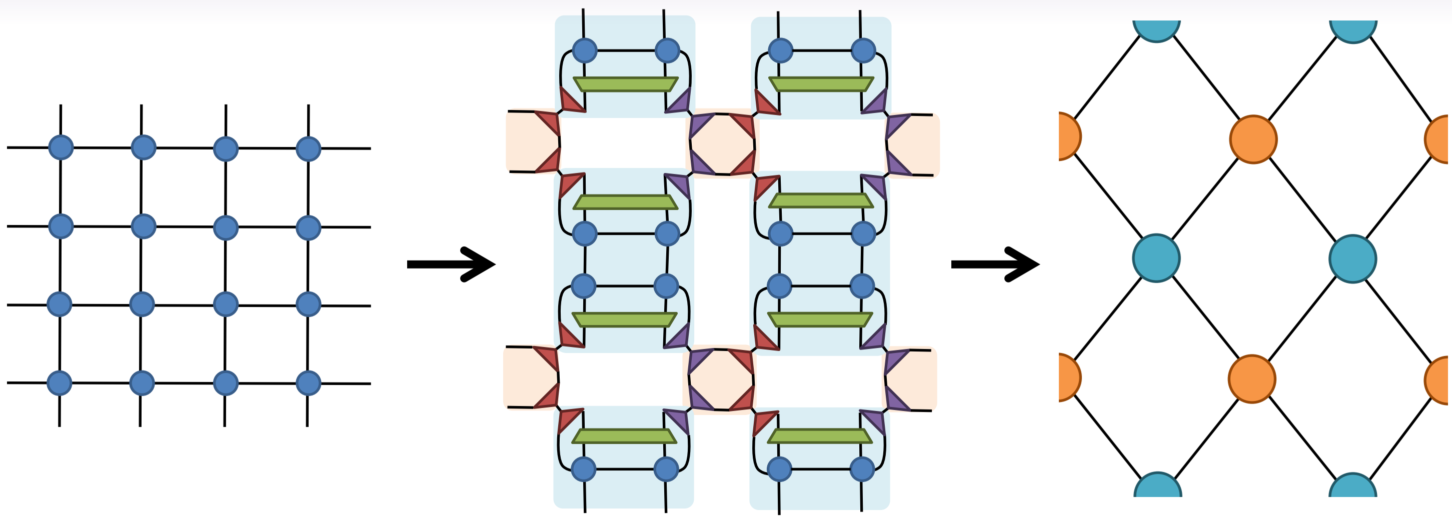
KITP, May 2015

Disentangling Tensor Networks



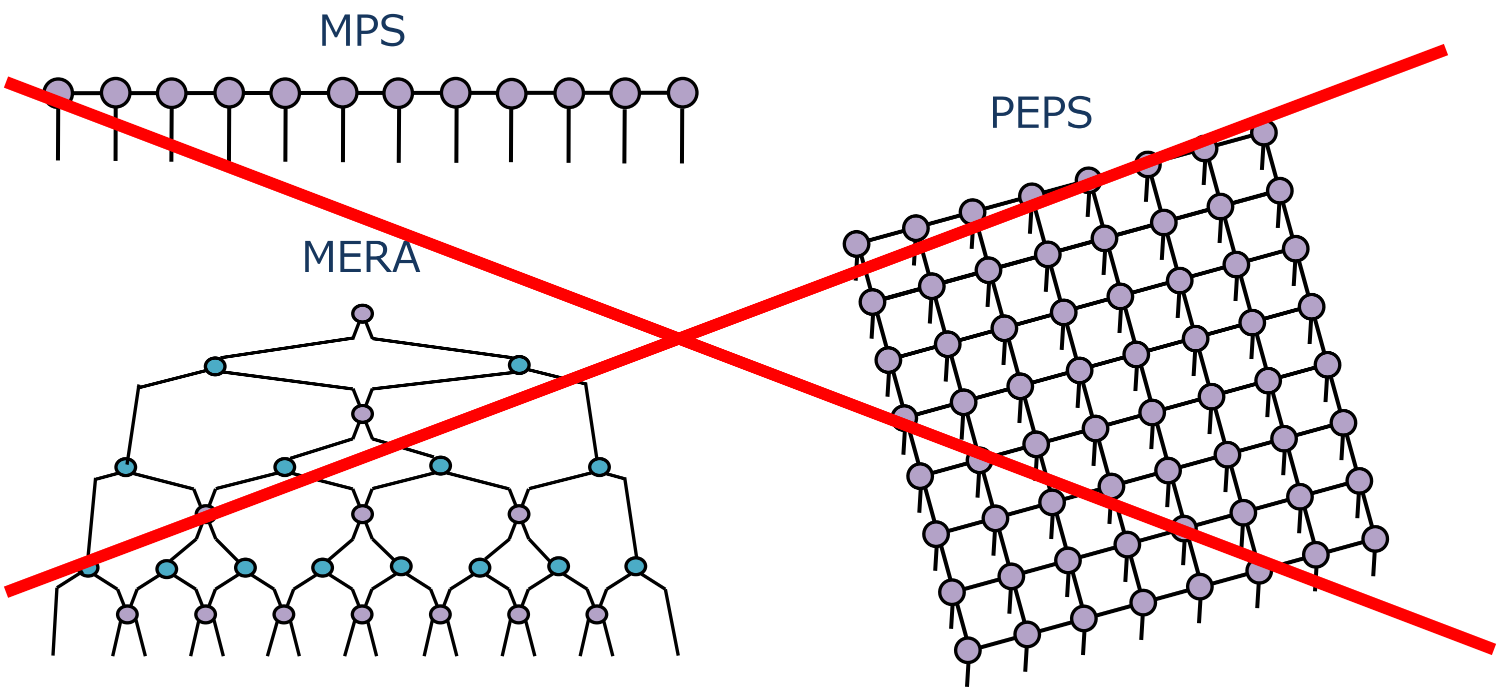
Glen Evenbly
Guifre Vidal

Tensor Network Renormalization, arXiv:1412.0732

Overview

Proper consideration of entanglement is important in the study quantum many-body physics

Tensor Network Ansatz: wavefunctions designed to reproduce ground state entanglement scaling



Today: consideration of entanglement in designing a real-space renormalization transformation

Outline: Tensor Network Renormalization

Overview

The set-up: Representation of partition functions and path integrals as tensor networks

Previous approaches: Levin and Nave's Tensor Renormalization Group (LN-TRG), conceptual and computation problems.

New approach: Tensor network renormalization (TNR): proper removal of all short-ranged degrees of freedom via disentanglers

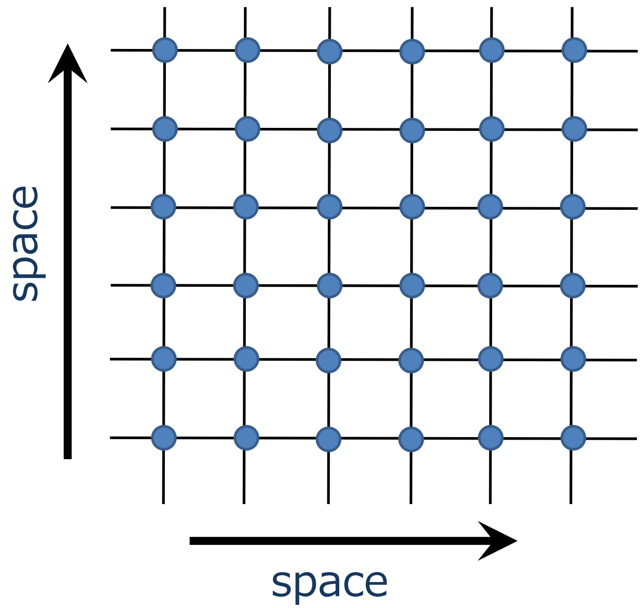
Benchmark results

Extensions

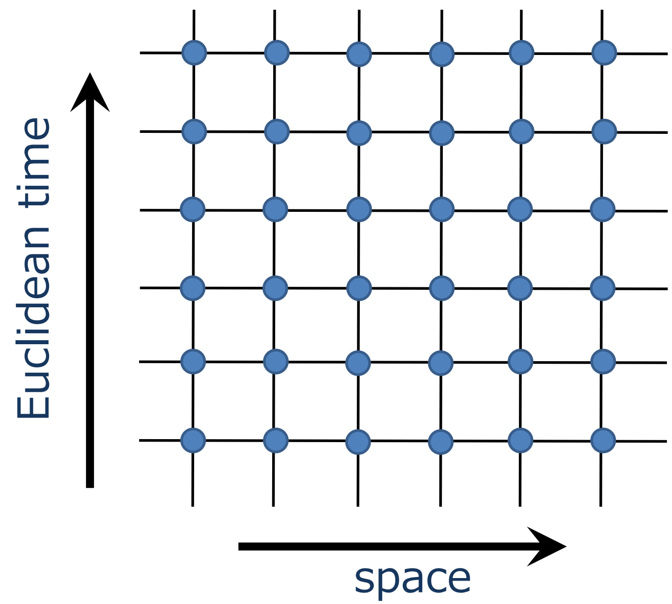
Overview

express many-body system as a tensor network:

partition function of 2D classical statistical model



Euclidean path integral of 1D quantum model

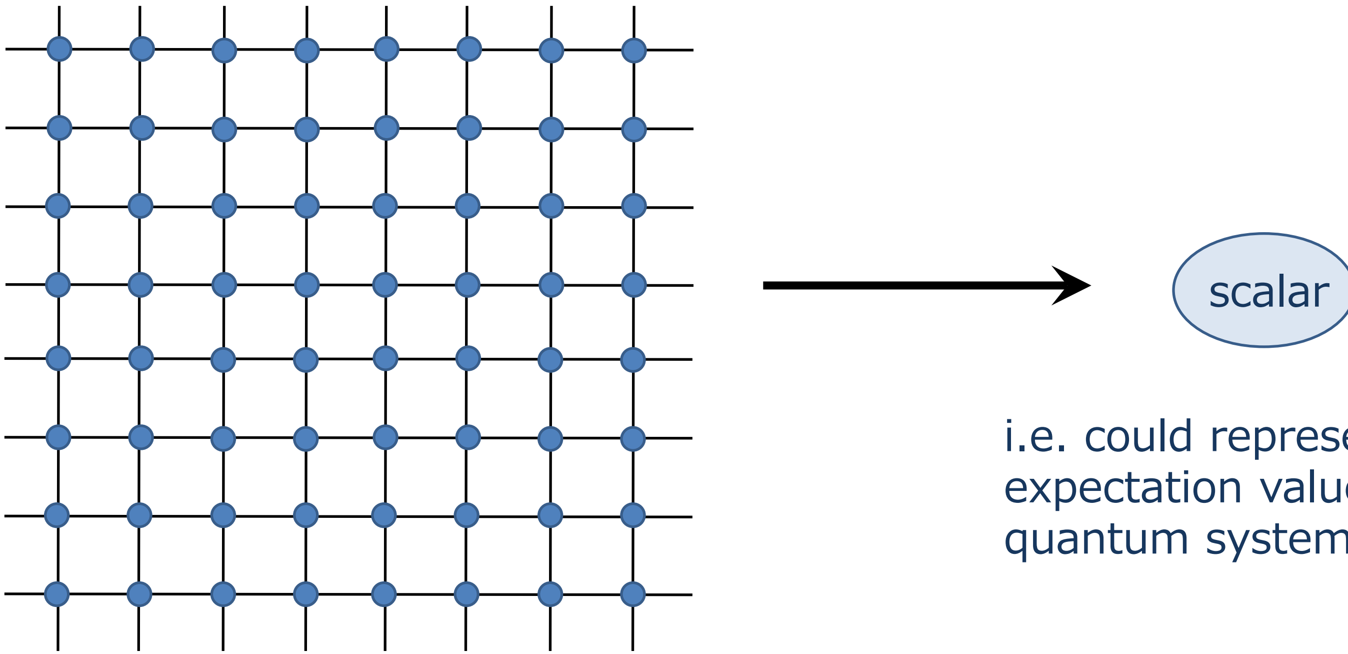


- tensors encode Boltzmann weights
- contraction of tensor network equals weighted sum over all microstates

- row of tensors encodes small evolution in imaginary time
- contraction of tensor network equals weighted sum over all trajectories

Overview

Goal: to contract the tensor network to a scalar:



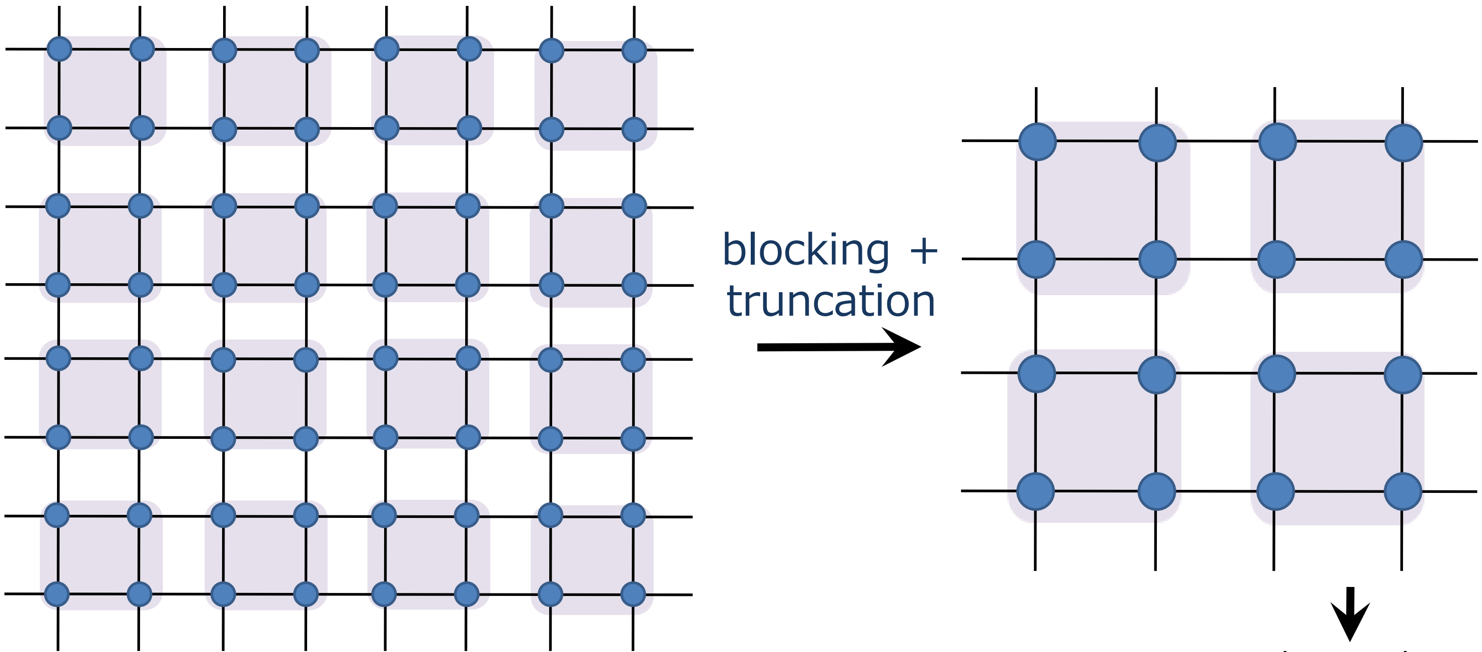
i.e. could represent an expectation value in the quantum system $\langle\psi|o|\psi\rangle$

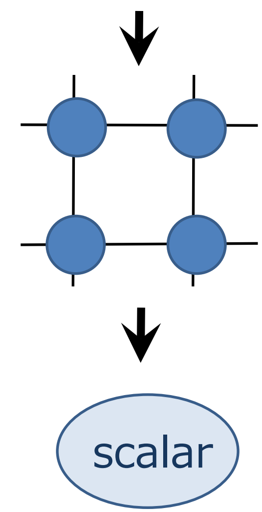
Approaches:

- Monte-carlo sampling
- Transfer matrix methods
- Real-space renormalization (coarse-graining)

Overview

Goal: to contract the tensor network to a scalar:





Approaches:

- Monte-carlo sampling
- Transfer matrix methods
- Real-space renormalization (coarse-graining)

Overview

Basic idea of RG:

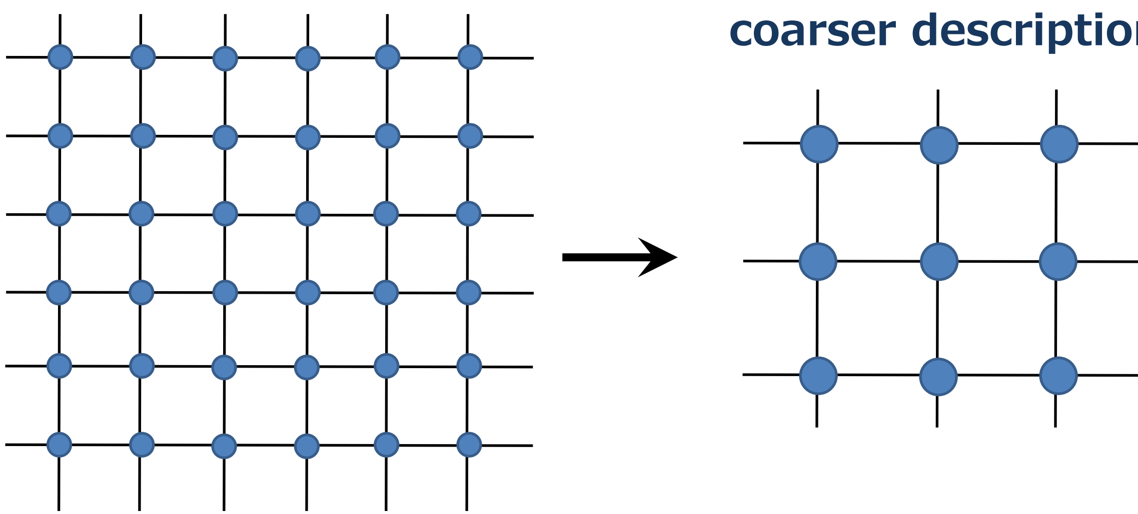
description in terms of very many microscopic degrees of freedom

Iterative RG transformations →

description in terms of a few effective (low-energy, long distance) degrees of freedom

each transformation removes short-range (high energy) degrees of freedom

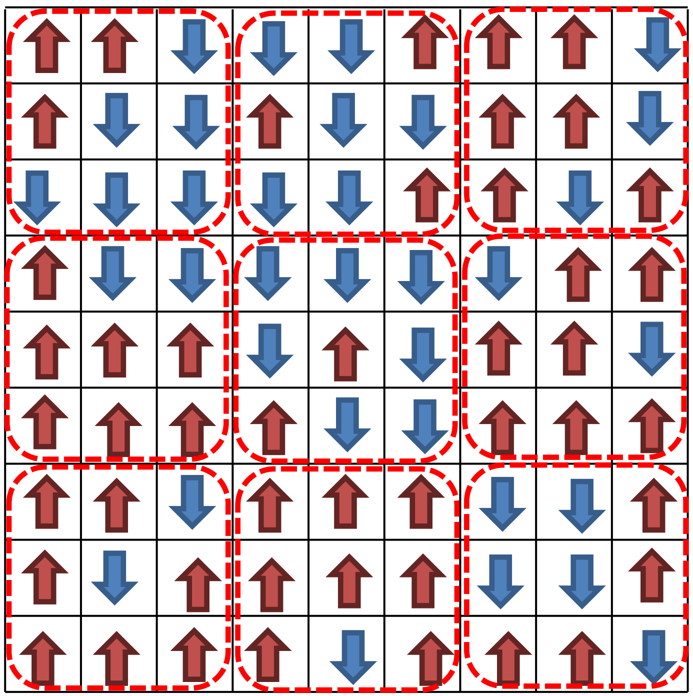
initial description → **coarser description**



Overview

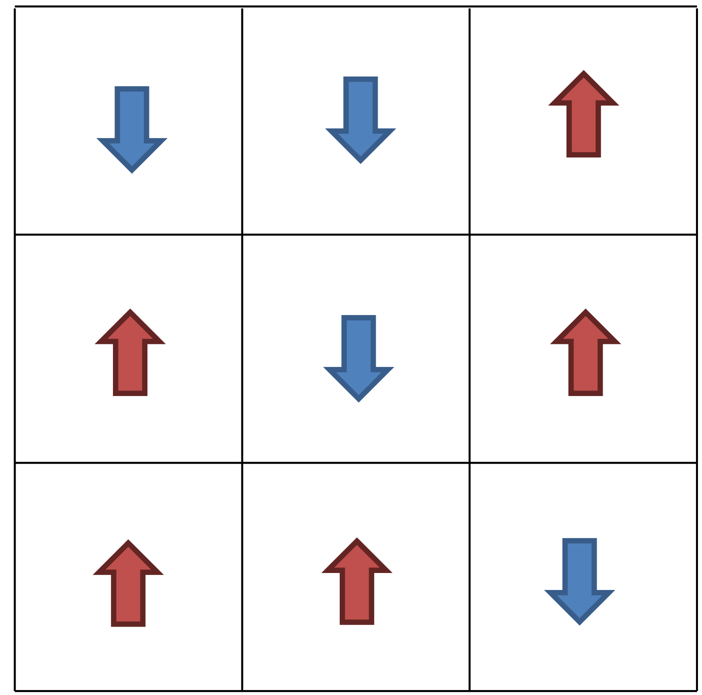
Early real-space RG: Kadanoff's "spin blocking" (1966)

lattice of classical spins



majority vote blocking

coarser lattice



initial description: $H(T, J)$

renormalized parameters: $H(T', J')$

…successful only for certain systems

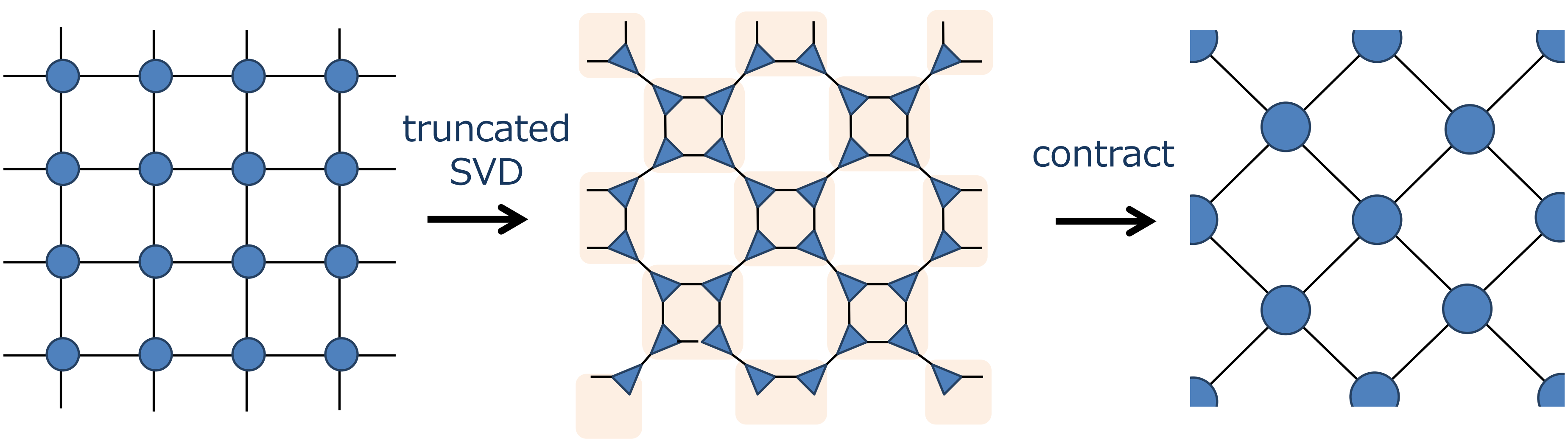
Overview

L.P. Kadanoff (1966): “Spin blocking”

spiritual successor

Key change: a more general prescription for deciding which degrees of freedom can safely be removed at each RG step

Levin, Nave (2006) : “Tensor renormalization group (LN-TRG)”



Overview

L.P. Kadanoff (1966): "Spin blocking"

spiritual successor

Key change: a more general prescription for deciding which degrees of freedom can safely be removed at each RG step

Levin, Nave (2006) : "Tensor renormalization group (LN-TRG)"

+ many improvements and generalizations:

Xie, Jiang, Weng, Xiang (2008): "Second Renormalization Group (SRG)"

Gu, Levin, Wen (2008): "Tensor Entanglement Renormalization Group (TERG)"

Gu, Wen (2009): "Tensor Entanglement Filtering Renormalization(TEFR)"

Xie, Chen, Qin, Zhu, Yang, Xiang (2012): "Higher Order Tensor Renormalization Group (HOTRG)"

Overview

L.P. Kadanoff (1966): "Spin blocking"

spiritual successor

Key change: a more general prescription for deciding which degrees of freedom can safely be removed at each RG step

Levin, Nave (2006) : "Tensor renormalization group (LN-TRG)"

Today: introduce new method of tensor RG (for partition functions and path integrals) that resolves significant **computational and conceptual problems** of previous approaches

Overview

Flaw with previous tensor RG methods (such as LN-TRG):

still contains microscopic "baggage"

original microscopic description → RG step → coarser effective description → RG step → coarser effective description → RG step

some short-ranged correlation propagated

some short-ranged correlation removed

Flaw: each RG step removes some (but not all) of the short-ranged degrees freedom

Consequences:

- Accumulation of short ranged detail can cause computational breakdown; cost scales exponentially in RG step! ✗
- Effective theory still contains unwanted microscopic detail; one does not recover proper structure of RG fixed points ✗

Overview

New approach: **"Tensor Network Renormalization (TNR)"** arXiv:1412.0732

A way of implementing real-space RG that addresses all short-ranged degrees of freedom at each RG step

original microscopic description $\xrightarrow{\text{RG step}}$ coarser effective description $\xrightarrow{\text{RG step}}$ coarser effective description $\xrightarrow{\text{RG step}}$

all short-ranged correlation removed | all short-ranged correlation removed | all short-ranged correlation removed

Advantages:

- **Proper RG flow is achieved**, TNR reproduces the correct structure of RG fixed points
- Prevents harmful accumulation of short-ranged detail, allowing for a sustainable RG flow:

exponential cost scaling (previous tensor RG) → constant cost (new approach, TNR)

Outline: Tensor Network Renormalization

Overview

The set-up: Representation of partition functions and path integrals as tensor networks

Previous approaches: Levin and Nave's Tensor Renormalization Group (LN-TRG), conceptual and computation problems.

New approach: Tensor network renormalization (TNR): proper removal of all short-ranged degrees of freedom via disentanglers

Benchmark results

Extensions

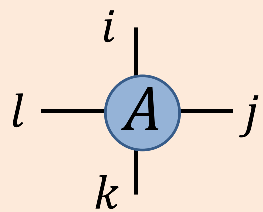
Overview: Tensor Networks

bond dimension

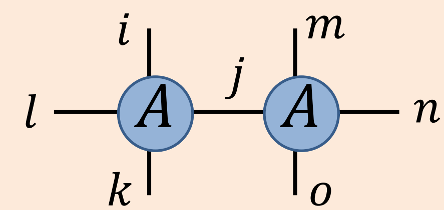
Let A_{ijkl} be a four index tensor with $i, j, k, l \in \{1,2,3, \dots, \chi\}$

i.e. such that the tensor is a $\chi \times \chi \times \chi \times \chi$ array of numbers

Diagrammatic notation:

$A_{ijkl} \longleftrightarrow$ 

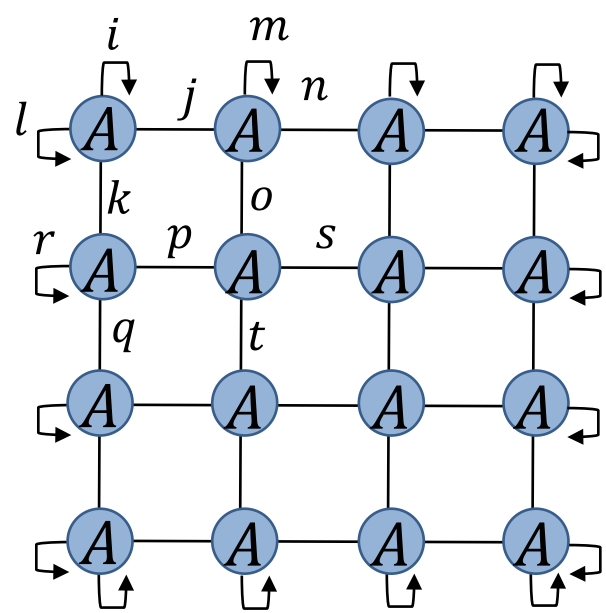
Contraction of two tensors:

$\sum_j A_{ijkl} A_{mnoj} \longleftrightarrow$ 

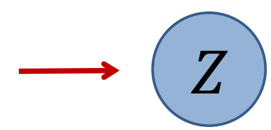
Square lattice network (PBC):

$$\sum_{ijklmn\dots} A_{ijkl} A_{mnoj} A_{kpqr} A_{ostp} \dots$$

$$\equiv \mathrm{tTr}\left(\bigotimes_{x=1}^{N} A \right) = \mathrm{Z}$$

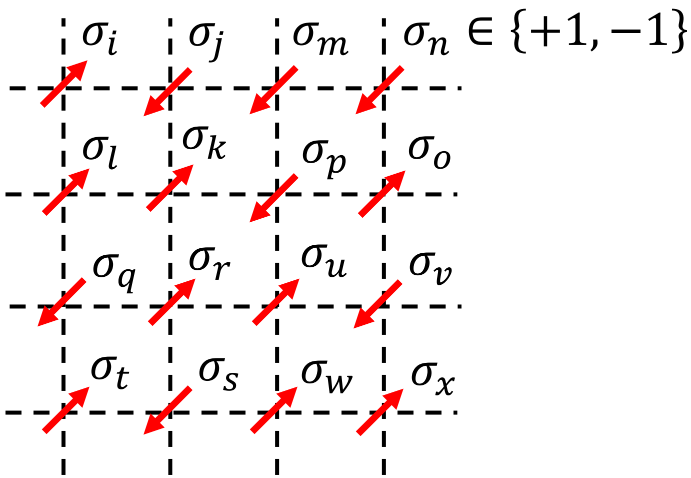


Contracts to a scalar:

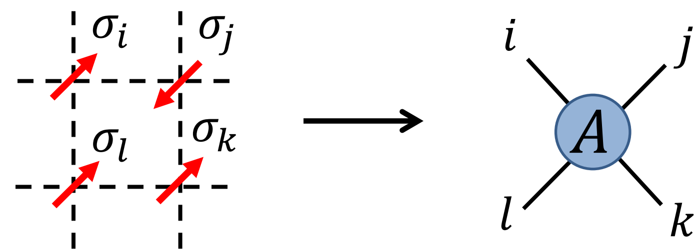


Partition functions as Tensor Networks

Square lattice of Ising spins:



Encode the Boltzmann weights of a plaquette of spins in a four-index tensor



Hamiltonian functional for Ising ferromagnet:

$$H(\{\sigma\}) = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

Partition function:

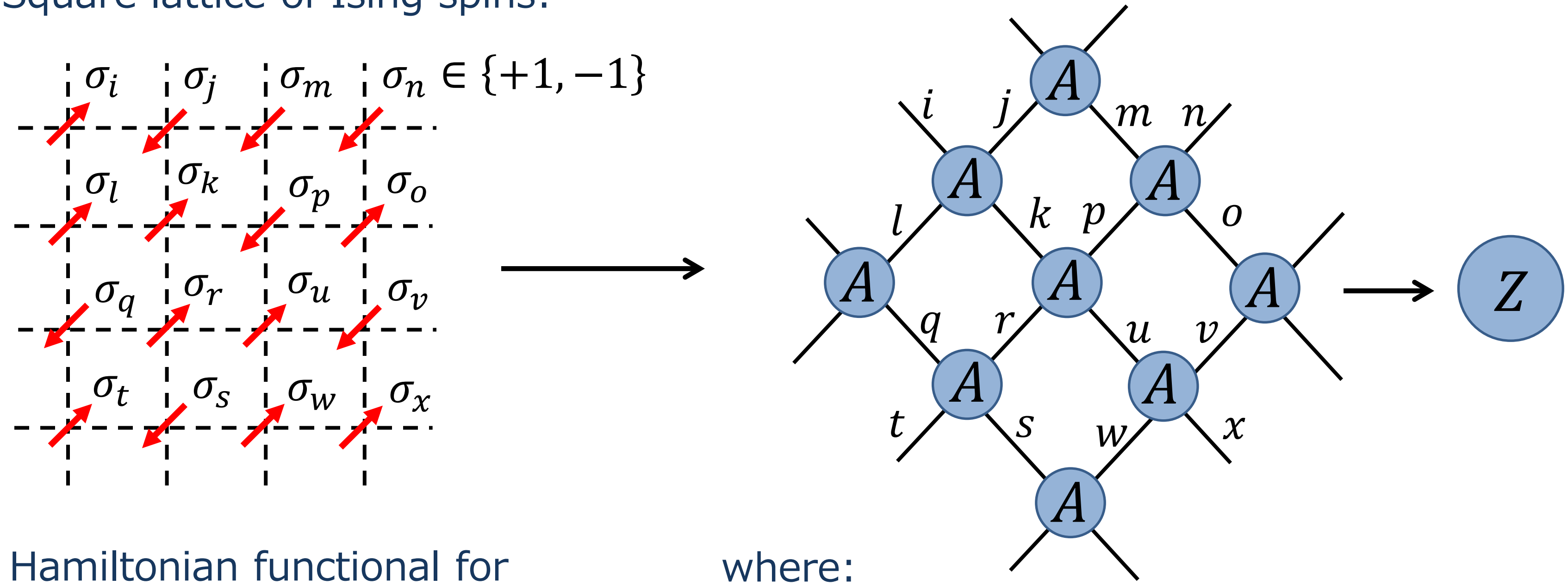
$$Z = \sum_{\{\sigma\}} e^{-H(\{\sigma\})/T}$$

where:

$$A_{ijkl} = e^{(\sigma_i\sigma_j + \sigma_j\sigma_k + \sigma_k\sigma_l + \sigma_l\sigma_i)/T}$$

Partition functions as Tensor Networks

Square lattice of Ising spins:



Hamiltonian functional for Ising ferromagnet:

$$H(\{\sigma\}) = -\sum_{\langle i,j\rangle} \sigma_i \sigma_j$$

Partition function:

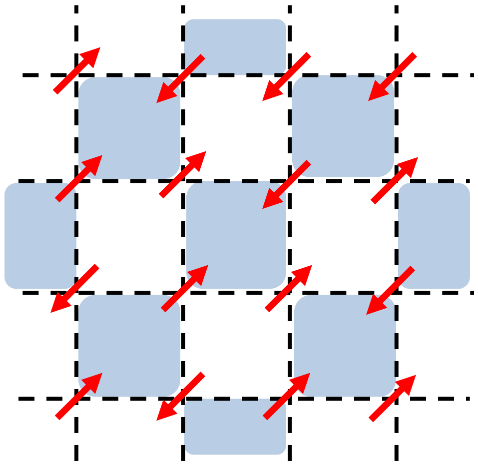
$$Z = \sum_{\{\sigma\}} e^{-H(\{\sigma\})/T}$$

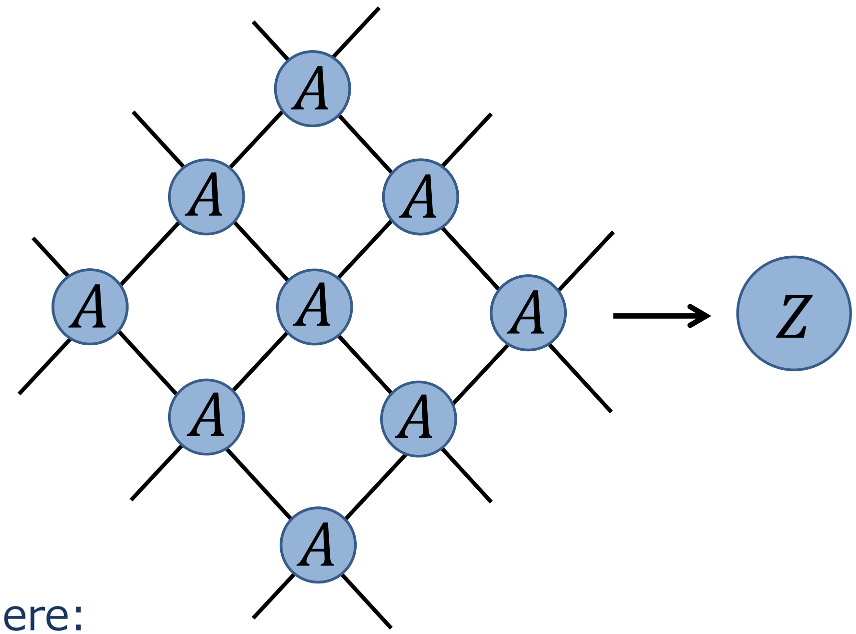
where:

$$A_{ijkl} = e^{(\sigma_i\sigma_j + \sigma_j\sigma_k + \sigma_k\sigma_l + \sigma_l\sigma_i)/T}$$

Partition functions as Tensor Networks

Square lattice of Ising spins:





Hamiltonian functional for Ising ferromagnet:

$$H(\{\sigma\}) = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

where:

$$A_{ijkl} = e^{(\sigma_i\sigma_j+\sigma_j\sigma_k+\sigma_k\sigma_l+\sigma_l\sigma_i)/T}$$

Partition function:

$$Z = \sum_{\{\sigma\}} e^{-H(\{\sigma\})/T} = \mathrm{tTr}\left(\bigotimes_{x=1}^{N} A\right)$$

Partition function given by contraction of tensor network

Path Integrals as Tensor Networks

Nearest neighbour Hamiltonian for a 1D quantum system:

$$H = \sum_r h(r, r+1) = \sum_{r\ \mathrm{even}} h(r, r+1) + \sum_{r\ \mathrm{odd}} h(r, r+1)$$

$$= H_{\mathrm{even}} + H_{\mathrm{odd}}$$

Evolution in imaginary time yields projector onto ground state:

$$|\psi_{\mathrm{GS}}\rangle\langle\psi_{\mathrm{GS}}| = \lim_{\beta\to\infty}\left[e^{-\beta H}\right]$$

Expand in small time steps:

$$\lim_{\beta\to\infty}\left[e^{-\beta H}\right] = e^{-\tau H} e^{-\tau H} e^{-\tau H} e^{-\tau H} \ldots$$

Suzuki-Trotter expansion:

$$e^{-\tau H} = e^{-\tau H_{\mathrm{even}}} e^{-\tau H_{\mathrm{odd}}} + o(\tau^2)$$

Path Integrals as Tensor Networks

Separate Hamiltonian into even and odd terms:

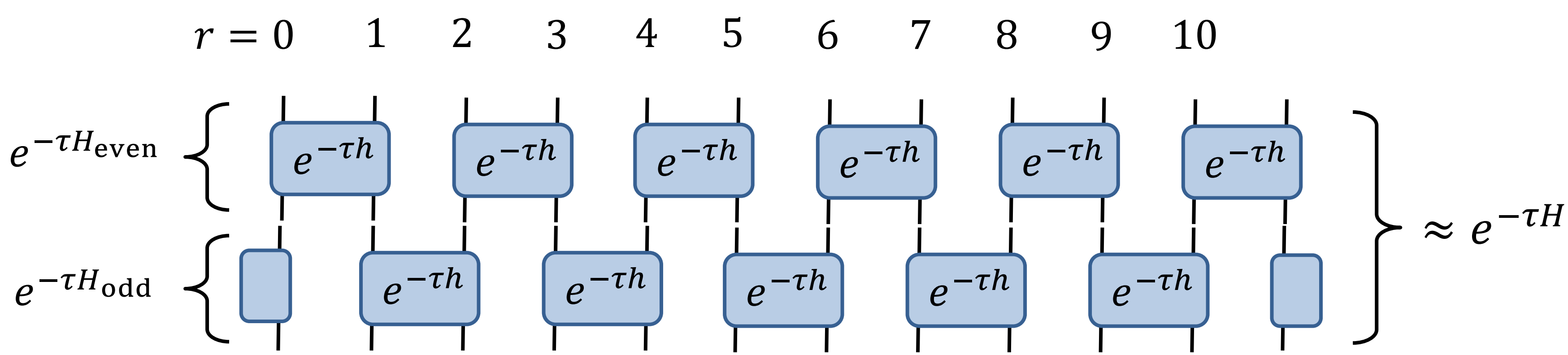
$$H = \sum_{r\ \text{even}} h(r, r+1) + \sum_{r\ \text{odd}} h(r, r+1) \ = H_{\text{even}} + H_{\text{odd}}$$

Expand path integral in small discrete time steps:

$$\lim_{\beta\to\infty} \left[e^{-\beta H}\right] = e^{-\tau H} e^{-\tau H} e^{-\tau H} e^{-\tau H} \dots$$

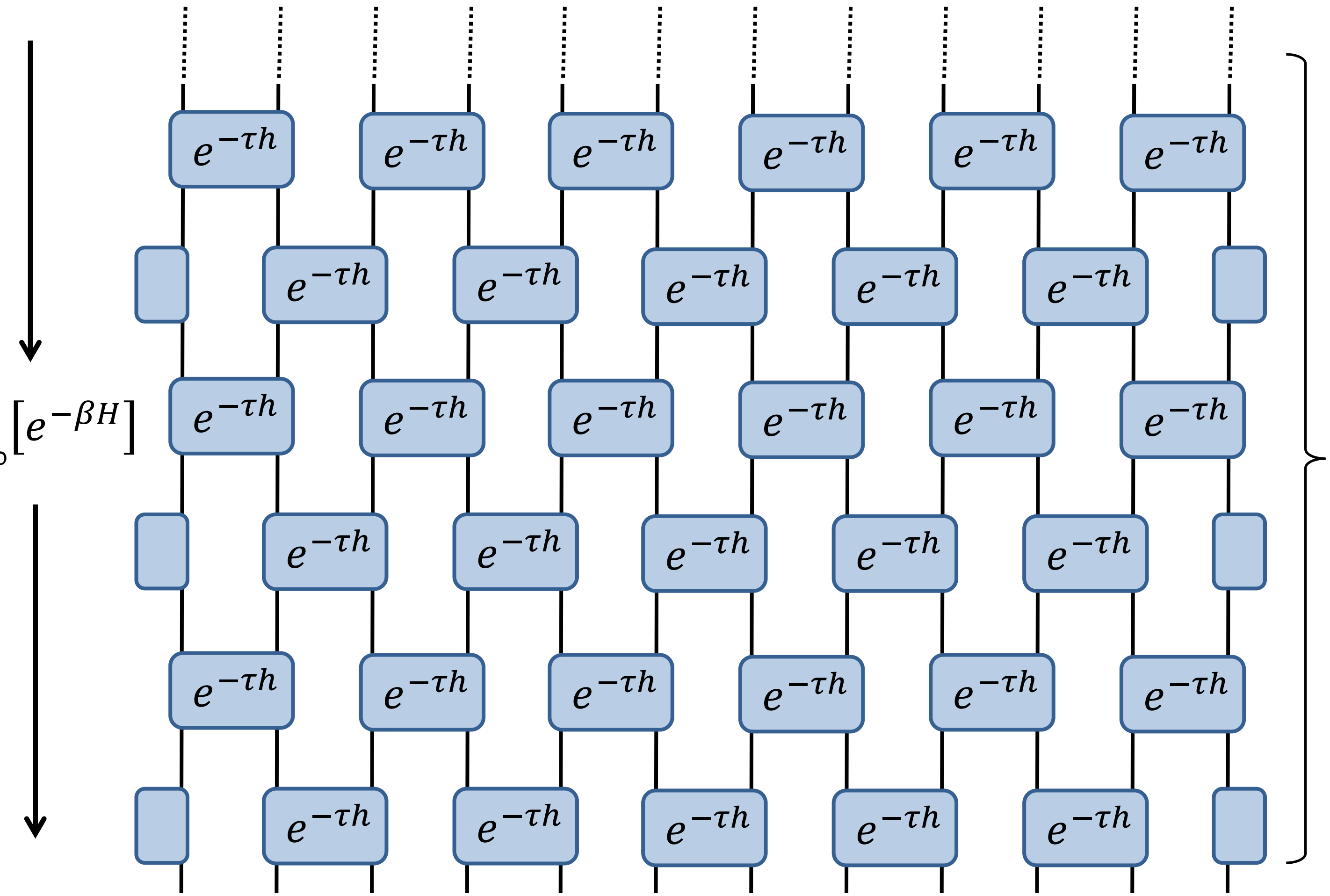
$$e^{-\tau H} = e^{-\tau H_{\text{even}}} e^{-\tau H_{\text{odd}}} + o(\tau^2)$$

Exponentiate even and odd separately :



Path Integrals as Tensor Networks

$$\lim_{\beta\to\infty}\left[e^{-\beta H}\right]$$

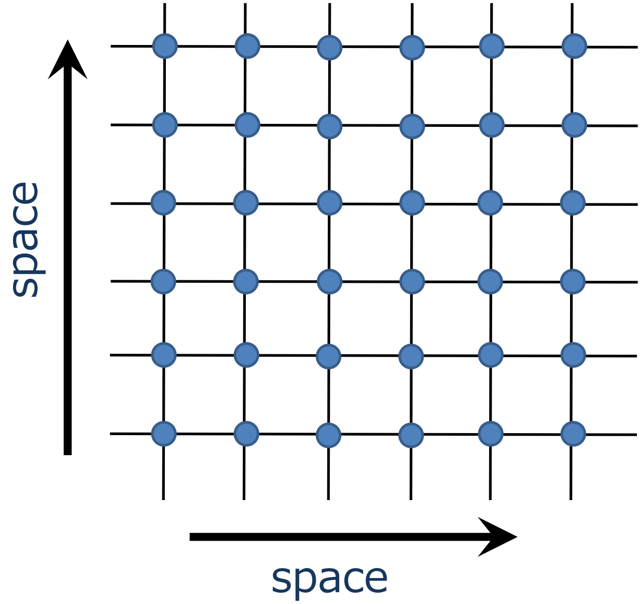


$$|\psi_{\mathrm{GS}}\rangle\langle\psi_{\mathrm{GS}}|$$

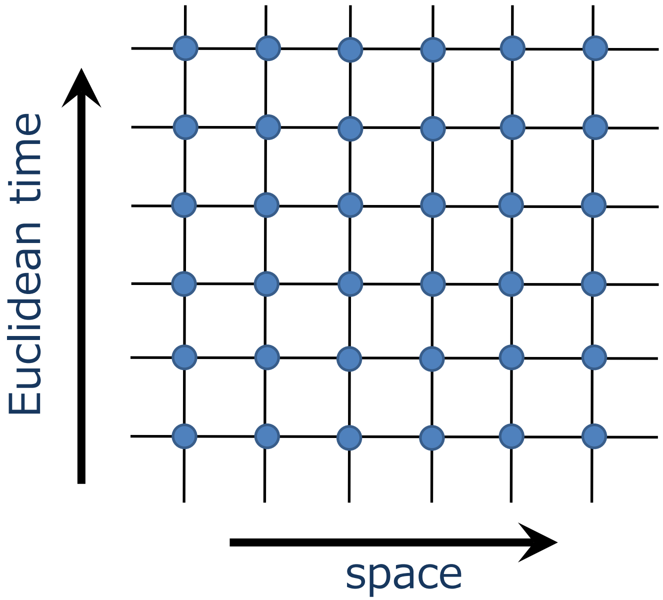
Overview

encode many-body systems as a tensor network:

partition function of 2D classical statistical model



Euclidean path integral of 1D quantum model



- tensors encode Boltzmann weights
- contraction of tensor network equals weighted sum over all microstates

- row of tensors encodes small evolution in imaginary time
- contraction of tensor network equals weighted sum over all trajectories

Outline: Tensor Network Renormalization

The set-up: Representation of partition functions and path integrals as tensor networks

Previous approaches: Levin and Nave's Tensor Renormalization Group (LN-TRG), conceptual and computation problems.

New approach: Tensor network renormalization (TNR): proper removal of all short-ranged degrees of freedom via disentanglers

Benchmark results

Extensions

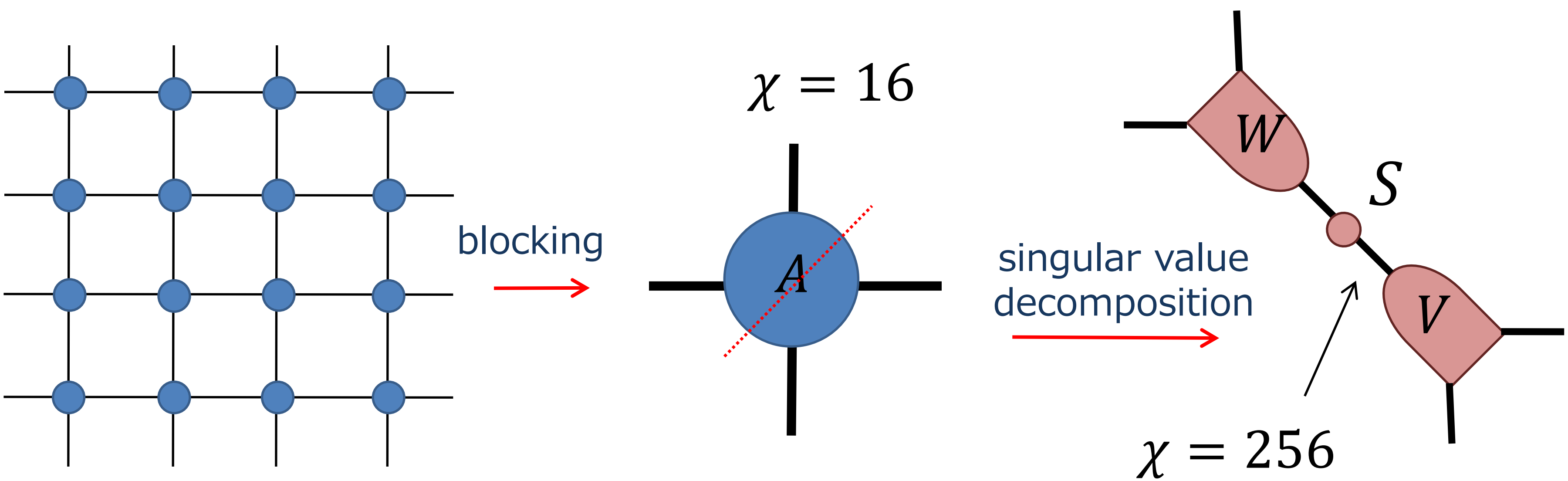
Tensor Renormalization Group (LN-TRG)

Levin, Nave (2006)

Tensor renormalization group (LN-TRG) is a method for coarse-graining tensor networks based upon **blocking** and **truncation steps**

Example of blocking + truncation: 2D classical Ising (critical temp)

- take a (4 x 4) block of tensors from the partition function
- contract to a single tensor; each (16-dim) index describes the state of four classical spins
- can the block tensor be truncated?



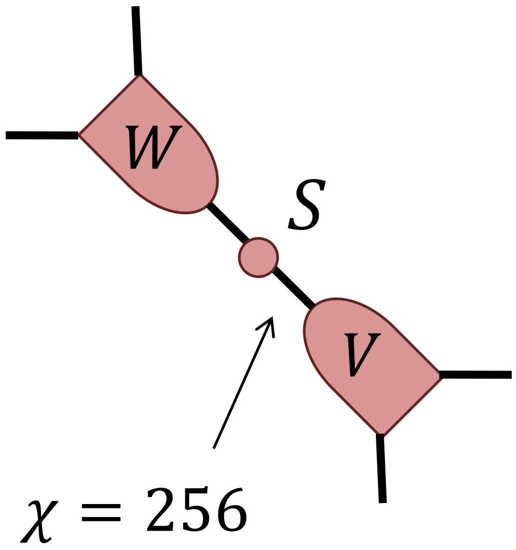
Tensor Renormalization Group (LN-TRG)

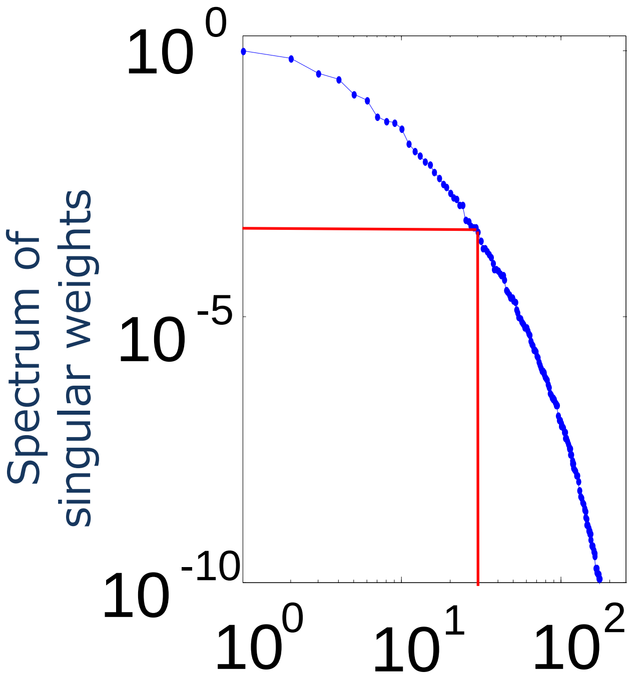
Levin, Nave (2006)

Tensor renormalization group (LN-TRG) is a method for coarse-graining tensor networks based upon **blocking** and **truncation steps**

Example of blocking + truncation: 2D classical Ising (critical temp)

- take a (4 x 4) block of tensors from the partition function
- contract to a single tensor; each (16-dim) index describes the state of four classical spins
- can the block tensor be truncated? **Yes!**

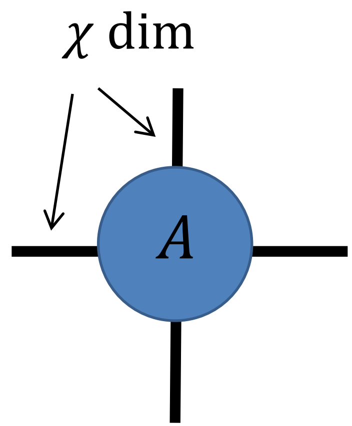


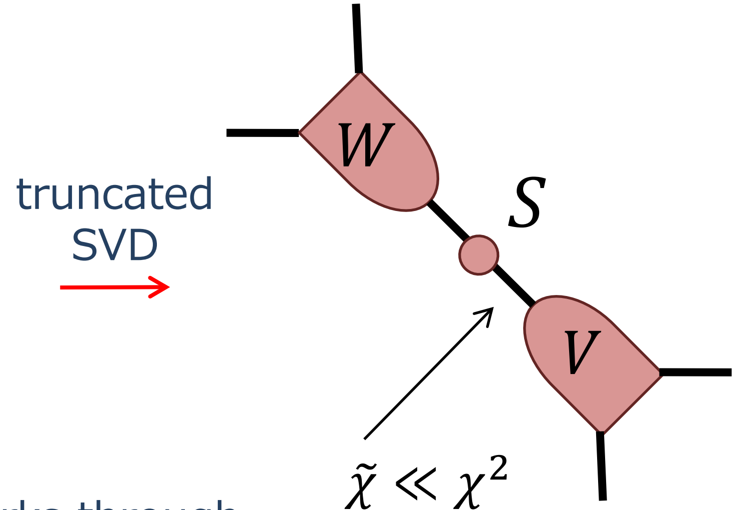


Only keeping the largest 30 singular values yields truncation error ($\sim 10^{-3}$):

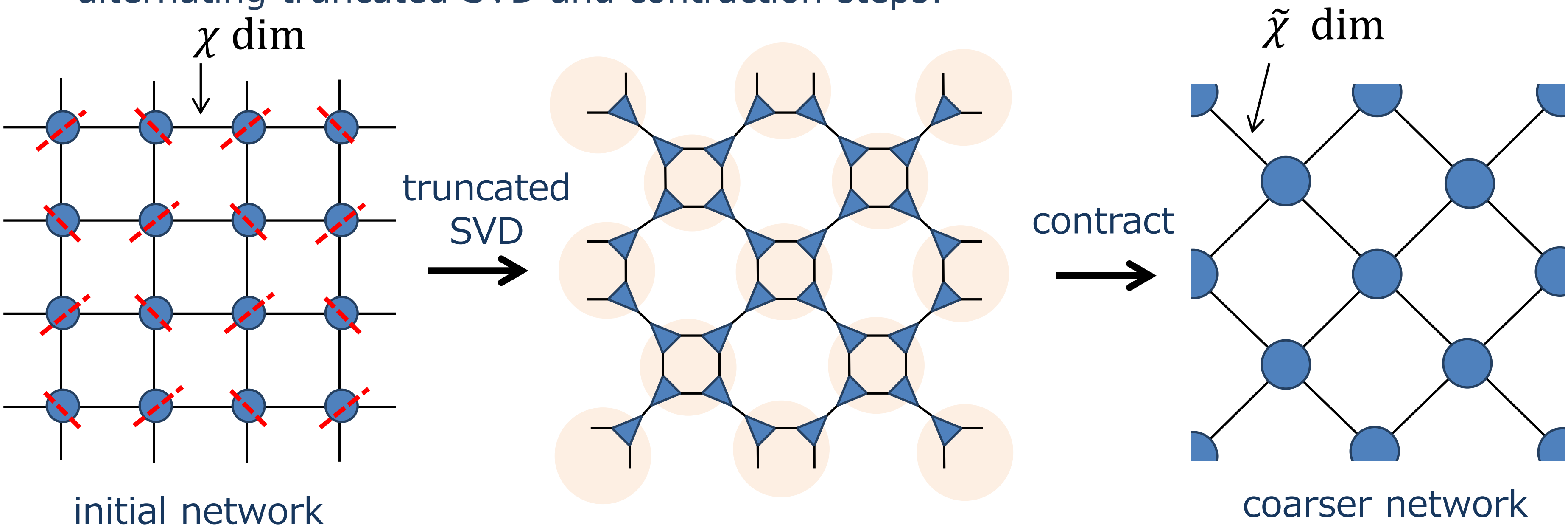
Tensor Renormalization Group (LN-TRG)

discard singular values smaller than desired truncation error δ



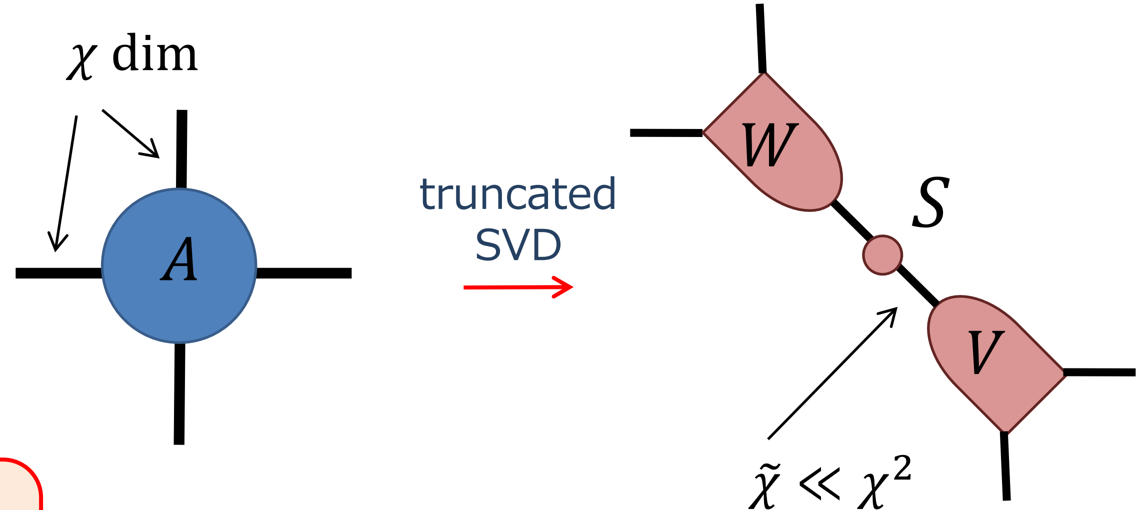


Tensor Renormalization Group (LN-TRG) works through alternating truncated SVD and contraction steps:



Tensor Renormalization Group (LN-TRG)

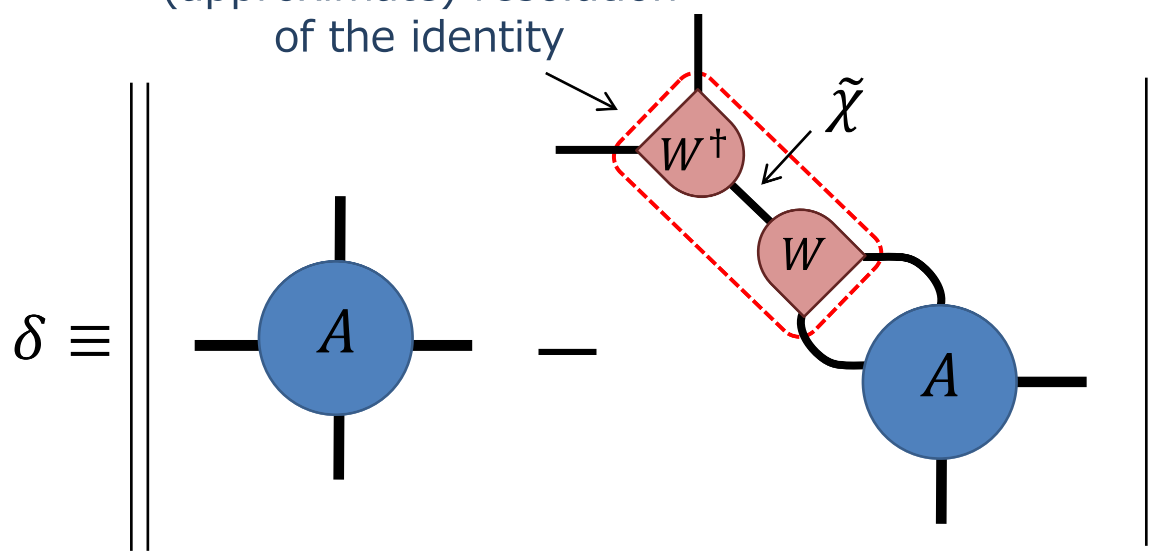
discard singular values smaller than desired truncation error δ



alternative approach:
implement truncation through projector of the form $W^{\dagger}W$ for isometric W

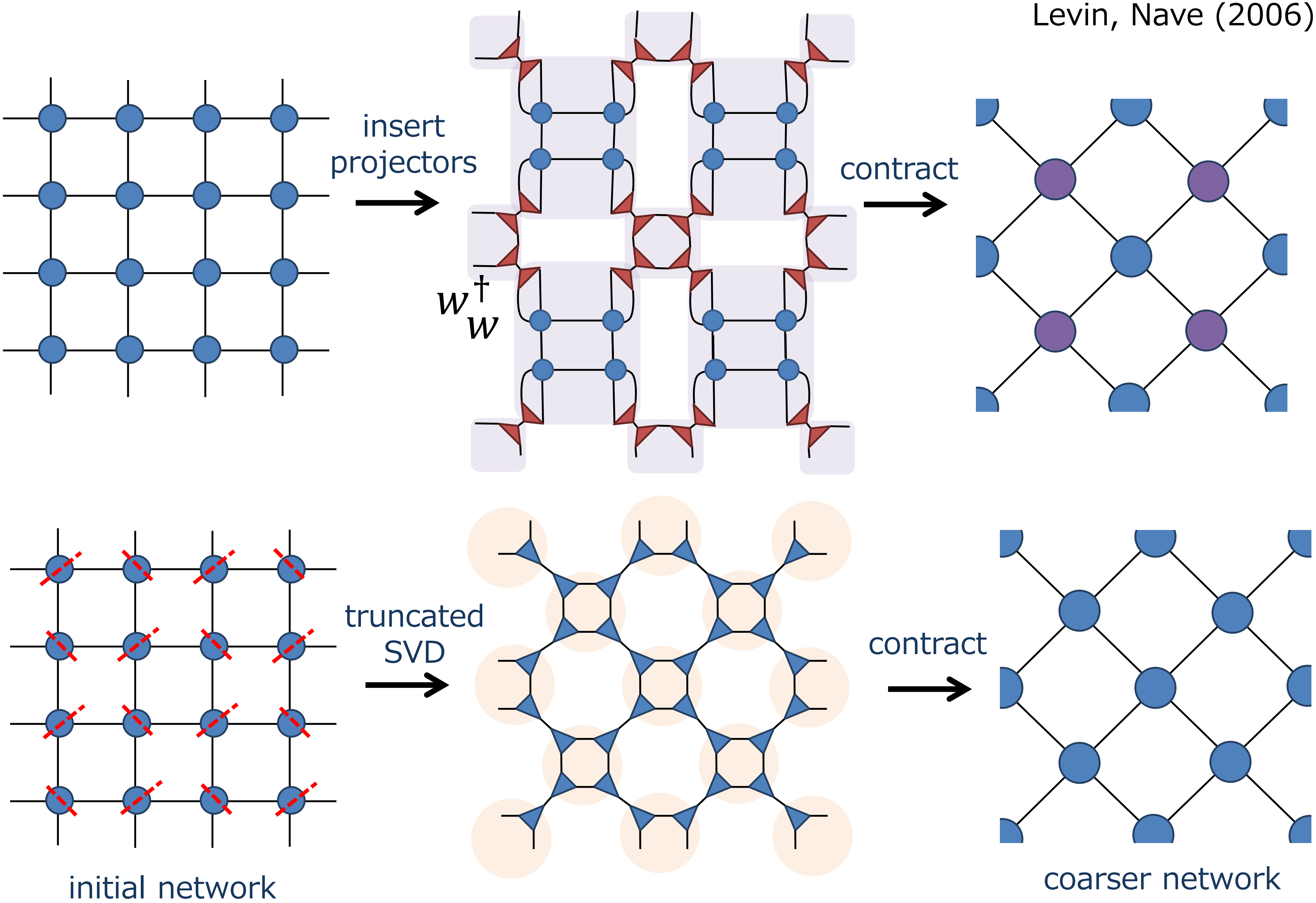
projector acts as a (approximate) resolution of the identity

i.e. choose isometry W to minimise truncation error δ



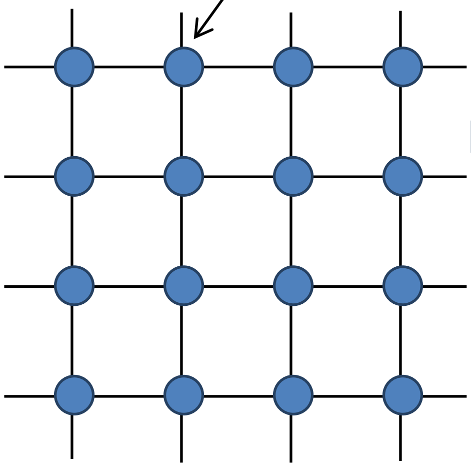
Tensor Renormalization Group (LN-TRG)

Levin, Nave (2006)

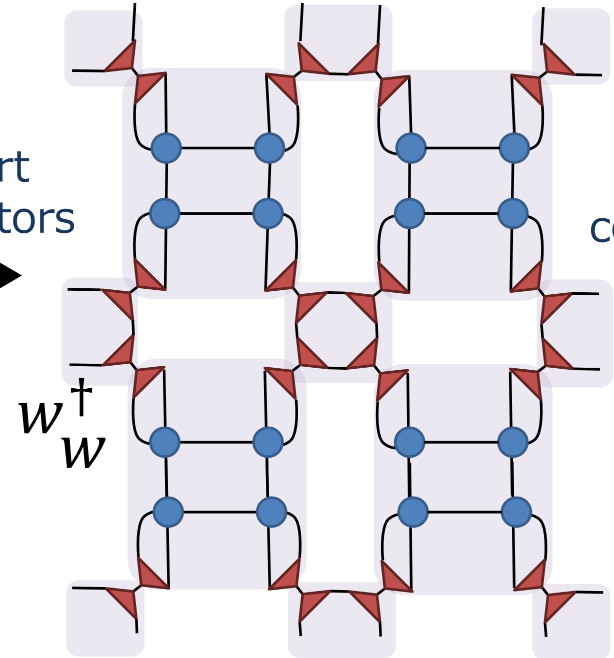


Tensor Renormalization Group (LN-TRG)

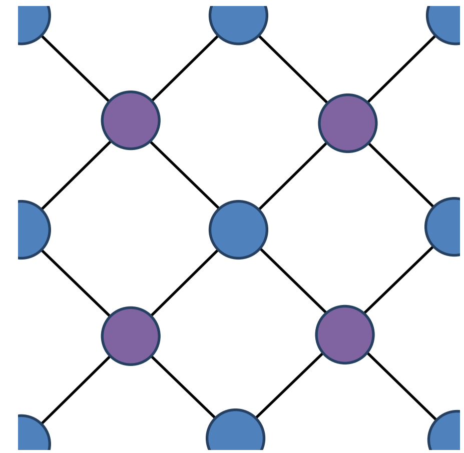
Levin, Nave (2006)



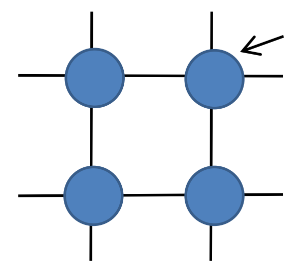
insert projectors $\rightarrow$



contract $\rightarrow$



$\downarrow$ repeat

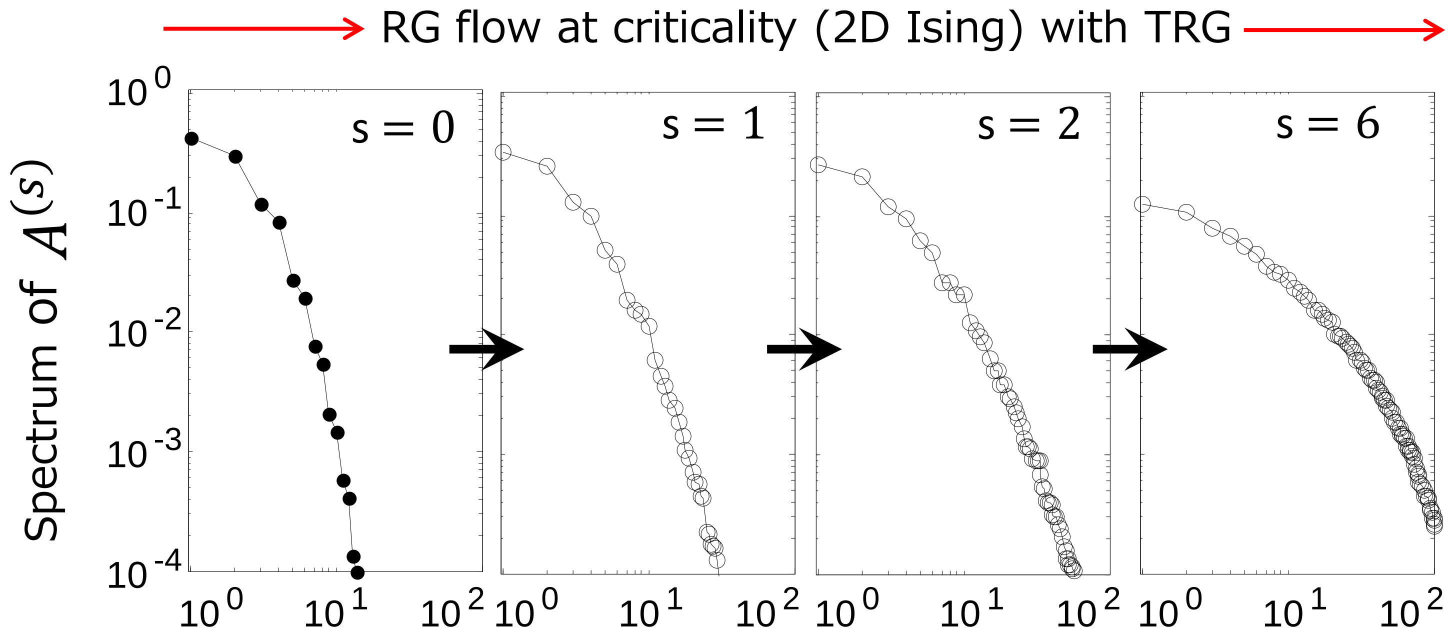


generates RG flow in the space of tensors: $A^{(0)} \rightarrow A^{(1)} \rightarrow \cdots \rightarrow A^{(s)} \rightarrow \cdots$

- is this RG flow sustainable?
- does it converge to the expected fixed points?

Tensor Renormalization Group (LN-TRG)

RG flow in the space of tensors: $A^{(0)} \to A^{(1)} \to A^{(2)} \to \cdots \to A^{(s)} \to \cdots$



Bond dimension χ required for truncation error < 10^{-3}: ~10 → ~20 → ~40 → >100

Cost of iteration: $O(\chi^6)$ $1 \times 10^6 \to 6 \times 10^7 \to 4 \times 10^9 \to > 10^{12}$

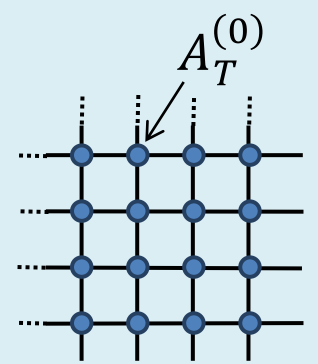
Cost of LN-TRG scales exponentially with RG iteration!

Tensor Renormalization Group (LN-TRG)

RG flow in the space of tensors: $A^{(0)} \to A^{(1)} \to A^{(2)} \to \cdots \to A^{(s)} \to \cdots$

Consider 2D classical Ising ferromagnet at temperature T:

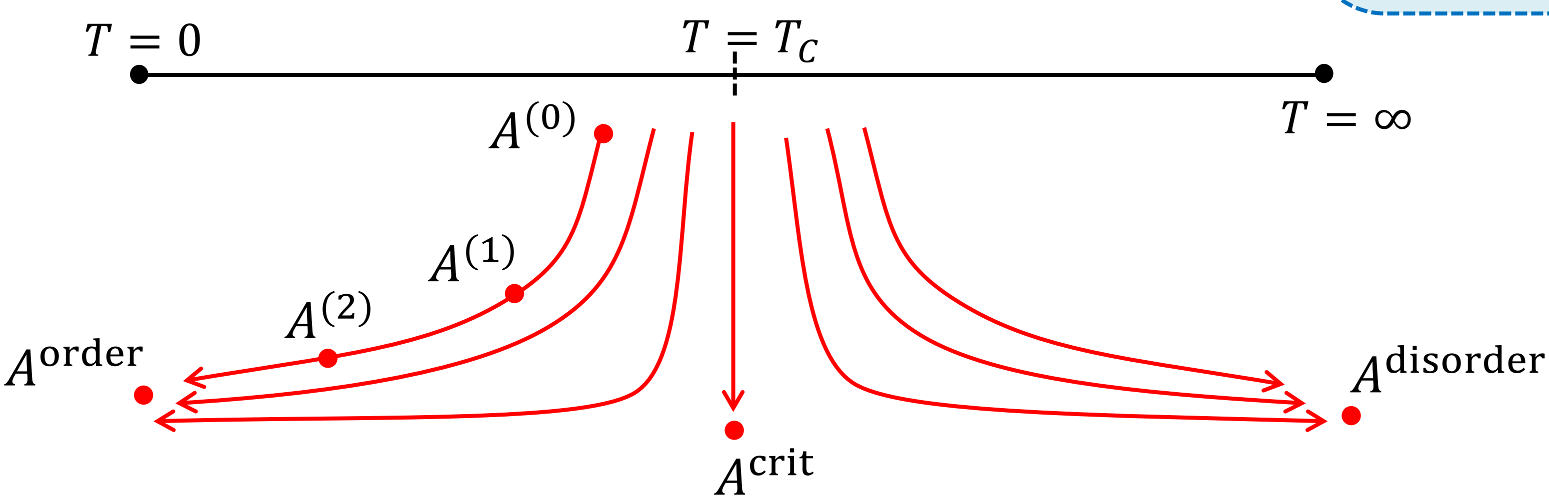
Encode partition function (temp T) as a tensor network:



Phases:

$T < T_C$	ordered phase
$T = T_C$	critical point (correlations at all length scales)
$T > T_C$	disordered phase

Proper RG flow:



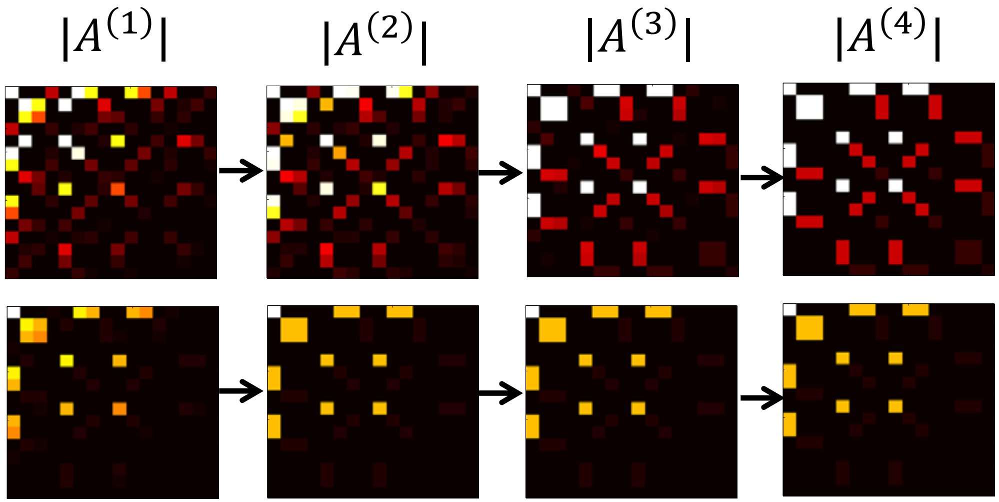
Proper RG flow: 2D classical Ising

Numerical results, Tensor renormalization group **(LN-TRG):**

disordered phase

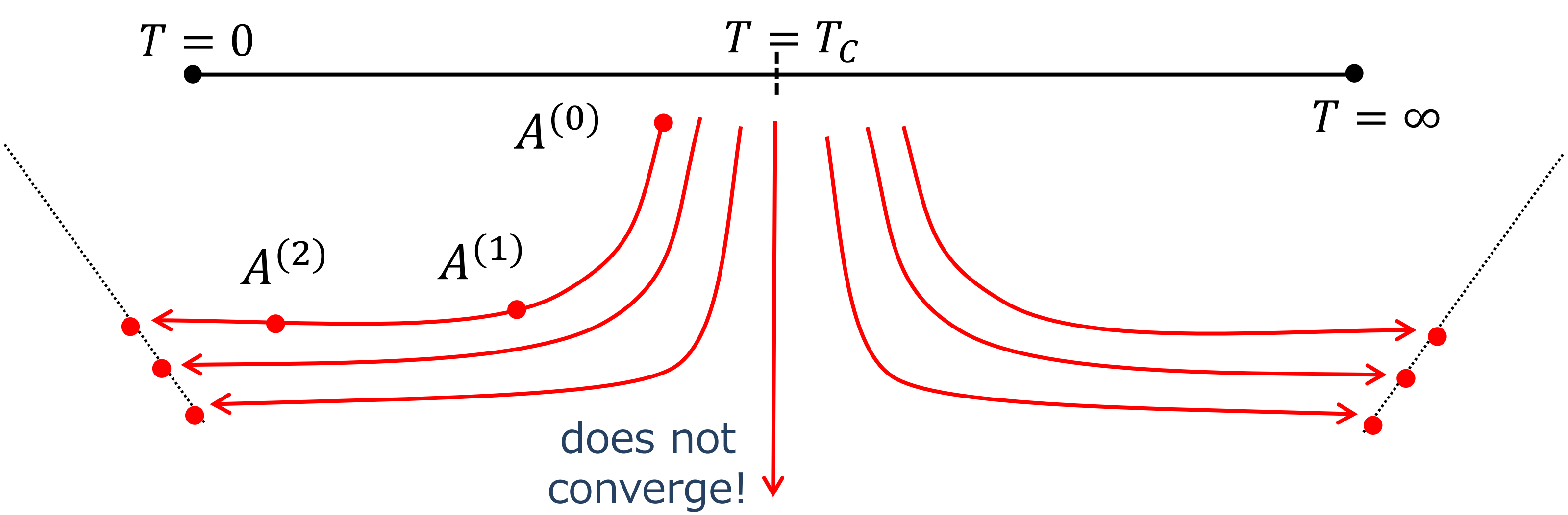
$T = 1.1\,T_c$

$T = 2.0\,T_c$



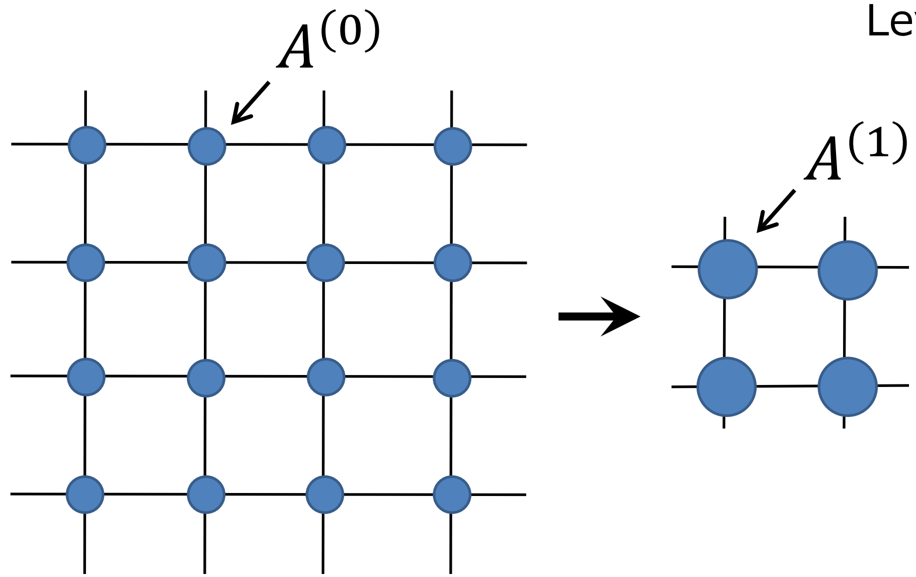
LN-TRG converges to different fixed point tensors

LN-TRG does not give proper RG flow:



Tensor Renormalization Group (LN-TRG)

Levin, Nave (2006)



LN-TRG generates an RG flow in the space of tensors

RG flow in the space of tensors: $A^{(0)} \to A^{(1)} \to A^{(2)} \to \cdots \to A^{(s)} \to \cdots$

LN-TRG can be very powerful and useful numerically but...

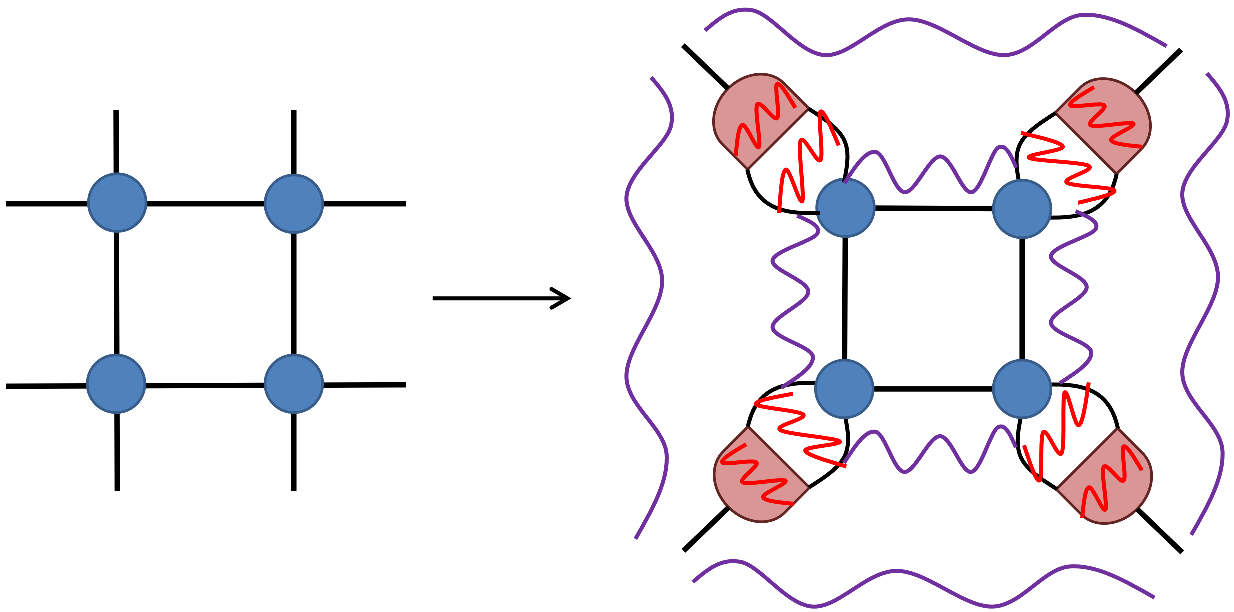
- does not reproduce a proper RG flow
- computational breakdown when near or at criticality

can we understand this?

Tensor Renormalization Group (LN-TRG)

Levin, Nave (2006)

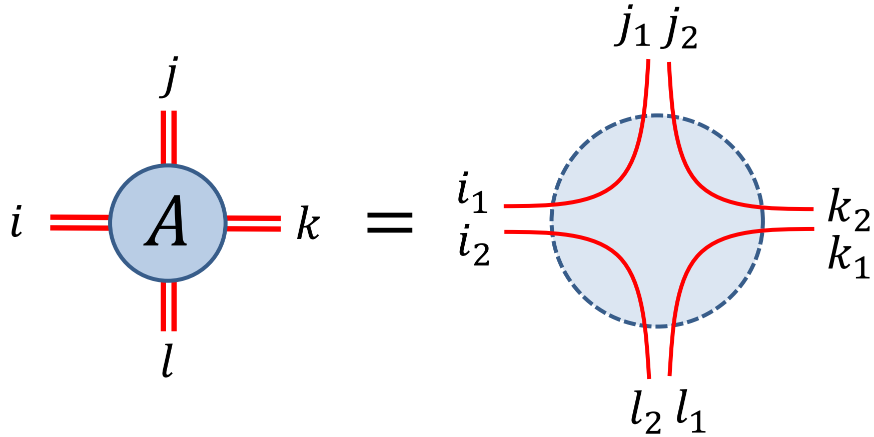
basic step
of LN-TRG:



- isometries remove some (but not all!) short-ranged correlated degrees of freedom
- LN-TRG fails to remove some short-ranged correlations, which propagate to next length scale

Example: corner-double line (CDL) tensors

Fixed points of LN-TRG



Imagine "A" is a special tensor such that each index can be decomposed as a product of smaller indices,

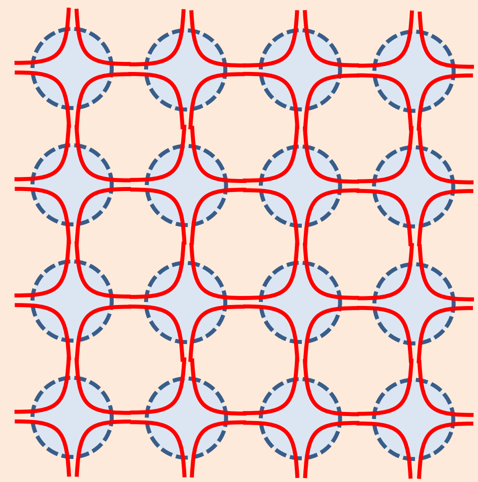
$$A_{ijkl} = A_{(i_1 i_2)(j_1 j_2)(k_1 k_2)(l_1 l_2)}$$

such that certain pairs of indices are perfectly correlated:

$$A_{(i_1 i_2)(j_1 j_2)(k_1 k_2)(l_1 l_2)} \equiv \delta_{i_1 j_1} \delta_{j_2 k_2} \delta_{k_1 l_1} \delta_{l_2 i_2}$$

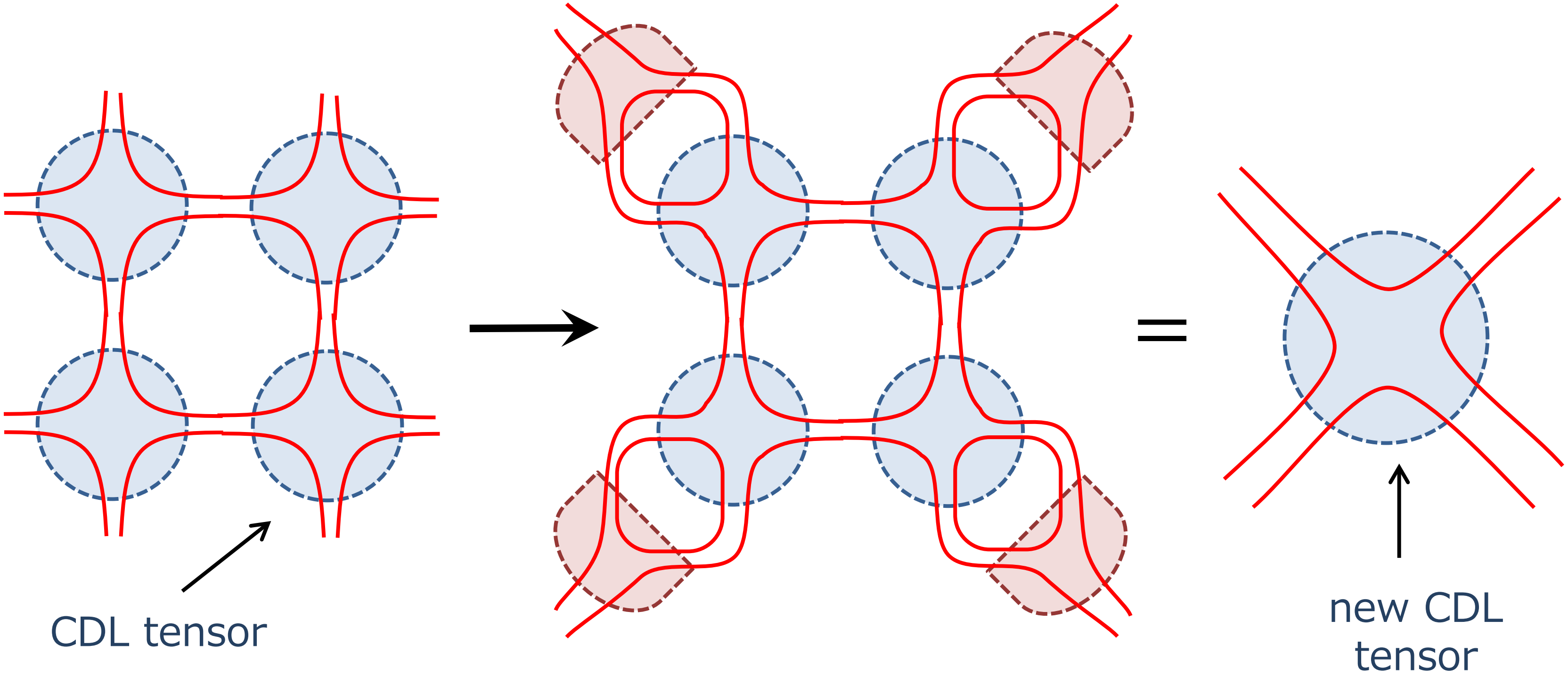
These are called corner double line (CDL) tensors. CDL tensors are fixed points of TRG.

Partition function built from CDL tensors represents a state with short-ranged correlations



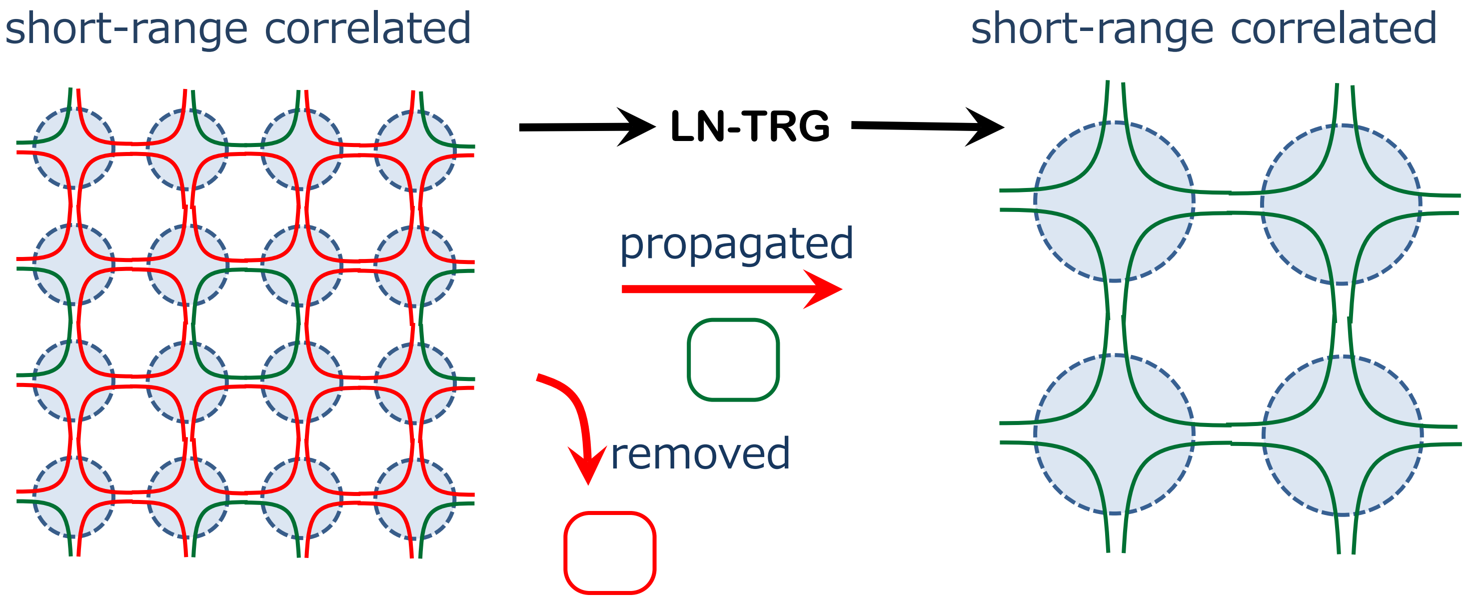
Fixed points of LN-TRG

single iteration of LN-TRG:



Some short-ranged always correlations remain under LN-TRG!

Fixed points of LN-TRG



TRG removes some short ranged correlations, but···

others are artificially promoted to the next length scale

- always retains some of the microscopic (short-ranged) details
- can cause computational breakdown when near criticality

Is there some way to 'fix' tensor renormalization such that all short-ranged correlations are addressed?

Outline: Tensor Network Renormalization

The set-up: Representation of partition functions and path integrals as tensor networks

Previous approaches: Levin and Nave's Tensor Renormalization Group (LN-TRG), conceptual and computation problems.

New approach: Tensor network renormalization (TNR): proper removal of all short-ranged degrees of freedom via disentanglers

Benchmark results

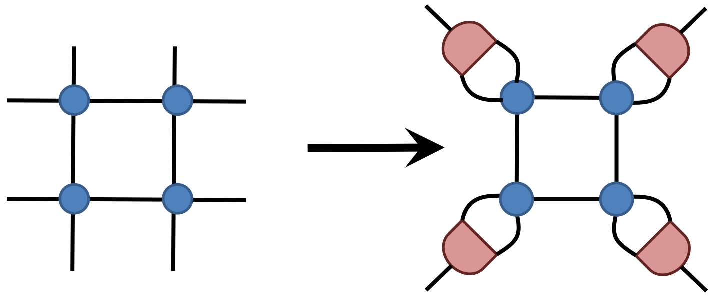
Extensions

Tensor Network Renormalization

arXiv:1412.0732

previous RG schemes for tensor networks based upon **blocking:**

i.e. isometries responsible for combining and truncating indices

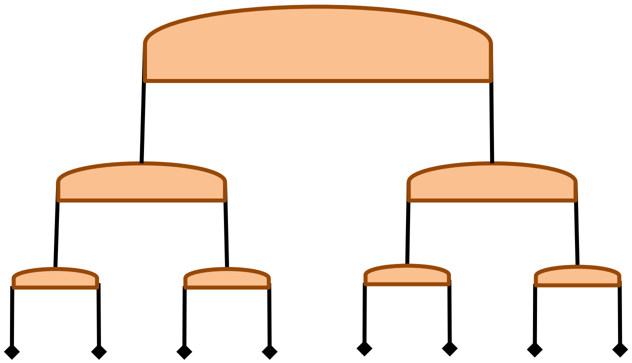


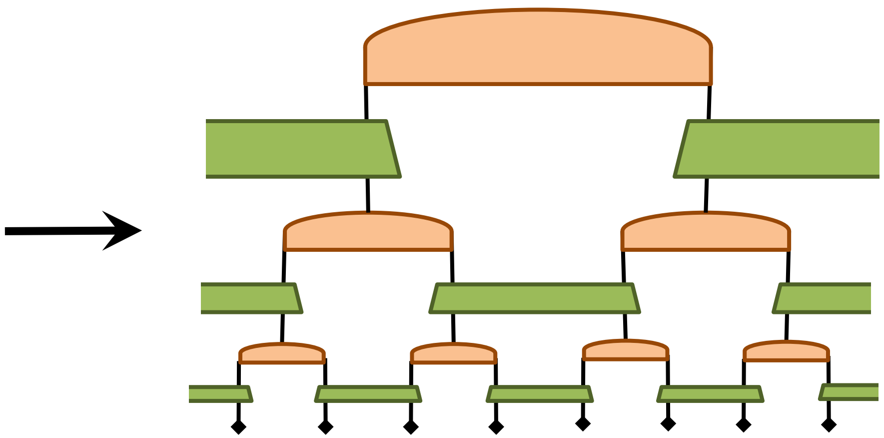
but blocking alone fails to remove short-ranged degrees of freedom...

...can one incorporate some form of unitary disentangling into a tensor RG scheme?

Tree tensor network (TTN)

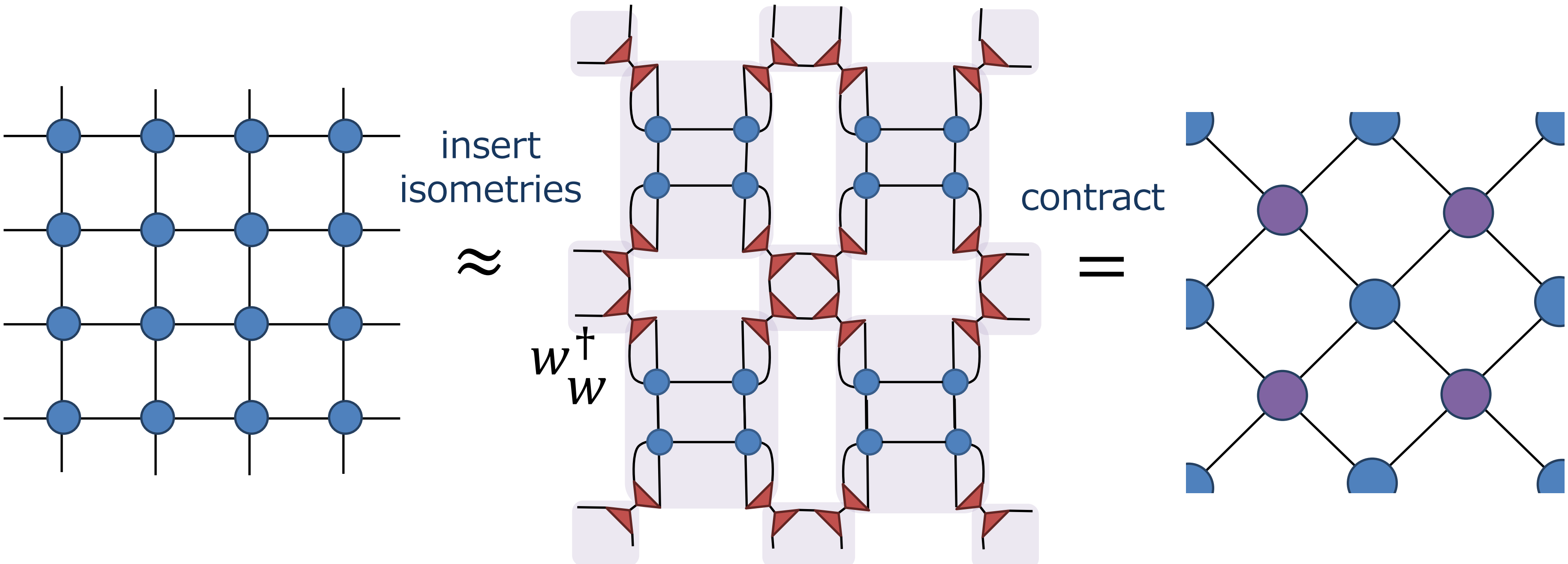
Multi-scale entanglement renormalization ansatz (MERA)





Tensor Network Renormalization

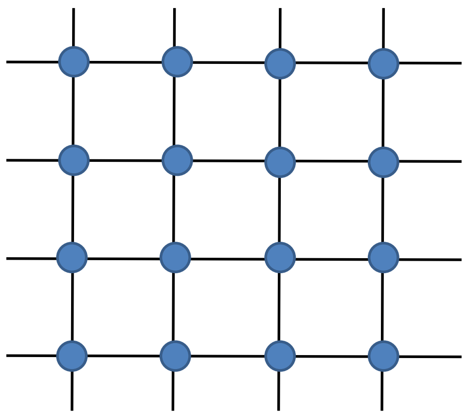
arXiv:1412.0732

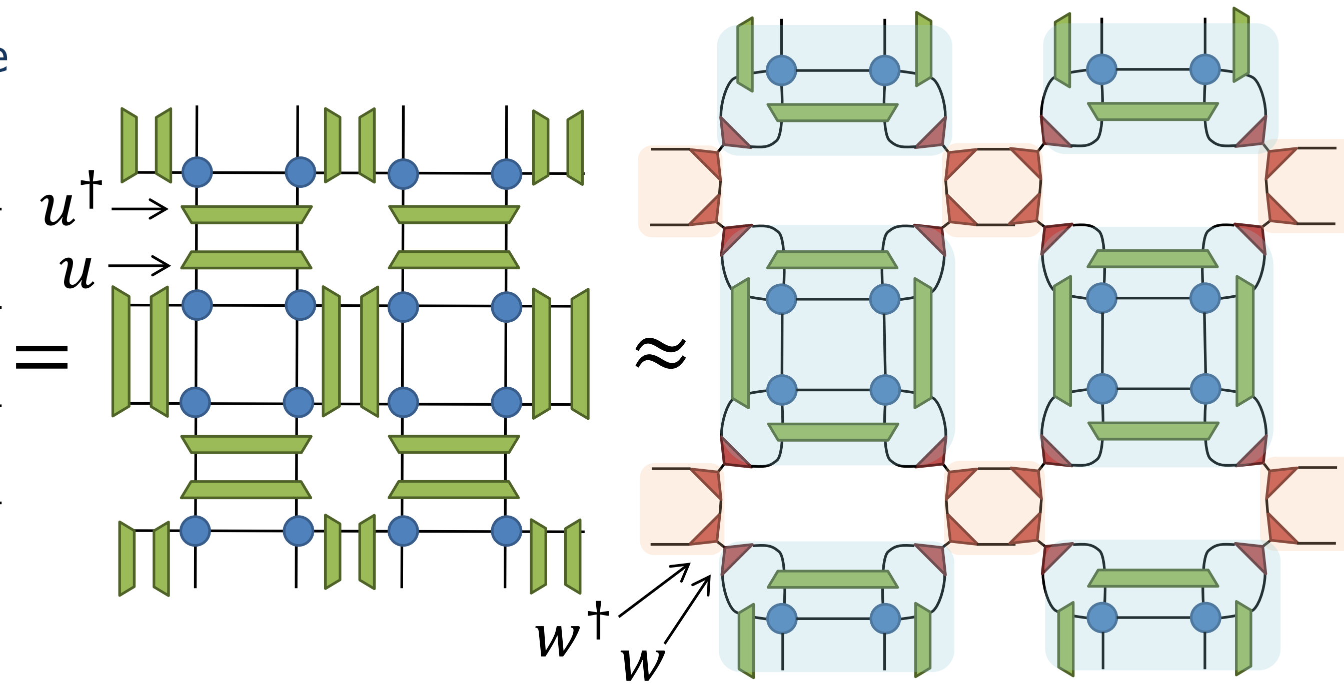


Tensor Network Renormalization

arXiv:1412.0732

initial square lattice tensor network:



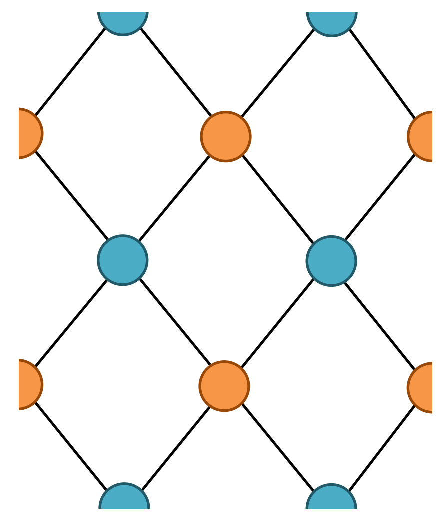


exact step: insert conjugate pairs of unitaries: $u^\dagger u = I$

approximate step: insert conjugate pairs of isometries: $w^\dagger w$

exact step: contract

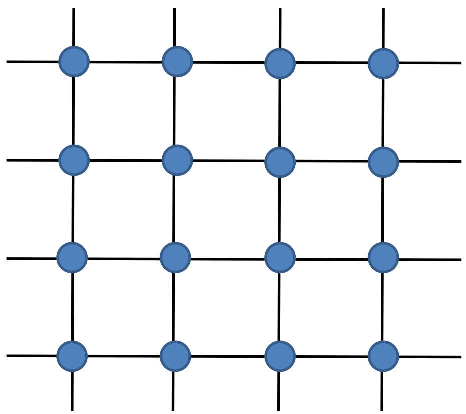
coarser network:

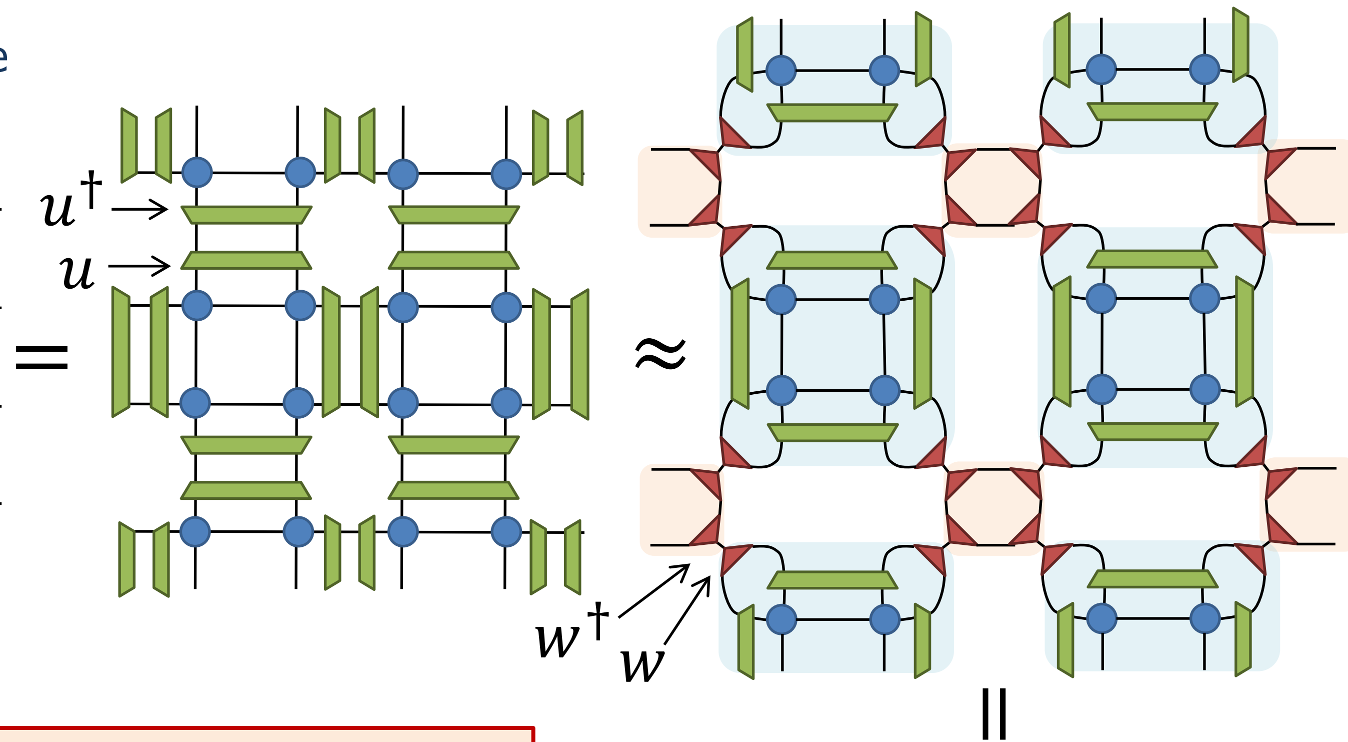


Tensor Network Renormalization

arXiv:1412.0732

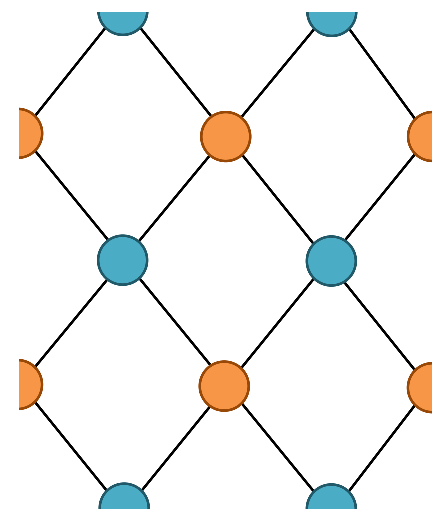
initial square lattice tensor network:





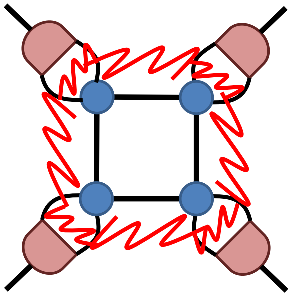
is it possible that the additional **disentangling step** is enough to remove all short-ranged degrees of freedom?

coarser network:



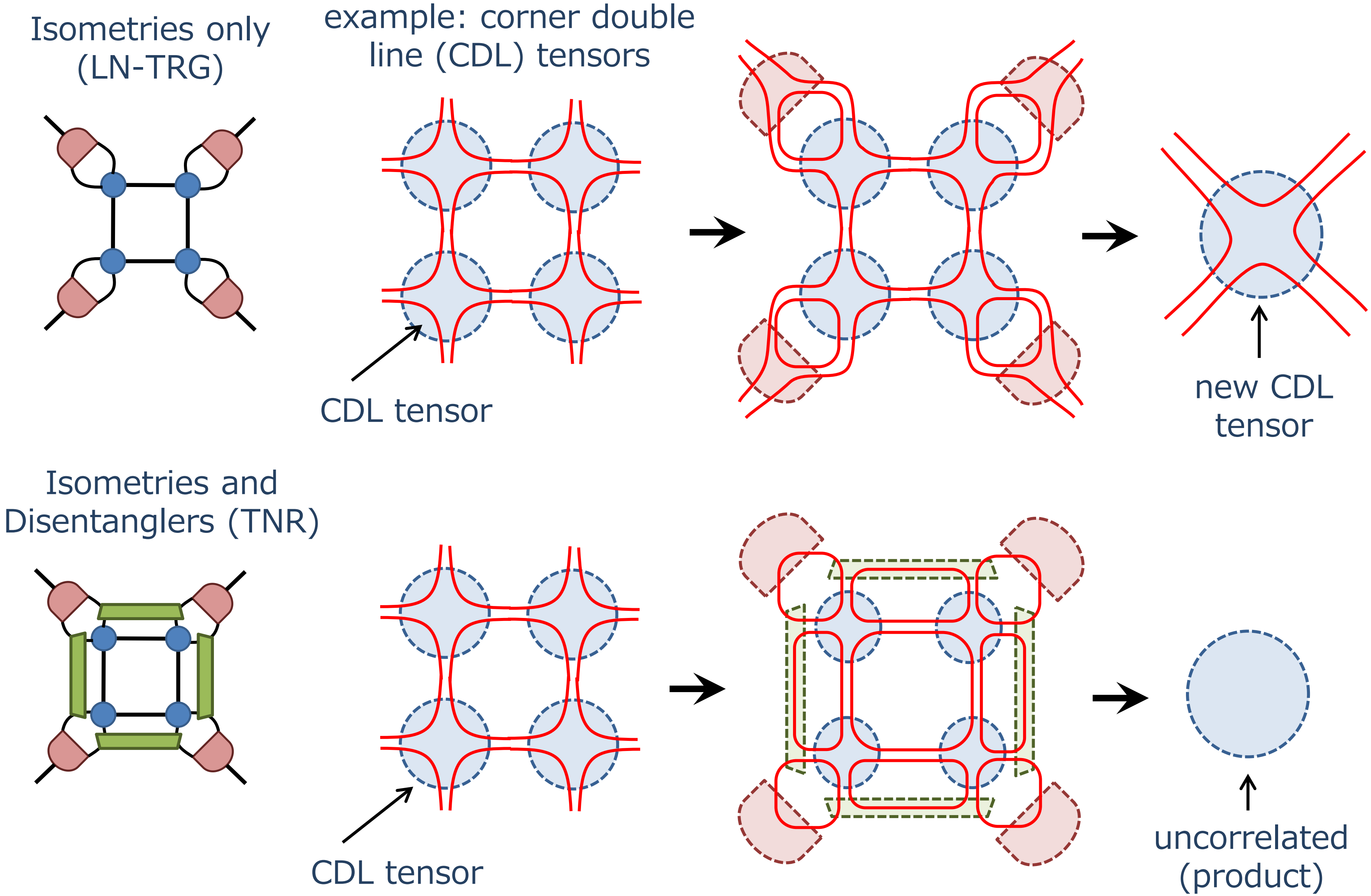
Corner double line tensors revisited

Isometries only
(LN-TRG)



- can remove some short-ranged correlated degrees of freedom
- but fails to remove others

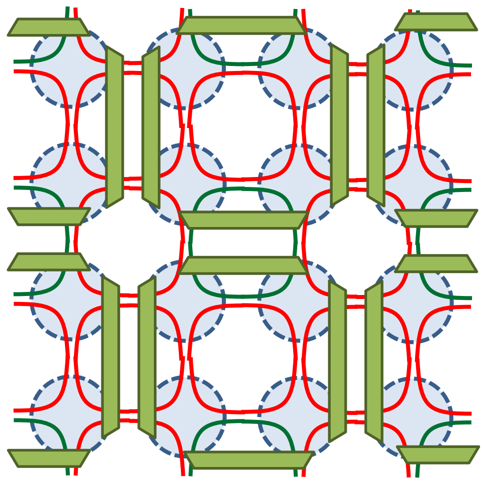
Corner double line tensors revisited



Corner double line tensors revisited

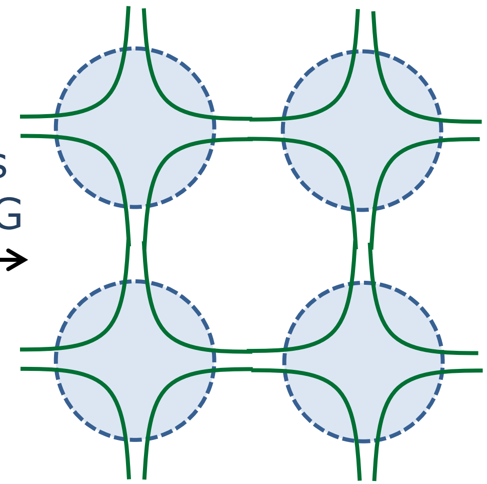
short-range correlated

network of CDL tensors



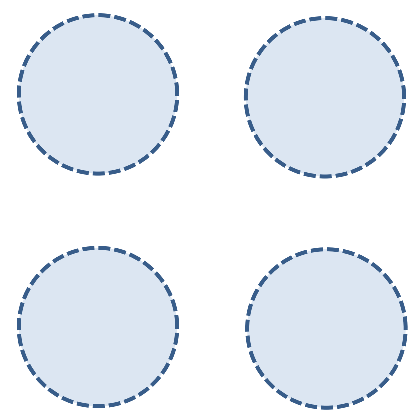
previous tensor RG

short-range correlated



TNR

trivial (product) state

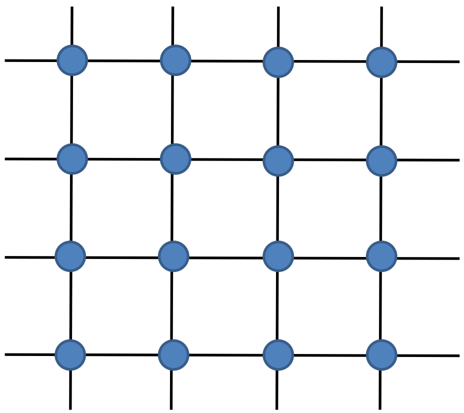


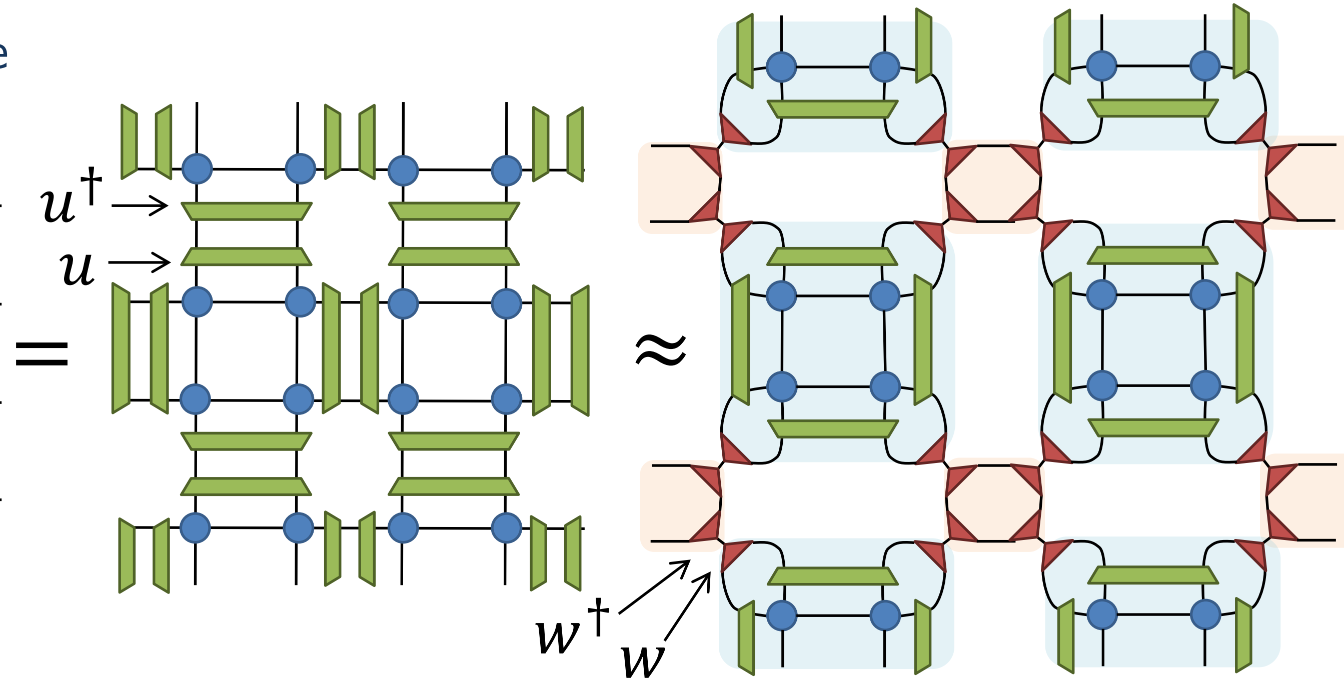
TNR coarse-grains a short-range correlated network into a trivial (product) network as desired!

Tensor Network Renormalization

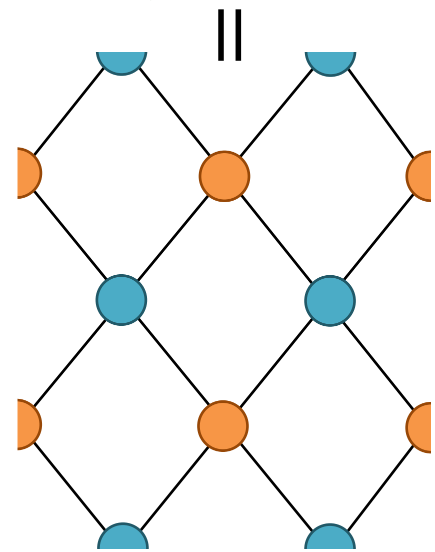
arXiv:1412.0732

initial square lattice tensor network:





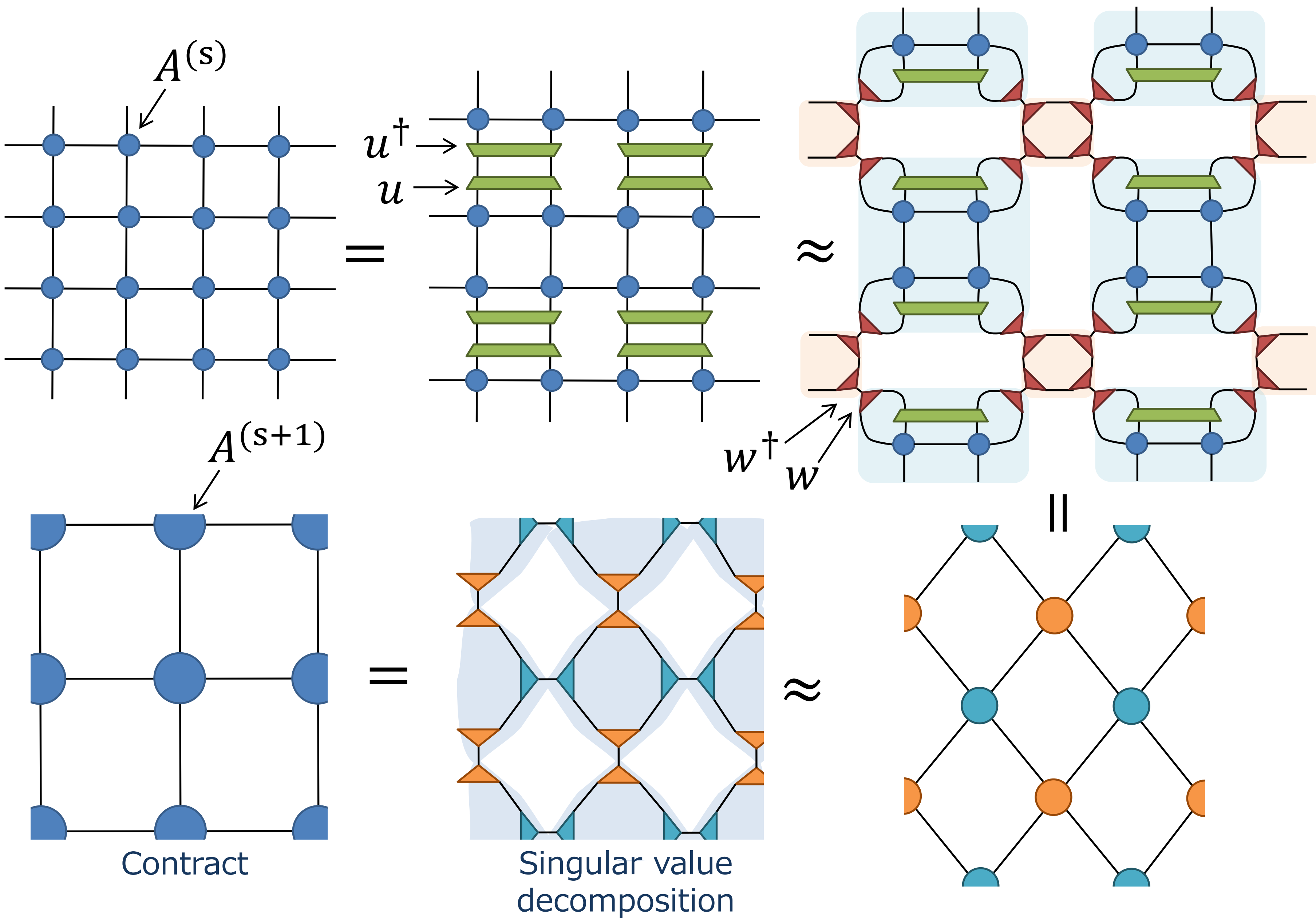
slight modification: we want to include the **minimal amount of disentangling** (sufficient to address all short-range degrees of freedom)



coarser network

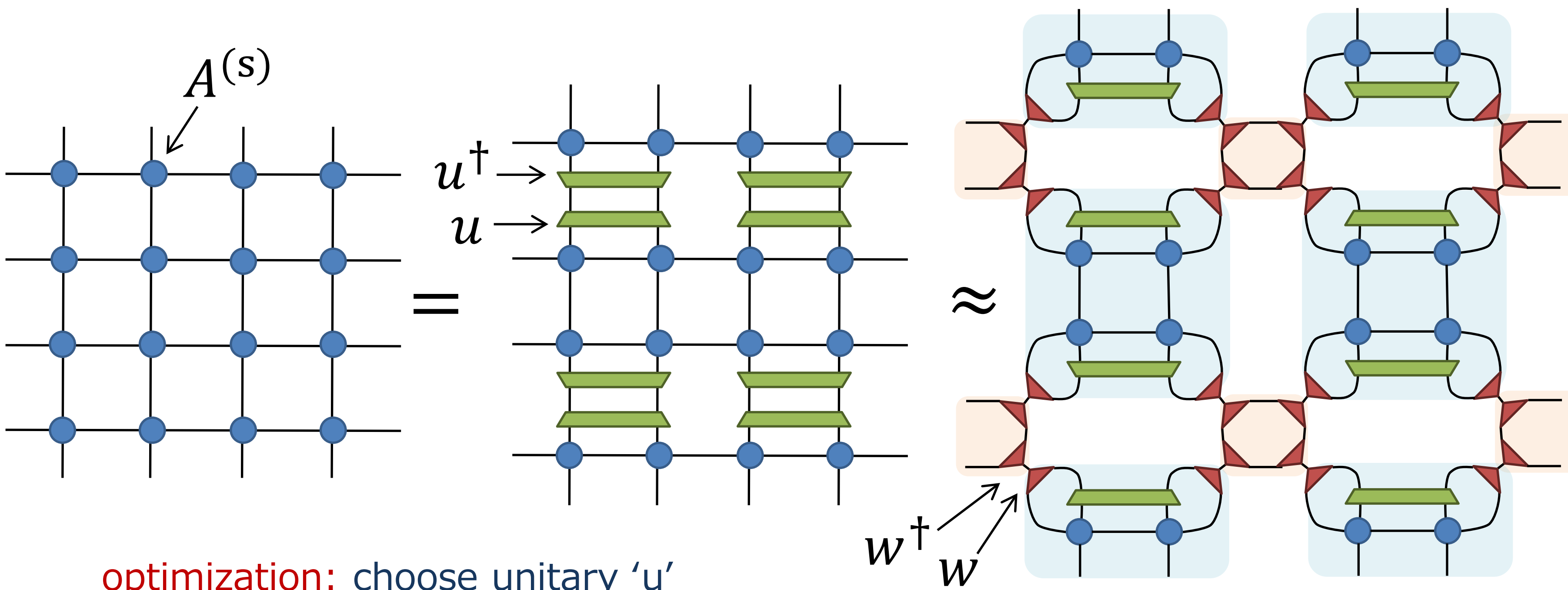
Tensor Network Renormalization

arXiv:1412.0732

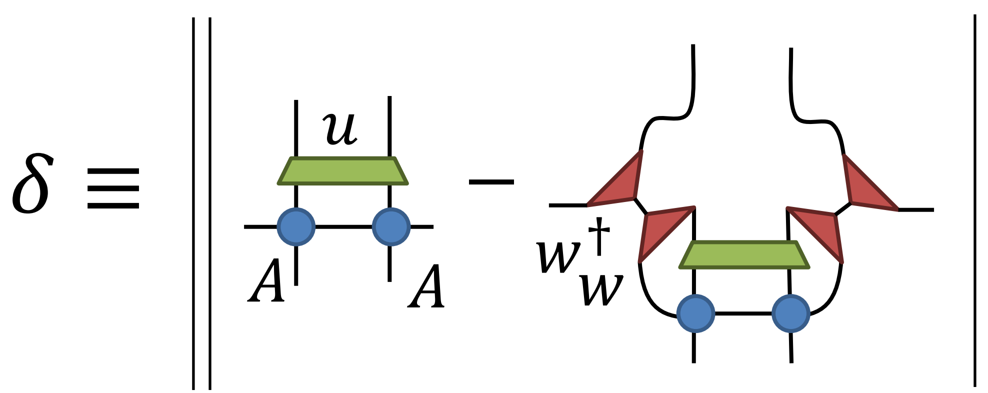


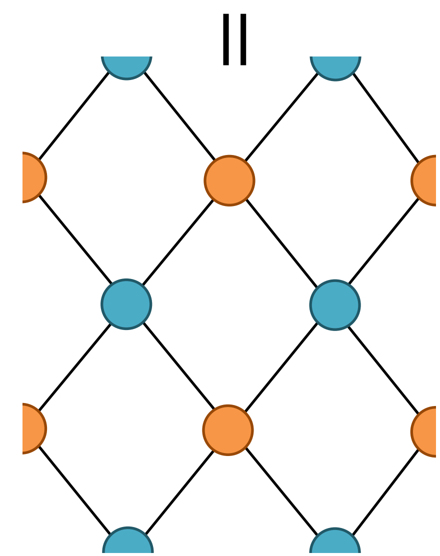
Tensor Network Renormalization

arXiv:1412.0732



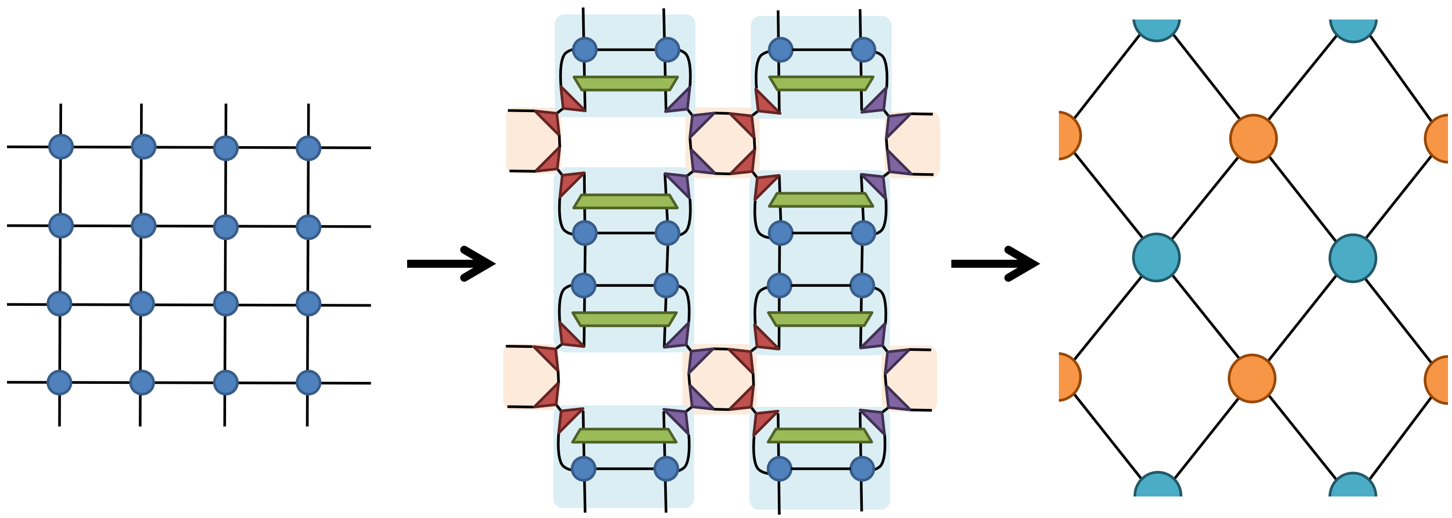
optimization: choose unitary ‘u’ and isometric ‘w’ to minimize the truncation error δ





Tensor Network Renormalization

arXiv:1412.0732



Tensor network renormalization (TNR):
RG for tensor networks designed to address **all short-ranged degrees of freedom** at each step

- works in simple examples (networks with only short-range correlations)
- does it work in more challenging / interesting cases? (such as in critical systems, which possess correlations at all length scales)

Outline: Tensor Network Renormalization

The set-up: Representation of partition functions and path integrals as tensor networks

Previous approaches: Levin and Nave's Tensor Renormalization Group (LN-TRG), conceptual and computation problems.

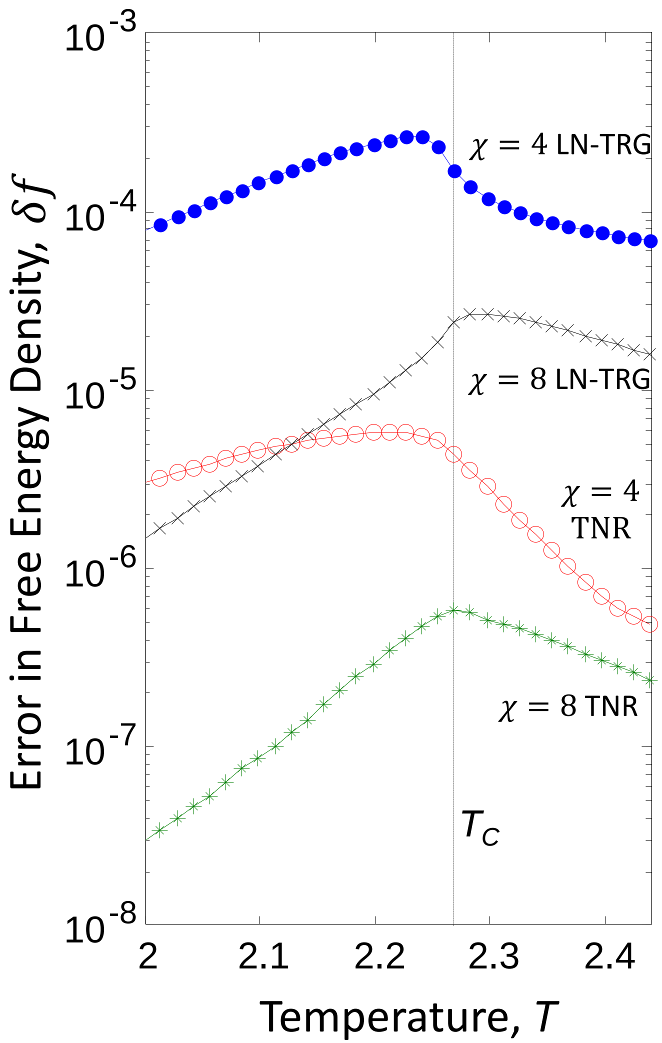
New approach: Tensor network renormalization (TNR): proper removal of all short-ranged degrees of freedom via disentanglers

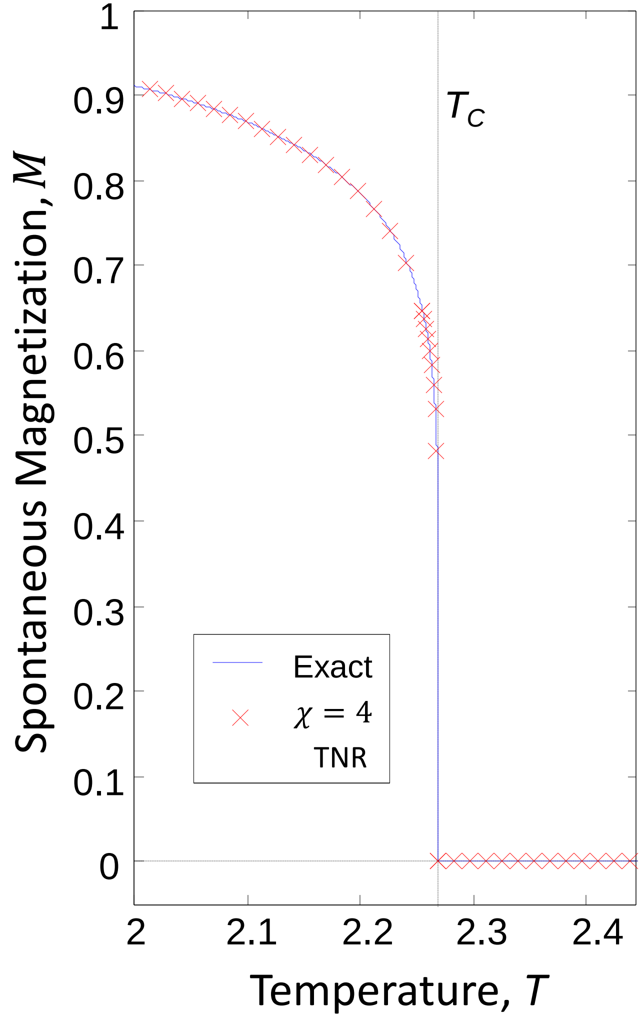
Benchmark results

Extensions

Benchmark numerics:

2D classical Ising model on lattice of size: $2^{12} \times 2^{12}$

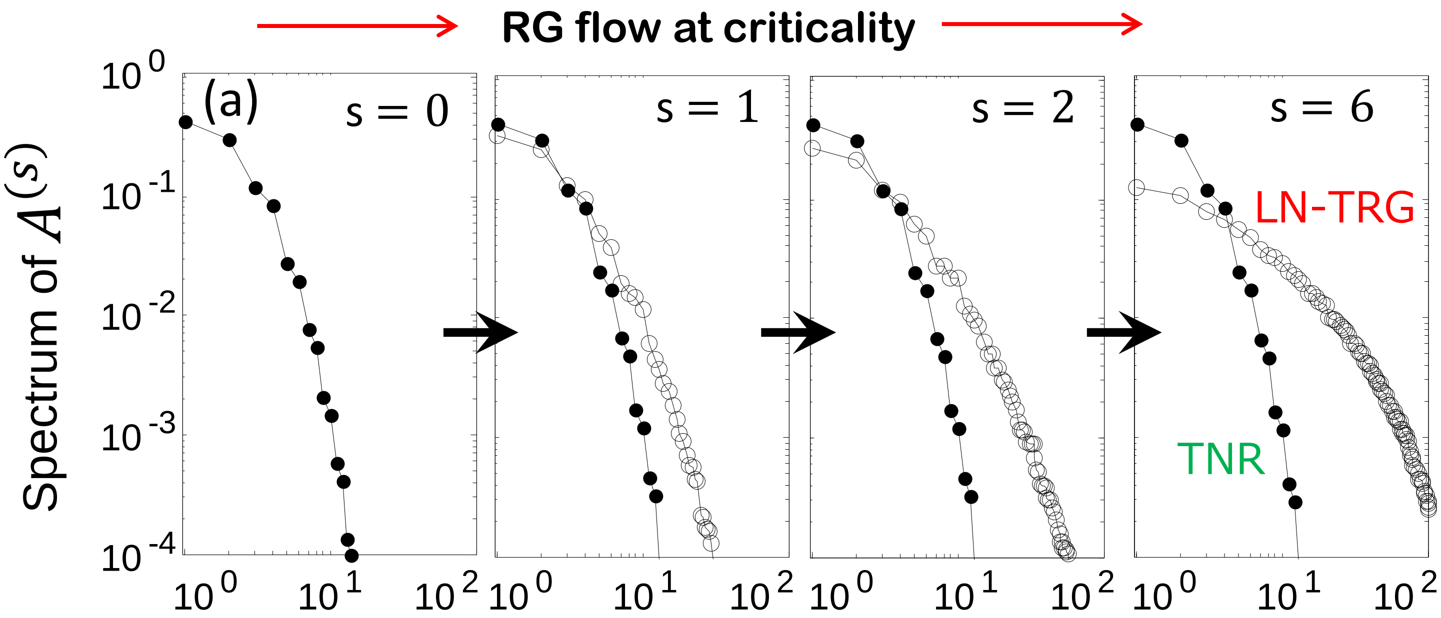




Sustainable RG flow

Old approach (LN-TRG) vs new approach (TNR)

Does TRG give a sustainable RG flow?



Bond dimension χ required to maintain fixed truncation error (~10^{-3}):

LN-TRG:	**~10** →	**~20** →	**~40** →	**>100**
TNR:	**~10** →	**~10** →	**~10** →	**~10**

Computational costs:

LN-TRG, $O(\chi^6)$:	1×10^6 →	6×10^7 →	4×10^9 →	$> 10^{12}$
TNR $O(k\chi^6)$:	5×10^7 →	5×10^7 →	5×10^7 →	5×10^7

exponential

↓

constant

Tensor Renormalization Group (LN-TRG)

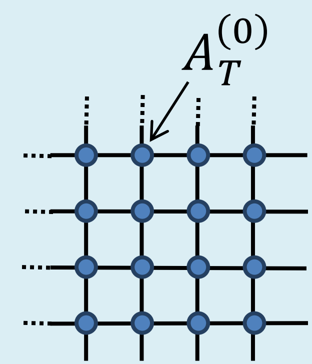
RG flow in the space of tensors: $A^{(0)} \rightarrow A^{(1)} \rightarrow A^{(2)} \rightarrow \cdots \rightarrow A^{(s)} \rightarrow \cdots$

Consider 2D classical Ising ferromagnet at temperature T:

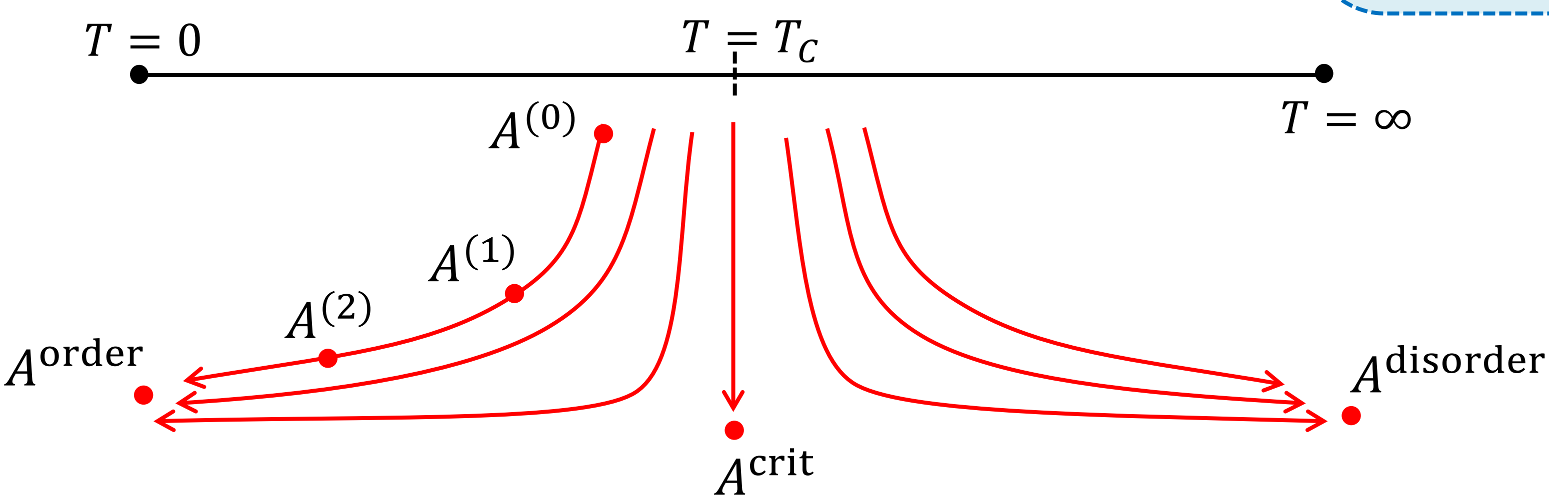
Phases:

$T < T_C$	ordered phase
$T = T_C$	critical point (correlations at all length scales)
$T > T_C$	disordered phase

Encode partition function (temp T) as a tensor network:



Proper RG flow:



Proper RG flow: 2D classical Ising

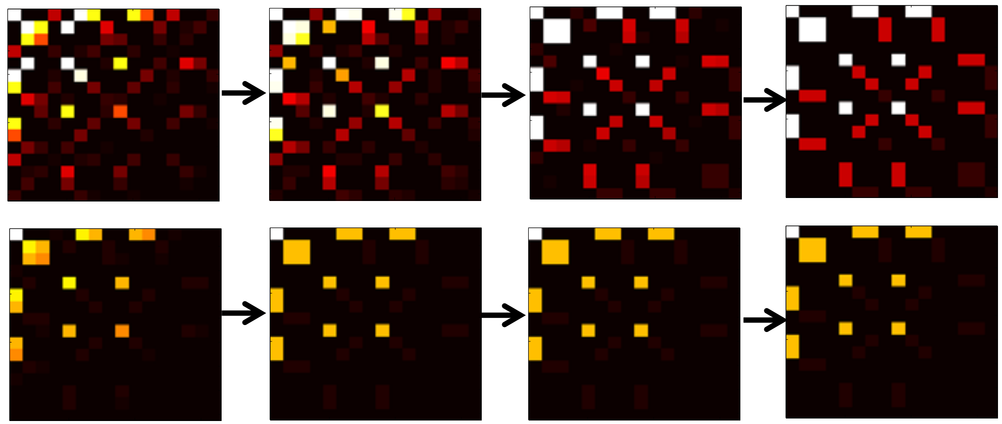
Old Approach: Tensor renormalization group **(LN-TRG):**

disordered phase

$|A^{(1)}|$ $|A^{(2)}|$ $|A^{(3)}|$ $|A^{(4)}|$

$T = 1.1\, T_c$

$T = 2.0\, T_c$



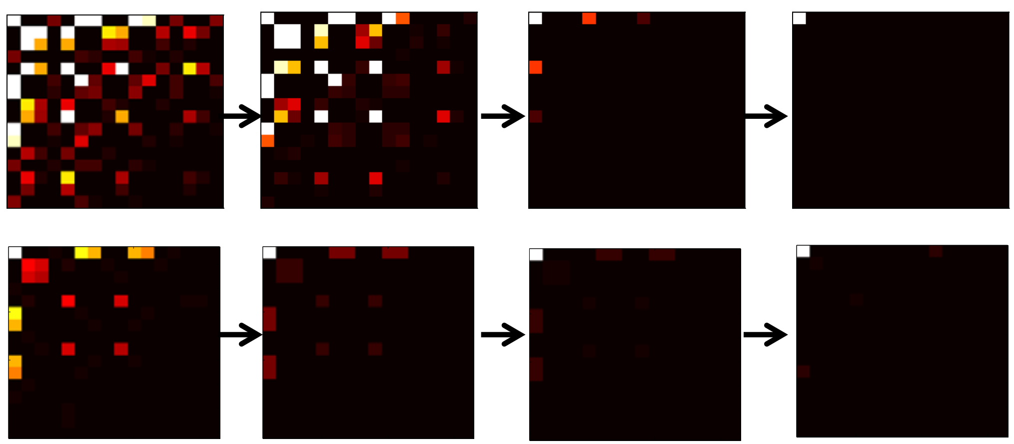
CDL tensor fixed points

still contain microscopic "baggage"

New Approach: Tensor Network Renormalization **(TNR):**

$T = 1.1\, T_c$

$T = 2.0\, T_c$

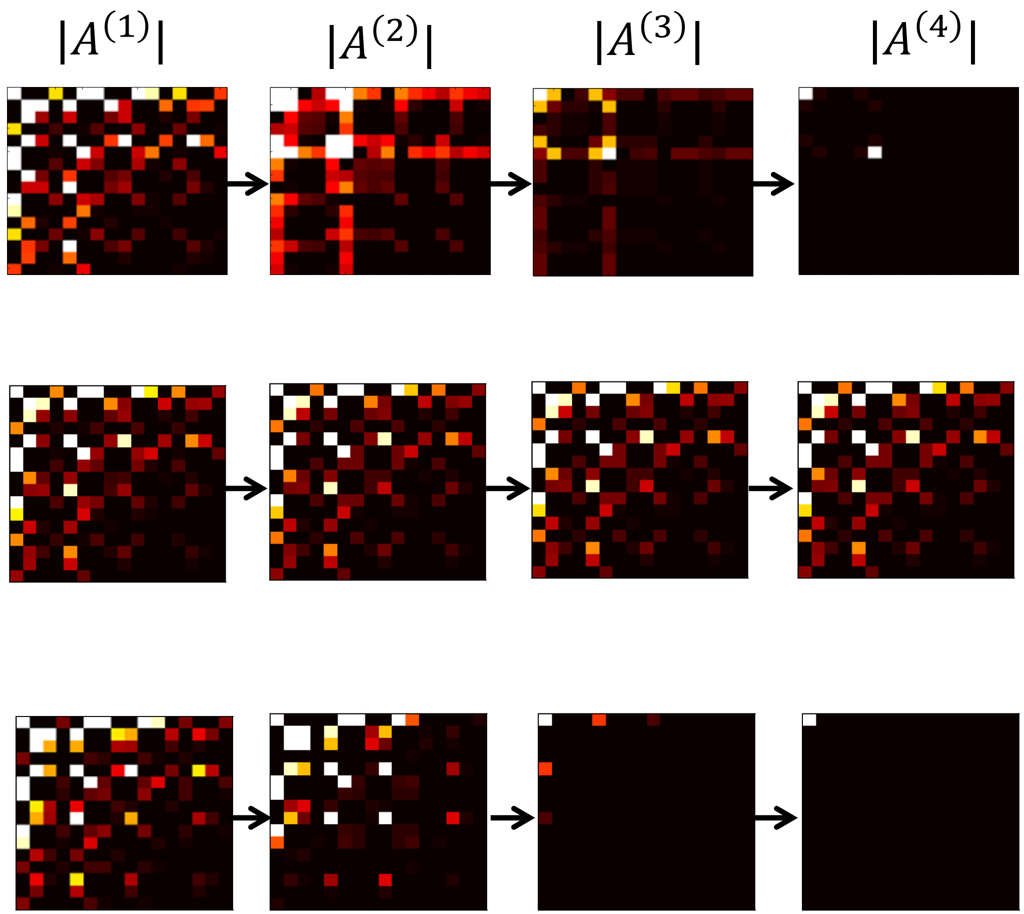


fixed points contain only universal information about the phase

Proper RG flow: 2D classical Ising

New Approach: Tensor Network Renormalization (TNR):

- Converges to one of three RG fixed points, consistent with a proper RG flow



sub-critical

$T = 0.9\, T_c$

ordered (Z2) fixed point

critical

$T = T_c$

critical (scale-invariant) fixed point

super-critical

$T = 1.1\, T_c$

disordered (trivial) fixed point

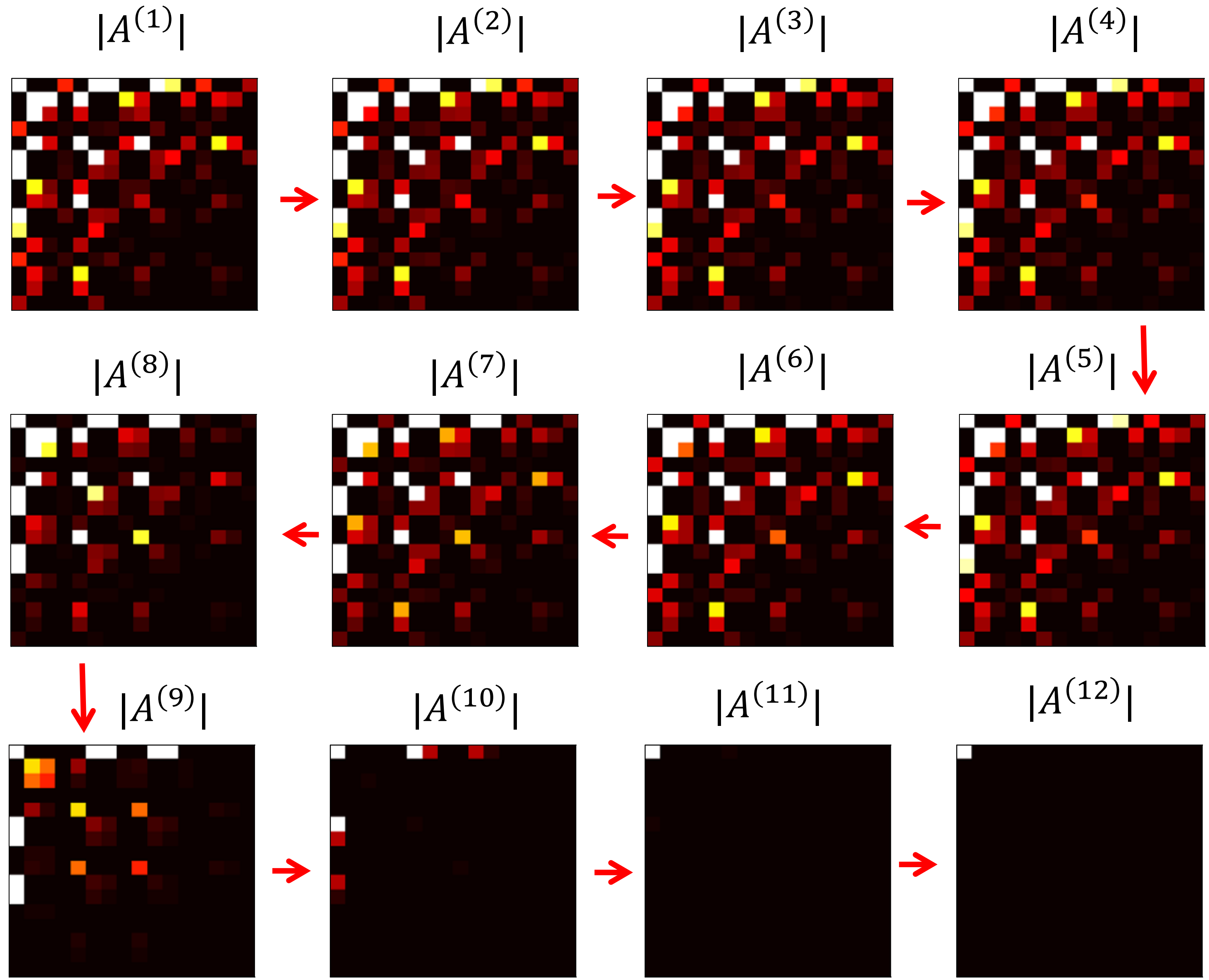
Proper RG flow: 2D classical Ising

more difficult!

$T = 1.002\, T_c$

TNR bond dimension:

$\chi = 4$



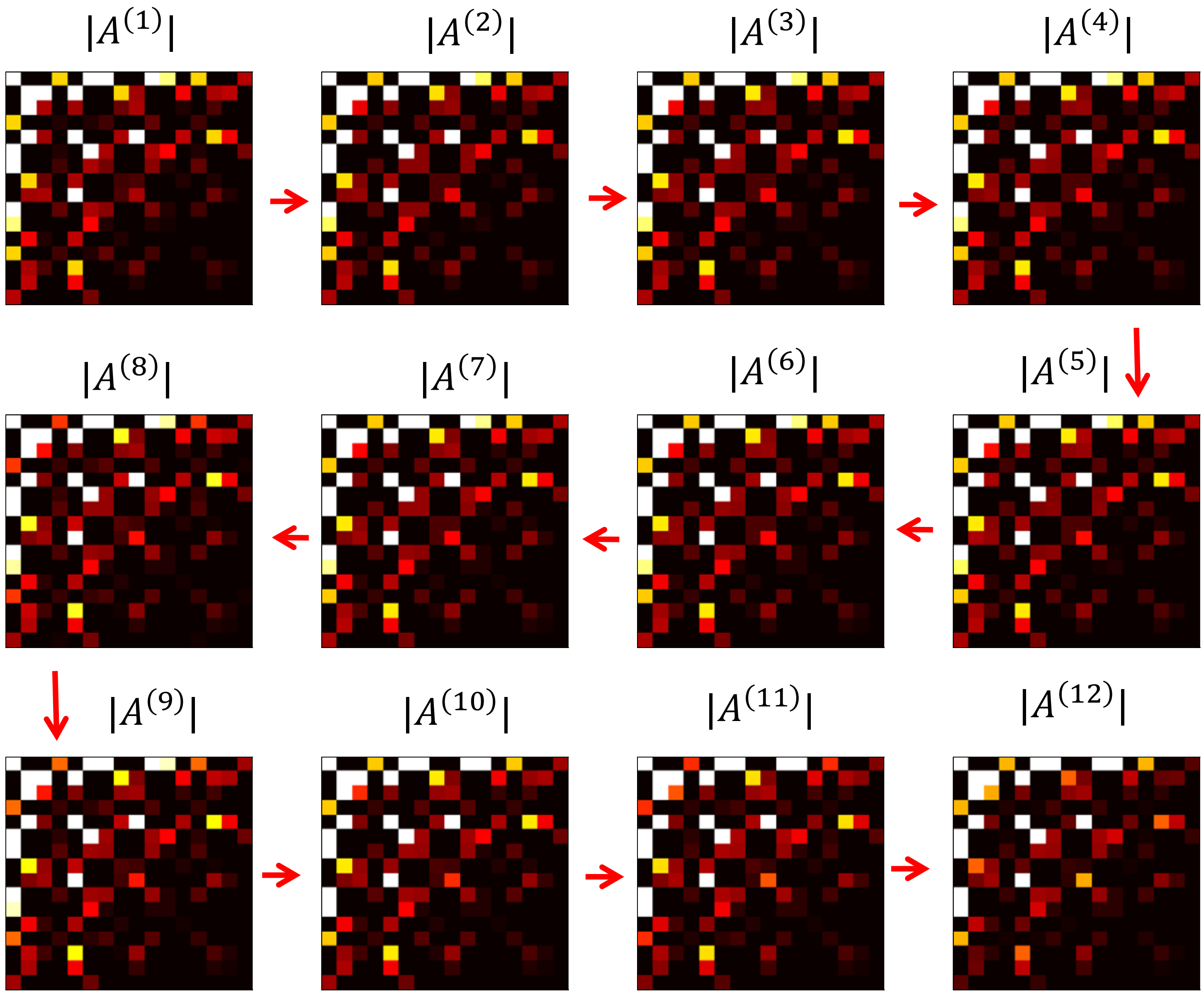
Proper RG flow: 2D classical Ising

critical point:

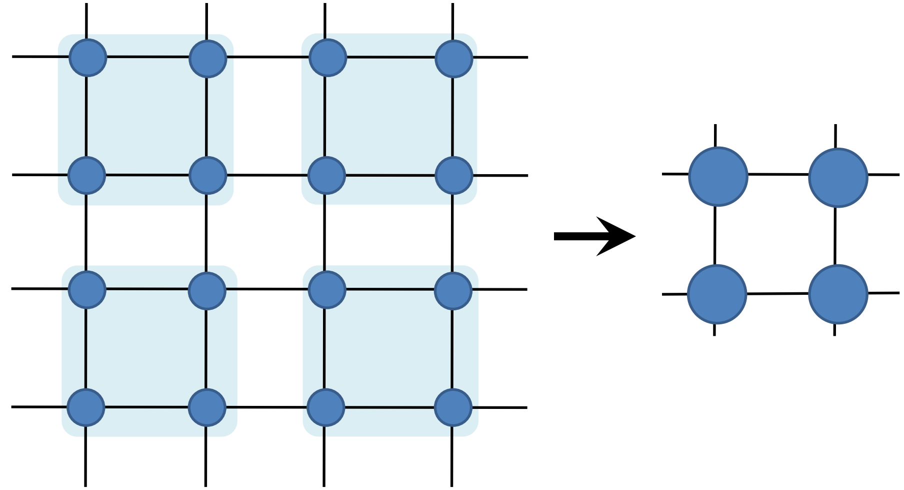
$T = T_c$

TNR bond dimension:

$\chi = 4$



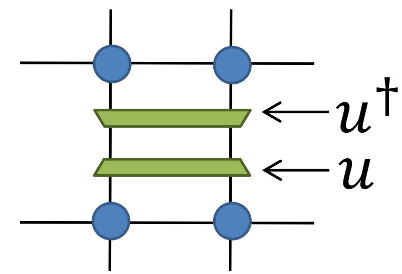
Summary



- We have discussed implementation of real-space RG for tensor networks
- Demonstrated that previous methods (e.g. Levin Nave TRG) do not generate a proper RG flow

cause: failure to address all short-range degrees of freedom at each RG step

Tensor Network Renormalization, arXiv:1412.0732



uses disentangling to address all short-range degrees of freedom at each RG step

- Proper RG flow: correct structure of RG fixed points
- Computationally sustainable RG flow

future work: implementation in higher dimensions, for contraction of PEPS, for study of impurity CFTs...

Outline: Tensor Network Renormalization

The set-up: Representation of partition functions and path integrals as tensor networks

Previous approaches: Levin and Nave's Tensor Renormalization Group (LN-TRG), conceptual and computation problems.

New approach: Tensor network renormalization (TNR): proper removal of all short-ranged degrees of freedom via disentanglers

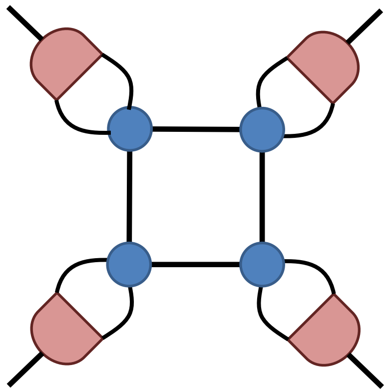
Benchmark results

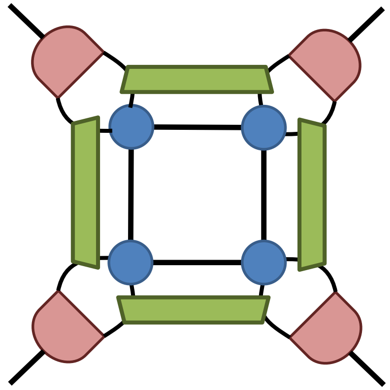
Extensions

TNR yields the MERA

(Evenbly, Vidal, arXiv:1502.05385)

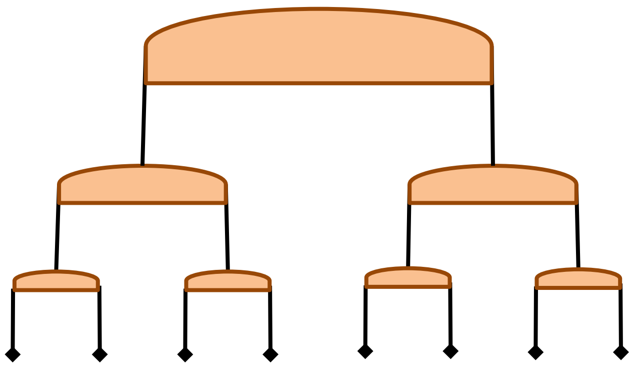
Tensor Renormalization Group (LN-TRG) **vs** Tensor Network Renormalization (TNR)

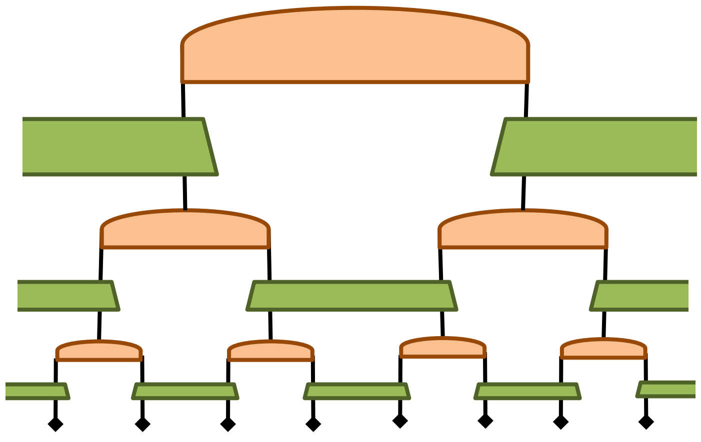




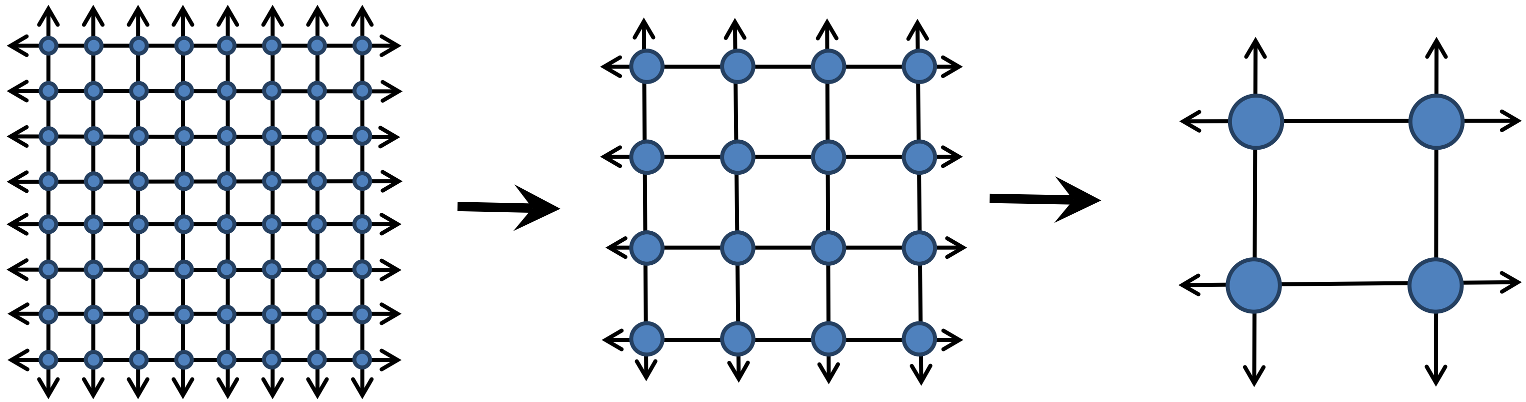
Analogous to:

Tree tensor network (TTN) **vs** Multi-scale entanglement renormalization ansatz (MERA)



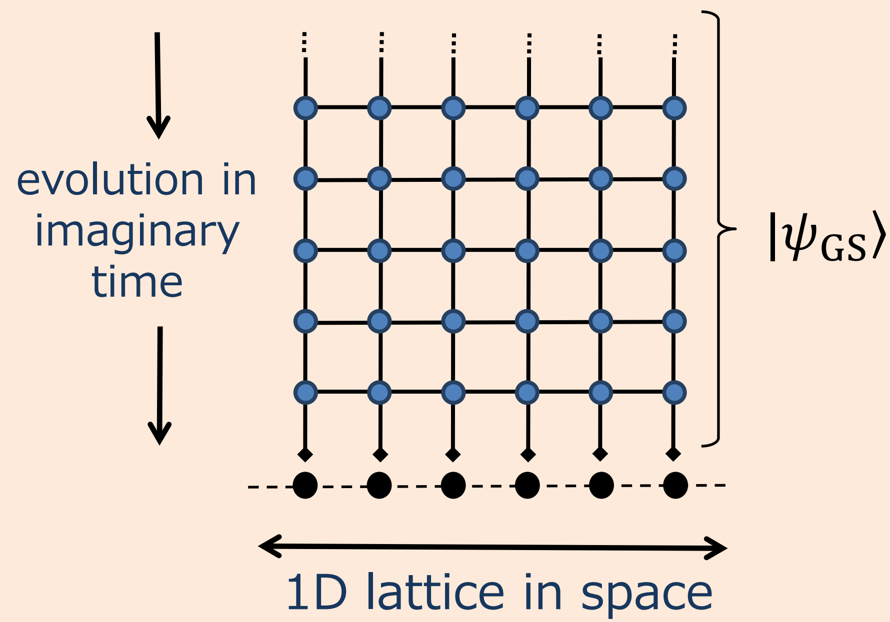


TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)

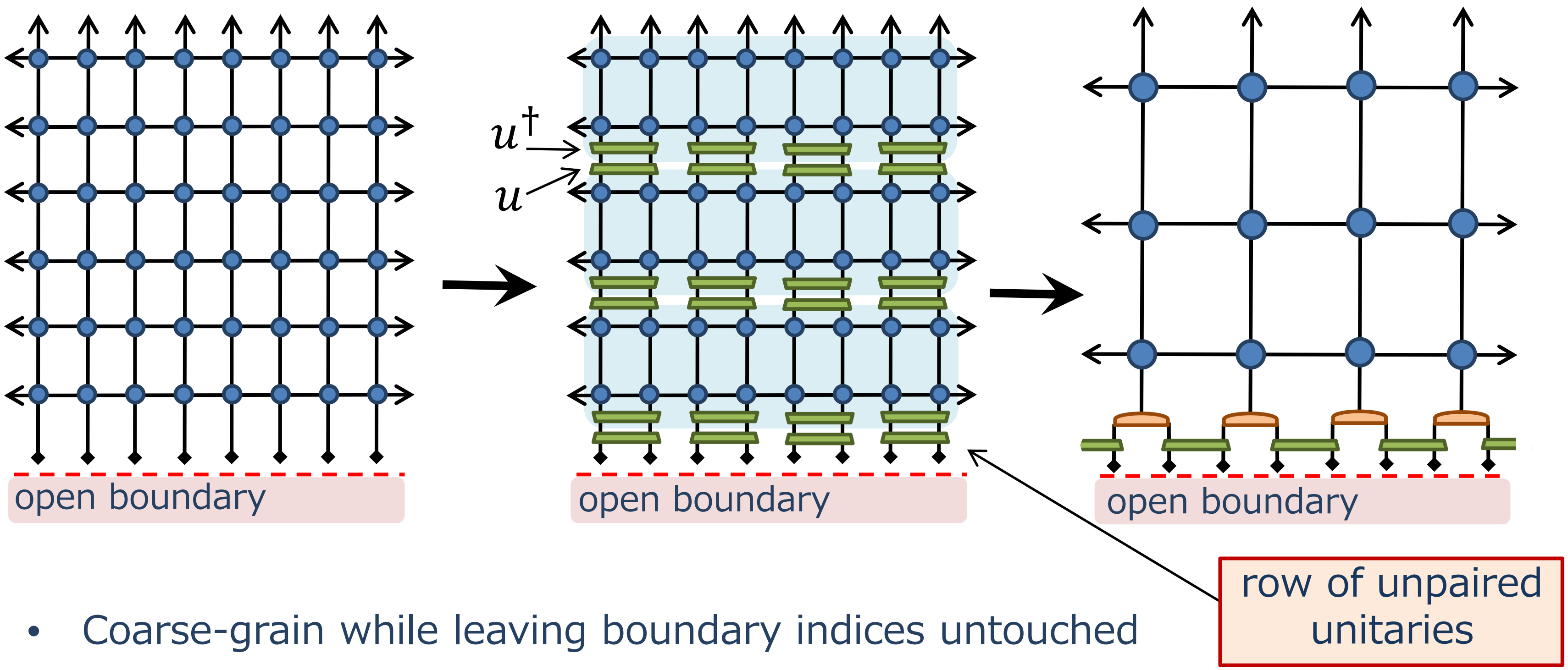


network with PBC

open boundaries?

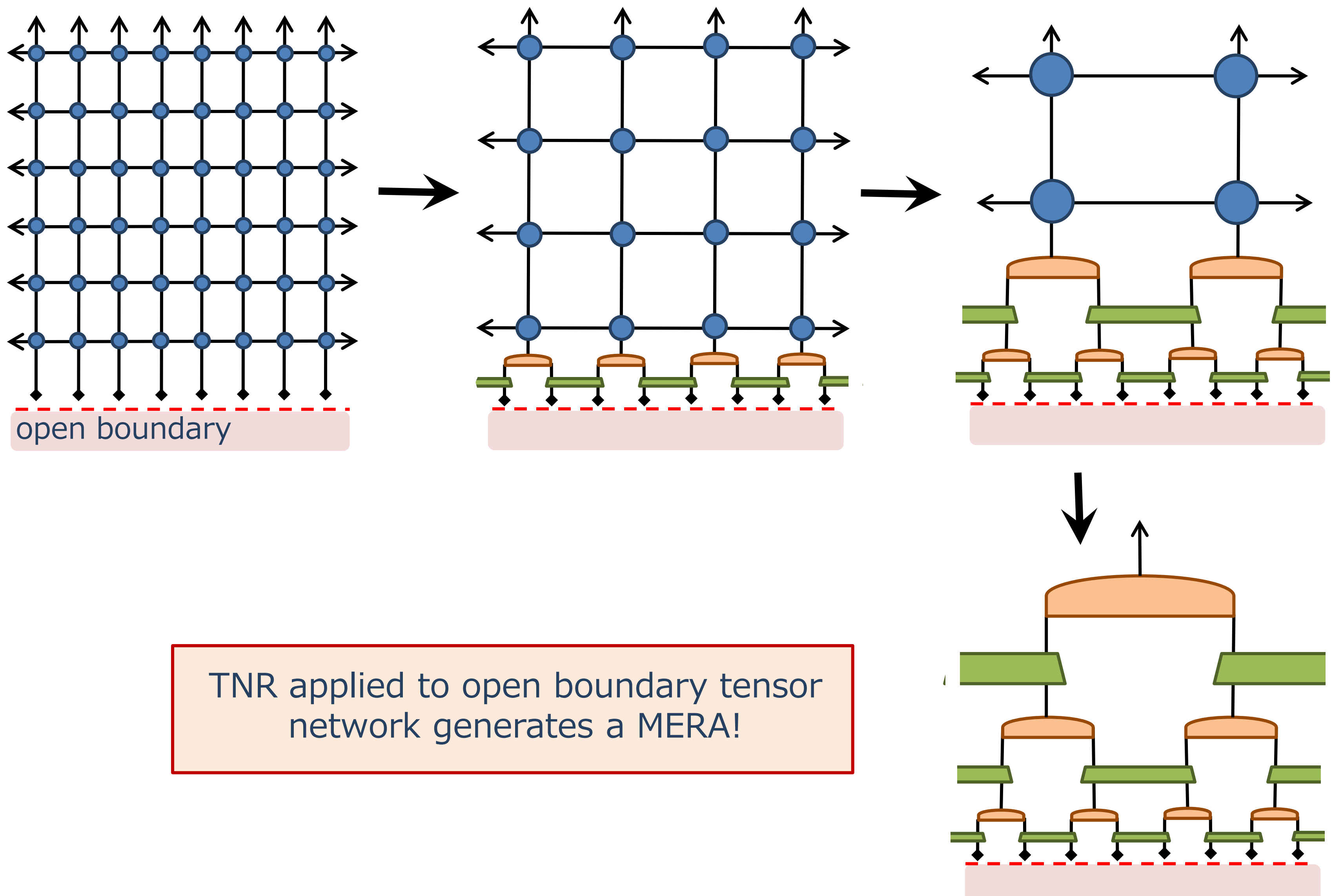


TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)



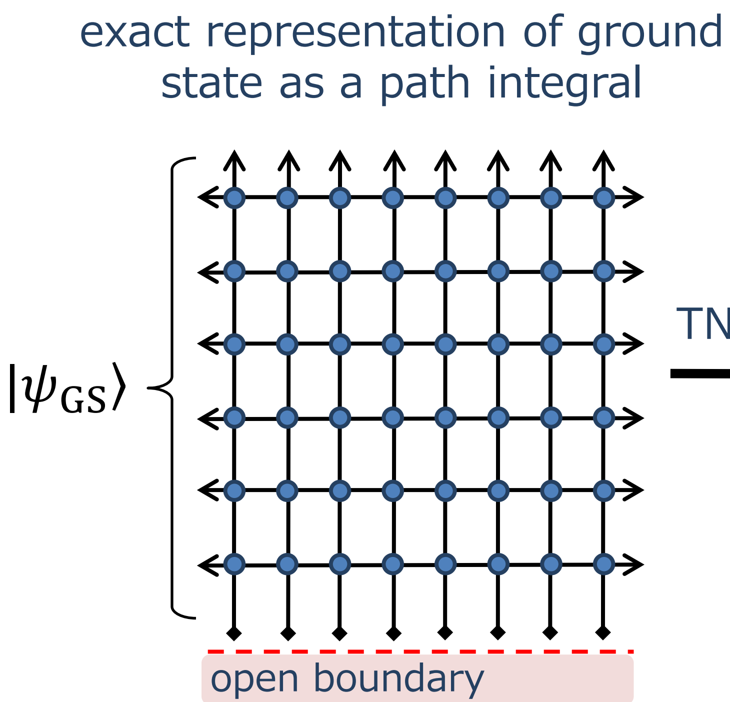
- Coarse-grain while leaving boundary indices untouched
- Disentanglers and isometries are inserted in conjugate pairs, eventually becoming a part of the coarse-grained tensors
- But a row of unpaired disentanglers and isometries remains on the open boundary···

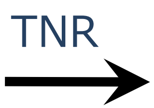
TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)

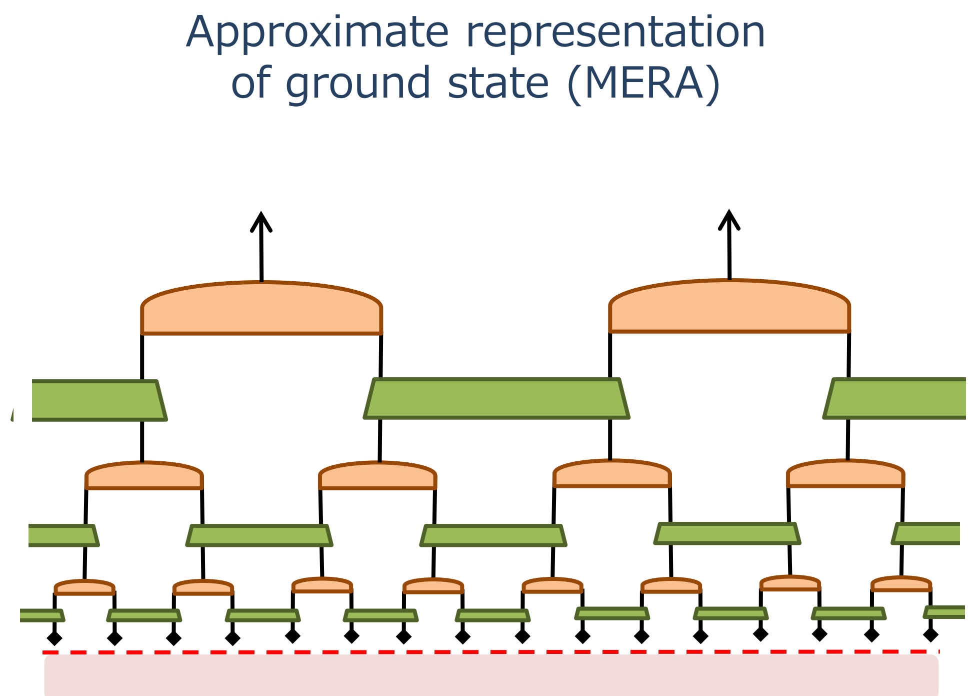


TNR applied to open boundary tensor network generates a MERA!

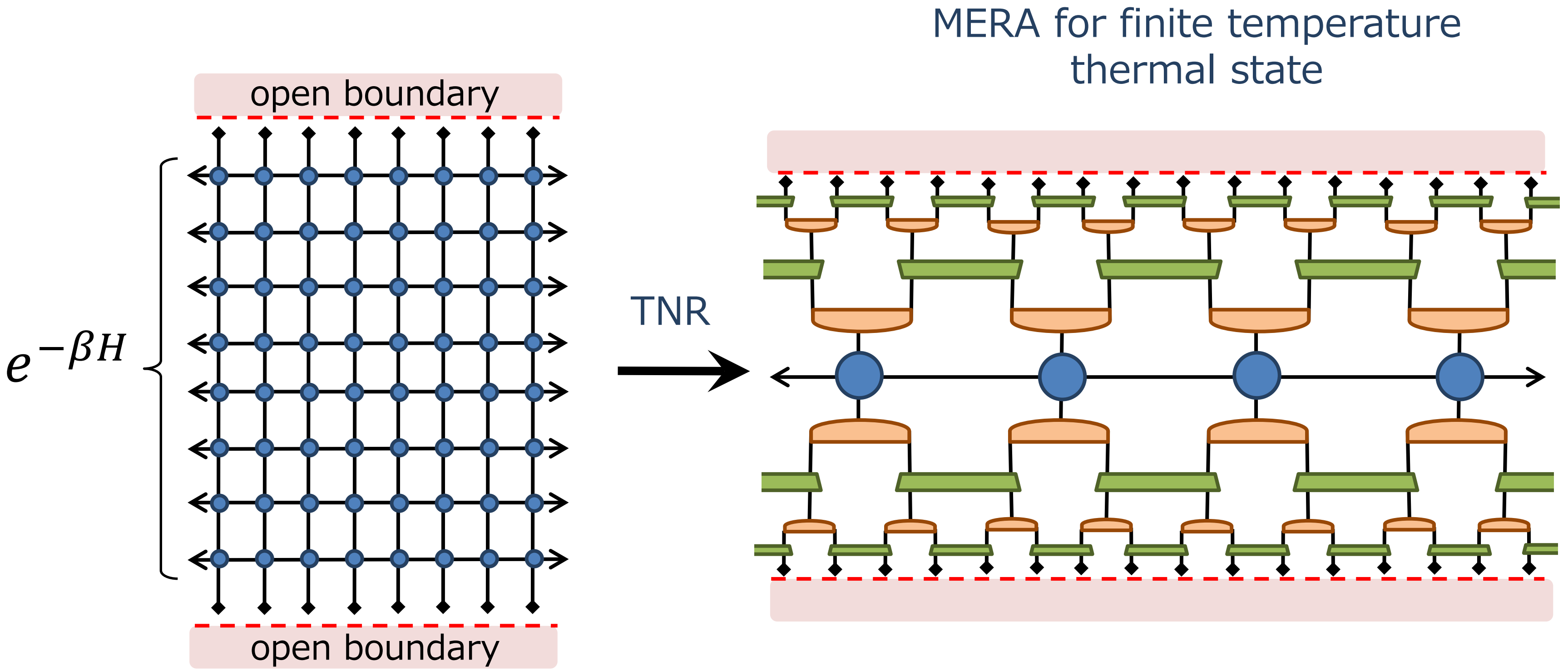
TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)





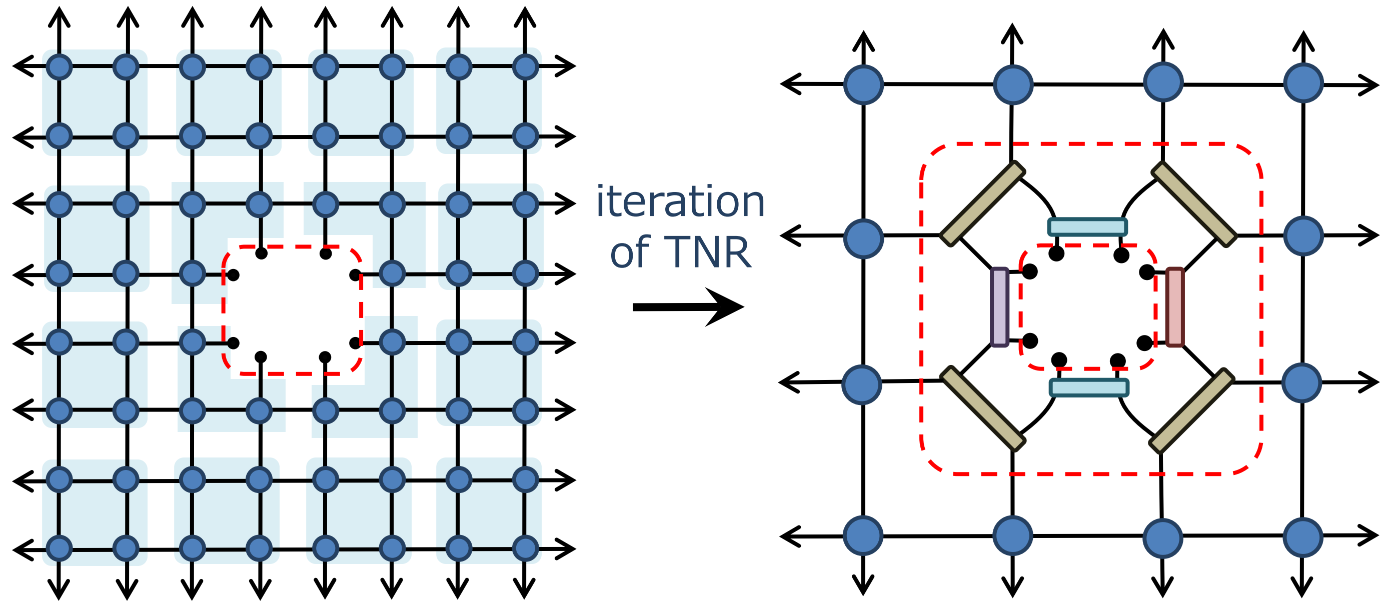


TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)



TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)

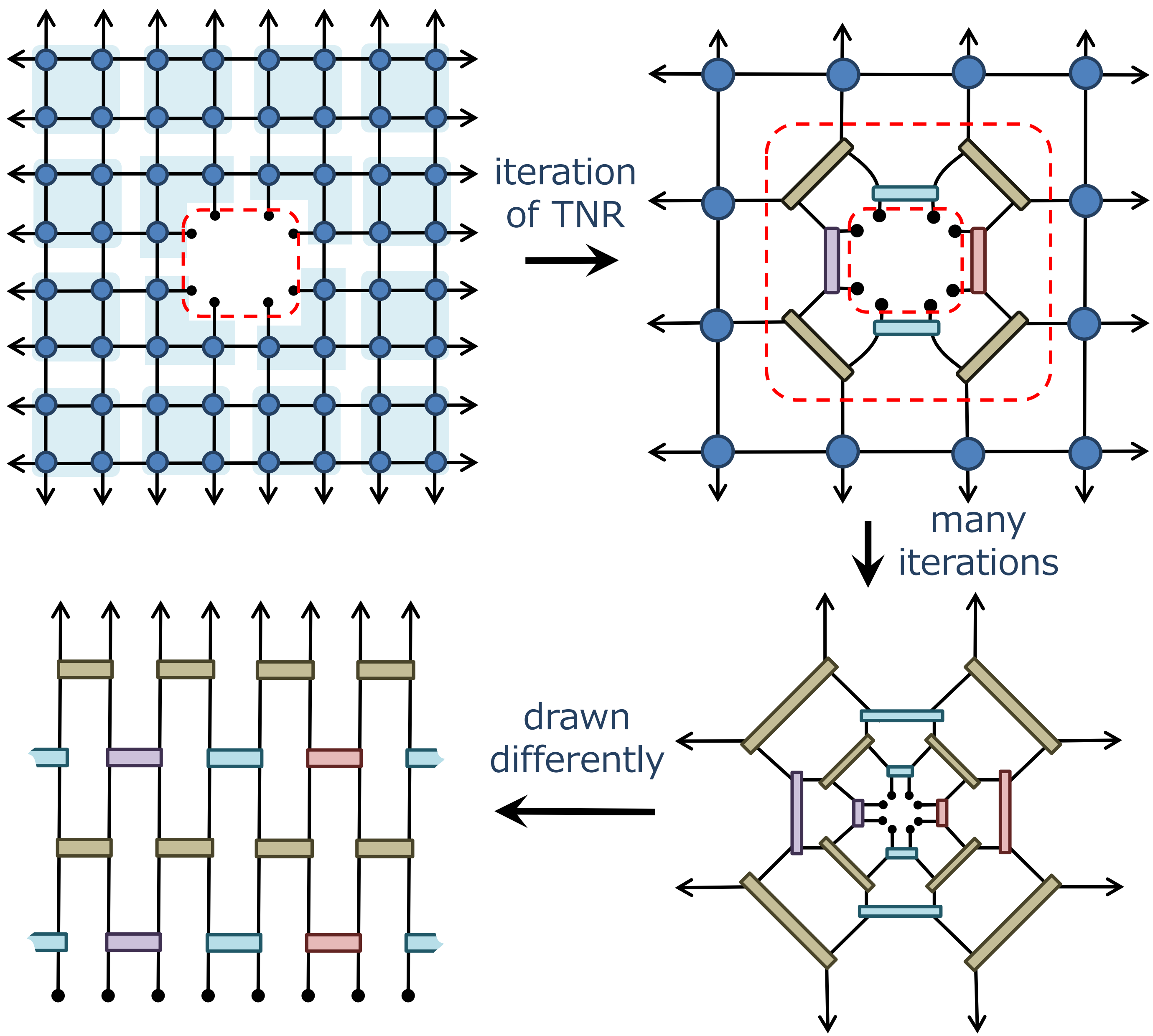
Tensor network with open ‘hole’:



coarse-grain as much as possible
(subject to leaving indices
around the hole untouched)

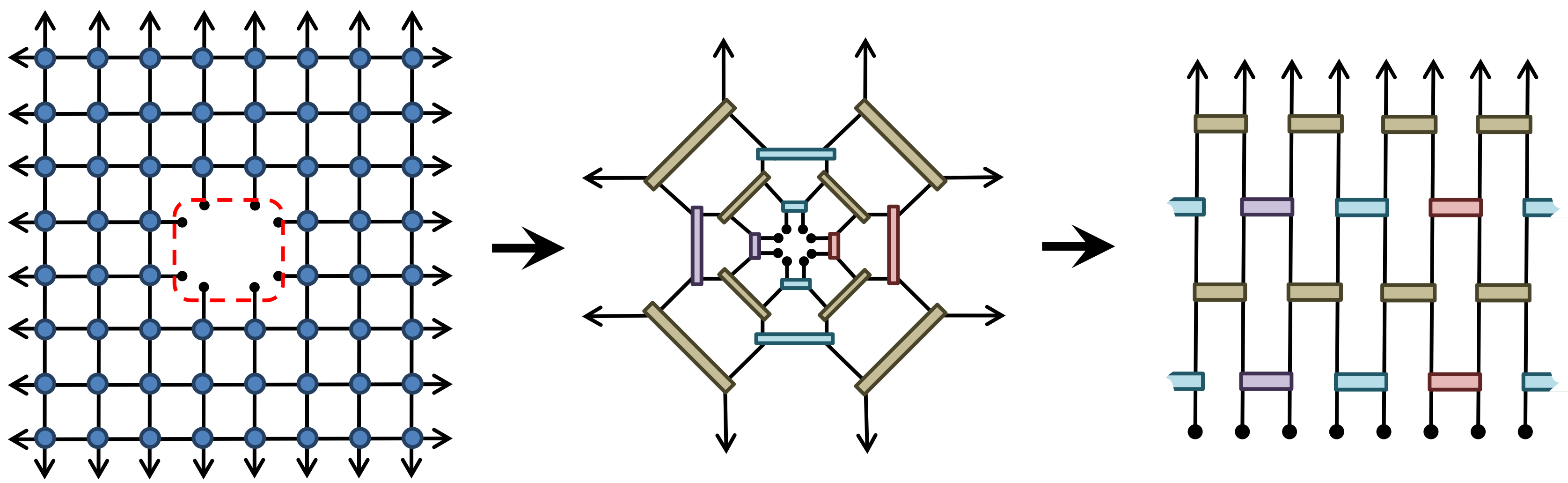
TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)

Tensor network
with open ‘hole’:

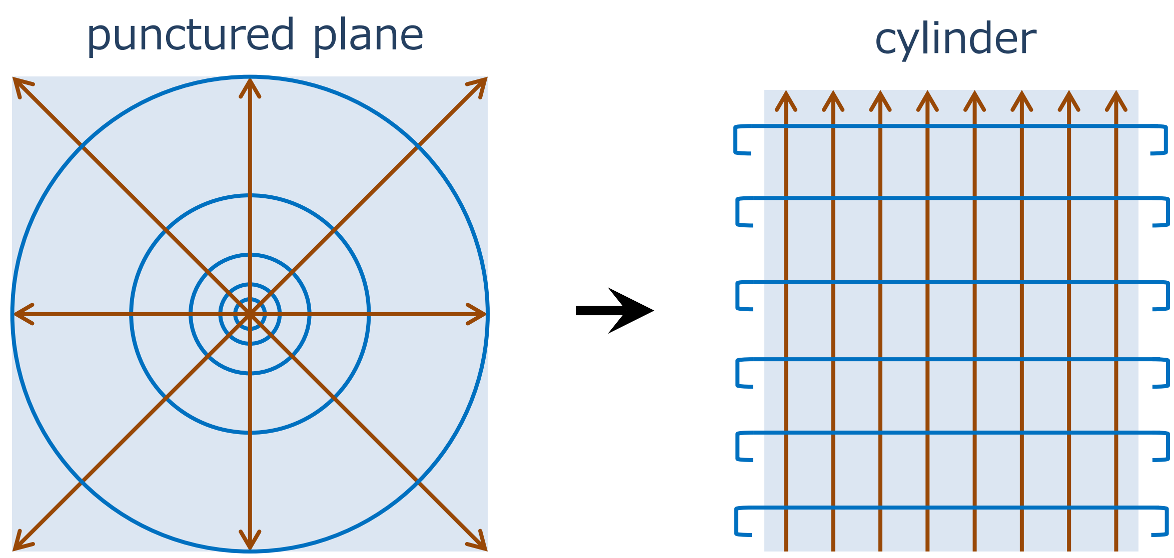


TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)

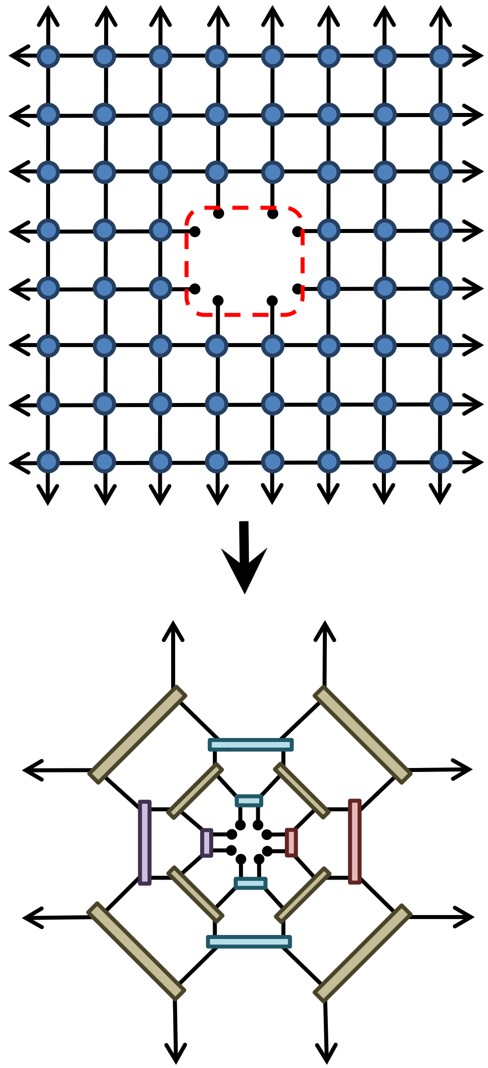
TNR for network with hole:



Logarithmic transform in CFT:



TNR yields the MERA (Evenbly, Vidal, arXiv:1502.05385)



diagonalize transfer operator:

Scaling dimensions from partition function of critical Ising