Title: Renormalization of tensor networks using graph independent local truncations

Date: Nov 06, 2017 04:00 PM

URL: http://pirsa.org/17110108

Abstract: <p><span style="font-size:11.0pt;font-family:"Calibri",sans-serif;
mso-fareast-font-family:"Times New Roman";mso-bidi-font-family:"Times New Roman";
color:black;mso-ansi-language:EN-CA;mso-fareast-language:EN-US;mso-bidi-language:
AR-SA">I will describe our recent work from 1709.07460, where we introduce a new renormalization group algorithm for tensor networks. The algorithm is based on a novel understanding of local correlations in a tensor network, and a simple method to remove such correlations from any network. It performs comparably with the best competing algorithms on 2D/(1+1)D systems, but is significantly simpler to implement, and easier to generalize to different lattices and graphs, including to higher dimensions. I will begin the talk by discussing renormalization group methods for tensor networks in general, then describe our algorithm and its advantages, show some benchmark results, and finally comment on the status of implementing real-space renormalization for 3D tensor networks.</p>

Renormalization of tensor networks using graph independent local truncations

Markus Hauru, Clement Delcamp, Sebastian Mizera

arXiv: 1709.07460

2017-11-06, Perimeter Institute

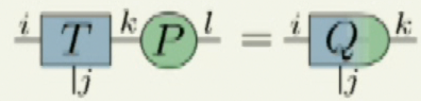
Abstract:

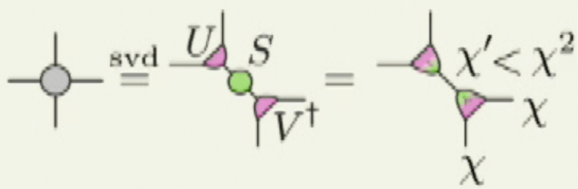
I will describe our recent work from 1709.07460, where we introduce a new renormalization group algorithm for tensor networks. The algorithm is based on a novel understanding of local correlations in a tensor network, and a simple method to remove such correlations from any network. It performs comparably with the best competing algorithms on 2D/(1+1)D systems, but is significantly simpler to implement, and easier to generalize to different lattices and graphs, including to higher dimensions. I will begin the talk by discussing renormalization group methods for tensor networks in general, then describe our algorithm and its advantages, show some benchmark results, and finally comment on the status of implementing real-space renormalization for 3D tensor networks.

Tensor network basics

$= A_{ijkl}$

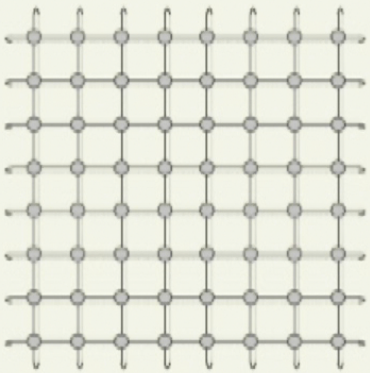
$$\sum_k T_{ijk} P_{kl} = Q_{ijl}$$





$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 10^{-9} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

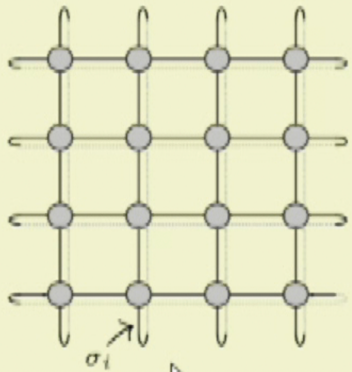
RG for tensor networks

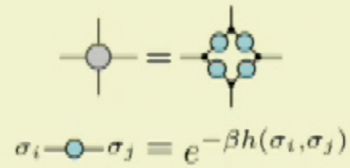


A partition function as a TN

Classical partition function

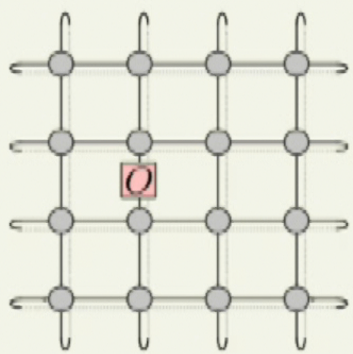
$$Z = \sum_{\sigma_1,\sigma_2,\ldots} \prod_{\langle i,j\rangle} e^{-\beta h(\sigma_i,\sigma_j)} =$$





Expectation value

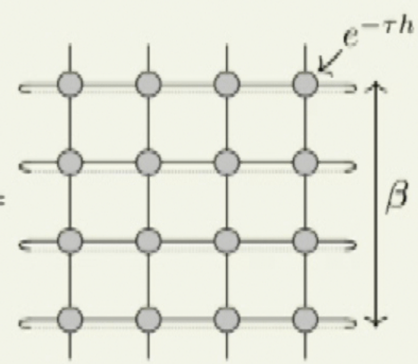
$$\langle O\rangle =$$



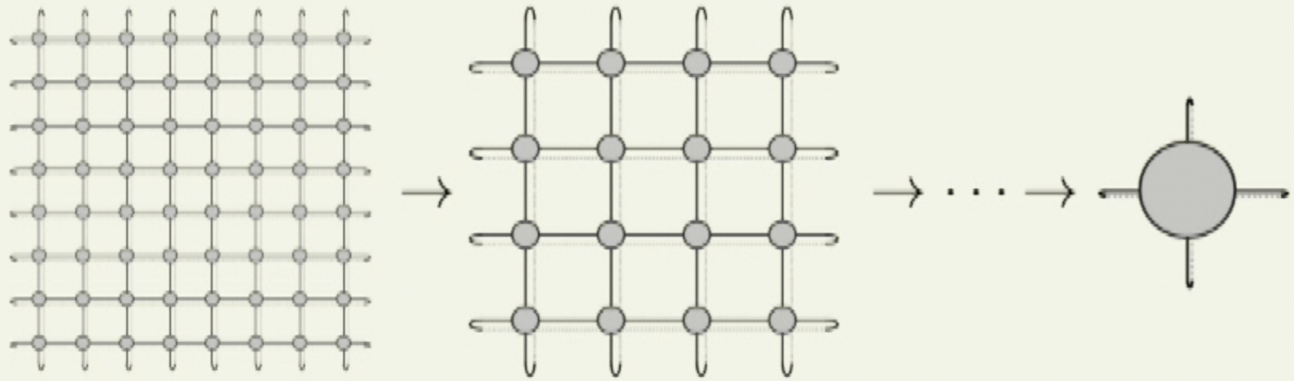
$$Z$$

Thermal quantum state

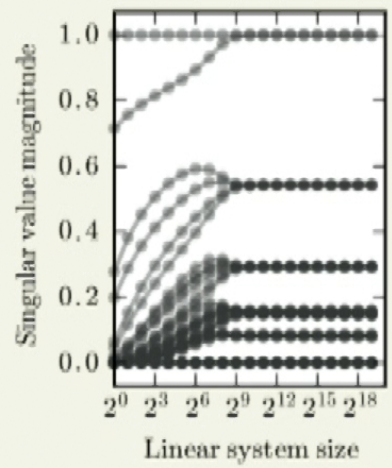
$$e^{-\beta H} =$$

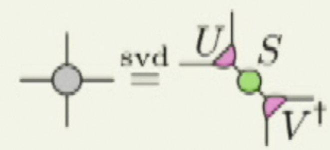


RG FOR TENSOR NETWORKS



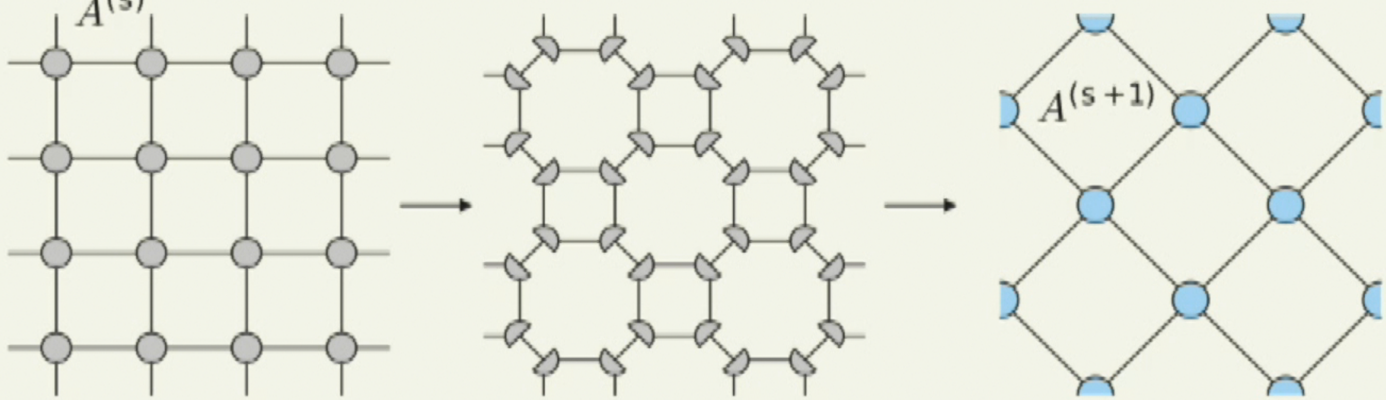
RG flows in the space of tensors!

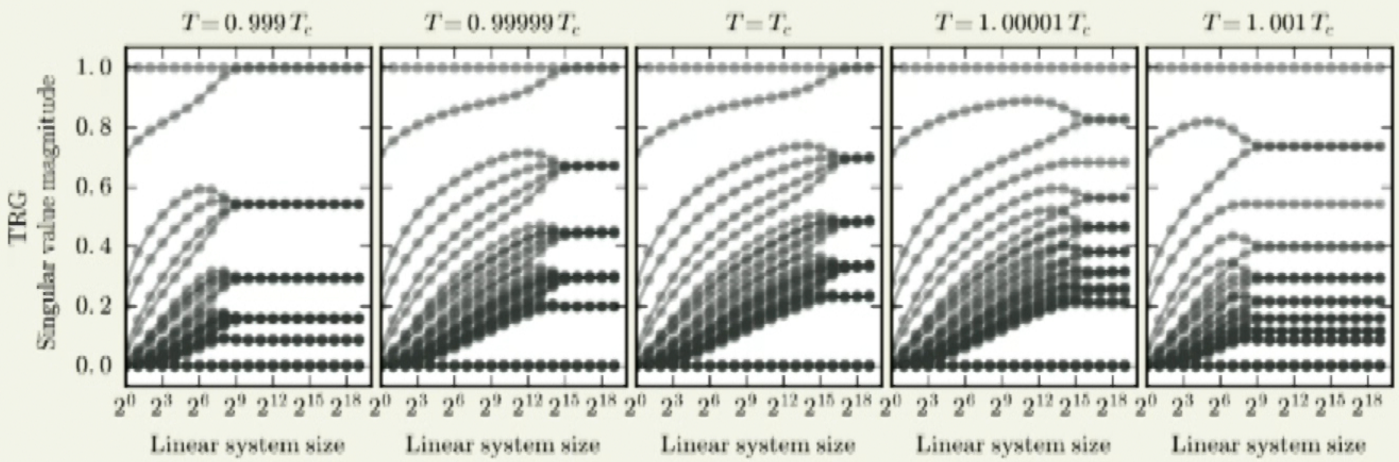




3/11

Tensor renormalization group (TRG)



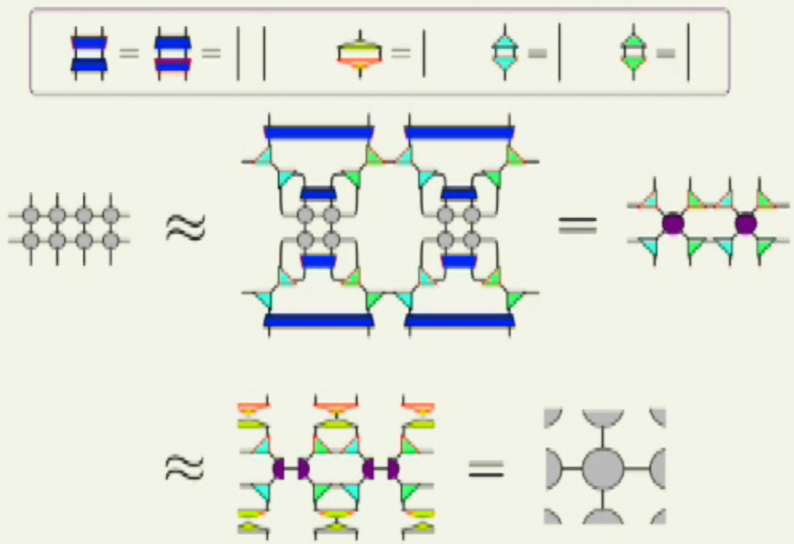


Levin & Nave, 0611687

4/11

Tensor network renormalization (TNR)

Solutions exist: Proper RG on tensor networks

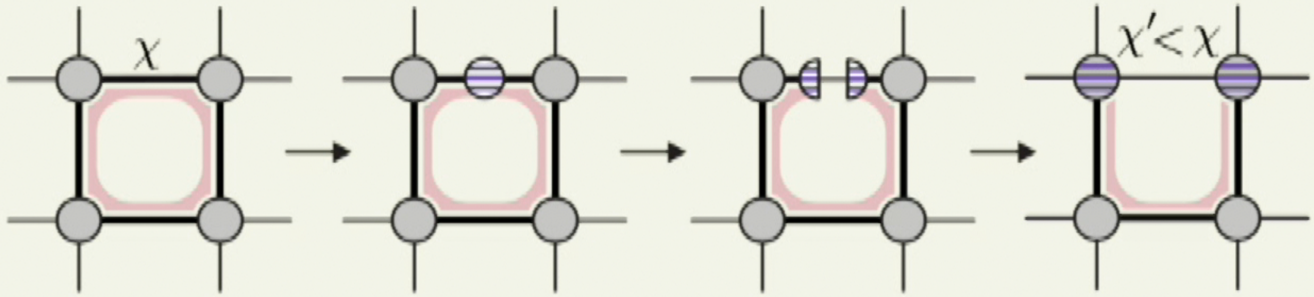


TNR, Evenbly & Vidal, 1412.0732

Loop-TNR, Yang et al., 1512.04938
TNR+, Bal et al., 1703.00365
Skeletonization, Ying, 1607.00050

5/11

A simpler, more generalizable way?



Aim: Remove local correlations by simply truncating a leg, with minimal modifications to the network.

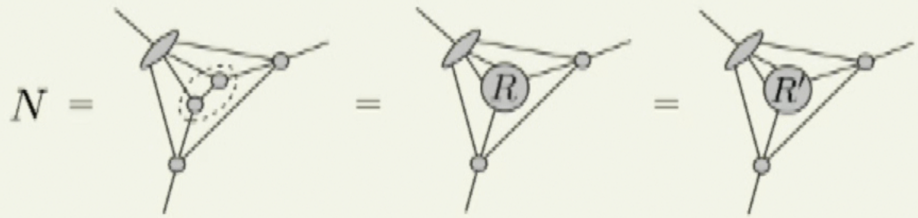
- Cheap
- Simple
- Works on any lattice/graph

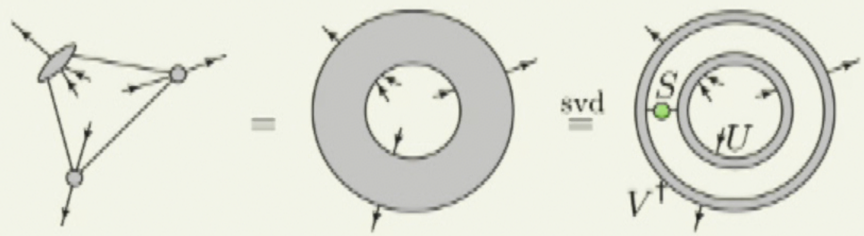
Solution: Graph independent local truncations (Gilt)

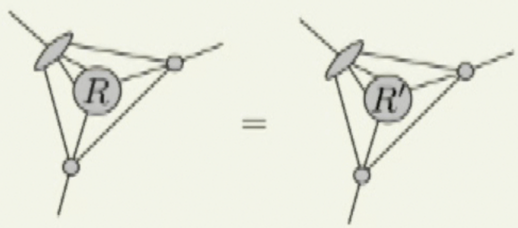
- "Please, tell me more!"

Allowed local changes in a TN

What are all the possible choices of R'?



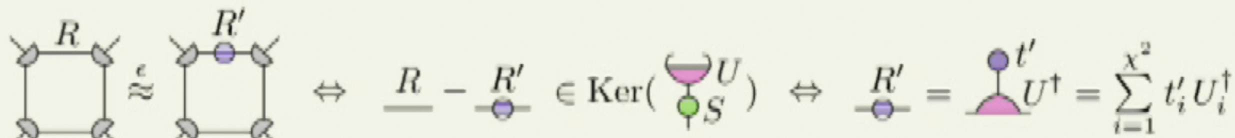




$\Leftrightarrow \quad R - R' \in \mathrm{Ker}(SU)$

(Gilt will be a special case of this generic stuff.)

Graph independent local truncation



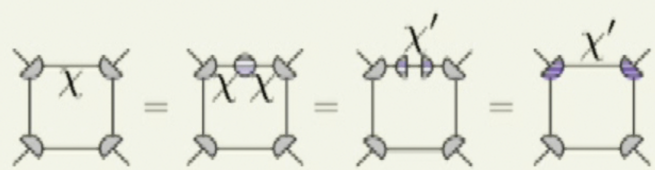
$$\frac{R'}{} = \ldots U^\dagger = \sum_{i=1}^{\chi^2} t'_i U_i^\dagger$$

with $t'_i = \mathrm{Tr}\, U_i$ for those i for which $S_i > \epsilon$.

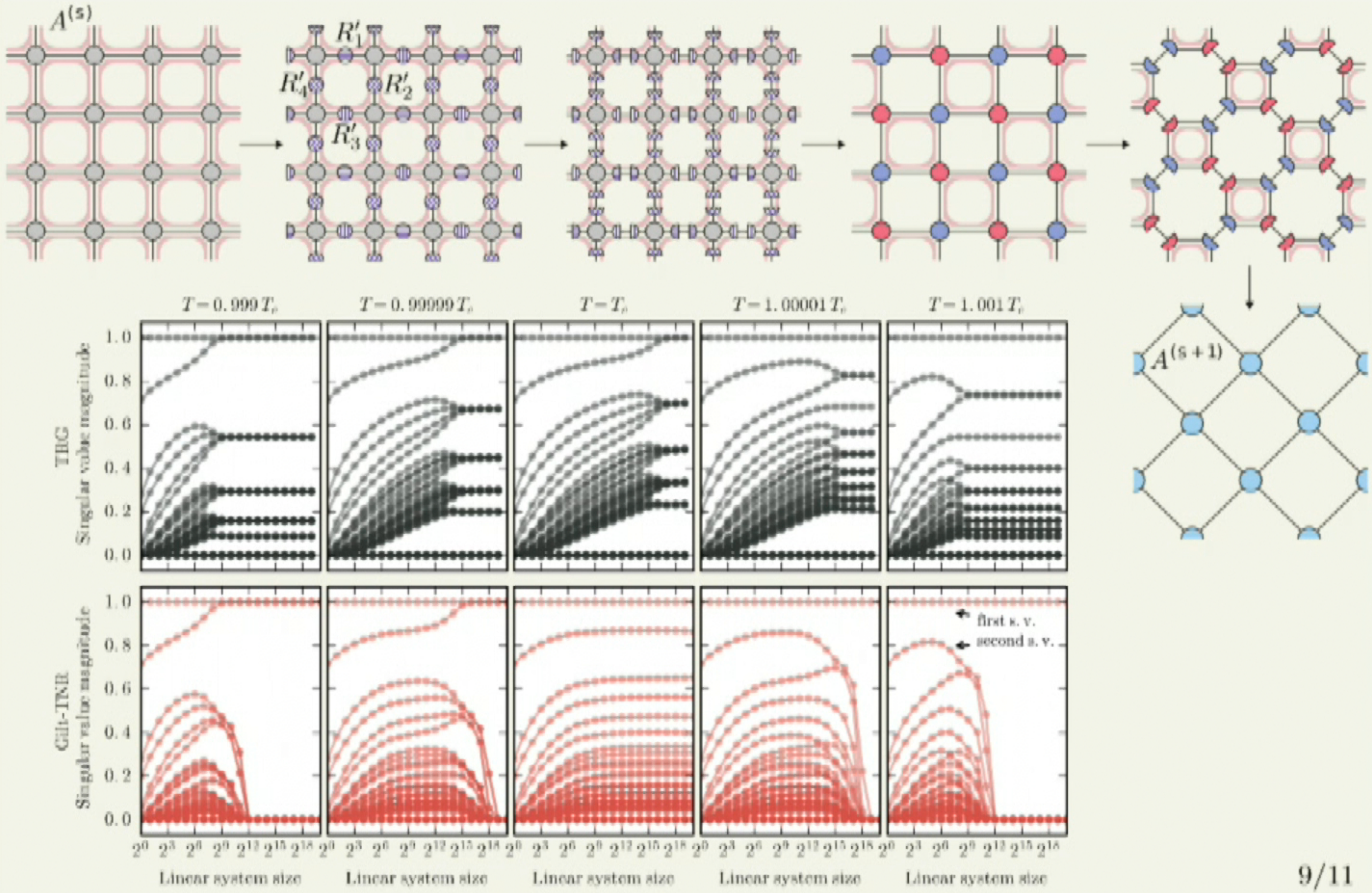
We are free to choose t'_i as we wish, for those i for which $S_i \leq \epsilon$!

We should choose them to minimize $\mathrm{rank}(R') = \chi'$.

That's tricky to do, so instead we minimize an exactly solvable cost function, that incentivizes similar solutions. It does the job, except for corner cases.

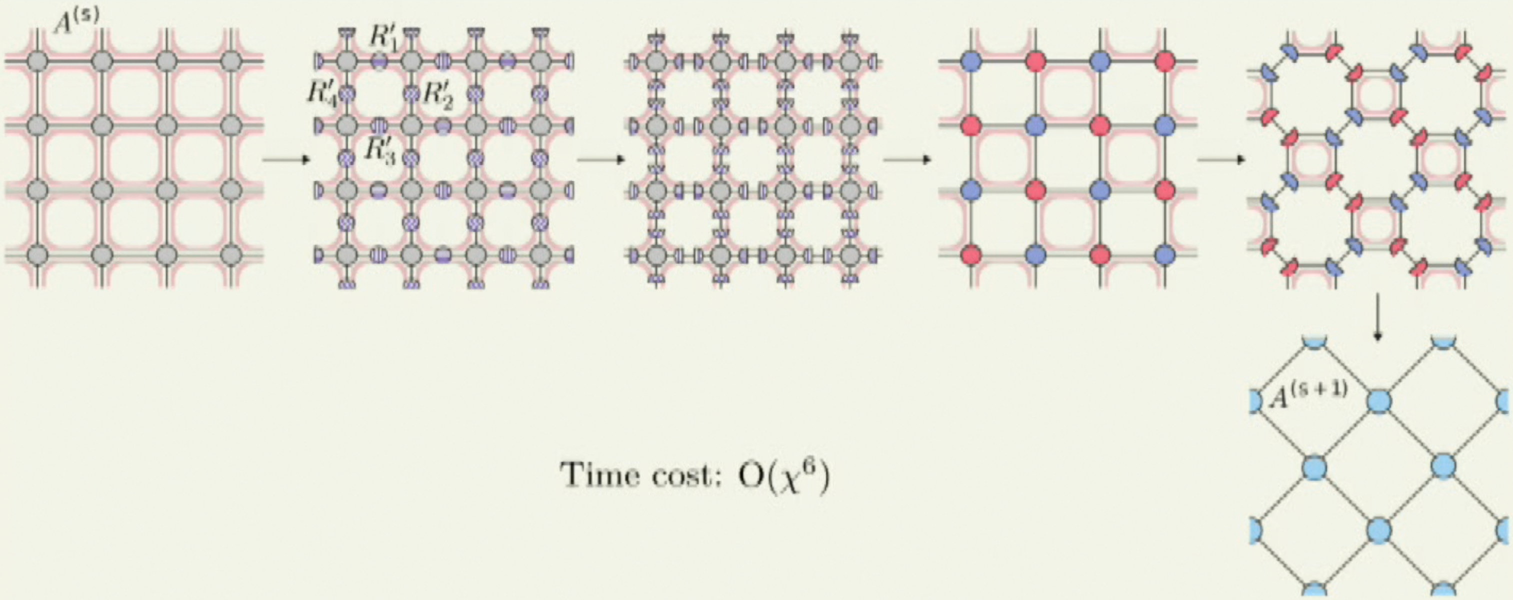


Gilt-TNR & 2D Classical Ising



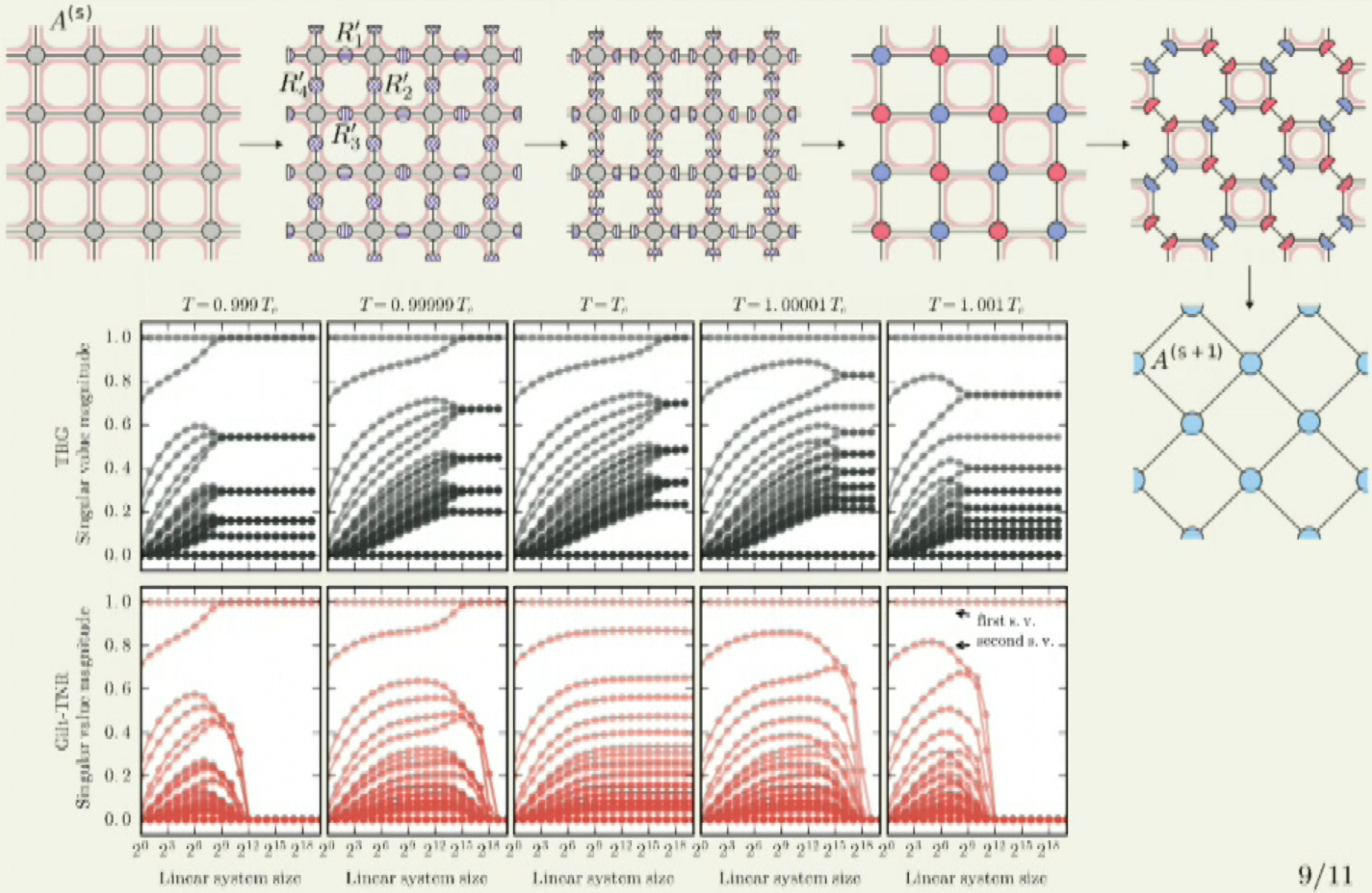
9/11

Gilt-TNR



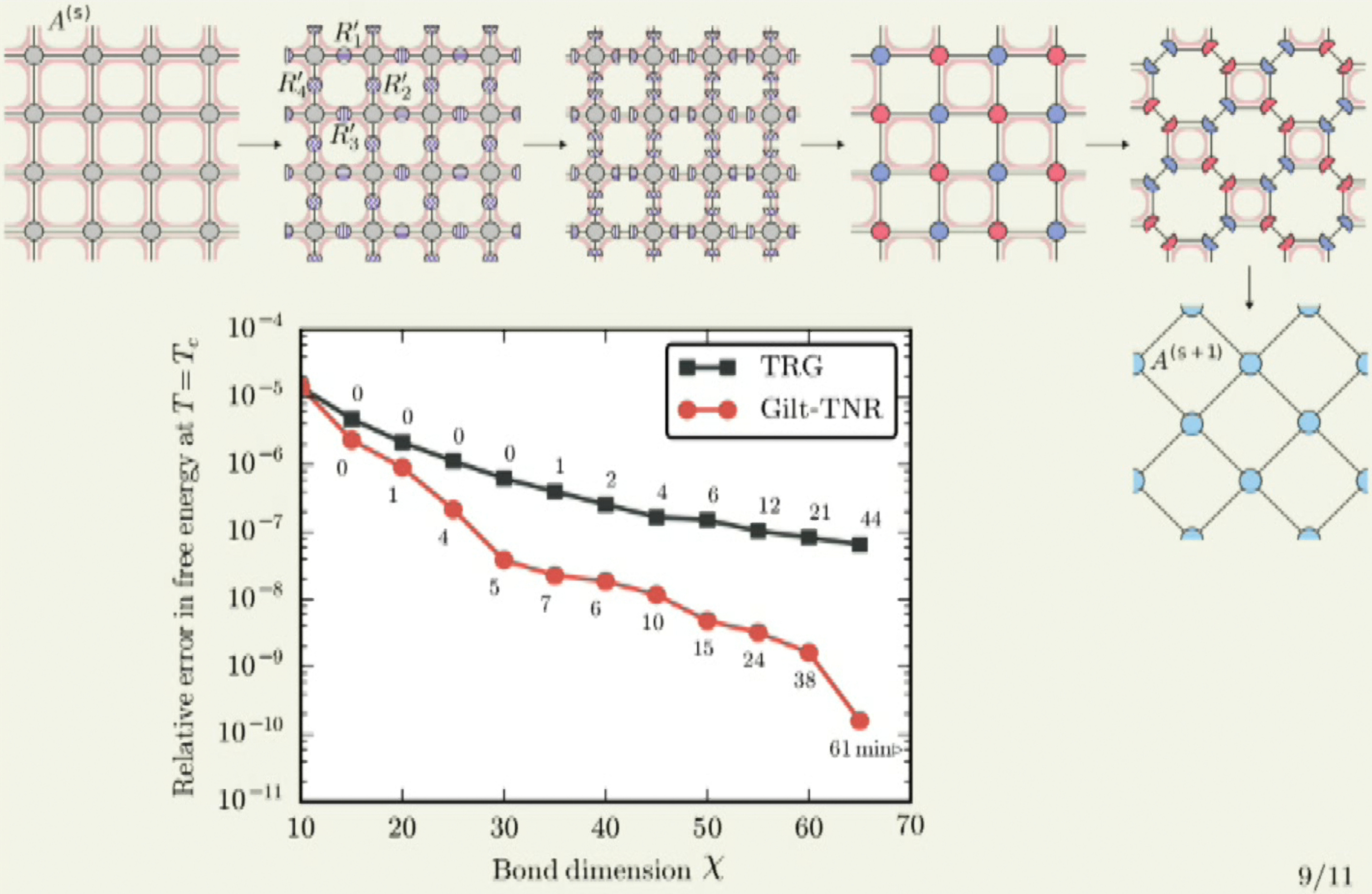
Time cost: $O(\chi^6)$

Gilt-TNR & 2D Classical Ising



9/11

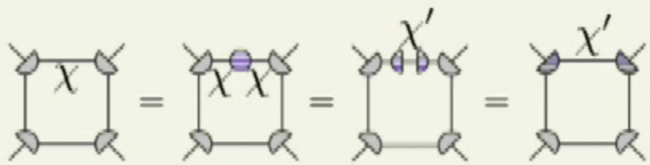
GILT-TNR & 2D CLASSICAL ISING



What have we achieved?

A proper tensor network RG method, based on Gilt, that

- performs comparably to the best in the market
- is simple to implement, requires no iterative optimization
- applies to any graph/lattice (with decent computational cost)



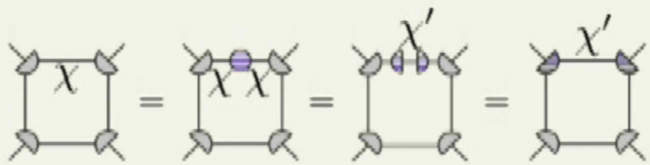
Next?

- What else could we use Gilt for, other than RG?
- 3D / (2+1)D

What have we achieved?

A proper tensor network RG method, based on Gilt, that

- performs comparably to the best in the market
- is simple to implement, requires no iterative optimization
- applies to any graph/lattice (with decent computational cost)



Next?

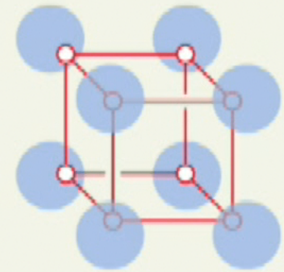
- What else could we use Gilt for, other than RG?
- 3D / (2+1)D

10/11

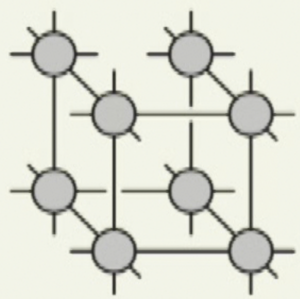
A word on 3D

Local correlations are much more abundant (area law).

$\Rightarrow$ Need to disentangle is more severe.

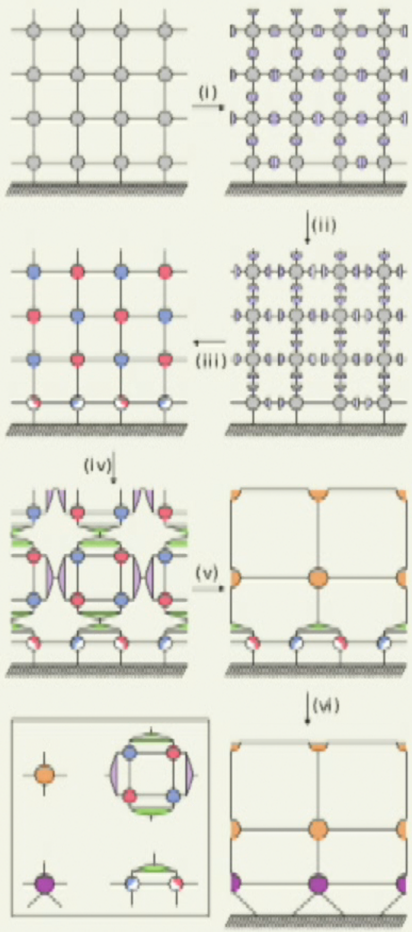


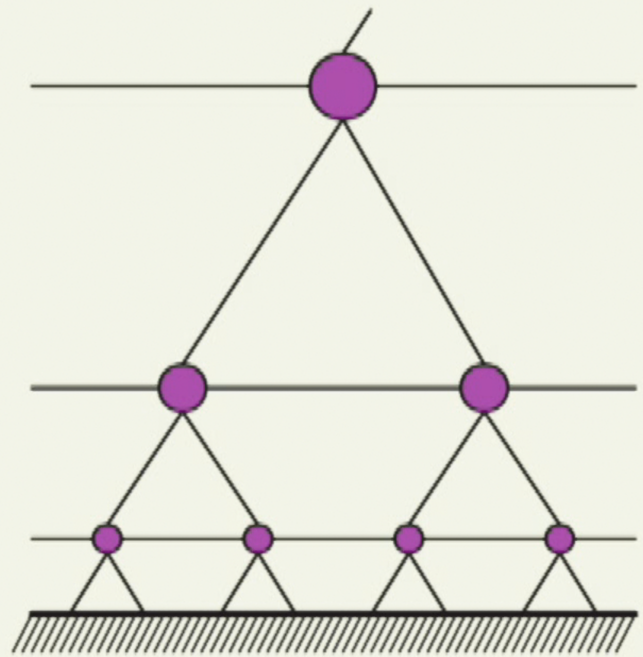
Gilt works the same, although cubes are more expensive than squares [$O(\chi^{12})$ or less].



Work in progress: https://github.com/Gilt-TNR/Gilt-TNR

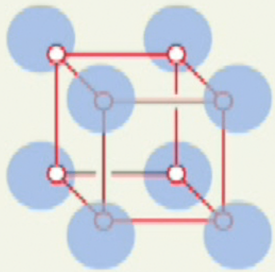
Quantum states



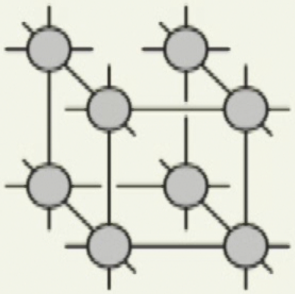


A word on 3D

Local correlations are much more abundant (area law).
$\Rightarrow$ Need to disentangle is more severe.



Gilt works the same, although cubes are more expensive than squares [$O(\chi^{12})$ or less].



Work in progress: https://github.com/Gilt-TNR/Gilt-TNR

11/11