Title: Tensor network renormalization

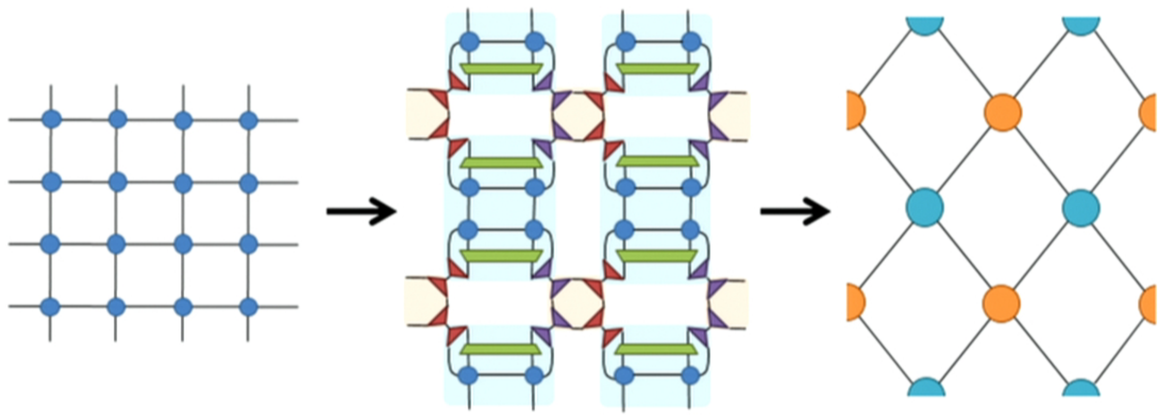
Date: Nov 19, 2014 11:00 AM

URL: http://pirsa.org/14110137

Abstract: <p>I will describe how to define a proper RG flow in the space of
tensor networks, with applications to the evaluation of classical
partition functions, euclidean path integrals, and overlaps of tensor
network states.</p>

Perimeter Institute, November 2014

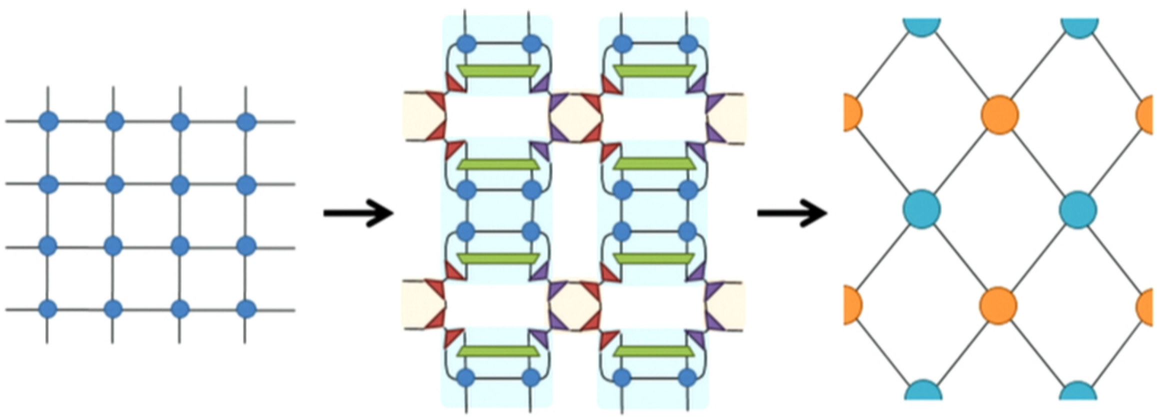
Tensor Network Renormalization



Glen Evenbly
Guifre Vidal

Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)

A new RG based method to contract tensor networks, with applications towards simulation of quantum and classical many-body systems



Overview: Tensor Networks

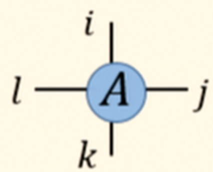
bond dimension

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

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

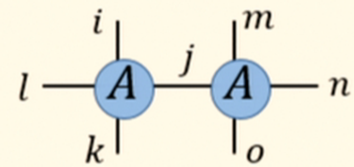
Diagrammatic notation:

A_{ijkl} $\longleftrightarrow$



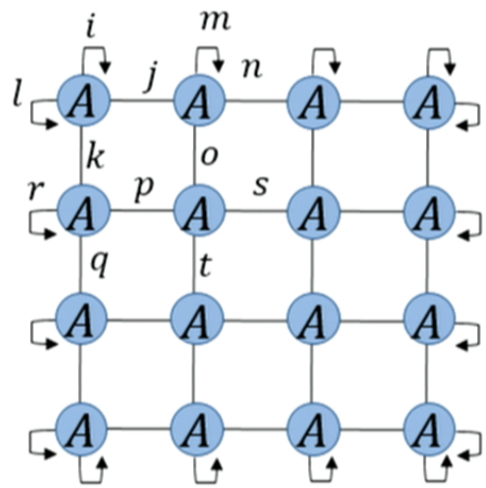
Contraction of two tensors:

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

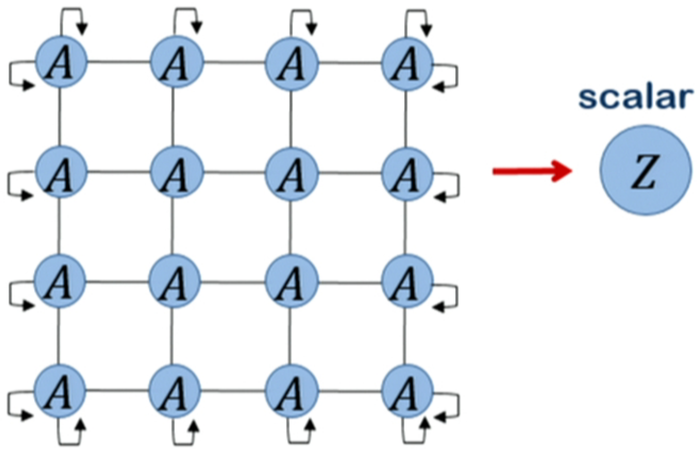


Square lattice network (PBC):

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



Overview: Tensor Networks



Task: we want a method for efficient (approximate) numerical evaluation of this scalar

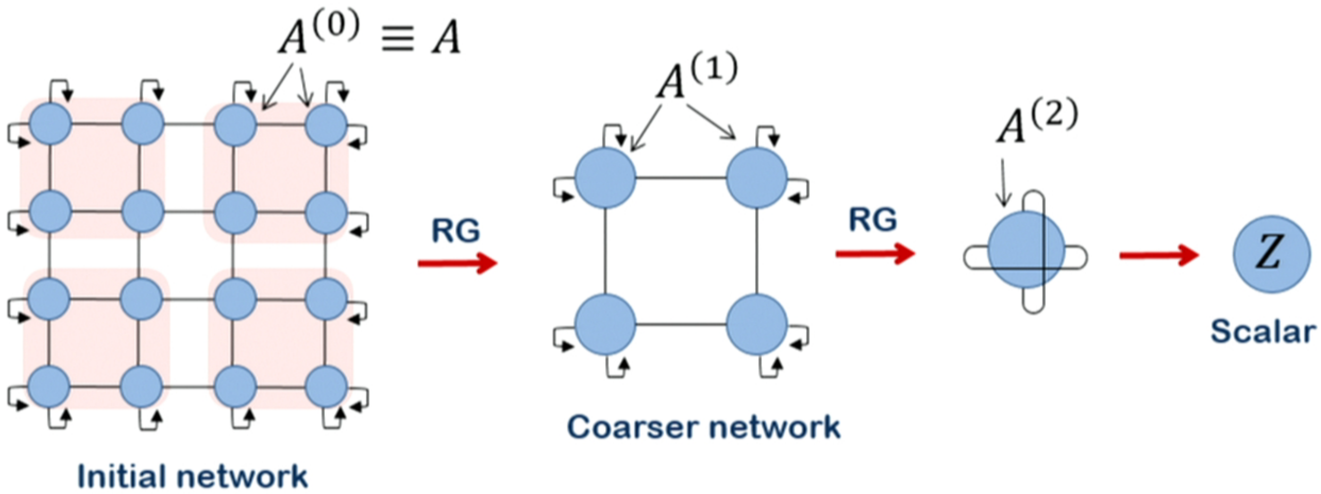
Contraction of D dim tensor network could allow one to:

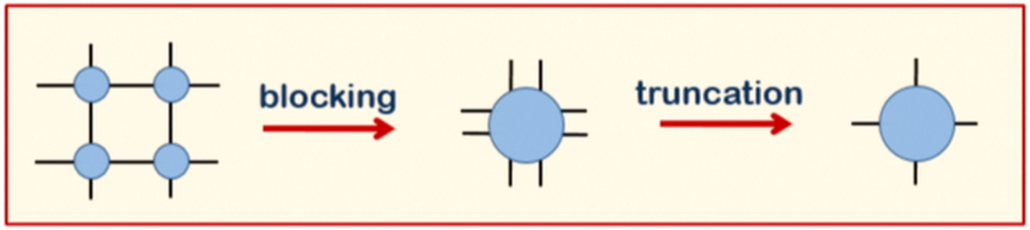
- compute properties of D dim classical many body systems (where the tensor network represents a partition function)
- compute properties of (D-1) dim quantum many body systems (where the tensor network represents the Euclidean path integral)
- plus other applications....

Overview: RG transformations of tensor networks

Many different strategies could be employed for contracting a tensor network

Today I consider approaches based upon successive use of renormalization group (RG) transformations:





Overview: RG transformations of tensor networks

Many different strategies could be employed for contracting a tensor network

Today I consider approaches based upon successive use of renormalization group (RG) transformations:

RG flow in the space of tensors:

$$\underbrace{A^{(0)} \rightarrow A^{(1)} \rightarrow A^{(2)} \rightarrow \cdots \rightarrow A^{(S)}}^{\log(N)\ \text{steps}} \rightarrow \overset{\text{scalar}}{Z}$$

Previous RG approaches:

- Tensor Renormalization Group (TRG) (Levin, Nave, 2006)
- Second Renormalization Group (SRG) (Xie, Jiang, Weng, Xiang, 2008)
- Tensor Entanglement Filtering Renormalization (TEFR) (Gu, Wen, 2009)
- Higher Order Tensor Renormalization Group (HOTRG) (Xie, Chen, Qin, Zhu, Yang, Xiang, 2012)

+ many more…

Overview: RG transformations of tensor networks

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

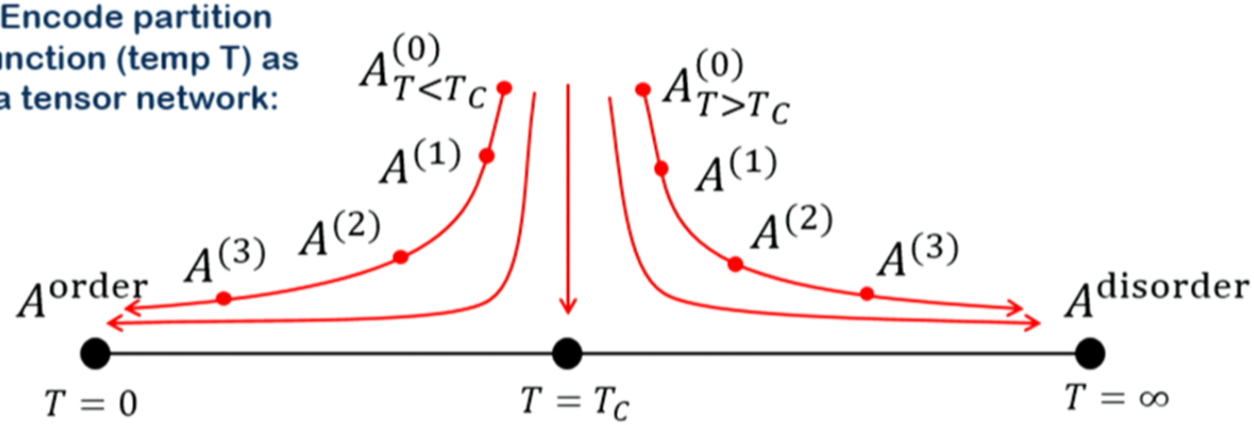
Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)

- an approach that generates a proper RG flow in the space of tensors

Consider 2D classical Ising ferromagnet at temperature T:

$T < T_C$	ordered phase (Z_2 symmetry broken)
$T = T_C$	critical point (correlations at all length scales)
$T > T_C$	disordered phase

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



Overview: RG transformations of tensor networks

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

Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)

- an approach that generates a proper RG flow in the space of tensors

Practical consequences (as a numerical method) at (or near) criticality: $T = T_C$

computational cost of iteration 's' of previous tensor RG schemes: $\longrightarrow$ $\text{cost} \sim \exp(s)$

computational cost of iteration 's' of TNR: $\longrightarrow$ $\text{cost} \sim \text{independent of } s$

Outline: Tensor Network Renormalization

Part I: Motivation

Representing partition functions of classical many body systems as tensor networks

Representing Euclidean path integrals of quantum many-body systems as tensor networks

Scalar products of PEPS

Part II: Previous RG schemes

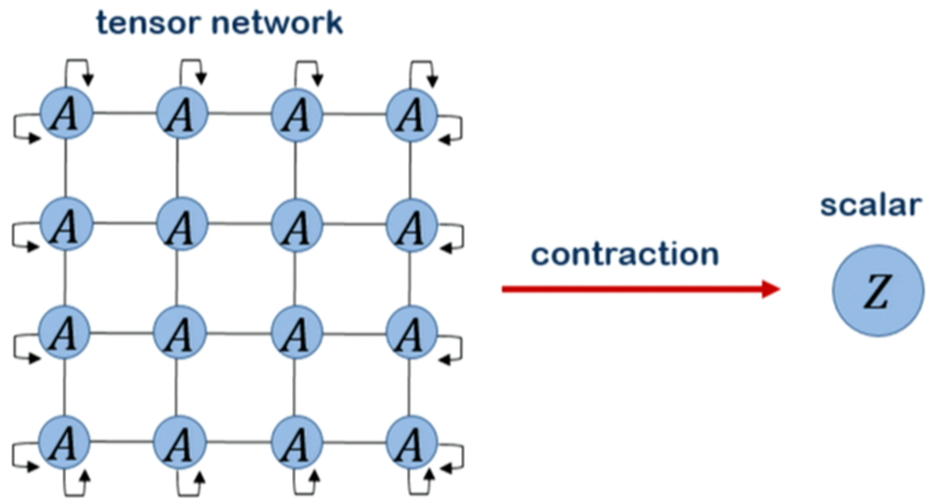
The tensor renormalization group (TRG) approach

Failure of previous schemes to give proper RG flow

Part III: Tensor Network Renormalization (TNR)

Formulation, benchmark results, other applications

Encoding many-body physics in tensor networks

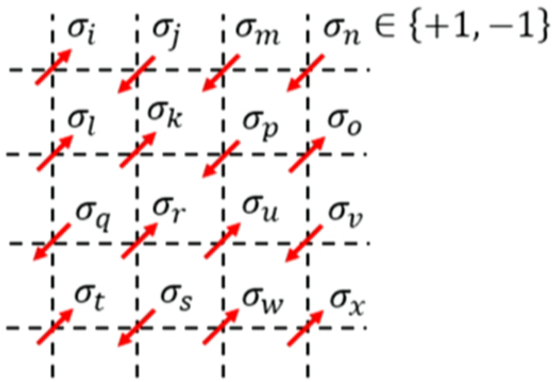


What is the physical relevance of this procedure?

(i) Computing information about classical many-body systems (via evaluation of the partition function)

Encoding partition functions as tensor networks

Square lattice of Ising spins:



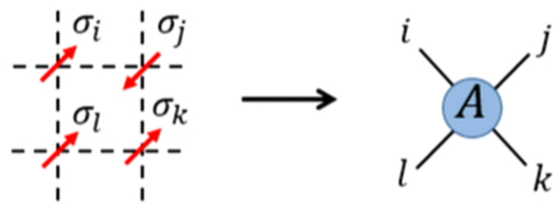
Hamiltonian functional for Ising ferromagnet:

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

Partition function:

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

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

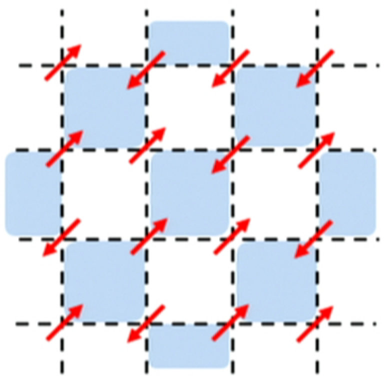


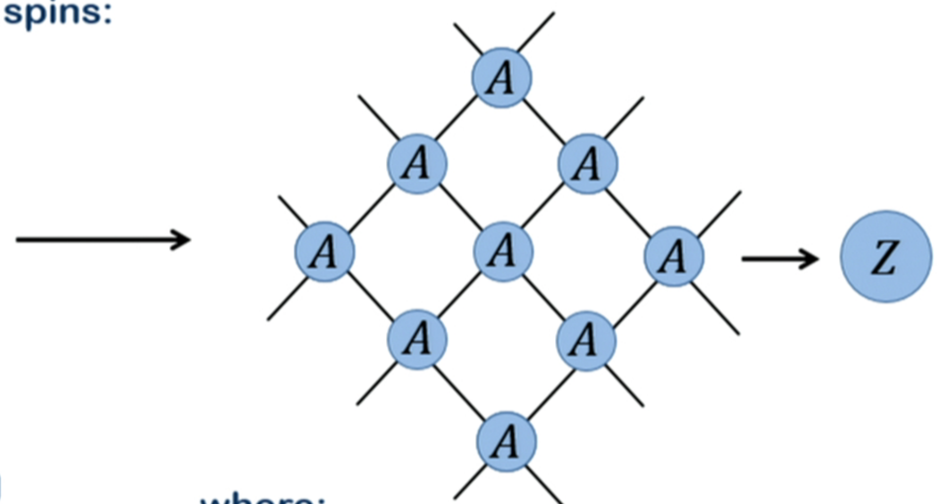
where:

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

Encoding 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:

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

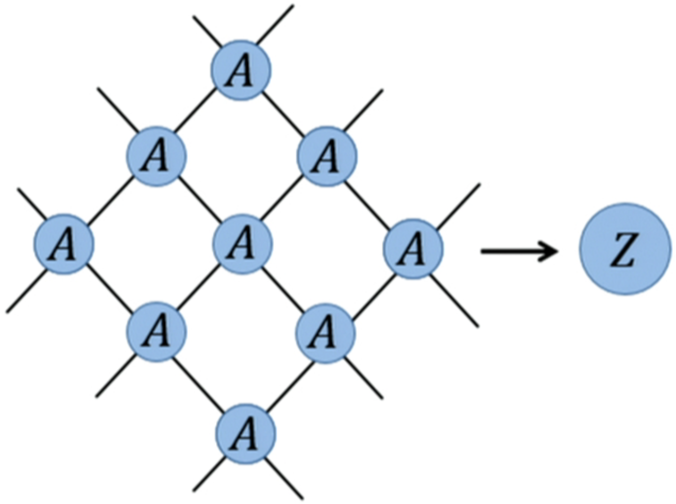
Partition function:

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

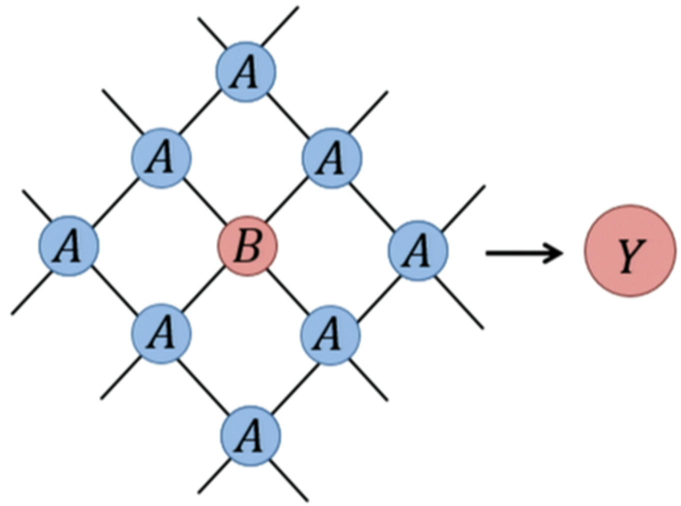
Partition function given by contraction of tensor network

Encoding partition functions as tensor networks

Partition function:

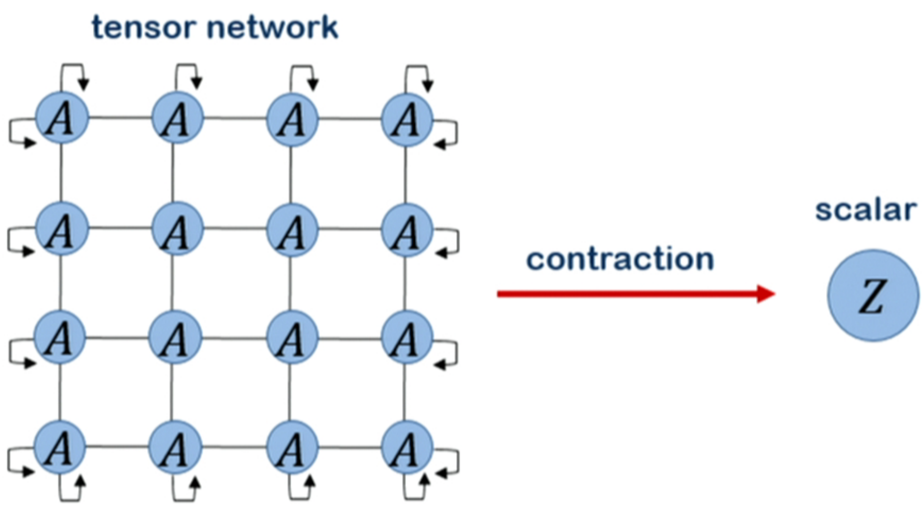


Replace a single tensor in the network:



Expectation value of local observable: $\langle o \rangle_\beta = \dfrac{Y}{Z}$

Encoding many-body physics in tensor networks



What is the physical relevance of this procedure?

(i) Computing information about classical many-body systems (via evaluation of the partition function)

(ii) Computing information about quantum many-body systems (via evaluation of the Euclidean path integral)

Encoding Euclidean path integrals as tensor networks

Nearest neighbour Hamiltonian for a 1D quantum system:

$$H = \sum_r h(r, r+1)$$

Evolution in imaginary time yields projector onto ground state:

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

Goal: express Euclidean path integral as a tensor network

$$\lim_{\beta\to\infty}\left[e^{-\beta H}\right] \longrightarrow \text{T.N.}$$

Separate into even and odd terms

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

Expand in small time steps

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

Where it is then seen

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

Encoding Euclidean path integrals as tensor networks

Separate Hamiltonian into even and odd terms:

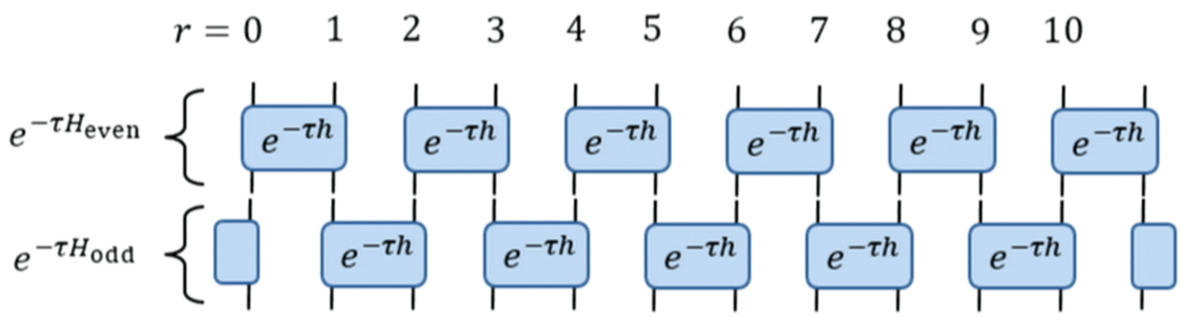
$$H = \sum_{r\ \text{even}} h(r, r+1) + \sum_{r\ \text{odd}} h(r, r+1) \ = H_{\text{even}} + H_{\text{odd}}$$

Expand path integral in small discrete time steps:

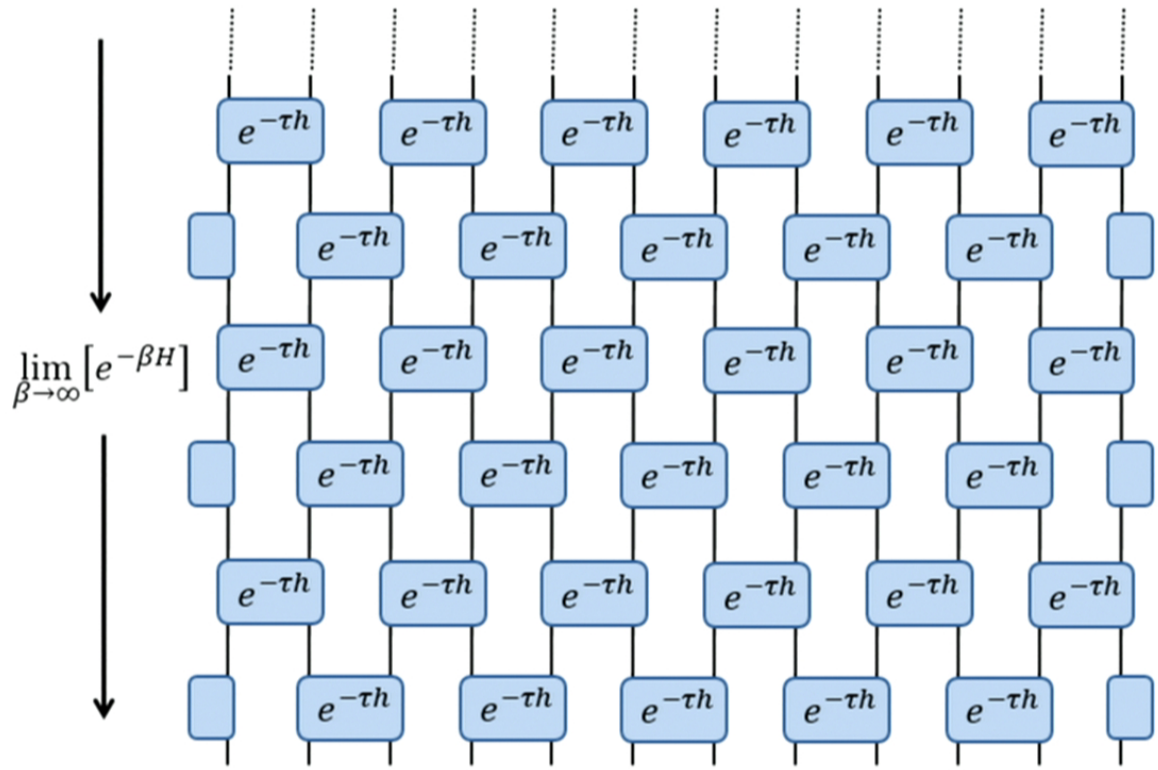
$$\lim_{\beta \to \infty} \left[e^{-\beta H} \right] = e^{-\tau H} e^{-\tau H} e^{-\tau H} e^{-\tau H} \ldots$$

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

Exponentiate even and odd separately :



Encoding Euclidean path integrals as tensor networks



Encoding Euclidean path integrals as tensor networks

Given 1D quantum Hamiltonian:

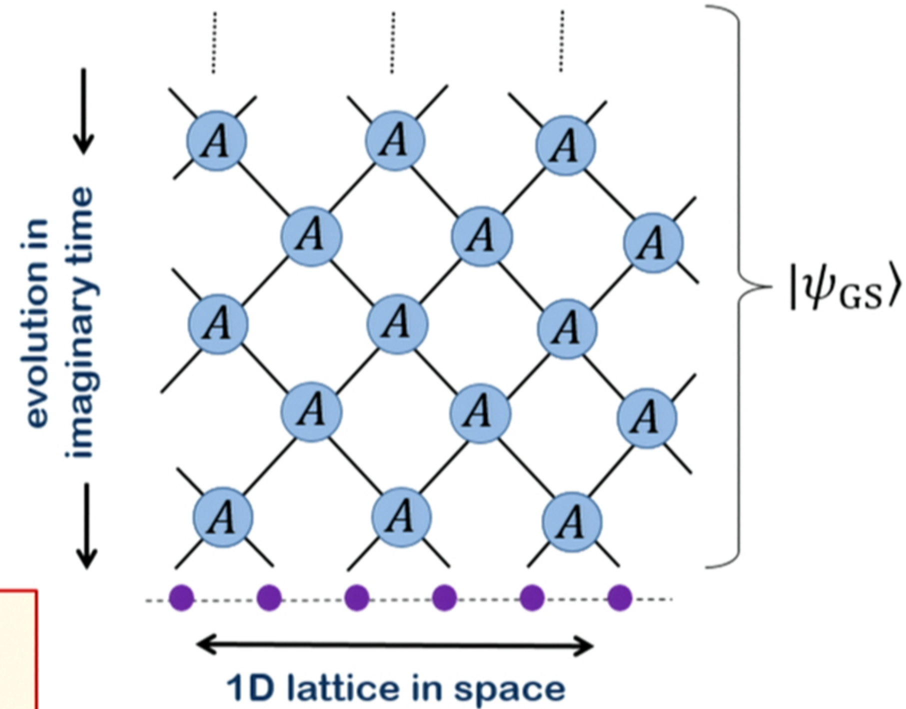
$$H = \sum_r h(r, r+1)$$

Set tensors:

$$A = \exp(-\tau h)$$

for sufficiently small time-step τ

The tensor network is a representation of the ground state $|\psi_{GS}\rangle$ of the quantum system



Encoding Euclidean path integrals as tensor networks

Given 1D quantum Hamiltonian:

$$H = \sum_r h(r, r+1)$$

Set tensors:

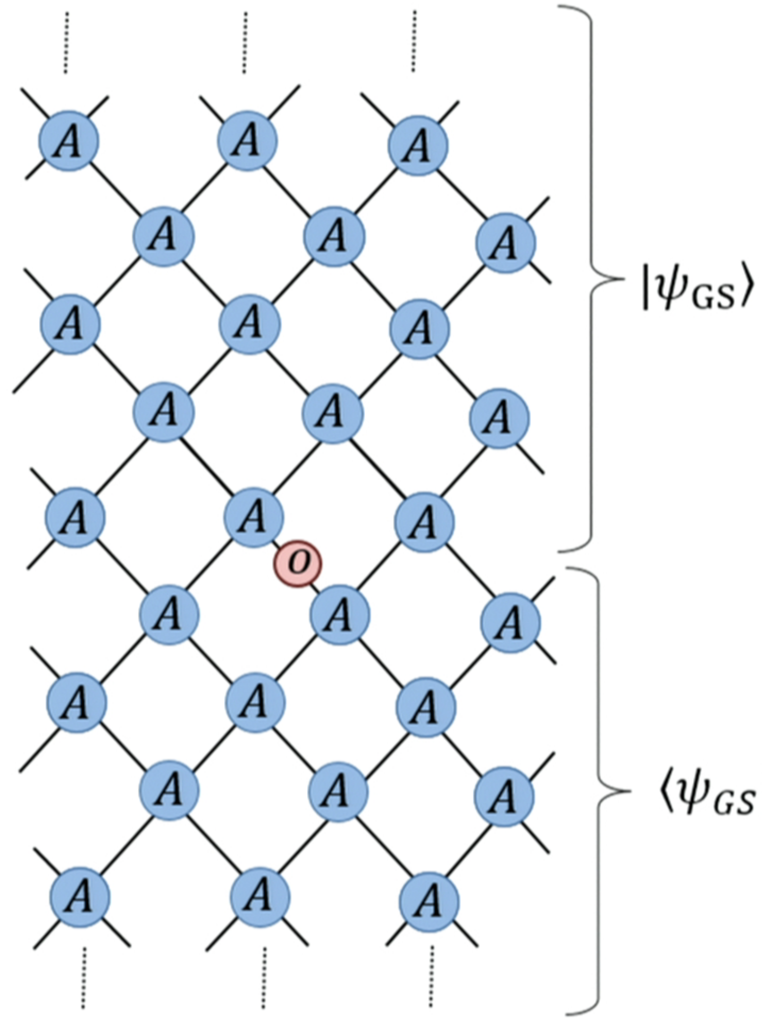
$$A = \exp(-\tau h)$$

for sufficiently small time-step τ

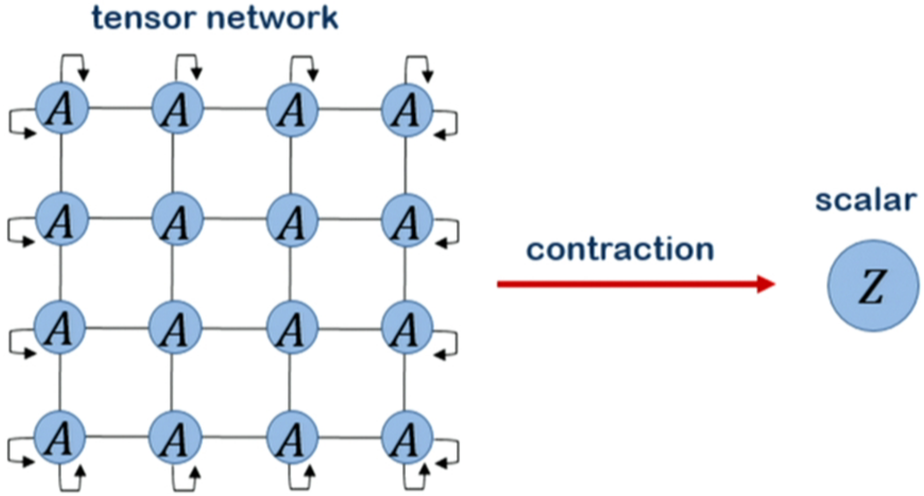
The tensor network is a representation of the ground state $|\psi_{GS}\rangle$ of the quantum system

Expectation value of local operator:

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



Encoding many-body physics in tensor networks



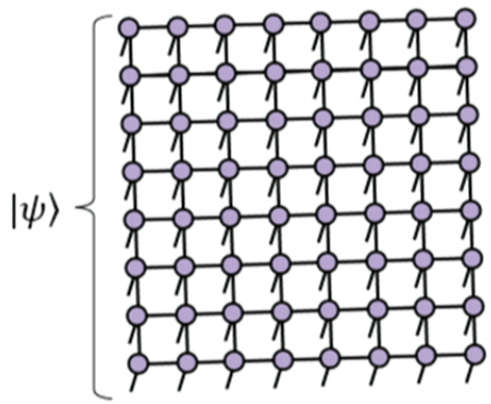
What is the physical relevance of this procedure?

(i) Computing information about classical many-body systems (via evaluation of the partition function)

(ii) Computing information about quantum many-body systems (via evaluation of the Euclidean path integral)

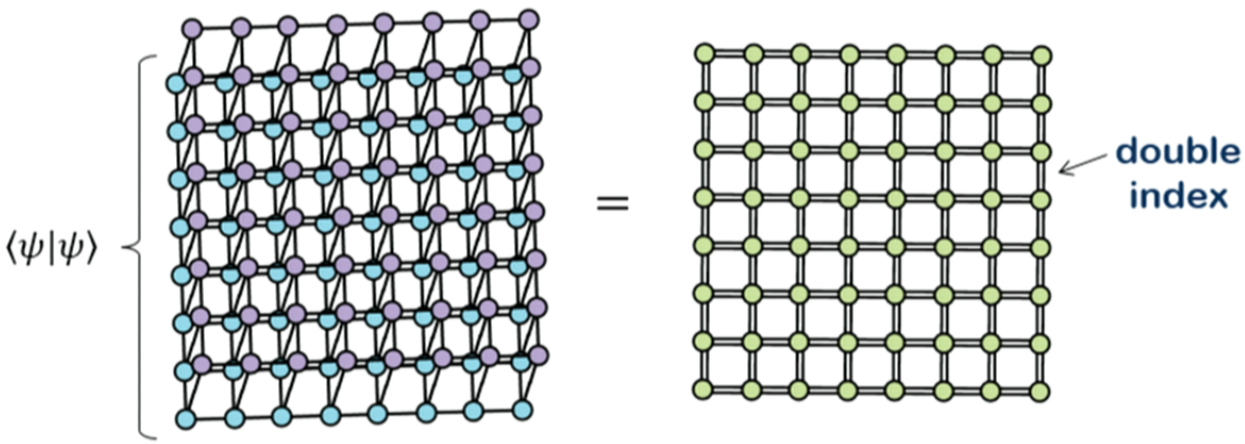
(iii) Contracting projected entangled pair states (PEPS)

Projected entangled pair states (PEPS)



Tensor network ansatz for
2D quantum systems

Important part of PEPS algorithms is
in evaluation of scalar products and
expectation values of local observables



Outline: Tensor Network Renormalization

Part I: Motivation

Representing partition functions of classical many body systems as tensor networks

Representing Euclidean path integrals of quantum many-body systems as tensor networks

Scalar products of PEPS

Part II: Previous RG schemes

The tensor renormalization group (TRG) approach

Failure of previous schemes to give proper RG flow

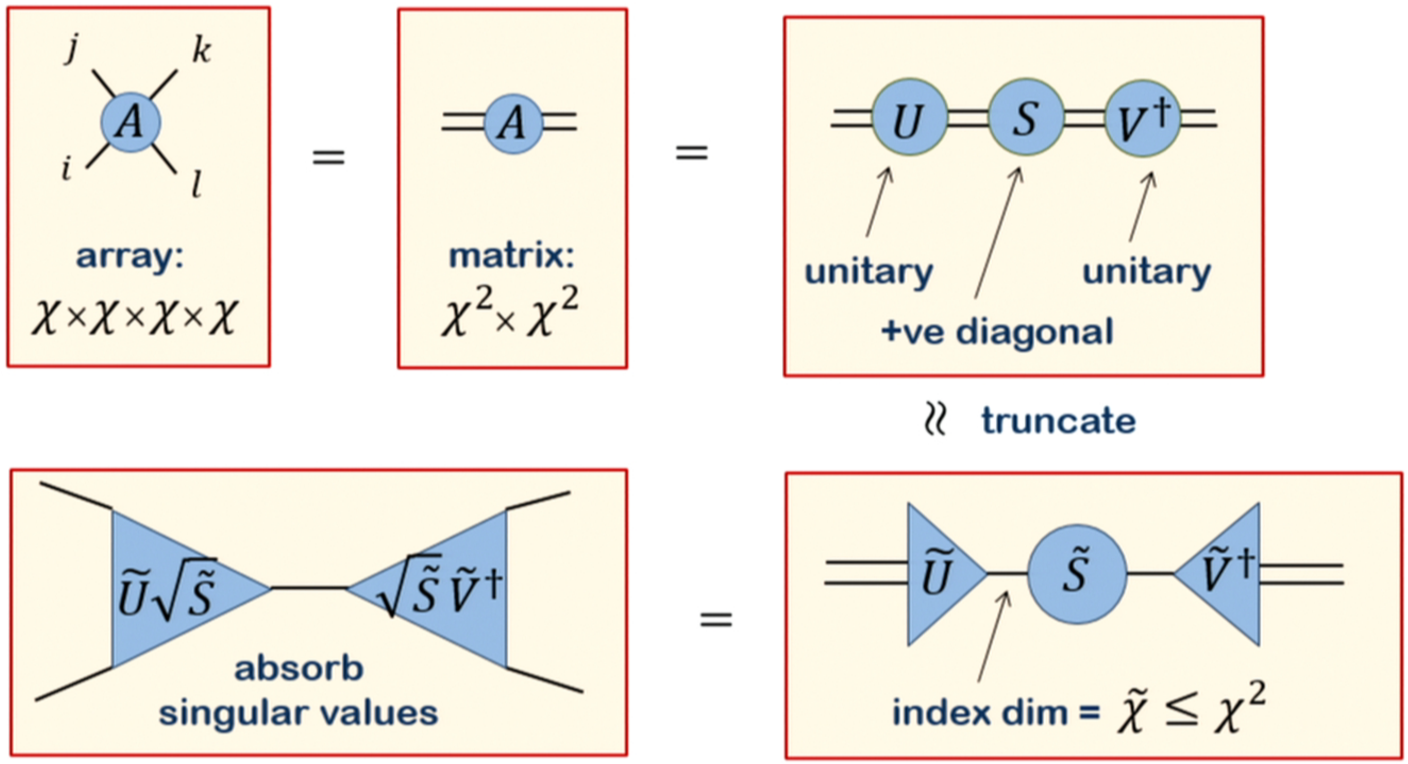
Part III: Tensor Network Renormalization (TNR)

Formulation, benchmark results, other applications

Tensor Renormalization Group (TRG) (Levin, Nave, 2006)

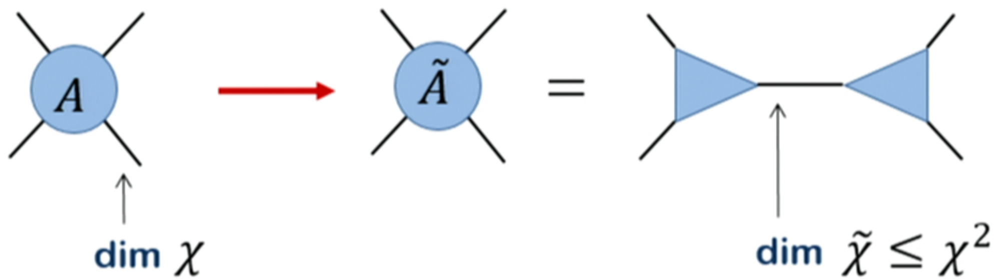
Preliminary: truncated singular value decomposition (SVD)

$$A_{ijkl} \xrightarrow{\text{reshape}} A_{(ij)(kl)} \xrightarrow{\text{SVD}} \sum_{m=1}^{\chi^2} U_{(ij)m} S_{mm} V_{m(kl)}$$



Tensor Renormalization Group (TRG) (Levin, Nave, 2006)

Preliminary: truncated singular value decomposition (SVD)

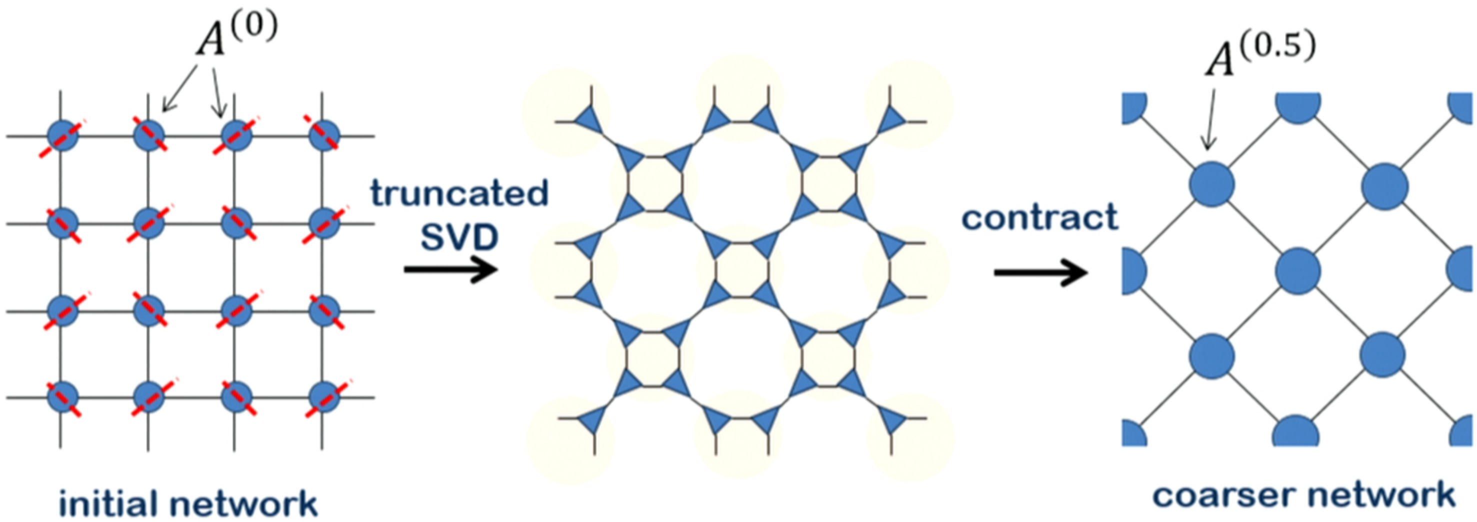


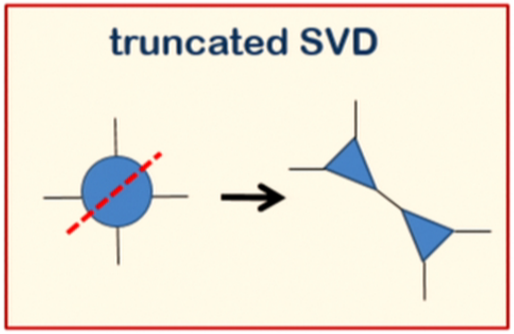
most accurate decomposition of a four index tensor into a pair of three index tensors (for a fixed bond dimension $\tilde{\chi}$)

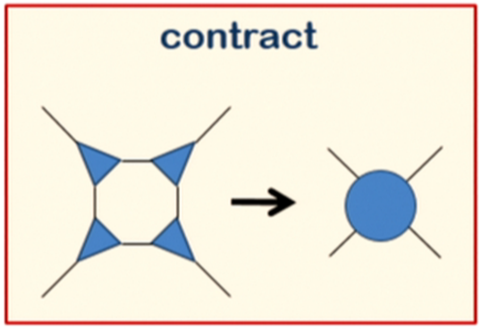
i.e. minimises: $\varepsilon = ||A - \tilde{A}||$

Tensor Renormalization Group (TRG)

(Levin, Nave, 2006)

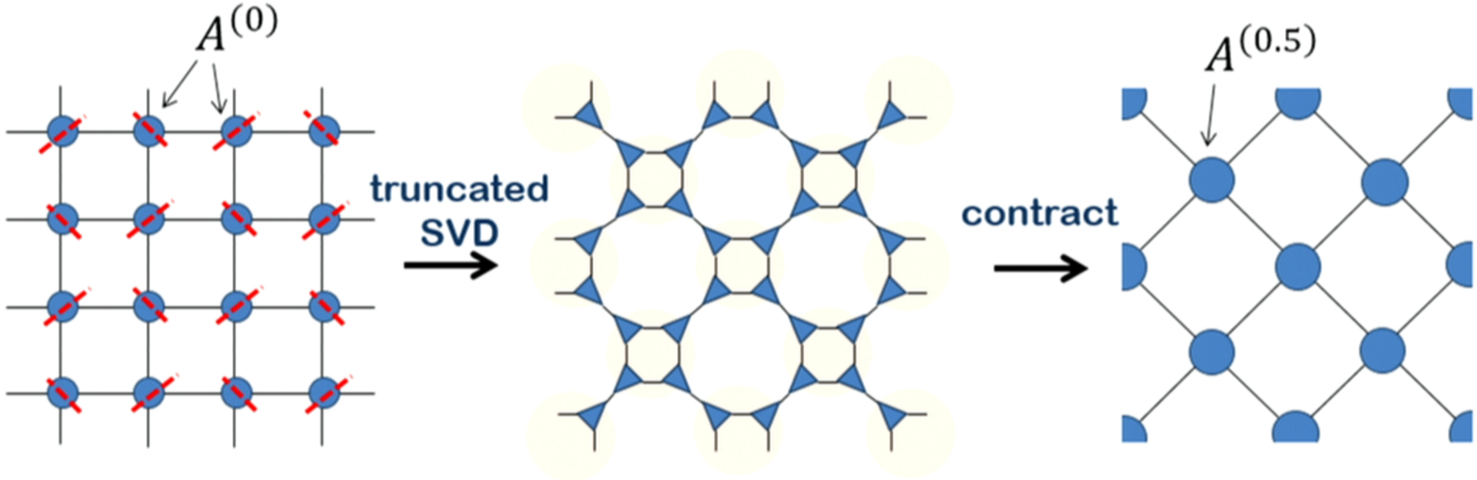






Tensor Renormalization Group (TRG)

(Levin, Nave, 2006)



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

TRG can be very powerful, but has significant flaws:

Conceptual flaw: TRG does not give proper RG flow

Computational flaw: TRG can not be iterated sustainably when at (or near) criticality

Tensor Renormalization Group (TRG) (Levin, Nave, 2006)

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

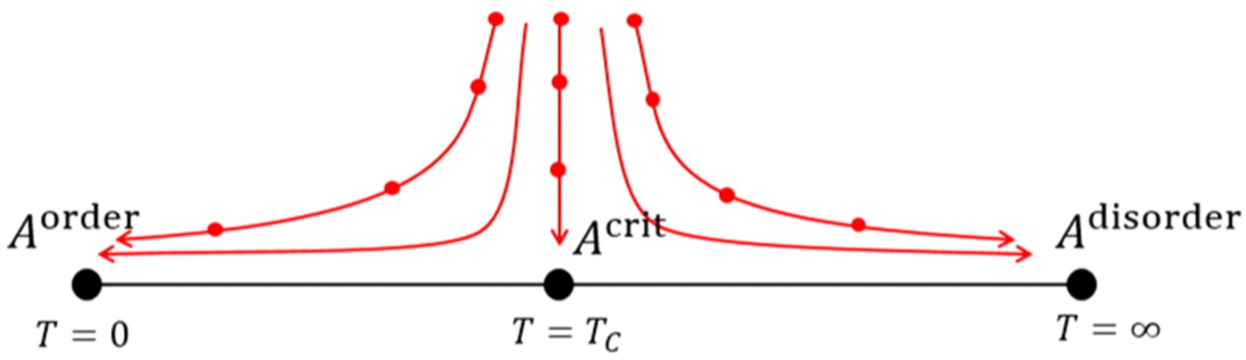
Conceptual flaw: TRG does not give proper RG flow

Consider TRG applied to the classical 2D Ising model:

Expect: The tensors should flow to one of three fixed point tensors, dependant on whether the temperature is below, at, or above the critical temperature

Find: Away from criticality, tensors in the same phase flow to different (temperature dependent) fixed points

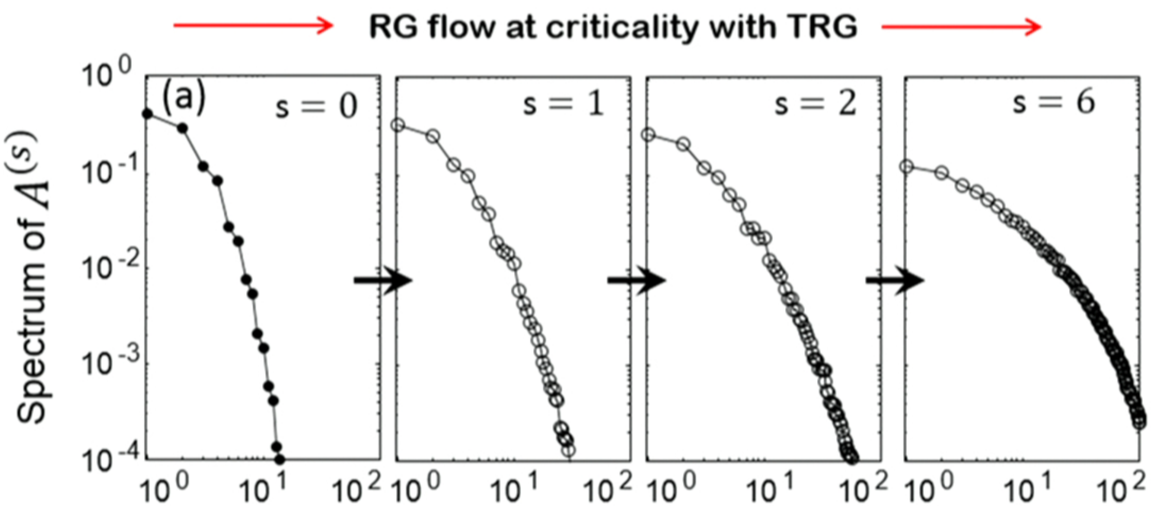
At criticality, tensors do NOT flow to a fixed point



Tensor Renormalization Group (TRG)

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

Computational flaw: TRG can not be iterated sustainably when at (or near) criticality

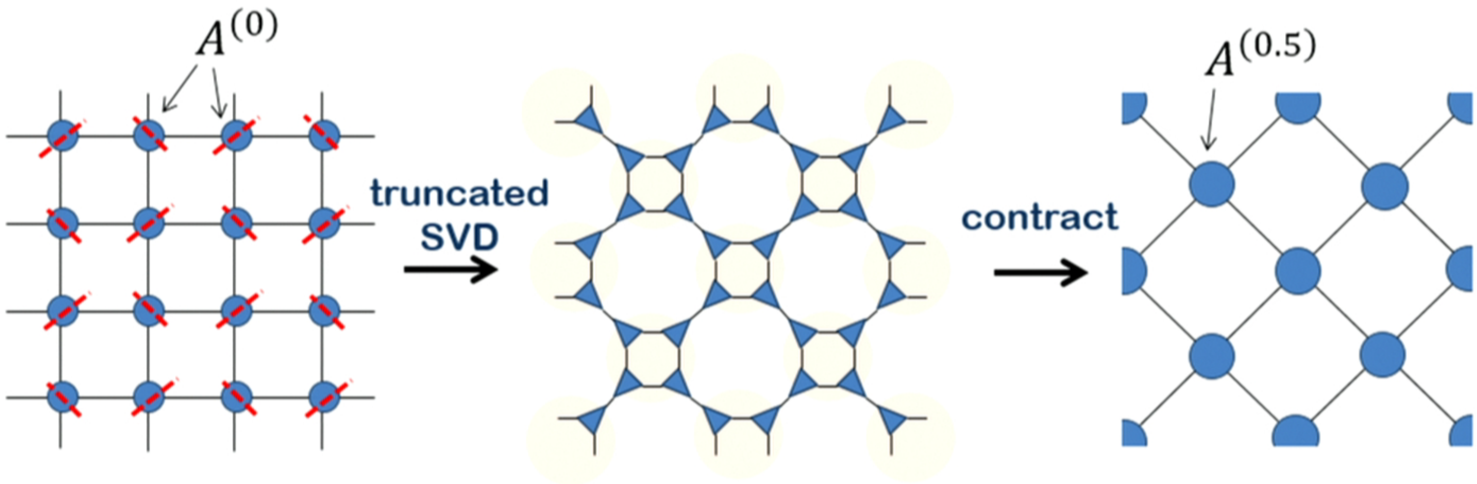


Bond dimension χ required to maintain fixed truncation error ($\sim 10^{-3}$): ~10 → ~20 → ~40 → >100

Cost of iteration, $O(\chi^5)$: $1 \times 10^5 \to 3 \times 10^6 \to 1 \times 10^8 \to > 10^{10}$

Tensor Renormalization Group (TRG)

(Levin, Nave, 2006)



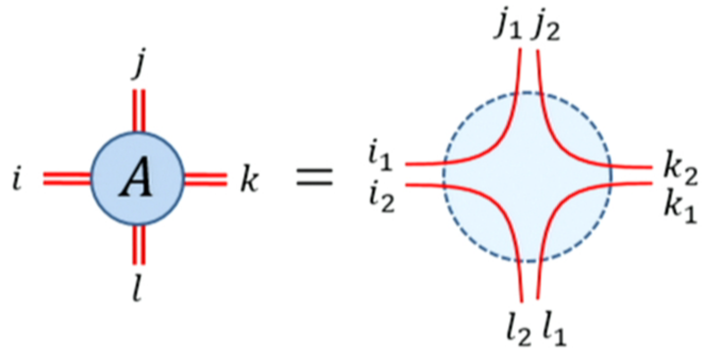
RG flow in the space of tensors: $A^{(0)} \rightarrow A^{(1)} \rightarrow A^{(2)} \rightarrow \cdots \rightarrow A^{(s)} \rightarrow \cdots$

TRG can be very powerful, but has significant flaws:

- **Conceptual flaw:** TRG does not give proper RG flow
- **Computational flaw:** TRG can not be iterated sustainably when at (or near) criticality

What is the origin of these flaws?

Fixed points of TRG



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

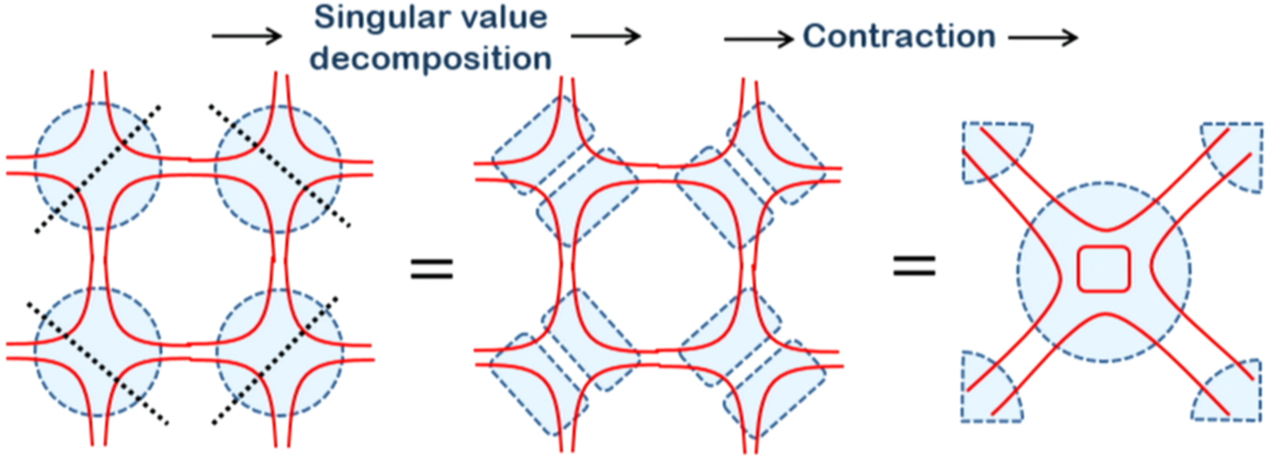
$$A_{ijkl} = A_{(i_1 i_2)(j_1 j_2)(k_1 k_2)(l_1 l_2)}$$

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.

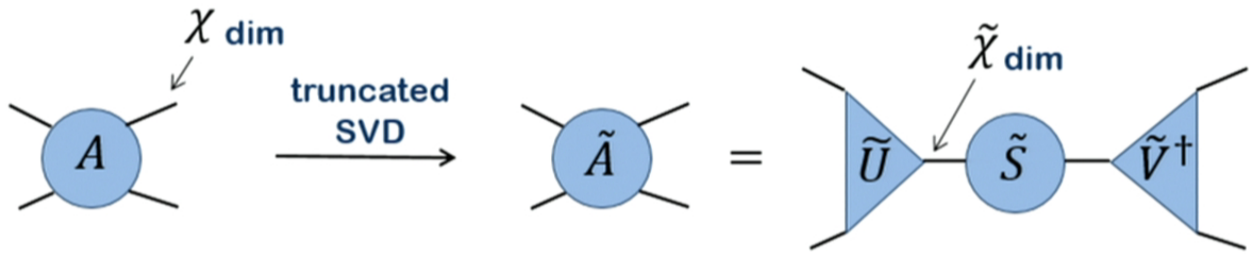
Fixed points of TRG

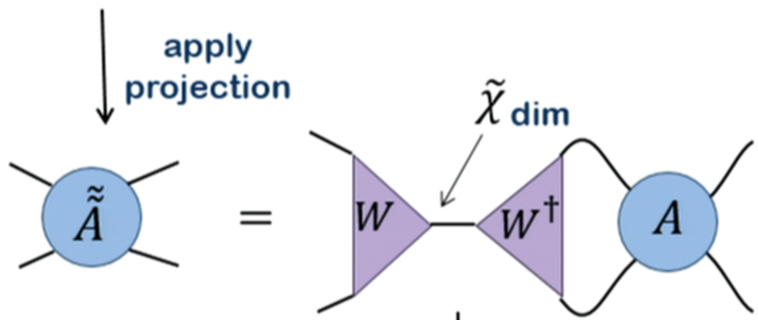


Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)

Change in formalism:

RG scheme based on SVD decompositions ⟷ RG scheme based on insertion of projectors into network

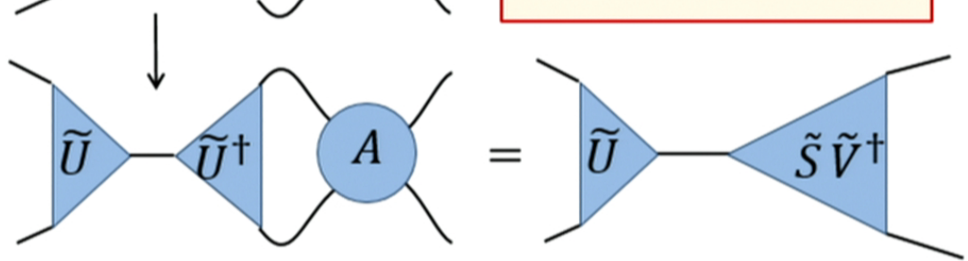




Want to choose 'w' as to minimize error:

$$\varepsilon = \|A - \tilde{\tilde{A}}\|$$

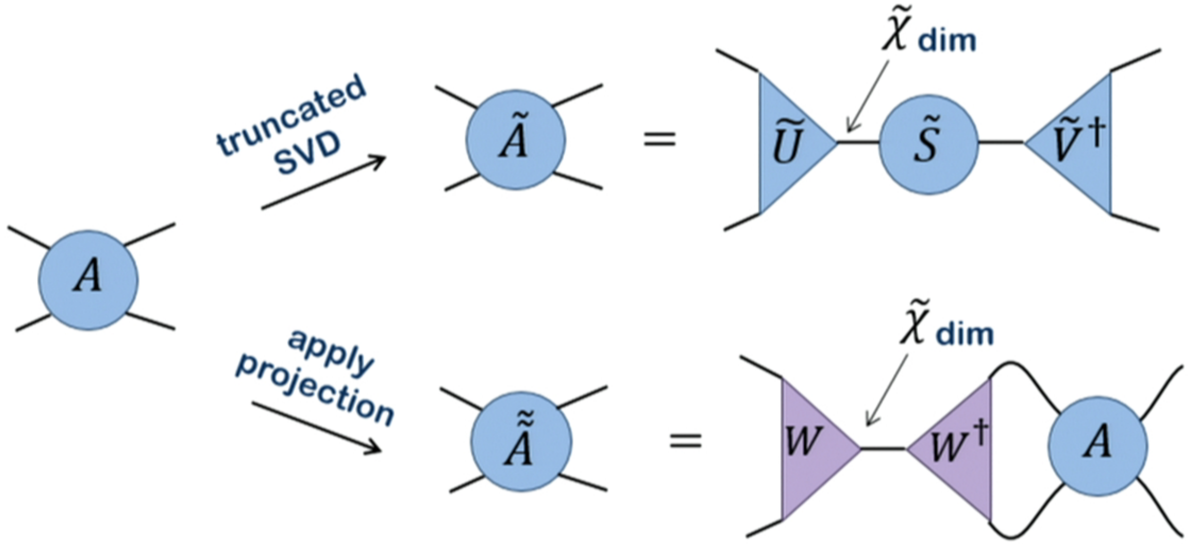
set: $W = \tilde{U}$



Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)

Change in formalism:

RG scheme based on SVD decompositions ⟷ RG scheme based on insertion of projectors into network



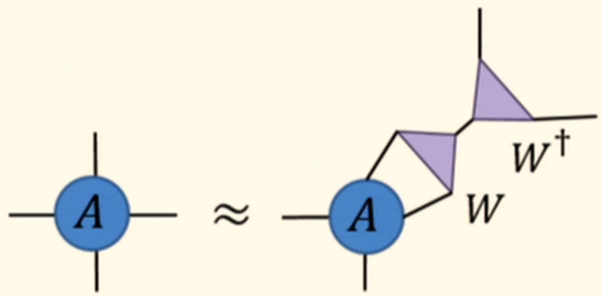
if isometry 'w' is optimised to act as an approximate resolution of the identity, then these two procedures are equivalent

Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)

Two key ingredients for TNR:

(1)

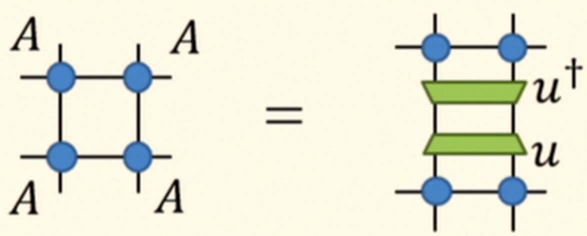
insertion of projectors:



Can mimic the effect of truncated SVD

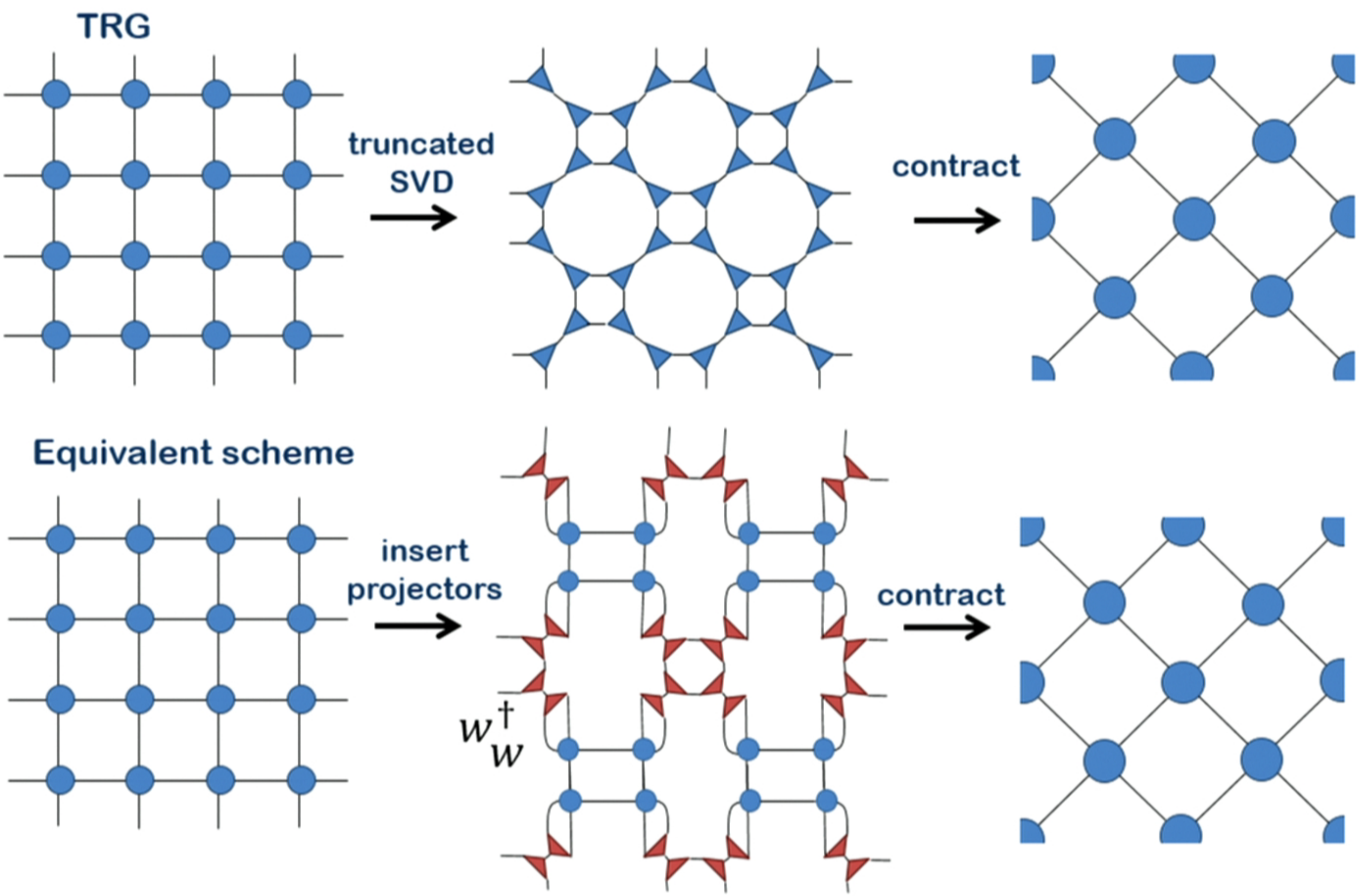
(2)

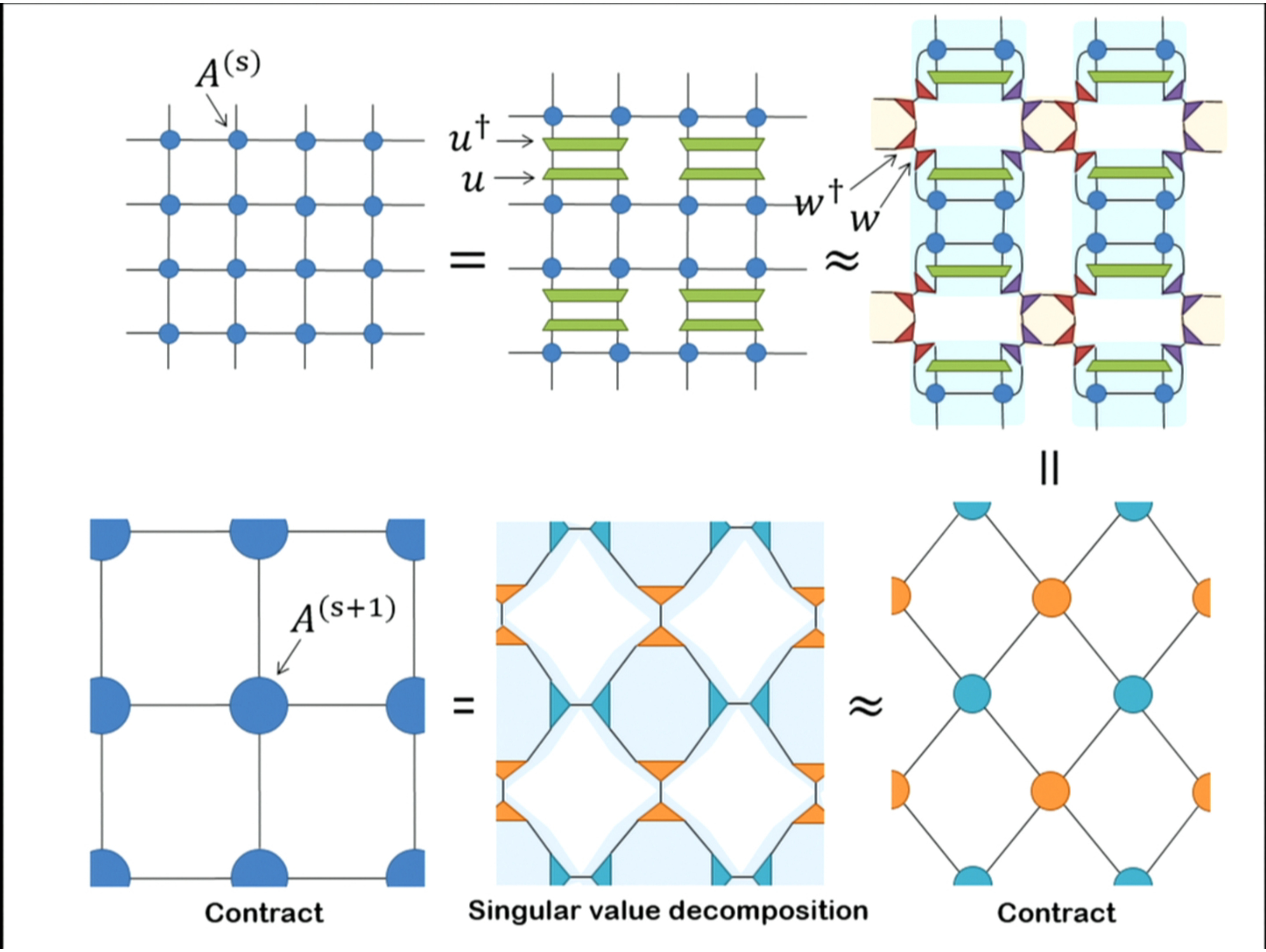
insertion of unitaries:



act as exact resolution of identity

Tensor Network Renormalization (TNR) (Evenbly, Vidal, *in prep*)





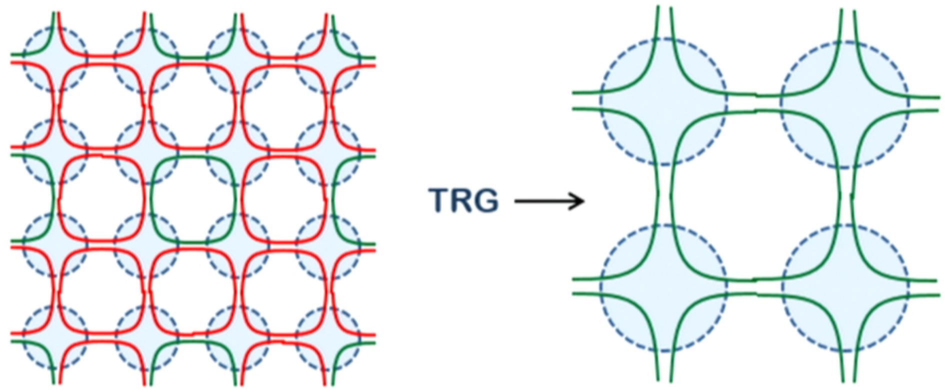
$A^{(s)}$
$u^\dagger$
u
$w^\dagger$
w
$A^{(s+1)}$
Contract
Singular value decomposition
Contract

Tensor Network Renormalization (TNR):

How does disentangling help?

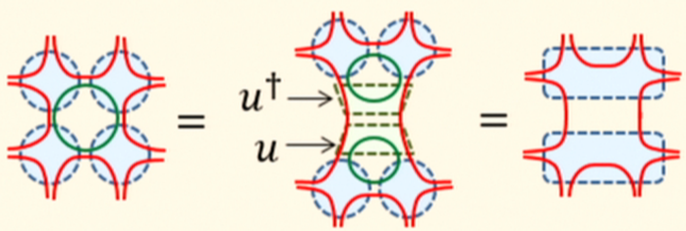
Consider CDL tensors…

short-range correlated   short-range correlated

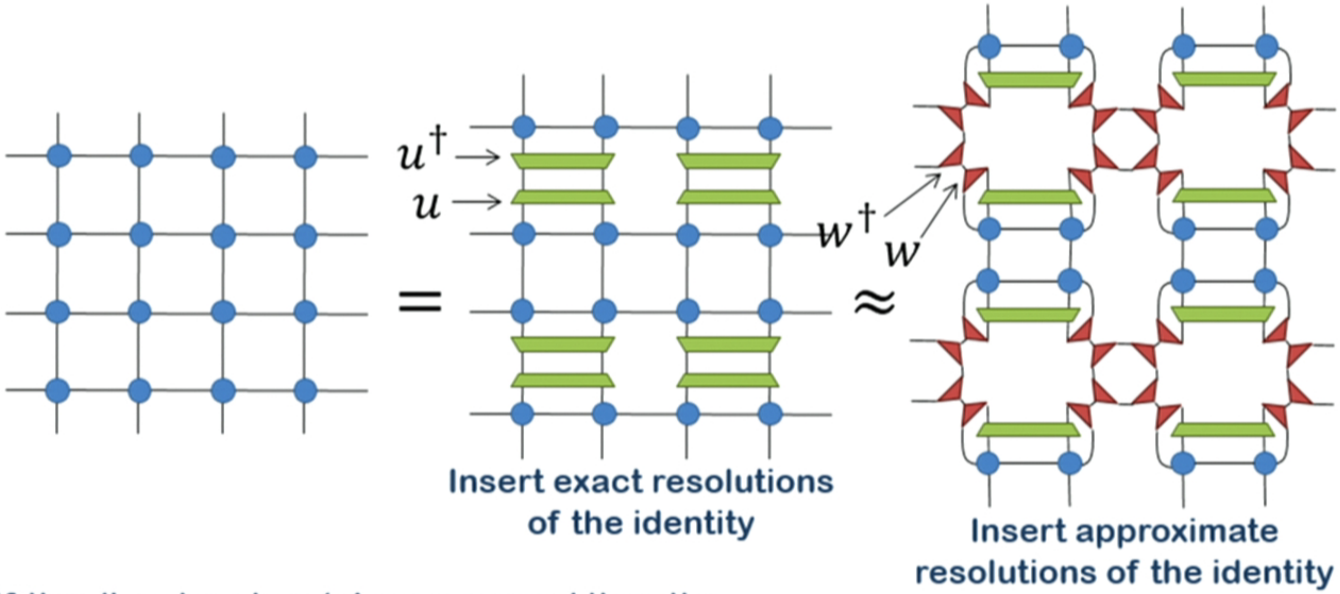


Key step of TNR algorithm:

Insert unitary disentanglers:



Tensor Network Renormalization (TNR):



- If the disentanglers 'u' are removed then the TNR approach becomes equivalent to TRG
- I will not here discuss the numeric algorithm required to optimize disentanglers 'u' and isometries 'w'

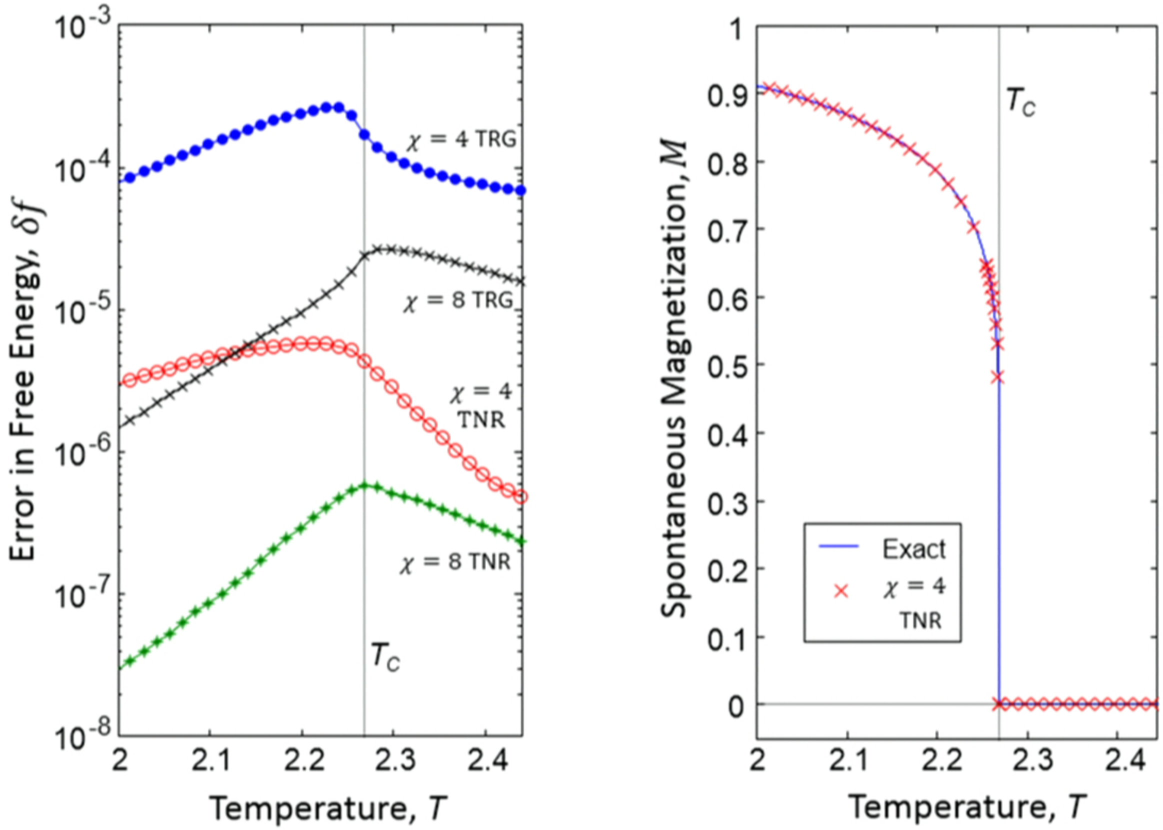
Does TNR fix the flaws of previous RG schemes?

Conceptually: want correct RG fixed points

Computationally: want sustainable RG flow

Benchmark numerics:

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



Benchmark numerics: RG flow of Ising model

Conceptual goal: Does TNR give proper structure of fixed points? **Yes!**

$$\left|\tilde{A}^{(0)}\right| \rightarrow \left|\tilde{A}^{(1)}\right| \rightarrow \left|\tilde{A}^{(2)}\right| \rightarrow \left|\tilde{A}^{(4)}\right|$$

sub-critical

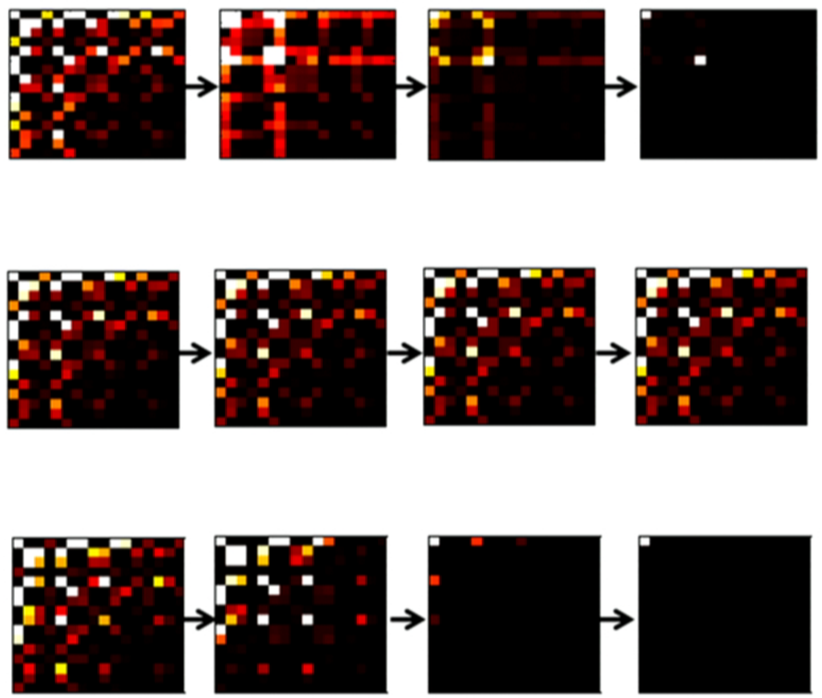
$T = 0.9\, T_c$

critical

$T = T_c$

super-critical

$T = 1.1\, T_c$



ordered fixed point

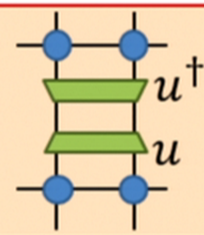
critical (scale-invariant) fixed point

disordered (trivial) fixed point

Summary

We have introduced an RG based method for contracting tensor networks: **Tensor Network Renormalization (TNR)**

key idea: use of unitary disentanglers to properly address all short-ranged degrees of freedom at each RG step



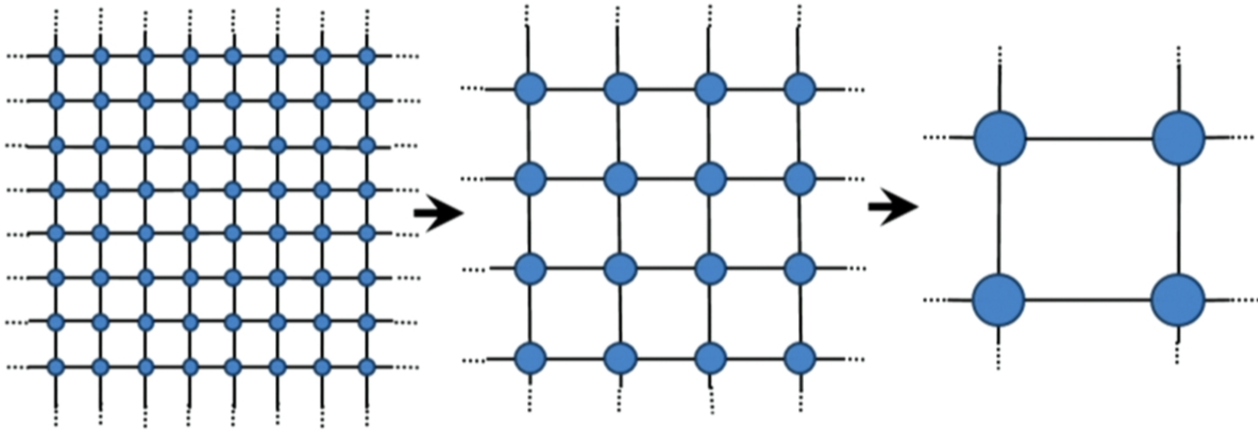
key features of TNR:

- Proper RG flow (gives correct RG fixed points)
- Sustainable RG flow (can iterate without increase in cost)

Direct applications to study of 2D classical and 1D quantum many-body systems, and for contraction of PEPS.

The same ideas can be implemented for higher dimensional tensors networks forming e.g. a simulation algorithm for 2D quantum many-body systems.

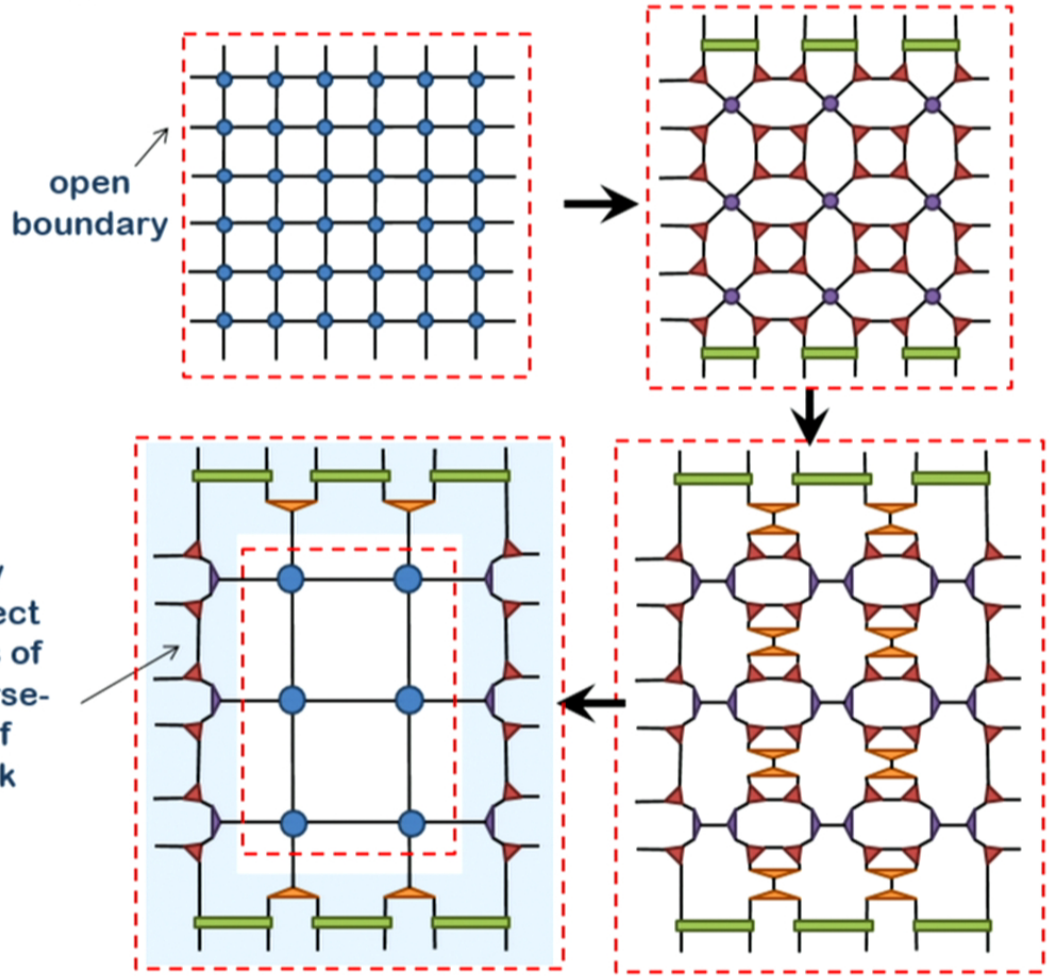
Other Applications



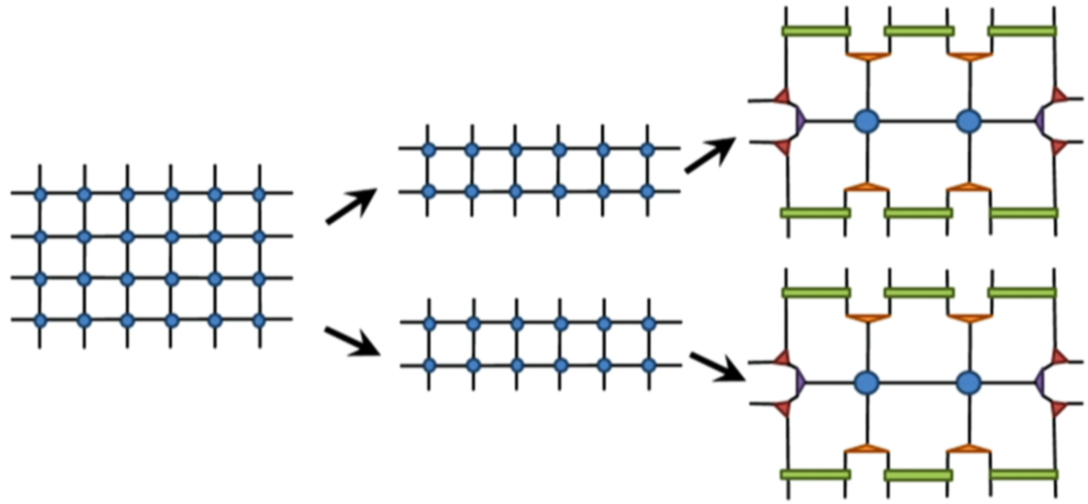
Thus far, we have considered RG of infinite tensor networks (or PBC)

What about TNR applied to tensor networks with open boundaries?

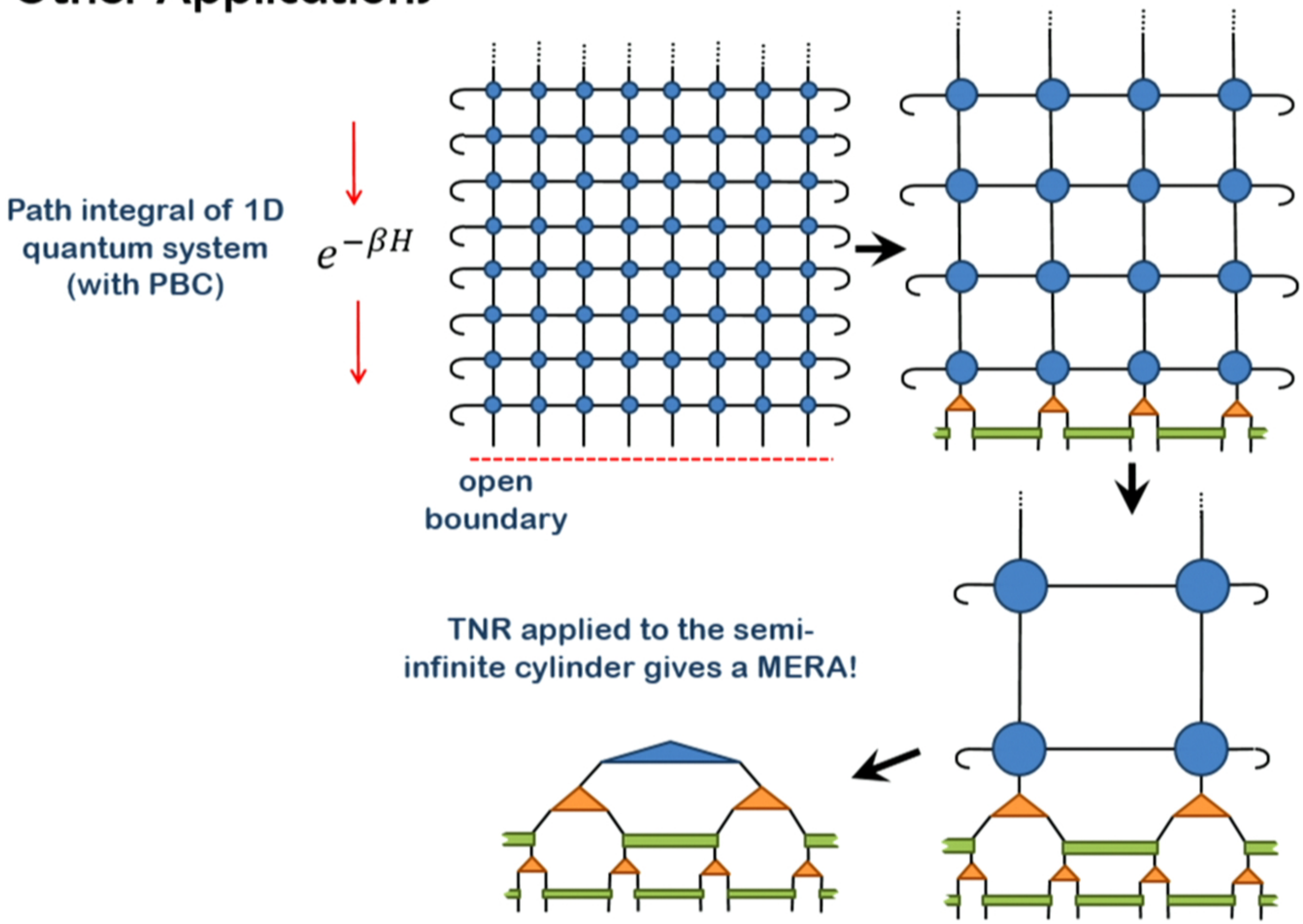
Other Applications



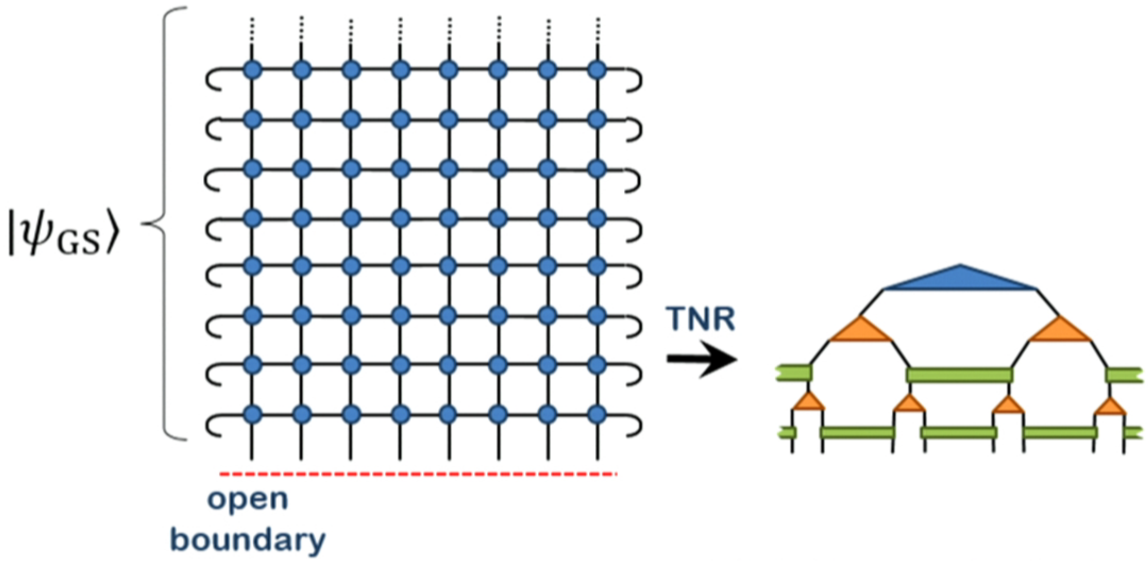
Other Applications



Other Applications



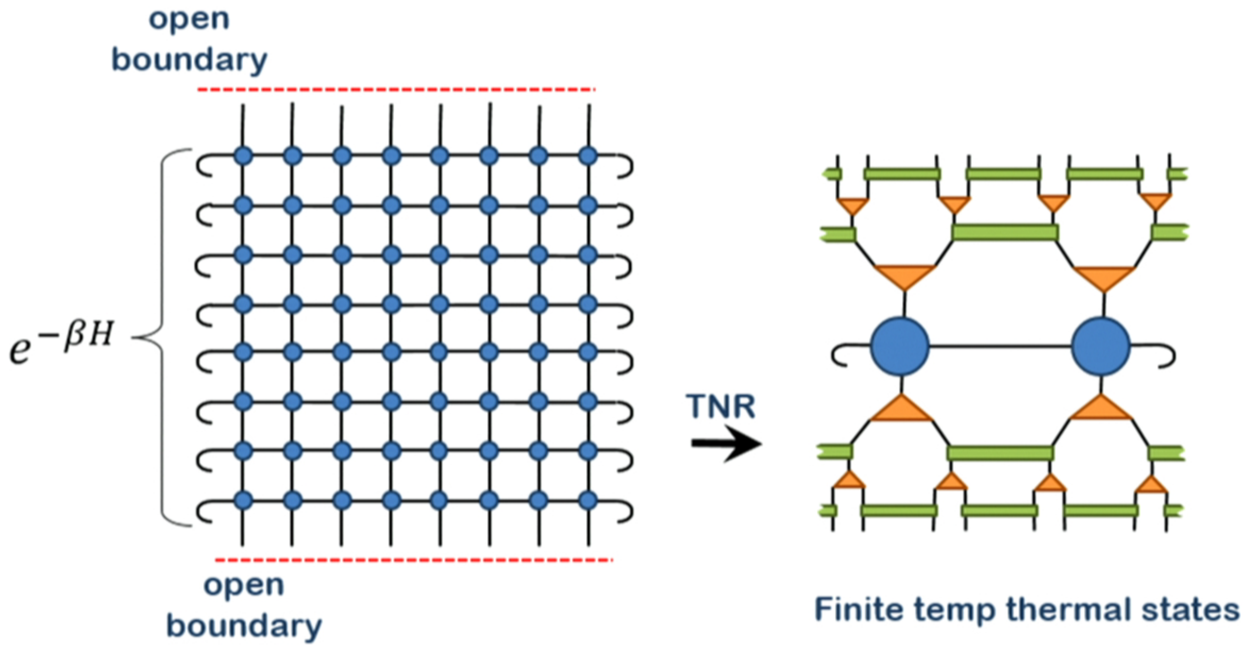
Other Applications



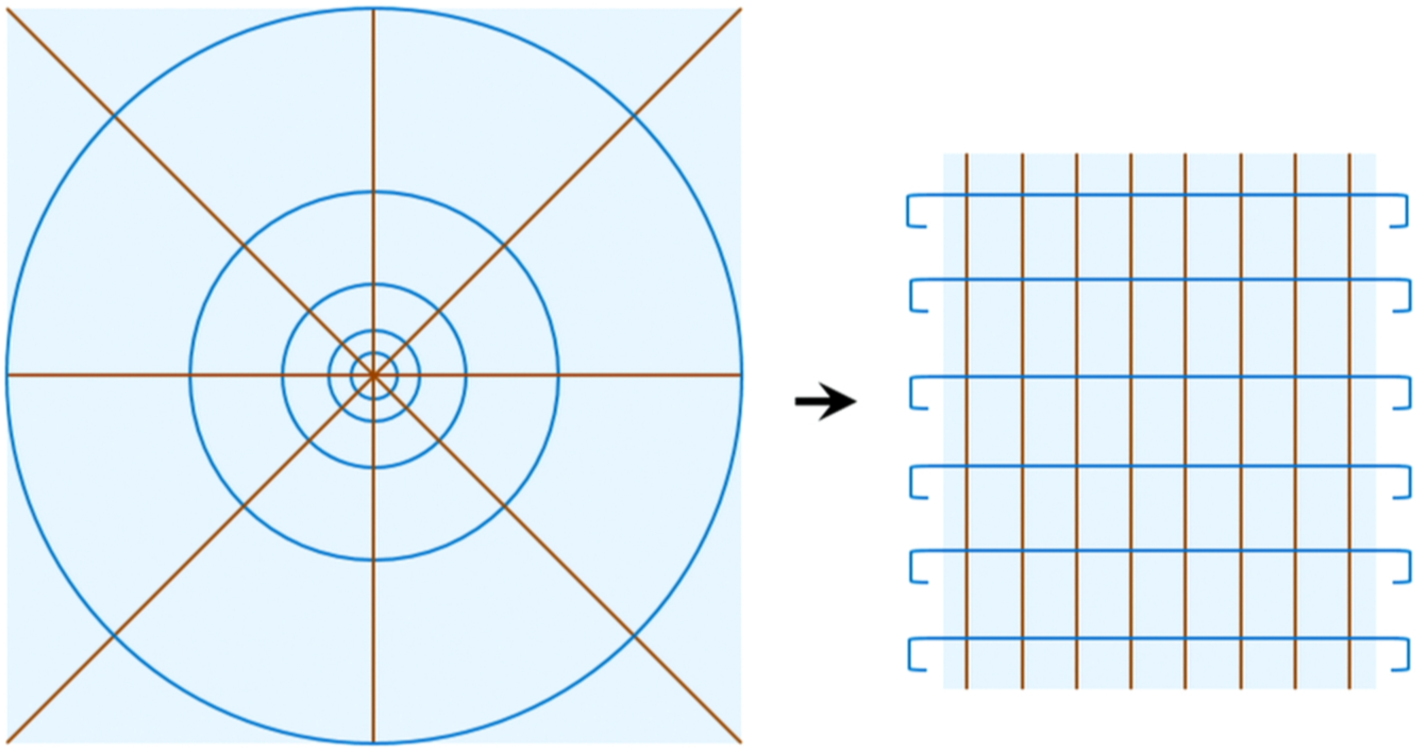
TNR allows one to obtain a MERA approximation to the ground state directly from the path integral

MERA is emergent from TNR

Other Applications

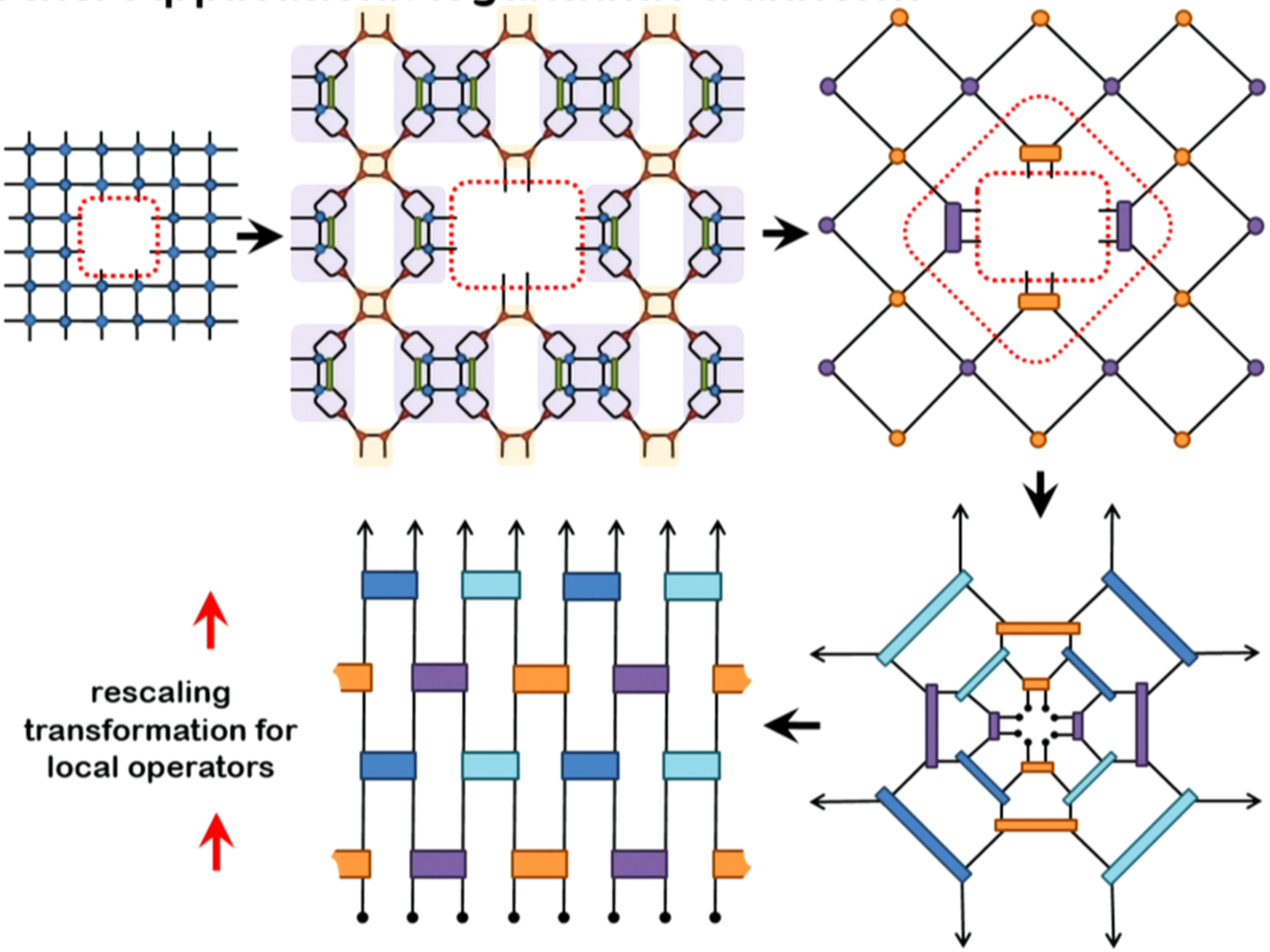


Other Applications: logarithmic transform

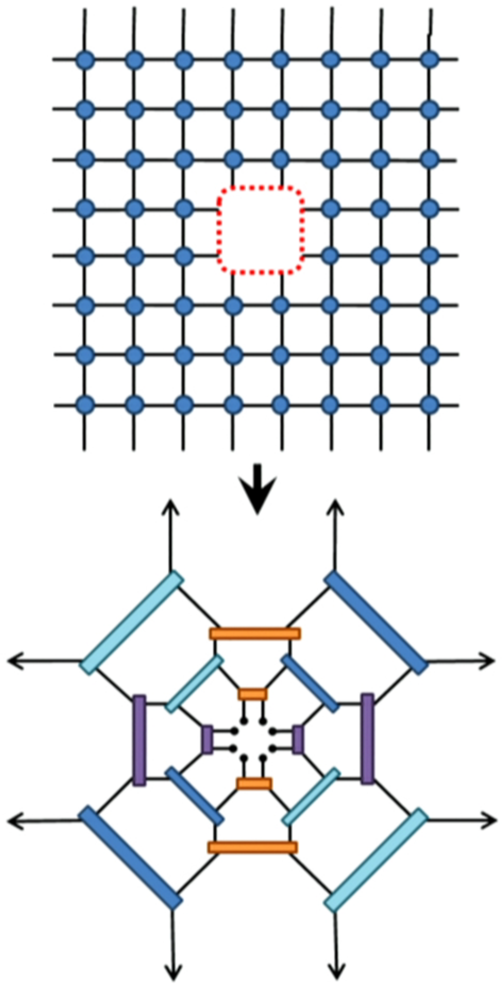


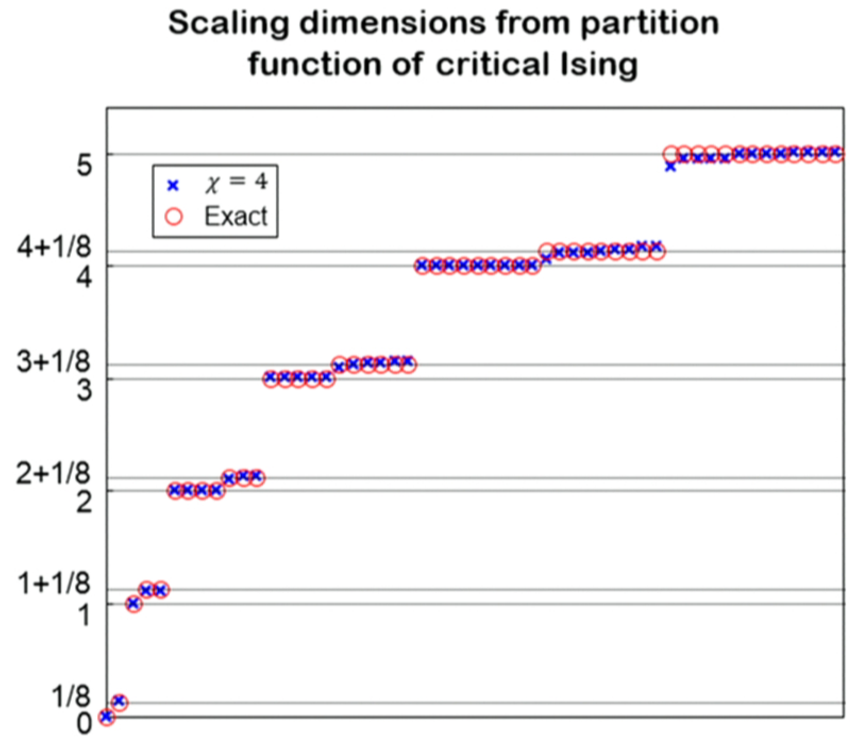
2D Conformal field theory

Other Applications: logarithmic transform



Other Applications: logarithmic transform

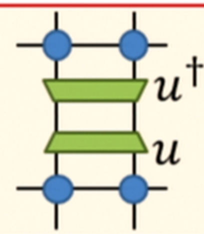




Summary

We have introduced an RG based method for contracting tensor networks: **Tensor Network Renormalization (TNR)**

key idea: use of unitary disentanglers to properly address all short-ranged degrees of freedom at each RG step



key features of TNR:

- Proper RG flow (gives correct RG fixed points)
- Sustainable RG flow (can iterate without increase in cost)

Direct applications to study of 2D classical and 1D quantum many-body systems, and for contraction of PEPS.

The same ideas can be implemented for higher dimensional tensors networks forming e.g. a simulation algorithm for 2D quantum many-body systems.