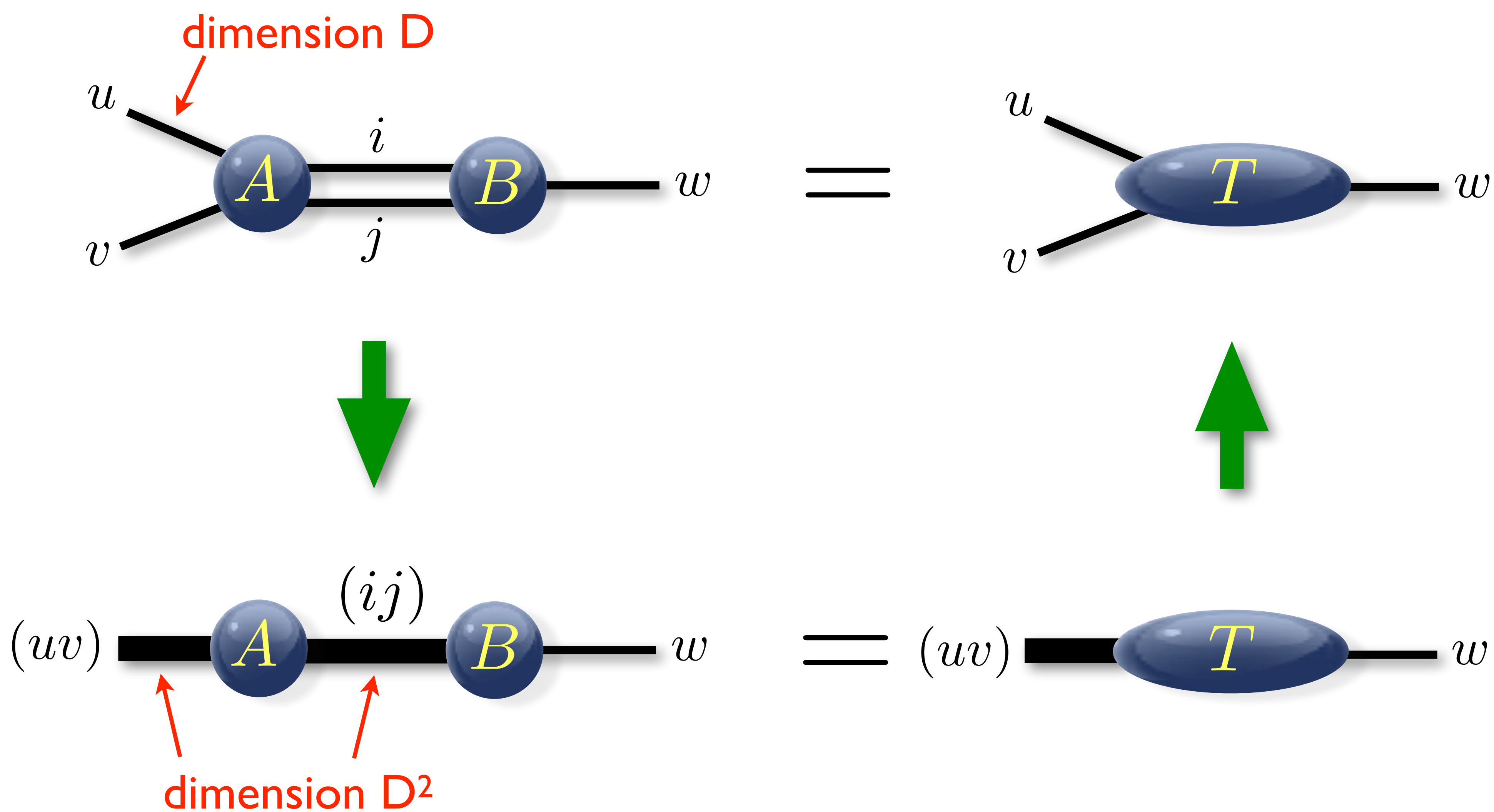


done!

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

Contracting a tensor network

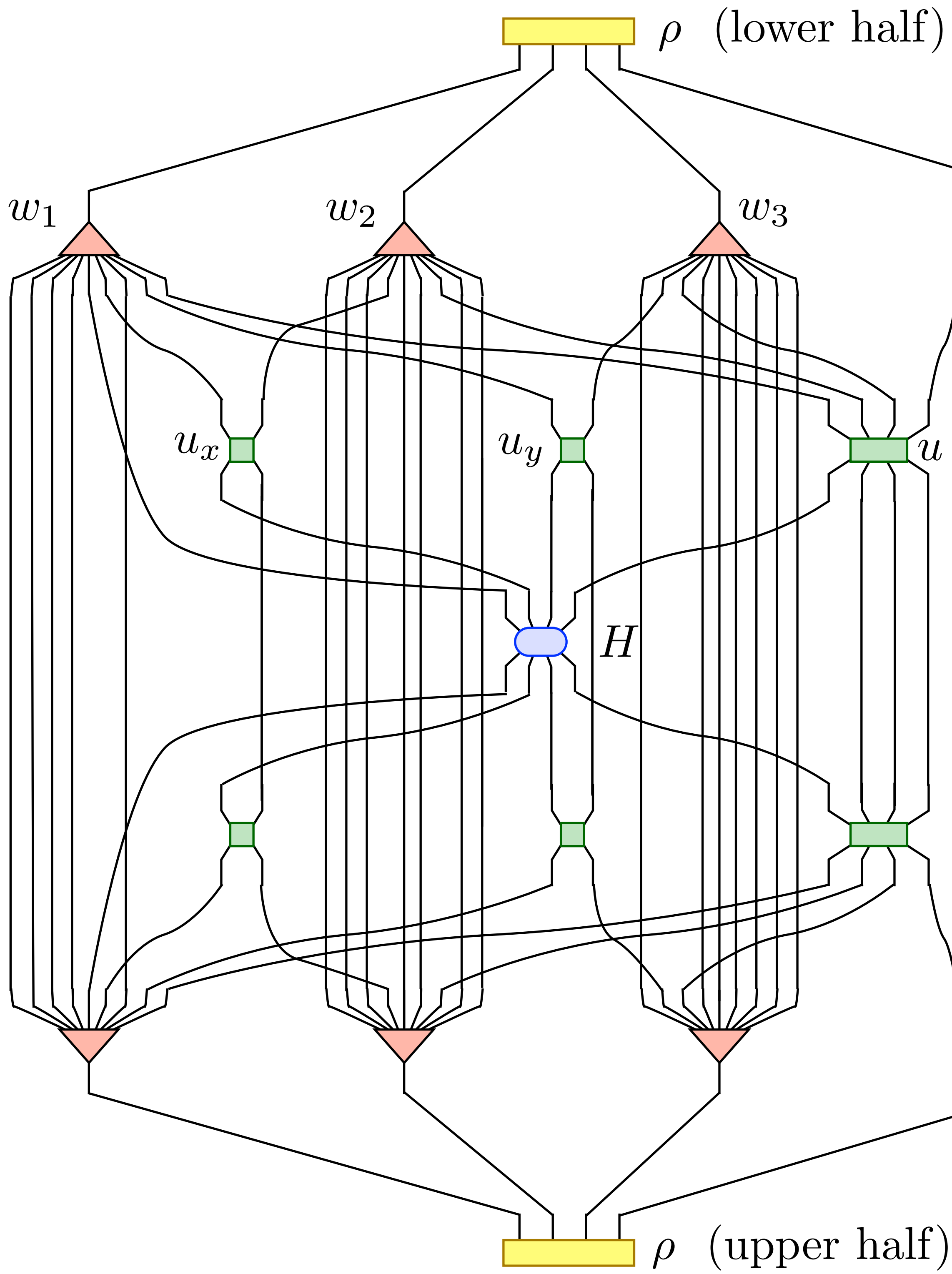
★ Reshape tensors into matrices and multiply them with optimized routines (BLAS)



cost D^5

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

Contraction: Example from the 2D MERA



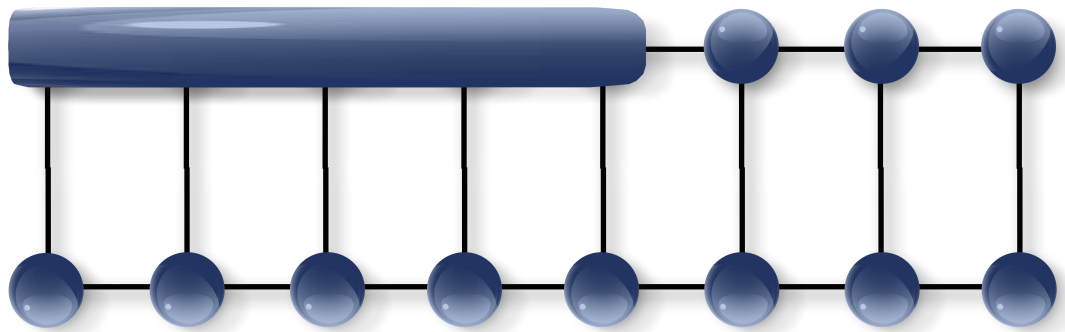
What is the optimal contraction order?

Use program to find optimal contraction, e.g. NETCON:

Pfeifer, Haegeman, Verstraete, PRE 90 (2014)

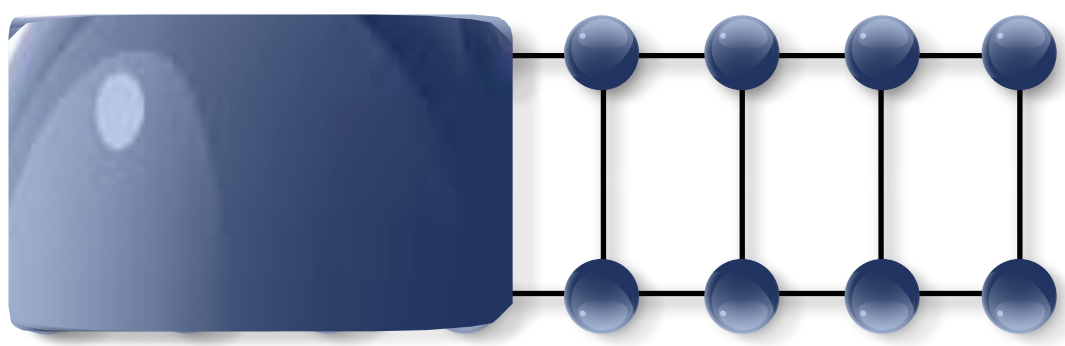
Contracting an MPS

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



BAD!

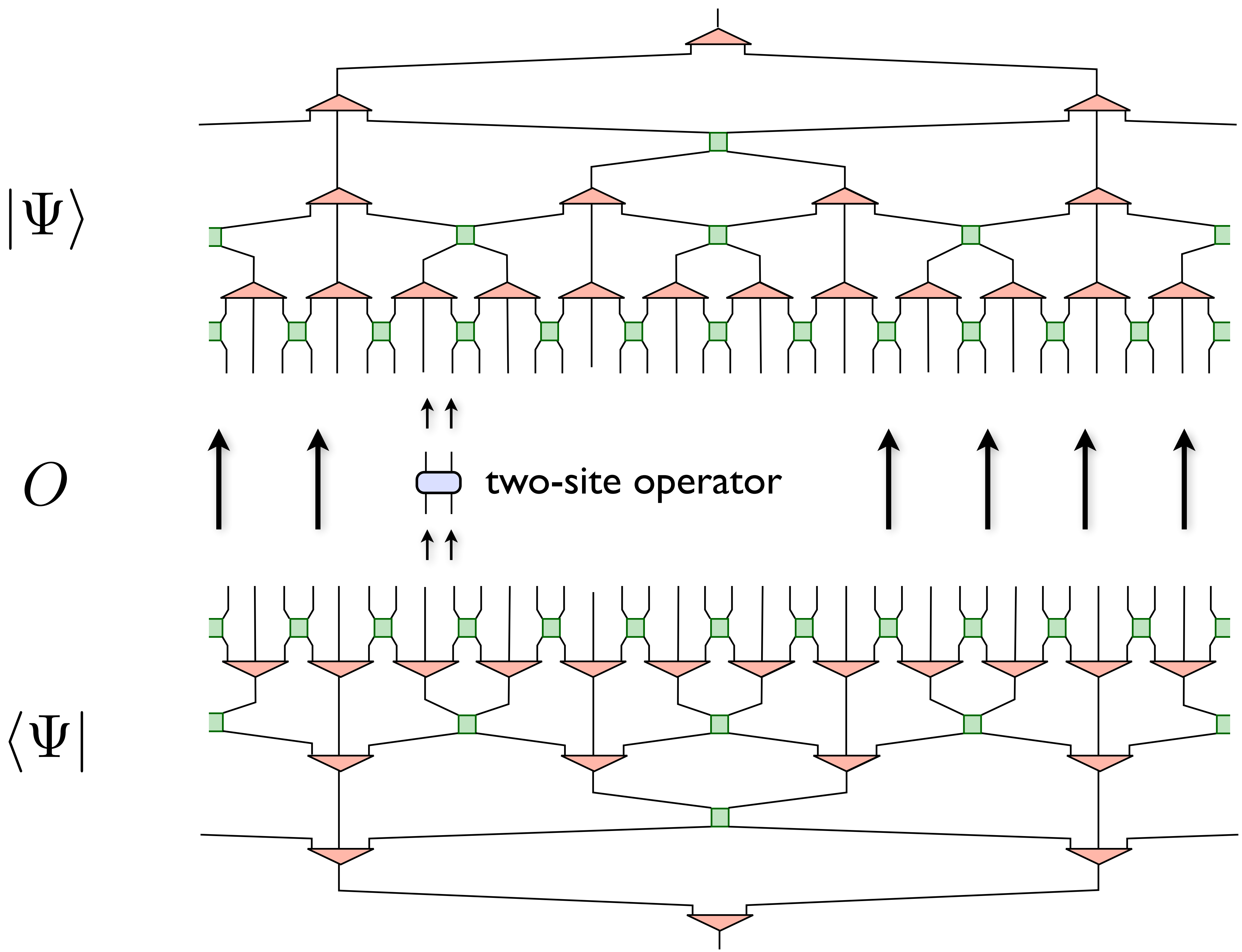
$$\langle \Psi | \Psi \rangle =$$



Good!

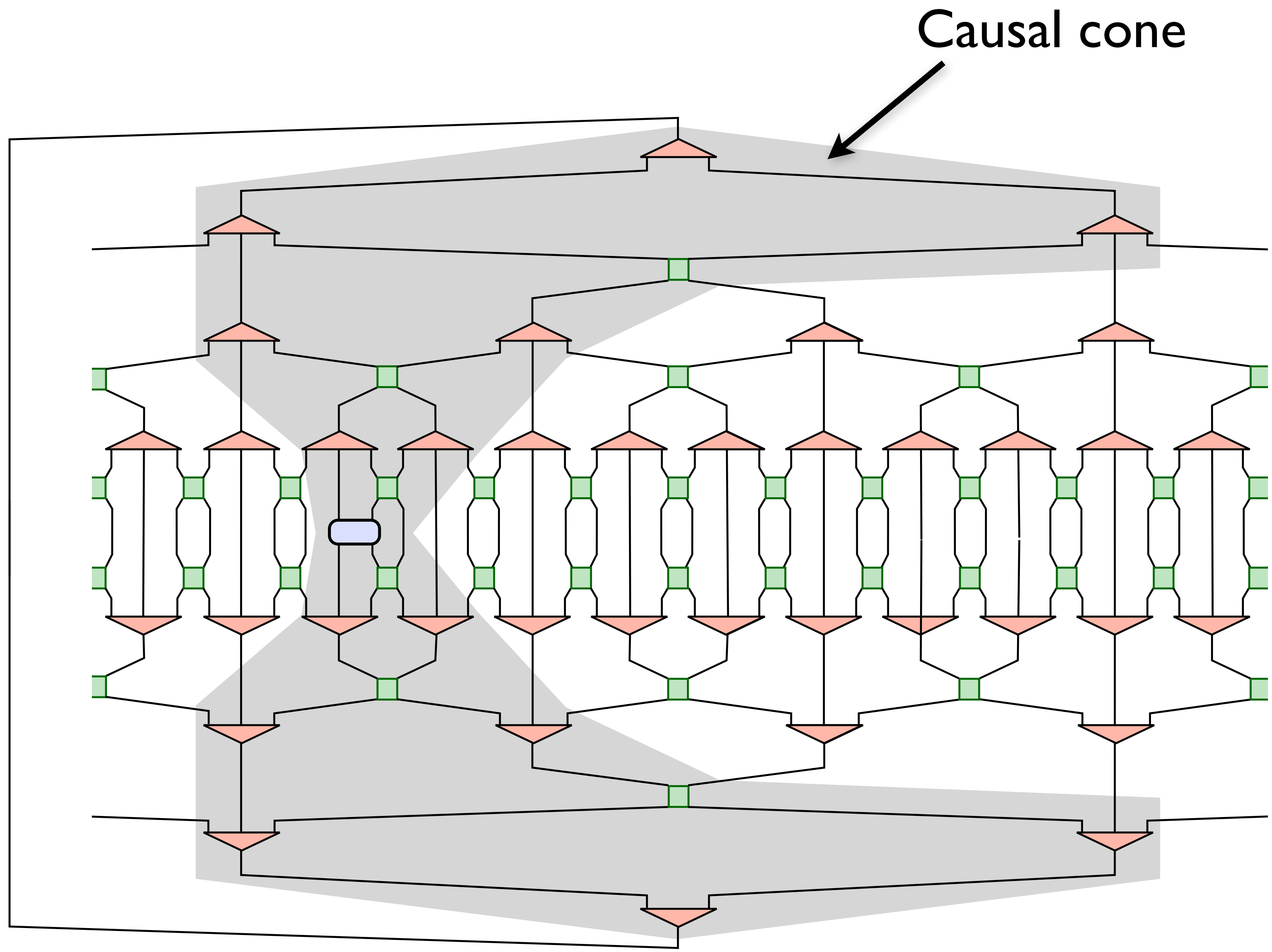
MERA: Contraction

Let's compute $\langle\Psi|O|\Psi\rangle$ O: two-site operator

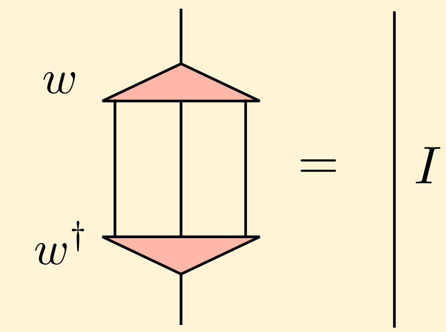


MERA: Contraction

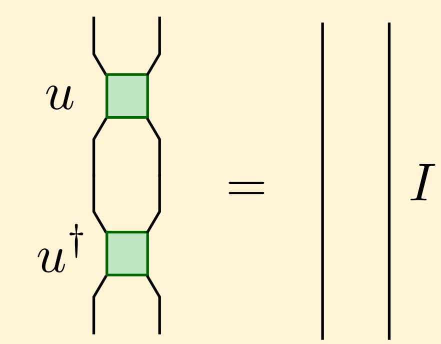
$\langle\Psi|O|\Psi\rangle$



Isometries are *isometric*

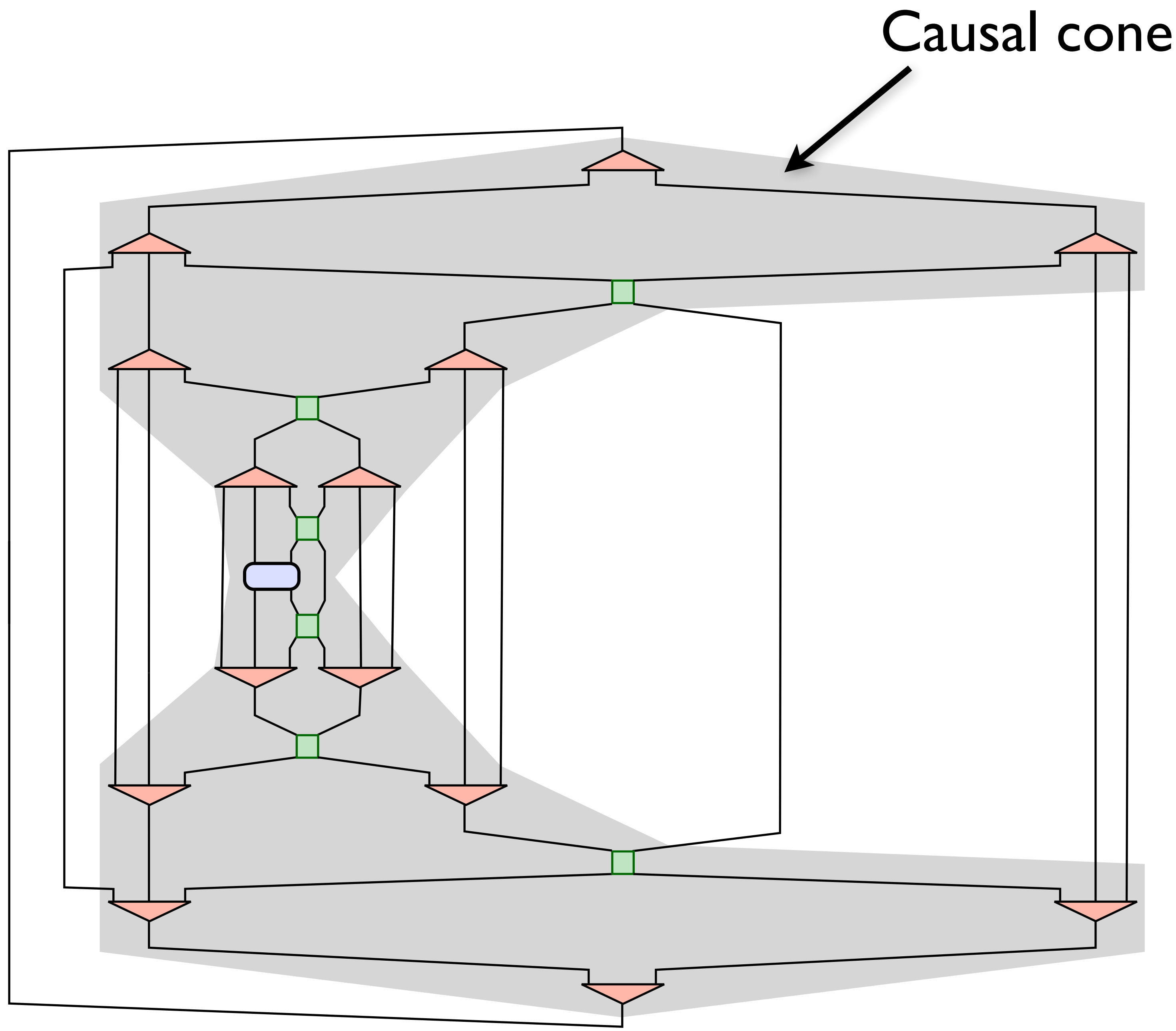


Disentanglers are *unitary*

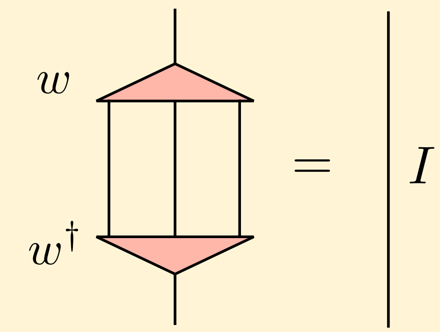


MERA: Contraction

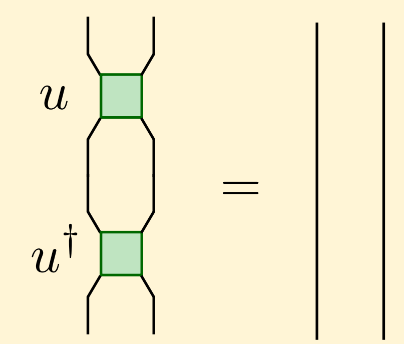
$\langle\Psi|O|\Psi\rangle$



Isometries are *isometric*

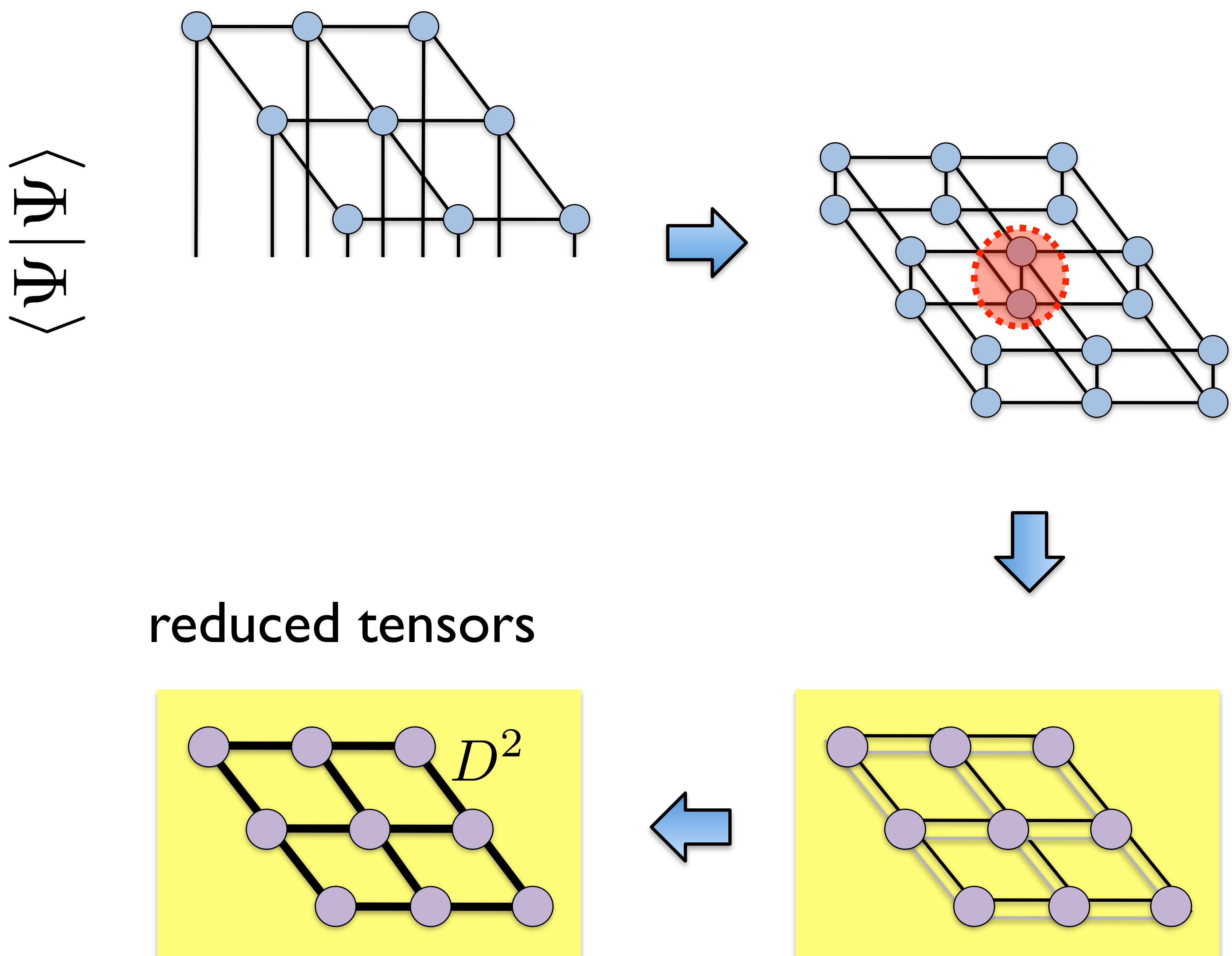


Disentanglers are *unitary*

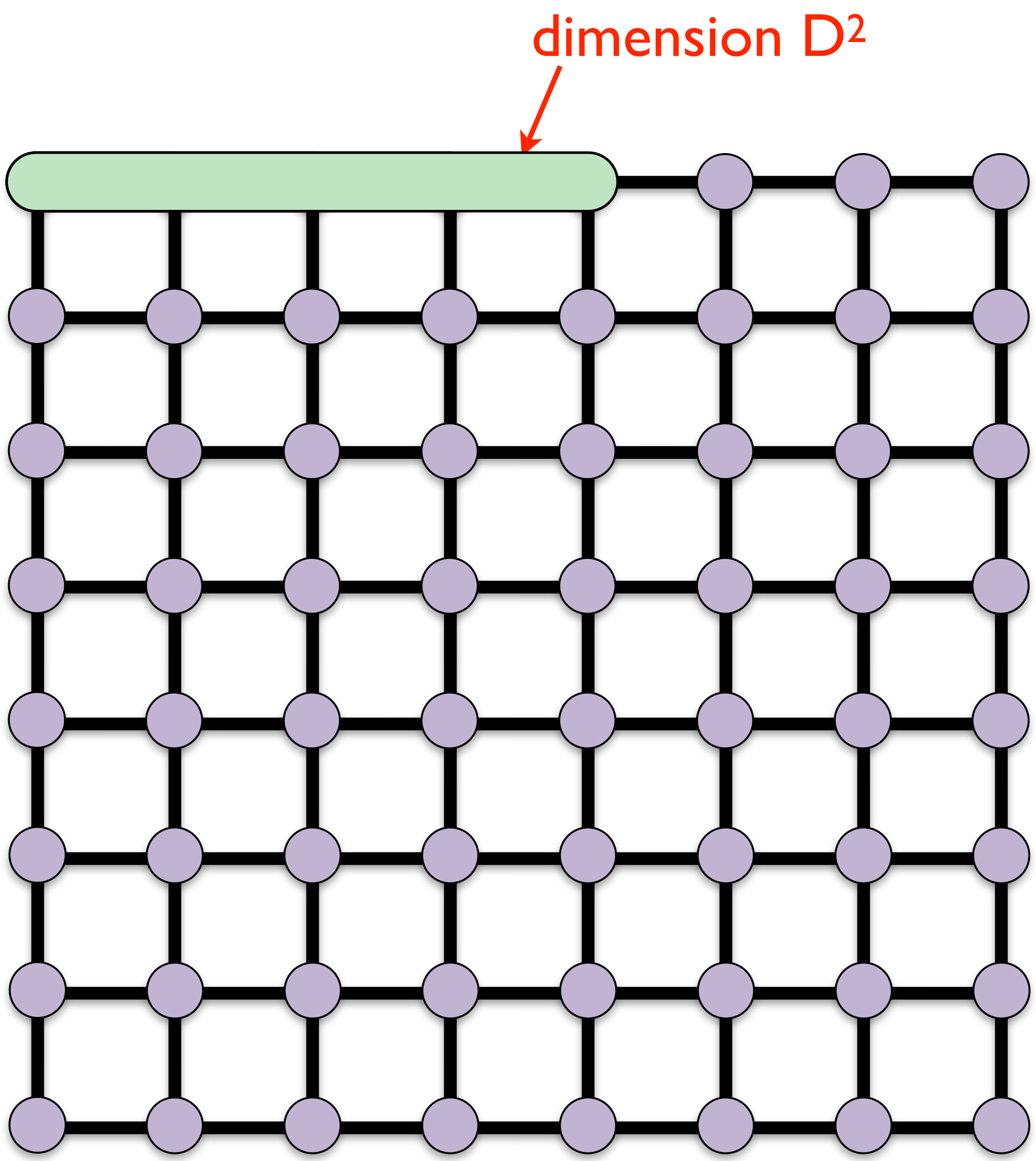


Efficient computation of expectation values of observables!

Contracting the PEPS



Contracting the PEPS



Problem: how do we contract this??

**no matter how we contract,
we will get intermediate
tensors with O(L) legs**

number of coefficients D^{2L}

Exponentially increasing with L!

NOT EFFICIENT

Contracting the PEPS

★ Exact contraction of an PEPS is exponentially hard!

→ *use controlled approximate contraction scheme*

MPS-MPO-based approaches

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)

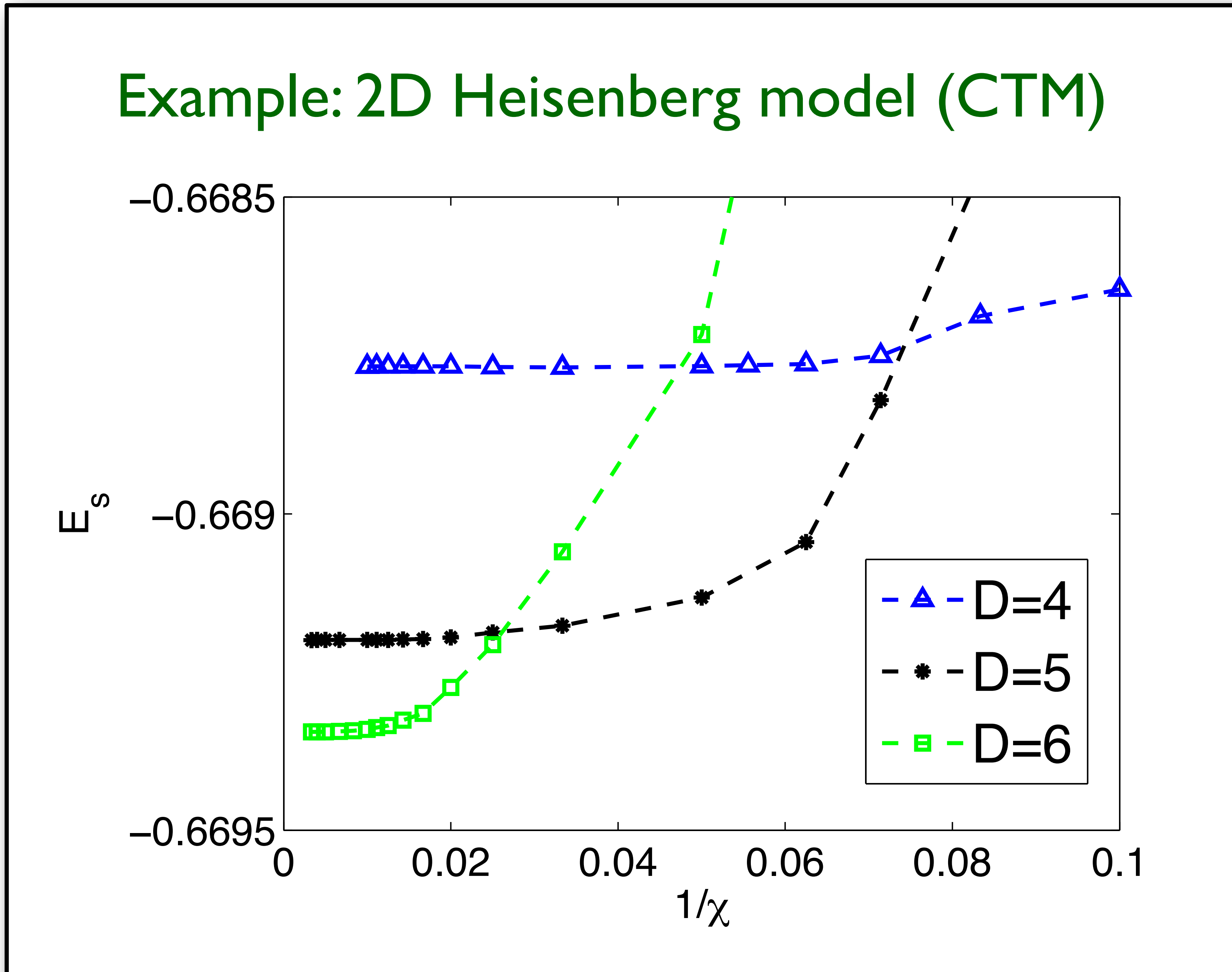
Loop-TNR:
Yang, Gu & Wen, PRL 118 (2017)

★ 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!!!

Contracting the PEPS

★ Exact contraction of an PEPS is exponentially hard!

→ *use controlled approximate contraction scheme*

MPS-MPO-based approaches

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)

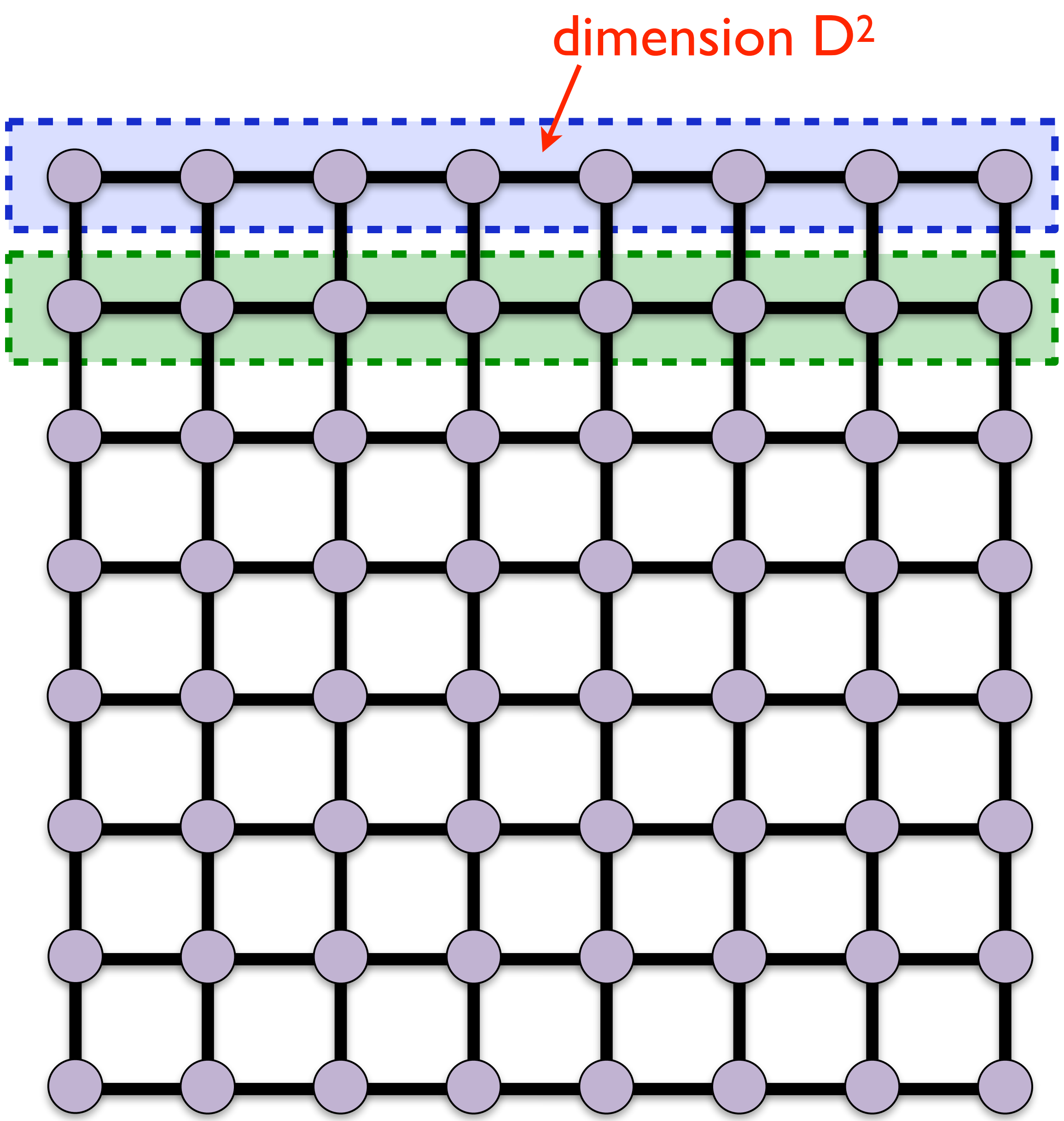
★ Accuracy of the approximate contraction is controlled by "boundary dimension" χ

★ Convergence in χ needs to be carefully checked

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

Contracting the PEPS using an MPS

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

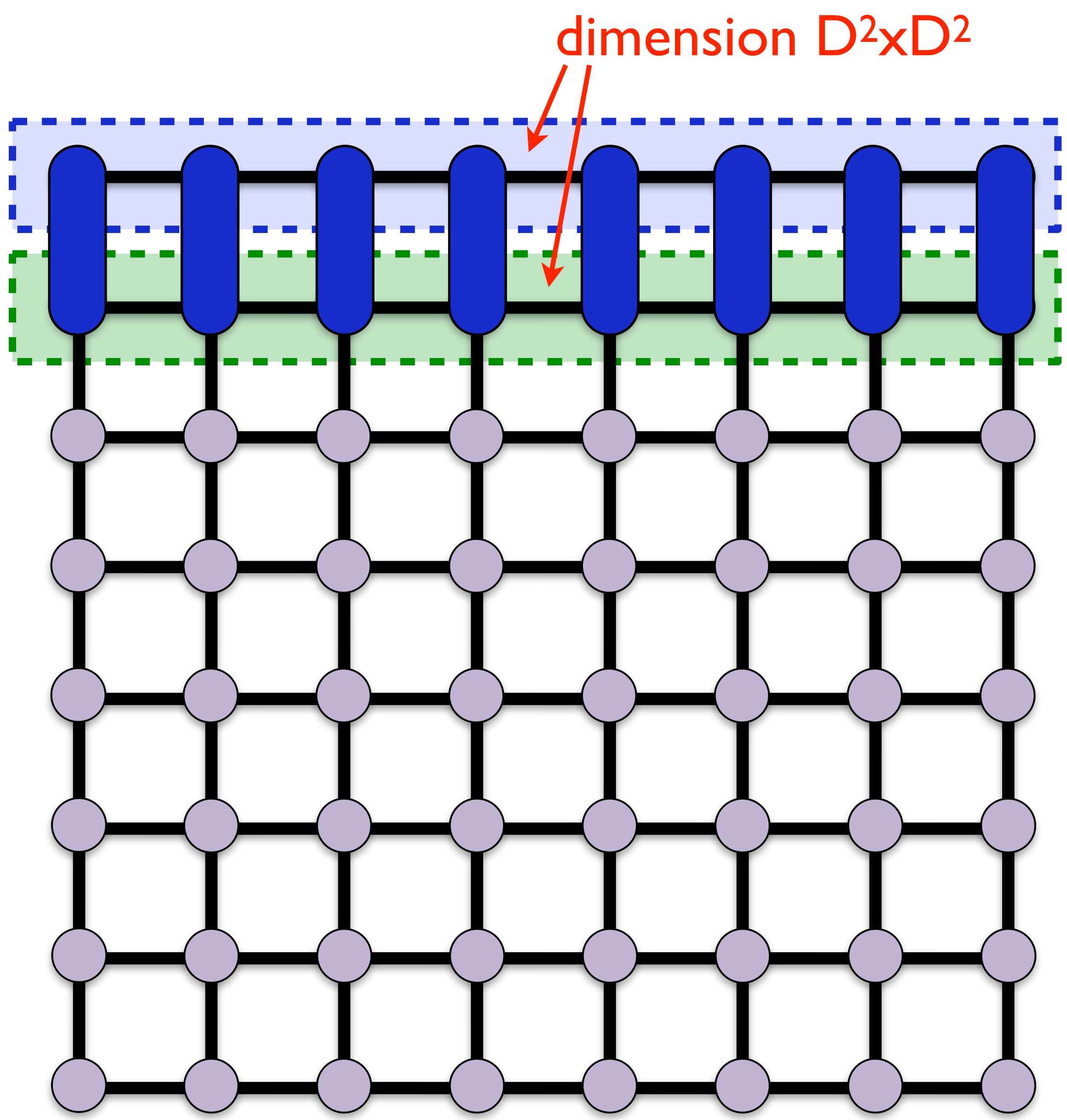


this is an MPS

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 $D^2 \times D^2$

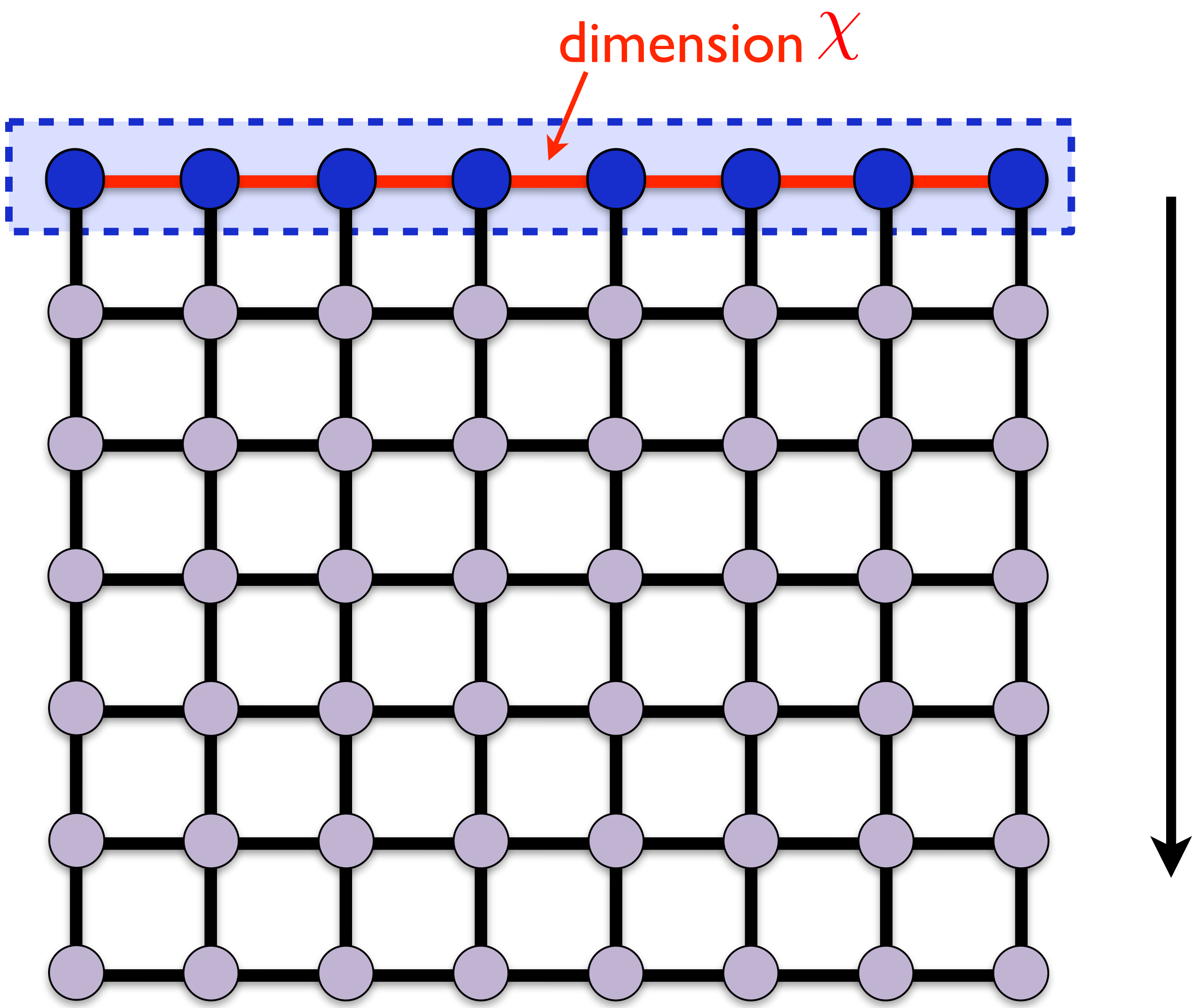
truncate the bonds to χ

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

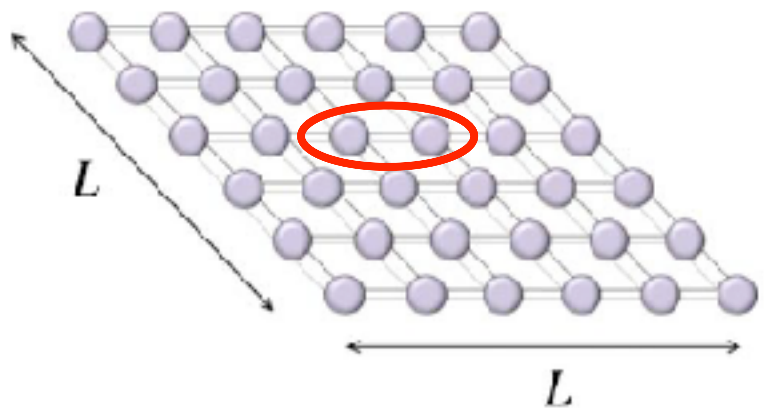
Verstraete, Murg, Cirac, Adv. in Phys. 57, 143 (2008)

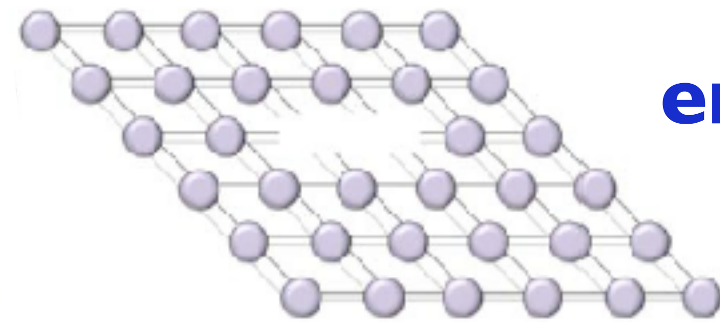


proceed...

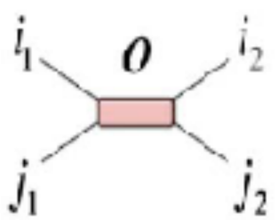
★ We can do this from several directions
★ Similar procedure when computing an expectation value

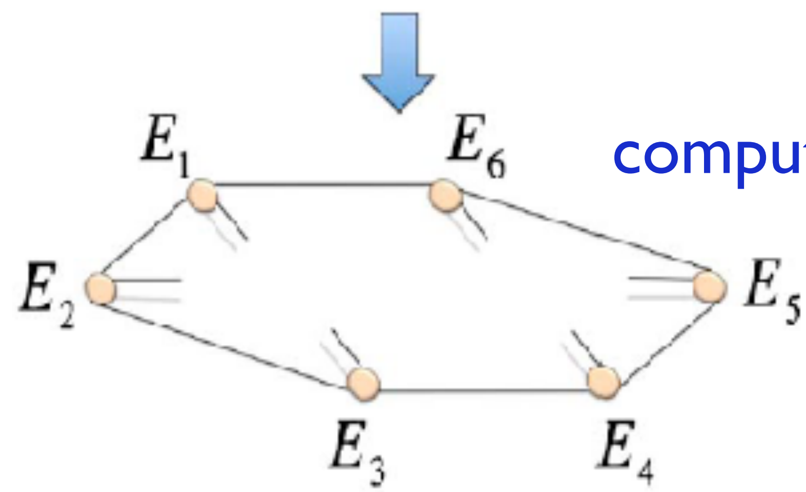
Compute expectation values



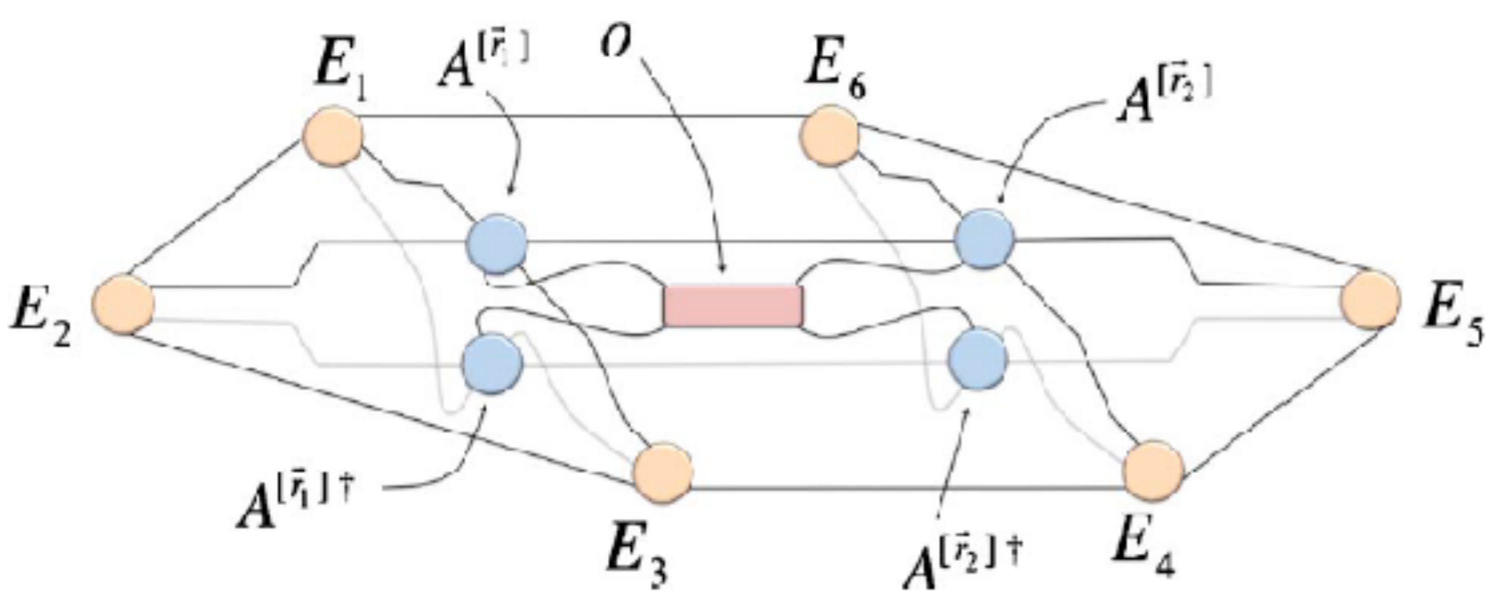


environment





compute environment approximately



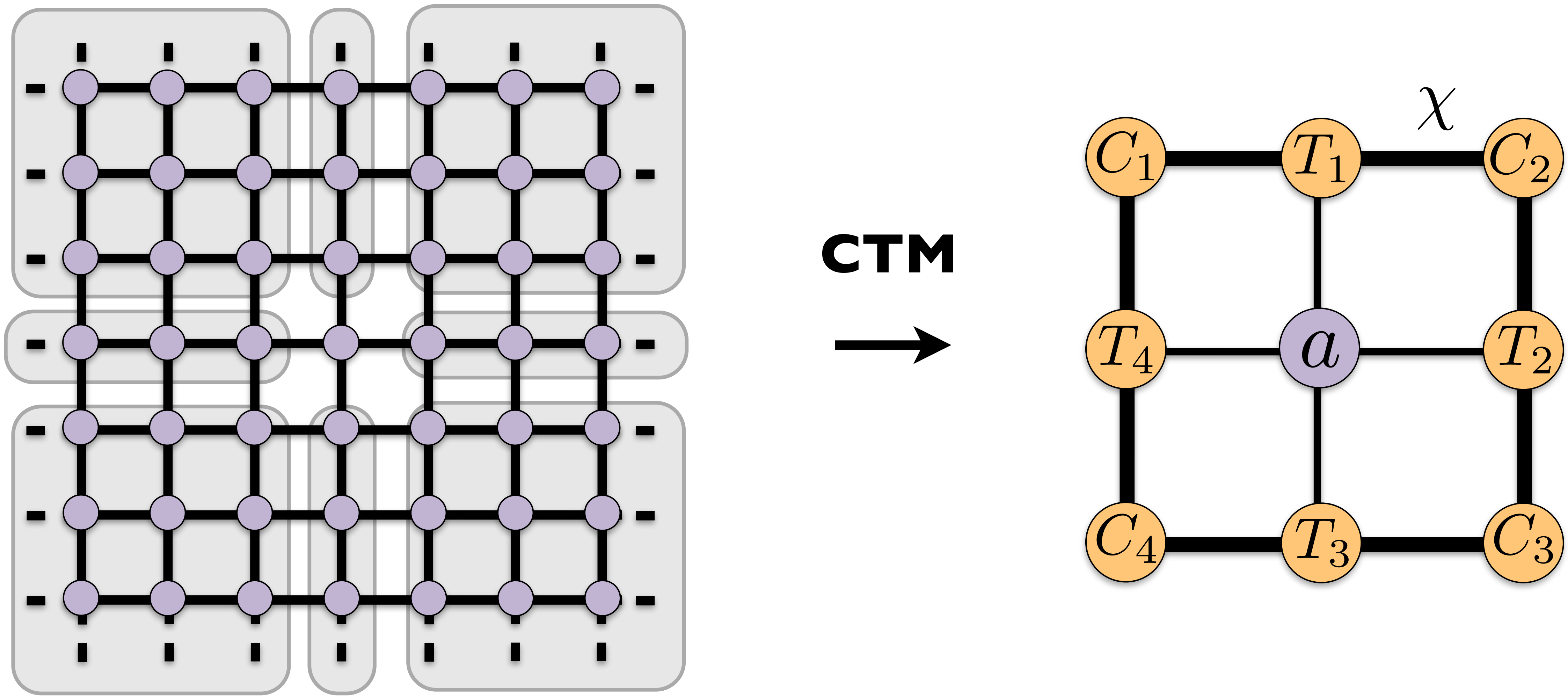
Connect two-body operator

Contract this network!

Figure taken from Corboz, Orús, Bauer, Vidal, PRB 81, 165104 (2010)

Contracting the iPEPS using the corner transfer matrix method

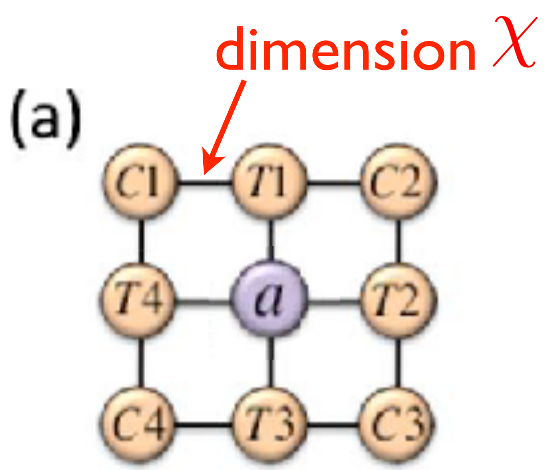
Nishino, Okunishi, JPSJ65 (1996)



- 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

Nishino, Okunishi, JPSJ65 (1996)
Orus, Vidal, PRB 80 (2009)

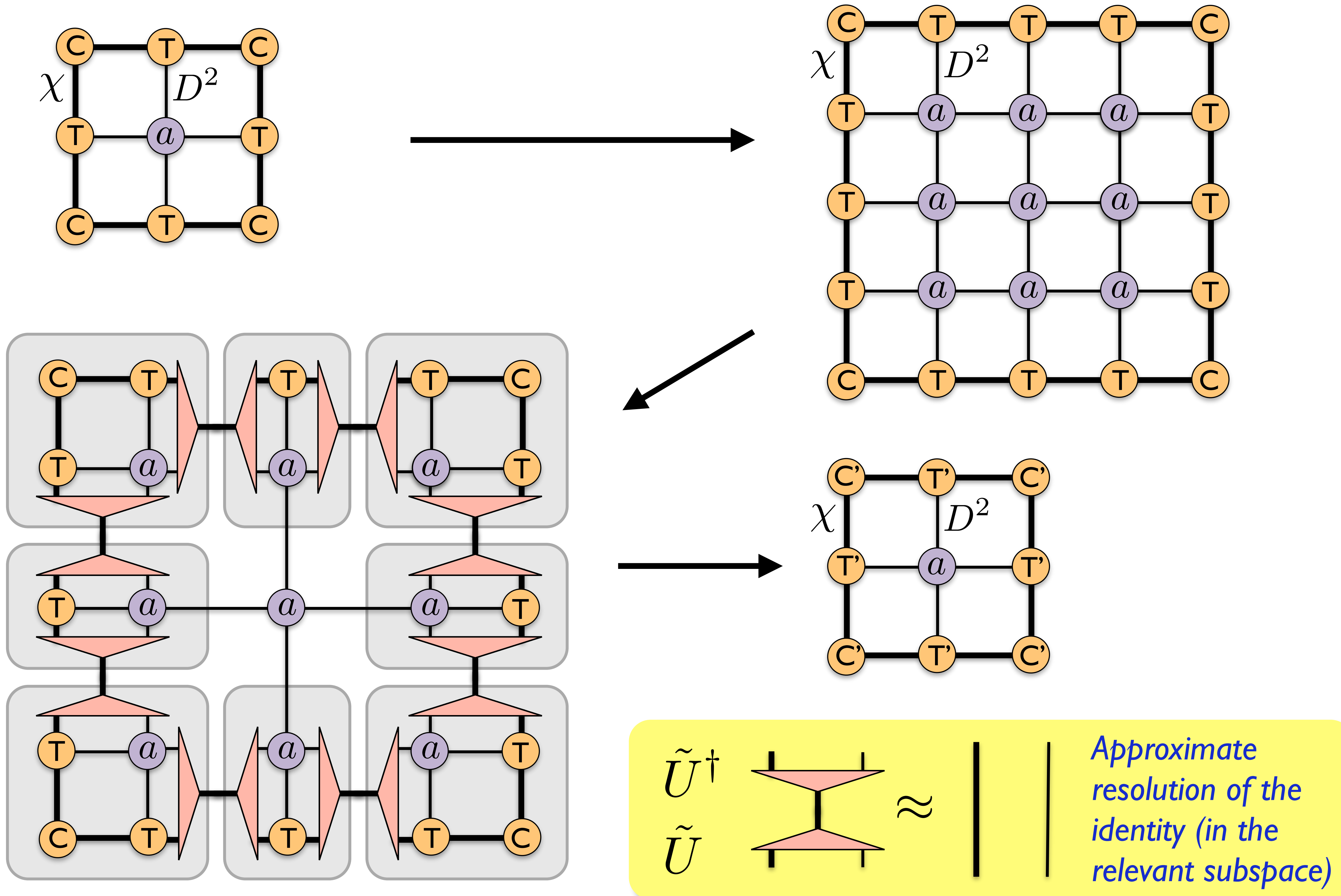


- ★ Let the system grow in all directions.
- ★ Repeat until convergence is reached
- ★ The boundary tensors form the **environment**
- ★ Can be generalized to arbitrary unit cell sizes

Corboz, et al., PRB 84 (2011)

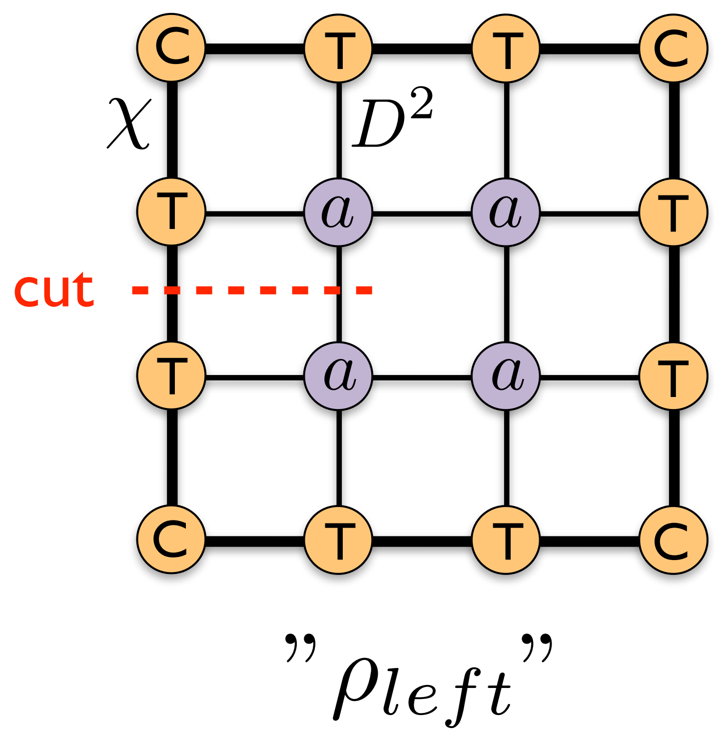
Simplest case: rotational symmetric tensors

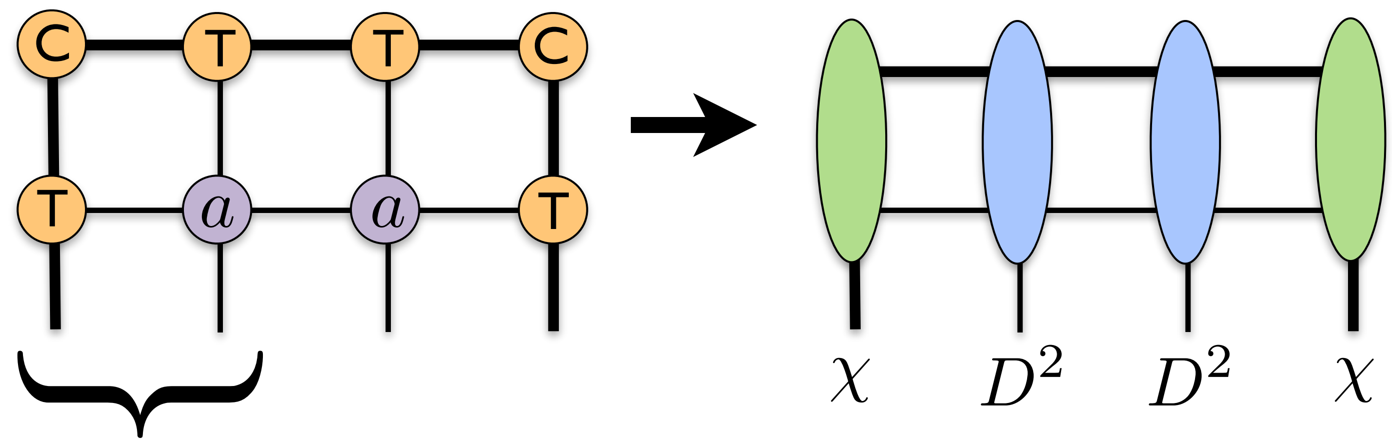
Nishino, Okunishi, JPSJ65 (1996)



Simplest case: rotational symmetric tensors

Nishino, Okunishi, JPSJ65 (1996)

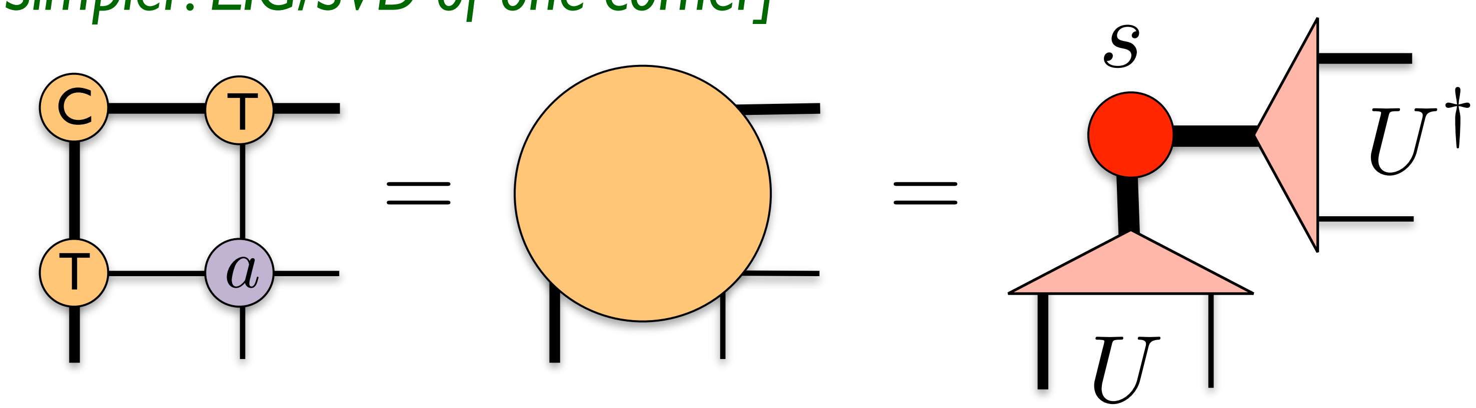




Relevant subspace?

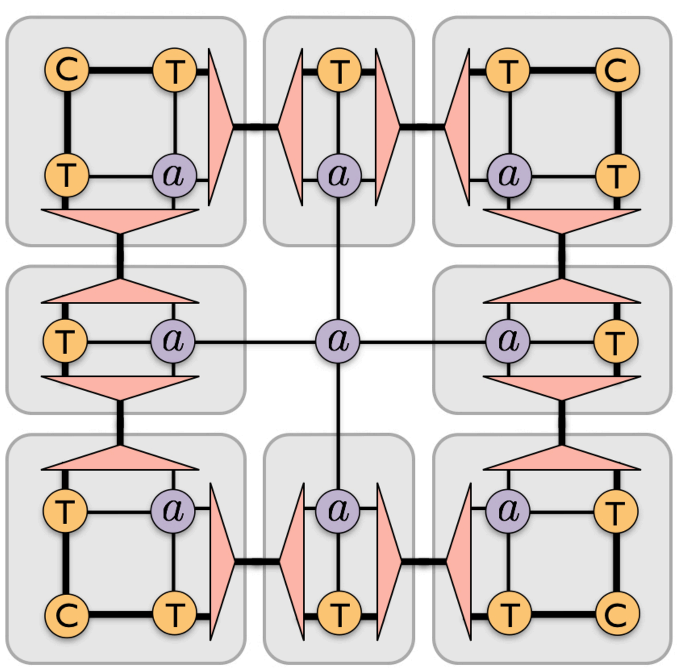
DMRG: Eigenvectors with largest eigenvalues of ρ_{left}

[Simpler: EIG/SVD of one corner]

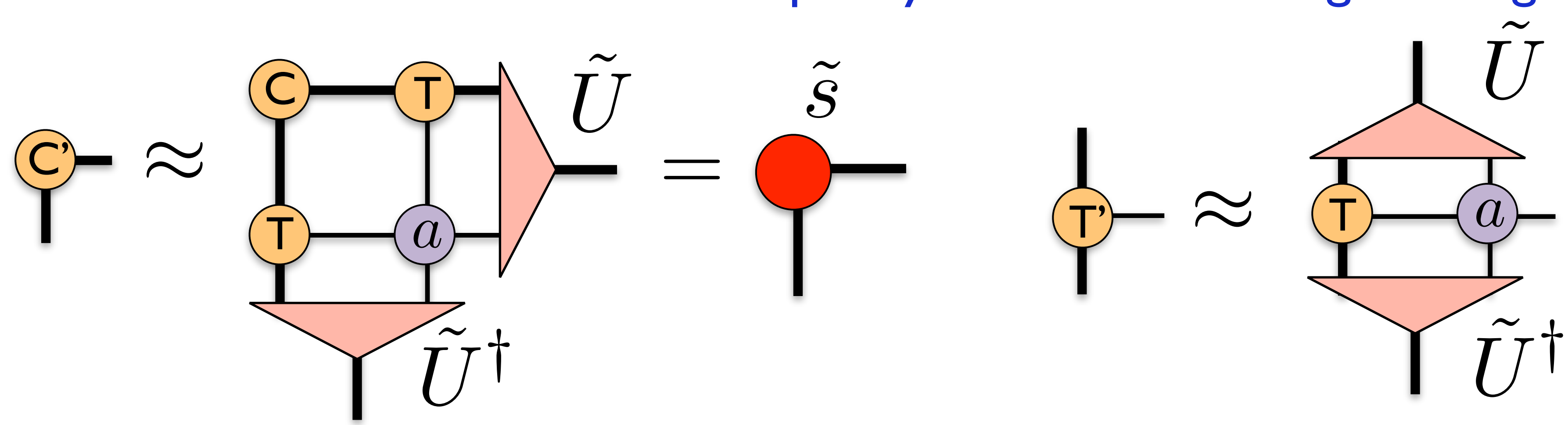


How can we best truncate from

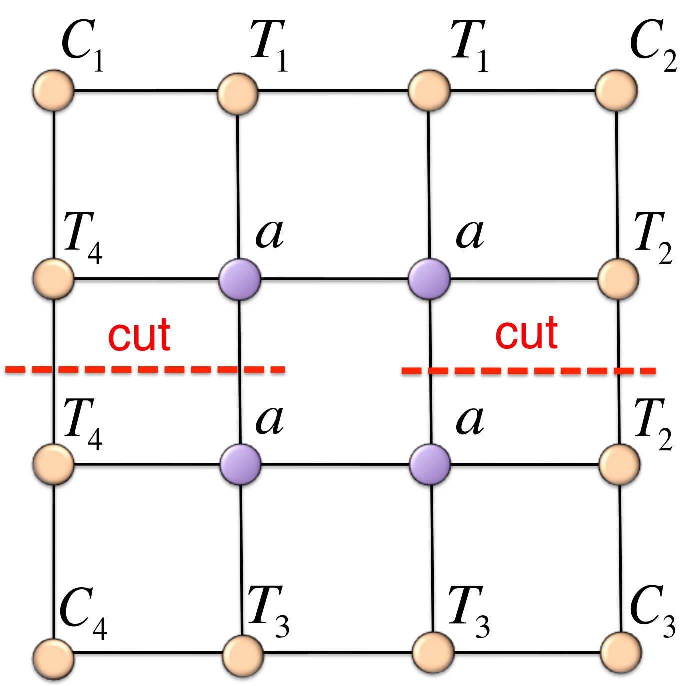
$$\chi D^2 \to \chi$$

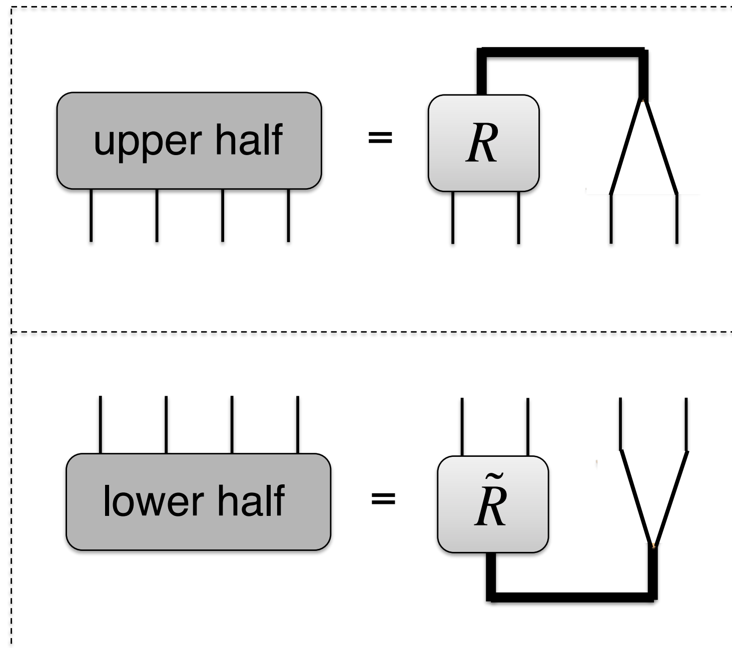


Renormalized tensors: keep only χ states with largest weight

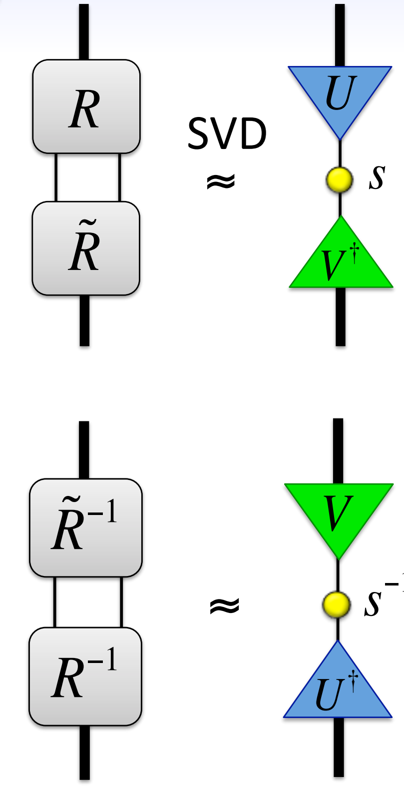


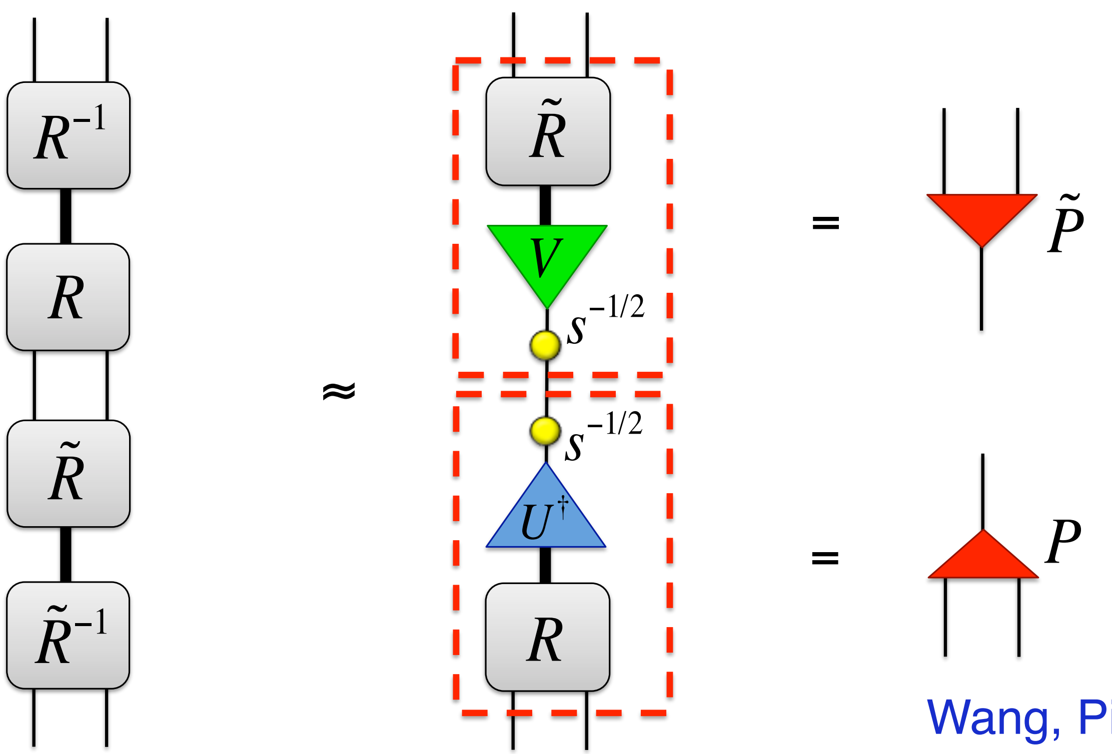
General case: Renormalization step (left move)





alternatively: only use upper left and lower left corners

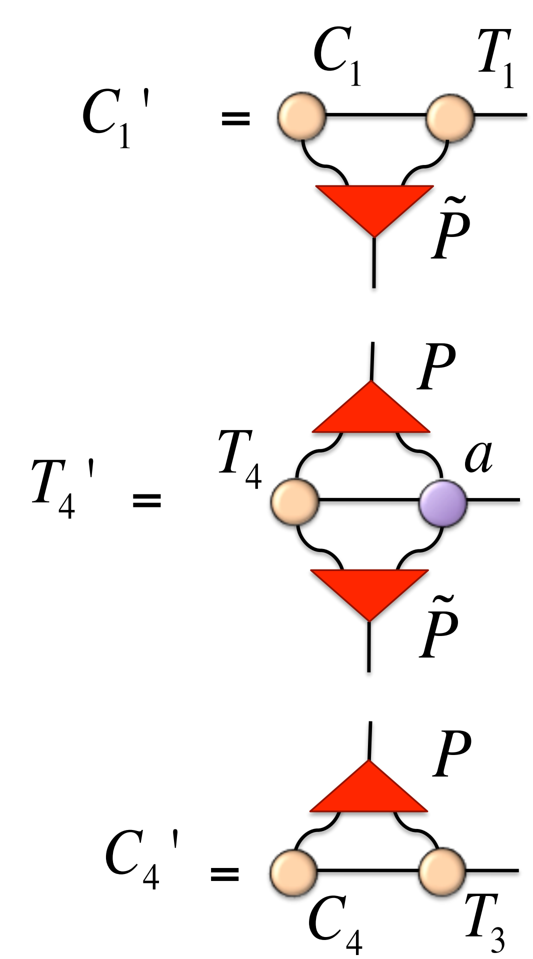




identity

approx. identity

projectors onto relevant subspace

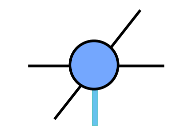


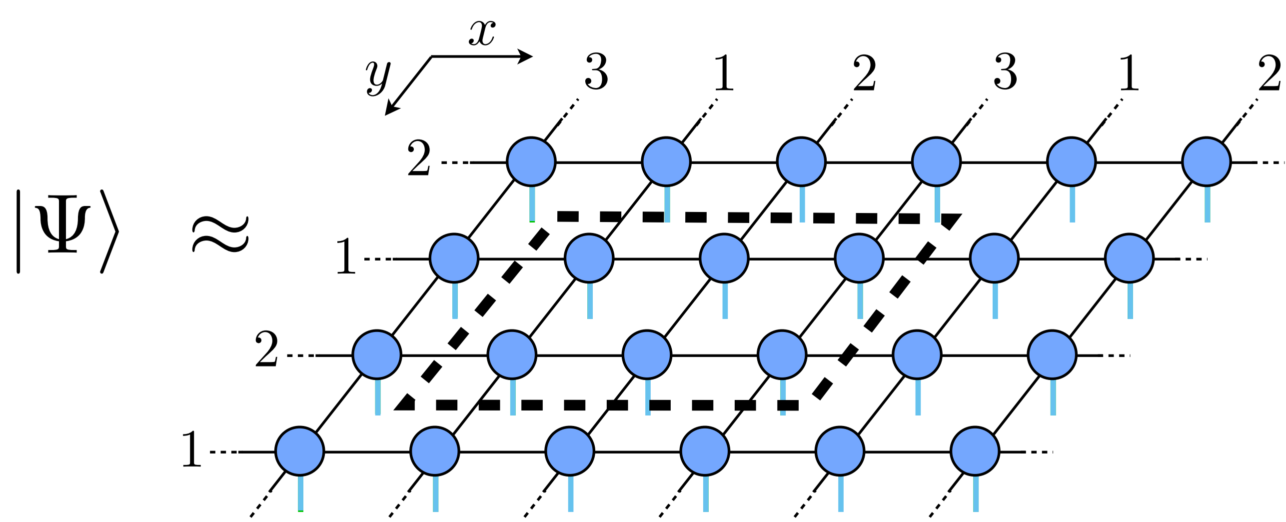
Wang, Pižorn & Verstraete, PRB 83 (2011)
Huang, Chen & Kao, PRB 86 (2012)
PC, Rice, Troyer, PRL 113 (2014)
T. Okubo, private comm.

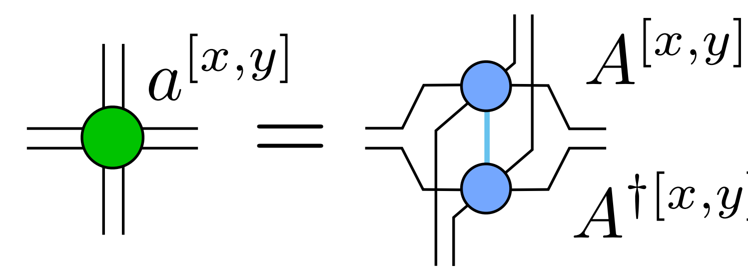
CTM with larger unit cells

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

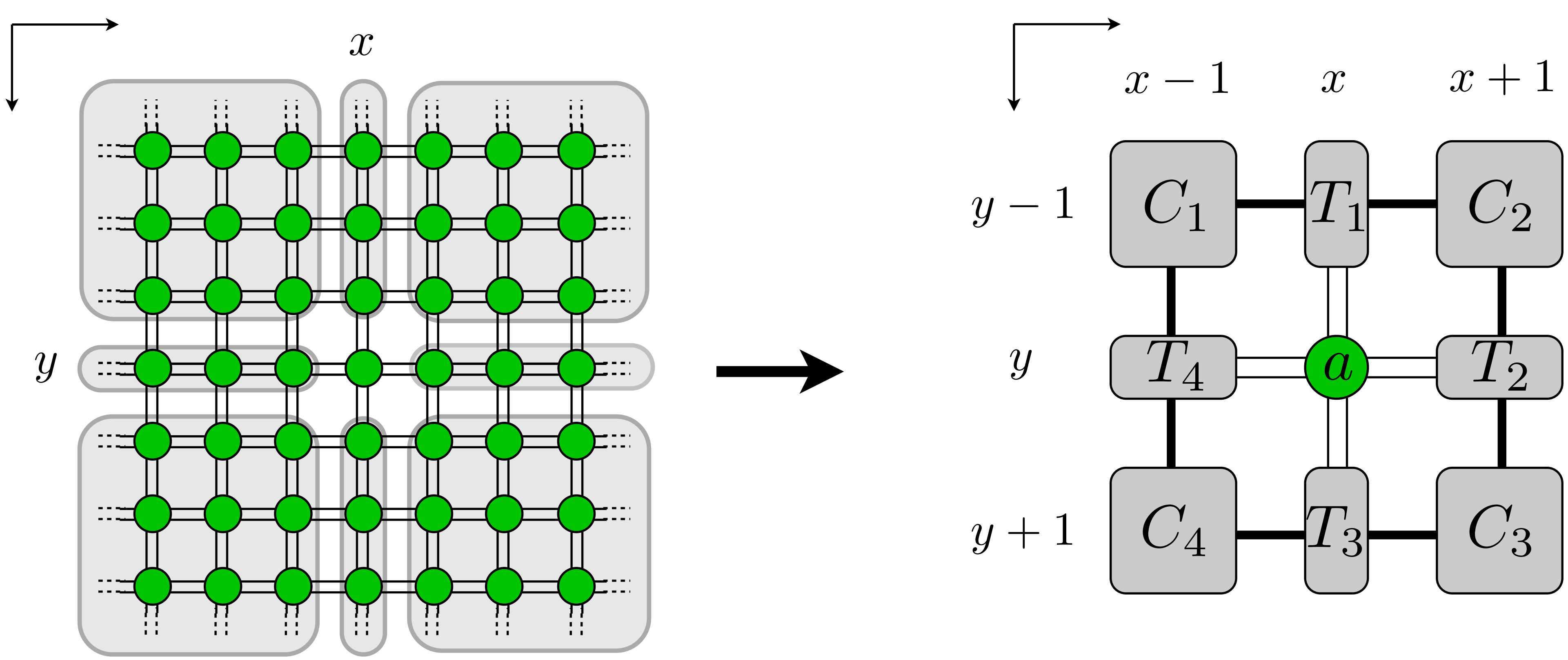
★ Each tensor has coordinates with respect to the unit cell: $A^{[x,y]}$





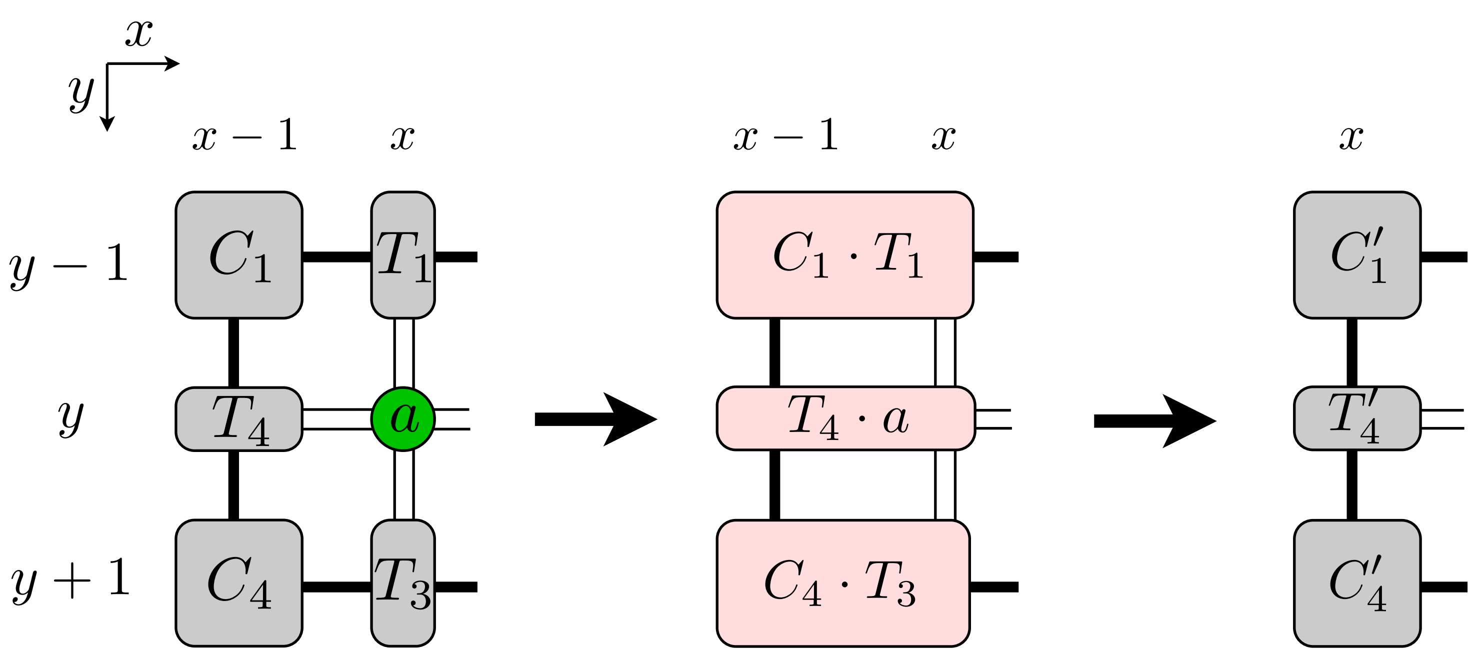


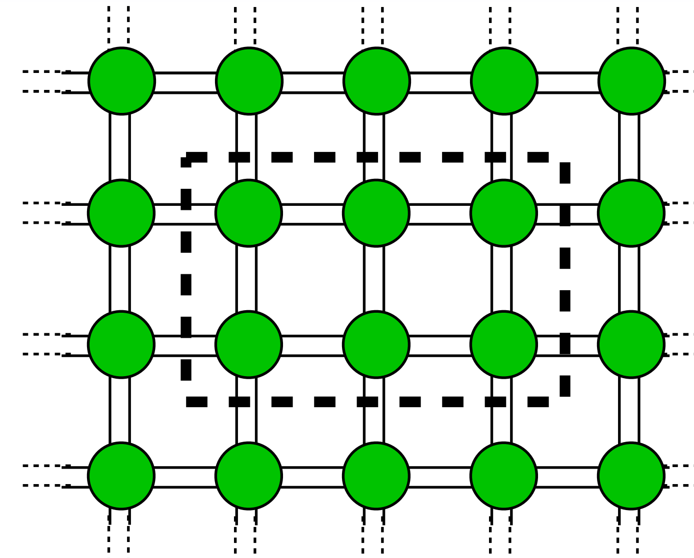
★ Keep a copy of every environment tensors C1, ... C4, T1, ..., T4 for each coordinate

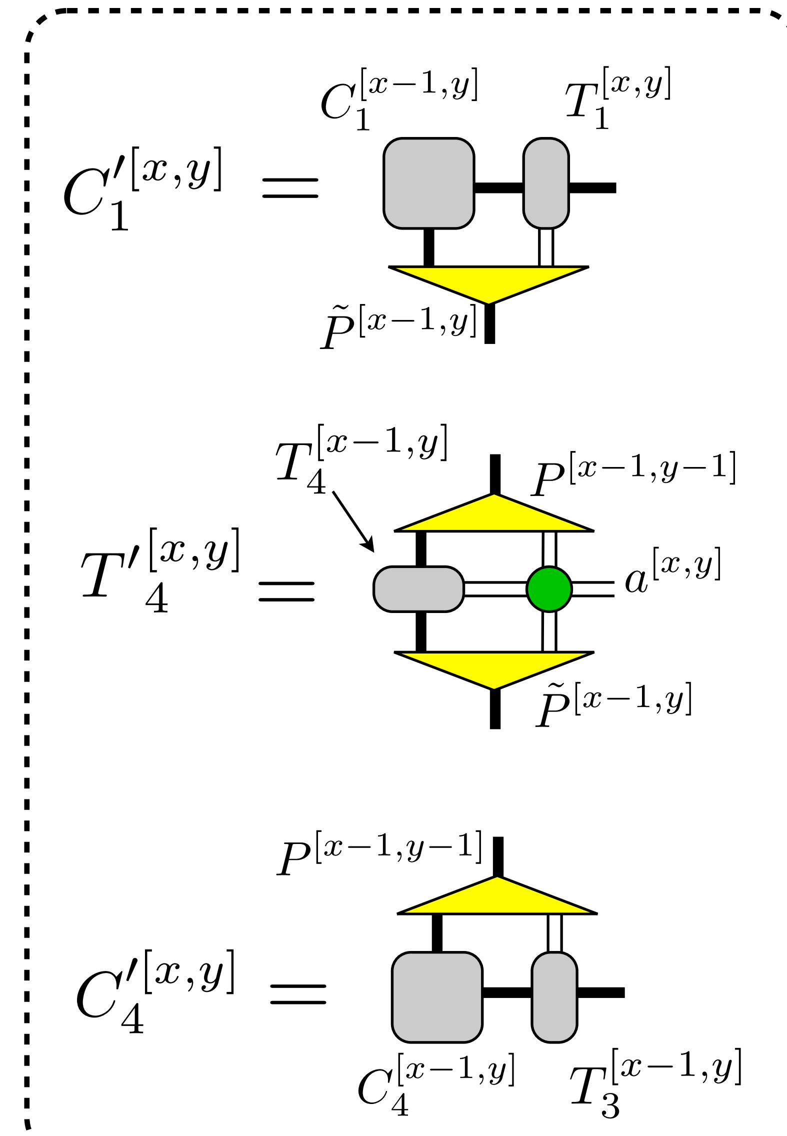


CTM with larger unit cells

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

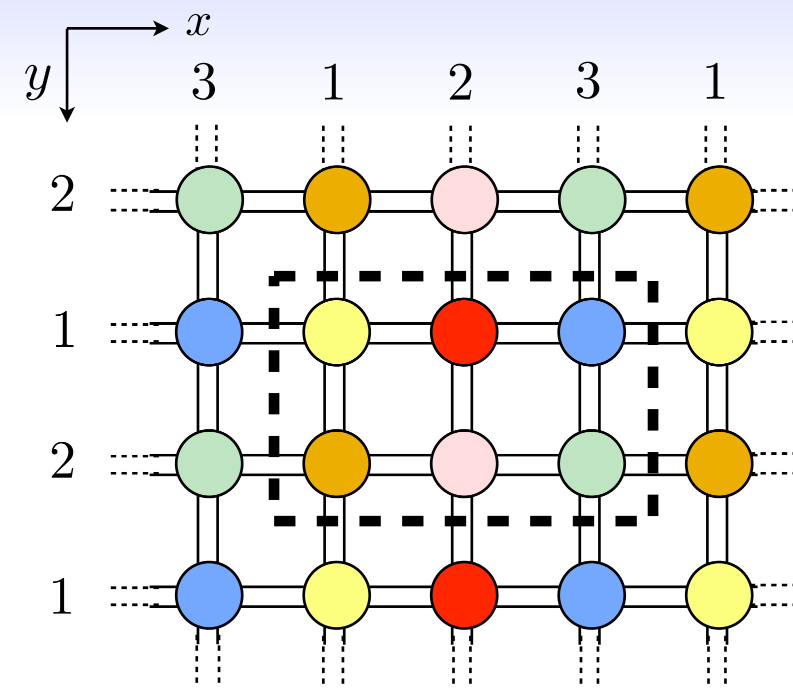




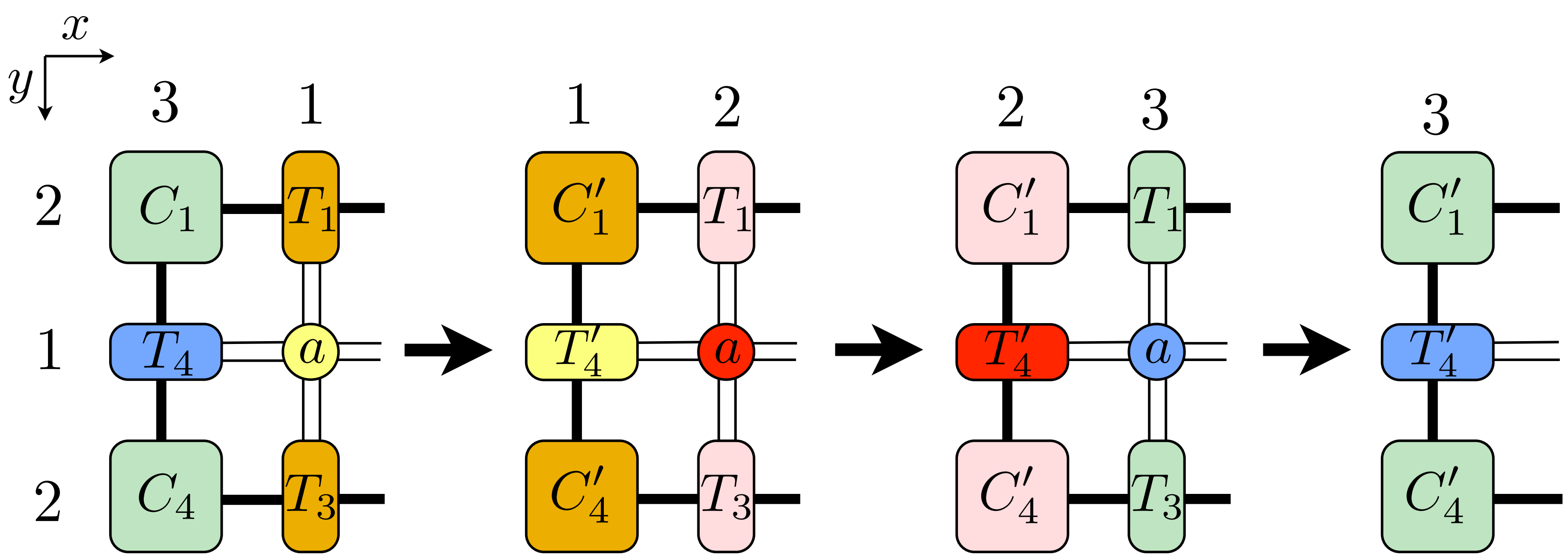


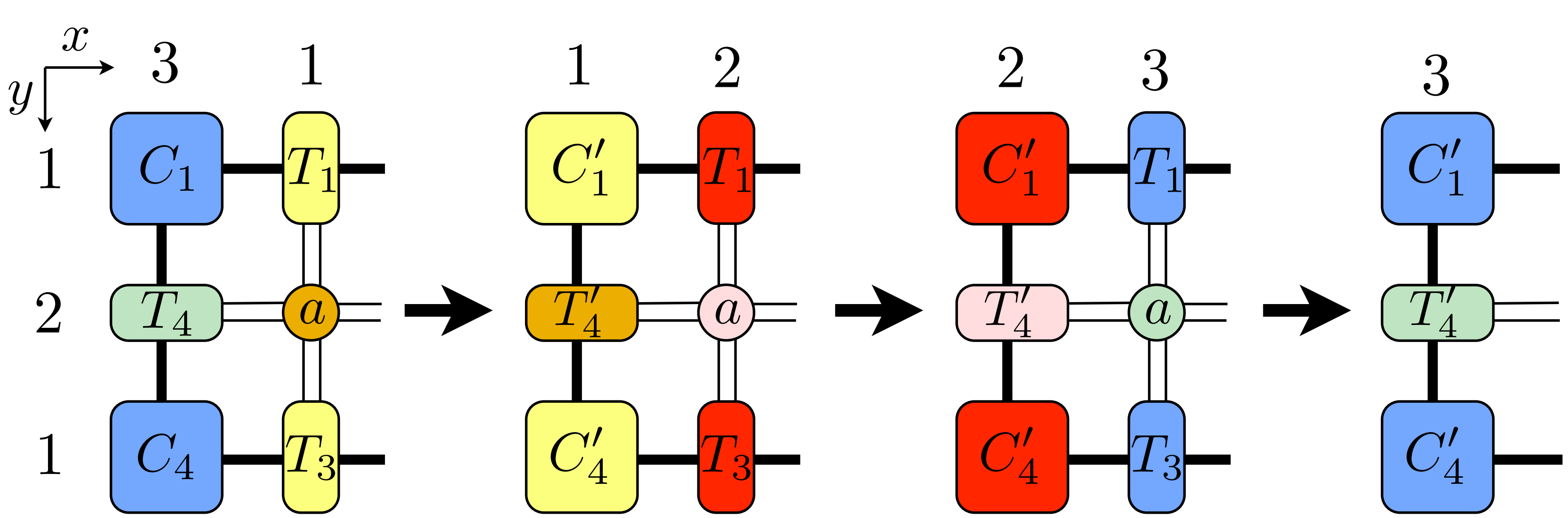
- Do for all $x \in [1, L_x]$
  - Do for all $y \in [1, L_y]$
    - Compute projectors $P^{[x-1,y]}$, $\tilde{P}^{[x-1,y]}$
  - Do for all $y \in [1, L_y]$
    - Compute updated environment tensors: $C_1'^{[x,y]}$, $C_4'^{[x,y]}$, $T_4'^{[x,y]}$

CTM with larger unit cells



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



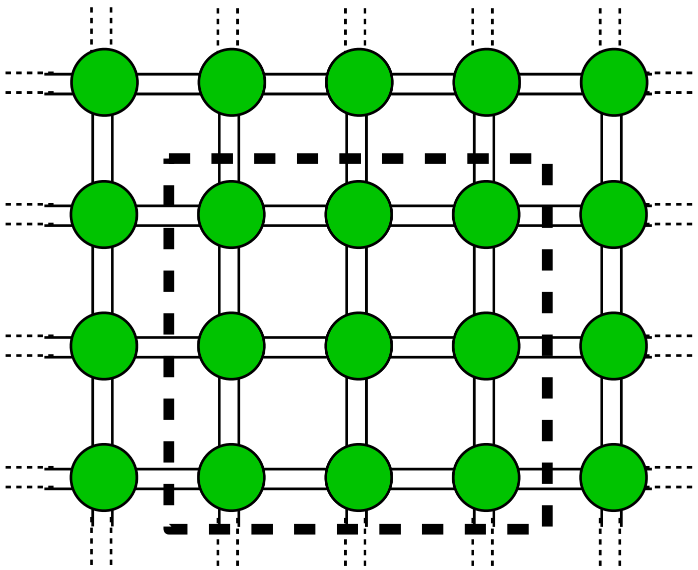


Completed left move of entire unit cell!

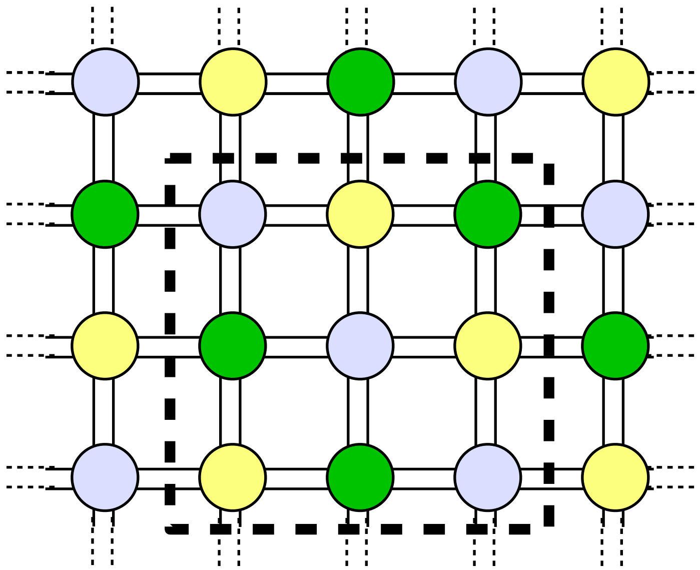
CTM with larger unit cells

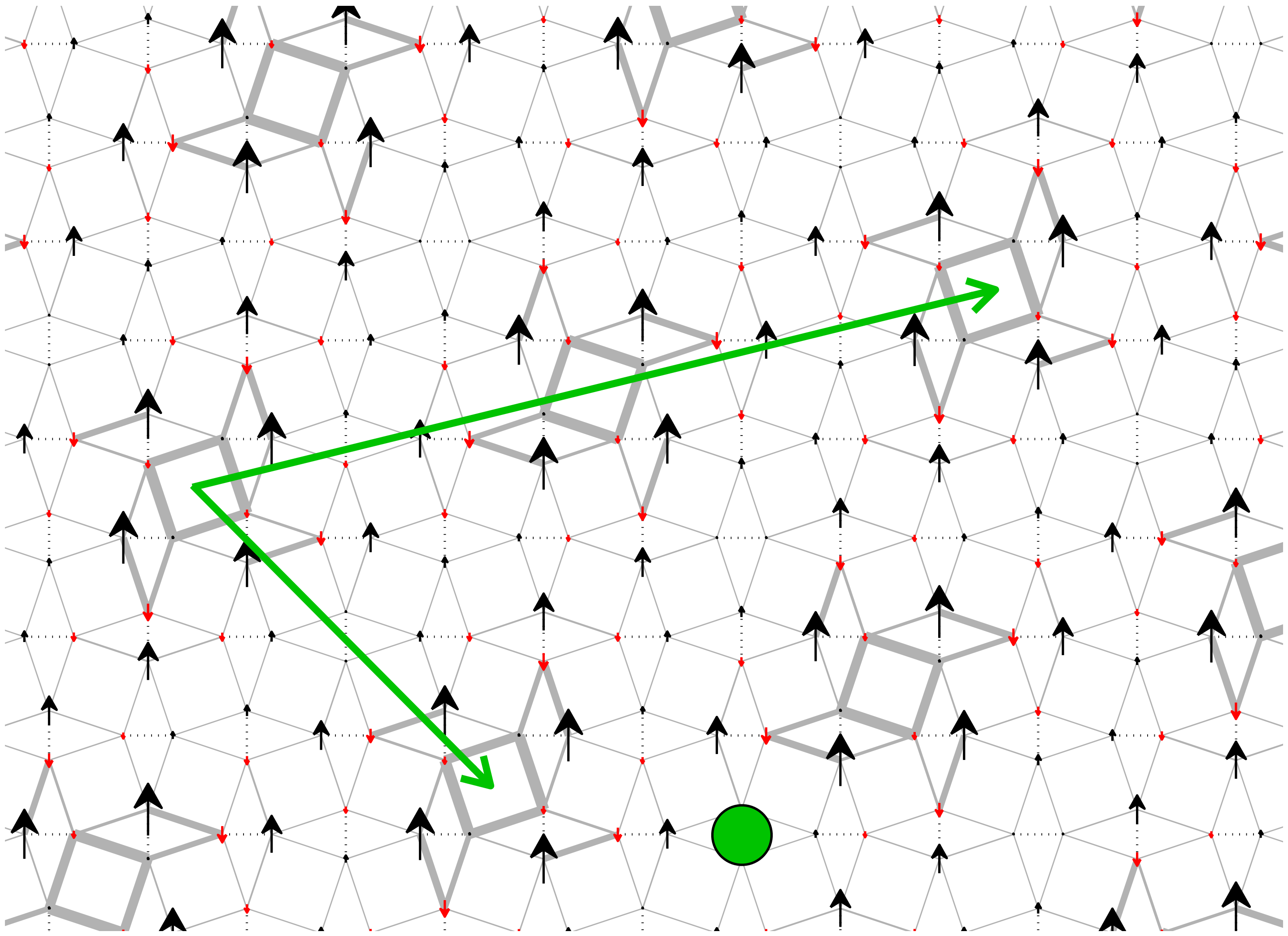
Other shapes than rectangular cell possible:

All 9 tensors different:



Only 3 different tensors:





Unit cell with 30 tensors (60 sites)

(example: Shastry-Sutherland model)

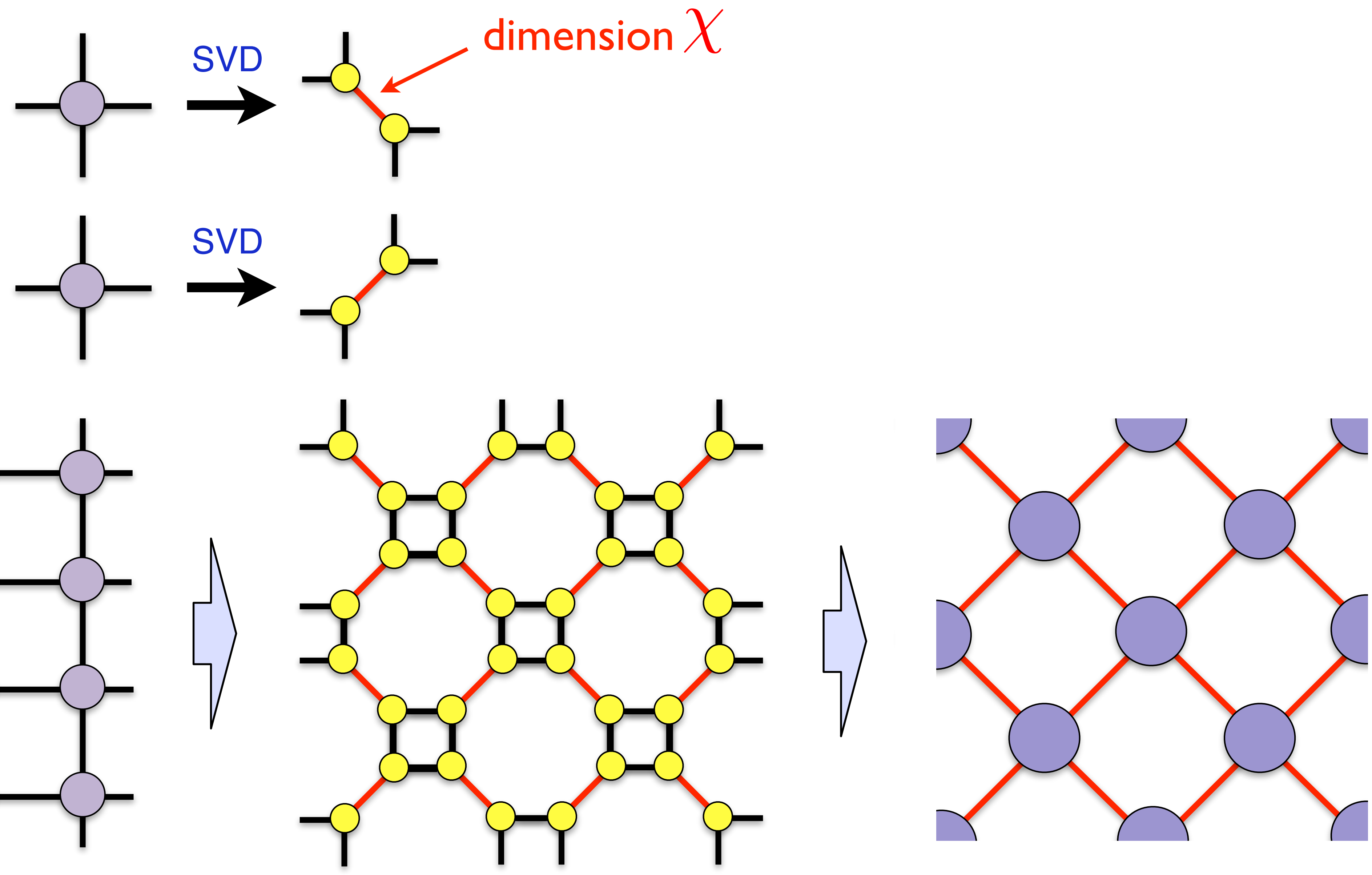
Contracting the PEPS/iPEPS using TRG

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

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

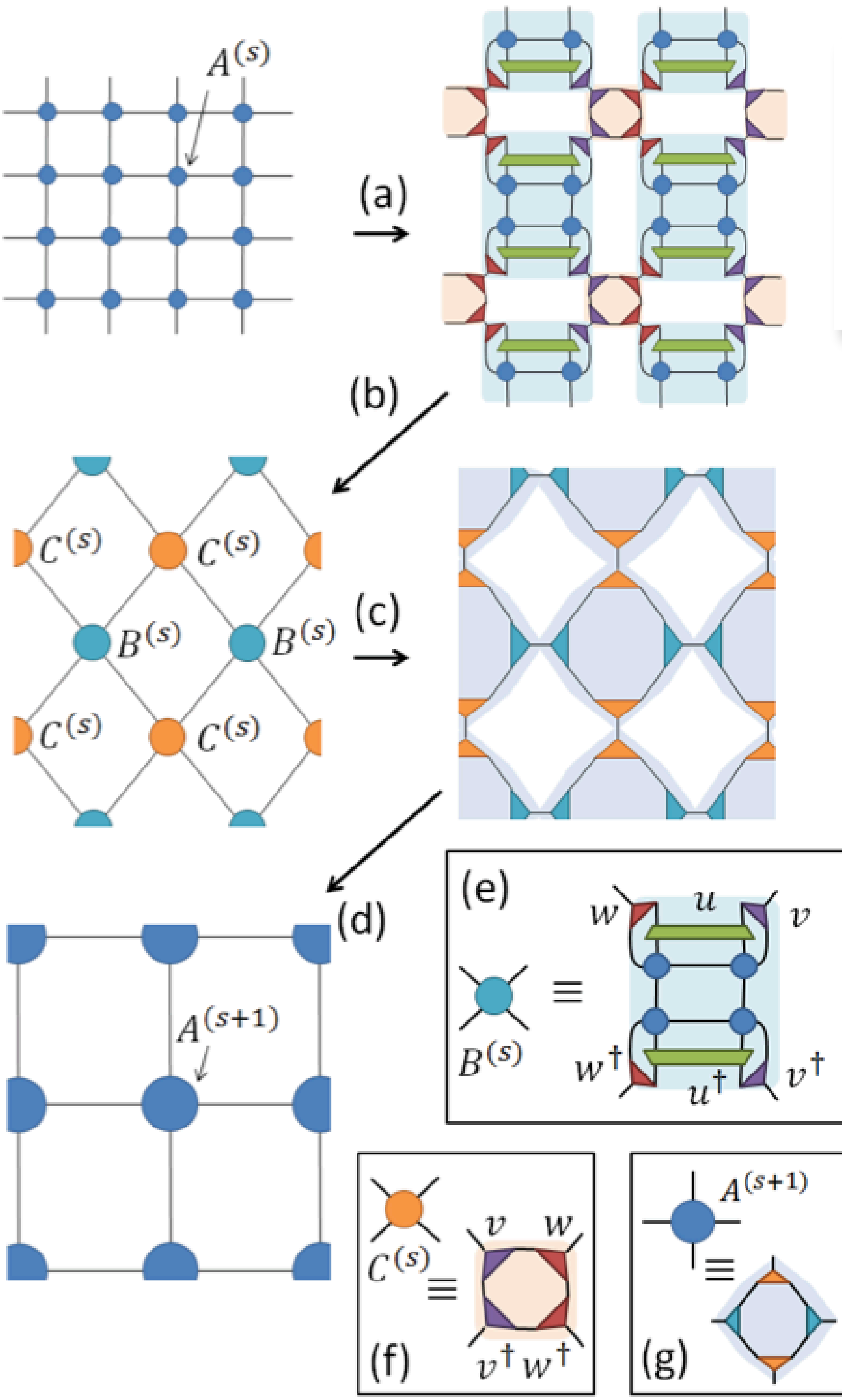
sublattice A:

sublattice B:



★ 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 approaches

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)

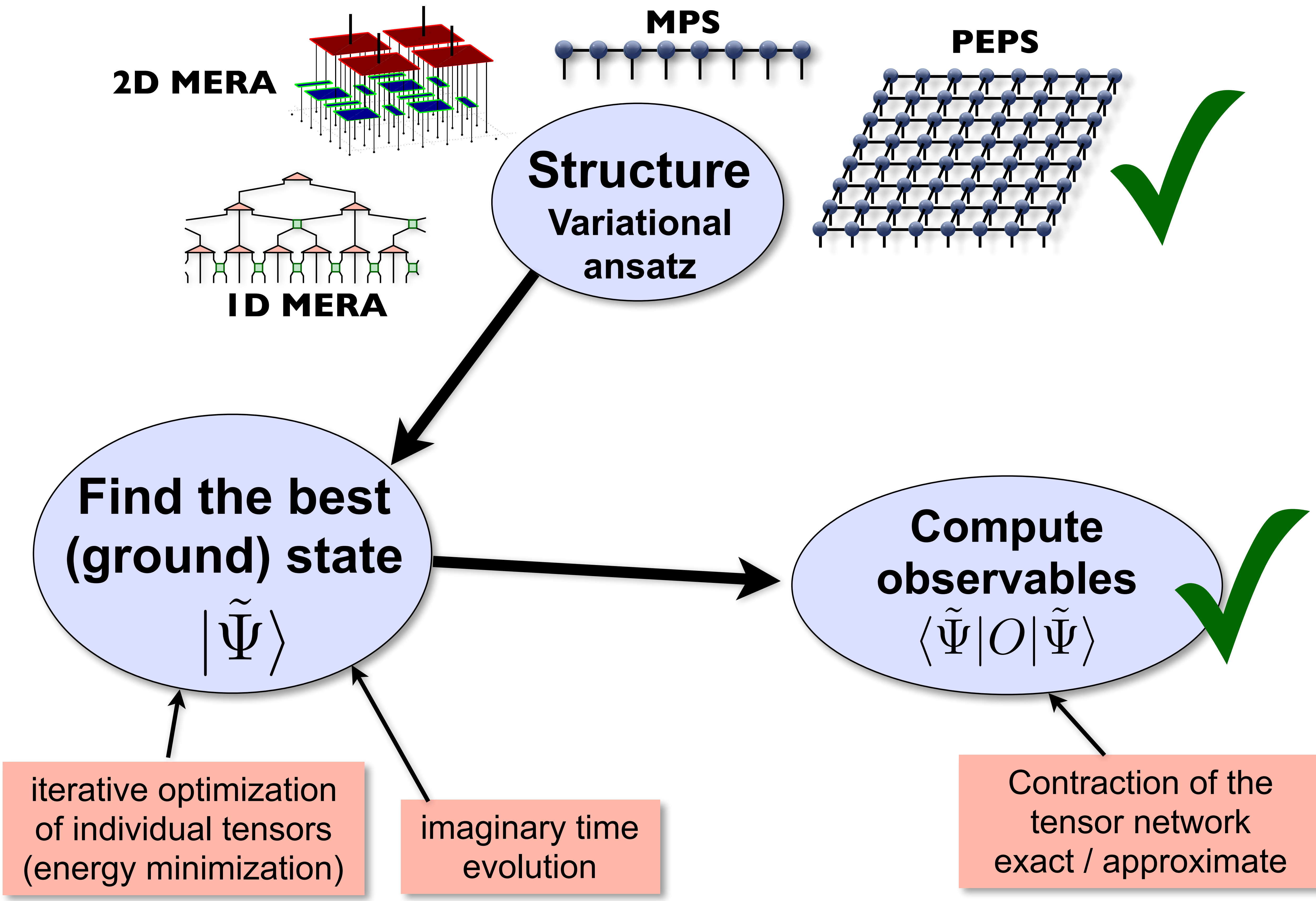
Loop-TNR:
Yang, Gu & Wen, PRL 118 (2017)

★ 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 $\chi \sim D^2$

Summary: Tensor network algorithm for ground state



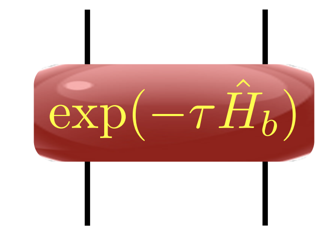
Optimization

Optimization via imaginary time evolution

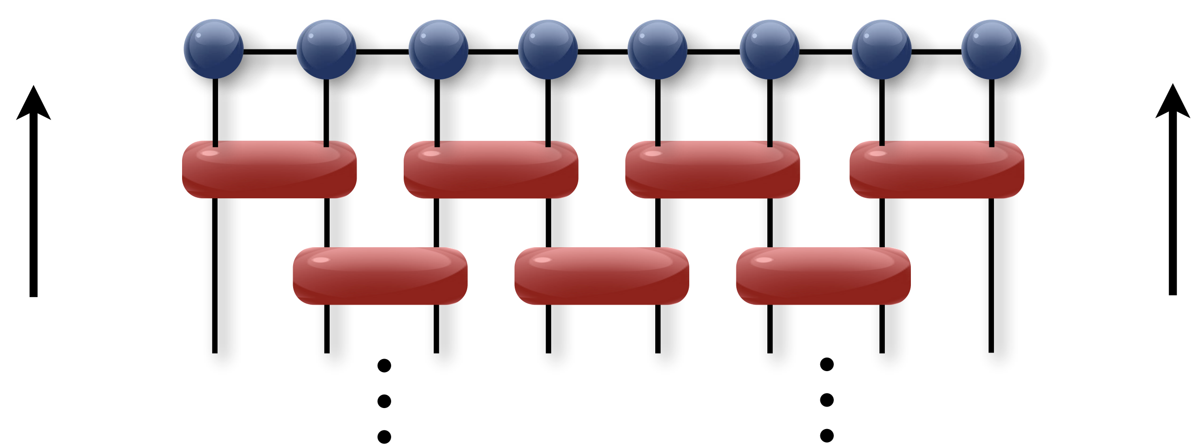
- Idea: $\exp(-\beta \hat{H})|\Psi_i\rangle \xrightarrow{\beta \to \infty} |\Psi_{GS}\rangle$

Trotter-Suzuki decomposition:

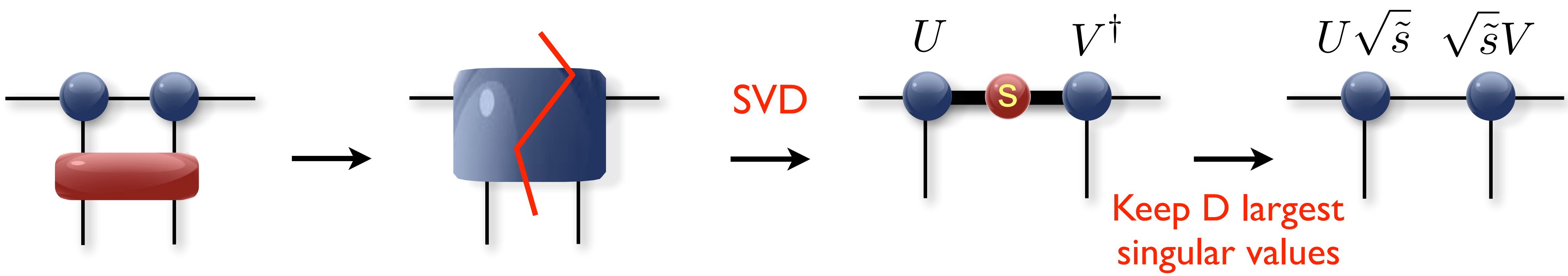
$$\exp(-\beta \hat{H}) = \exp(-\beta \sum_b \hat{H}_b) = \left(\exp(-\tau \sum_b \hat{H}_b)\right)^n \approx \left(\prod_b \exp(-\tau \hat{H}_b)\right)^n \qquad \tau = \beta/n$$



- 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

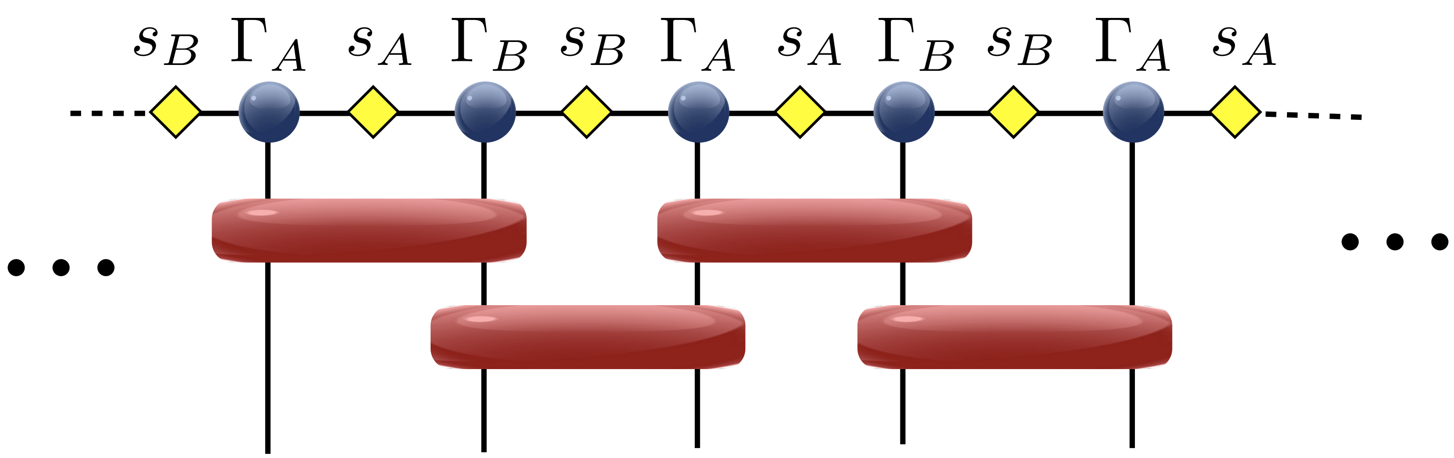
Optimization via imaginary time evolution

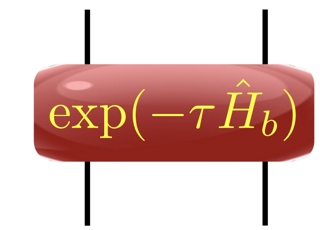
- Idea: $\exp(-\beta \hat{H})|\Psi_i\rangle \xrightarrow{\beta \to \infty} |\Psi_{GS}\rangle$

Trotter-Suzuki decomposition:

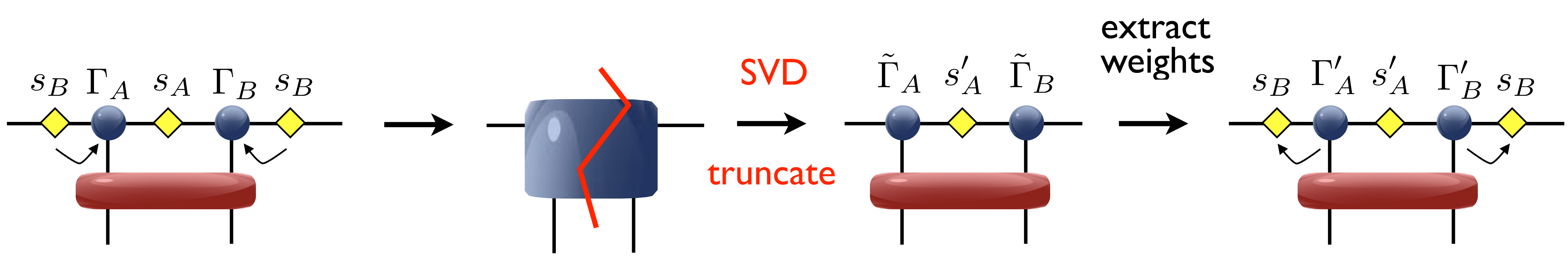
$$\exp(-\beta \hat{H}) = \exp(-\beta \sum_b \hat{H}_b) = \left(\exp(-\tau \sum_b \hat{H}_b)\right)^n \approx \left(\prod_b \exp(-\tau \hat{H}_b)\right)^n \qquad \tau = \beta/n$$

- 1D:



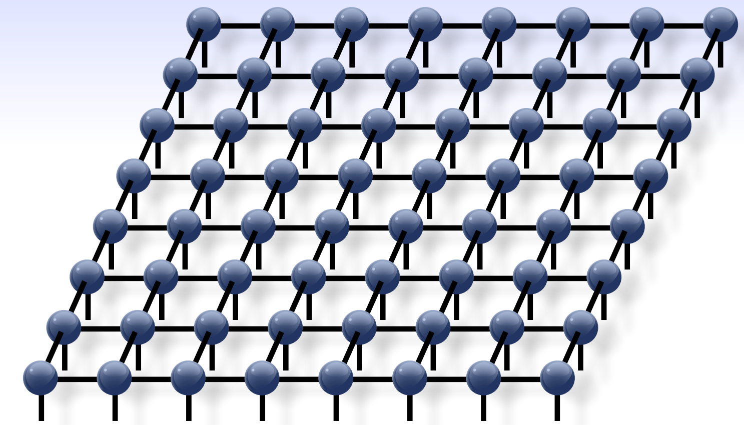


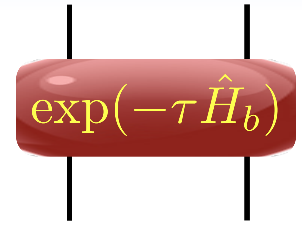
- At each step: apply a two-site operator to a bond and truncate bond back to *D*



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 (SVD)

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

Jordan et al, PRL 101 (2008)

★ Take the full wave function into account for truncation

★ optimal, but computationally more expensive

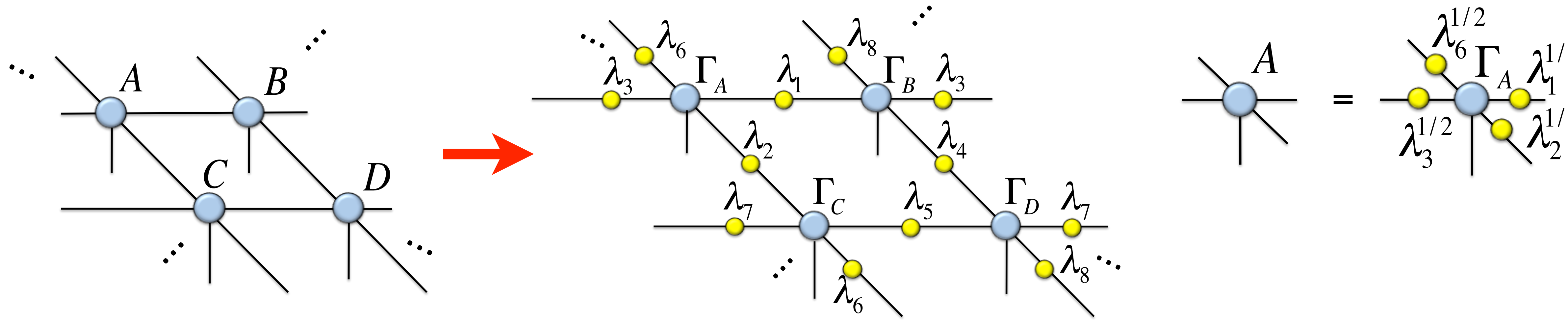
★ Fast-full update [Phien et al, PRB 92 (2015)]

Cluster update Wang, Verstraete, arXiv:1110.4362 (2011)

Optimization: simple update

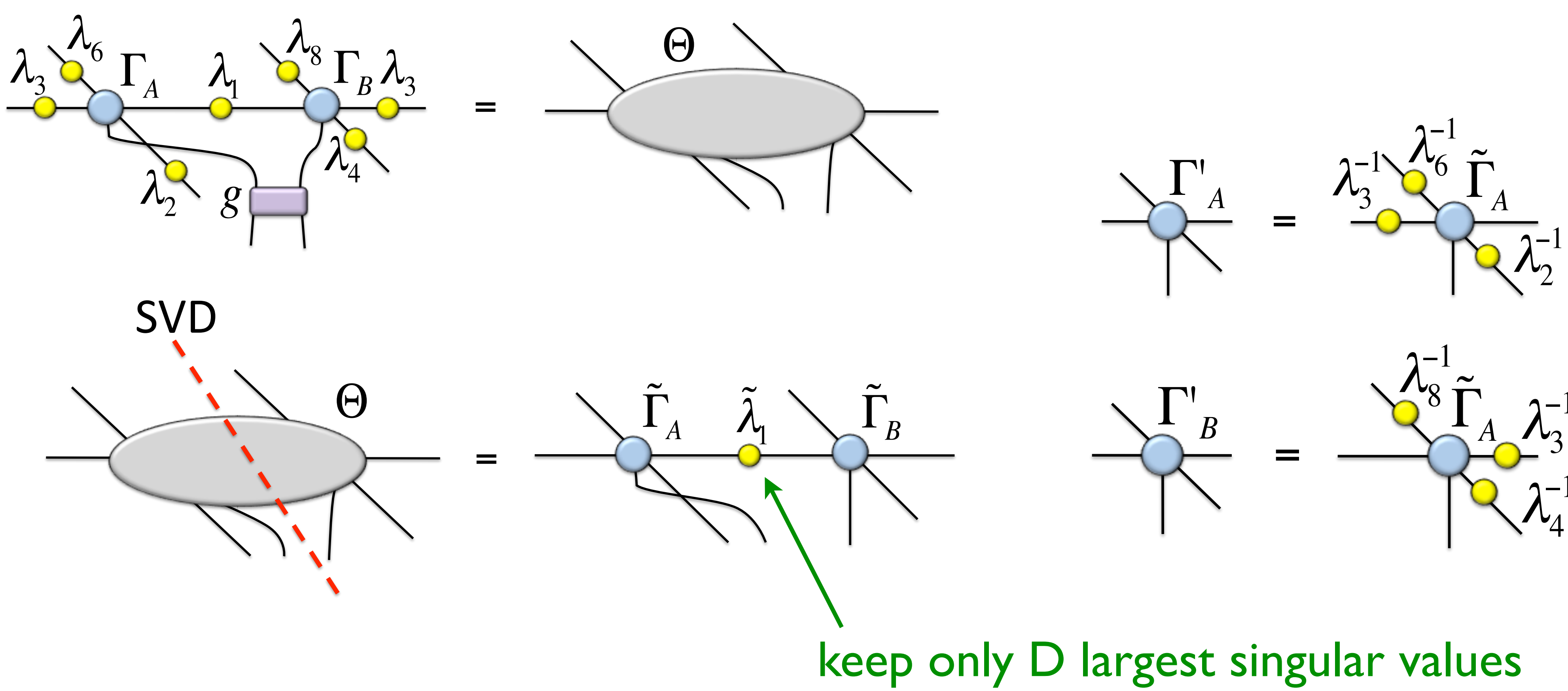
Jiang, et al., PRL 101, 090603 (2008)

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



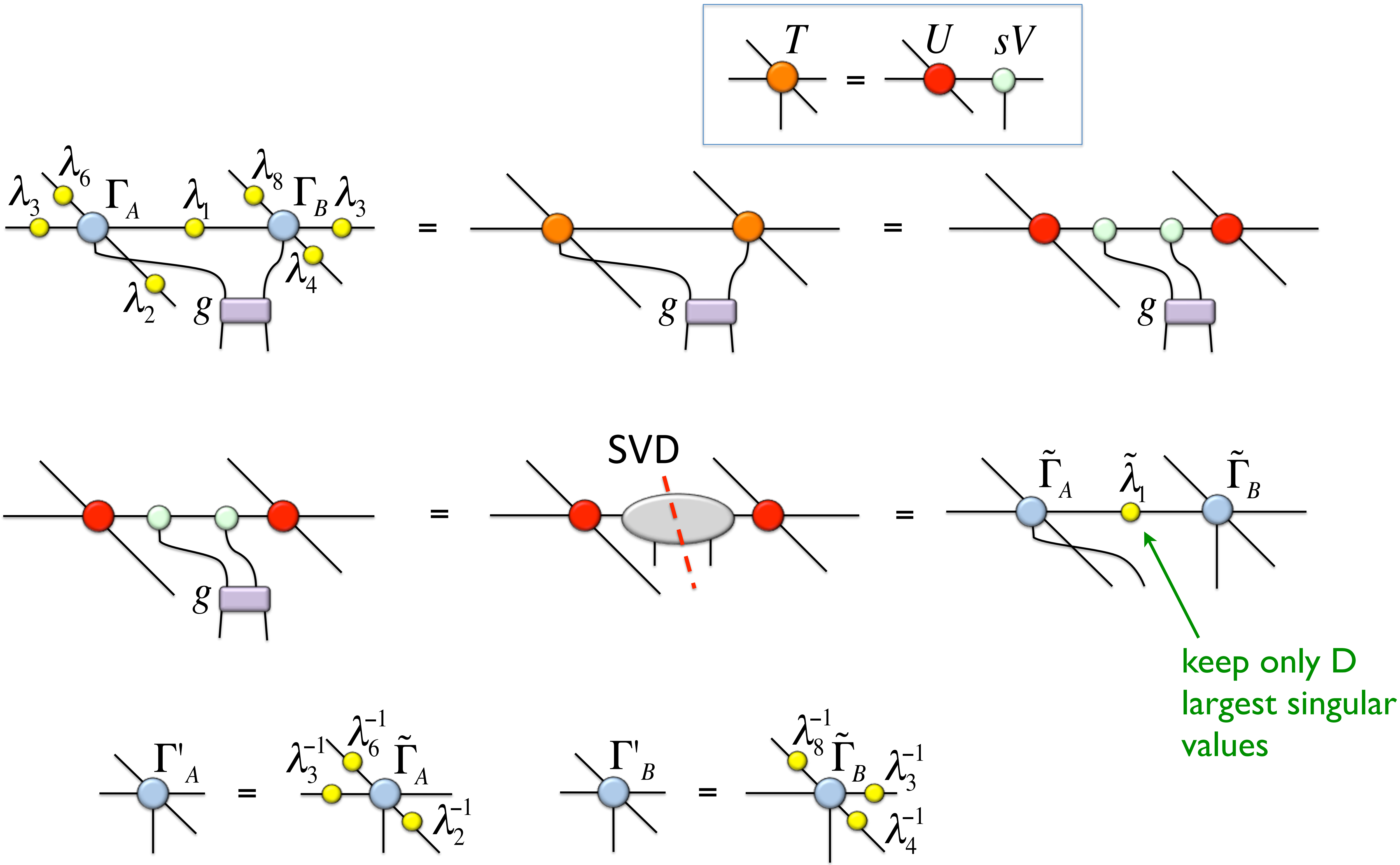
- Update works like in 1D with iTEBD (infinite time-evolving block decimation)

G. Vidal, PRL 91, 147902 (2003)



Trick to make it cheaper

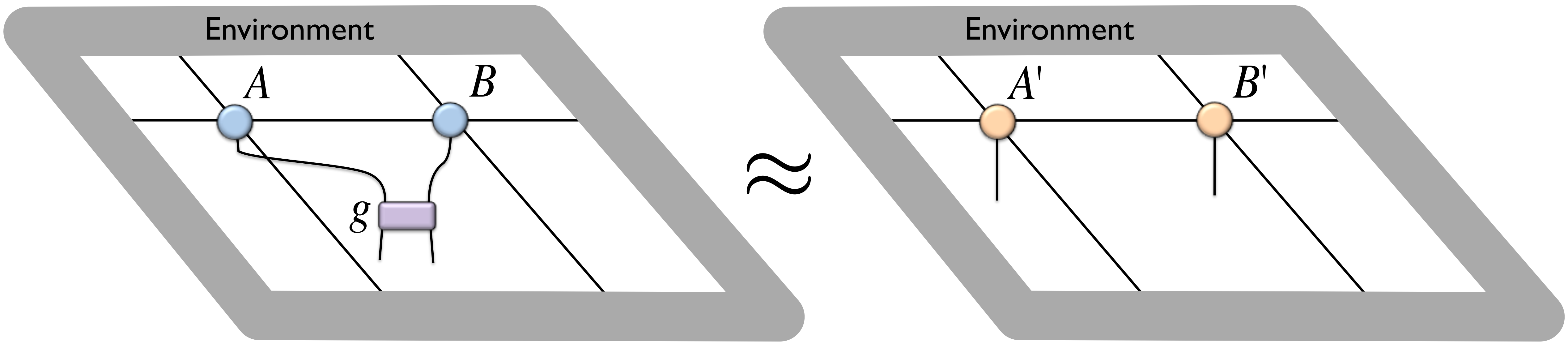
- Idea: Split off the part of the tensor which is updated



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



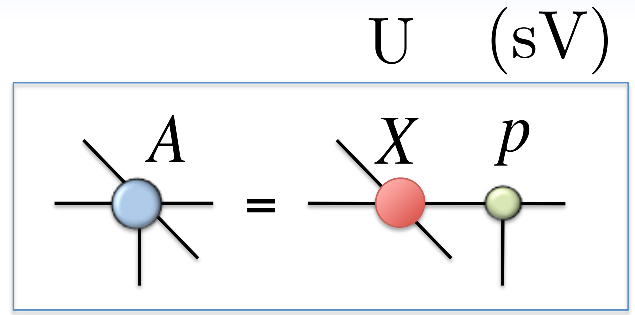
$$|\tilde{\Psi}\rangle = g|\Psi\rangle \quad \approx \quad |\Psi'\rangle$$

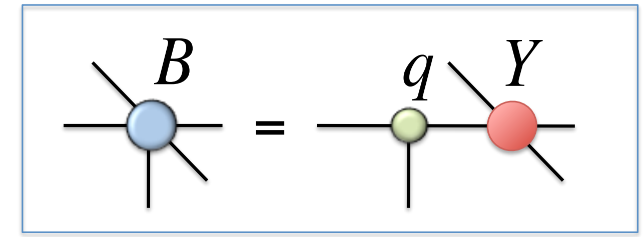
- Minimize $\quad || \, |\tilde{\Psi}\rangle - |\Psi'\rangle \, ||^2 = \langle\tilde{\Psi}|\tilde{\Psi}\rangle + \langle\Psi'|\Psi'\rangle - \langle\tilde{\Psi}|\Psi'\rangle - \langle\Psi'|\tilde{\Psi}\rangle$

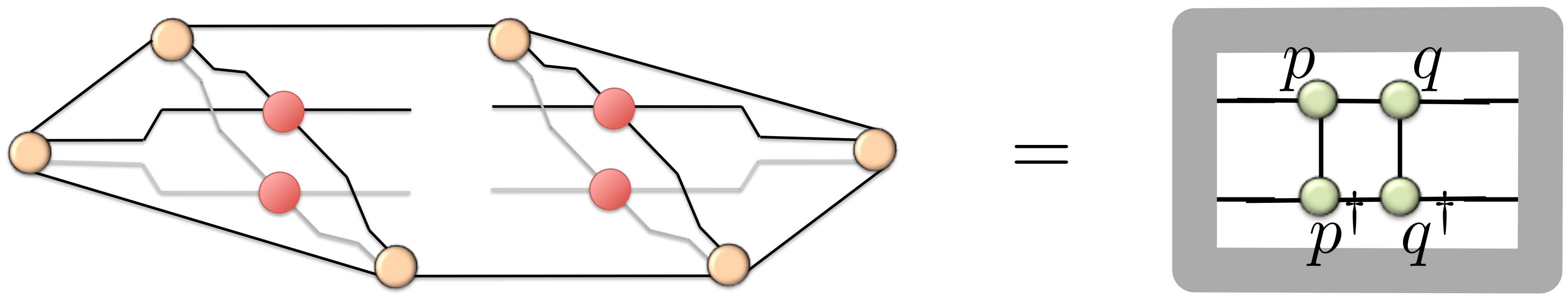
- Iteratively / CG / Newton / ...

Full-update: details

- Split off the part of the tensor which is updated





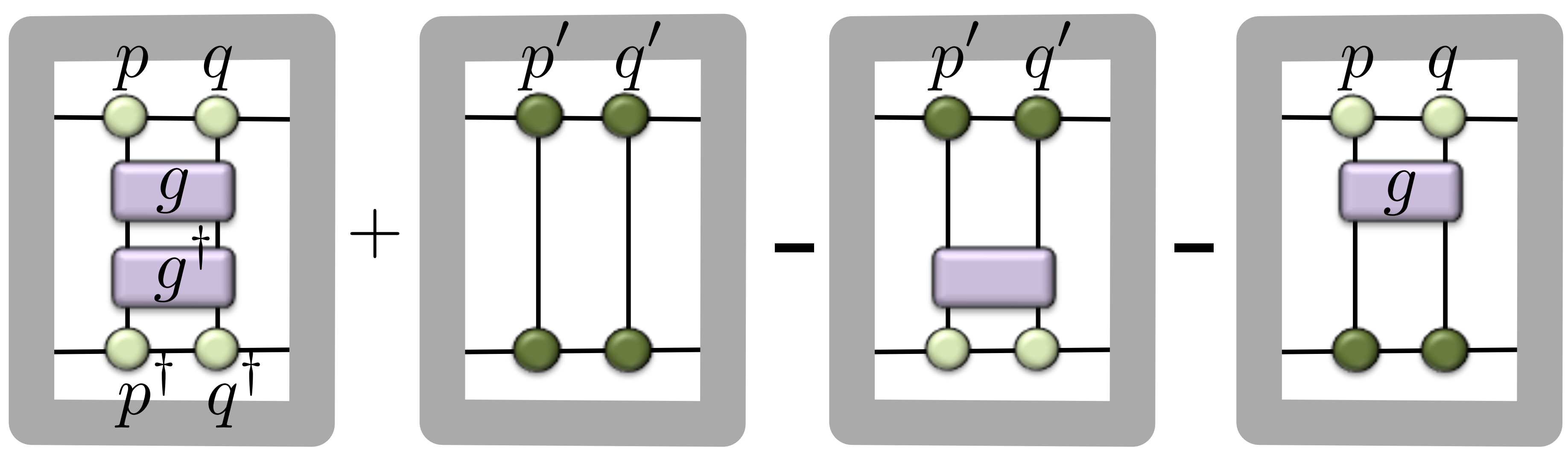


Environment of p and q tensors

$|\tilde{\Psi}\rangle = g|\Psi(p,q)\rangle \approx |\Psi'(p',q')\rangle$ find new p', and q' to minimize: $|| \; |\tilde{\Psi}\rangle - |\Psi'\rangle \; ||^2$

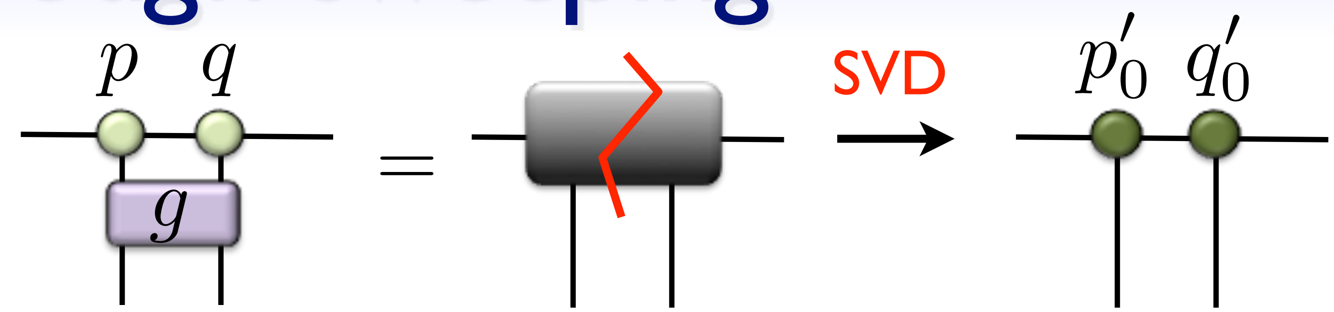
$$d(p',q') = \langle\tilde{\Psi}|\tilde{\Psi}\rangle + \langle\Psi'|\Psi'\rangle - \langle\tilde{\Psi}|\Psi'\rangle - \langle\Psi'|\tilde{\Psi}\rangle$$

"Cost-function"



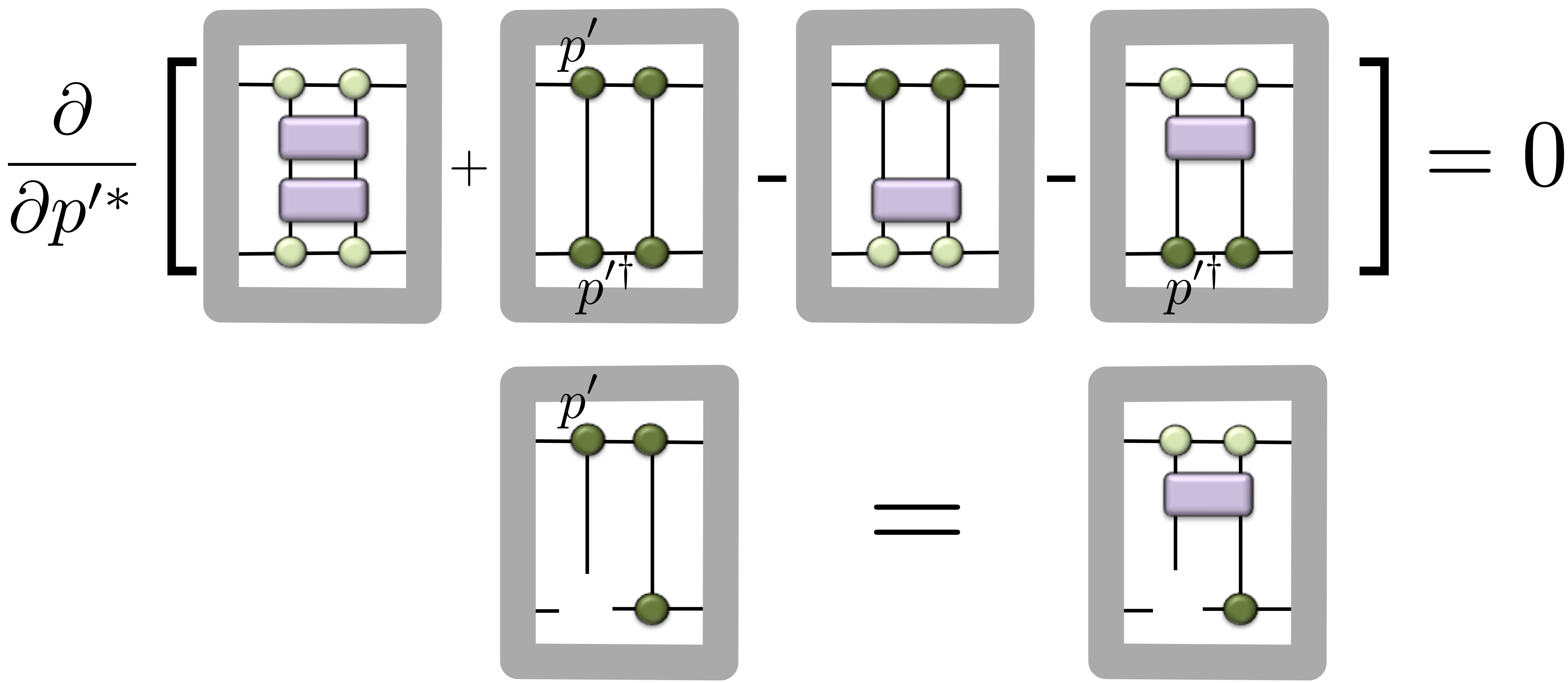
Finding p' and q' through sweeping

- Initial guess with SVD:



- Keep q' fixed and optimize with respect to p'

$$\frac{\partial}{\partial p'^*} d(p', q') = 0$$

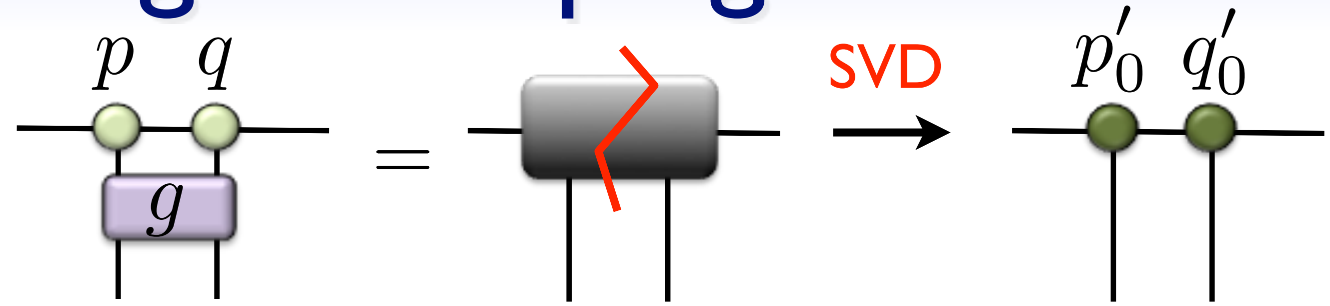


- Solve linear system:

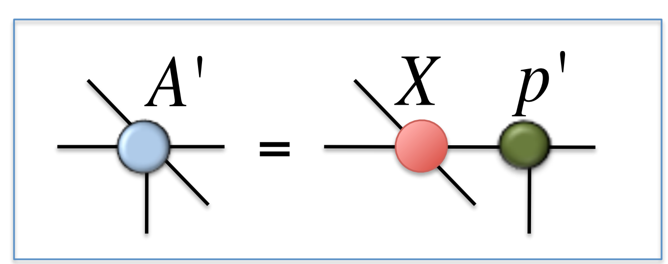
$$M p' = b \quad \rightarrow \quad \textbf{new p'}$$

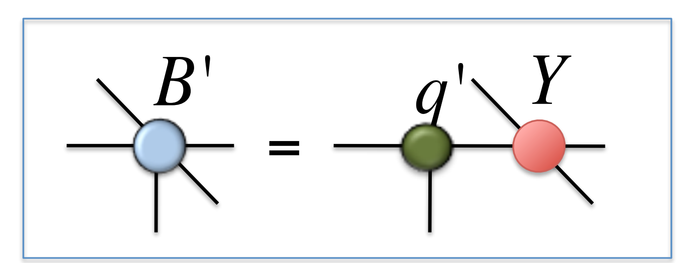
Finding p' and q' through sweeping

- Initial guess with SVD:



- Keep q' fixed and optimize with respect to p': $\frac{\partial}{\partial p'^*} d(p', q') = 0$
- Solve linear system: $Mp' = b \rightarrow$ **new p'**
- Keep p' fixed and optimize with respect to q': $\frac{\partial}{\partial q'^*} d(p', q') = 0$
- Solve linear system: $\tilde{M}q' = \tilde{b} \rightarrow$ **new q'**
- Repeat above until convergence in $d(p', q')$
- Retrieve full tensors again:

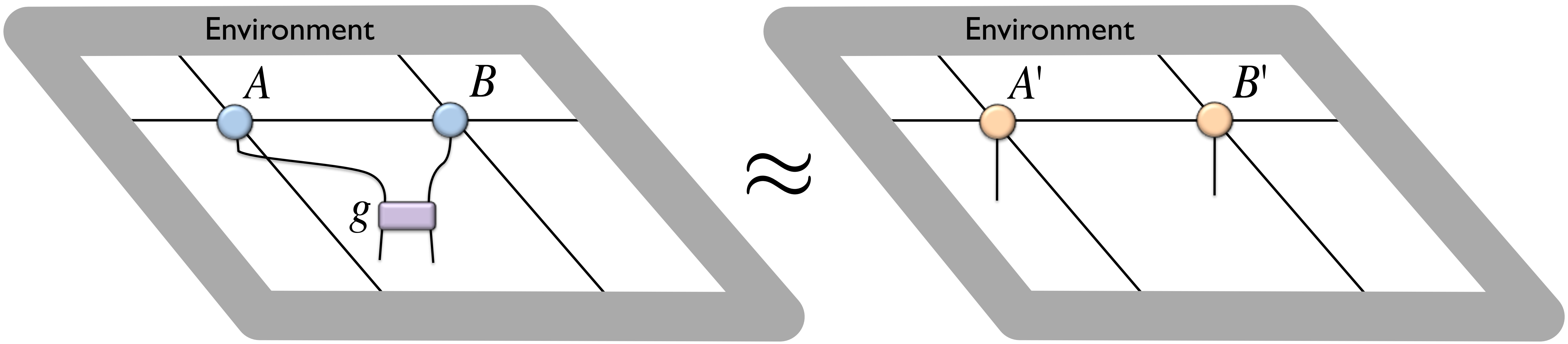




Optimization: full update

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 $|| \, |\tilde{\Psi}\rangle - |\Psi'\rangle \, ||^2 = \langle\tilde{\Psi}|\tilde{\Psi}\rangle + \langle\Psi'|\Psi'\rangle - \langle\tilde{\Psi}|\Psi'\rangle - \langle\Psi'|\tilde{\Psi}\rangle$
- 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

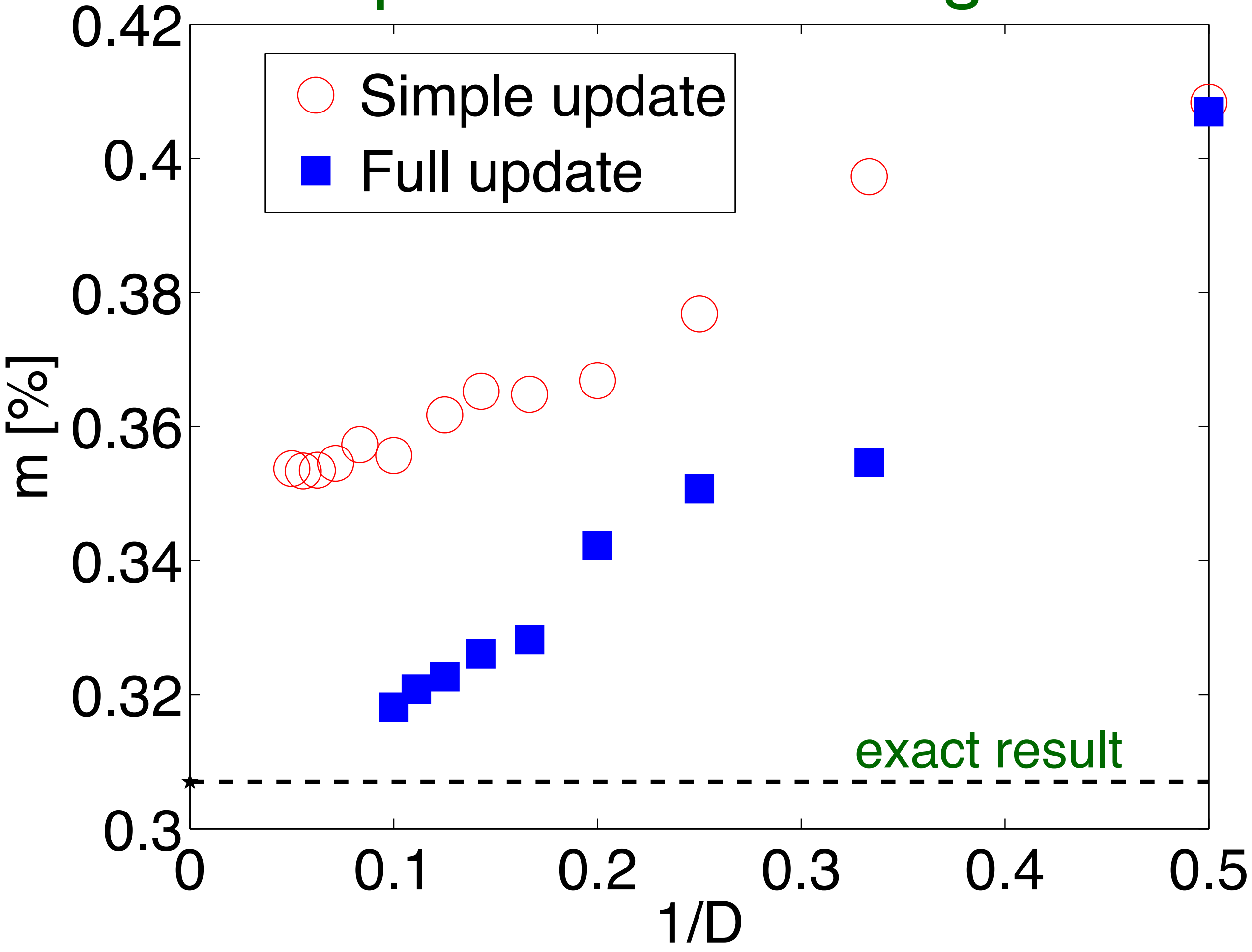
★ Cheap, but not optimal
(e.g. overestimates magnetization
in S=1/2 Heisenberg model)

full update

★ Take the full wave function into
account for truncation

★ optimal, but computationally more
expensive

Example: 2D Heisenberg model



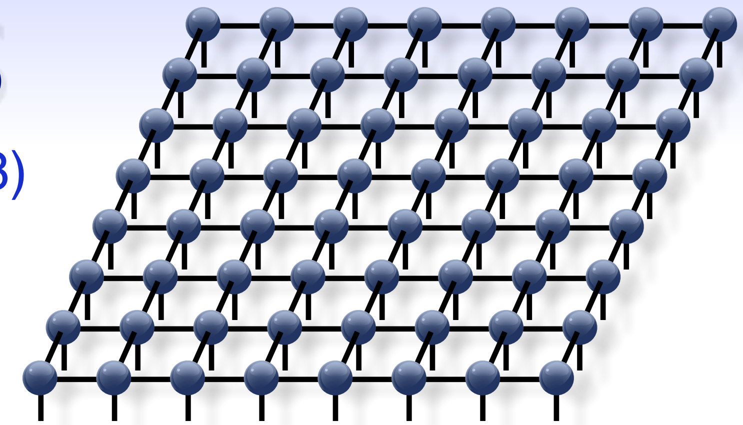
- Combine the two: Use simple update to get an initial state for the full update

- Don't compute environment from scratch but recycle previous one
→ **fast full update**

Phien, Bengua, Tuan, PC, Orus, PRB 92 (2015)

Variational optimization for PEPS

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



1. Select one of the PEPS tensors A

2. Optimize tensor A (keeping all the others fixed) by minimizing the energy:

tensor network including all Hamiltonian terms

tensor network from norm term

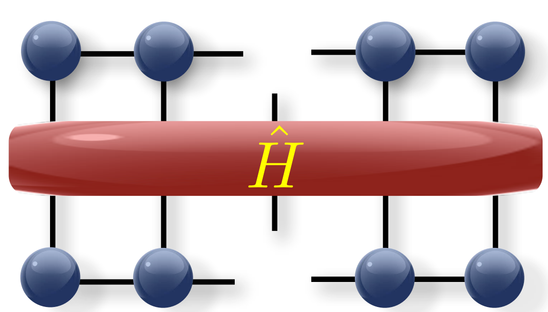
$$E = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \xrightarrow{\text{minimize}} \mathbf{H}\, x = E\, \mathbf{N}\, x$$

tensor A reshaped as a vector

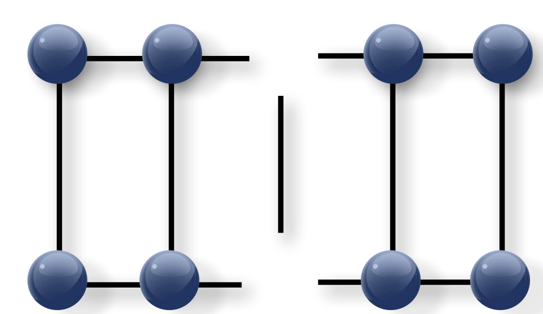
solve generalized eigenvalue problem

$\mathbf{H} =$

in 1D:

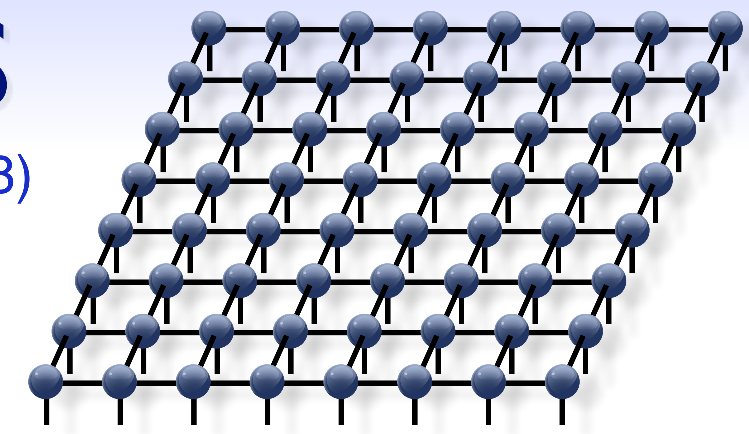


$\mathbf{N} =$



Variational optimization for PEPS

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



1. Select one of the PEPS tensors A

2. Optimize tensor A (keeping all the others fixed) by minimizing the energy:

tensor network including all Hamiltonian terms

tensor network from norm term

$$E = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \xrightarrow{\text{minimize}} \mathbf{H}\, x = E\, \mathbf{N}\, x$$

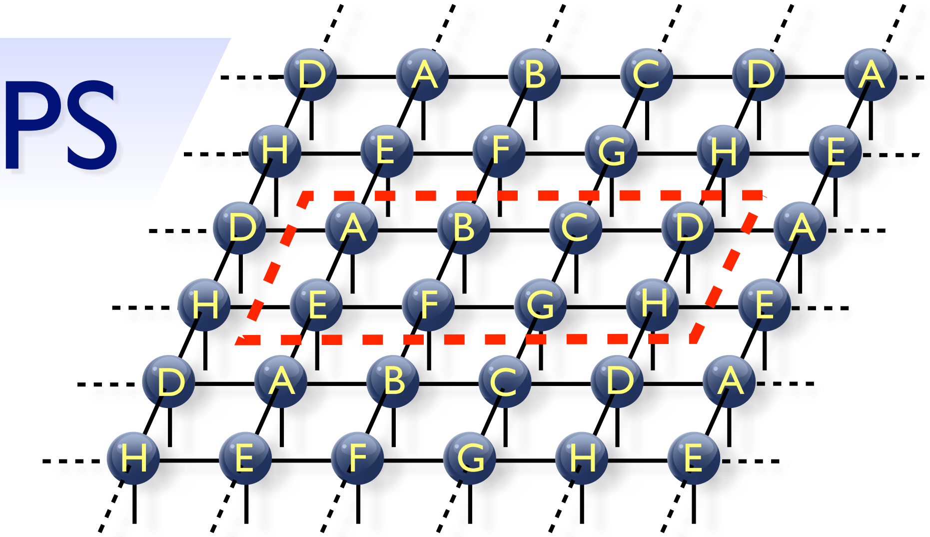
tensor A reshaped as a vector

solve generalized eigenvalue problem

3. Take the next tensor and optimize (keeping other tensors fixed)

4. Repeat 2-3 iteratively until convergence is reached

Variational optimization for iPEPS



Main challenges:

1. Need to take into account infinitely many Hamiltonian contributions

- ✦ Solution: use corner-transfer matrix method [PC, PRB 94 (2016)]
- ✦ Alternative: use "channel-environments" [Vanderstraeten et al, PRB 92; PRB 94 (2016)]
- ✦ Or: Use PEPO (similar to 3D classical) [cf. Nishino et al. Prog. Theor. Phys 105 (2001)]

2. Tensor A appears infinitely many times! (Min. problem highly non-linear)

- ✦ Take adaptive linear combination of old and new tensor [PC, PRB 94 (2016)] [see also Nishino et al. Prog. Theor. Phys 105 (2001), Gendiar et al. PTR 110 (2003)]
- ✦ Alternative: use CG approach [Vanderstraeten, Haegeman, PC, Verstraete, PRB 94 (2016)]

tensor network including all Hamiltonian terms

tensor network from norm term

$$E = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \xrightarrow{\text{minimize}} \mathbf{H}\, x = E\, \mathbf{N}\, x$$

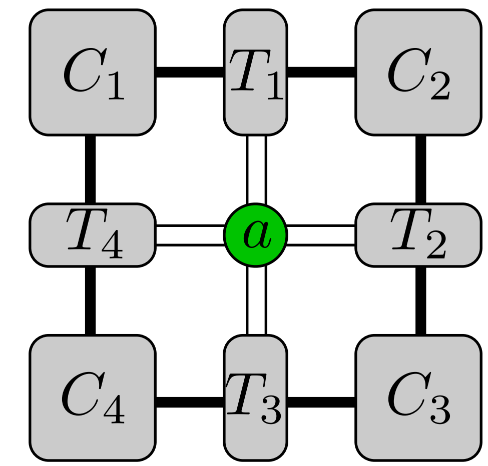
tensor A reshaped as a vector

H-environment

tensor network including all Hamiltonian terms

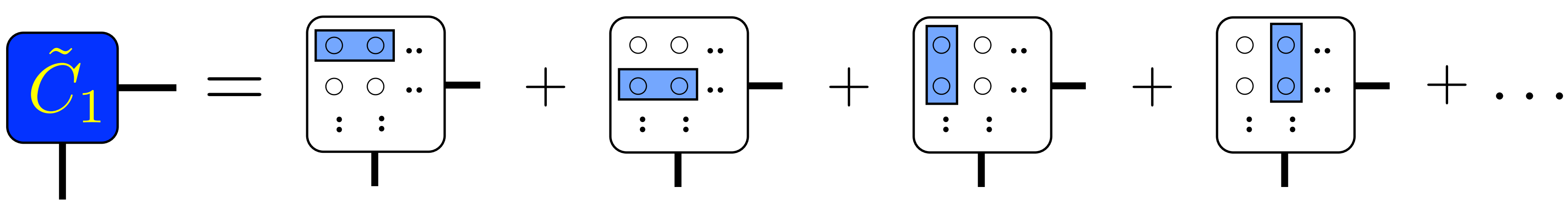
tensor network from norm

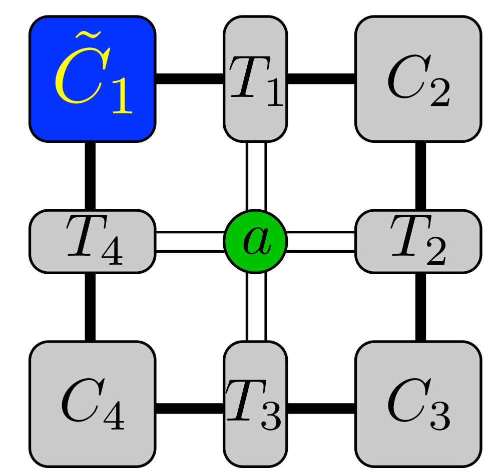
$$E = \frac{\langle\Psi|H|\Psi\rangle}{\langle\Psi|\Psi\rangle} \xrightarrow{\text{minimize}} \mathbf{H}\,x = E\,\mathbf{N}\,x$$



But how about H ??

- Need additional **H**-environment tensors:

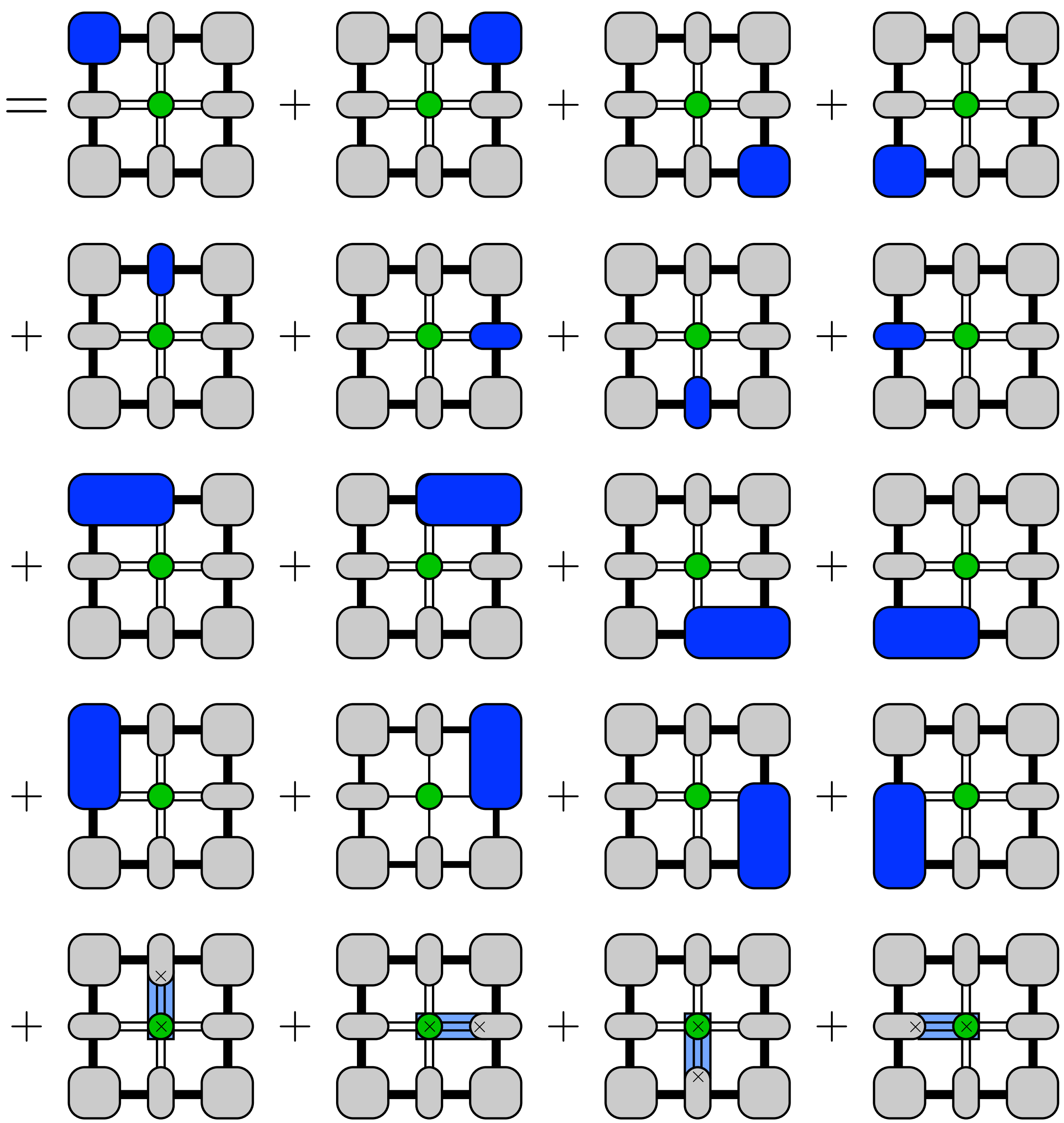




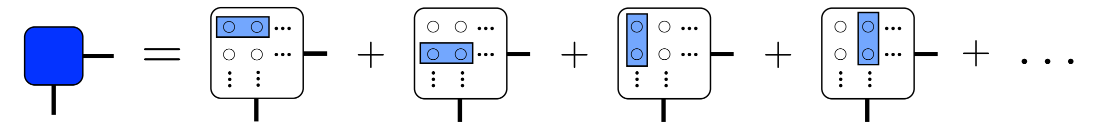
➡ taking into account all Hamiltonian contributions in the infinite upper left corner

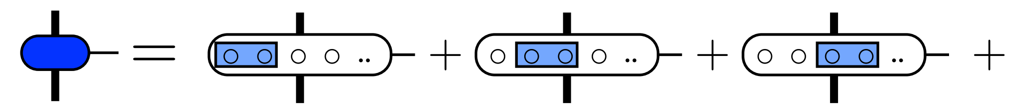
H-environment

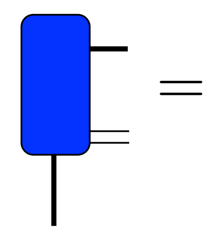
$$\langle \Psi | \hat{H} | \Psi \rangle =$$



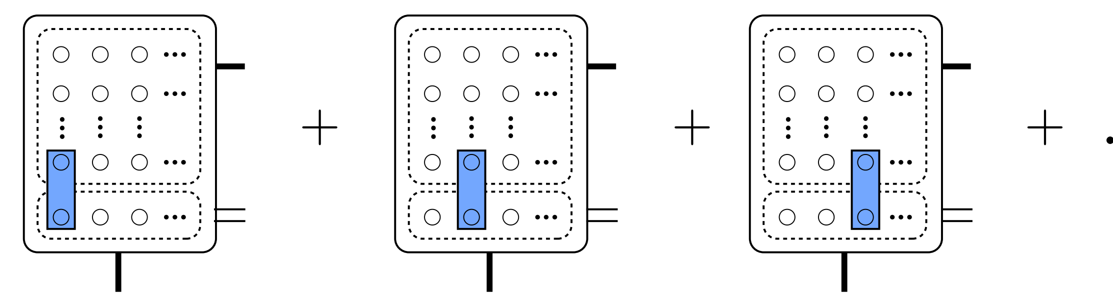
Corner terms







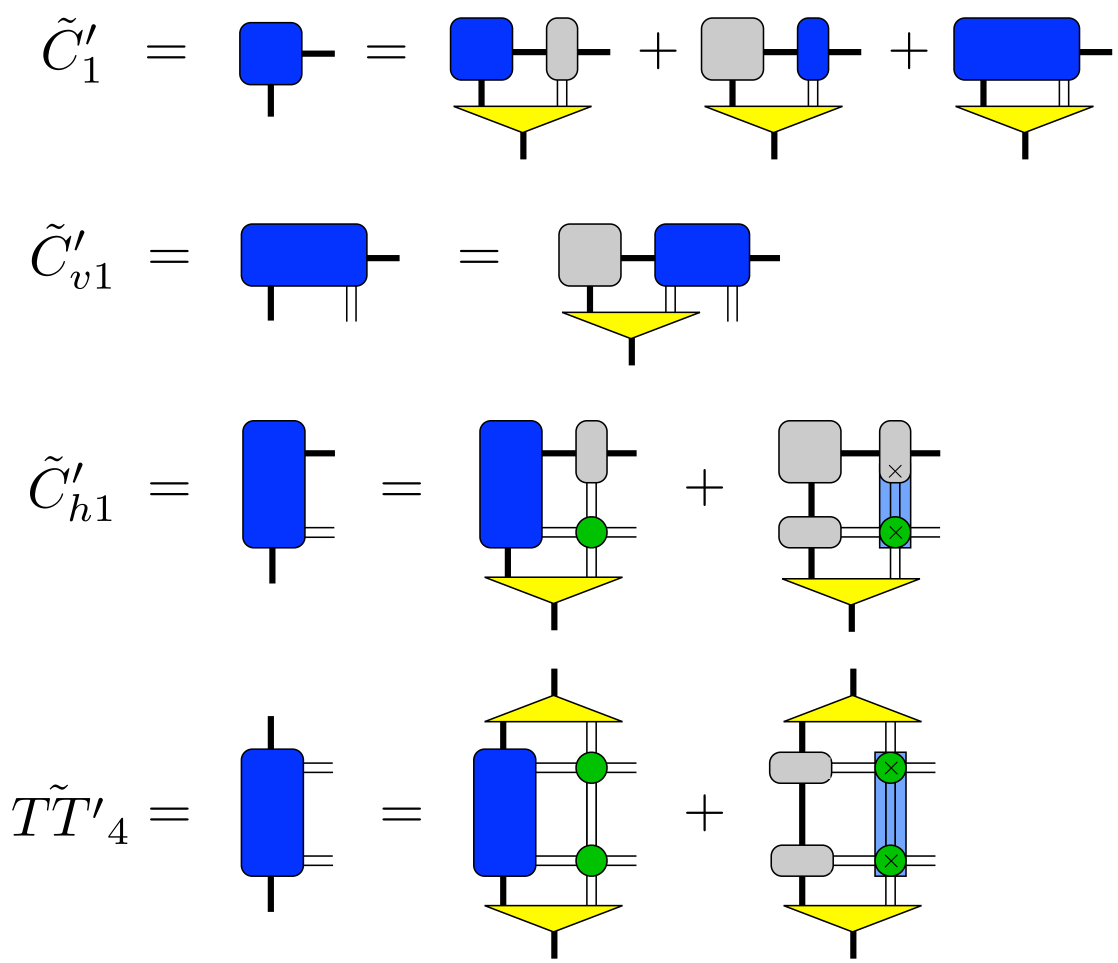
Terms between a corner and an edge tensor

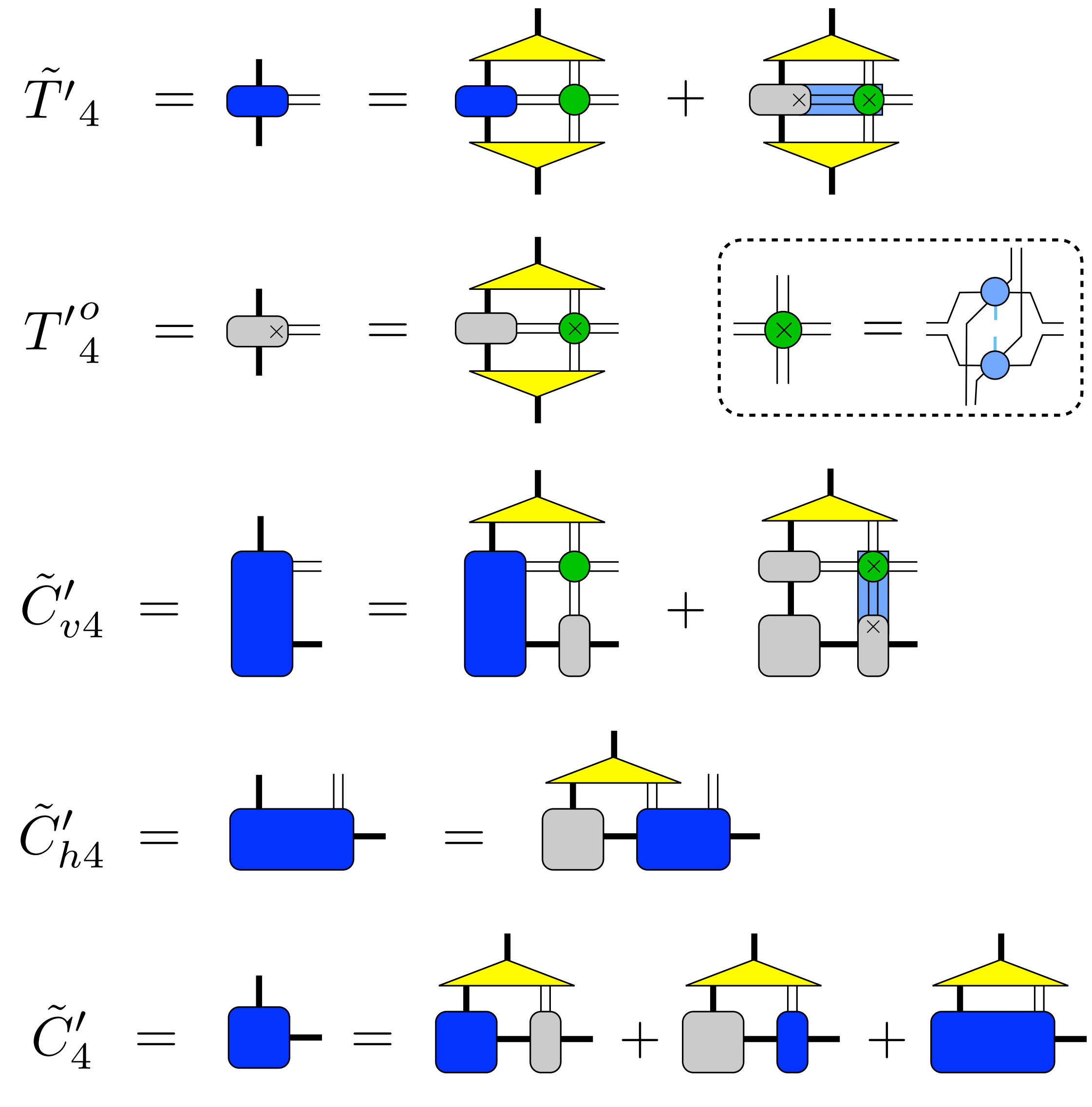


Local terms

H-environment: bookkeeping

CTM left move:

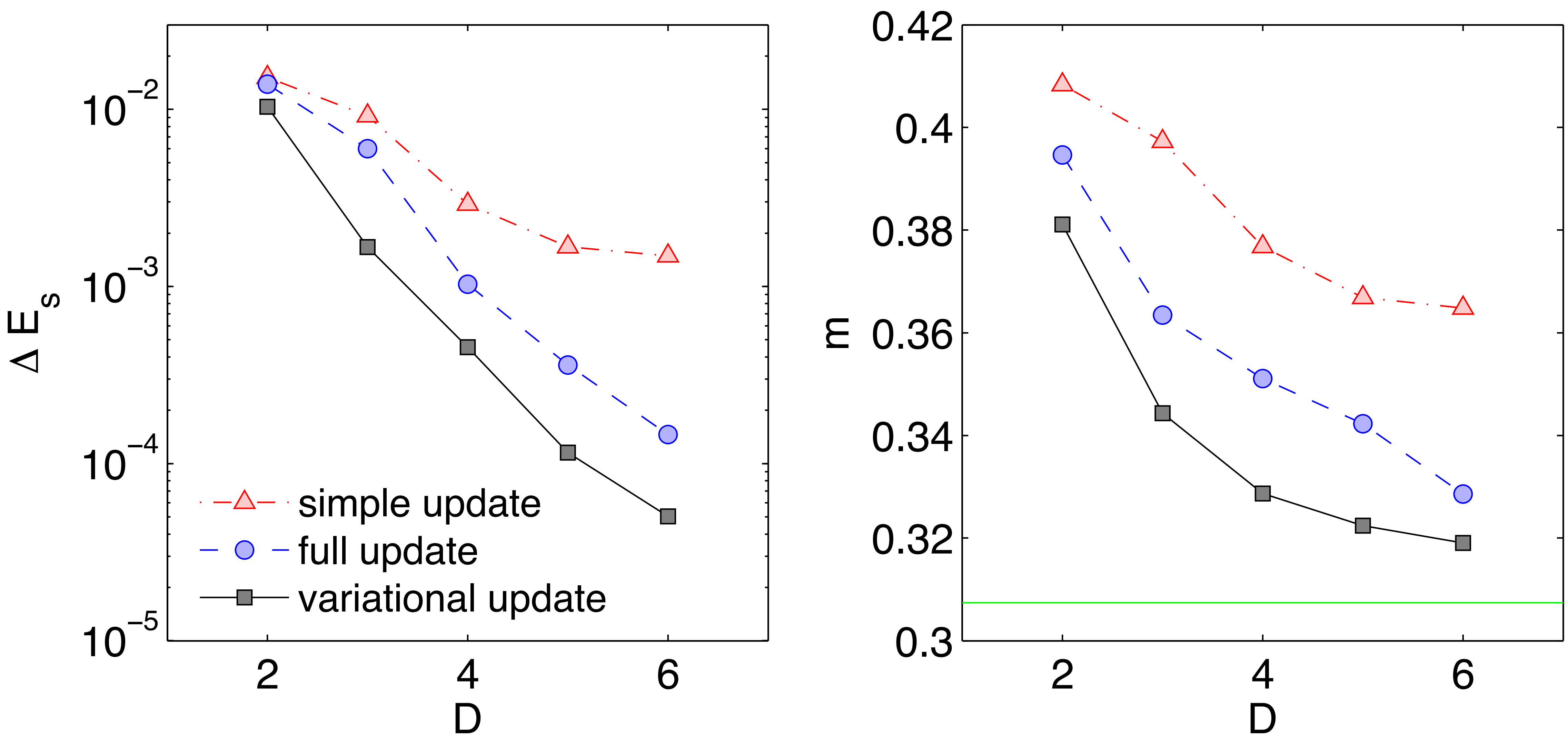




... and similarly for right-, top-, bottom-move

- We can sum up all Hamiltonian contributions in an iterative way

Comparison: Heisenberg model



- Energy and order parameter are substantially improved with the variational optimization
- 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 / variational optimization
  - ✦ DMRG-like sweeping: **higher accuracy**, similar cost as FFU
    PC, PRB 94 (2016)
  - ✦ CG-approach: **higher accuracy**, similar cost as FFU
    Vanderstraeten, Haegeman, PC, and Verstraete, PRB 94 (2016)
  - ✦ See also variational optimization in the context of 3D classical models
    Nishino et al. Prog. Theor. Phys 105 (2001), Gendiar et al. Prog. Theor. Phys 110 (2003)
  - ✦ ... more to explore...!

Summary: optimization in iPEPS

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 … efficiently using automa … or networks contraction a … nd efficient backpropaga … heat of the Ising model … the tensor renormalizati … of infinite projected ent … in 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.

DOI: 10.1103/PhysRevX.9.031041

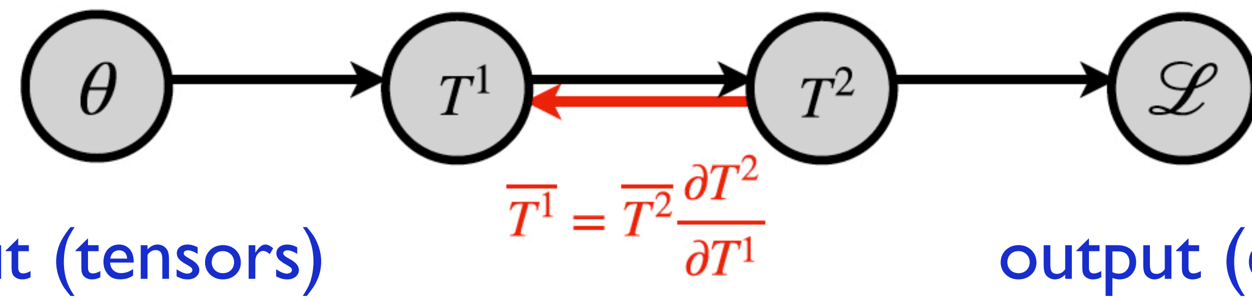
Subject Areas: Computational Physics, Condensed Matter Physics

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

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

See Juraj Hasik's talk on Thursday!

Summary: Tensor network algorithms (ground state)