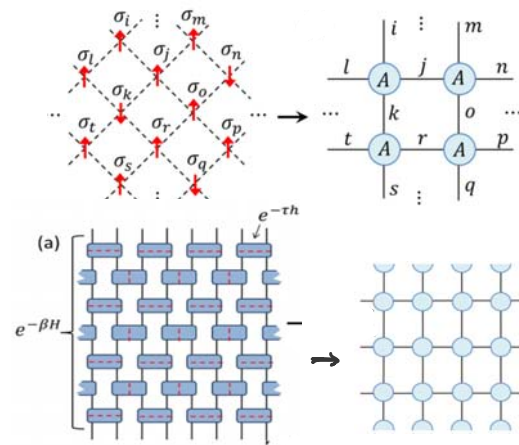[credit for course materials: Prof. Jan von Delft]

Tensor renormalization group (TRG)

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] propose 'second renormalization' (SRG), a global approach taking account renormalization of environmental tensor ('full update'). Reduced truncation error significantly.

[Xie2012] Different coarse-graining scheme, using higher-order SVD, employing both local and global optimization schemes.

[Zhao2016] coarse graining on finite lattices.

[Evenbly2019] propose core tensor renormalization group (CTRG) which rescales lattice size linearly (not exponentially), but at much lower cost, $O(\chi^4)$ (rather than $O(\chi^6)$). $\chi =$ bond dim.

1.   TRG for 2D classical lattice models
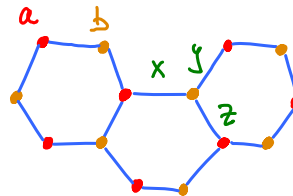[Levin2007]

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 IIB]

Honeycomb lattice

is bipartite:

[unit cell contains two sites, labeled a, b , and three bond directions: x, y, z]

Hamiltonian:

$$H = -\sum_{\langle \ell \ell' \rangle} \sigma_\ell \sigma_{\ell'} \qquad \sigma_\ell = \pm 1 \quad (\pm \text{ Ising variable})$$
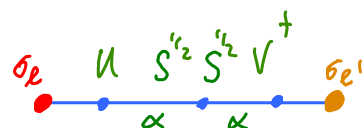
$\underbrace{\langle \ell \ell' \rangle}_{\text{nearest neighbors}}$, $\ell \in a, \ell' \in b$

Partition function:

$$Z = \sum_{\{\sigma\}} e^{-\beta H} = \sum_{\{\sigma\}} \prod_{\langle \ell \ell' \rangle} \underbrace{e^{+\beta \sigma_\ell \sigma_{\ell'}}}_{= \Theta_{\ell \ell'}} = \sum_{\{\sigma\}} \prod_{\langle \ell \ell' \rangle} \Theta_{\ell \ell'}$$

'Factorize' the dependence on $\sigma_\ell + \sigma_{\ell'}$ by performing an SVD:

$$\Theta_{\ell\ell'} = \sum_{\alpha=1,d} \underbrace{U_{\sigma_\ell\alpha}(S_\alpha)^{1/2}}_{\equiv Q^a_{\sigma_\ell\alpha}} \underbrace{(S_\alpha)^{1/2}V^\dagger_{\alpha\sigma_{\ell'}}}_{Q^b_{\sigma_{\ell'}\alpha}}$$

$\ell \downarrow a$ $\ell \downarrow b$

$(d\times d)$ $(d\times d\ \text{matrices})$

→ [Note: in following we will not distinguish upper/lower indices] (all numbers are real)

Advantage of this representation: spin dependence has been factorized.   → σ_ℓ is on a

Price to pay: additional 2D bond index, $\alpha \in \{1,2\}$ has been introduced. $\sigma_{\ell'}$ is on b

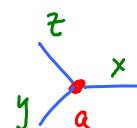Partition function now contains products of Q matrices on all sites. Because of SVD, a given Q matrix is localized to a given site. Hence, the sum over all spin values can be taken before the product and summed over for each site. To accomplish that, we group all Q's connected to some site ℓ on a lattice, and sum over σ_ℓ, for given $x,y,z \in \{1,2\}$ (bond indices)
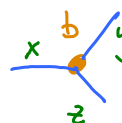
$$T^a_{(\ell)xyz} = \sum_{\sigma_\ell} Q^a_{\sigma_\ell x}\, Q^a_{\sigma_\ell y}\, Q^a_{\sigma_\ell z}$$

bond indices

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

$$T^b_{(\ell')xyz} = \sum_{\sigma_{\ell'}} Q^b_{\sigma_{\ell'} x}\, Q^b_{\sigma_{\ell'} y}\, Q^b_{\sigma_{\ell'} z}$$

Then partition function takes the form

$$Z = \sum_{\{\sigma\}} \prod_{\langle \ell, \ell' \rangle} \Theta_{\ell \ell'}$$

$$= tTr \prod_{\substack{\ell \in a \\ \ell' \in b}} T^a_{(\ell) x_\ell y_\ell z_\ell} \; T^b_{(\ell') x_{\ell'} y_{\ell'} z_{\ell'}}$$

↳ sum over virtual indices on all nearest-neighbor bonds

(6)
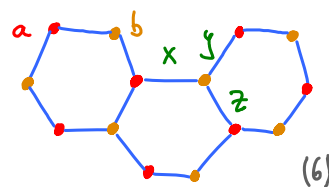
All statistical physics models with short-range interactions can be expressed as tensor

network models, i.e. $\quad Z = Tr \prod \underset{\text{product}}{T}$

[more examples in [Zhao2010, Sec II]

Contract out the tensor network by coarse-graining

'rewire': switch from vertices with external leg pairings (i,j), (l,k) to vertices with pairings
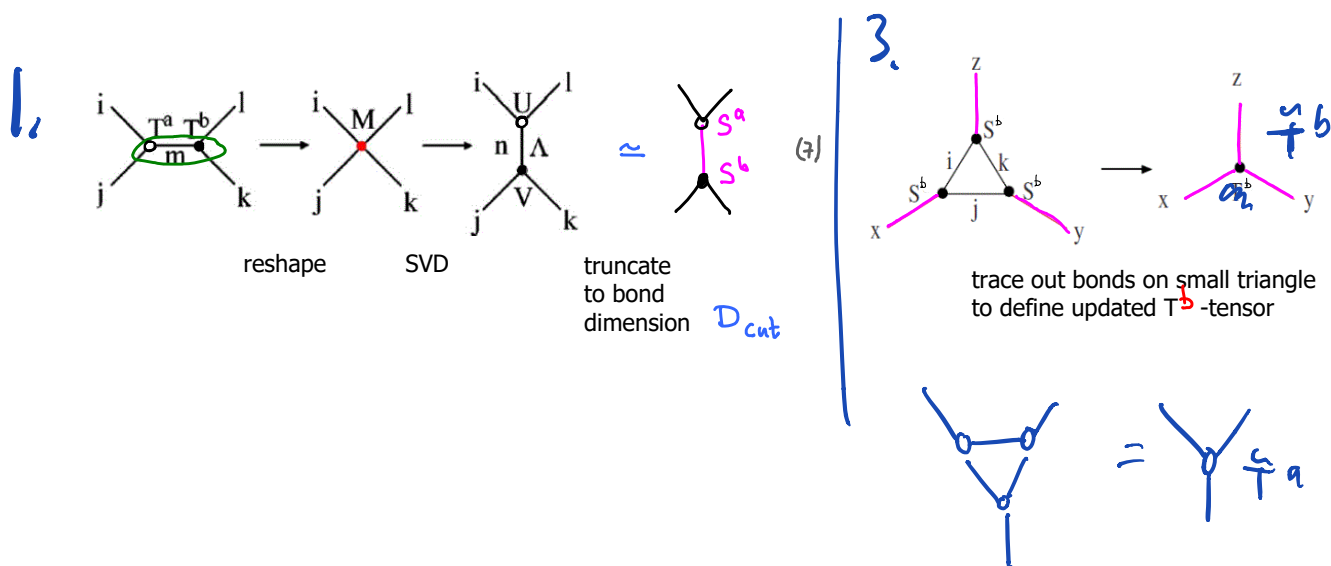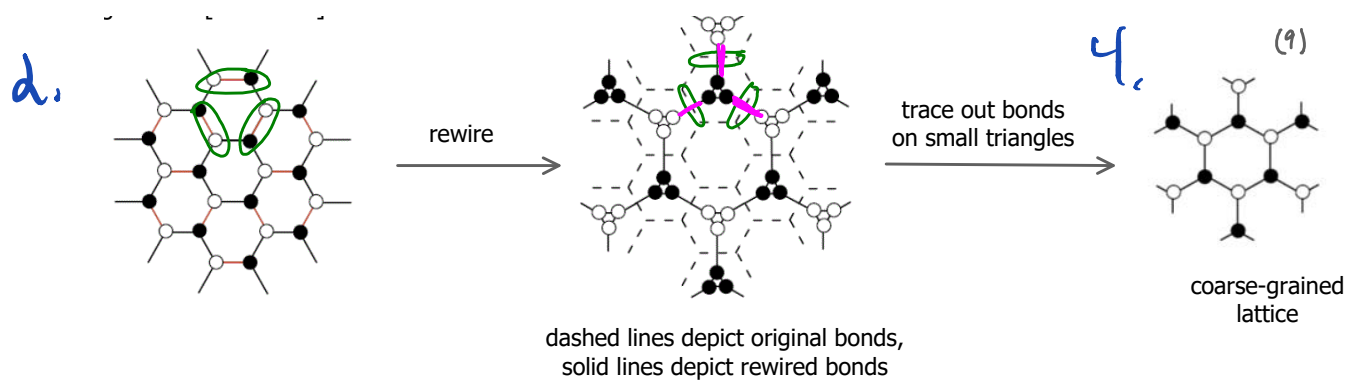
(i,l), (j,k)

1.

reshape   SVD   truncate
                               to bond
                               dimension D_{cut}

(7)

3.

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

Fig 3 from [Zhao2010]

d.

rewire

trace out bonds
on small triangles

4.

(9)

coarse-grained
lattice

dashed lines depict original bonds,
solid lines depict rewired bonds

Iterate this procedure, thereby coarse-graining lattice step by step, until T^a_i, T^b reach

fixed point values, T^{a*}_i, T^{b*}. Use these to compute partition function via

$Z =$

and from there the free energy per spin, and the magnetization, etc.

2. TRG for quantum lattice models
[Jiang2008]

Goal: compute ground state of 2D quantum lattice model

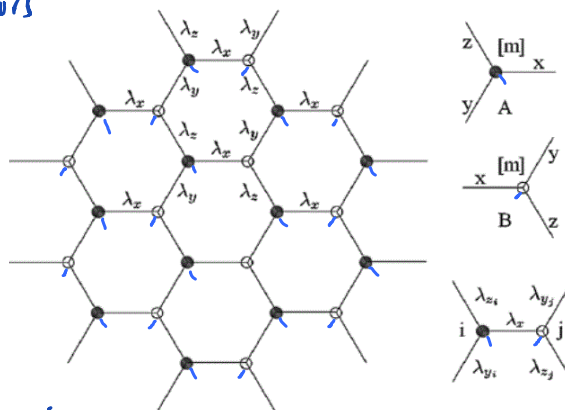Strategy: iterative projection via $e^{-H\tau}$, compress by 'simple update';

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

Model: $S = \frac{1}{2}$ Heisenberg on honeycomb lattice.

Tensor network ansatz:

$$|\psi\rangle = tTr \prod_{\ell \in b} \prod_{\ell' \in W} \overbrace{\lambda_{x_\ell} \lambda_{y_\ell} \lambda_{z_\ell}}^{\text{weight factors}}$$

$$\times A^{\sigma_\ell}_{x_\ell y_\ell z_\ell} B^{\sigma_{\ell'}}_{x_{\ell'} y_{\ell'} z_{\ell'}} |\sigma_\ell , \sigma_{\ell'}\rangle$$

$\ell \in b$ → black $\ell' \in W$ → white

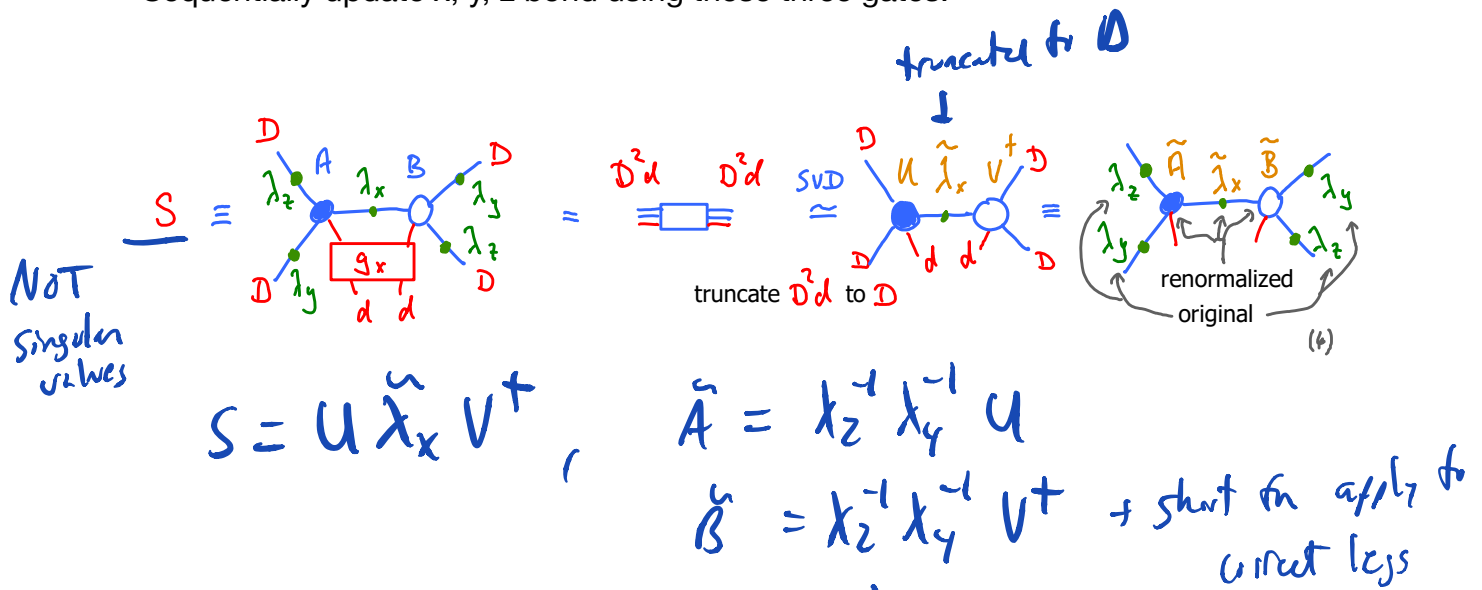$\lambda = d \times d$ matrices that contain info. on environment

Ground state projection via simple update

$$H = H_x + H_y + H_z$$

Suzuki-Trotter:
$$e^{-H\tau} \cong \underbrace{e^{-H_x t}}_{g_x} \underbrace{e^{-H_y t}}_{g_y} \underbrace{e^{-H_z t}}_{g_z} + O(\tau^2)$$

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

$$S = U \tilde{\lambda}_x V^\dagger ,$$

$$\tilde{A} = \lambda_z^{-1} \lambda_y^{-1} U$$

$$\tilde{B} = \lambda_z^{-1} \lambda_y^{-1} V^\dagger + \text{short for apply to correct legs}$$

'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.

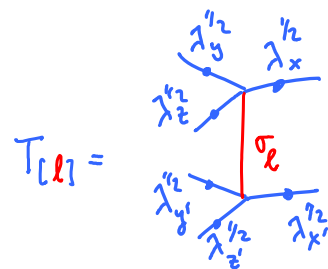-Similarly update y and z bonds. This concludes one iteration.

-Iterate simple update many times.

-Start with $\tau \sim 10^{-3}$, gradually reduce it to $\tau \sim 10^{-5}$
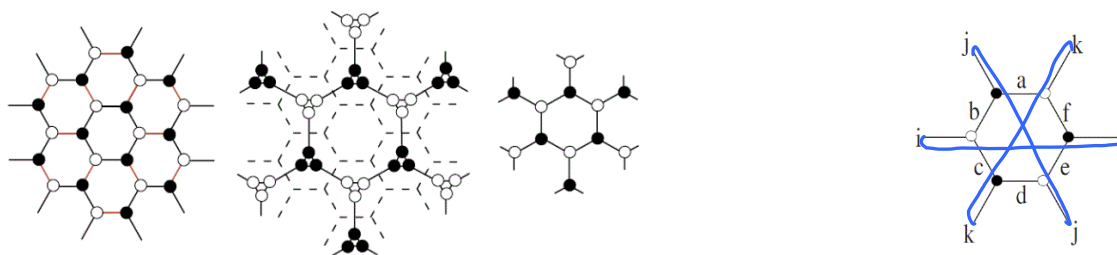
-Number of iterations needed until convergence: $10^5 - 10^6$

$\langle 4|4 \rangle$ is a double-layer tensor network.

Use TRG (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
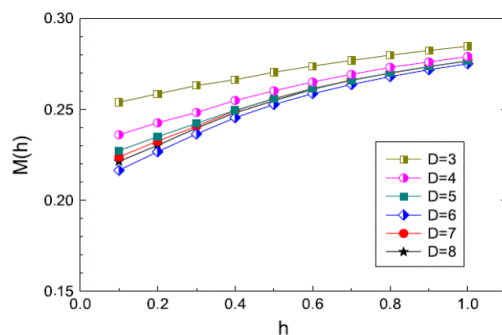
bond indices:

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 $h = 0$.

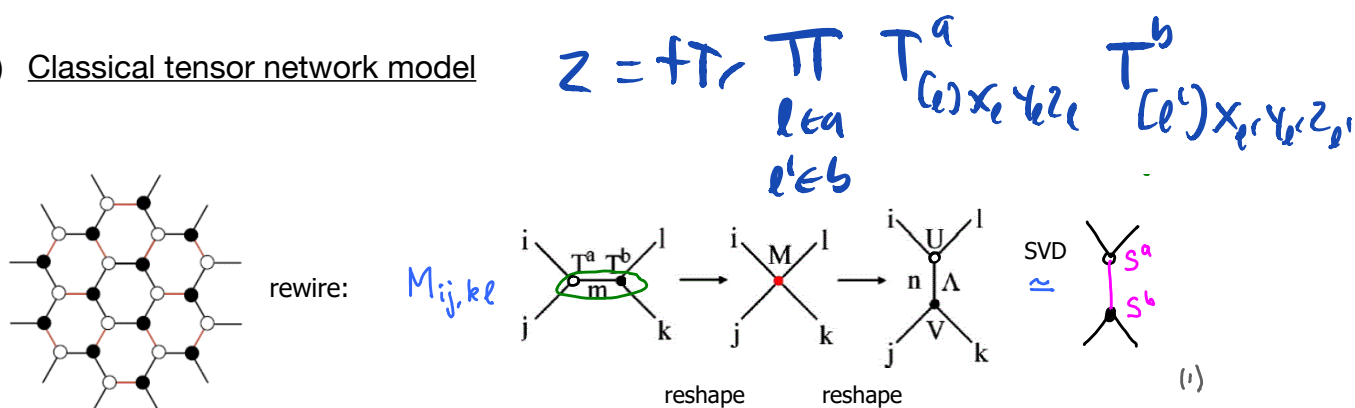| Method	E	M
Spin wave [12]	-0.5489	0.24
Series expansion [13]	-0.5443	0.27
Monte Carlo [14]	-0.5450	0.22
Ours $D = 8$	-0.5506	0.21 ± 0.01

3. Second renormalization (SRG) of tensor networks
[Xie2009, Zhao2010]

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.
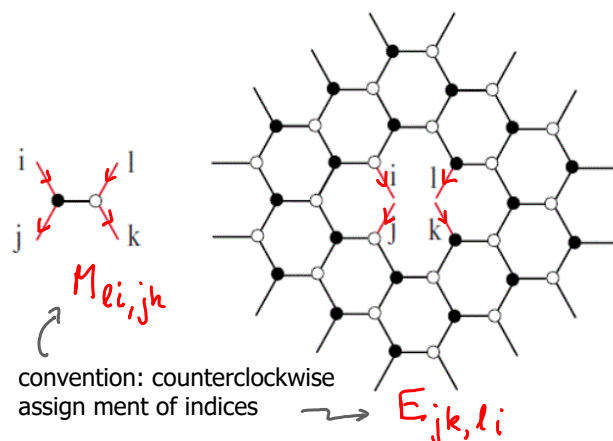
(i) Classical tensor network model

$$Z = f Tr \prod_{\substack{\ell \in a \\ \ell' \in b}} T^a_{(\ell) x_\ell, y_\ell, z_\ell} T^b_{(\ell') x_{\ell'}, y_{\ell'}, z_{\ell'}}$$

rewire: $M_{ij,k\ell}$

reshape   reshape

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

truncation error of partition function Z.

Renormalize environment

Partition function:

$$Z = Tr \, M E$$

$$= \sum_{ijk\ell} M_{\ell i, jk} E_{jk, \ell i}$$

$M_{\ell i, jk}$

convention: counterclockwise
assign ment of indices $\longrightarrow$ $E_{jk, \ell i}$

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 IIIB]

$$Z = M_{\ell i, jk}\, E_{jk, \ell i} =$$

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

$$\text{and}\quad \hat{M} = \Lambda^{1/2} V^\dagger M U \Lambda^{1/2}$$

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

It knows about environment via $U, \Lambda, V^\dagger , (E)$.

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

To this end, first invert relation between M and $\overset{\frown}{M}$, using $\quad u^\dagger u = v^\dagger v = I$

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

$$M = V \Lambda^{-1/4} \overset{\frown}{M} \Lambda^{-1/4} u^\dagger$$

$$\underbrace{\tilde{u} \tilde{\Lambda}^{1/4} \tilde{\Lambda}^{-1/4} \tilde{v}^\dagger}$$
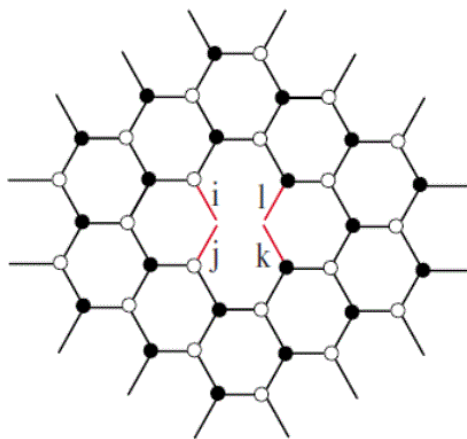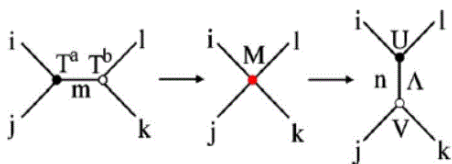$$\underbrace{\qquad}_{S^a} \underbrace{\qquad}_{S^b}$$

and write as product of two vertices: $=$

with indices: $M_{\ell i j k} = S^a_{\ell i \, n} \; S^b_{n j k}$

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

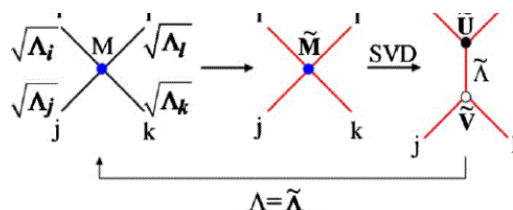<u>Computing environment tensor</u> E using simple update (mean-field approach)

$M = u \Lambda V^\dagger$ 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

$$E \overset{\sim}{=} \sqrt{\Lambda_i \Lambda_j \Lambda_k \Lambda_\ell}$$

- Compute $\overset{\smallfrown}{M}$, then do SVD: $\overset{\smallfrown}{M} = \hat{u} \, \tilde{\Lambda} \, \hat{v}^{+}$

  $\hookleftarrow$ new bond vectn

- Use new $\tilde{\Lambda}$ to recalculate $E_{,} \overset{\smallfrown}{M}_{,} \overset{\smallfrown}{\Lambda}$, etc.

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

needed).

(ii) Computing environment tensor E using finite lattices

The next best approach that improves on the "cheap environment" above is to make a

finite lattice approximation for the environment.

$$\delta f(T) = \left(1 - \frac{f(T)}{f_{exact}(T)}\right); \quad f = \text{free energy}$$

(a) 6 sites   (c) 16 sites

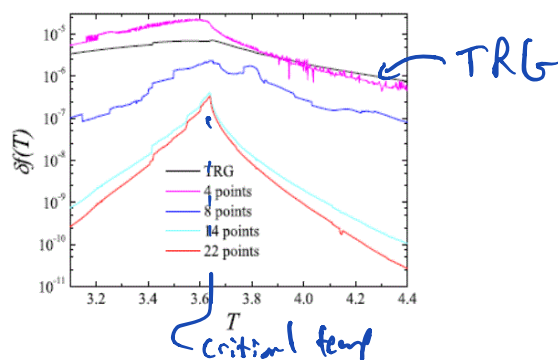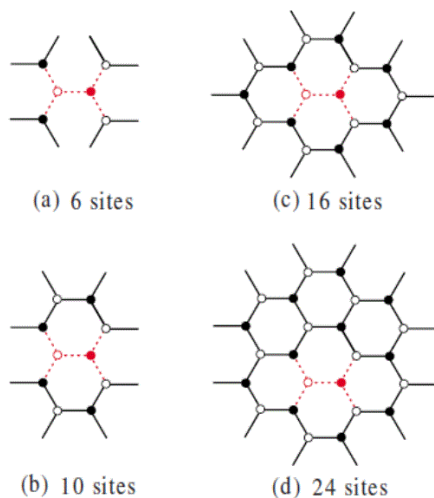(b) 10 sites   (d) 24 sites

$\hookleftarrow$ TRG

critical temp

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!

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

'Forward iteration':

(a) -> (b): Rewire environment $E_{jk, li}$ using data at

iteration n:

$$T^{(n)}_{\ } T^{(n)} = M^{(n)} = U^{(n)} \Lambda^{(n)} V^{(n)\dagger}$$

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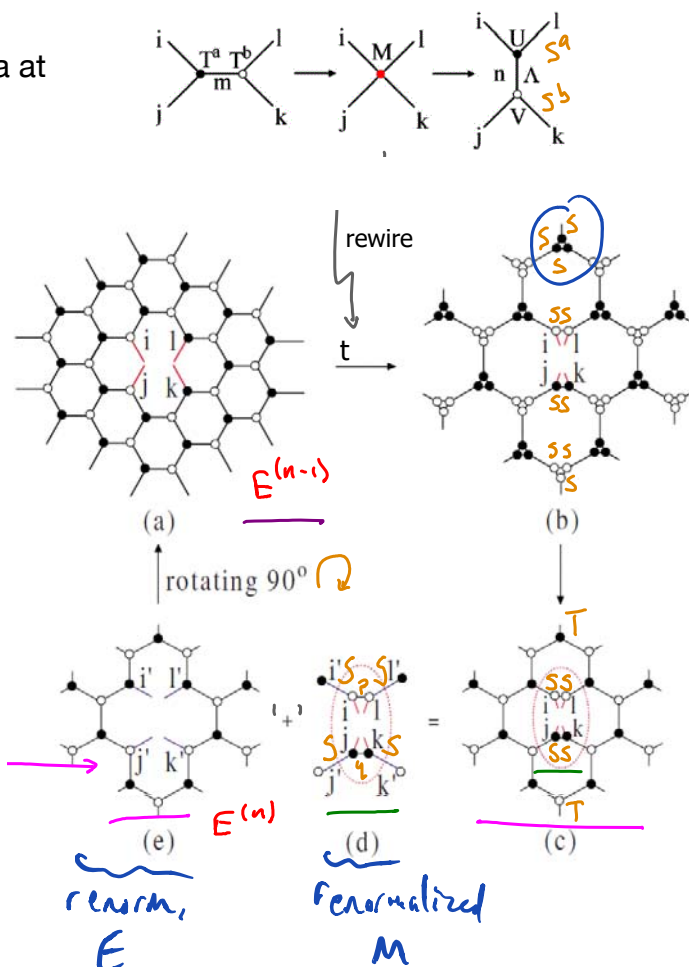
four S are left over

(c) -> (d) + (e): Identify new environment

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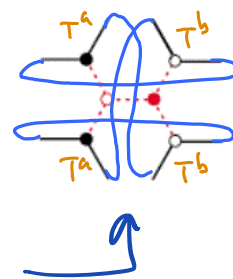
90 degrees, and rescaled.

Iteration relation:

$$E^{(n-1)}_{li,jk} = E^{(n)}_{l'i',j'k'} \; S^a_{l l' o'} \; S^a_{i i' o} \; S^b_{q j' j} \; S^b_{k' q k}$$

-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.

Then 'Backward iteration':   *put back S tensors that were absorbed into M.*

-Start from current values of tensors T_i^a, T_i^b and bond vectors Λ .

-Use them to compute $E^{[N]}, E^{(N-1)}$ etc, all the way back to $E^{(0)}$ = desired result. $= E$

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

have converged.

Results for SRG (2nd renormalization) for classical 2D systems
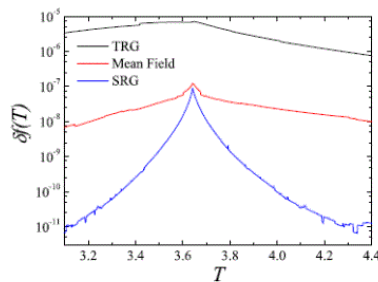
Ising model on triangular lattice:

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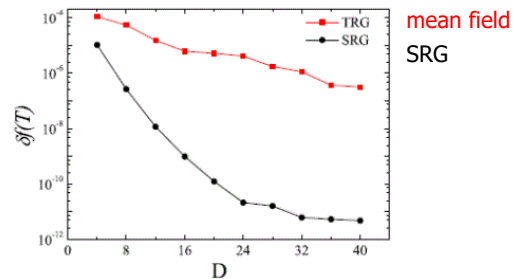
FIG. 13. (Color online) The relative error of the free energy as a function of the truncation dimension D_{cut} for the Ising model on triangular lattices obtained using the TRG (black) and SRG (blue), respectively. T=3.2.

critical state is hardest to simulate   error drops with increasing D faster for SRG

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.

SRG yields more stable results than TRG!
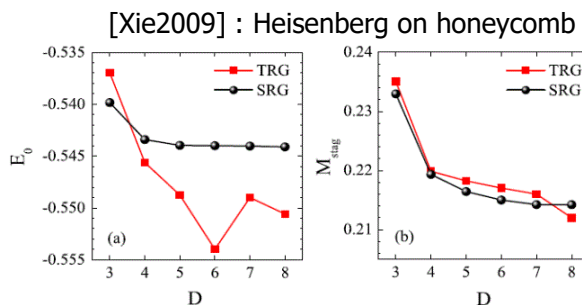
[Xie2009] : Heisenberg on honeycomb

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.
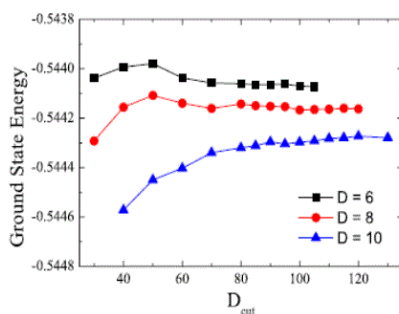
[Zhao2010]

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.
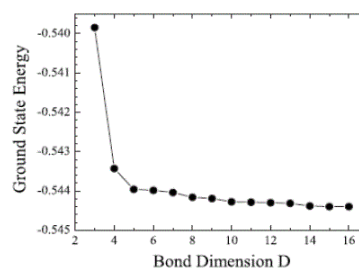
FIG. 20. The ground-state energy of the Heisenberg model on a honeycomb lattice as a function of the bond dimension D obtained by the SRG with $D_{cut}=130$.
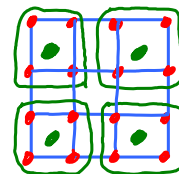
$E^{SRG} = -0.54440$ $E^{QMC} = -0.54455(20)$

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

For reference: very recent results [Lan2019, core tensor renormalization group] reduces cost of TRG from $O(D^6) \rightarrow O(D^4)$

# Graph-independent local truncation (GILT)

Goal: improve TRG by <u>fully</u> removing local correlations, including

those in local loops.

Strategy: devise truncation scheme based on 'environment spectrum'

of local tensors.

1. Motivation

TRG is a concrete, quantitative implementation, for lattice models, of Wilson's RG idea:

$$ T \xrightarrow{\text{Coarse graining}} T' $$

initial tensor   renormalized tensor
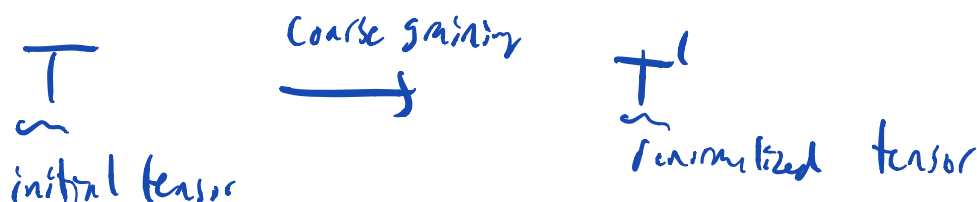
This generalizes Kadanoff's block-spin RG, $H(g) \to H'(g) \stackrel{\sim}{=} H(g')$, which approximates

coarse-grained system by same Hamiltonian, but parametrized by rescaled coupling.

TRG instead allows the form of the Hamiltonian (or corresponding tensors) to change.

BUT: TRG, as formulated by Levin and Nave, does not <u>fully</u> remove all local

correlations.

Reason: it is based on SVD of local tensors, so removes local correlations only for '<u>tree</u> — no loops

tensor networks'. Effect of environment is not included (SRG/full update is an attempt

to do that). As a result, fixed point tensors still include some information from short-

range physics. Hence, TRG does not yield 'proper RG flow' (which should eliminate all

short-range physics).

Needed: schemes that <u>fully</u> remove local correlations at each length scale.

Some key players in this quest:

Levin himself pointed out that TRG fails for 'corner double-line tensors' (CDL). (Public talks, 2007)

[GU2009] Gu & Wen: clearly identify above problem, proposed 'tensor entanglement filtering renormalization' (TERG) to remedy it. This led to discovery of 'symmetry protected topological order', and a classification thereof via structure of fixed point tensors.

[Evenbly2015] Evenbly & Vidal: propose 'tensor network renormalization' (TNR), which goes beyond TRG by including 'disentanglers', allowing removal of all short-ranged correlations at each length scale.

[Evenbly2015a] Evenbly & Vidal: show that TNR generates a MERA (multiscale entanglement renormalization ansatz) structure. (MERA was proposed in [Vidal2007, 2007a, Evenbly 2009]

[Evenbly 2017] Detailed description of TNR, including strategies for optimizing disentanglers (carried over from MERA, as described in [Evenbly 2009]). High cost. $\sim O(D^{2\tau})$

[Yang2017], Yang, Gu, Wen: propose loop optimization for TNR (loop-TNR), which is more effective than TERG. Also more effective and cheaper than TNR.

[Ying2017] Proposes 'tensor network sceletonization' (TNS). Separate steps for coarse-graining and removal of local correlations. Needs costly iterative optimization. Blind to nature of local correlations.

[Bal2017] Bal, Marien, Haegeman, Verstraete: propose TNR+, similar in spirit to [Yang2017], but using element-wise purely positive tensors.

[Evenbly2018] Proposes 'canonical form' and 'optimal truncations' for tensor networks

with closed loops. Optimization scheme include environment information. Simpler than

optimization scheme of MERA/TNR. Performance: better than TNR, loop TNR,

comparable to GILT.

[Hauru2018] Hauru, Delcamp, Mizera: propose 'graph-independent local

truncations' (GILT) of individual bonds of network, based on analysis of environment

spectrum. Very simple, clear scheme. Outperforms all previous 2D approaches. Even

applied to 3D systems!

2. Why is TRG insufficient?
[Hauru2018, Sec 2]

Representing partition function as classical model as a tensor network (graphical

argument)
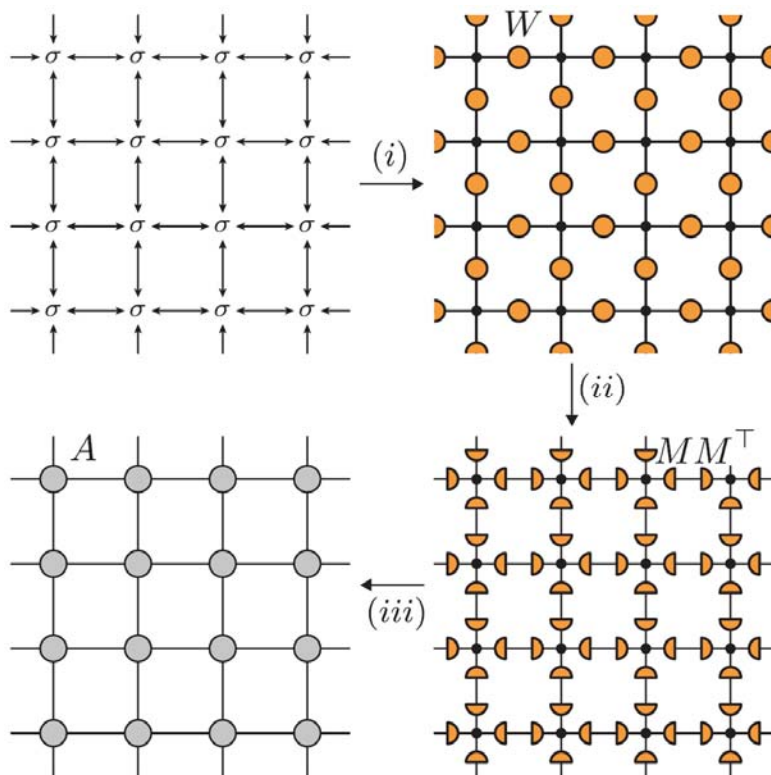
-each site hosts a classical

configuration variables, σ

-(i) each bond is

characterized by

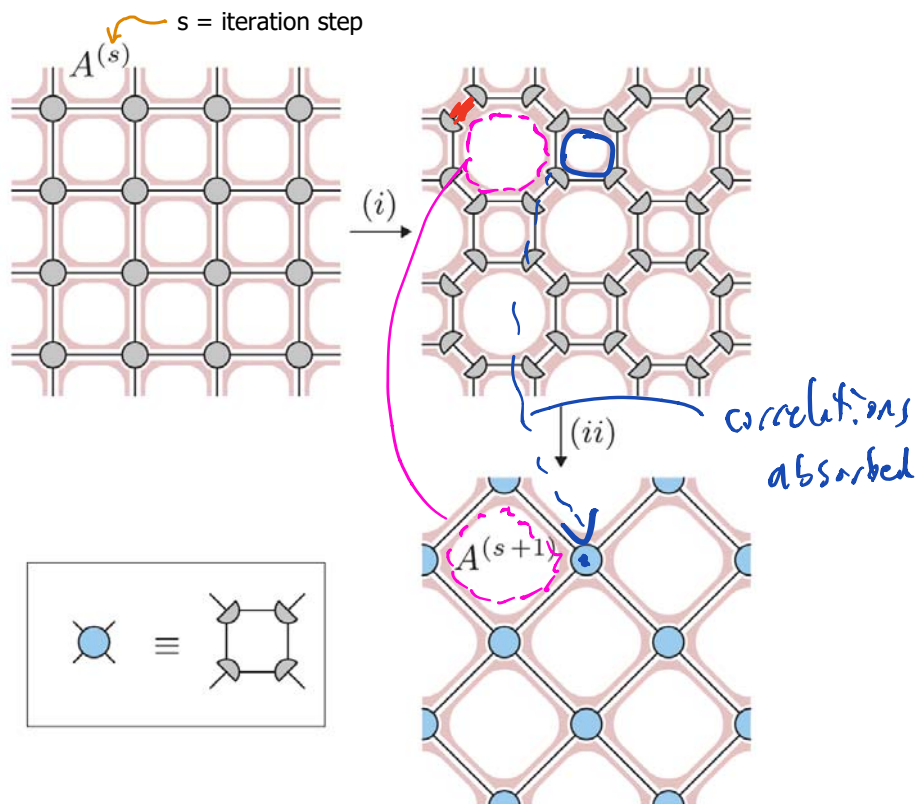Boltzmann weight, $W_a e^{-\beta h_{ij}}$

-(ii) SVD $W = M M^{\top}$ ~classical system

-(iii) at each vertex, contract

bonds to obtain 4-leg

tensor , A.

Single iteration of TRG for square lattice:

(i) Use truncated SVD to split 4-leg tensor $A^{(s)}$ along different diagonals

(ii) Contract sets of 3-leg tensors to obtain new 4-leg tensors.

s = iteration step

$A^{(s)}$

(i)

(ii) correlations absorbed

$A^{(s+1)}$

Shaded red loops represent short-range loop correlations. Step (ii) 'captures' half of red loops; effect is encoded in $A^{(s+1)}$. But other half of the red loops remain, and become ~~come~~ nearest-neighbor correlations of coarse-grained tensors.

This violates principle of RG that coarse-grained description should not include short-ranged (UV) details.

As a result, fixed-point tensors depend on non-universal details, such as exact value of temperature. (Only T < Tc or T > Tc should matter).

Corner double-line (CDL) tensors: an extreme example where TRG fails completely — Gu, Wen

2009
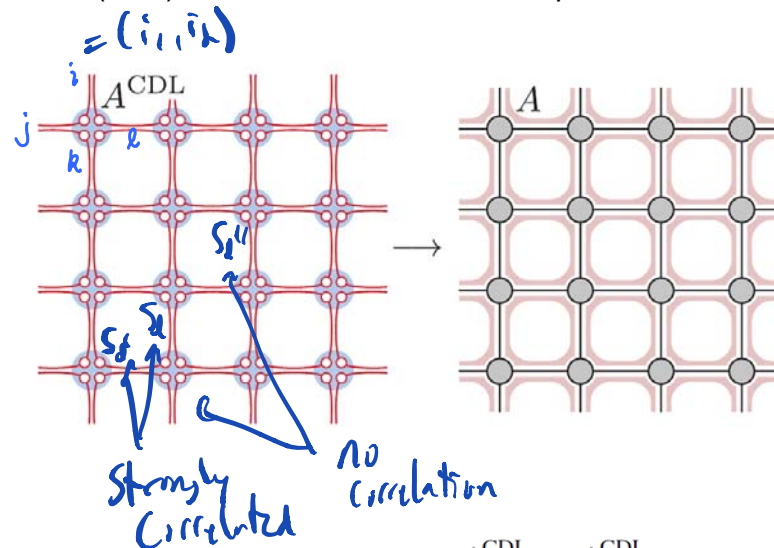
Each index is two-fold composite:

$$A_{ijkl}^{CDL} \equiv A_{(i_1 i_2)(j_1 j_2)(k_1 k_2)(l_1 l_2)}^{CDL}$$
$$= M_{i_1 j_2} M_{j_1 k_2} M_{k_1 l_2} M_{l_1 i_2}$$

Two observables on same plaquette are strongly correlated. Observables on different

plaquettes are totally uncorrelated.

CDL model is a toy model for purely short-ranged physics: it encodes no correlations

at length scales larger than a lattice spacing. Under a proper RG, it should flow to a

trivial fixed point.

However, TRG leaves CDL tensors invariant, i.e. CDL model is fixed point of RG

transformation.

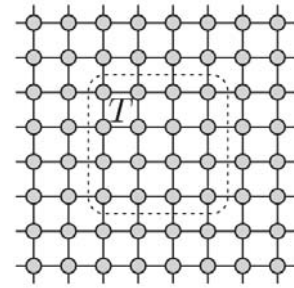This example illustrates: TRG fails to remove loop correlations!

Terminology for this problem: 'accumulation of local, or short-range, correlations'.

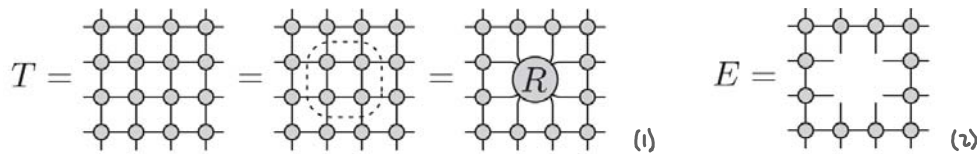This problem gets worse in higher dimensions....

3. Environment spectrum
[Hauru2018, Sec 3]

Consider a local neighborhood, T, of a global network.

Goal: make changes (typically truncations) to a local
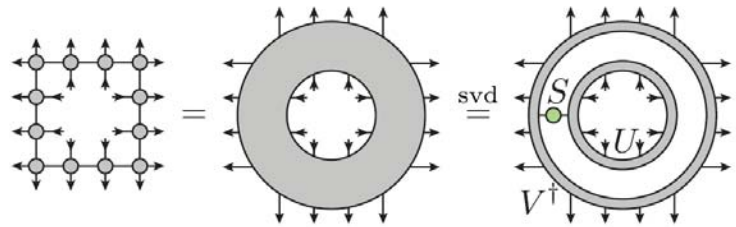
subnetwork, R, of T, without affecting T.

$$T = \quad = \quad = \boxed{R} \quad (1) \qquad E = \quad (2)$$

E = T / R = environment that is left upon removing R from T.
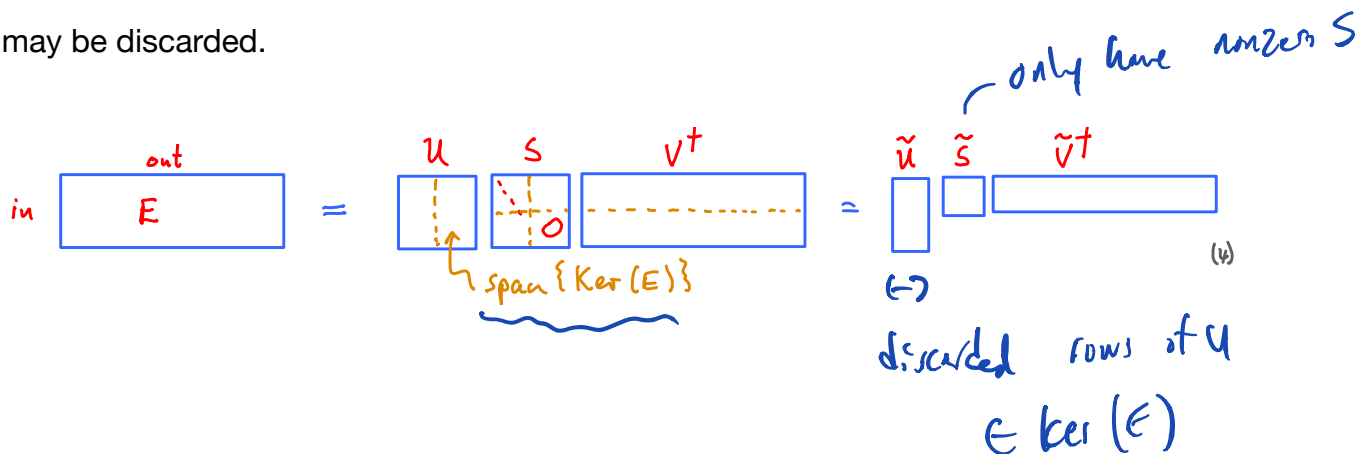
Do SVD, $E = USV^{\dagger}$

between incoming legs (coming

from R)

and outgoing legs (going to

outside):

Singular values = 'environment spectrum' (quantifies how much R affects outside

world)
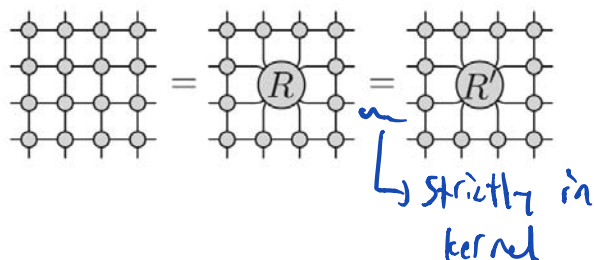
The kernel of E (the incoming subspace that is mapped to 0) is irrelevant to outside and

may be discarded.

only have nonzero S

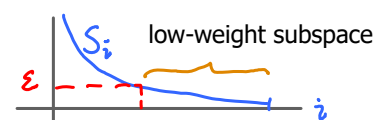$$\underset{in}{\boxed{ E }_{out}} = \boxed{U}\boxed{\begin{matrix}S\\ & 0\end{matrix}}\boxed{V^{\dagger}} = \boxed{\tilde{U}}\boxed{\tilde{S}}\boxed{\tilde{V}^{\dagger}} \quad (4)$$

span {Ker(E)}

(←)

discarded rows of U

$\in$ Ker(E)

If $R - R' \in \ker(E)$

then replacement $R \to R'$

does not affect outside world.

↳ strictly in kernel

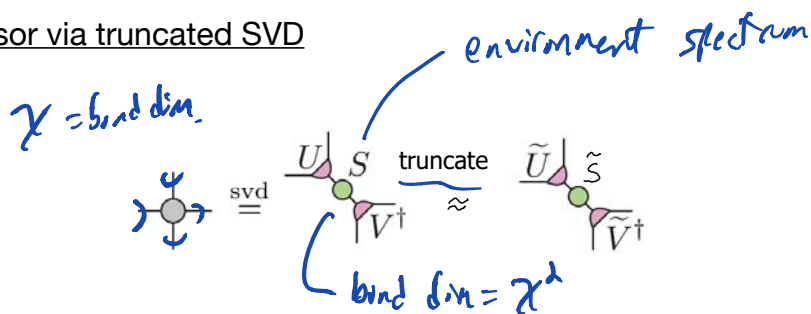If $R \cdot R' \in$ 'low weight subspace of E' (where, say, $S_i < \varepsilon$)

then $R \to R'$ affects outside world only weakly.

Simple example: splitting 4-leg tensor via truncated SVD

Consider first step of TRG:

$\chi = $ bond dim.

environment spectrum

bond dim $= \chi^d$

This can be formulated as truncation of environment spectrum:

Choose R = two legs = product of two

identity matrices:

$$R = -|\quad, \quad T = \text{(diagram)} = E$$

E = T, since cutting identity matrices from outer legs does nothing.

If R is replaced by R' = projector removing low-weight subspace of E,

(with intermediate bond dimension $\chi' < \chi^d$)

$$R' = \chi \overset{\chi'}{\diagdown} \chi \equiv \tilde{U}\tilde{U}^\dagger$$

projectors

then we recover SVD-truncation:
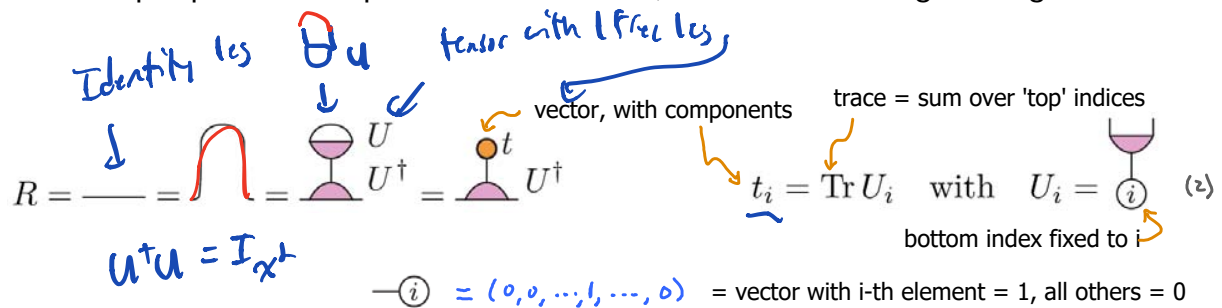

→ bond dim $= \chi'$
  $< \chi^2$

4. GILT: Graph-independent local truncation

Let R = leg in network T. For concreteness, let T = plaquette:
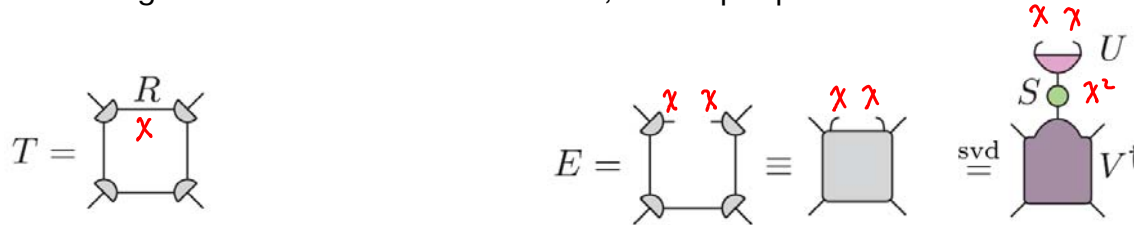
Environment spectrum S quantifies which part of R-space matters only for physics

internal to plaquette. To exploit this information, make basis change on leg R:

Identity leg θu tensor with 1 free leg,

$\downarrow$ $\swarrow$ vector, with components   trace = sum over 'top' indices

$$R = \underline{\quad} = \int \bigcap = U^\dagger = t \, U^\dagger \qquad t_i = \mathrm{Tr}\, U_i \quad \text{with} \quad U_i = \hat{\imath} \qquad (2)$$

$u^\dagger u = I_{\chi^2}$

bottom index fixed to i

$\hat{\imath} = (0, 0, \cdots, 1, \cdots, 0)$ = vector with i-th element = 1, all others = 0

Inserting environment definition into above, we see that environment matrix S, which is

diagonal, multiplies elements

of T,

$= u^\dagger u = \mathbb{1}$

$$T = \sum_i t_i S_i V_i^\dagger$$

$$T = \qquad = \qquad (1) \qquad = \qquad =$$
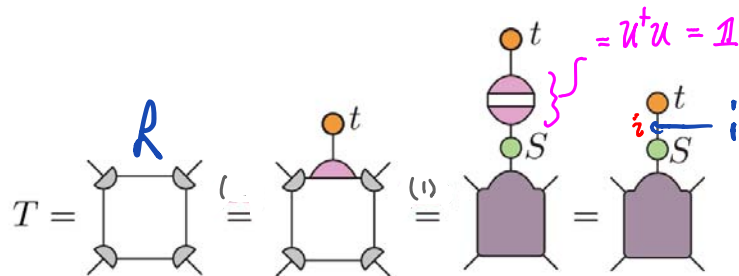
So, if we replace t_i by some other value, t_i', differing

from t_i only for $i \in$ low-weight-subspace of E, $S_i < \varepsilon; \ 0 < i < \chi^2$

then T changes only by $O(\varepsilon)$.

$t'_i = $ truncated
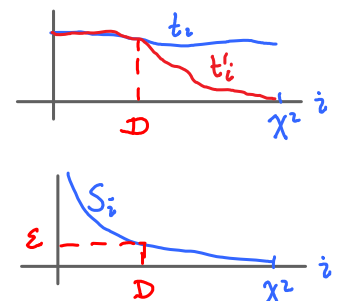
Correspondingly, we can replace leg, $R = I_\chi$

by a matrix $R' = \ominus = $ [diagram] u^i

while T hardly changes:

[diagram] $\approx$ [diagram] $= T$

$\quad \hookrightarrow + O(\varepsilon).$
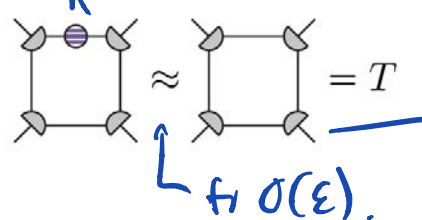
This freedom can be exploited to make matrix R' have as low rank, χ' , as

possible. Low is desirable, since SVD of R', followed by redefinition of neighboring

sites, brings down bond dimension to　　　　.

original
↓

[diagram] χ $\overset{(29)}{\approx}$ [diagram] χ χ $\overset{\text{svd}}{=\!=}$ [diagram] χ' $=$ [diagram] χ'
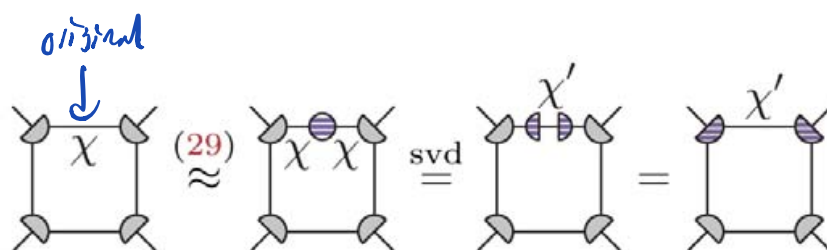
Optimizing the choice of t'

[Hauru2018, App B]

The rank of $R' = $ [diagram] $\overset{t'}{\bullet}$ $U^\dagger$ $= \sum\limits_{i=1}^{\chi^2} t'_i u_i^\dagger$ as a function of t'_i .

is a complicated cost function to minimize. Simpler alternative: minimize

$$C_{norm} = \|R'\|^2 = \text{Tr}\left[R' R'^\dagger\right] = \sum_{i=1}^{\chi^2} s_i'^2$$

singular values of R', not E!

lowercase

Rationale: minimizing the cost function requires reducing the individual s_i' and the

more of them come close to zero, the smaller the rank of R'.

Moreover,

$$C_{norm} = Tr\left(R'R'^{\dagger}\right) = \sum_{ij} Tr\left(t_i u_i^{\dagger} \underbrace{u_j \cdot t_j^*}_{\delta_{ij}}\right)$$

$$= \sum_{i=1}^{\chi^2} |t_i'|^2$$

So, those elements t_i^i that we are free to choose (associated with low-weight

subspace of E) should be chosen as small as possible.

On the other hand, the replacement of R by R' causes an error.

linear in t' - t

$$C_{error} = \left\| \begin{matrix} \end{matrix} - \begin{matrix} \end{matrix} \right\|^2 = \left\| \begin{matrix} t' \\ S \end{matrix} - \begin{matrix} t \\ S \end{matrix} \right\|^2$$

$$= \left\| \begin{matrix} t' \\ S \end{matrix} - \begin{matrix} t \\ S \end{matrix} \right\|^2 = \sum_{i=1}^{\chi^2} |t_i' - t_i|^2 S_i^2 . \qquad (12)$$

of E

This error should remain $O(\varepsilon)$. Thus minimize combined cost function:

$$C_{tot} = C_{error} + C_{norm} = \sum_{i=1}^{\chi^2} \left[|t_i' - t_i|^2 S_i^2 + \varepsilon^2 |t_i'|^2 \right]$$

This is minimized by choosing

$$t_i^i = t_i \cdot \frac{S_i^2}{\varepsilon^2 + S_i^2}$$

(14)
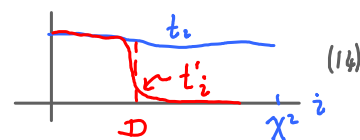
Note: other algorithms require variational

optimization → costly !

Summary of GILT algorithm for truncating leg R in environment E:

(i) SVD the environment E to obtain the unitary U and environment spectrum S (*)

(ii) Compute the traces $t_i = Tr\, U_i$

(iii) Choose the vector t' as above, and compute matrix $R' = \sum_{i=1}^{\chi^2} t'_i u'_i$

(iv) SVD R' as

lowercase $R' = u s v^\dagger$

Repeat (ii)-(iv) on same bond, now with R' as input:

(v) Once singular values S of R' have

converged, multiply $u\sqrt{s}, \sqrt{s}\, v^\dagger$ into neighboring tensors.

(*) We only need U and S, so instead of full SVD of $E = u s v^\dagger$ (with $\to s, t\, v^\dagger$), it suffices to

compute eigenvalue decomposition of the Hermitian matrix
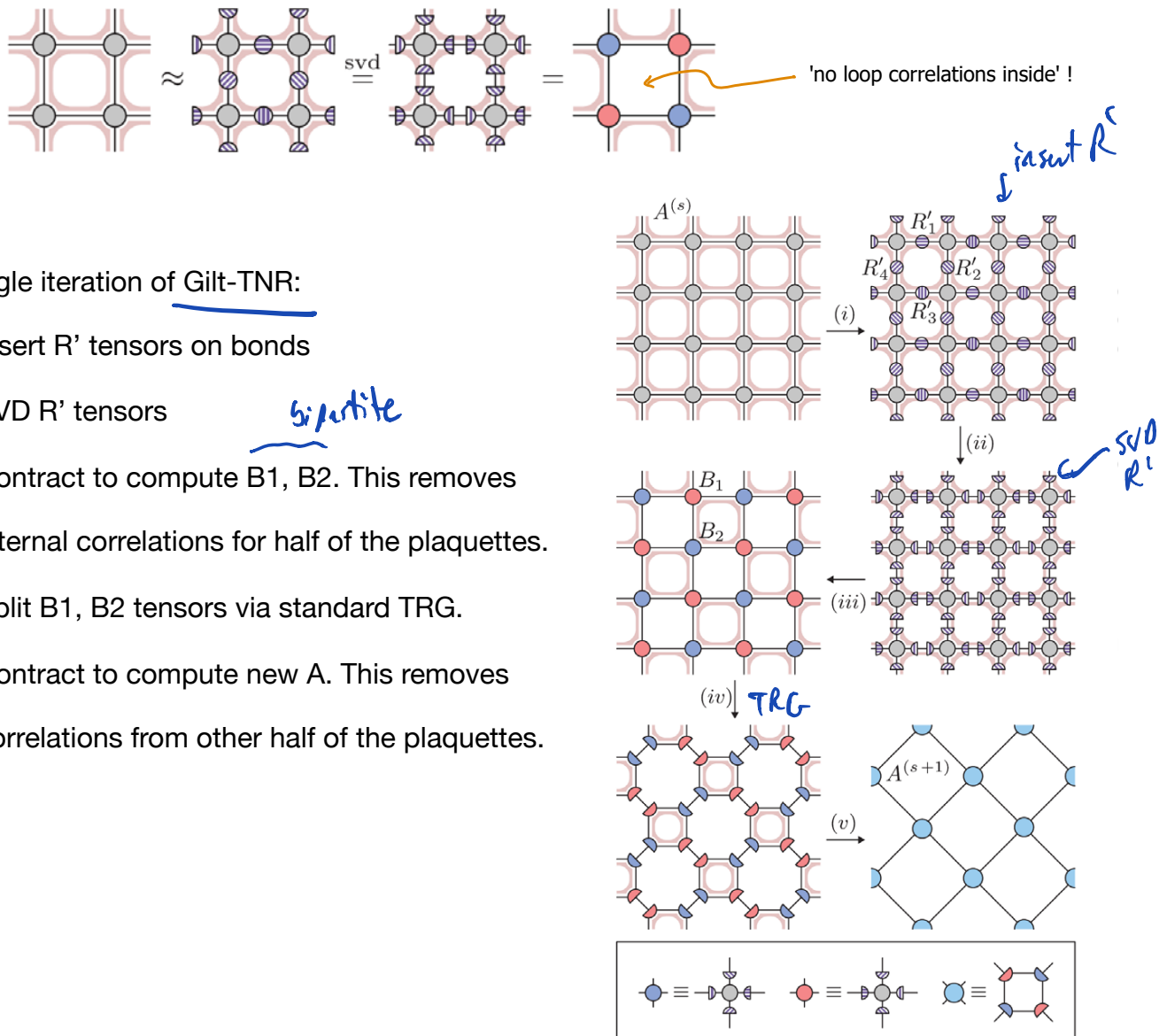
$$E E^\dagger = u s^2 u^\dagger = u s v^\dagger v s u^\dagger$$

This is computationally much cheaper, and for square plaquette reduces cost to $O(\chi^6)$.

5. GILT Tensor network renormalization (Gilt-TNR)

Problem with TNR was: local correlations are dealt with properly only around every other plaquette. Remedy: before each TRG step, apply Gilt to all four bonds of problematic plaquettes.

The four matrices R' must be created in serial, not parallel, since each one modifies environment of the others. Together, they truncate away any details internal to plaquette, by modifying tensors at corners.

'no loop correlations inside' !

insert R'

A single iteration of Gilt-TNR:

(i)   Insert R' tensors on bonds

(ii) SVD R' tensors   *bipartite*

(iii) Contract to compute B1, B2. This removes

      internal correlations for half of the plaquettes.

(iv) Split B1, B2 tensors via standard TRG.

(v) Contract to compute new A. This removes

      correlations from other half of the plaquettes.

SVD R'

Main advantage of Gilt-TNR over rivals:

-'simplicity and generality' (minimal working implementation takes ~100 lines of code)

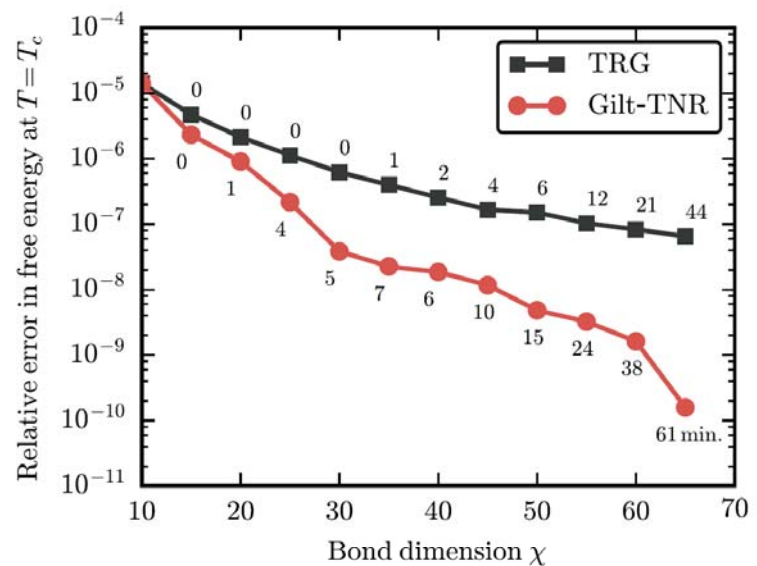-No iterative optimization of truncation.

-Graph does not change.

Applying Gilt to non-square lattices requires simply changing the neighborhood T used

for Gilt step.

-So efficient it has already been applied to 3D Ising model.

6. Benchmark results
[2D classical Ising model]

-TRG and Gilt-TNR obtained

with same code; for TRG, Gilt

was turned off.

-Gilt-TNR orders of magnitude

more accurate than TRG

-at only moderate increase in

run-time.

RG flow of TNR and Gilt-TNR

two dominant singular values

one dominant singular value

$T = 0.999\,T_c$ $T = 0.99999\,T_c$ $T = T_c$ $T = 1.00001\,T_c$ $T = 1.001\,T_c$

first s. v.
second s. v.

TRG
Singular value magnitude

Gilt-TNR
Singular value magnitude

Linear system size   Linear system size   Linear system size   Linear system size   Linear system size

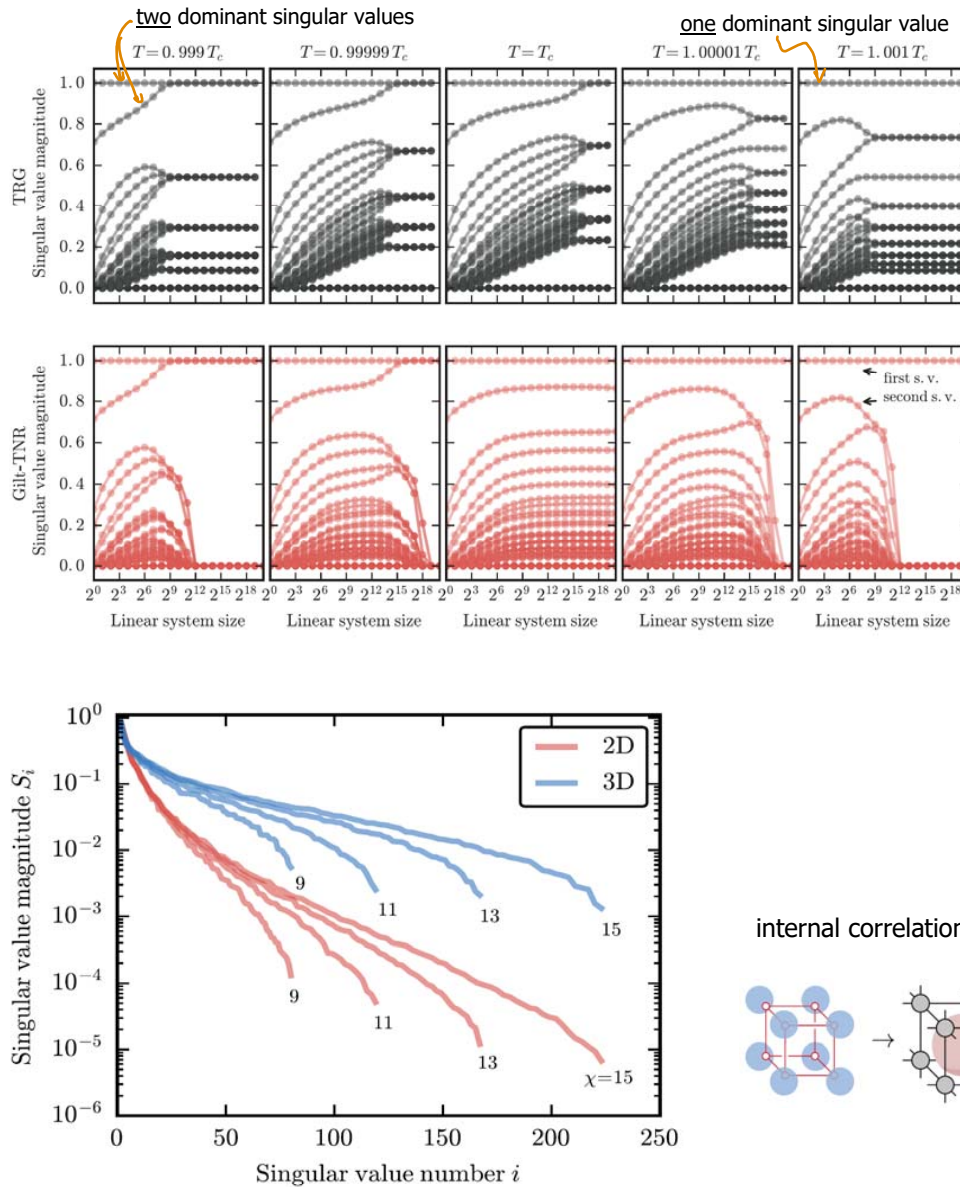internal correlation:

$\rightarrow$

FIG. 7. Typical environment spectra of a single leg with respect to a square plaquette in 2D (bottom four spectra) and a cube in 3D (top four spectra), each labeled with the corresponding bond dimension χ. Recall that in each spectrum, the large values correspond to parts of the vector space of the leg that are relevant for physics outside the plaquette or the cube, whereas small values signify contributions relevant only for short-range details. The spectra in 3D can be seen to decay much more slowly, indicating that larger bond dimensions are necessary, before truncations with a small error are possible. The behavior of the spectra as χ is increased, is also somewhat different in 2D and 3D. In 2D the longer spectra have more values mainly at the bottom end, whereas in 3D new values appear almost throughout the whole spectrum. The example spectra shown here are for the Ising model, from systems that have been coarse grained thrice. Many other choices of system sizes would yield qualitatively similar results, and the same overall difference between 2D and 3D can also be seen with

$T = $ R and $E = $