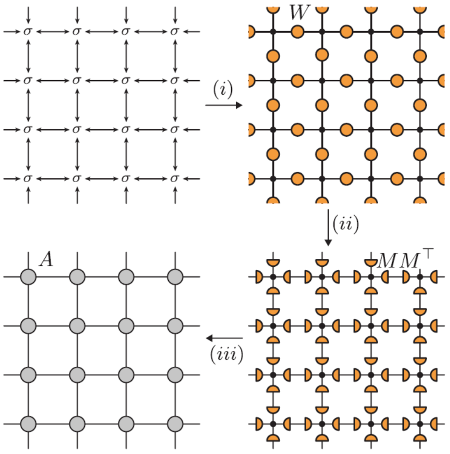
Tensor Renormalization Group (TRG) and related schemes (VUMPS, CTM, FCTM)

TRG-I.1

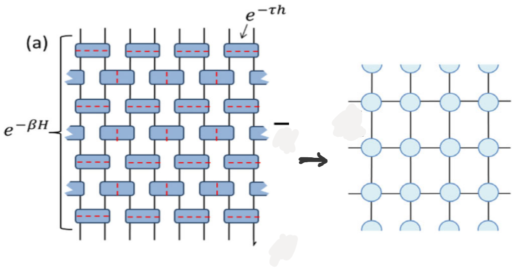
Goal: Compute 2D contractions by coarse-graining RG schemes (instead of transfer matrix schemes)

Applications:

Partition functions of
2D classical models:



Imaginary time evolution of
1D quantum models:



[Levin2007] Levin, Nave: proposed original idea for TRG for classical lattice models.
Local approach: truncation error is minimized only locally.

[Jiang2008] Jiang, Weng, Xiang: adapted Levin-Nave idea to 2D quantum ground state projection via imaginary time evolution. Local approach: truncation is done via 'simple update'. TRG is used to compute expectation values.

[Xie2009] Jiang, Chen, Weng, Xiang; and [Zhao2010] Zhao, Xie, Chen, Wei, Cai, Xiang:
Propose 'second renormalization' (SRG), a global approach taking account renormalization of environmental tensor ('full update'). Reduced truncation error significantly.

[Xie2012] Xie, Qin, Zhu, Yang, Xiang: different coarse-graining scheme, using higher-order SVD, employing both local and global optimization schemes.

[Zhao2016] Zhao, Xie, Xiang, Imada: coarse-graining on finite lattices.

[Evenbly2019] Lan, Evenbly: propose core tensor renormalization group (CTRG), which rescales lattice size linearly (not exponentially), but at much lower cost, $\mathcal{O}(\chi^4)$ (rather than $\mathcal{O}(\chi^6)$).

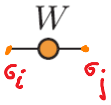
1. Tensor renormalization group (TRG)

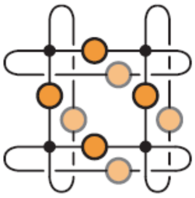
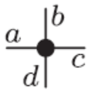
following [Hauru2018]

TRG-I.1

Spin Hamiltonian: $H(\{\sigma\}) = \sum_{\langle i,j\rangle} h(\sigma_i,\sigma_j)$, $h(\sigma_i,\sigma_j) = -\sigma_i\sigma_j$, $\sigma_i \in \{\uparrow,\downarrow\} = \{+1,-1\}$ (1)

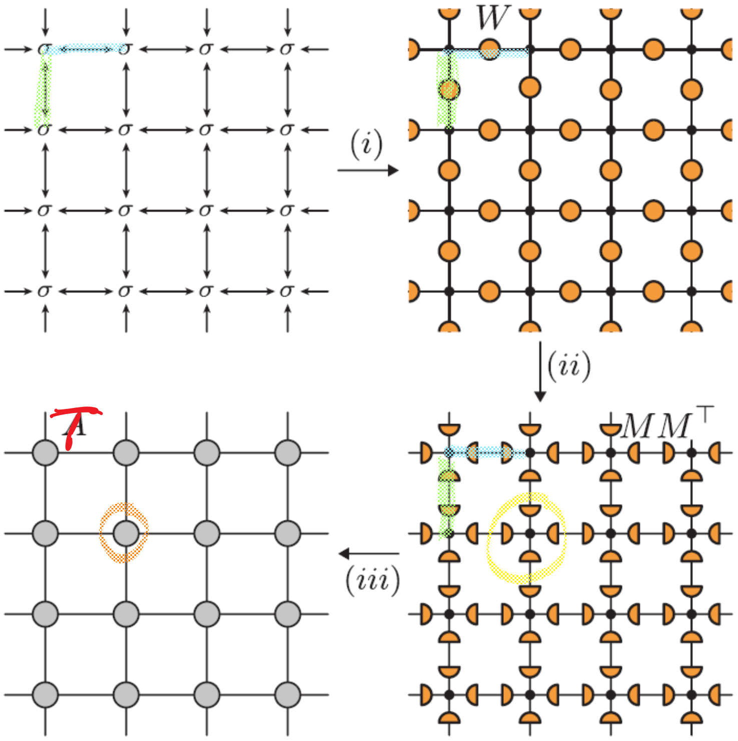
Classical partition function: $Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} = \sum_{\{\sigma\}} \bigotimes_{\langle i,j\rangle} W_{\sigma_i\sigma_j}$ (2)

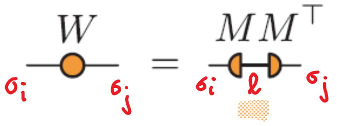
Bond weights: $W_{\sigma_i\sigma_j} = e^{-\beta h(\sigma_i,\sigma_j)} = \begin{pmatrix} W_{\uparrow\uparrow} & W_{\uparrow\downarrow} \\ W_{\downarrow\uparrow} & W_{\downarrow\downarrow} \end{pmatrix} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix} =:$  (3)

For 2x2 lattice (with periodic conditions): $Z =$  with $\delta_{abcd} =$  (4)

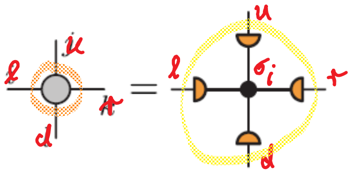
For inifinite 2D lattice,

we obtain a 2D tensor network:



 (5)

$$M = \begin{pmatrix} \sqrt{\cosh\beta} & \sqrt{\sinh\beta} \\ \sqrt{\cosh\beta} & -\sqrt{\sinh\beta} \end{pmatrix} \qquad (6)$$

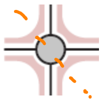
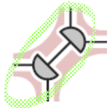
$T_{lurd} =$  (7)

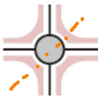
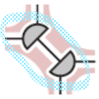
$$= \sum_{ijkl} \delta_{ijkl} M_{il} M_{ju} M_{kr} M_{ld} \qquad (8)$$

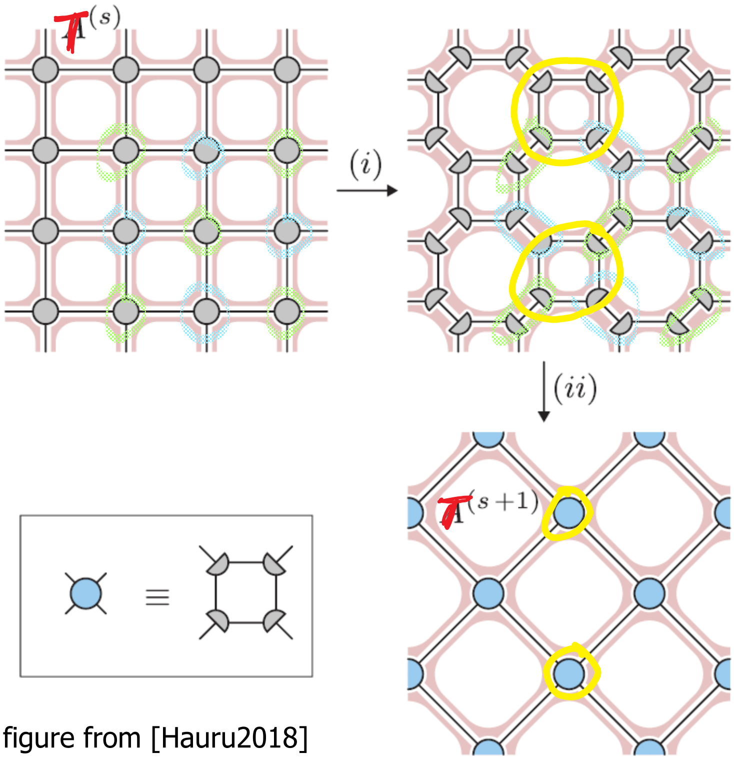
$$= \sum_{\sigma} M_{\sigma l} M_{\sigma u} M_{\sigma r} M_{\sigma d} \qquad (9)$$

Technical challenge: contract this infinite tensor network!

Do SVD on T in two different ways:

$T =$  $\approx$  $= U\sqrt{S}\sqrt{S}V^{\dagger}$ (10)

$T =$  $\approx$  $= U'\sqrt{S'}\sqrt{S'}V'^{\dagger}$ (11)



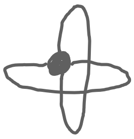
(12)

figure from [Hauru2018]

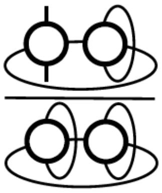
(ignore red shading)

Iterate until $T^{(s)}$ converges

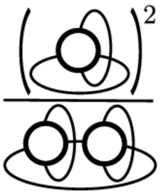
(reaches fixed point)

$Z = \mathrm{Tr}\, T^{(\infty)} =$  (13)

Structure of T^{∞} can be used to characterize different phases [Gu2009].

Proxy for thermal density matrix: $\Gamma =$  $\Rightarrow$ eigenvalues λ_α (14)

von Neumann entropy: $S = -\sum_\alpha |\lambda_\alpha| \log(|\lambda_\alpha|)$ (15)

Degeneracy counter: $X =$  has different values in trivial or non-trivial phases (16)

TRG has issues:	does not fully remove local loop correlations (see [Hauru2018])
	computing 'environment' of given site involve tracking all layers of the iteration scheme

2. 2D contractions via Variational Uniform Matrix Product States (VUMPS)

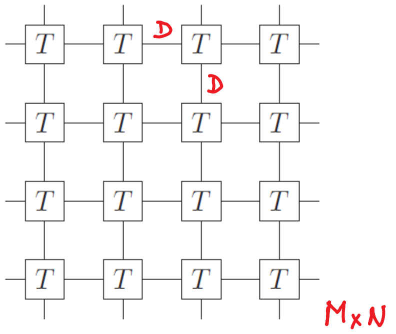
[Fishman2018]

TRG-I.2

Goal: contract $M \times N$ tensor network (for given T); ultimate take $N \times M \to \infty \times \infty$

Partition function:

$Z_{M,N} :=$

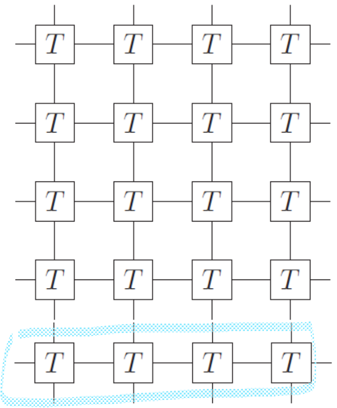


$=: (\kappa^N)^M$ (1)

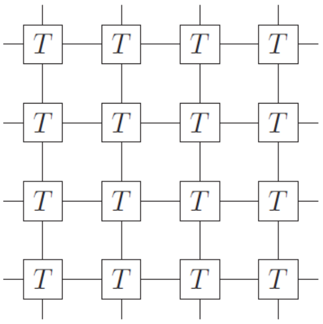
κ = partition function per site

each row contributes a factor κ^N

$Z_{M+1,N} =$



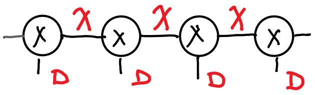
$\approx Z_{M,N} \cdot \kappa^N =$



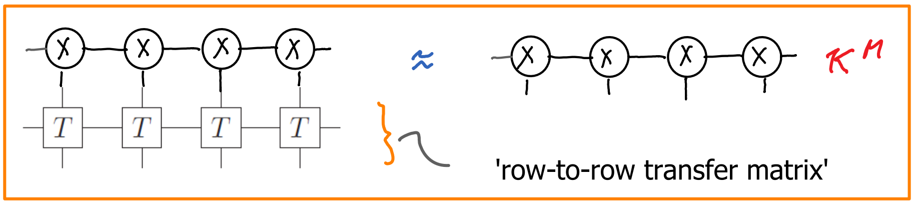
$\cdot \kappa^N$ (2)

[$\approx$ becomes $=$ for $M, N \to \infty$]

In limit $M \to \infty$, represent $Z_{M,N}$ by an 'upper boundary MPS':



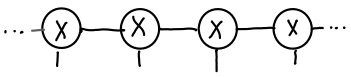
Then:



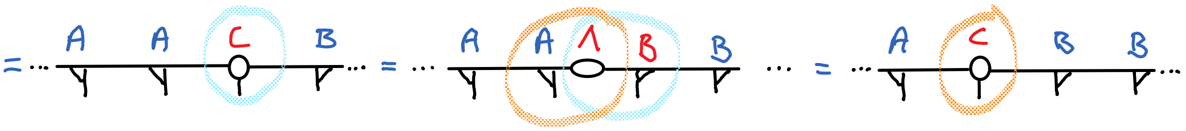
'row-to-row transfer matrix'

'fixed-point condition' (3)

In limit, $N \to \infty$,

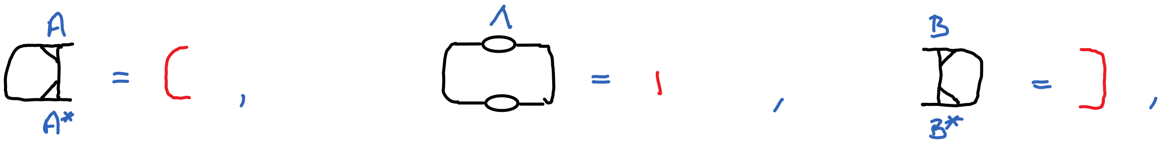


is translationally invariant. Express it in canonical form:



(4)

with



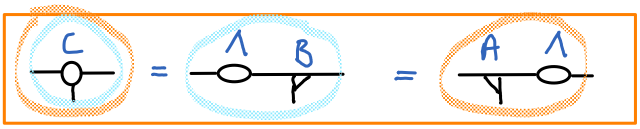
left-normalization

overall normalization

right-normalization

(5)

while C, Λ satisfy the 'gauge conditions':

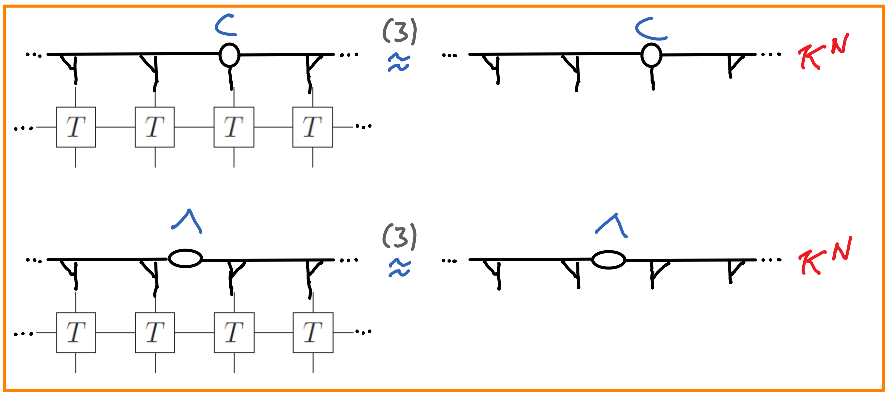


. (6)

which must hold on all sites.

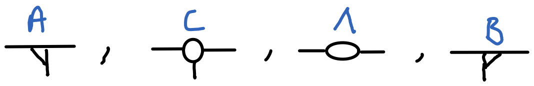
Fixed-point condition (3) implies:

Similarly:



(7)

(8)

Given T , (6,7,8) are to be solved for 

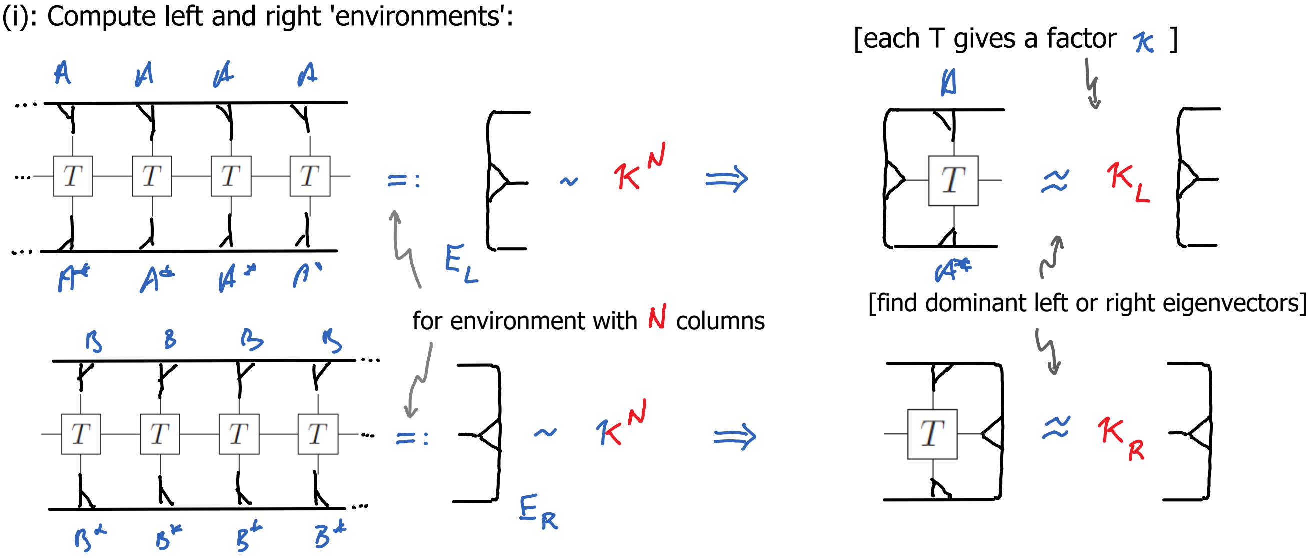
So contraction of infinite tensor network has been reduced to self-consistent solution of four equations!

(6,7,8) have the same structure as when finding ground state of infinite uniform system.

So, solution strategy developed for 'variational uniform matrix product states' (VUMPS) applies:

Repeat following three steps until convergence [with A, C, Λ, B from previous iteration as input]:

(i): Compute left and right 'environments':



[each T gives a factor κ]

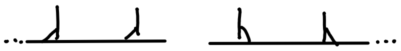
for environment with N columns

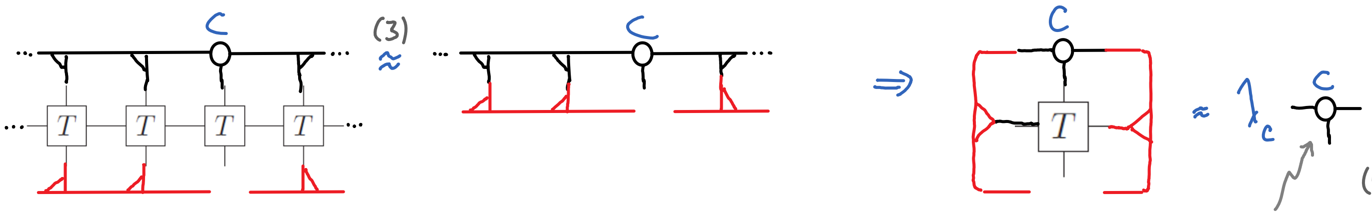
[find dominant left or right eigenvectors]

(9)

(10)

(ii) Solve for central tensor and bond tensor:

(7) contracted with  and expressed through environmental tensors, implies:

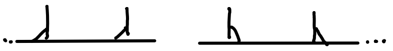


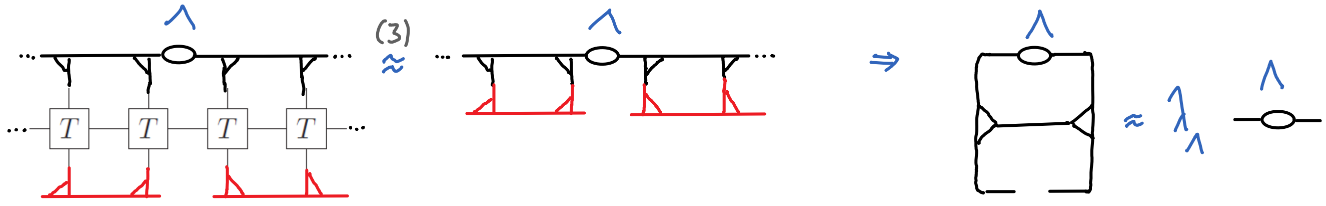
(11)

[find dominant left eigenvector]

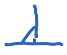
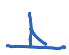
(8) contracted with

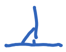
[find dominant left eigenvector]

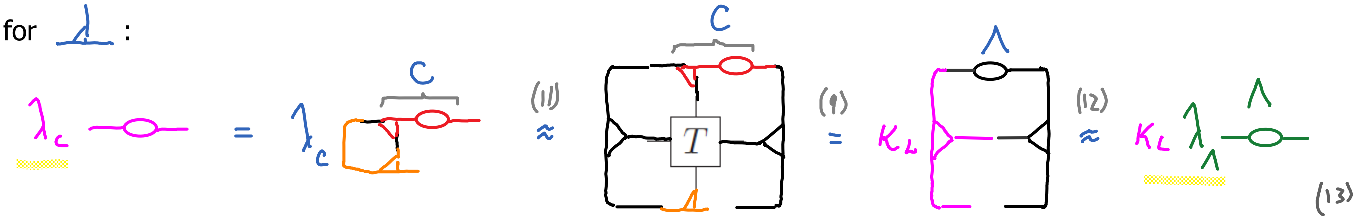
(8), contracted with  and expressed through environmental tensors, implies:

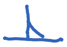


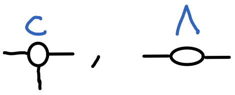
[find dominant right eigenvector]

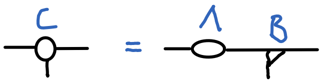
At or near fixed point: $\lambda_C \approx \lambda_\Lambda K_L \approx \lambda_\Lambda K_R$ [this follows by contracting (11) with  or ]

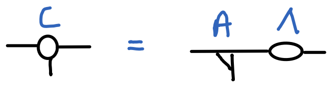
E.g. for  :



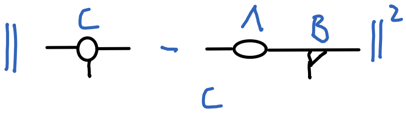
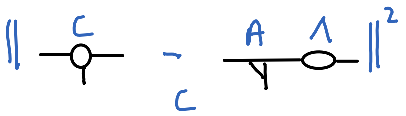
and similarly for  (check yourself!)

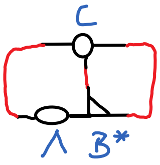
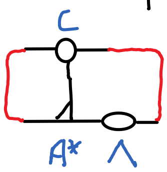
(iii) From  found in (ii), find new that best satisfy (6),

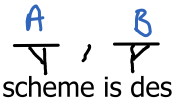


and


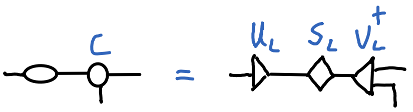
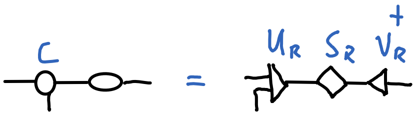
(14)

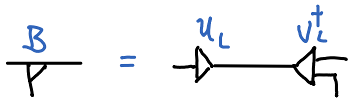
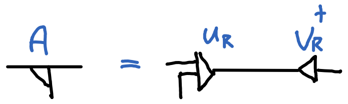
i.e. that minimize  and  (15)

i.e. that maximize  and  (15)

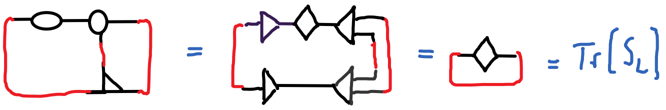
subject to the isometry conditions (5) on  . This is a 'constrained optimization' (i.e. tricky!) problem.
A sophisticated 'Riemannian optimization' scheme is described in [Hauru2021], [Li2023]
In [Fishman2018], it was treated the following 'pragmatic' way, that works reasonably well but is not optimal:

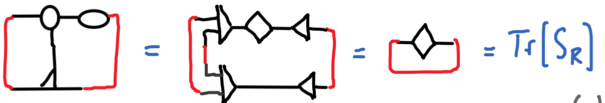
To maximize (15) subject to constraints (5),

do SVDs  , and  (16)

and choose new  , and  (17)

Reason: if we then insert (16) and (17) into (15), we obtain simply a sum over all singular values
(the largest number one can hope to get from such a contraction):



and


(18)

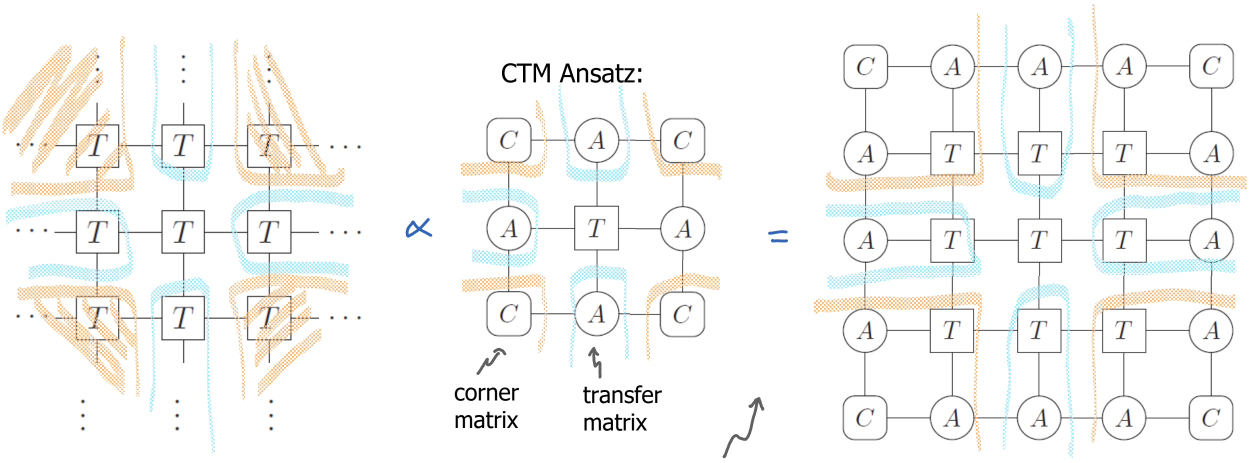
Repeat (i), (ii), (iii) until convergence, measured, e.g., by change in singular values of Λ .

3. Fixed Point Corner Transfer Matrix (FPCM)

following [Fishman18]

TRG-I.3

Ansatz: express infinite 2D network through finite number of tensors: (assume reflection symmetry)

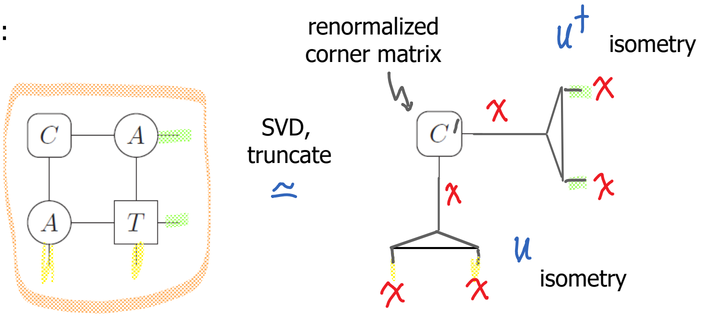


fixed point condition

(1)

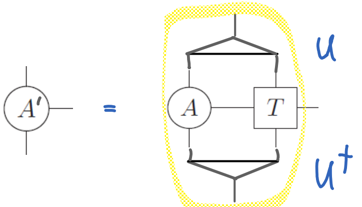
Iteratively following two 'renormalization 'steps:

(i) SVD the 'expanded corner' to obtain renormalized corner and projectors:



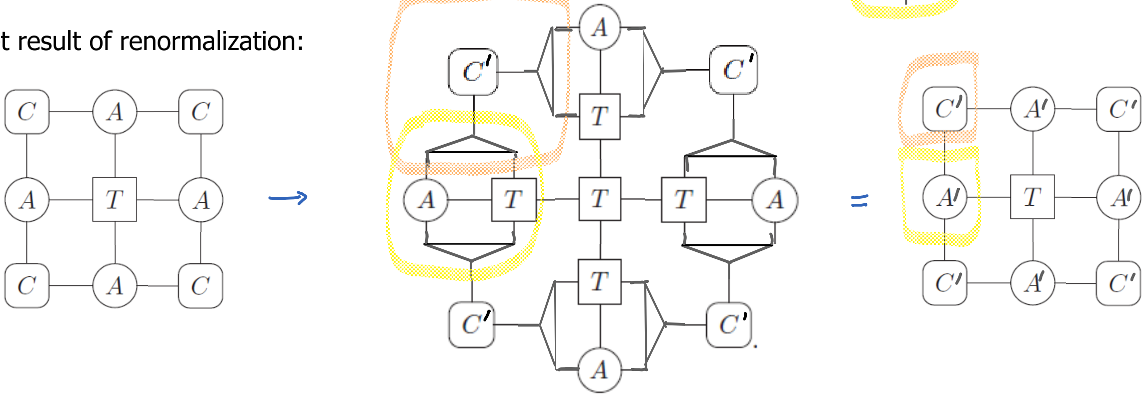
(2)

(ii) Use projectors to obtain renormalized transfer matrix:



(3)

Net result of renormalization:

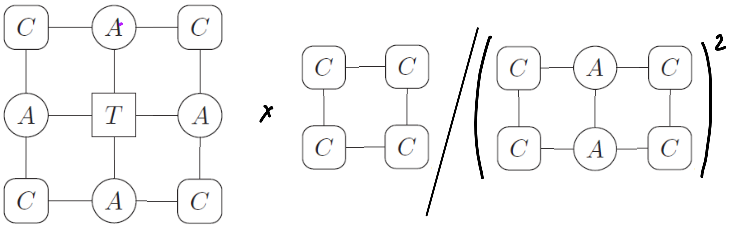


(4)

Iterate until convergence, i.e. until $A' = A$, $C' = C$

Partition function per site:

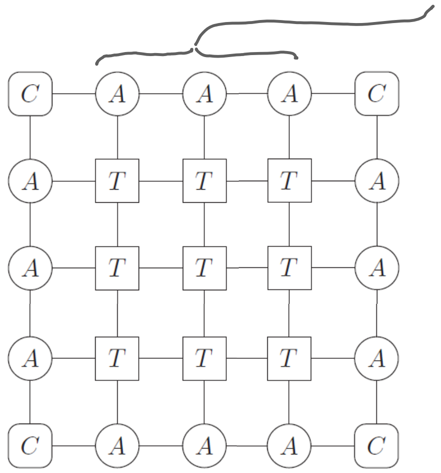
$\kappa :=$



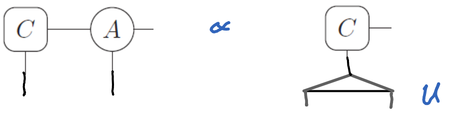
(5)

'divide out four A's and eight C's

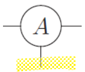
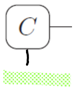
Enforce translational symmetry on boundary MPS built from A s:



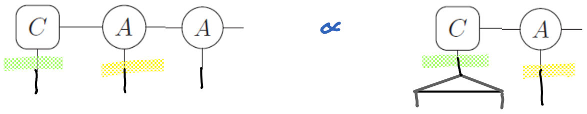
To this end, impose the 'pulling through' gauge condition:



(6)

then each  has the same left neighbor, 

for example:

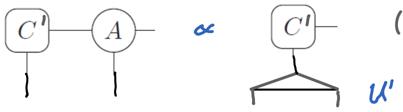


(7)

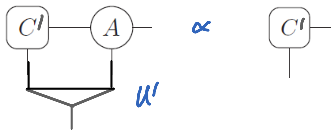
FPCM imposes this gauge via fixed point conditions, by iterating the following two steps until convergence:

i.e. until $A' = A$, $C' = C$

(i) Given A, find isometry U' and (symmetric) C' approximately satisfying 'pulling through' gauge condition: [cf. (6)]



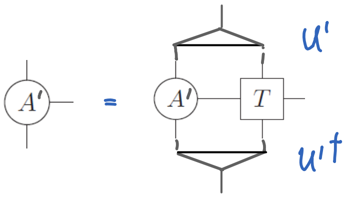
(8) or equivalently



(9)

(how to achieve this: see below)

(ii) Use A and new U' to find new A' by solving the following fixed-point equation [with Arnoldi method]:

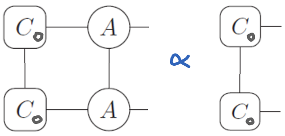


(It may be necessary to symmetrize A' by hand.)

(10)

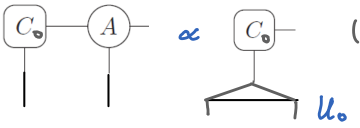
Details for step (i):

(i.a) Compute initial C_0^2 as dominant (normalized) eigenvector of $A^\dagger A$: As eigenvector of a transfer matrix, this C_0^2 is positive and Hermitian (up to numerical errors). Obtain C_0 as its square root (e.g. via an eigendecomposition). However, this C_0 is not yet properly gauged.



(11)

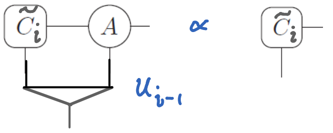
(i.b) Obtain U_0 through polar decomposition of $C_0 A = U_0 C_0'$: Pulling-through condition (1) would hold if C_0' were equal to C_0. Conversely, $\|C_0 - C_0'\|$ quantifies the degree of violation of Eq. (6).



(12)

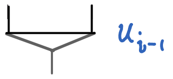
Obtain C_i and U_i from A and U_{i-1} by iterating following two steps until convergence (starting from $i=1$):

(i.c) Obtain $\tilde{C}_i$ as dominant (normalized) eigenvector of 'mixed transfer matrix' $U_{i-1}^\dagger A$ [cf. Eq. (9)],

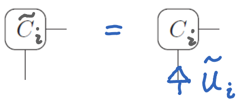


(13)

'mixed transfer matrix' $U^{\dagger}_{i-1} A$ [cf. Eq. (9)],



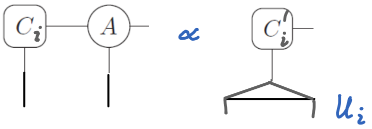
and extract a positive hermitian factor C_i from $\tilde{C}_i$ using a polar decomposition: $\tilde{C}_i = \tilde{U}_i C_i$



(14)

(i.d) Obtain U_i through polar decomposition of $C_i A = U_i C_i'$:

When $\| C_i - C_i' \|$ is small enough, terminate loop, and set $C = C_i$, $U' = U_i$.



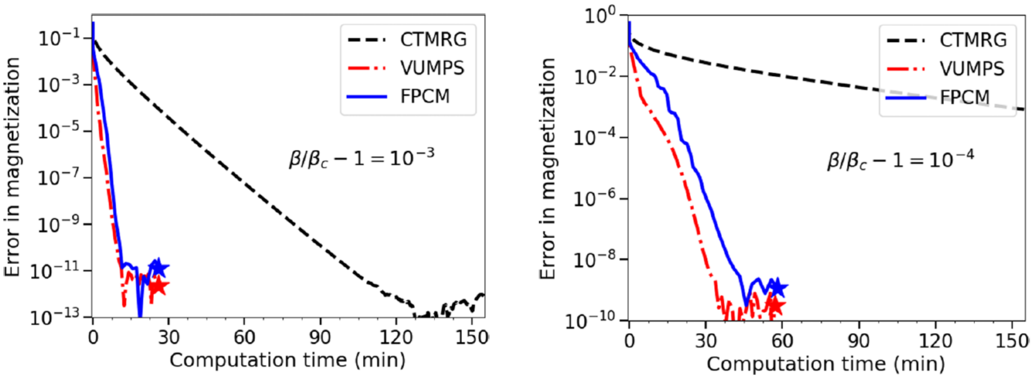
(15)

Technical remark: a polar decomposition can be obtained via SVD:

$$A = USV^{\dagger} = \underbrace{(USU^{\dagger})}_{\text{hermitian, positive}}(UV^{\dagger}) = (UV^{\dagger})\underbrace{(VSV^{\dagger})}_{\text{hermitian, positive}} \quad (16)$$

Results:
2D classical
Ising model:

Main message:
fixed-point methods
VUMPS and FPCM
are faster than
CTMRG!



4.TRG for 2D classical lattice models (optional)

[Levin2007] Levin, Nave

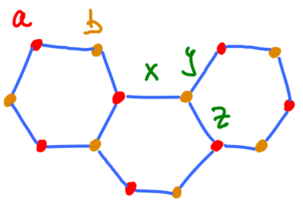
TRG-I.4

Goal: compute partition function of 2D classical model.

Strategy: Express partition function as 2D tensor network, contract it by coarse-graining procedure.

Example 2D classical Ising model on honeycomb lattice [Zhao2010, Sec. II.B]

Honeycomb lattice is bipartite:



unit cell contains two sites, labeled a, b

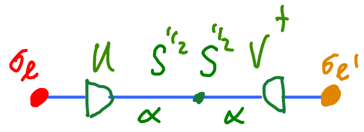
three bond directions: x, y, z

Hamiltonian: $H = - \sum_{\langle l,l' \rangle} \sigma_l \sigma_{l'}$, $\sigma_l = \pm 1$ Ising variable (1)

nearest neighbors, with $l \in a$, $l' \in b$

Partition function: $$Z = \sum_{\{\vec{\sigma}\}} e^{-\beta H} = \sum_{\{\vec{\sigma}\}} \prod_{\langle l,l' \rangle} \underbrace{e^{\beta \sigma_l \sigma_{l'}}}_{:= \Theta_{ll'}} = \sum_{\{\vec{\sigma}\}} \prod_{\langle l,l' \rangle} \Theta_{ll'} \qquad (2)$$

'Factorize' the dependence on σ_l and $\sigma_{l'}$ by performing an SVD:



$$\Theta_{\underset{a}{\underset{\cap}{l}}\,\underset{b}{\underset{\cap}{l'}}} = \sum_{\alpha \in 1,2} \underbrace{U_{\sigma_l \alpha} (S_\alpha)^{1/2}}_{:= \underset{(2\times 2)}{Q^a_{\sigma_l \alpha}}} \underbrace{(S_\alpha)^{1/2} V^\dagger_{\alpha \sigma_{l'}}}_{\underset{(2\times 2)}{Q^b_{\sigma_{l'} \alpha}}} \qquad (3)$$

matrices

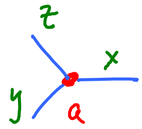
classical model: no need to distinguish upper/lower indices

Advantage of this representation: spin dependence has been factorized.

Price to pay: additional 2-dimensional bond index, $\alpha \in \{1,2\}$ has been introduced.

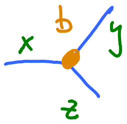
Group all Q's connected to site l on a -lattice, and sum over σ_l , for given $x, y, z \in \{1,2\}$

$$T^a_{[l]xyz} = \sum_{\sigma_l} Q^a_{\sigma_l x} Q^a_{\sigma_l y} Q^a_{\sigma_l z} \qquad (4)$$

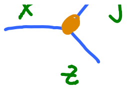


Ditto for site l' on b -lattice, sum over $\sigma_{l'}$:

$$T^b_{[l']xyz} = \sum_{\sigma_{l'}} Q^b_{\sigma_{l'} x} Q^b_{\sigma_{l'} y} Q^b_{\sigma_{l'} z} \qquad (5)$$



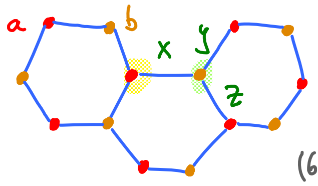
$$T_{[\ell']xyz} = \sum_{\sigma_{\ell'}} Q_{\sigma_{\ell'}x}\, Q_{\sigma_{\ell'}y}\, Q_{\sigma_{\ell'}z} \qquad (5)$$



Then partition function takes the form

$$Z = \sum_{\{\vec\sigma\}} \prod_{\langle \ell,\ell'\rangle} \Theta_{\ell\ell'} = \mathrm{Tr} \prod_{\ell\in a,\, \ell'\in b} T^a_{[\ell]x_\ell y_\ell z_\ell}\, T^b_{[\ell']x_{\ell'} y_{\ell'} z_{\ell'}} \qquad (6)$$

sum over virtual indices on $\underbrace{\text{all (suitably contr}}$acted) nearest-neighbor bonds

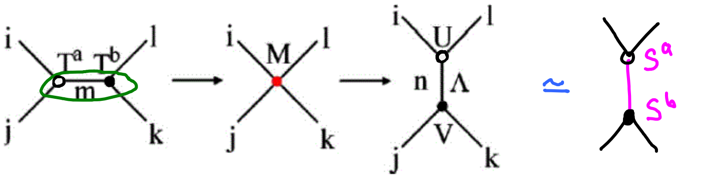


All statistical physics models with short-range interactions can be expressed as tensor network models, i.e.

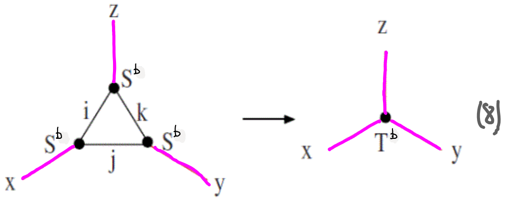
$Z = \mathrm{Tr} \prod T$ (for more examples, see [Zhao2010, section II]).

Contract out the tensor network by course-graining [Levin2007]

'rewire': switch from T-vertices with external leg pairings (i,j), (l,k) to S-vertices with pairings (i,l), (j,k):

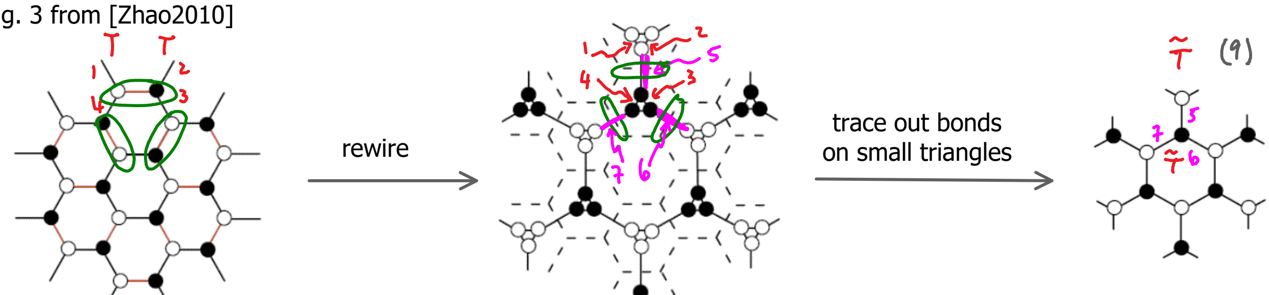


reshape SVD truncate to bond dimension D_{cut}



coarse-grain:
trace out bonds on small triangle
to define updated T^b -tensor

Fig. 3 from [Zhao2010]

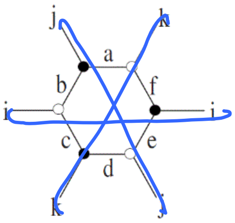


dashed lines depict original bonds,
solid lines depict rewired bonds

coarse-grained
lattice

Iterate this procedure, thereby coarse-graining lattice step by step, until T^a, T^b reach fixed point values,

T^{a*}, T^{b*}. Use these to compute partition function via $Z =$



and from there the free energy per spin, $F = -\frac{1}{N\beta}\ln Z$

and the magnetization, etc.

5. TRG for quantum lattice models (optional)

[Jiang2008] Jiang, Weng, Xiang

TRG-I.5

Goal: compute ground state of 2D quantum lattice model

Strategy: iterative projection via $e^{-H\tau}$, compress by 'simple update';

compute $\langle\psi|\psi\rangle$ and $\langle\psi|\hat{O}|\psi\rangle$ using TRG of Levin & Nave.

Model: $S=1/2$ Heisenberg on honeycomb lattice.

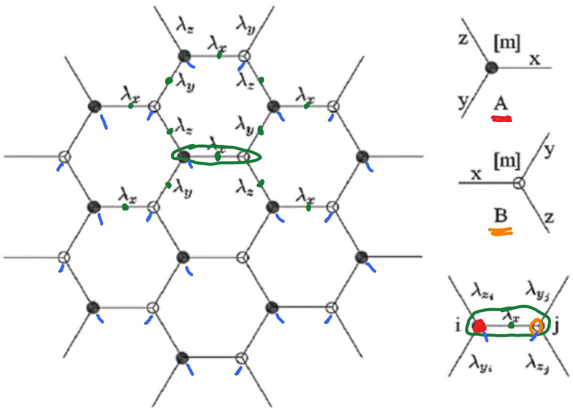
iPEPS-type tensor network Ansatz for ground state:

$$|\psi\rangle = \mathrm{Tr}\prod_{l\in b}\prod_{l'\in w} \overbrace{\lambda_{x_l}\lambda_{y_l}\lambda_{z_l}}^{\text{weight factors associated with bc}} \underbrace{A^{\sigma_l}_{x_l y_l z_l} B^{\sigma_{l'}}_{x_{l'} y_{l'} z_{l'}}}_{\text{tensors associated with vertices}} |\sigma_l \sigma_{l'}\rangle \qquad (1)$$

black

white

vertices: A or B tensors
bonds: diagonal λ-tensors (weights)

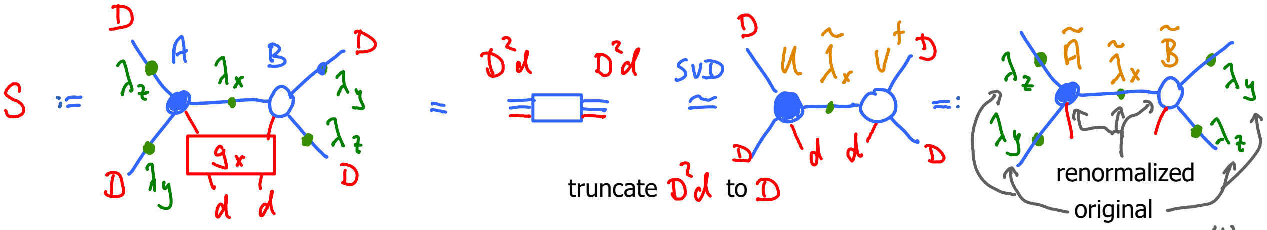


Ground state projection via simple update

$$H = H_x + H_y + H_z \qquad \text{(living on x, y, or z bonds)} \qquad (2)$$

Suzuki-Trotter: $$e^{-H\tau} \simeq \underbrace{e^{-H_x\tau}}_{g_x}\underbrace{e^{-H_y\tau}}_{g_y}\underbrace{e^{-H_z\tau}}_{g_z} \qquad (3)$$

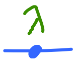
Sequentially update x, y, z bonds using these three gates.



(4)

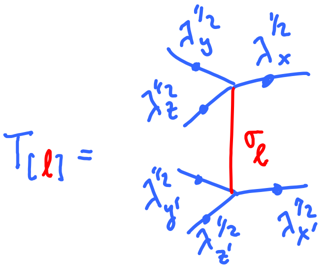
$$S \simeq U\tilde{\lambda}_x V^\dagger, \qquad \tilde{A} = \lambda_z^{-1}\lambda_y^{-1}U, \qquad \tilde{B} = \lambda_z^{-1}\lambda_y^{-1}V^\dagger \qquad (5)$$

SVD, truncate

'simple update': outer legs of contain  , which account for the 'environment' of

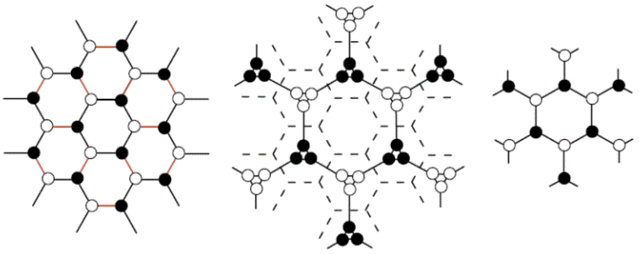
in mean-field fashion. Without including these λ factors in definition of S, procedure does not converge.

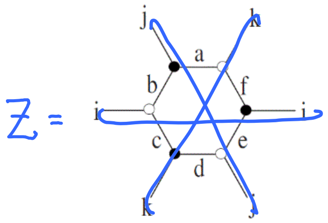
- Similarly update y and z bonds. This concludes one iteration.
- Iterate simple update many times.
- Start with $\tau \sim 10^{-3}$, gradually reduce it to $\tau \sim 10^{-5}$.
- Number of iterations needed until convergence: $10^5 - 10^6$



$\langle \Psi | \Psi \rangle$ is a double-layer tensor network.

Use TRG (á la Levin & Nave) to contract bond indices of double-layer network:





Start with a finite system, and iterate until only six sites are left; then trace out final bond indices.

Results [Jiang2008]

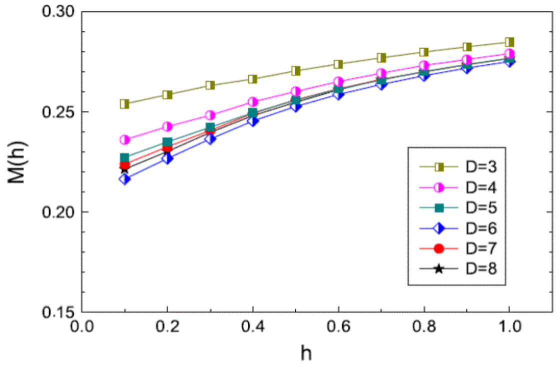


FIG. 5 (color online). The staggered magnetization $M(h)$ as a function of the staggered magnetic field, at different D.

TABLE II. Comparison of our results with those obtained by other approaches for the ground state energy per site E and the staggered magnetization M of the Heisenberg model with $h = 0$.

Method	E	M
Spin wave [12]	−0.5489	0.24
Series expansion [13]	−0.5443	0.27
Monte Carlo [14]	−0.5450	0.22
Ours $D = 8$	−0.5506	0.21 ± 0.01

6. Second renormalization (SRG) of tensor network states (optional)

[Xie2009],
more details: [Zhao2010]

TRG-I.6

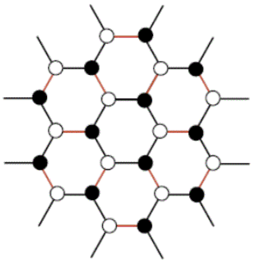
Goal: include influence of environment when doing update 'global optimization', 'full update'.

Two applications: (i) partition function of classical 2D models
(ii) 2D quantum ground states

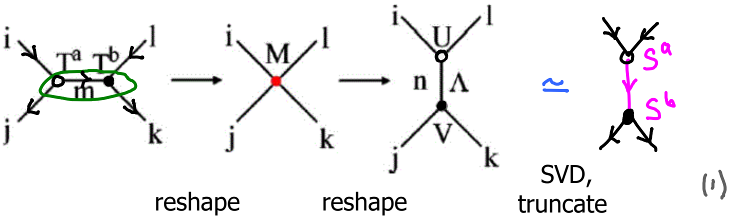
(i) Classical tensor network model

(TRG.1.6)

$$Z = \mathrm{Tr} \prod_{l\in a,\, l'\in b} T^a_{[l]x_l y_l z_l} \, T^b_{[l']x_{l'} y_{l'} z_{l'}}$$



rewire: M^{li}_{kj}

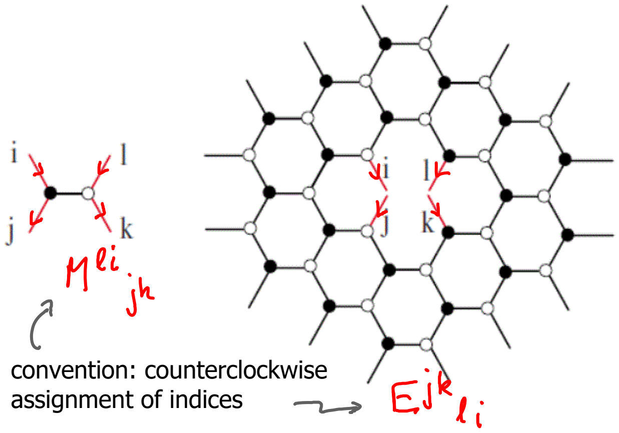


SVD minimizes truncation error for rewiring M. However, we should minimize truncation error of Z.

Renormalize environment

Partition function:

$$Z = \mathrm{Tr}\, ME = \sum_{ijkl} M^{li}_{\cdot jk} E^{jk}_{\cdot li} \qquad (2)$$



convention: counterclockwise assignment of indices

Goal: minimize truncation error of Z.

Strategy:

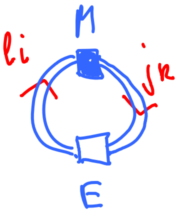
(i) Compute E
- (a) cheap mean-field approach ('single update')
- (b) on finite lattices
- (c) more expensive forward/backward TRG ('full update')

(ii) Do SVD on ME, Let's discuss (ii) first.

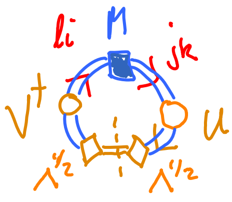
Minimize truncation error of ME

[Zhao2010, Sec. III.B]

$Z = M^{li}_{\cdot jk} E^{jk}_{\cdot li} =$



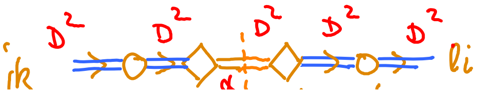
$\overset{\text{SVD}}{=}$



$= \mathrm{Tr}\, \tilde{M} \simeq \mathrm{Tr}\, \tilde{\tilde{M}}$ (3)

SVD
truncate

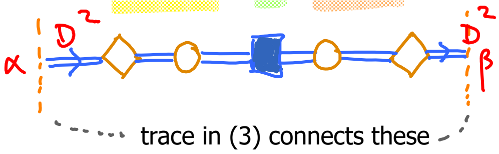
with $E \overset{\text{SVD}}{=} U \Lambda V^\dagger$

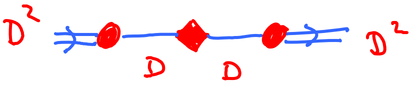


(4)

$U \quad \Lambda^{1/2} \quad \Lambda^{1/2} \quad V^\dagger$

and $\tilde{M} \equiv \Lambda^{1/2} V^\dagger M U \Lambda^{1/2}$ $\quad \overset{\text{SVD}}{\simeq} \quad \tilde{U} \hat{\Lambda} \tilde{V}^\dagger = \tilde{\tilde{M}}$ (5)





truncate from D^2 back to D (6)

Since $Z = \mathrm{Tr}\, \tilde{M}$, this truncation directly controls error in partition function!

It knows not only about M, but also about its environment, via $U, \Lambda, V^\dagger$

Now express M in terms of truncated objects, $\tilde{U}, \tilde{\Lambda}, \tilde{V}^\dagger$

To this end, first invert relation between M and $\tilde{\tilde{M}}$, using $\quad U^\dagger U = V^\dagger V = 1$

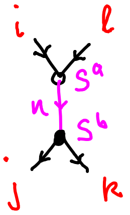
$$M \overset{(5)}{=} V \Lambda^{-1/2} \overbrace{\tilde{\tilde{M}}}^{} \Lambda^{-1/2} U^\dagger$$

then insert truncated version of $\tilde{\tilde{M}}$:

$$\underbrace{V \Lambda^{-1/2} \, \tilde{U} \tilde{\Lambda}^{1/2}}_{} \underbrace{\tilde{\Lambda}^{1/2} \tilde{V}^\dagger \, \Lambda^{-1/2} U^\dagger}_{}$$

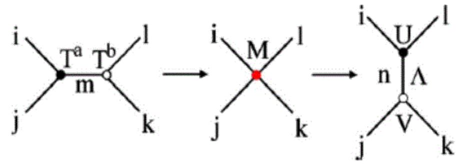
and write as product of two vertices: $= S^a \quad S^b$

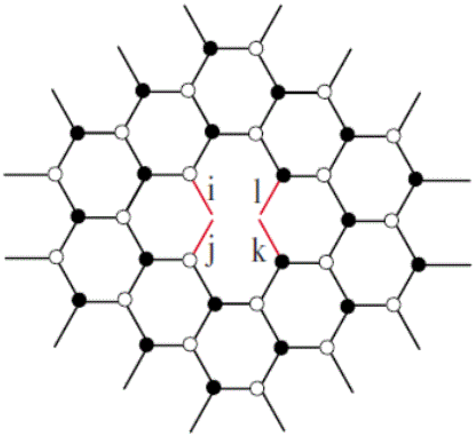
with indices: $M^{li}{}_{jk} = S^{a,li}{}_{n} \; S^{b,n}{}_{jk}$



Now we return to (i): actually computing the environment

<u>(a) Computing environment tensor E using simple update (mean-field approach)</u> [Xie2009]



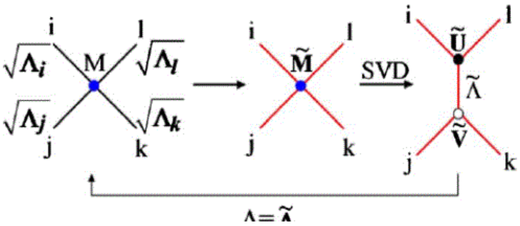


$M = U \Lambda V^\dagger$ defines the 'singular bond vector' Λ, which measures entanglement between two sites. It can be used directly to obtain a cheap, mean-field approximation of environment ('simple update'):

- Take $E \simeq \sqrt{\Lambda_i \Lambda_j \Lambda_k \Lambda_l}$

- Compute $\tilde{M}$, then do SVD: $\tilde{M} = \tilde{U} \tilde{\Lambda} \tilde{V}^\dagger$

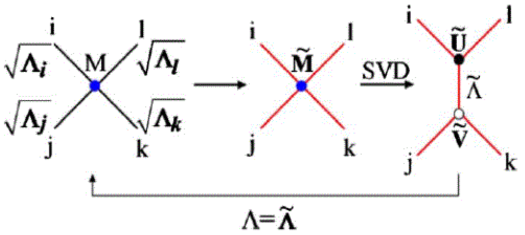
new bond vector



- Take $E \simeq \sqrt{\Lambda_i \Lambda_j \Lambda_k \Lambda_l}$

- Compute $\tilde{M}$, then do SVD: $\tilde{M} = \tilde{U}\tilde{\Lambda}\tilde{V}^{\dagger}$

new bond vector

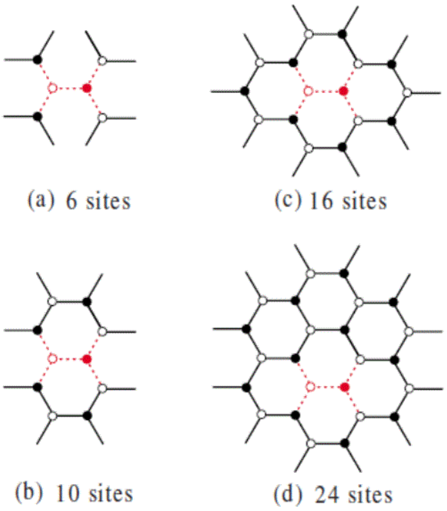


- Use new $\Lambda = \tilde{\Lambda}$ to recalculate E, $\tilde{M}$, $\tilde{\Lambda}$, etc.

- Iterate until convergence (typically 2 to 3 iterations suffice; near critical point, more are needed).

(b) Computing environment tensor E using finite lattices

$$\delta f(T) = 1 - \frac{f(T)}{f_{exact}(T)}$$



(a) 6 sites (c) 16 sites

(b) 10 sites (d) 24 sites

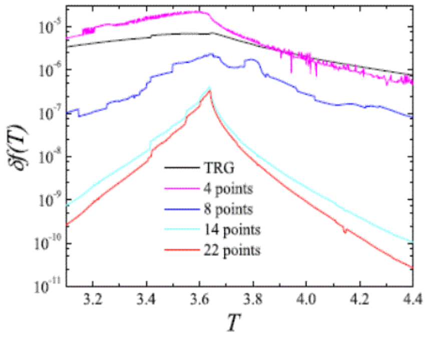


FIG. 10. (Color online) Relative errors of the free energy for the Ising model on a triangular lattice obtained by considering the second renormalization effect from four finite environment lattices which contains 4, 8, 14, and 22 sites, respectively. The configurations of these environments are shown in Fig. 9. The TRG result is also shown for comparison.

Including even just a few environmental sites already leads to big improvements!

(c) Computing environment tensor E using TRG [Zhao2010]

'Forward iteration':

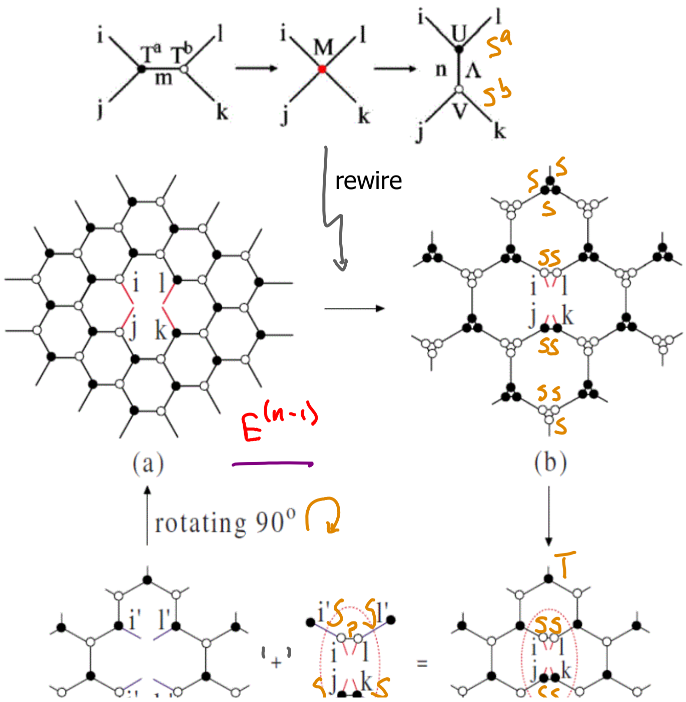
(a) → (b): Rewire environment E^{jk}_{li}
using data at iteration n:

$$T^{[n]}T^{[n]} = M^{[n]} = U^{[n]}\Lambda^{[n]}V^{\dagger[n]}$$

(b) → (c): Trace out small triangles, $T = SSS$
four S are left over

(c) → (d) + (e): Identify new environment

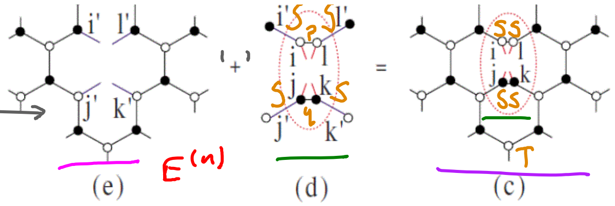
(e) looks same as (a), only rotated by 90 degrees, and rescaled.



Iteration relation expressing

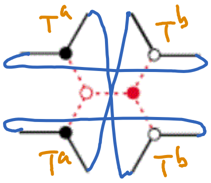
degrees, and rescaled.

Iteration relation expressing old through new environment:



$$\underline{E^{[n-1]}_{li,jk}} = \underline{E^{(n]}_{l'i',j'k'}}\, \underline{S^a_{lpl'}\, S^a_{ii'p}\, S^b_{qj'j}\, S^b_{k'qk}}$$

- Start with a very large but finite number of sites.
- Iterate until only 4 environmental sites are left:
- Compute final environment, $E^{(N)}$, by tracing out open indices:



$$E^{(N)} = \mathrm{Tr}\; T^a T^b T^a T^b$$

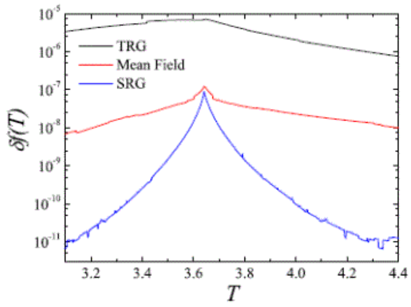
'Backward iteration':

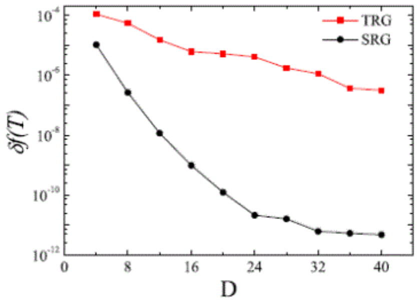
- Start from current values of tensors T^a, T^b and bond vectors Λ .
- Use them to compute $E^{(N)}$, $E^{(N-1)}$, etc., all the way back to $E^{(0)} = E$ = desired result.

This completes step (i). Now go to step (ii), compute $\tilde{M}, \tilde{\Lambda}, M$, and iterate, until $\tilde{\Lambda}$ have converged.

Results for SRG (2nd renormalization) for classical 2D system

Ising model on triangular lattice:





mean field

SRG

FIG. 12. (Color online) Comparison of the relative error of the free energy for the Ising model on triangular lattices obtained using TRG (red), the mean-field approximated SRG (blue), and the SRG (black) methods with D_{cut}=24, respectively. The critical temperature is T_c=4/ln 3.

FIG. 13. (Color online) The relative error of the free energy as a function of the truncation dimension D_{cut} for the Ising model on triangular lattices obtained using the TRG (black) and SRG (blue), respectively. T=3.2.

critical state is hardest to simulate

error drops with increasing D much more quickly for SRG than TRG

Results for SRG (2nd renormalization) for quantum ground state search

Optimize by imaginary time evolution; contractions performed using SRG.

Compute expectation values such as $\langle \psi | \psi \rangle$, $\langle \psi | \hat{O} | \psi \rangle$ using SRG, too.

[Xie2009] : Heisenberg on honeycomb

SRG yields more stable results than TRG!

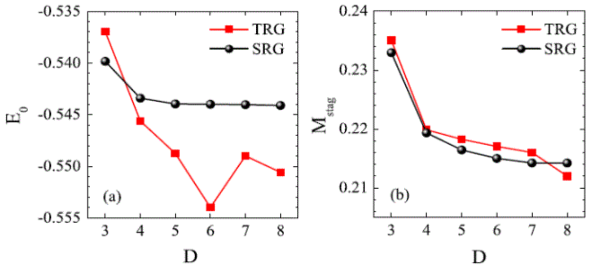


FIG. 5 (color online). (a) The ground state energy per site E_0 and (b) the staggered magnetization M_{stag} as functions of the bond degrees of freedom D on honeycomb lattices.

[Zhao2010]

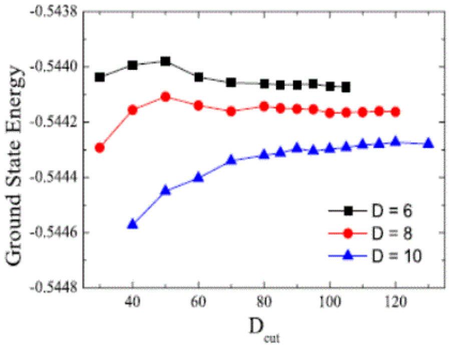


FIG. 19. (Color online) The SRG result of the ground-state energy as a function of the truncation dimension D_{cut} for the Heisenberg model on a honeycomb lattice. D is the bond dimension of the wave function.

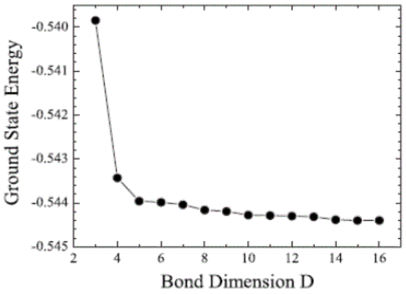


FIG. 20. The ground-state energy of the Heisenberg model on a honeycomb lattice as a function of the bond dimension D obtained by the SRG with D_{cut}=130.

$E^{SRG} = -0.54440 \qquad E^{QMC} = -0.54455(20)$

Energy does not decrease with D_cut, because imaginary time-evolution / SRG is not variational!

7. Core tensor renormalization group (optional) [Lan2019]

TRG-I.7

Goal: reduce computational cost of TRG from $\mathcal{O}(\chi^6)$ to $\mathcal{O}(\chi^4)$

Strategy: shrink lattice linearly rather than exponentially with each coarse-graining step.

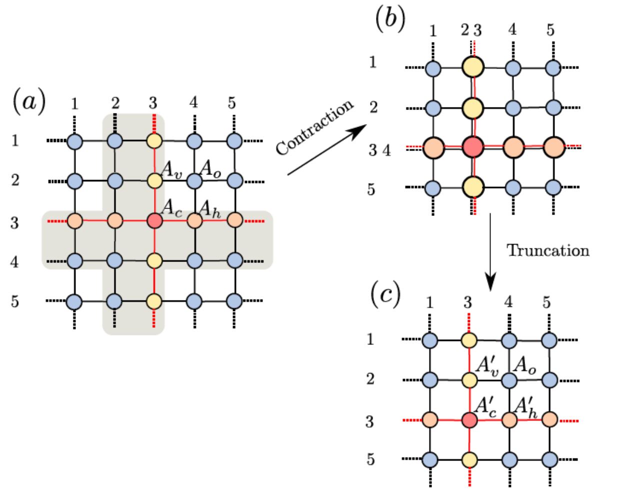


FIG. 1. A depiction of the CTRG iteration, which maps an $L \times L$ lattice of tensors to an $(L-1) \times (L-1)$ lattice. (a) The initial network is everywhere composed of copies of the bulk tensor A_0, except for a single 'core' row and column containing tensors $\{A_c, A_h, A_v\}$ as indicated. (b) An adjacent row and column of the network has been contracted into the core row/column, thus growing the index dimension of the core tensors. (c) The indices of the core tensors are truncated to dimension χ, as to obtain new core tensors $\{A'_c, A'_h, A'_v\}$.

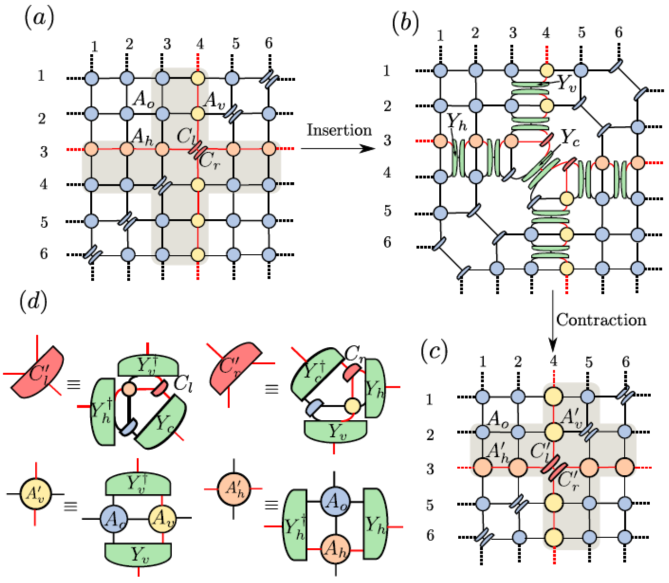


FIG. 2. At iteration of the CTRG algorithm. (a) The initial square lattice network is homogeneous except for a core row/column which contains core tensors $\{A_v, A_h, C_l, C_r\}$ and a diagonal line through the core along in which the bulk tensors have been decomposed into products of 3-index tensors. (b) Pairs of isometries $\{Y_v, Y_h, Y_c\}$ and their conjugates have been inserted into the core row/column of the network. (c) Isometries are contracted with their neighboring tensors, effectively absorbing a bulk row/column into the core row/column, as to produce new core tensors $\{A'_v, A'_h, C'_l, C'_r\}$. (d) Definitions of the new core tensors.

$$M = USV^\dagger \approx usv^\dagger \Rightarrow MM^\dagger = US^2U^\dagger$$

$$uu^\dagger M = uu^\dagger USV^\dagger = usv^\dagger$$

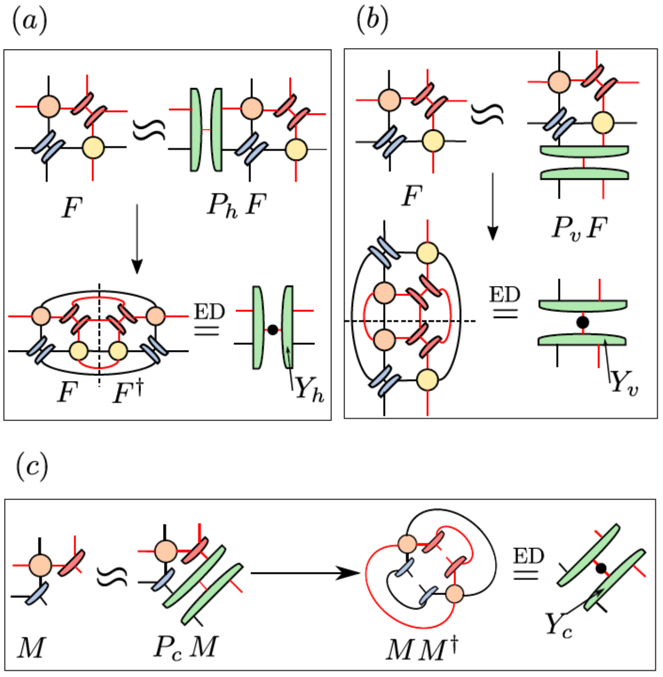


FIG. 3. (a) The projector $P_h \equiv Y_h Y_h^\dagger$ should be chosen to (approximately) leave invariant the network F, which is the network formed from the central tensors of the initial lattice in Fig. 2(a). The optimal isometry Y_h is formed by taking the eigenvalue decomposition (ED) of $FF^\dagger$, when F is viewed as a matrix between its left two and remaining indices, and truncating to retain only the χ dominant eigenvectors. (b) The optimal isometry Y_v is obtained from the ED of $FF^\dagger$, when F is viewed as a matrix between its bottom two and remaining indices. (c) The optimal isometry Y_c is obtained from the ED of $MM^\dagger$, when M is half of the F network.

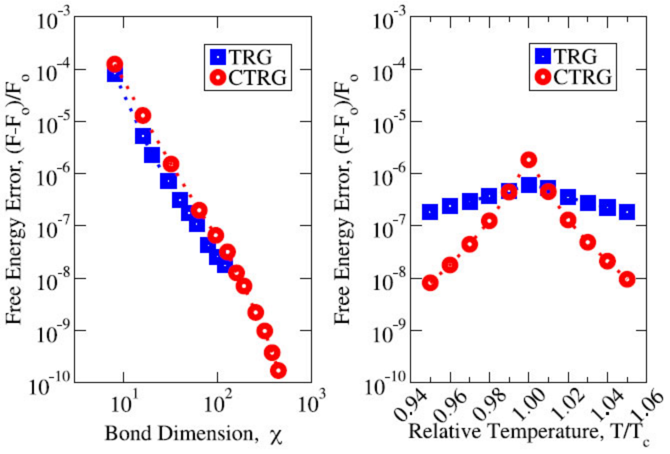


FIG. 5. (a) A comparison of the accuracy of the free energy density produced by TRG and CTRG for the Ising model on an infinite strip of width $L = 128$ sites at critical temperature. Both methods produce comparable accuracy for the same bond dimension χ, with TRG giving only slightly more accurate energies. (b) Comparison between TRG and CTRG for accuracy of the free energy density as a function of temperature with fixed bond dimension $\chi = 30$.