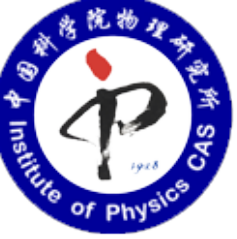


中国科学院物理研究所
Institute of Physics Chinese Academy of Sciences

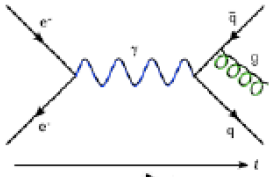
Tensor Network Contraction Schemes

Hai-Jun Liao(廖海军)
IOP, CAS
navyphysics@iphy.ac.cn

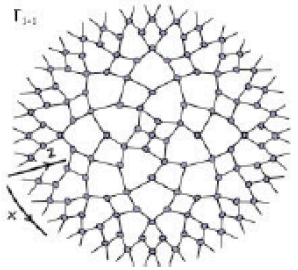
5/6/2019, Song Shan Lake

Tensor Network Everywhere

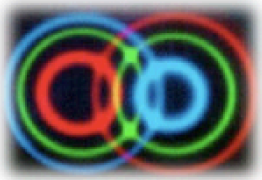
HEP and lattice gauge theories



Quantum gravity, string theory and AdS/CFT

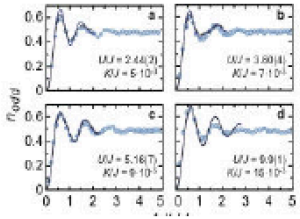


Quantum information and computation



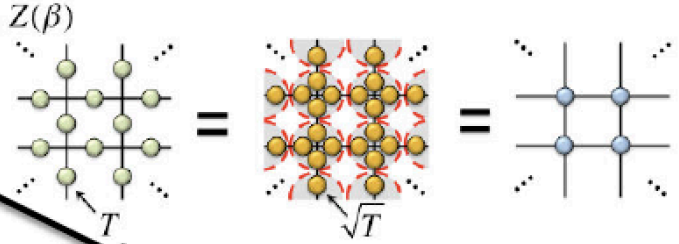
Strongly correlated systems

Quantum simulations

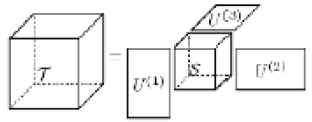


Entanglement and Tensor Networks

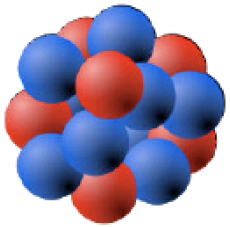
Classical statistical mechanics



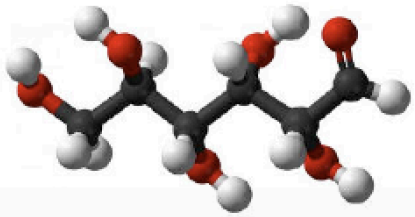
Numerical tensor calculus



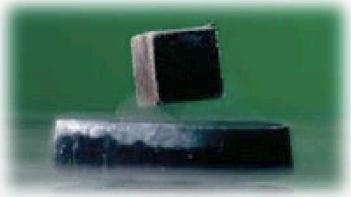
Nuclear physics



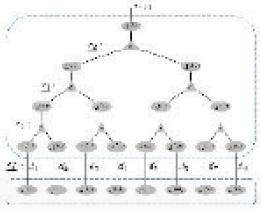
Quantum chemistry

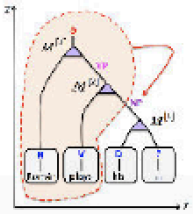


Materials science



AI, Deep learning & Linguistics



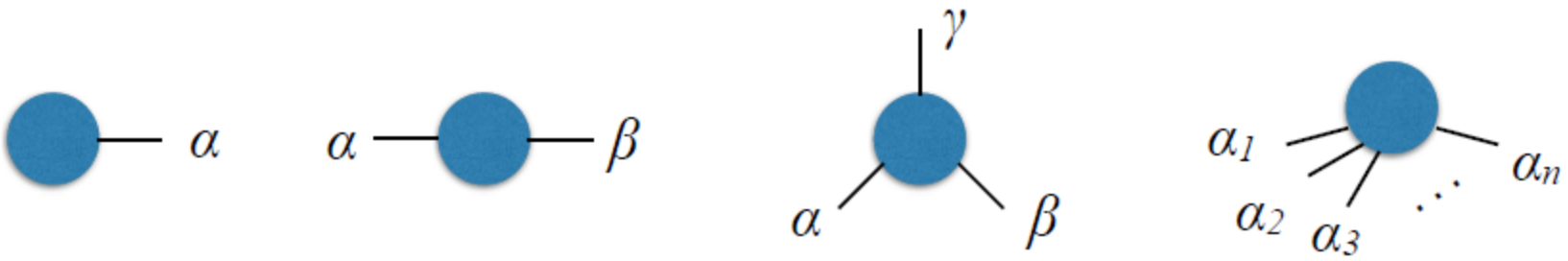


Notation

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_N \end{bmatrix} \qquad B = \begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn} \end{bmatrix}$$

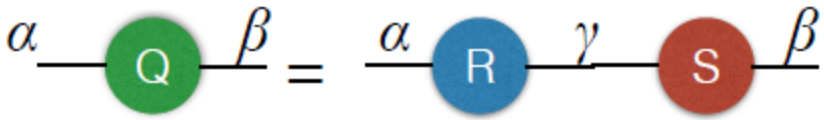
vector | matrix | rank-3 tensor | rank-n tensor

A_α $B_{\alpha\beta}$ $C_{\alpha\beta\gamma}$ $T_{\alpha_1\alpha_2\alpha_3\ldots\alpha_n}$

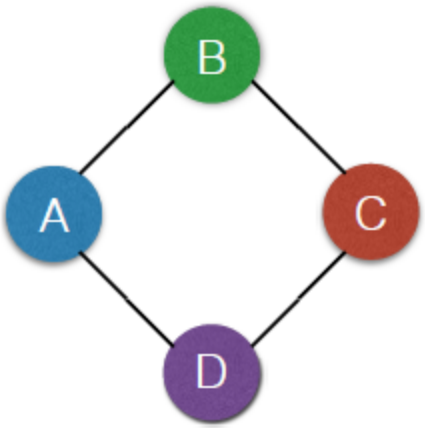


Notation

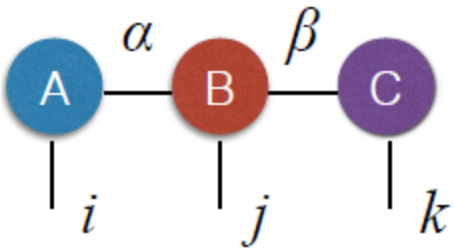
product of tensors (matrices)



$$Q_{\alpha\beta} = \sum_{\gamma} R_{\alpha\gamma} S_{\gamma\beta}$$



$$\mathrm{Tr}(ABCD)$$



- Internal lines are summed over
- External lines are external indices

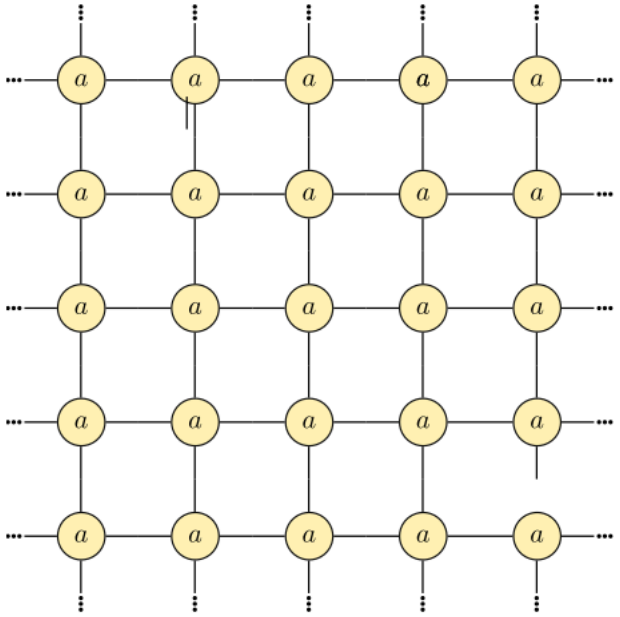
$$T_{ijk} = \sum_{\alpha\beta} A_{\alpha i} B_{\alpha\beta j} C_{\beta k}$$

Tensor Network Representation of Partition Function

Partition Function of a D-dimenstional classical or D+1 quantum model can be represented by a Tensor Network

$$\mathrm{Tr}\, \mathrm{e}^{-\beta H} = \sum_{ijkl\cdots} T_{jfei} T_{hgjk} T_{qklr} T_{lits} \cdots = \mathrm{tTr} \otimes_{\boldsymbol{i}} T.$$

$$\int D\varphi(\boldsymbol{x}, t)\, \mathrm{e}^{-\int \mathcal{L}} = \sum_{\{\phi_i\}} \prod T_{\phi_i \phi_j \phi_k \phi_l} \equiv \mathrm{tTr} \otimes_{\boldsymbol{i}} T$$



Tensor Network Representation of 2D Ising Model

For example, 2D Ising model, which has Onsager solution： $T_c = \frac{2}{\log 1+\sqrt{2}} \approx 2.269$

$$H(\{\sigma\}) = -J\sum_{i,j}(\sigma_{i,j}\sigma_{i+1,j} + \sigma_{i,j}\sigma_{i,j+1}),$$


Lars Onsager

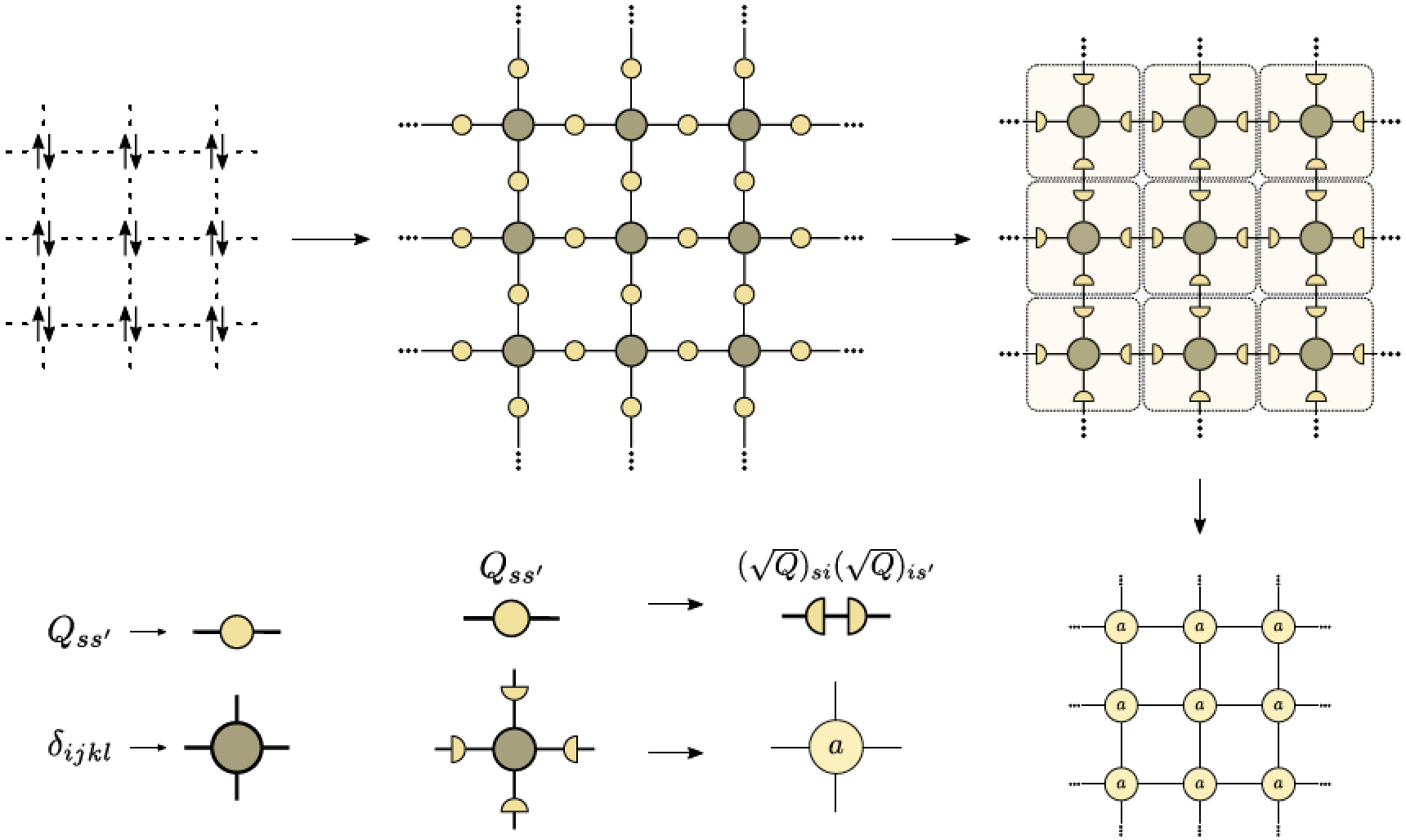
$$Z(\beta) = \sum_{\{\sigma\}} e^{-\beta H(\sigma)} = \sum_{\{\sigma\}} \prod_{<i,j>} e^{\beta\sigma_i\sigma_j} = \sum_{\{\sigma\}} \prod_{<i,j>} Q_{s,s'}.$$

$$Q_{s,s'} = \begin{matrix} s,s' \\ -1 \\ 1 \end{matrix} \begin{matrix} \quad -1 \qquad\quad 1 \\ \begin{pmatrix} \exp(\beta) & \exp(-\beta) \\ \exp(-\beta) & \exp(\beta) \end{pmatrix} \end{matrix} = (\sqrt{Q})_{s,i}(\sqrt{Q})_{i,s'}$$

$$\sqrt{Q} = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{\cosh\beta} + \sqrt{\sinh\beta} & \sqrt{\cosh\beta} - \sqrt{\sinh\beta} \\ \sqrt{\cosh\beta} - \sqrt{\sinh\beta} & \sqrt{\cosh\beta} + \sqrt{\sinh\beta} \end{pmatrix}.$$

Tensor Network Representation of 2D Ising Model

$$Z(\beta) = \sum_{\{\sigma\}} e^{-\beta H(\sigma)} = \sum_{\{\sigma\}} \prod_{<i,j>} e^{\beta \sigma_i \sigma_j} = \sum_{\{\sigma\}} \prod_{<i,j>} Q_{s,s'}.$$

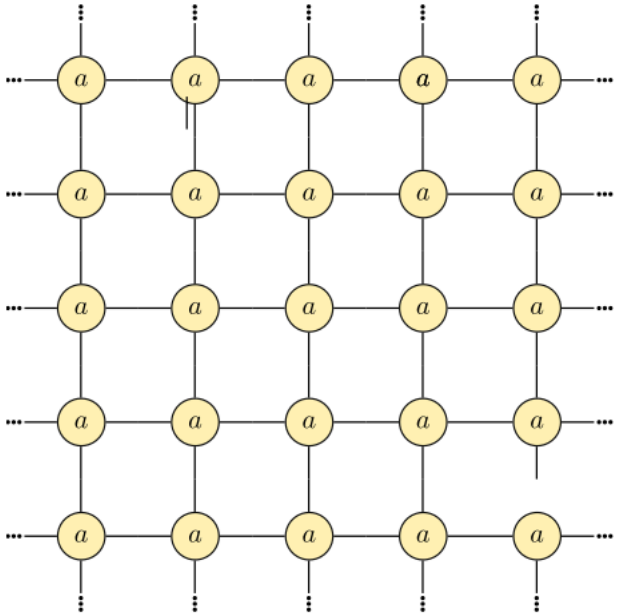


Partition Function Calculated by Contracting a TN

Partition Function of a D-dimenstional classical or D+1 quantum model can be represented by a Tensor Network

$$\mathrm{Tr}\, \mathrm{e}^{-\beta H} = \sum_{ijkl\cdots} T_{jfei} T_{hgjk} T_{qklr} T_{lits} \cdots = \mathrm{tTr} \otimes_i T.$$

$$\int D\varphi(\boldsymbol{x}, t)\, \mathrm{e}^{-\int \mathcal{L}} = \sum_{\{\phi_i\}} \prod T_{\phi_i \phi_j \phi_k \phi_l} \equiv \mathrm{tTr} \otimes_i T$$



Calculation of Partition Function is converted to contract a Tensor Network

Controlled approximate contraction schemes

Coarse Graining

① TRG: Levin & Nave, PRL (2007)
② SRG: Z. Y. Xie, *et. al.* PRL(2009)
③ HOTRG: Z. Y. Xie, *et. al.* PRB(2012)
④ HOSRG: Z. Y. Xie, *et. al.* PRB(2012)
⑤ EV-TNR: Evenbly & Vidal, PRL(2015)
⑥ Loop-TNR: S. Yang, *et. al.* PRL(2017)

Transfer Matrix-Based

⑦ TMRG: Bursill, Xiang & Gehring, JPCM(1996); X. Wang & T. Xiang, PRB(1997)
⑧ TEBD: G. Vidal, PRL(2007)
⑨ VUMPS: Zauner-Stauber, et. al. PRB(2012)

Corner Transfer Matrix-Based

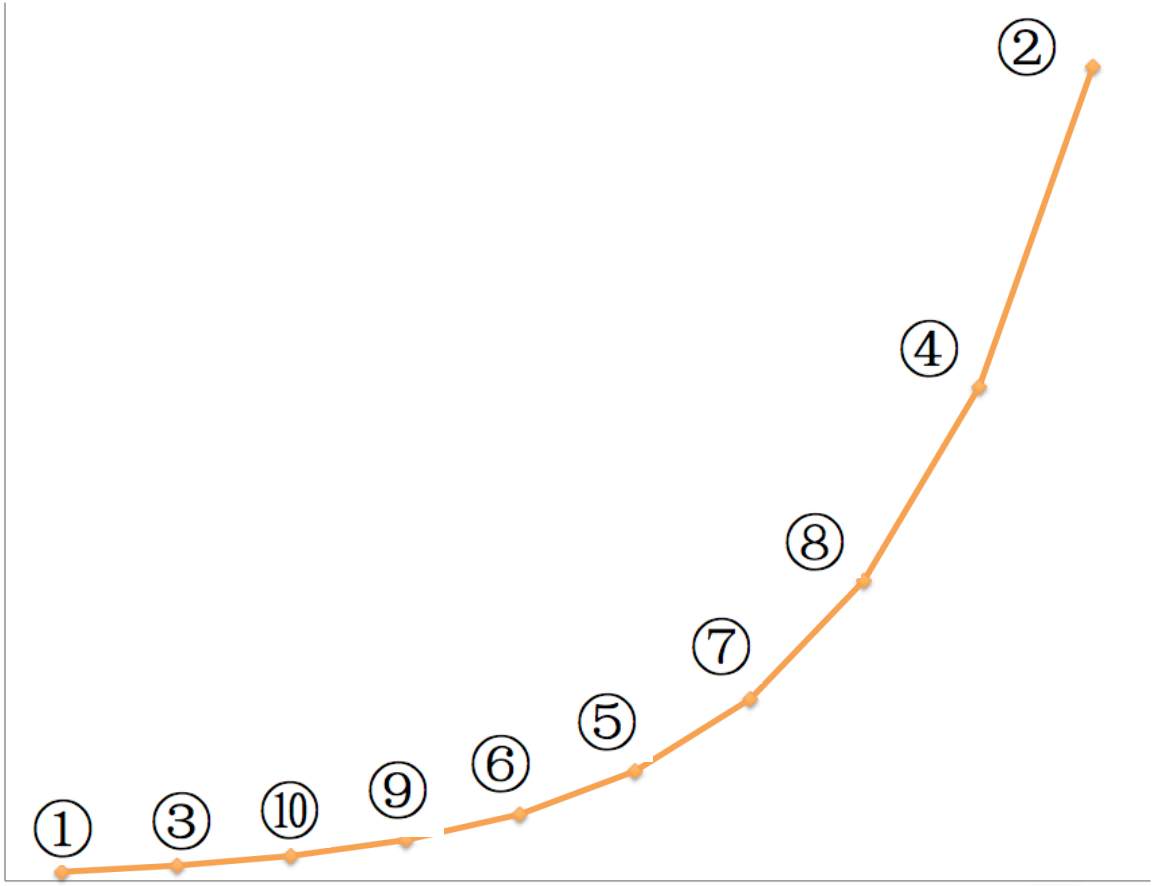
⑩ sym-CTMRG: Nishino, Okunishi, JPSJ (1996)
⑩ directional-CTM: Orus, Vidal, PRB(2009)
⑩ General-CTMRG: Corboz, Mila PRL(2014)
⑩ Fixed-point CTMRG: Fishman,et. al. PRB(2018)

Learning Curve for Various Methods

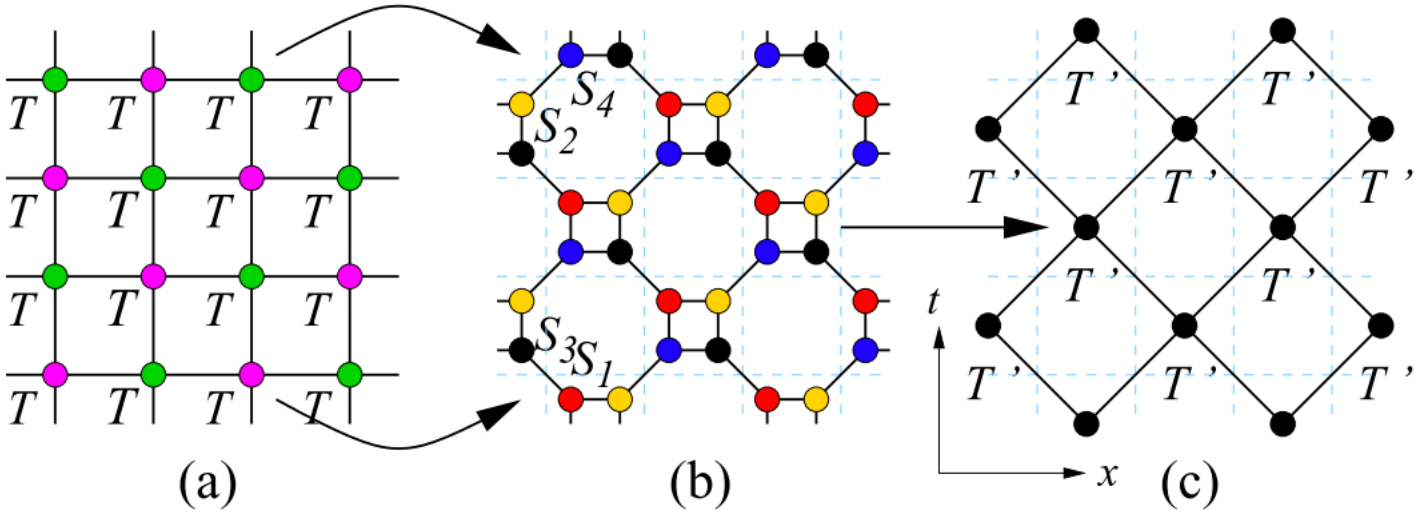
① TRG:
② SRG:
③ HOTRG:
④ HOSRG:
⑤ EV-TNR:
⑥ Loop-TNR:

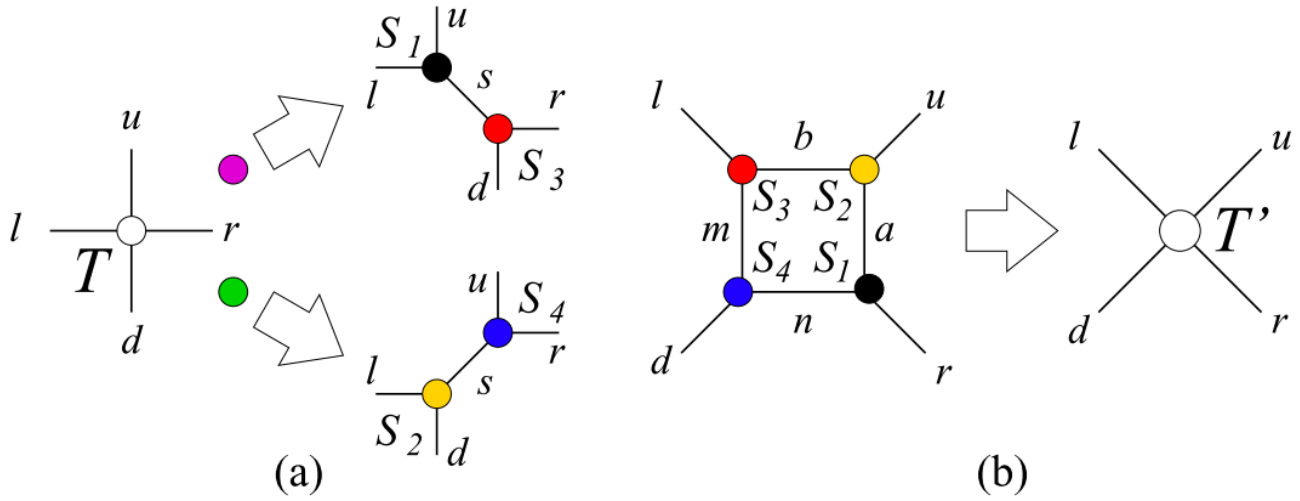
⑦ TMRG:
⑧ TEBD:
⑨ VUMPS:

⑩ sym-CTMRG:
⑩ directional-CTM:
⑩ General-CTMRG:
⑩ Fixed-point CTMRG:



Tensor Renormalization Group (TRG)





M. Levin and C. P. Nave, PRL. 99, 120601 (2007).

Tensor Renormalization Group (TRG)

Partition function Z:

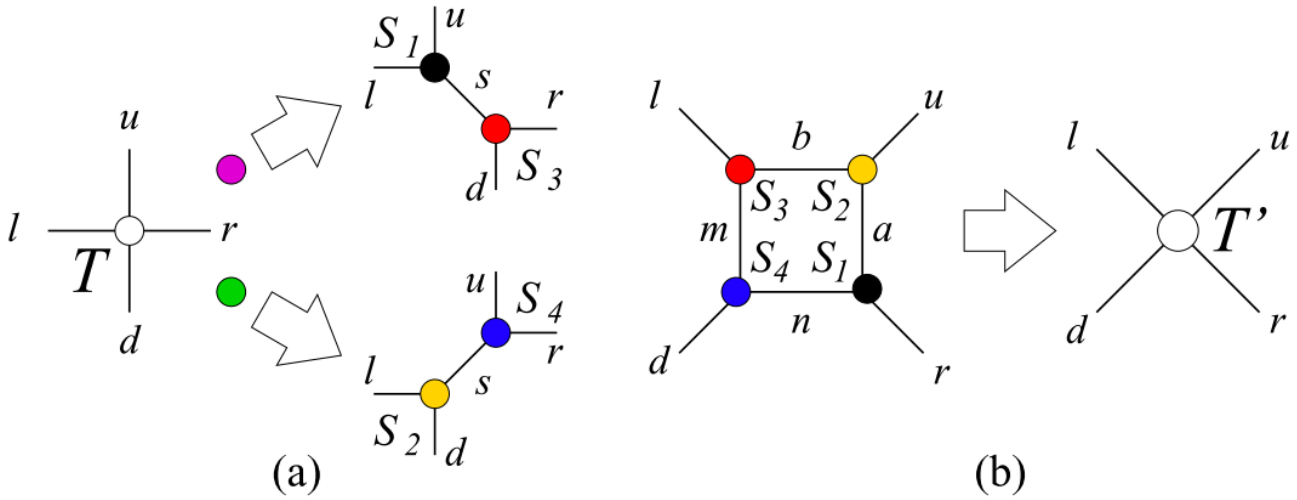
$$
\begin{aligned}
Z &= Tr(T_0^a T_0^b T_0^a T_0^b \ldots) \\
&= Tr(T_1^a T_1^b T_1^a T_1^b \ldots) \cdot \lambda_1^{N/2} \\
&= Tr(T_2^a T_2^b T_2^a T_2^b \ldots) \cdot \lambda_1^{N/2} \lambda_2^{N/2^2} \\
&= \ldots \\
&= Tr(T_n^a T_n^b) \cdot \prod_{i=1}^{n} \lambda_i^{N/2^i}
\end{aligned}
$$

$$
\frac{\log(Z)}{N} = \sum_i \frac{\log(\lambda_i)}{2^i} + \frac{\log(R)}{2 \times 2^n}
$$

M. Levin and C. P. Nave, PRL. 99, 120601 (2007).

Tensor Renormalization Group (TRG)

Key of TRG algorithm is the local truncation by SVD

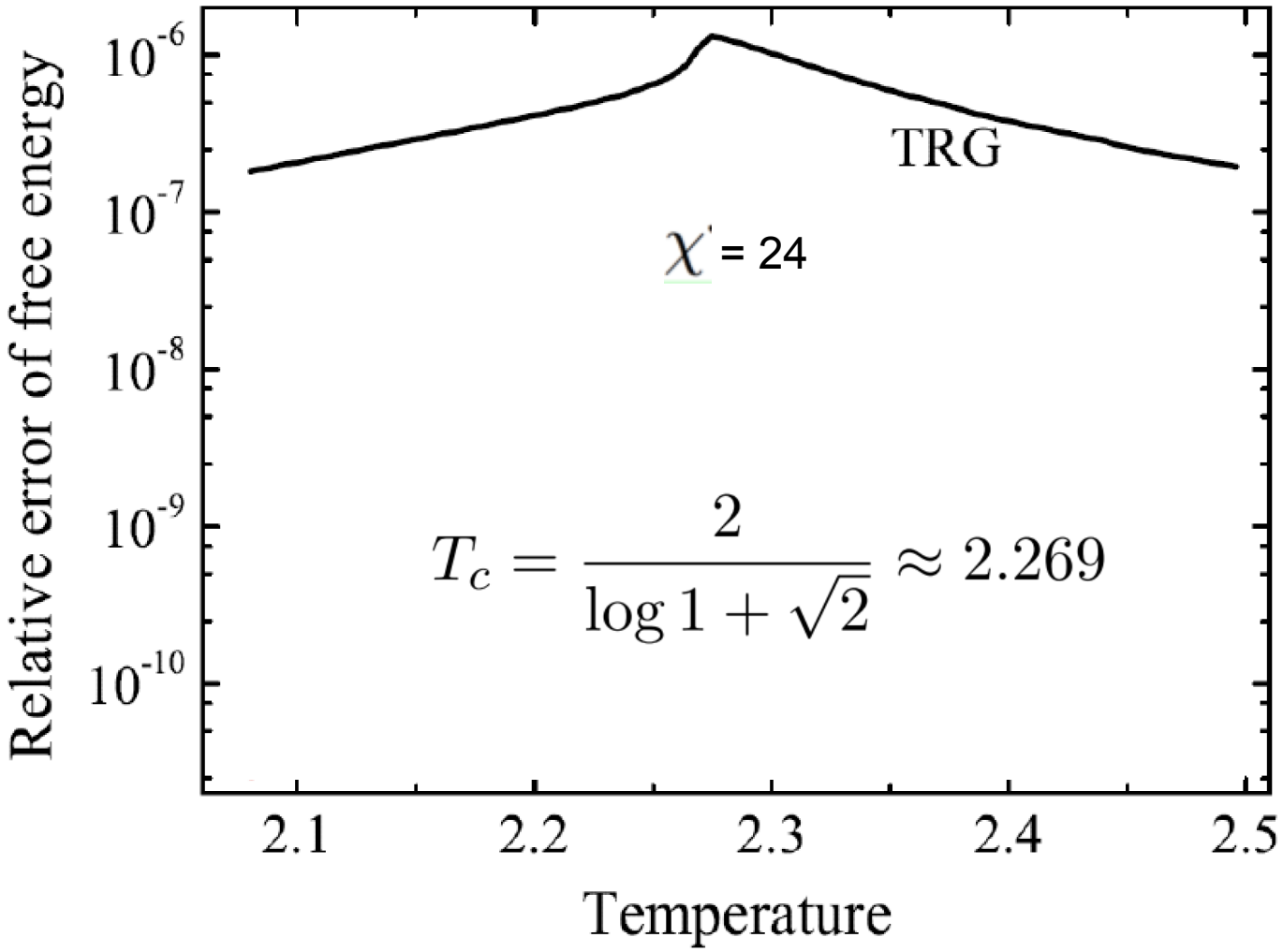


The Eckart-Young theorem：
Optimal low-rank approximation of a matrix is its SVD

G. Eckart and G. Young, Psychutnetiku 1: 211-218 (1936).

TRG result for 2D Ising model

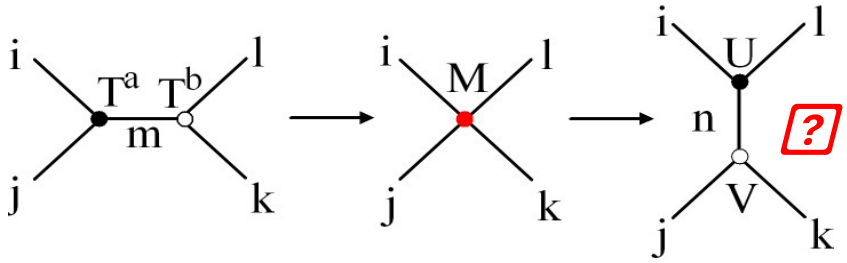
$$f = \frac{-\log(2)}{2\beta} - \frac{1}{\beta 2\pi}\int_0^{\pi} \log\left[\cosh(2J)\cosh(2J) + \frac{1}{k}\sqrt{1+k^2-2k\cos(2\theta)}\right] d\theta$$



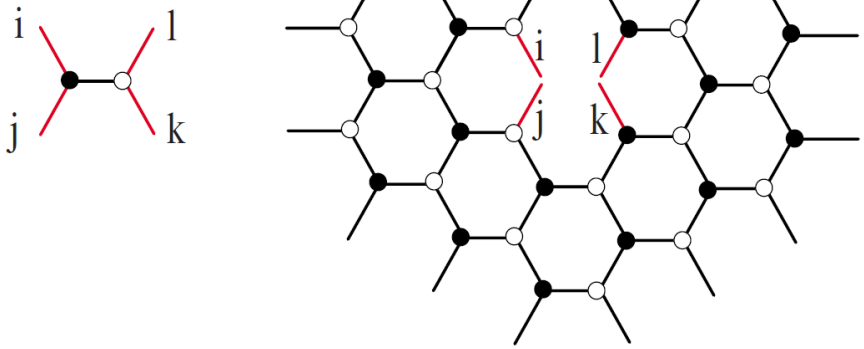
Second Renormalization Group

Second Renormalization Group (SRG)

Xie et al. PRL 103, (2009)



$$Z = Tr\left(MM^{env}\right)$$



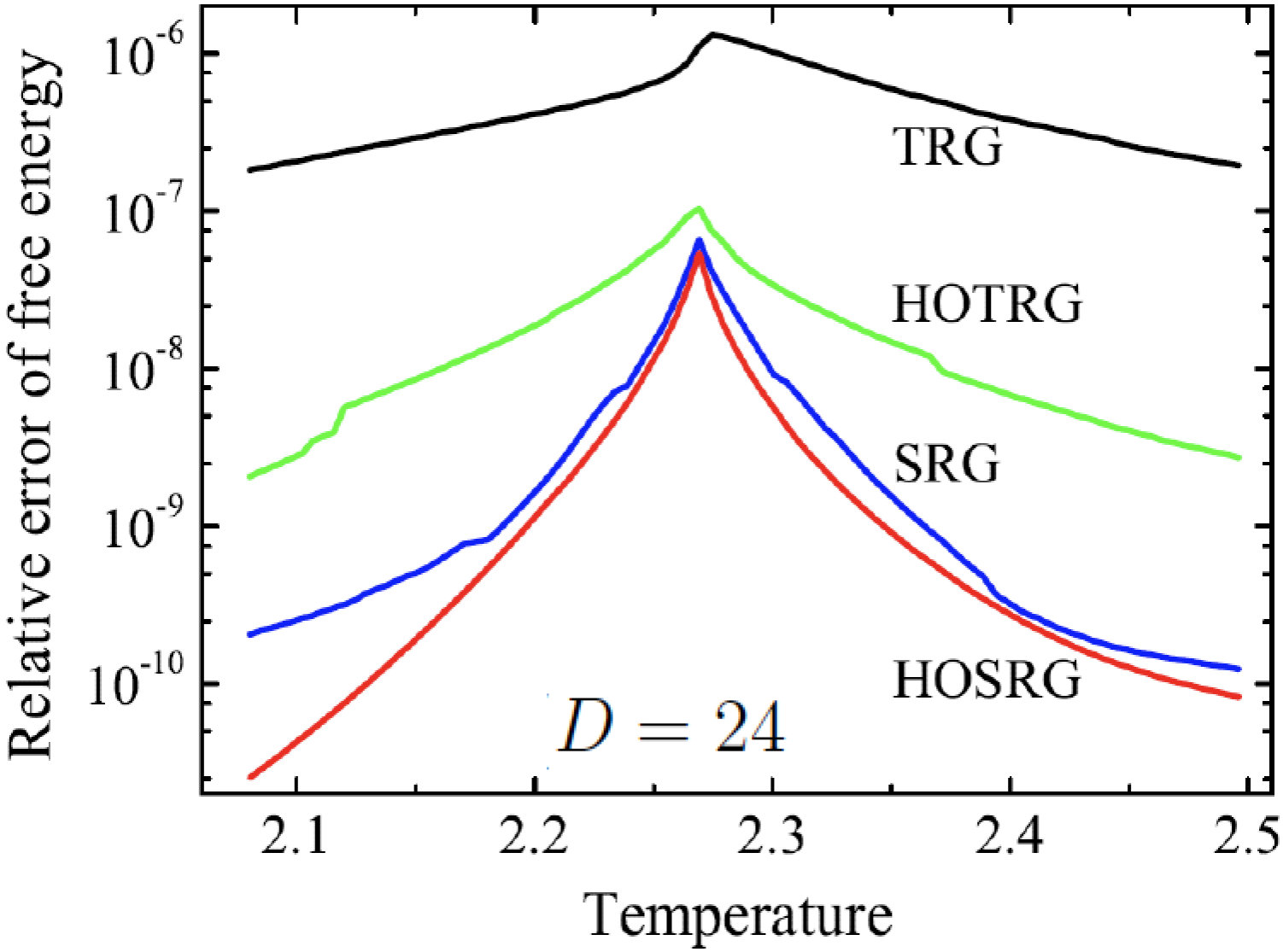
- TRG: local optimization by SVD

Z not M!

- SRG: approximate global optimization by considering environment

HOSRG: Environment calculated by HOTRG, **Z. Y. Xie, *et. al.* PRB(2012)**

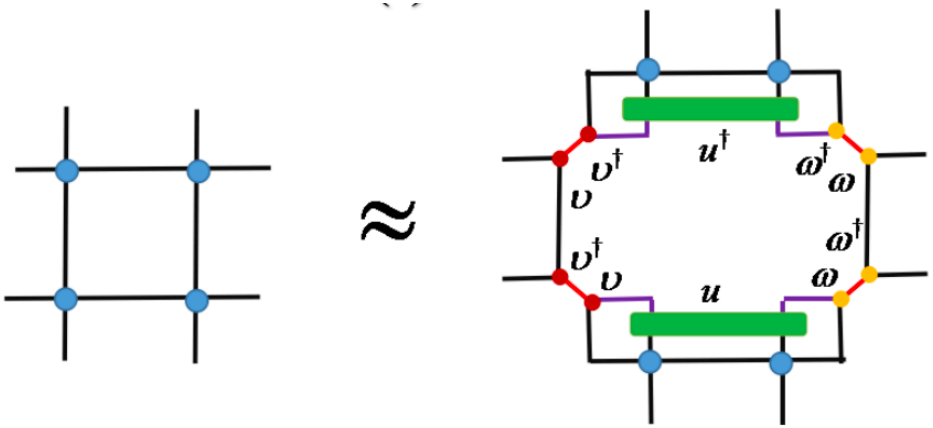
SRG/HOSRG result for 2D Ising model



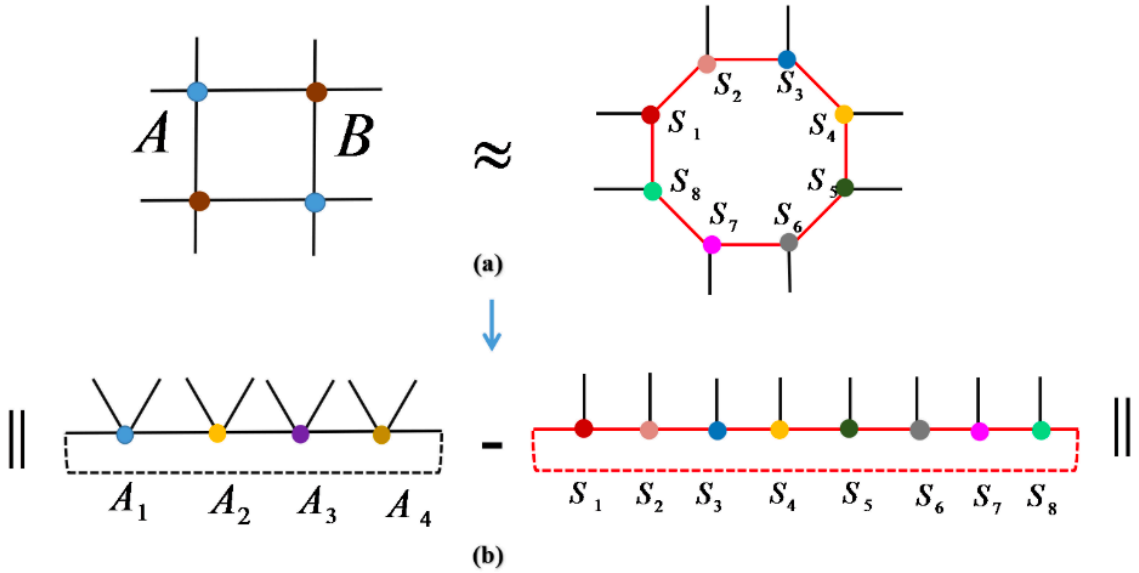
Z. Y. Xie, *et. al.* PRB(2012)

Tensor Network Renormalization

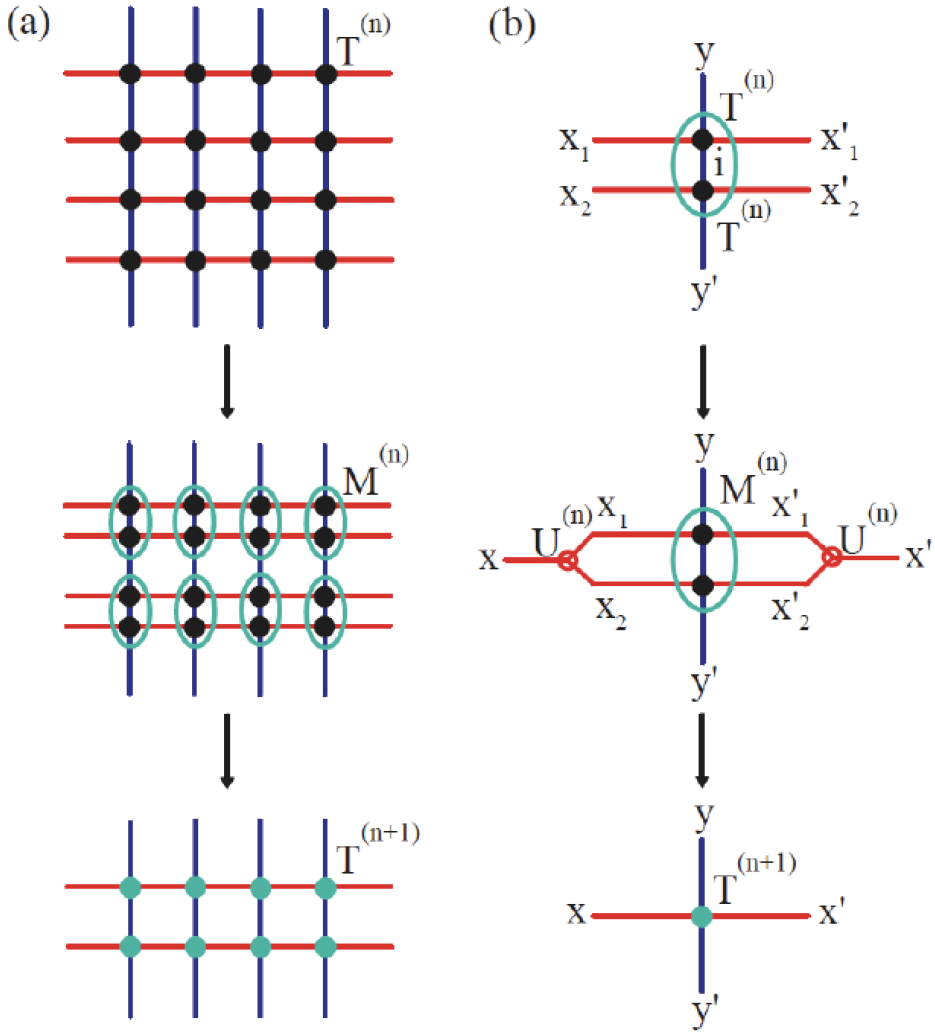
Tensor Network Renormalization (TNR) Evenbly & Vidal, PRL(2015)



Loop-TNR: S. Yang, *et. al.* PRL(2017)



Higher-Order TRG(HOTRG)



Key Idea: HOSVD

Lathauwer, et. al., (2000)

$$M^{(n)}_{xx'yy'} = \sum_{ijkl} S_{ijkl} U^{L}_{xi} U^{R}_{x'j} U^{U}_{yk} U^{D}_{y'l},$$

(1) all orthogonality,

$$\langle S_{:,j,:,:} \mid S_{:,j',:,:} \rangle = 0, \qquad \text{if } j \neq j',$$

where $\langle S_{:,j,:,:} \mid S_{:,j',:,:} \rangle$ is the inner-product o
sub-tensors.

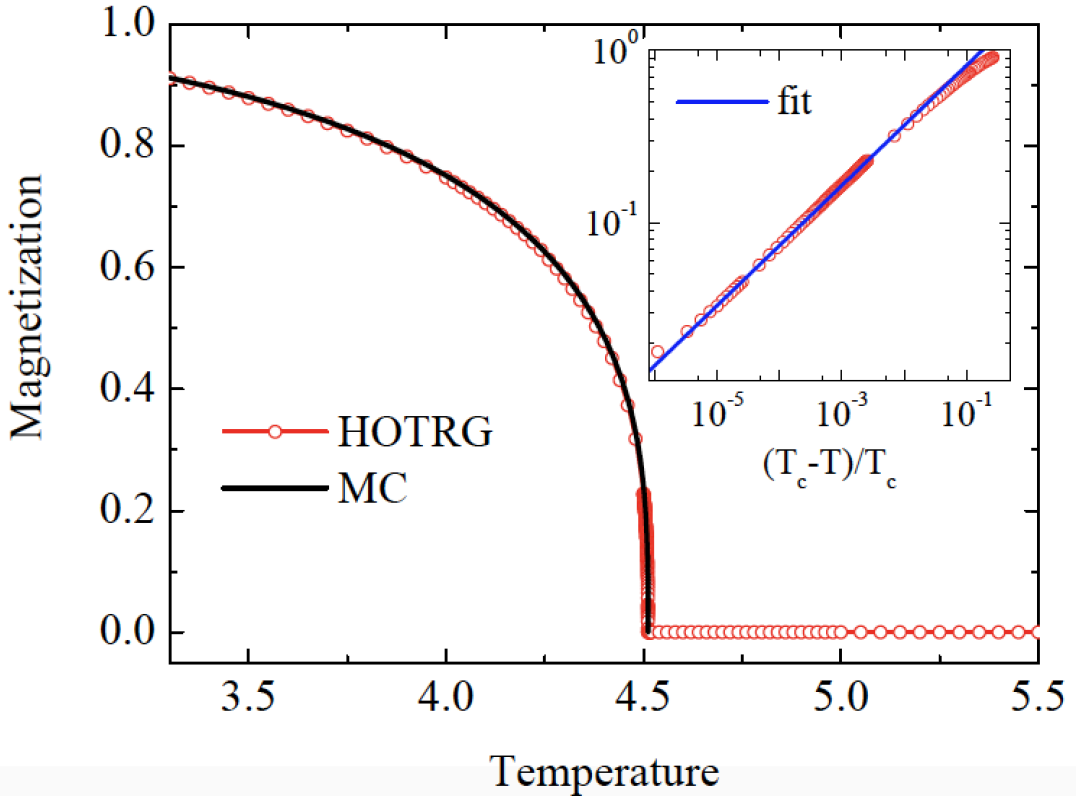
(2) pseudo-diagonal,

$$|S_{:,j,:,:}| \geq |S_{:,j',:,:}|, \qquad \text{if } j < j',$$

Z. Y. Xie, *et. al.* PRB(2012)

3D Classical Ising Model by HOTRG

Only HOTRG can successfully deal with 3D problem



method	T_c	
Monte Carlo [33]	4.511523	
Monte Carlo [25]	4.511528	
Monte Carlo [35]	4.511525	
Monte Carlo [36]	4.511516	
Series Expansion [34]	4.511536	
KWA [41]	4.5788	former NRG
CTMRG [13]	4.5704	
CTMRG [40]	4.5392	
TPVA [14]	4.554	
Algebraic variation [37]	4.547	
HOTRG(D=16, from U)	4.511544	
HOTRG(D=16, from M)	4.511546	

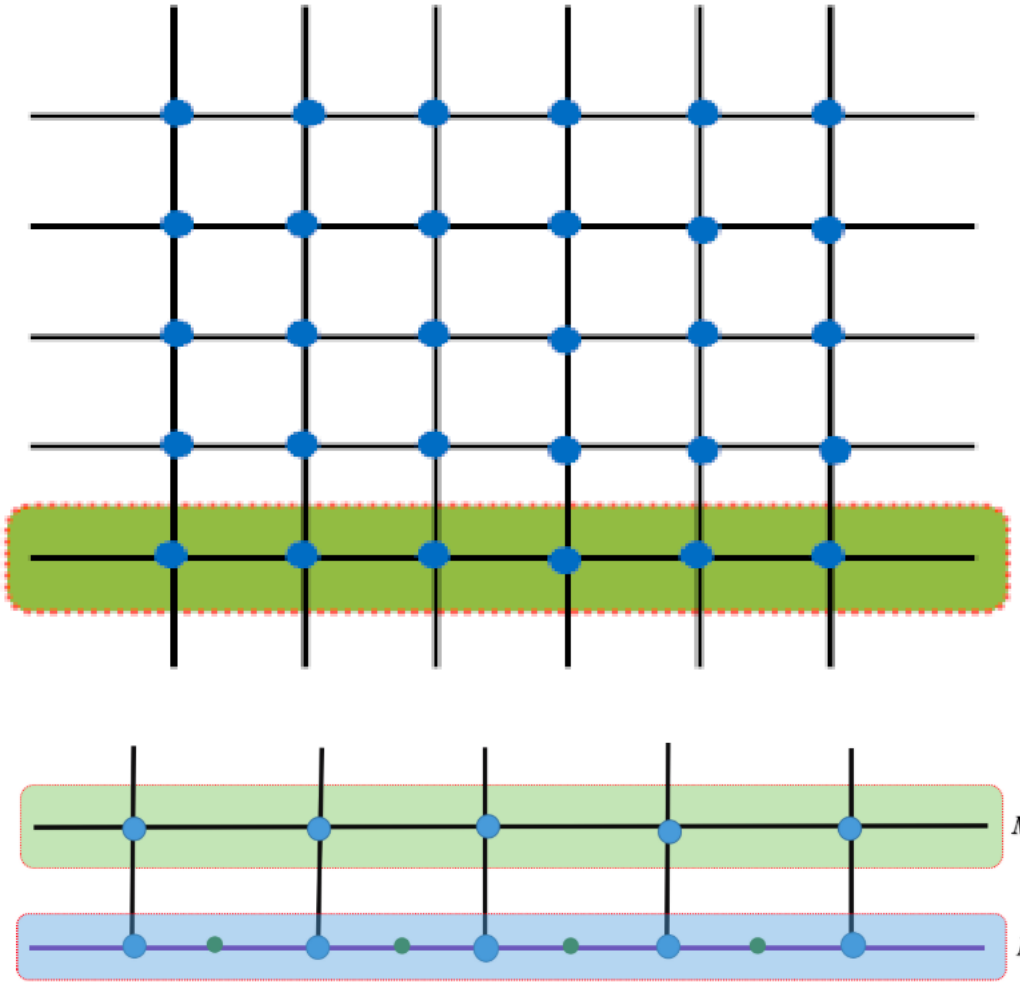
Monte Carlo (2003)[43]	4.5115248(6)
Monte Carlo (2010)[44]	4.5115232(17)
HOTRG ($D = 16$) (2012)[24]	4.511544
HOTRG ($D = 23$, this work)	4.51152469(1)

Z.Y. Xie, PRB 86, 045139 (2012)
S. Wang, et al, CPL 31, 070503 (2014)

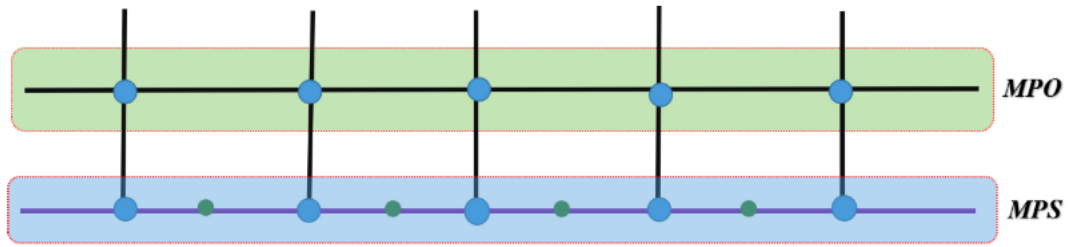
Transfer Matrix-Based Method

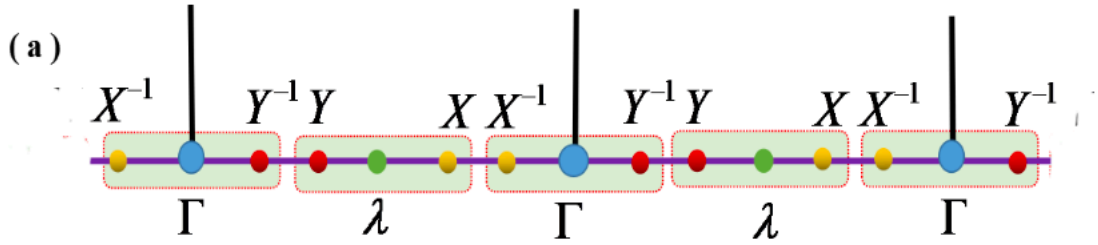
Transfer Matrix-Based Methods include TMRG, TEBD, VUMPS

Transfer Matrix

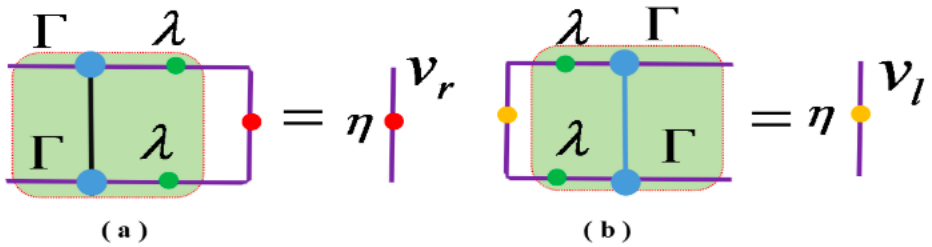


Transfer Matrix-Based Method----TEBD





Canonical Form:

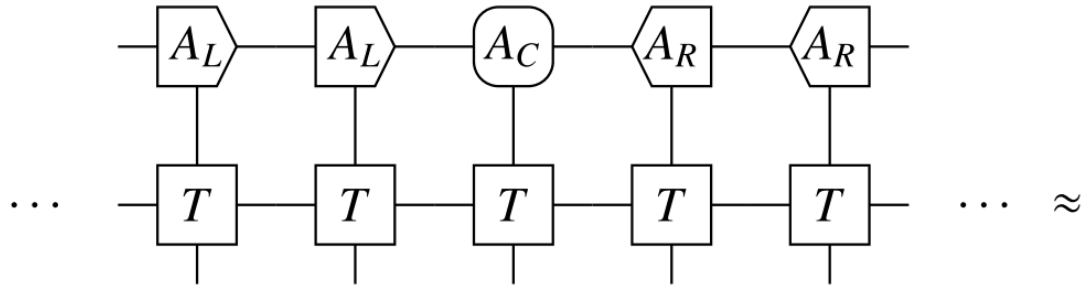


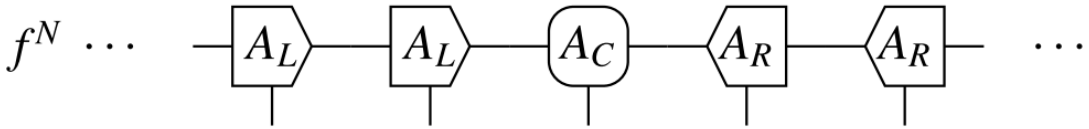
G. Vidal, PRL(2007); R. Orús and G. Vidal, PRB(2008)

Transfer Matrix-Based Method----VUMPS

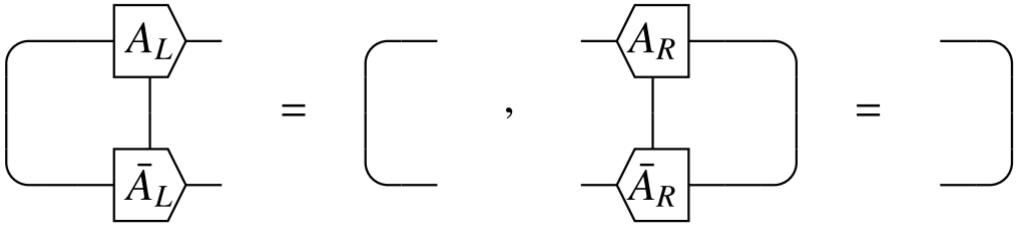
Fixed-point MPS

Transfer Matrix

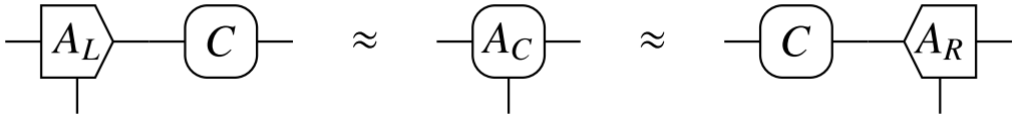




Mix Canonical Form



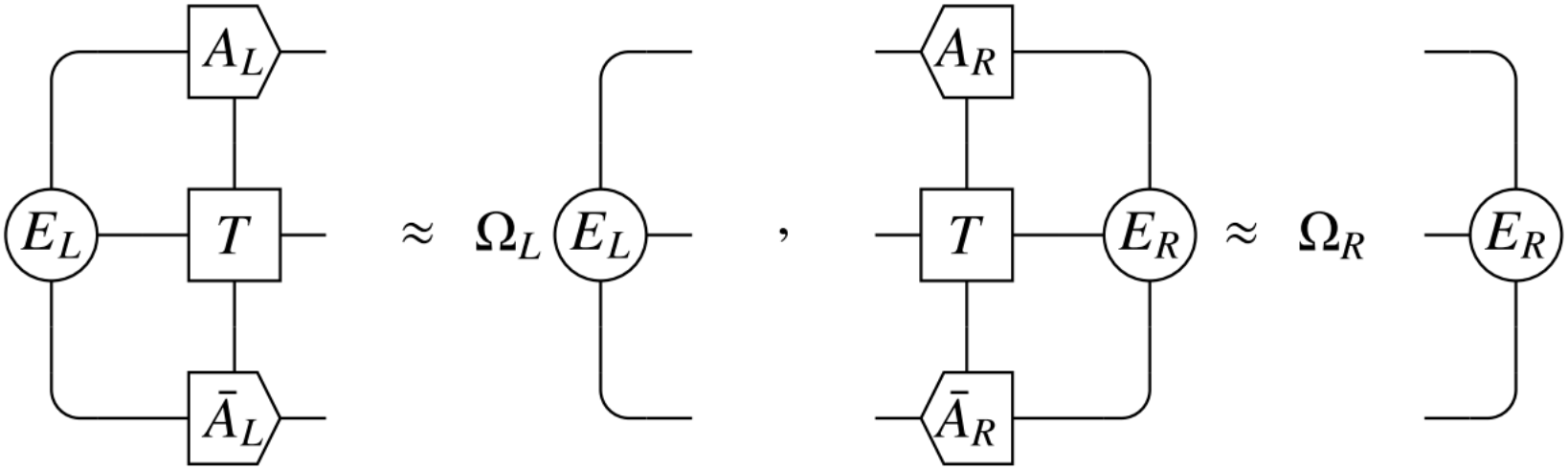
Fixed-point Equation



Transfer Matrix-Based Method----VUMPS

To find the fixed-point A_L, A_R, and A_C tensors, the VUMPS algorithm mainly includes the following four steps:

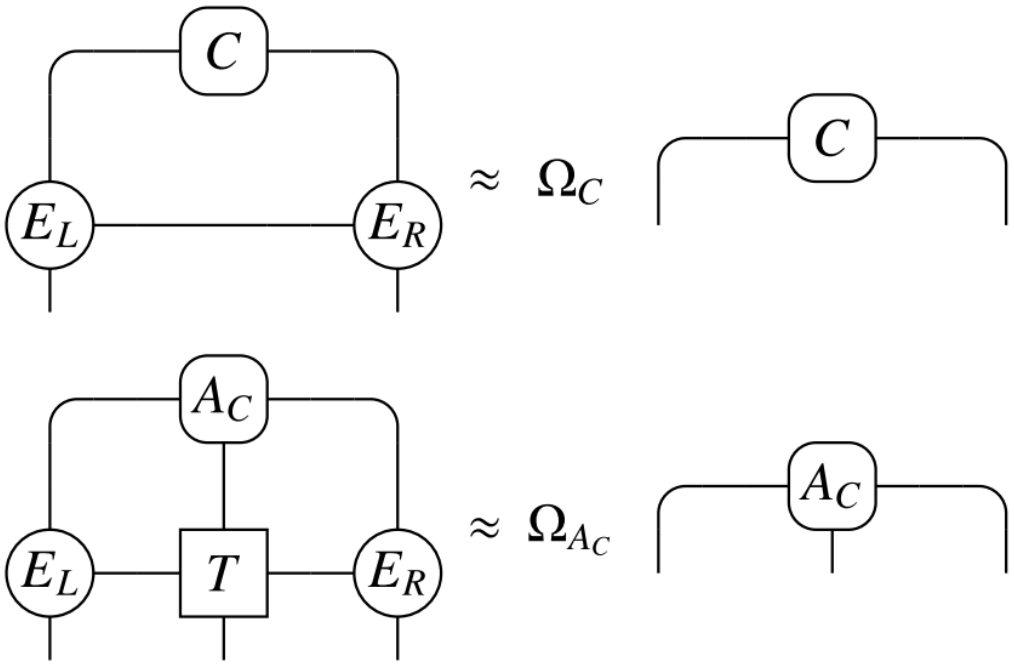
1. Find the left and right environments E_L and E_R by power or Arnoldi method:



(13)

Transfer Matrix-Based Method----VUMPS

2. Find the central tensors C and A_C by power or Arnoldi method:

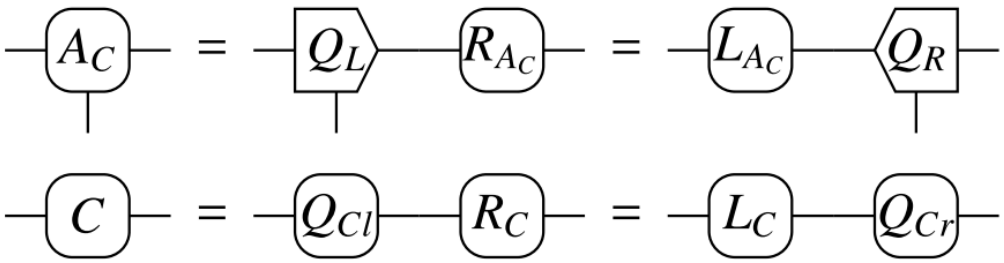


(14)

where $\Omega_{A_C}/\Omega_C \approx \Omega_L \approx \Omega_R$ is approximately equal to the free energy density f near or at the fixed point.

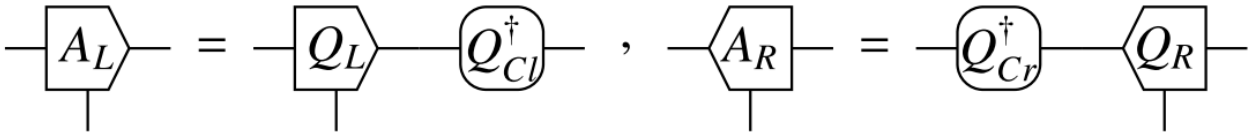
Transfer Matrix-Based Method----VUMPS

3. Obtain new A_L and A_R from A_C and C using the polar decompositions or QR decompositions [34, 35],



(15)

where Q_L, Q_R, Q_{Cl} and Q_{Cr} are unitary matrices, and the diagonal elements of R_{A_C}, L_{A_C}, R_C and L_C are positive to fix gauge, then obtain

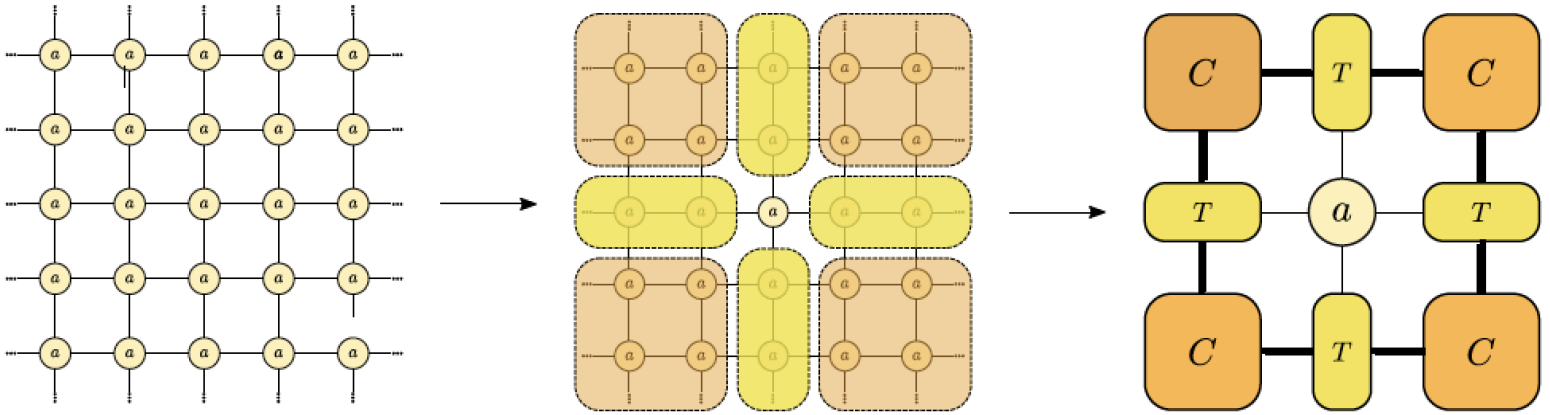


(16)

4. Repeat 1-3 until the error of the fixed-point Eq. (12) is enough small.

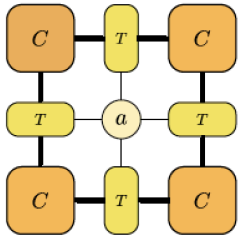
Corner Transfer Matrix-Based Method---symCTMRG

Corner Transfer Matrix RG (CTMRG) method



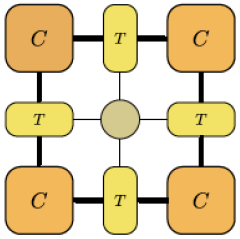
(a)

$$Z =$$



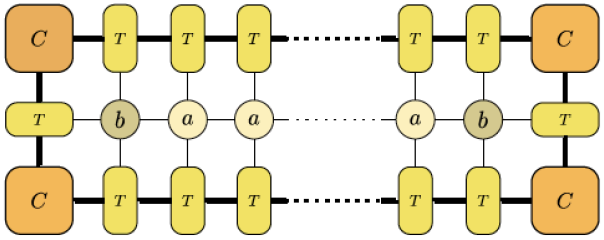
(b)

$$m = \frac{1}{Z}$$



(c)

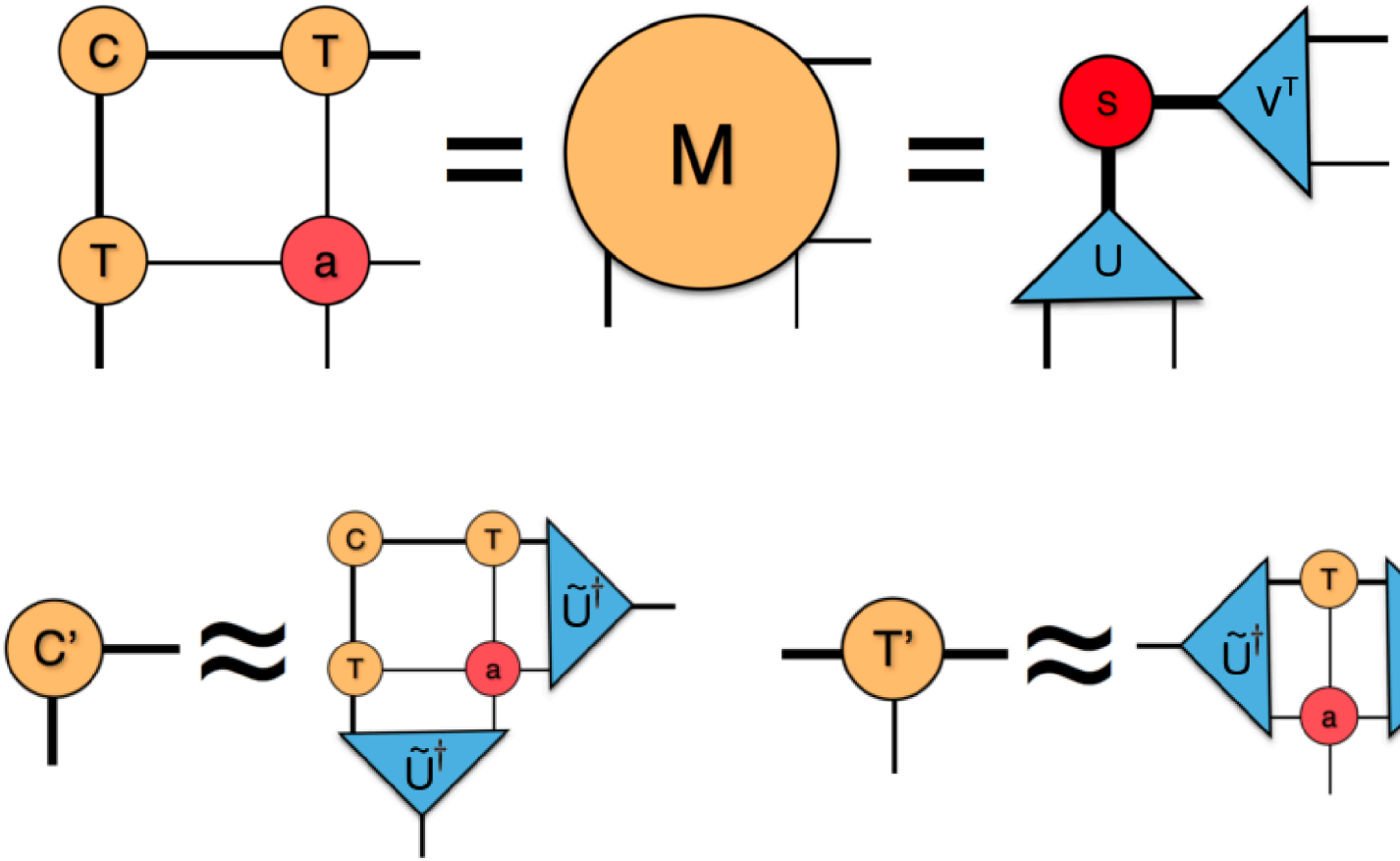
$$< \sigma_{i,j}\sigma_{i+N,j} > = \frac{1}{Z}$$



Nishino, Okunishi, JPSJ(1996)

Corner Transfer Matrix-Based Method---symCTMRG

Find Isometry and Update Corner and Edge Tensors:



Nishino, Okunishi, JPSJ(1996)

Computation Cost(CC) & Memory Cost(MC) of Various Methods

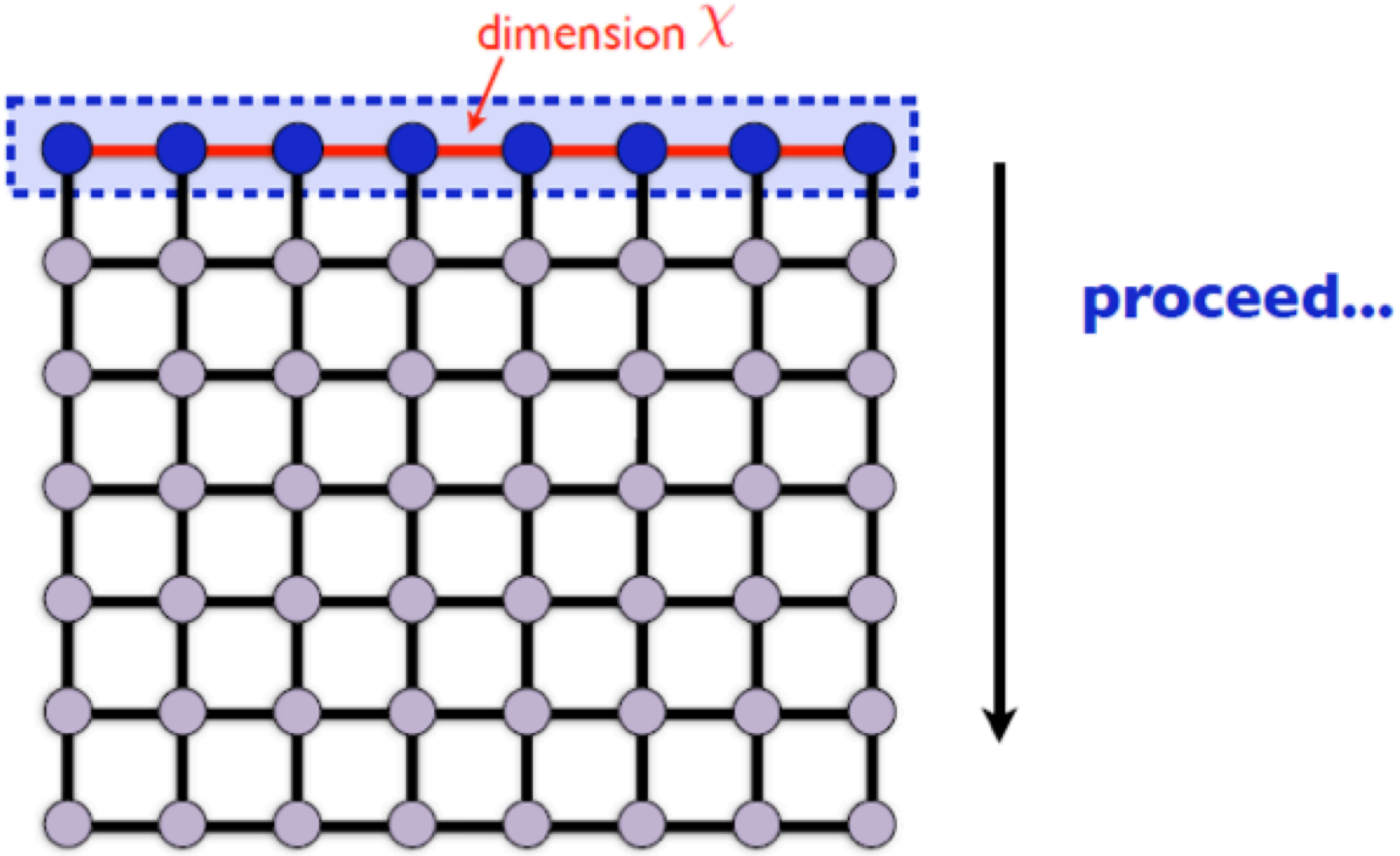
		CC	MC
Coarse Graining	① TRG:	χ^6	χ^4
	② SRG:	χ^6	χ^4
	③ HOTRG:	χ^7	χ^4
	④ HOSRG:	χ^8	χ^6
	⑤ EV-TNR:	χ^7	χ^5
	⑥ Loop-TNR:	χ^6	χ^4
Transfer Matrix-Based	⑦ TMRG:	χ^3	χ^2
	⑧ TEBD:		
	⑨ VUMPS:		
Corner Transfer Matrix-Based	⑩ sym-CTMRG:	χ^3	χ^2
	⑩ directional-CTM:		
	⑩ General-CTMRG:		
	⑩ Fixed-point CTMRG:		

Big room for improvement. **Many possible extensions.**

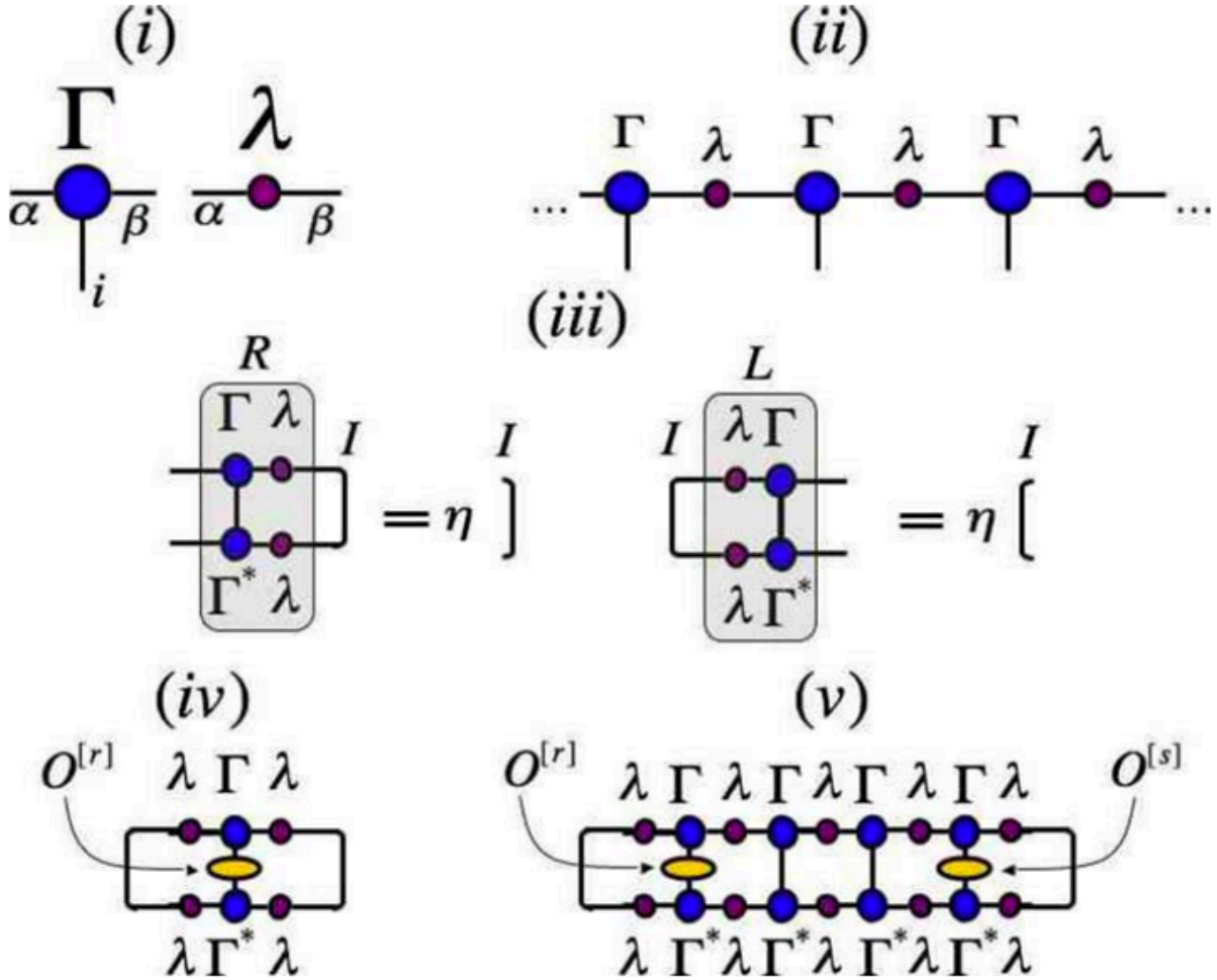
It's an exciting time to work on tensor networks!

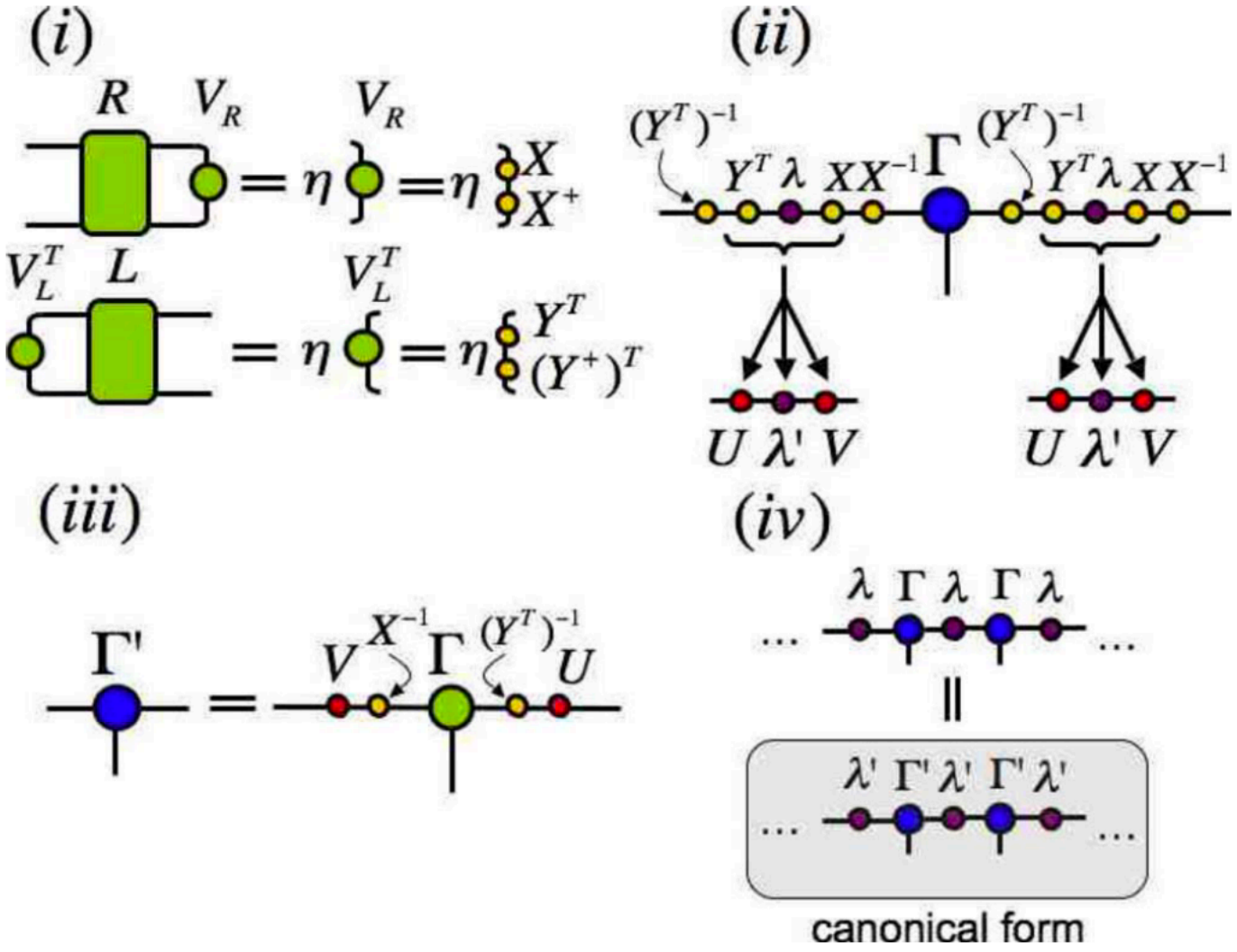
Thanks for your attention!

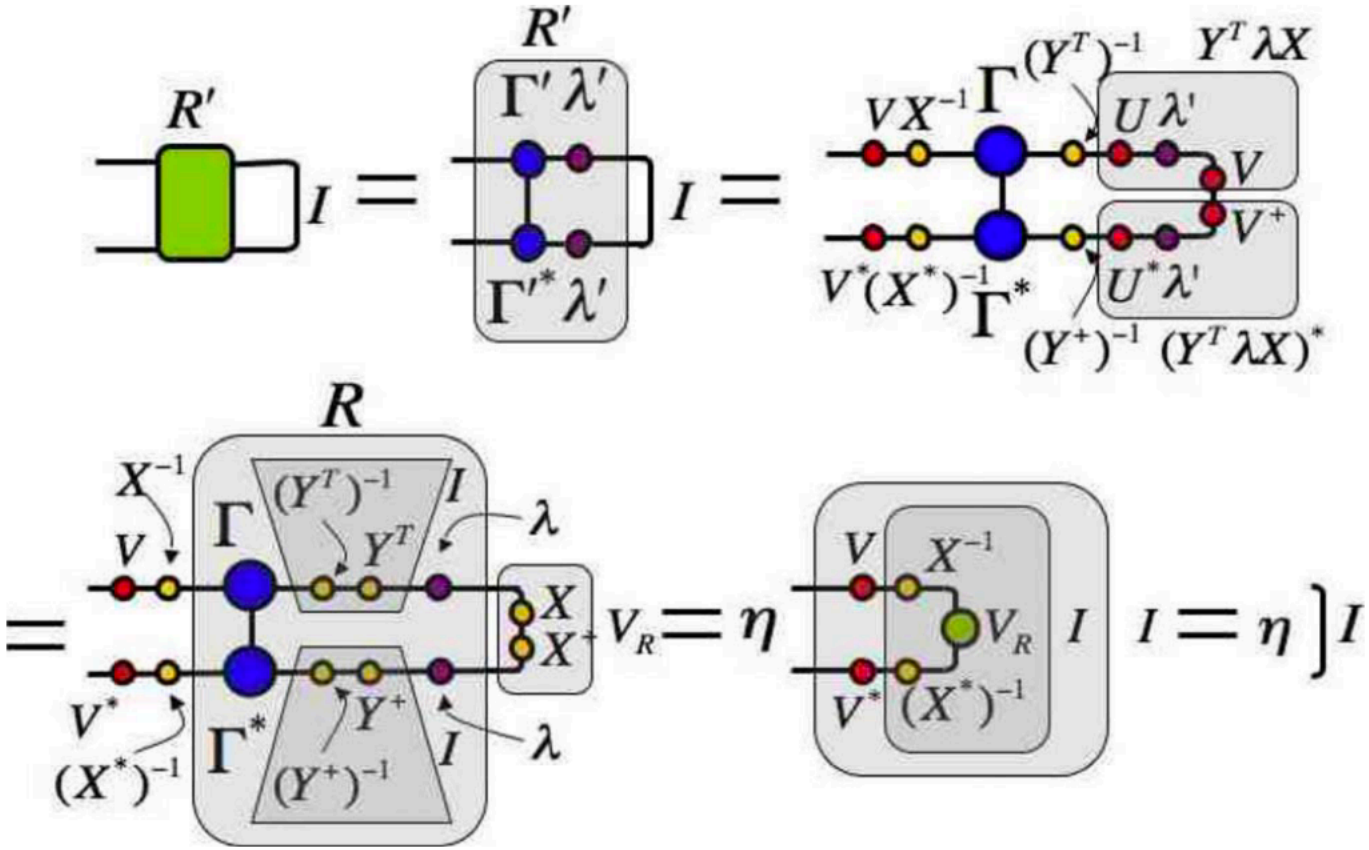
TEBD (also called Boundary-MPS method)



G. Vidal, PRL(2007)


(i)
Γ
λ
α
β
α
β
i
(ii)
Γ λ Γ λ Γ λ
...
...
(iii)
R
Γ λ
I
Γ* λ
= η
I
L
λ Γ
I
λ Γ*
= η
I
(iv)
O[r]
λ Γ λ
λ Γ* λ
(v)
O[r]
O[s]
λ Γ λ Γ λ Γ λ Γ λ
λ Γ* λ Γ* λ Γ* λ Γ* λ


(i)
R
V_R
V_R
$= \eta$
$= \eta$
X
X^+
V_L^T
L
V_L^T
$= \eta$
$= \eta$
Y^T
$(Y^+)^T$
(ii)
$(Y^T)^{-1}$
$Y^T \lambda$ XX^{-1} Γ
$(Y^T)^{-1}$
$Y^T \lambda$ XX^{-1}
U λ' V
U λ' V
(iii)
Γ'
=
V
X^{-1}
Γ
$(Y^T)^{-1}$
U
(iv)
λ Γ λ Γ λ
...
...
=
λ' Γ' λ' Γ' λ'
...
...
canonical form


R'
I =
R'
Γ' λ'
Γ'* λ'
I =
VX^-1 Γ
(Y^T)^-1
Y^T λX
U λ'
V
V^+
V*(X*)^-1 Γ*
U* λ'
(Y^+)^-1
(Y^T λX)*
=
R
X^-1
V
Γ
(Y^T)^-1
I
Y^T
λ
X
X^+
V_R = η
V*
(X*)^-1
Γ*
Y^+
(Y^+)^-1
I
λ
V
X^-1
V_R
I
V*
(X*)^-1
I = η]I