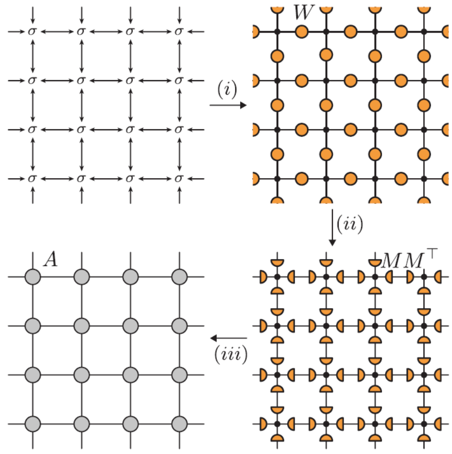
Tensor Renormalization Group (TRG) and related schemes (VUMPS, CTM, FCTM)

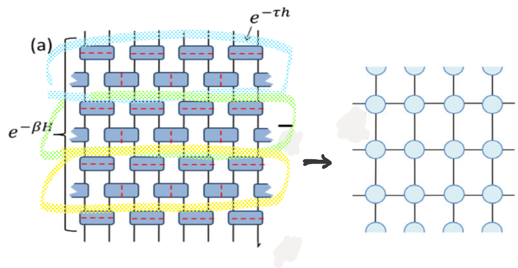
Goal: Compute 2D contractions by coarse-graining RG schemes (instead of transfer matrix schemes)

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]

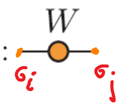
TRG-I.1

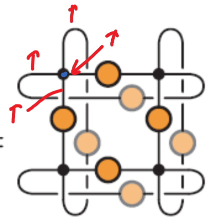
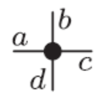
Spin Hamiltonian: $H(\{\sigma\}) = \sum_{\langle i,j \rangle} h(\sigma_i, \sigma_j)$

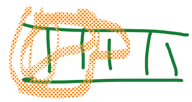
$\sigma_i = \{\uparrow, \downarrow\} = \{+1, -1\}$

$h(\sigma_i, \sigma_j) = -\sigma_i \sigma_j J_{ij}$

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

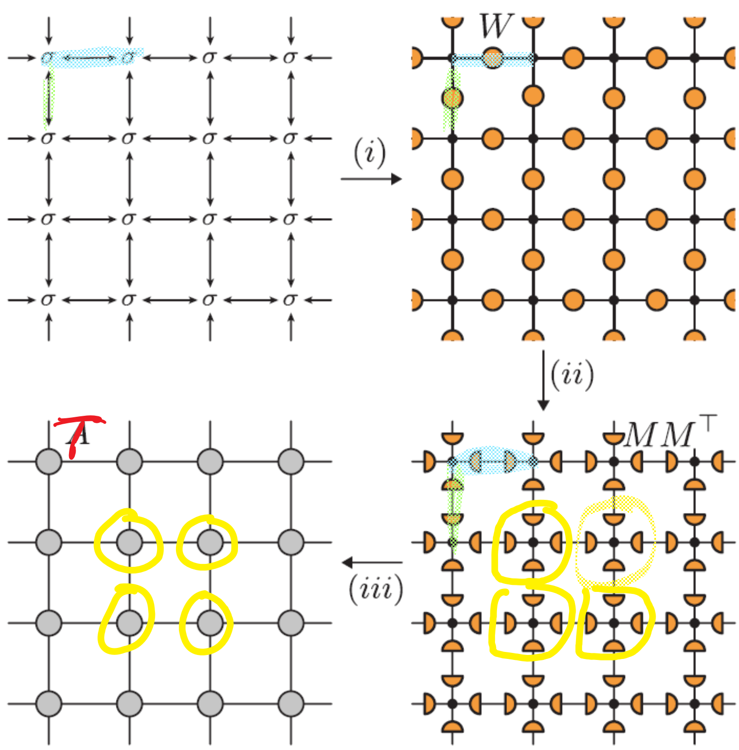
Bond weights: $W_{\sigma_i \sigma_j} = e^{-\beta h(\sigma_i, \sigma_j)} = \begin{pmatrix} W_{\uparrow\uparrow} & W_{\uparrow\downarrow} \\ W_{\downarrow\uparrow} & W_{\downarrow\downarrow} \end{pmatrix} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix} =:$ 

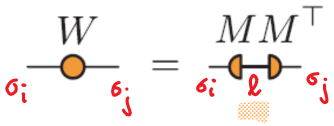
For 2x2 lattice (with periodic conditions): $Z =$  with $\delta_{abcd} =$ 



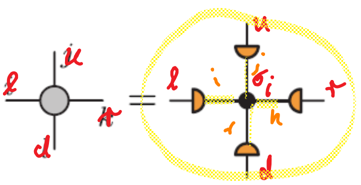
For inifinite 2D lattice,

we obtain a 2D tensor network:





$$M = \begin{pmatrix} \sqrt{\cosh \beta} & \sqrt{\sinh \beta} \\ \sqrt{\cosh \beta} & -\sqrt{\sinh \beta} \end{pmatrix}$$

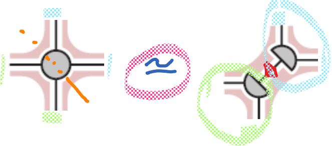
$T_{lurd} =$ 

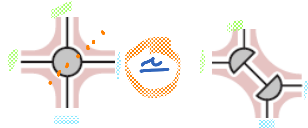
$$= \sum_{ijkl \in \{+,-1\}} \delta_{ijkl} M_{il} M_{ju} M_{kr} M_{ld}$$

$$= \sum_{\sigma} M_{\sigma l} M_{\sigma u} M_{\sigma r} M_{\sigma d}$$

Technical challenge: contract this infinite tensor network!

Do SVD on T in two different ways:

$T =$  $= U \sqrt{S} \sqrt{S} V^{\dagger}$

$T =$  $= U' \sqrt{S'} \sqrt{S'} V'^{\dagger}$

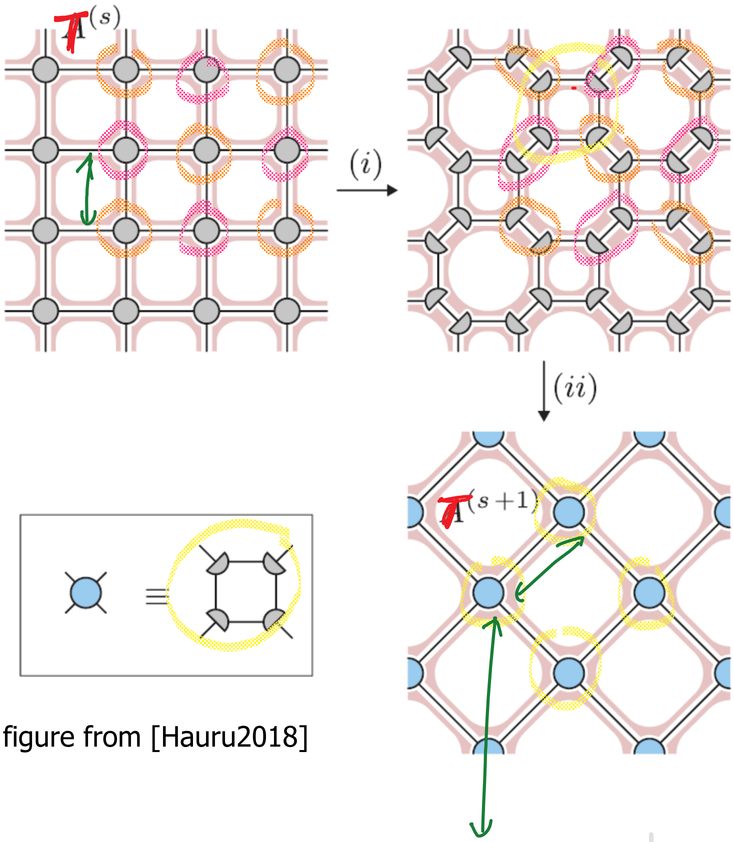
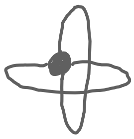


figure from [Hauru2018]

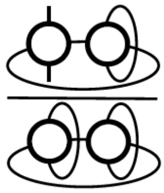
(ignore red shading)

Iterate until $T^{(s)}$ converges as $s \to \infty$

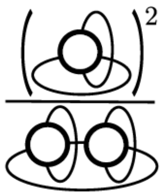
(reaches fixed point)

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

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

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

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

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

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

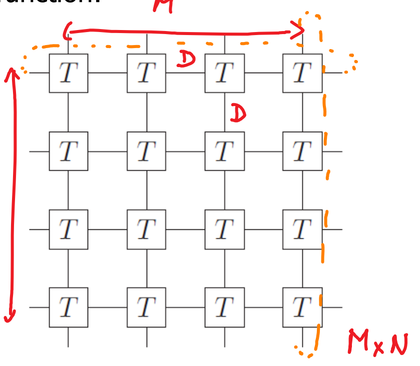
2. 2D contractions via Variational Uniform Matrix Product States (VUMPS)

[Fishman2018]

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

Partition function:

$Z_{M,N} :=$

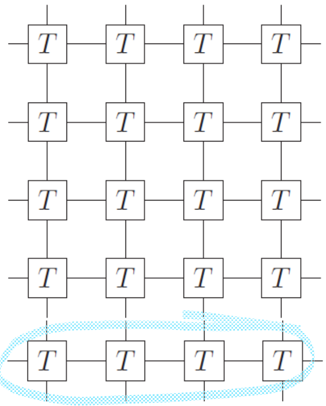


$=: (K^M)^N$ (1)

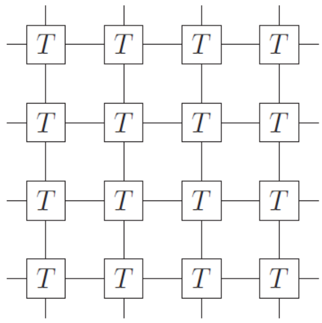
$K =$ partition function per site

each row contributes a factor K^M

$Z_{M,N+1} =$



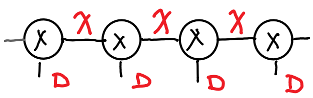
$\approx Z_{M,N} \cdot K^M =$



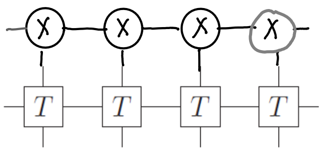
$\cdot K^M$ (2)

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

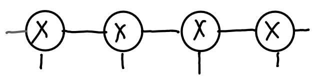
In limit $N \to \infty$, represent $Z_{M,N}$ by an 'upper boundary MPS':



Then:



$\approx$

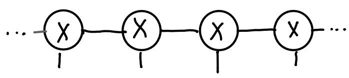


K^M

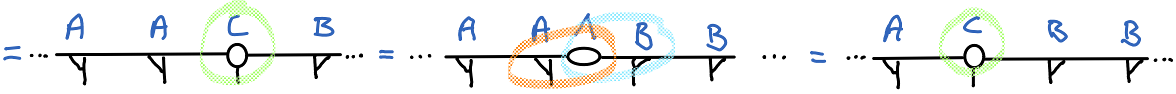
'row-to-row transfer matrix'

'fixed-point condition' (3)

In limit, $M \to \infty$,

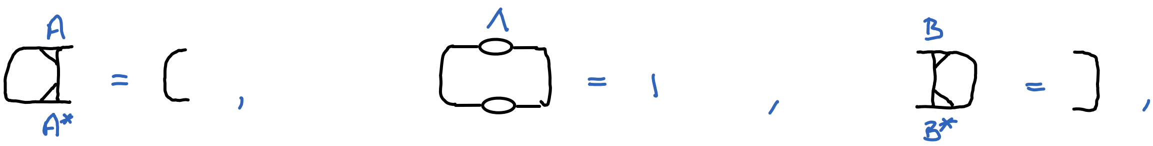


is translationally invariant. Express it in canonical form:



(4)

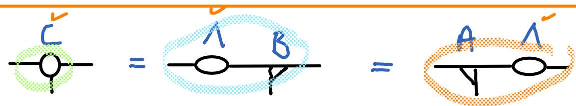
with



(5)

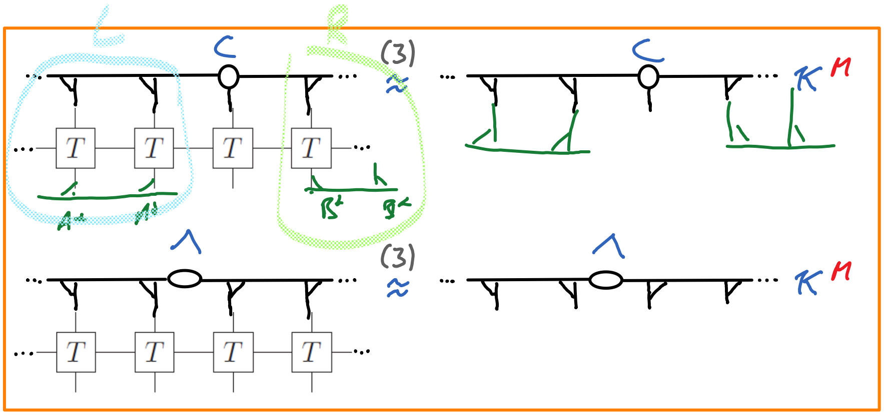
left-normalization, overall normalization, right-normalization

while C, Λ satisfy the 'gauge conditions':
which must hold on <u>all</u> sites.



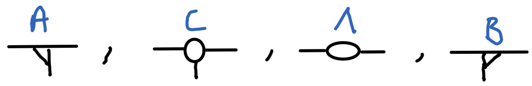
(6)

Fixed-point condition (3) implies:



(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!

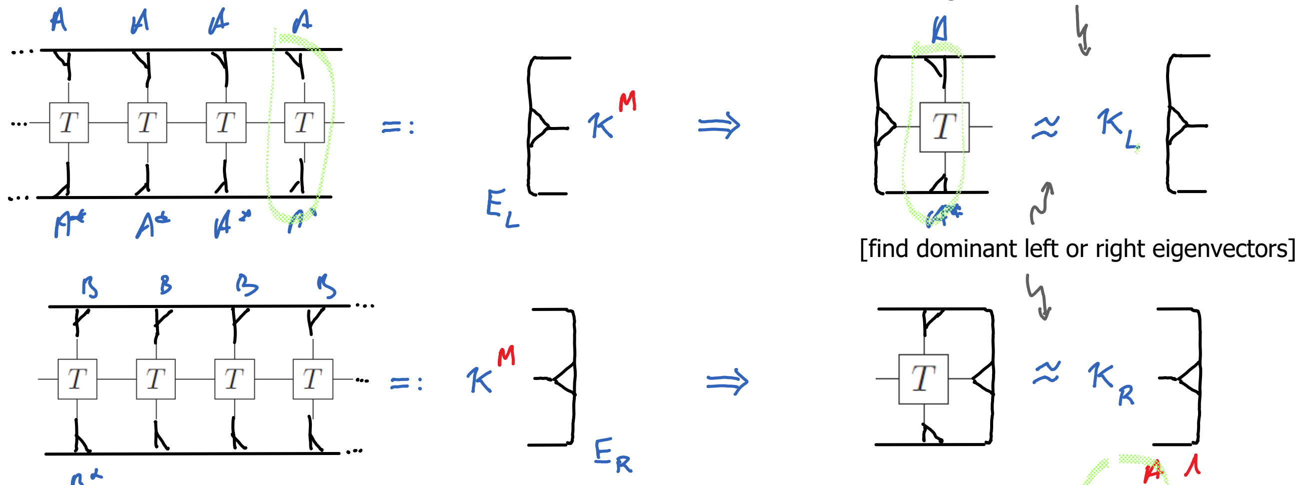
(6,7,8) have the same structure as when finding ground state of infinite uniform system.

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]

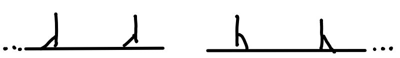


[find dominant left or right eigenvectors]

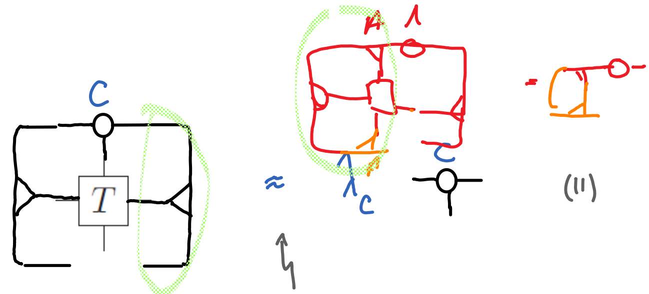
(9)

(10)

(ii) Solve for central tensor and bond tensor: C Λ

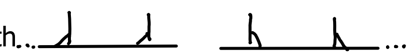
(7) contracted with 

and expressed through environmental tensors, implies:

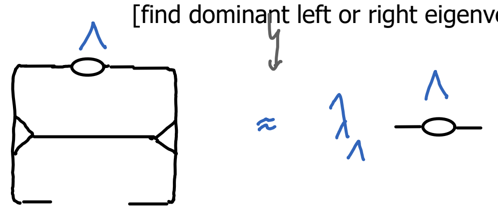


(11)

[find dominant left or right eigenvectors]

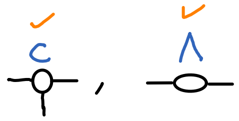
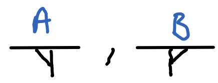
(8), contracted with 

and expressed through environmental tensors, implies:



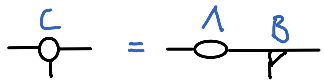
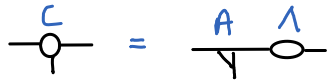
(12)

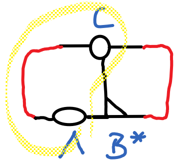
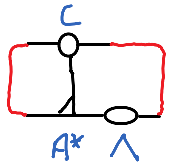
At or near fixed point: $\lambda_C \approx \lambda_\Lambda \kappa_L \approx \lambda_\Lambda \kappa_R$ [this follows by contracting (11) with  or ]

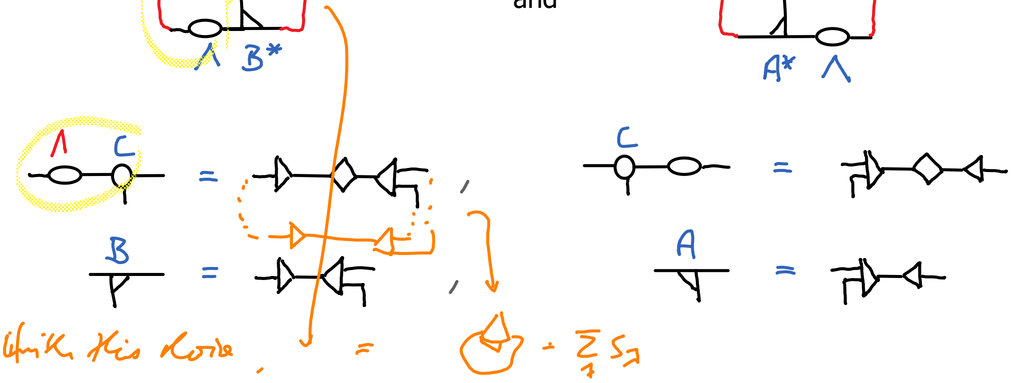
(iii) From  found in (ii), find new  that best satisfy (6), $\|C - A\Lambda\|^2$

$\|C - \Lambda B\|^2$

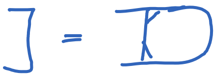
$\|C\|^2 + \|\Lambda B\|^2 - 2\|C \cdot \Lambda B\|$

 and  (13)

i.e. that maximize  and  (14)

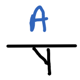
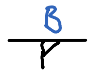
To that end, do SVDs:  (15)

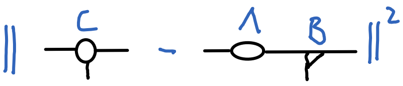
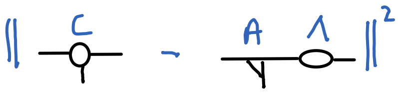
and choose new (16)



think this contr. $= \sum_\lambda s_\lambda$

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

There may be alternative schemes for finding optimal isometries  and  that satisfy (13), see 'Riemannian optimization', see [Hauru2021], [Li2023]. Those papers discuss how to optimize a cost function w.r.t. a tensor satisfying an isometry condition. Here, the cost functions would be

 and  (17)

and the isometry conditions are Eqs. (5).