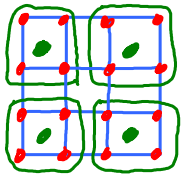
TRG-II: Graph-Independent Local Truncations (Gilt)

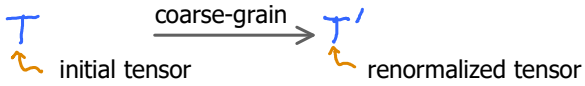


Goal: improve TRG by fully removing local correlations, including those in local loops.

Strategy: devise truncation scheme based on 'environment spectrum' of local tensors.

1. Motivation

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



This generalizes Kadanoff's block-spin RG, $H(g) \rightarrow H'(g) \simeq H(g')$, 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 & Nave, does not fully remove all local correlations.
Reason: it is based on SVD of local tensors, so removes local correlations only for 'tree 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 fully 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. Very influential paper!

[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 approach) structure. (MERA was proposed in [Vidal2007], [Vidal2007a], and first described in detail by Evenbly & Vidal in [Evenbly2009].)

[Evenbly2017] Detailed description of TNR, including strategies for optimizing disentanglers (carried over from MERA, as described in [Evenbly2009]). The schemes are very costly!] next time

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

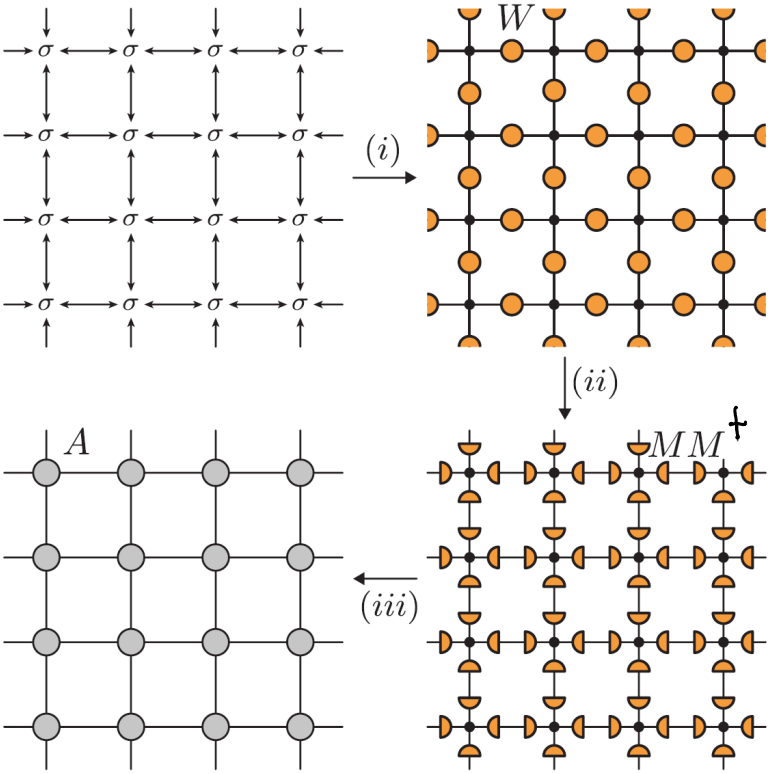
[Bal2017] Bal, Mariën, Haegeman, Verstraete: propose TNR+, similar in spirit to [Yang2017], but using element-wise purely positive tensors.

[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. Was even applied for 3D (!!) system.] today

[Evenbly2018] Proposes 'canonical form' and 'optimal truncations' for tensor networks with closed loops. Optimization scheme includes environment information. Simpler than optimization scheme of MERA/TNR. Performance: better than TNR and loop-TNR, comparable to GILT.

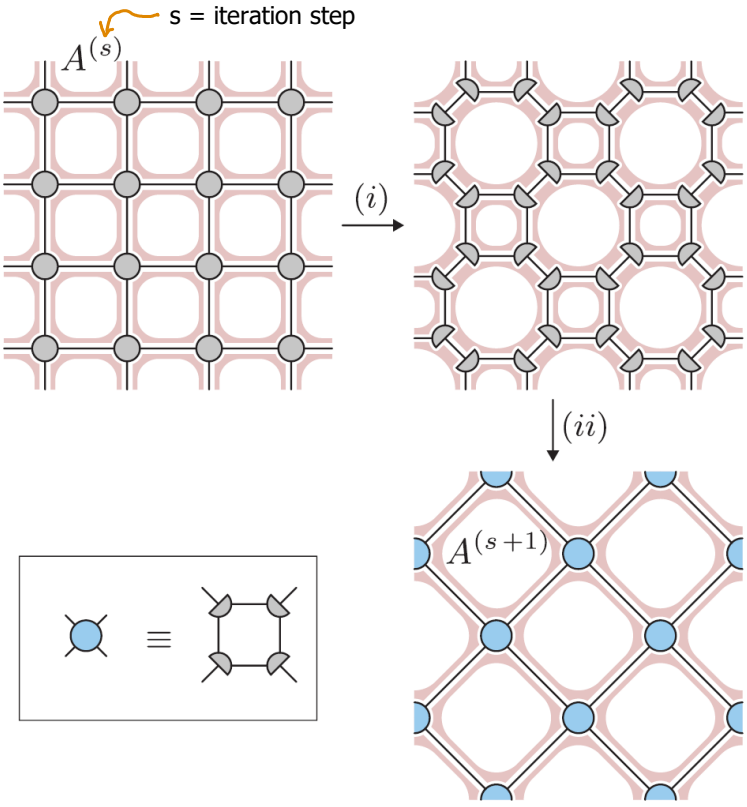
2. Why is TRG insufficient? [Hauru2018, Sec. 2]

Representing partition function of classical model as a tensor network (graphical argument):



- each site hosts a classical configuration variable
- (i) each bond is characterized by Boltzmann weight, W
- (ii) SVD: $W = M M^T$
- (iii) at each vertex, contract bonds to obtain 4-leg tensor, A

Single iteration of TRG algorithm for square lattice:



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

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

Shaded red loops represent 'short-range loop correlations'.

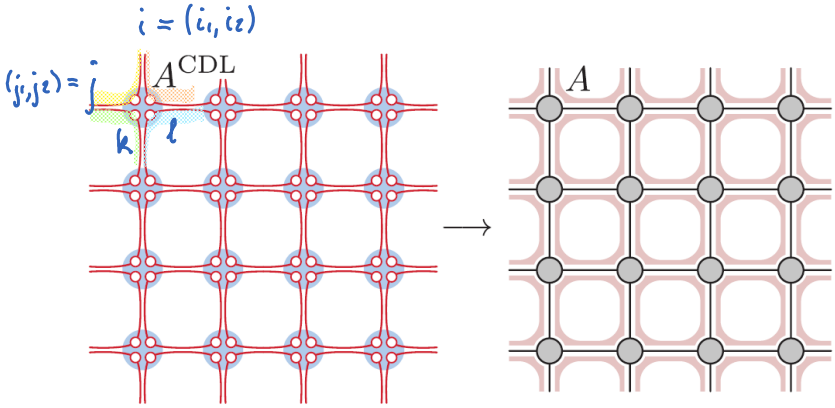
Step (ii) 'captures' half of the red loops: their effect is encoded in $A^{(s+1)}$.

But other half of the red loops remain, and become 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. (For a proper RG scheme, only T < Tc or T > Tc should matter.

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



Each index is two-fold composite:

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

local tensor has product form!

Two observables on same plaquette are strongly correlated.

Observables on different plaquettes are totally uncorrelated.

CDL model is 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!

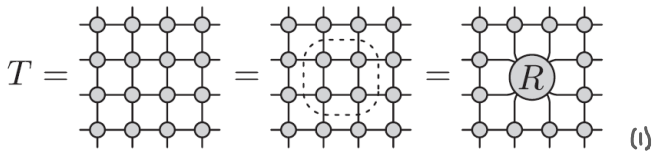
Terminology for this problem: 'accumulation of local, or short-range, correlations'.

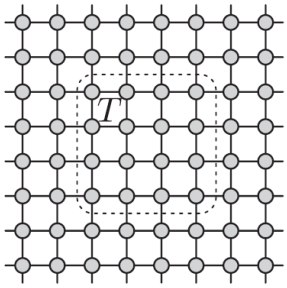
This problem gets worse in higher dimensions…

3. Environment spectrum [Hauru2018, Sec. 3]

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

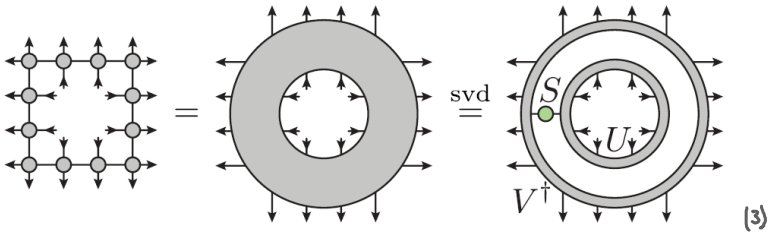
Goal: make changes (typically truncations) to a local subnetwork, R, of T, without affecting T.





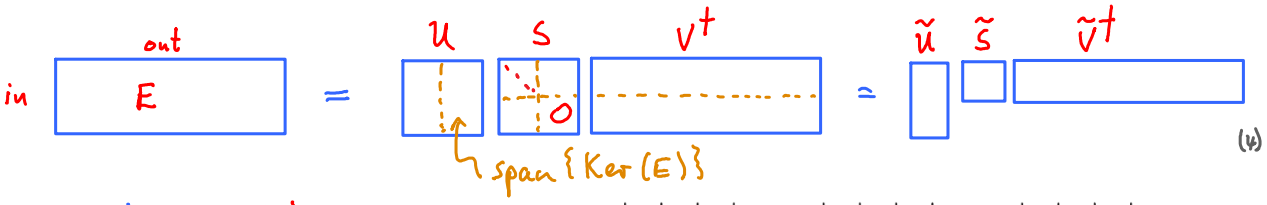
E= T / R = environment that is left upon removing R from T

Do SVD, $E = USV^\dagger$, between incoming legs (coming from R) and outgoing legs (going to outside):

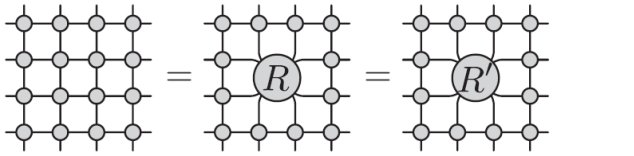


Singular values = 'environmental spectrum' (quantifies how much R affects outside world)

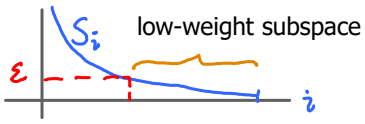
The kernel of E (the incoming subspace that is mapped to 0) is irrelevant to outside and may be discarded.



If $R - R' \in \mathrm{Ker}(E)$ then replacement $R \to R'$ does not affect outside world.

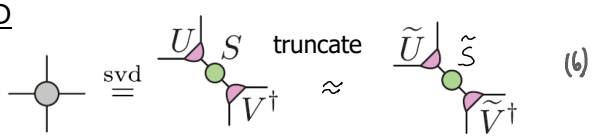


If $R - R' \in$ 'low-weight subspace of E' (where, say, $S_i < \varepsilon$) then $R \to R'$ affects outside world only weakly.



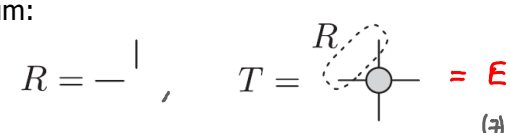
Simple example: splitting 4-leg tensor via truncated SVD

Consider first step of TRG:



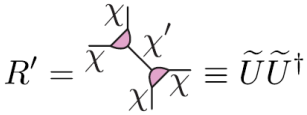
This can be formulated as truncation of environment spectrum:

Choose R = two legs = product of two identity matrices:



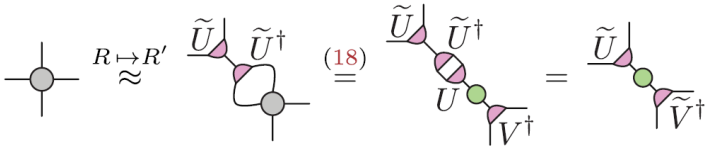
E = T, since cutting identity matrices from outer legs does nothing.

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



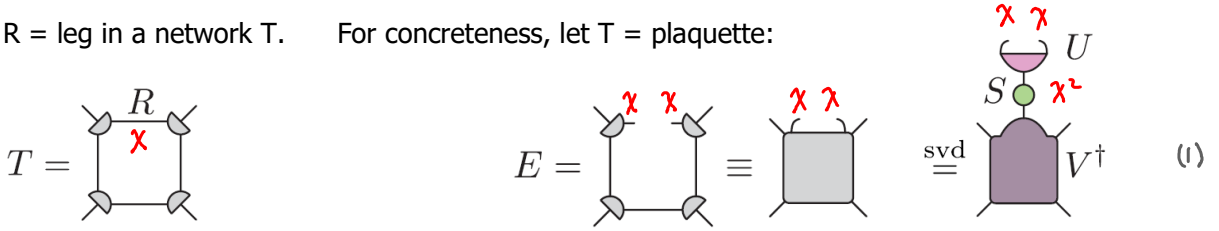
(with intermediate bond dimension $\chi' < \chi^2$)

then we recover SVD-truncation (6):



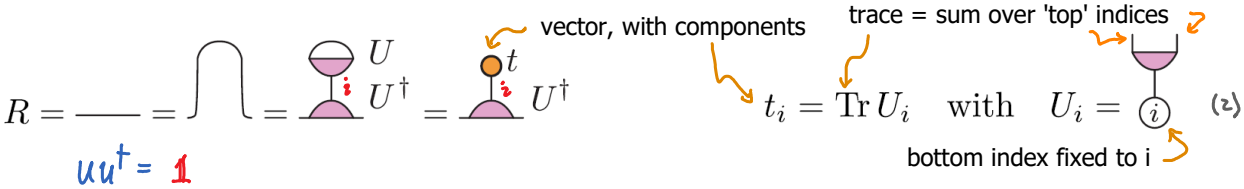
4. Gilt: Graph-independent local truncation

Let R = leg in a network T. For concreteness, let T = plaquette:



(1)

Goal: reduce bond dimension of R, without affecting system beyond T. Environment spectrum S controls information flow from R to outside. So, express leg R in SVD 'eigenbasis' of S:



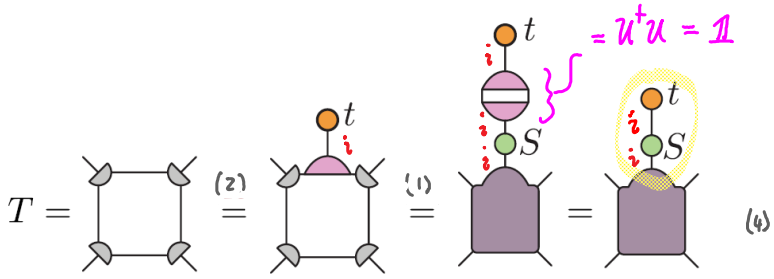
(2)

$uu^\dagger = \mathbb{1}$ on bond space

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

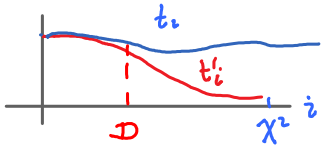
Inserting (2) into (1), we see that environment matrix, S, which is diagonal, multiplies elements of t_i,

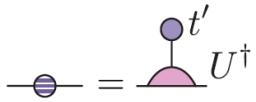
$$T = \sum_i t_i S_i V_i^\dagger \qquad (3)$$

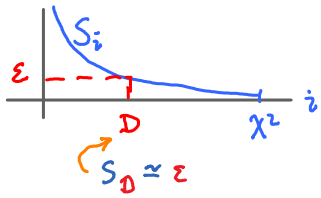


(4)

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, $S_i < \varepsilon$, $D < i < \chi^2$, then T changes only by $O(\varepsilon)$.

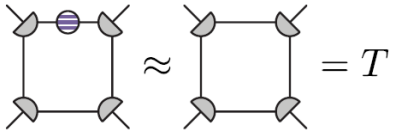


Correspondingly we can replace leg, $R = \mathbb{1}$, by a matrix $R' =$  (6)

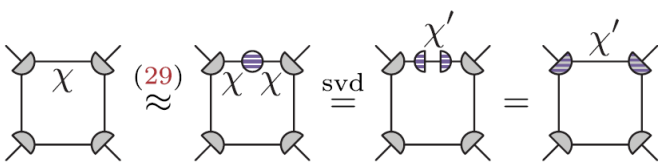


(7)

while T hardly changes:

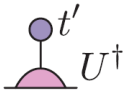


This freedom can be exploited to make matrix R' have as low rank, χ', as possible. Low is desirable, since SVD of R', followed by redefinition of neighboring sites, brings down bond dimension to χ':



(8)

Optimizing the choice of t' [Haure2018, App. B]

The rank of $R' =$  $= \sum_{i=1}^{\chi^2} t_i' U_i^\dagger$ as function of the coefficients t_i' (9)

is a complicated cost function to minimize. Simpler alternative: minimize singular values of R' (not E !)

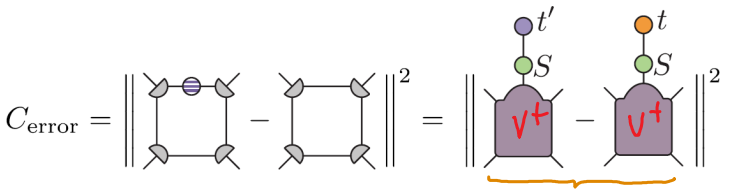
$$C_{norm} = \|R'\|^2 = \mathrm{Tr}\left[R' R'^\dagger\right] = \sum_{i=1}^{\chi^2} s_i'^2 \qquad \left[\begin{array}{c} R' = u s' v^\dagger \\ R' R'^\dagger = u s'^2 v^\dagger \\ \mathrm{tr}(R' R'^\dagger) = \mathrm{tr}\, s'^2 \end{array}\right] \quad (10)$$

Rationale: minimizing this cost function requires reducing the individual s_i
and the more of them come close to zero, the smaller the rank of R'.

Moreover, $C_{norm} = \mathrm{Tr}\, R' R'^\dagger = \mathrm{Tr}\left(t_i' \underbrace{u_i^\dagger u_i}_{=1} t_i'^*\right) = \sum_{i=1}^{\chi^2} |t_i'|^2$ (11)

So, those elements t_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:



This error should remain $O(\varepsilon)$.

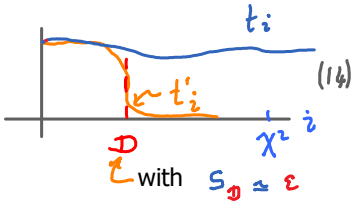
$$\overset{V^\dagger V = \mathbb{1}}{=} \left\| \begin{array}{c} t' \\ S \end{array} - \begin{array}{c} t \\ S \end{array} \right\|^2 = \sum_{i=1}^{\chi^2} |t_i' - t_i|^2 S_i^2 \,. \quad (12)$$

Thus, minimize combined cost function:

$$C_{total} = C_{error} + \varepsilon^2 C_{norm} = \sum_{i=1}^{\chi^2} \left[|t_i' - t_i|^2 S_i^2 + \varepsilon^2 |t_i'|^2 \right] \quad (13)$$

This is minimized by choosing $t_i' = t_i \dfrac{S_i^2}{\varepsilon^2 + S_i^2}$ (14)

[in effect, discard singular values $s_i < \varepsilon$]

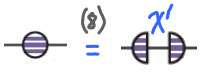


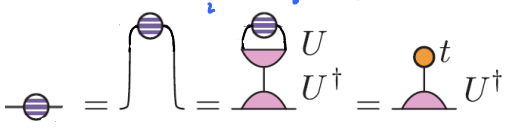
Summary of GILT algorithm for truncating a leg R in an environment E

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

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

(iii) Choose the vector t' according to (14) and compute the matrix $R' = \sum_i^{\chi^2} t_i' U_i^\dagger$

(iv) SVD R' as $R' = u s v^\dagger$, 

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, (8).

(*) We only need U and S, so instead of full SVD of $E = u s v^\dagger$, it suffices to compute
eigenvalue decomposition of the Hermitian matrix $EE^\dagger$ as $EE^\dagger = u s^2 u^\dagger$
This is computationally much cheaper, and for square plaquette reduces cost to $O(\chi^6)$.

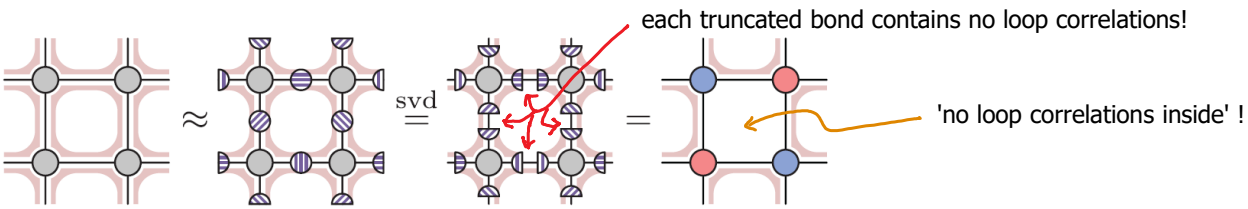
5. Gilt Tensor network renormalization (Gilt-TNR)

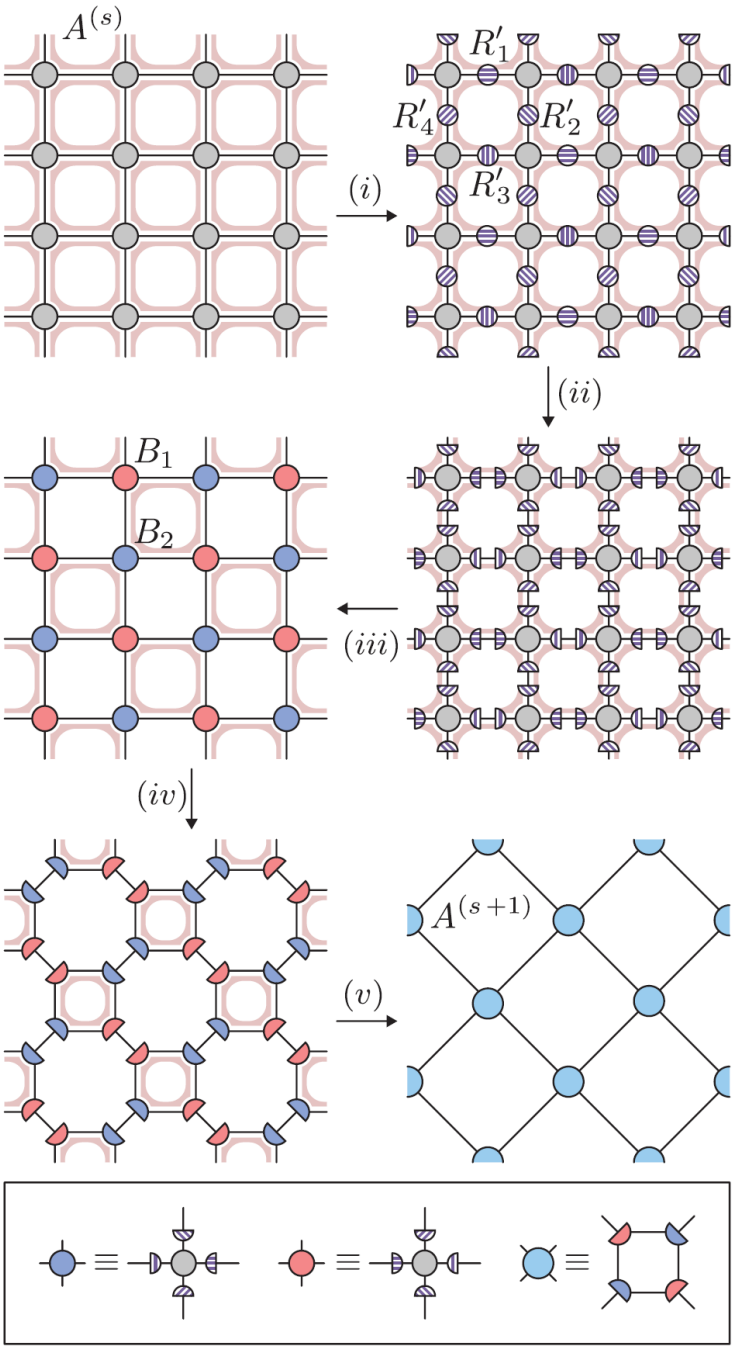
TRG-II.5

Problem with TRG 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 others. Together, they truncate away any details internal to plaquette, by modifying tensors at corners:





A single iteration of Gilt-TNR:

(i) Insert R' tensors on bonds
(ii) SVD R' tensors
(iii) Contract to compute B1, B2.
This removes internal correlations for half the plaquettes.
(iv) Split B1, B2 tensors via standard TRG.
(v) Contract to compute new A.
This removes correlations for other half of the plaquettes.

Main advantage of Gilt-TNR over rivals:

- ist 'simplicity and generality'. ('minimal working implementation takes mere hundred lines).
- 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 simple, that it has already been applied to 3D (Ising model).

6. Benchmark results [2D classical Ising model]

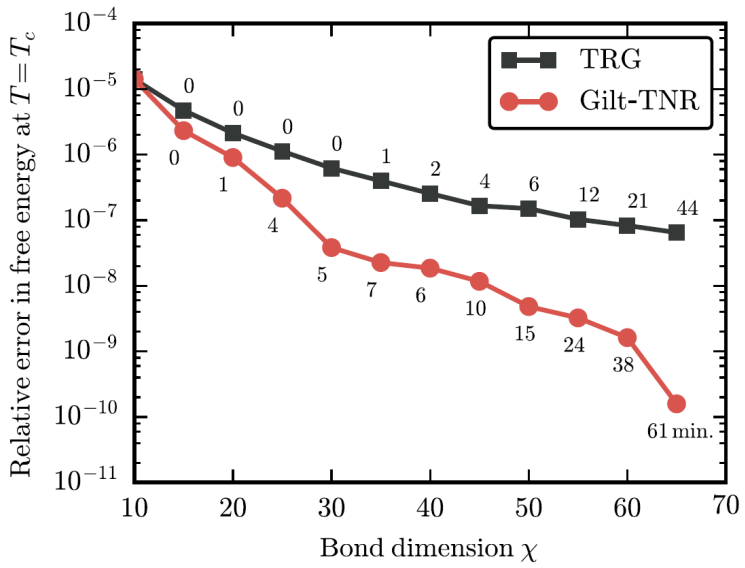


FIG. 5. Error in free energy of the critical 2D classical Ising model at different bond dimensions, for TRG and for Gilt-TNR. The numbers next to the data points are total running times in minutes, for the simulation consisting of 25 iterations of the algorithm. The exact running times of course depend on hardware and implementation details, but worth noting is the relatively small difference between the TRG and Gilt-TNR algorithms. Even though adding Gilt into the algorithm slows it down a bit, this is more than compensated for in the quality of the results. For the Gilt-TNR results shown here, the parameter ϵ has been chosen to be 8×10^{-7}. Note that this is not the optimal choice of ϵ for this whole range of χ. Instead, one should vary ϵ as one varies χ, making it smaller as χ grows. It is only for simplicity of presentation that we have chosen to stick to a single value of ϵ that performs well over the whole range of χ's shown.

- TRG and Gilt-TNR obtained with same code, for TRG the Gilt-algorithm was simply turned off
- Gilt-TNR orders of magnitude more accurate than TRG,
- at only moderate increase in run-time

TABLE I. First few scaling dimensions of the Ising CFT, as obtained by diagonalizing a transfer matrix on a cylinder/torus [28]. In all these cases the cylinder consists of two coarse-grained sites, but the amount of coarse-graining varies. In the Gilt-TNR results a linear system size of 2^8 sites has been used, and ϵ was chosen to be 4×10^{-9}. We are able to reach similar quality as with TNR and Loop-TNR, with moderate computational effort (the simulation in question finished in a little less than 12 h on the machines we use, cf. Footnote 9).

Exact	TRG $\chi = 120$	TNR $\chi = 24$	Loop-TNR $\chi = 24$	Gilt-TNR $\chi = 120$
0.125	0.124993	0.1250004	0.12500011	0.12500015
1	1.0002	1.00009	1.000006	1.00002
1.125	1.1255	1.12492	1.124994	1.12504
1.125	1.1255	1.12510	1.125005	1.12506
2	2.002	1.9992	1.9997	2.0002
2	2.002	1.99986	2.0002	2.0002
2	2.003	2.00006	2.0003	2.0003
2	2.002	2.0017	2.0013	2.0004

Comparison against other algorithms:

TRG: [Levin2007]
TNR: [Evenbly2015], [Evenbly2017]
Loop-TNR: [Yang2017]

TNR, Loop-TNR, Gilt-TNR all have computational cost $\sim O(\chi^6)$
For same χ, TNR and Loop-TNR are more accurate, but they have very much larger run-times, due to iterative optimization procedures.

For $\chi^{gilt} > \chi^{TNR}$ but run times are comparable, accuracy is comparable.

two dominant singular values
$T = 0.999\,T_c$ $T = 0.99999\,T_c$ $T = T_c$ $T = 1.00001\,T_c$
one dominant singular value
$T = 1.001\,T_c$

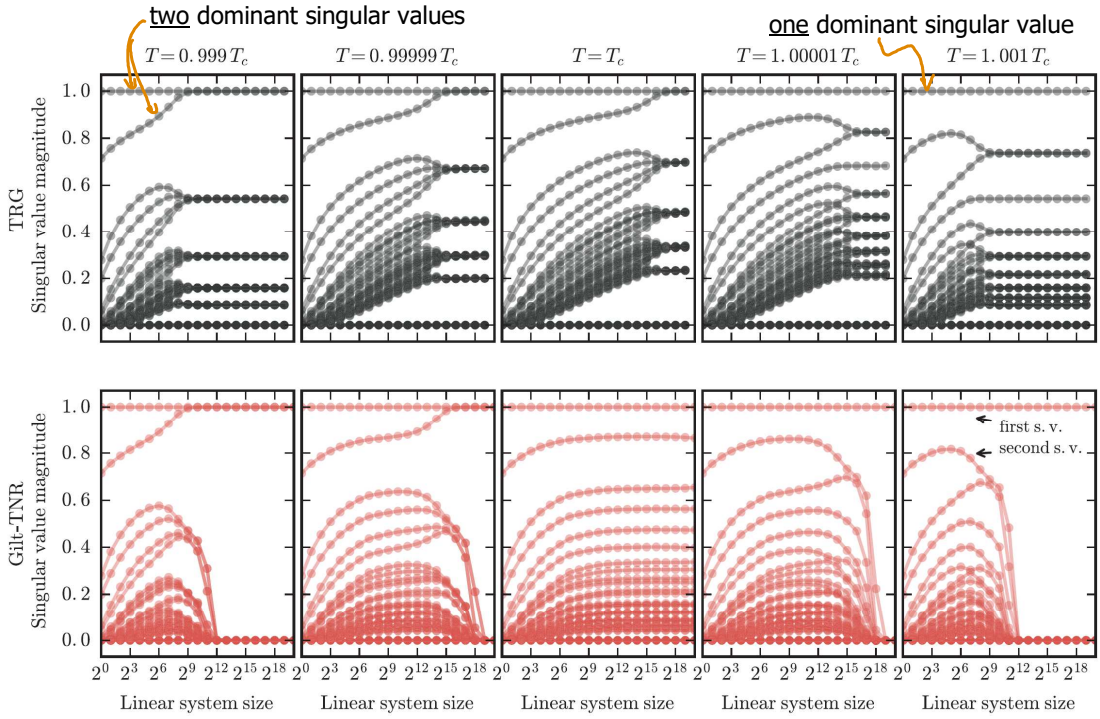


FIG. 6. RG flow of the coarse-grained tensors, illustrated for TRG (top row) and Gilt-TNR (bottom row) for five different temperatures. he horizontal axis is the linear system size, or in other words the number of RG transformations applied. At each system size, the data points re the 60 largest singular values of the coarse-grained tensor, with the same decomposition as that shown in (18). Thus each of the lines follows ie development of one of the singular values along the RG flow. These singular values provide a rough, basis independent characterization of ie structure of the tensor. Note how, for TRG, the spectrum is different at every temperature, even at the end of RG flow, when a fixed point as been reached. In contrast, for Gilt-TNR, on both sides of the critical point the RG flow ends in a trivial fixed point characteristic of that hase, with either one or two dominant singular values. At the critical point a complex fixed point structure is found that comes from the CFT. his critical fixed point is maintained over several orders of magnitude in linear system size. These results were obtained with $\chi = 110$ for oth TRG and Gilt-TNR, and $\epsilon = 5 \times 10^{-8}$ for Gilt-TNR.

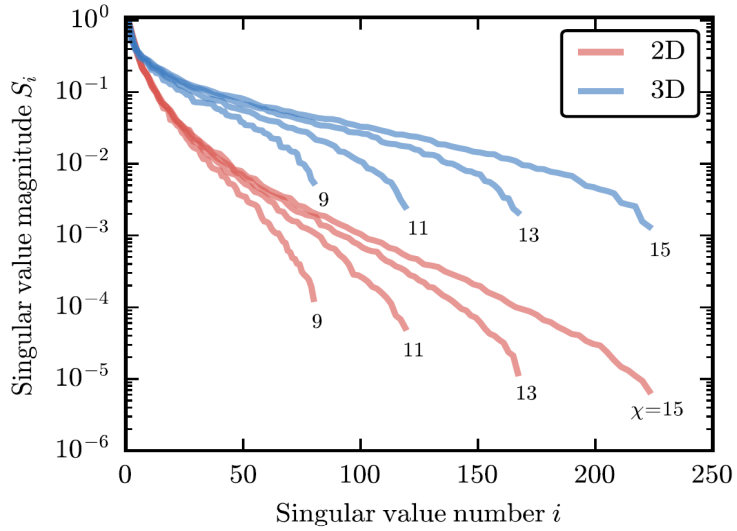
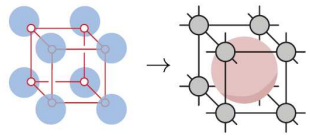
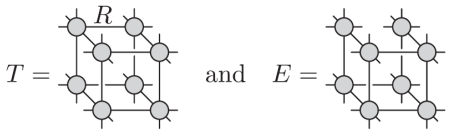


FIG. 7. Typical environment spectra of a single leg with respect to a square plaquette in 2D (bottom four spectra) and a cube in 3D (top four spectra), each labeled with the corresponding bond dimension χ. Recall that in each spectrum, the large values correspond to parts of the vector space of the leg 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

internal correlation:





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 the three-state Potts model.