Welcome to the ITensor CCQ School

FLATIRON
INSTITUTE

Goals of the School

- Fundamental theory (tensor networks, algorithms)

- Understanding strengths and weaknesses of TN

- Learning the ITensor software – hands on experience

Goals of the School

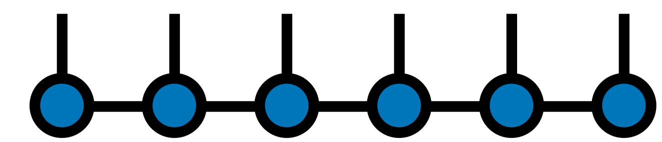
This is for <u>you</u>!

Please ask questions if anything is unclear…

Schedule

Day One:

Talk One: DMRG Overview
Talk Two: ITensors.jl and running DMRG
Hands-on: Using and applying DMRG



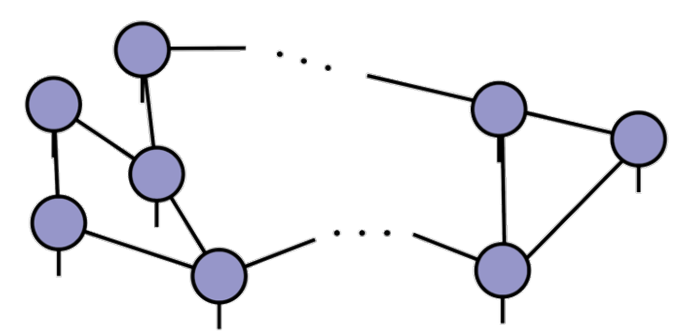
Day Two:

Talk Three: Time evolution with MPS
Talk Four: ITensors.jl under the hood: tensor algebra
Hands-on: Time Evolution of MPS

$$e^{-\mathrm{i}Ht}\,|\,\psi(0)\rangle = |\,\psi(t)\rangle$$

Day Three:

Talk Five: Quantics and MPS beyond wave functions
Talk Six: Tensor networks in higher dimensions
Hands-on: Belief Propagation on 2D Tensor Networks



This Talk

- Tensors
- Tensor Networks
- The DMRG algorithm

Tensors
and
Tensor Networks

Is exponential compression possible?

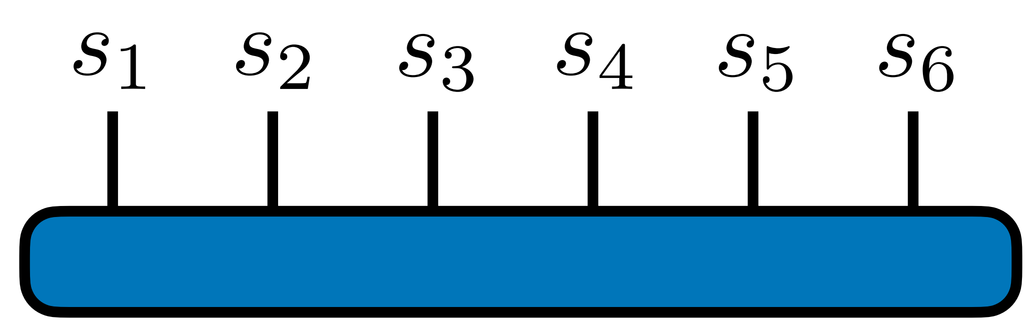
Tensor Networks

General wavefunction of n qubits

$$|\Psi\rangle = \sum_{s_1 s_2 s_3 \cdots s_n} \Psi^{s_1 s_2 s_3 \cdots s_n} |s_1 s_2 s_3 \cdots s_n\rangle \qquad s_j \in 0, 1$$

Amplitudes form a big tensor!

$$\Psi^{s_1 s_2 s_3 s_4 s_5 s_6} =$$



Tensor Networks

What is a tensor?

vector $$v = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \qquad v_2 = 3$$

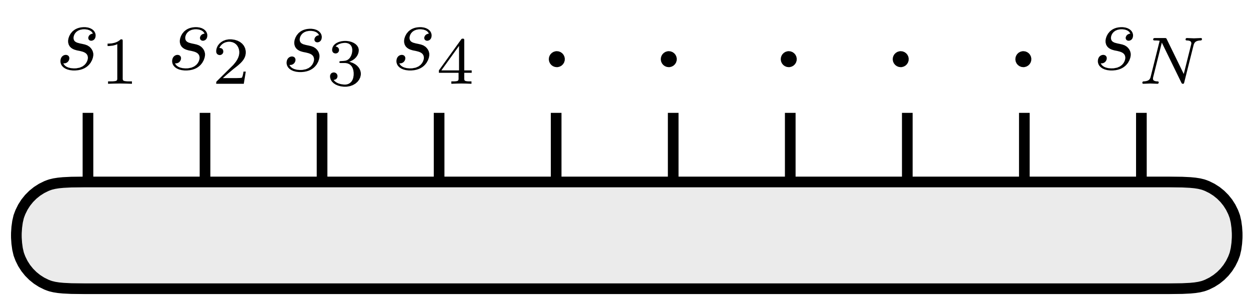
matrix $$M = \begin{bmatrix} 5 & 7 \\ 8 & 9 \end{bmatrix} \qquad M_{12} = 7$$

order-3 tensor $$T = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} {\color{gray}5} & 7 \\ {\color{gray}3} & 5 \end{bmatrix} \qquad T_{112} = 5$$

Tensor Networks

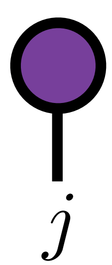
N-index tensor = shape with N lines

$$T^{s_1 s_2 s_3 \cdots s_N} =$$

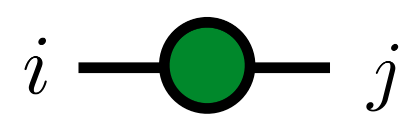


Low-order examples:

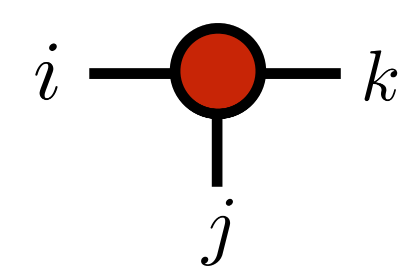
v_j



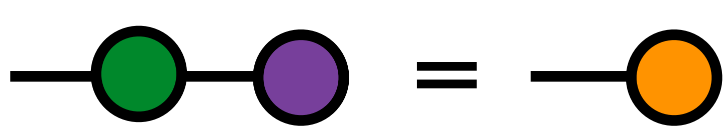
M_{ij}



T_{ijk}



Joining wires means contraction:

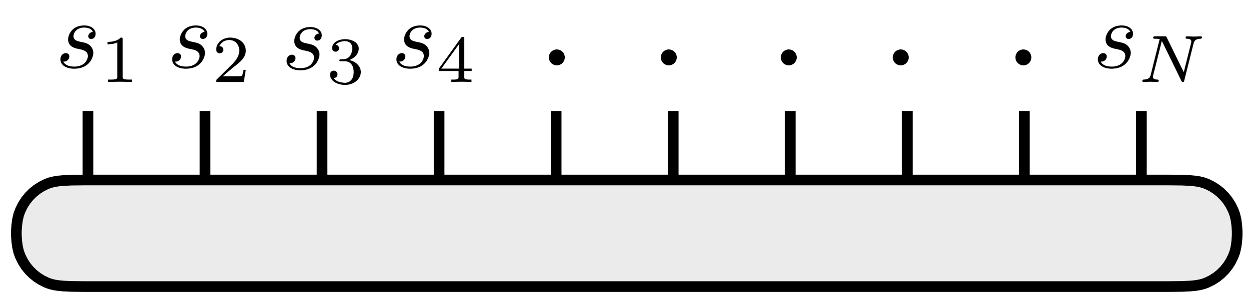


$$\sum_j M_{ij} v_j = w_i$$

Tensor Networks

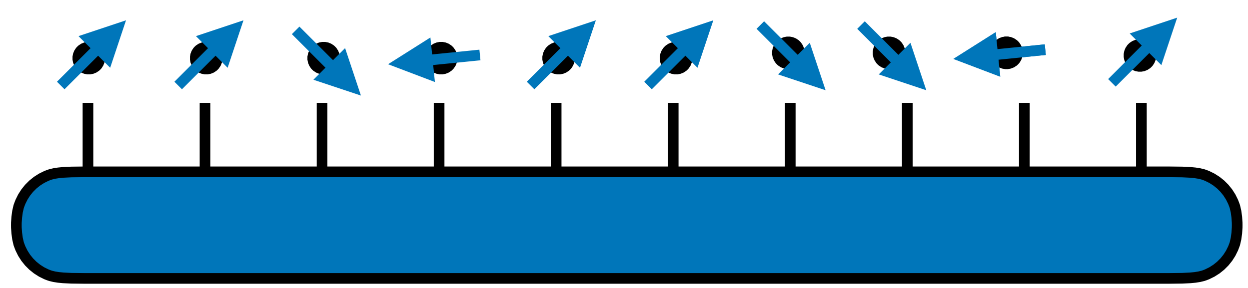
A problem: N-index tensor exponential to store

$$T^{s_1 s_2 s_3 \cdots s_N} =$$



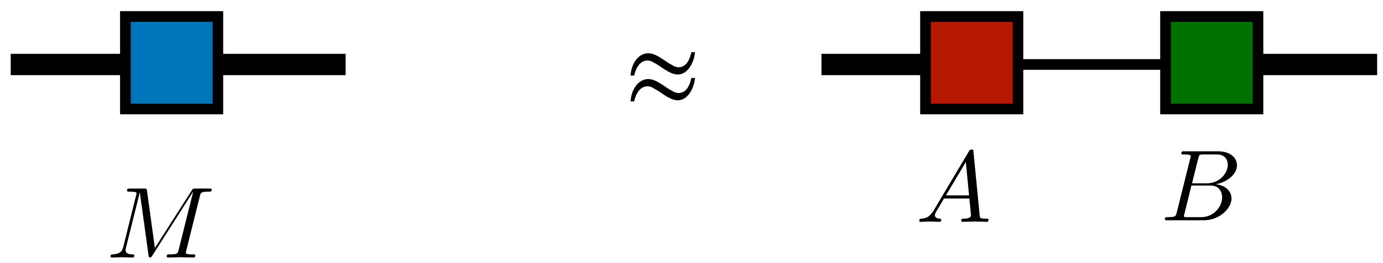
A version of the "many-body problem"

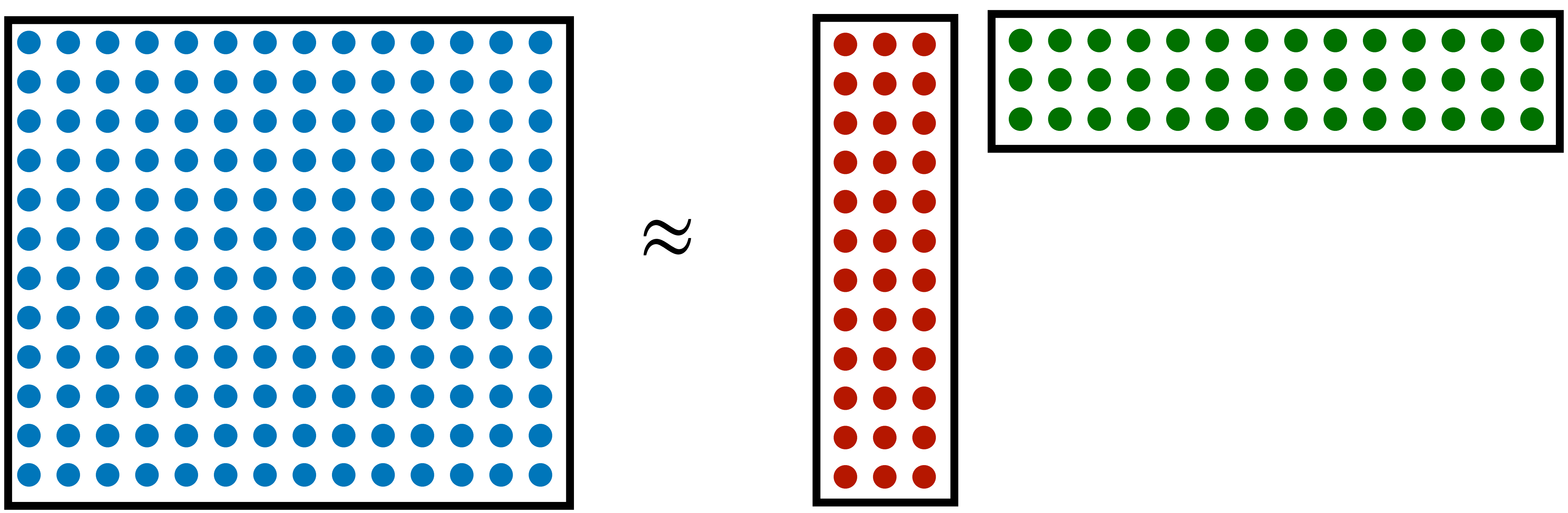
$$|\Psi\rangle =$$



Tensor Networks

For matrices, can perform low-rank compression

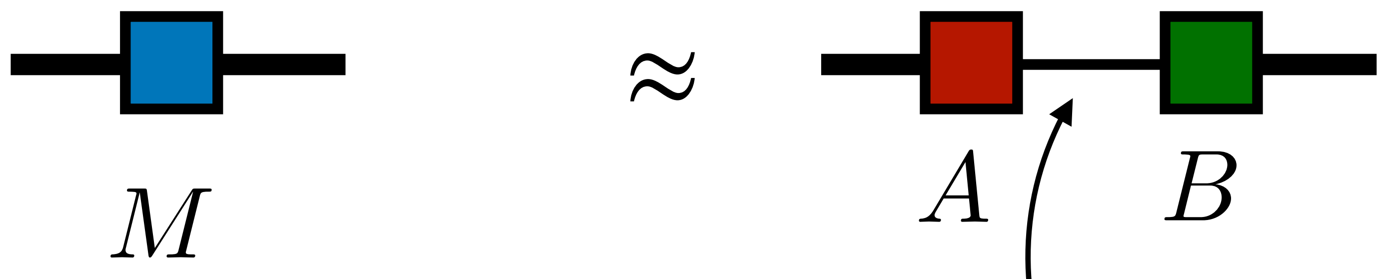




Less memory and fewer operations

Tensor Networks

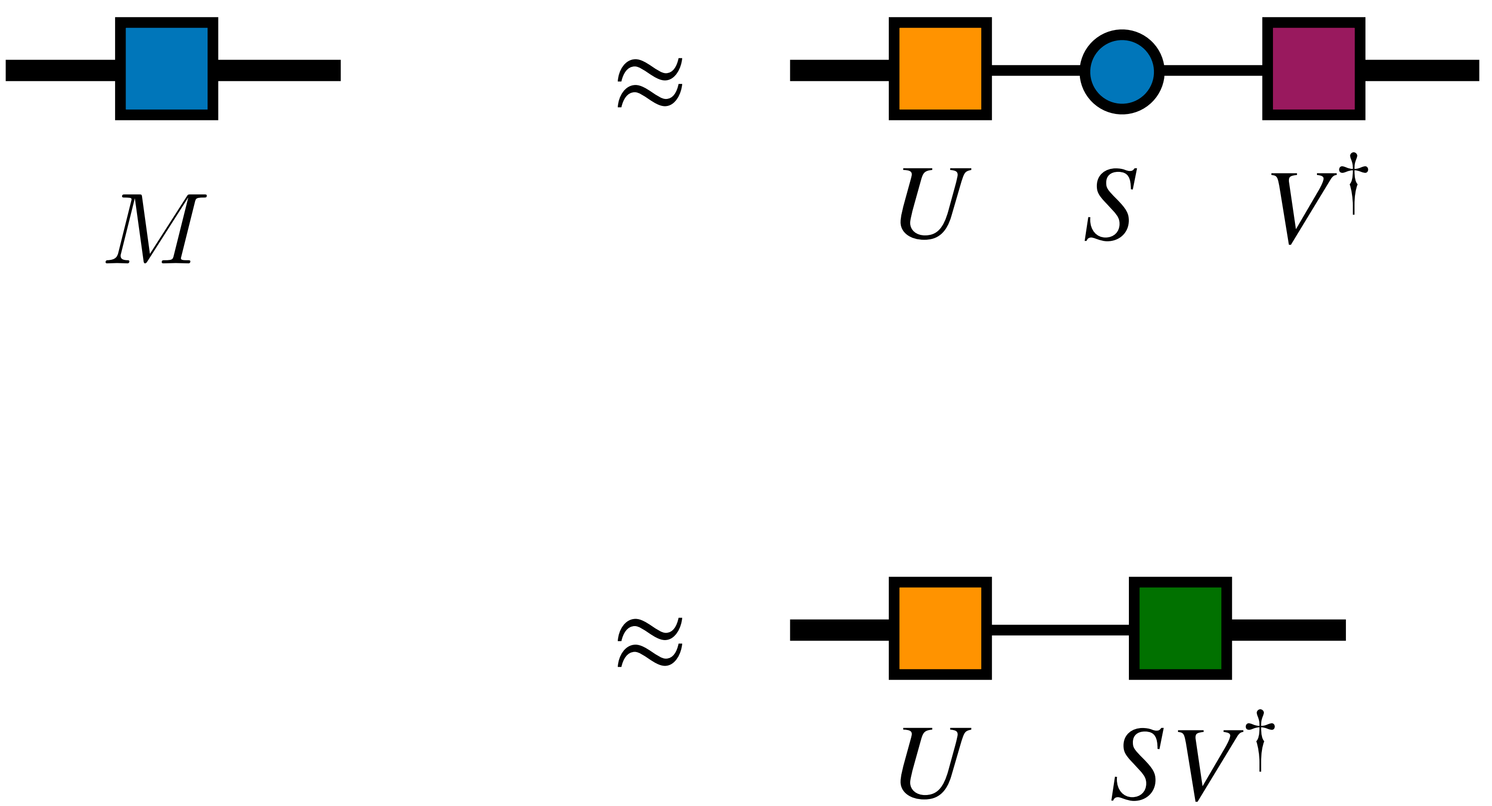
For matrices, can perform low-rank compression



size of shared index called the "__rank__" of the factorization

Tensor Networks

Optimal solution (smallest rank) found by SVD



Low-rank: Test Your Knowledge

Is this matrix low rank? Or high rank?
Why?

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Low-rank: Test Your Knowledge

Is this matrix low rank? Or high rank?
Why?

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

product of Nx1 and 1xN matrices

Low rank
Columns linearly dependent

Low-rank: Test Your Knowledge

Is this matrix low rank? Or high rank?
Why?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Low-rank: Test Your Knowledge

Is this matrix low rank? Or high rank?
Why?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

Low rank
Outer product of two vectors

Low-rank: Test Your Knowledge

Is this matrix low rank? Or high rank?
Why?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Low-rank: Test Your Knowledge

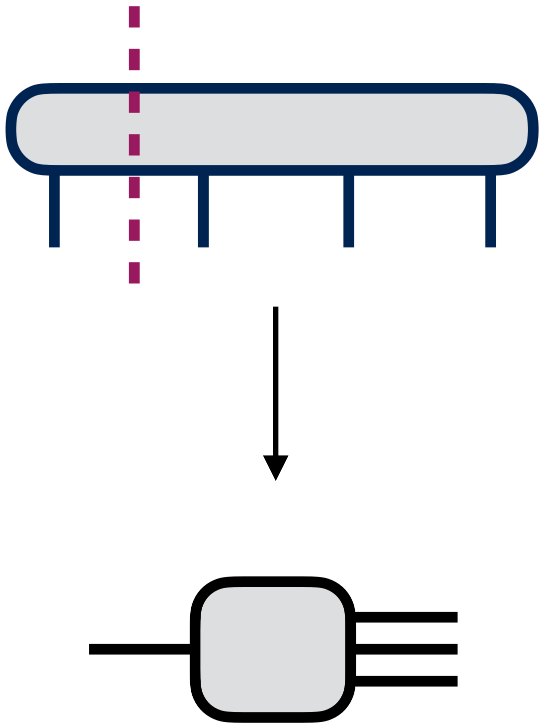
Is this matrix low rank? Or high rank?
Why?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

High rank (full rank)
All columns orthogonal / maximally linearly independent

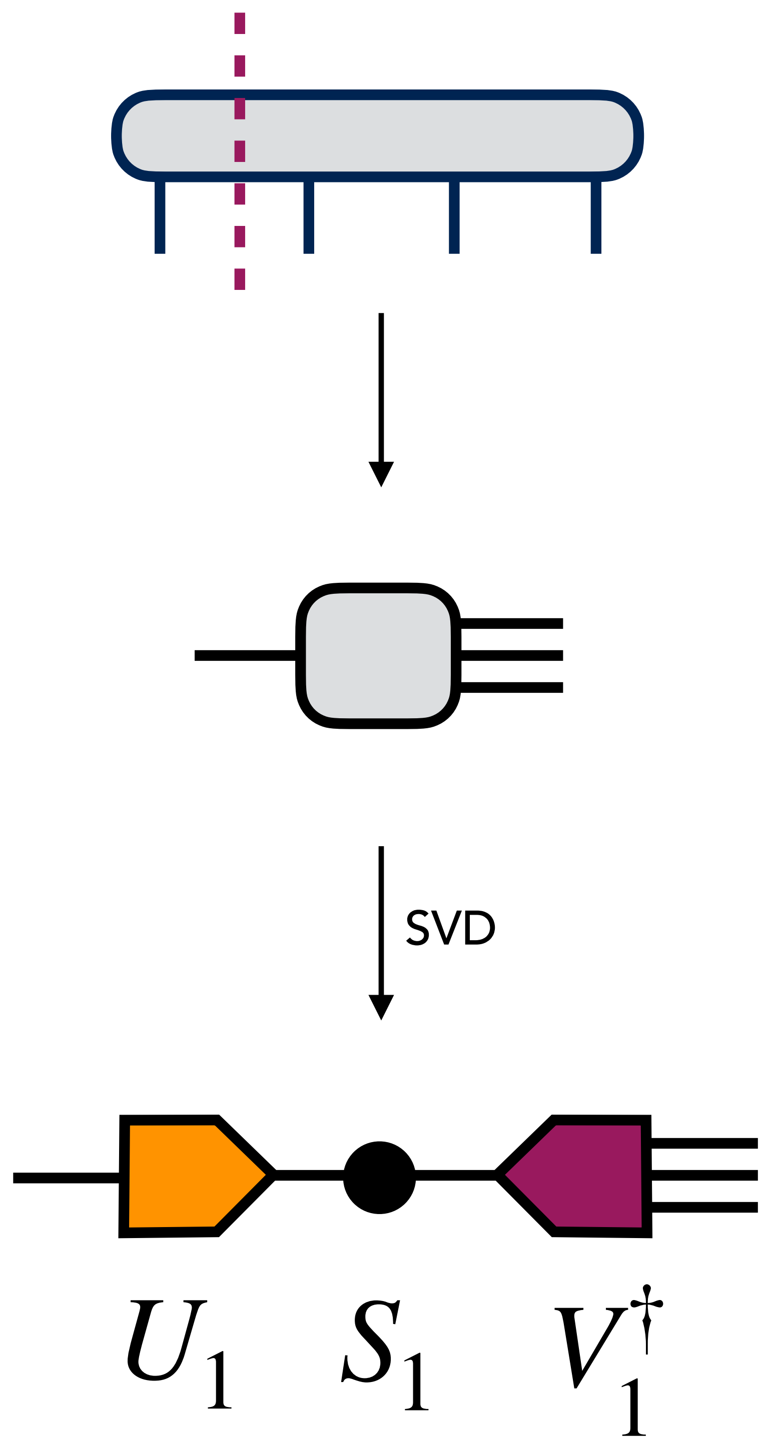
How to generalize SVD to tensors?

Reshape as a matrix:



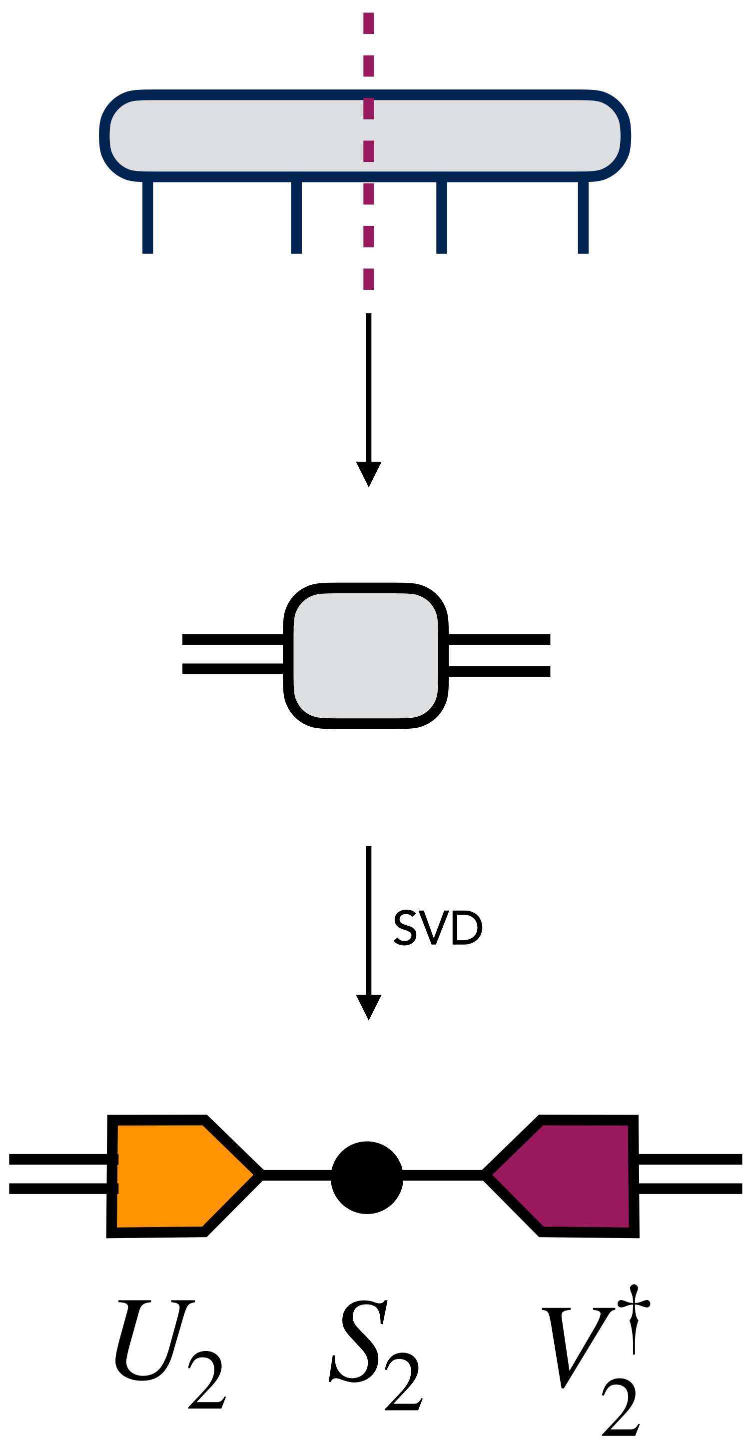
Generalizing SVD to tensors

Reshape as a matrix:



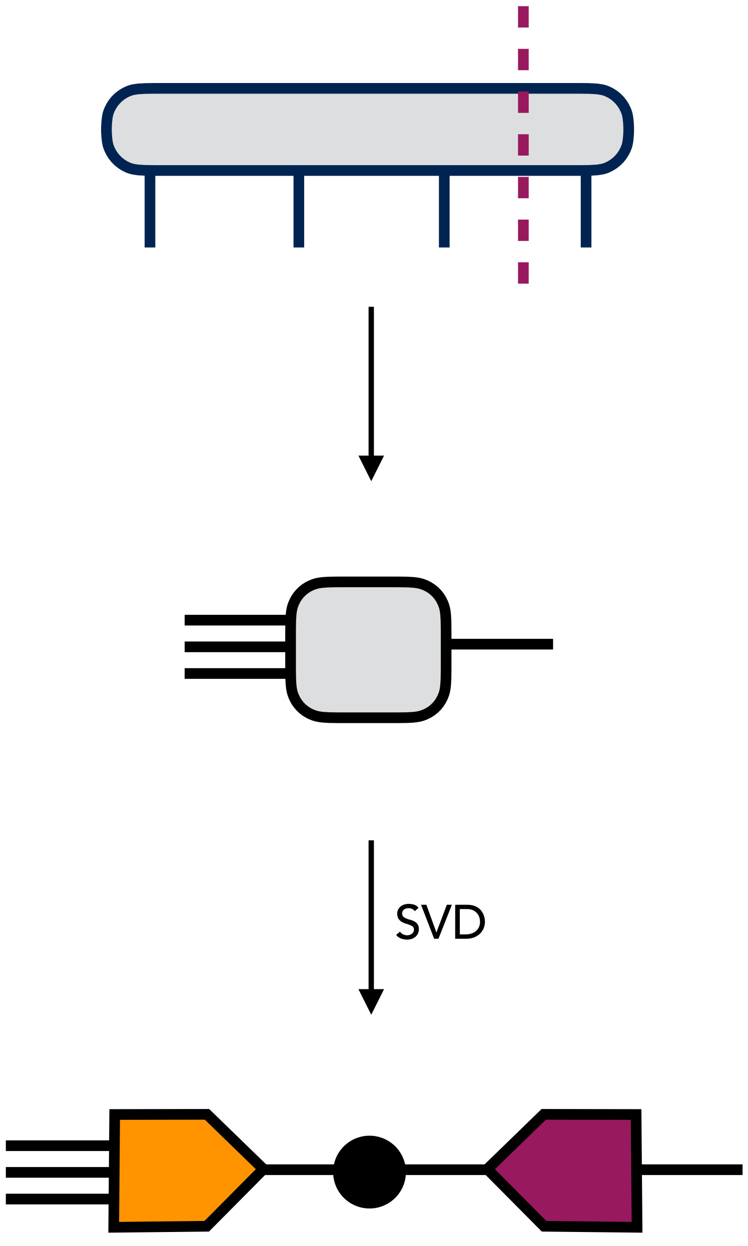
Generalizing SVD to tensors

Other partitions:



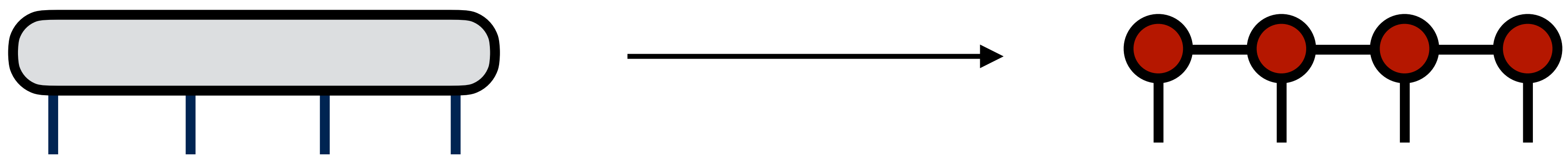
Generalizing SVD to tensors

Other partitions:



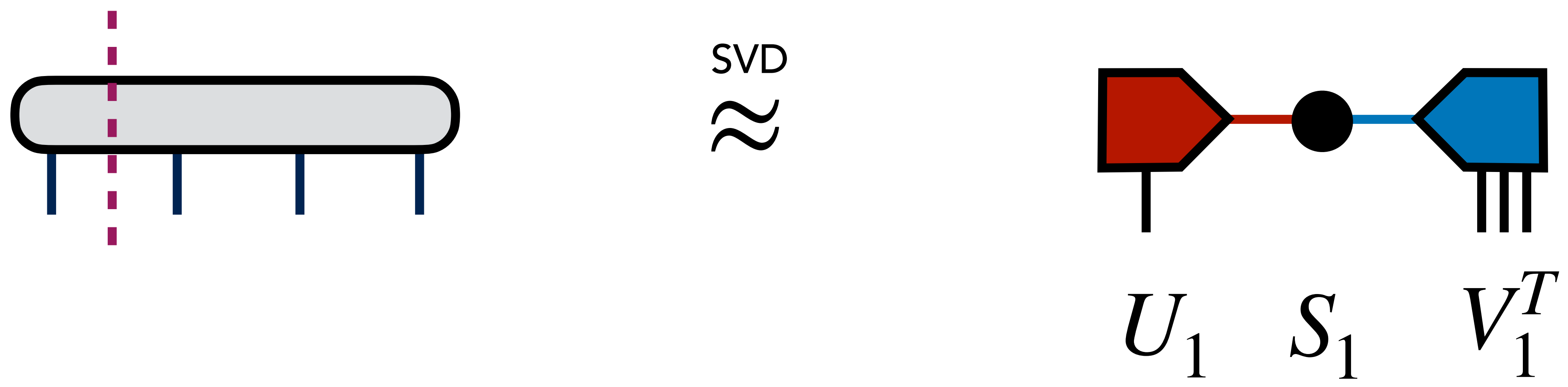
U_3 S_3 $V_3^\dagger$

Factorizations lead to tensor networks

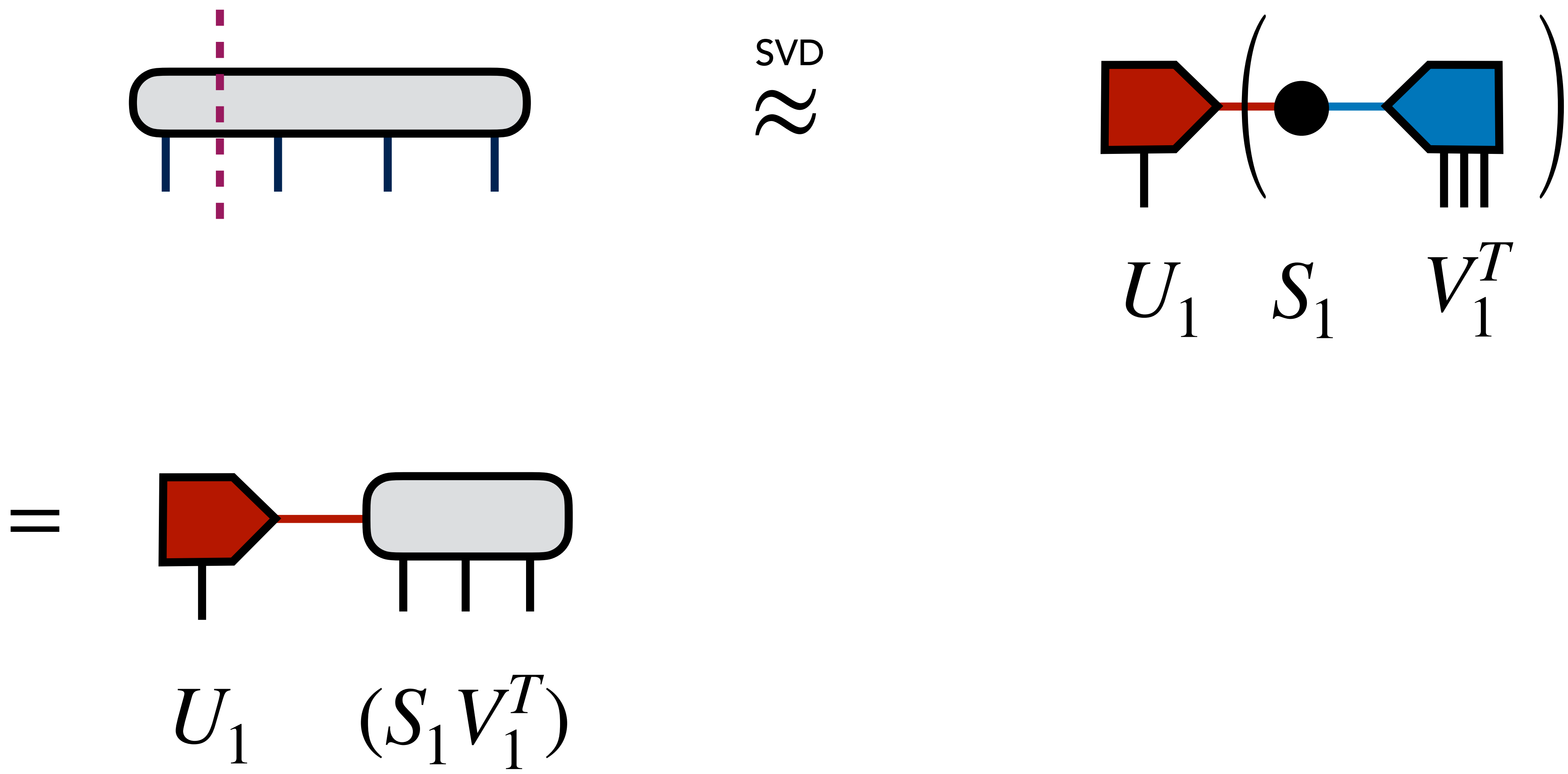


Consider sequence of SVD's...

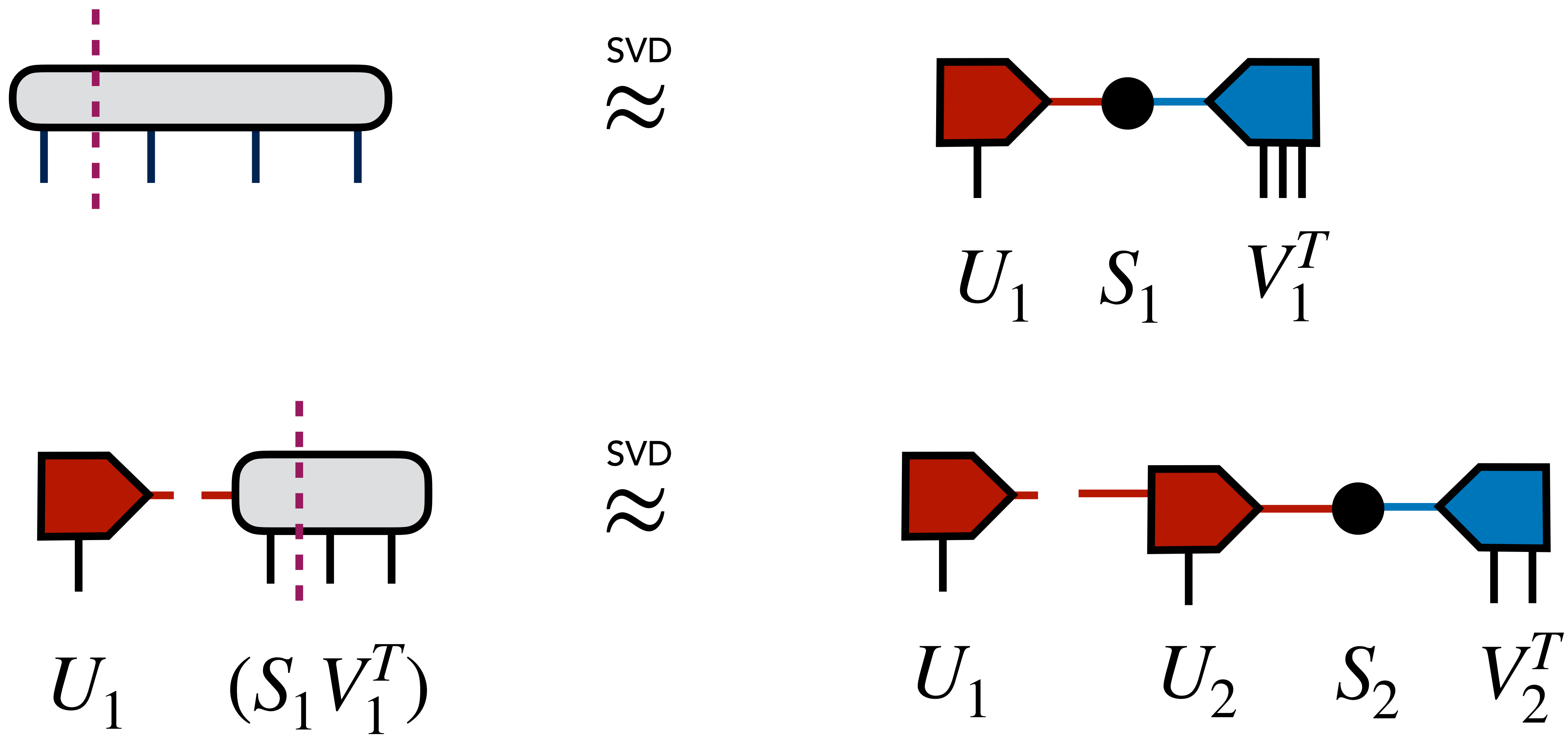
1. SVD first index from rest



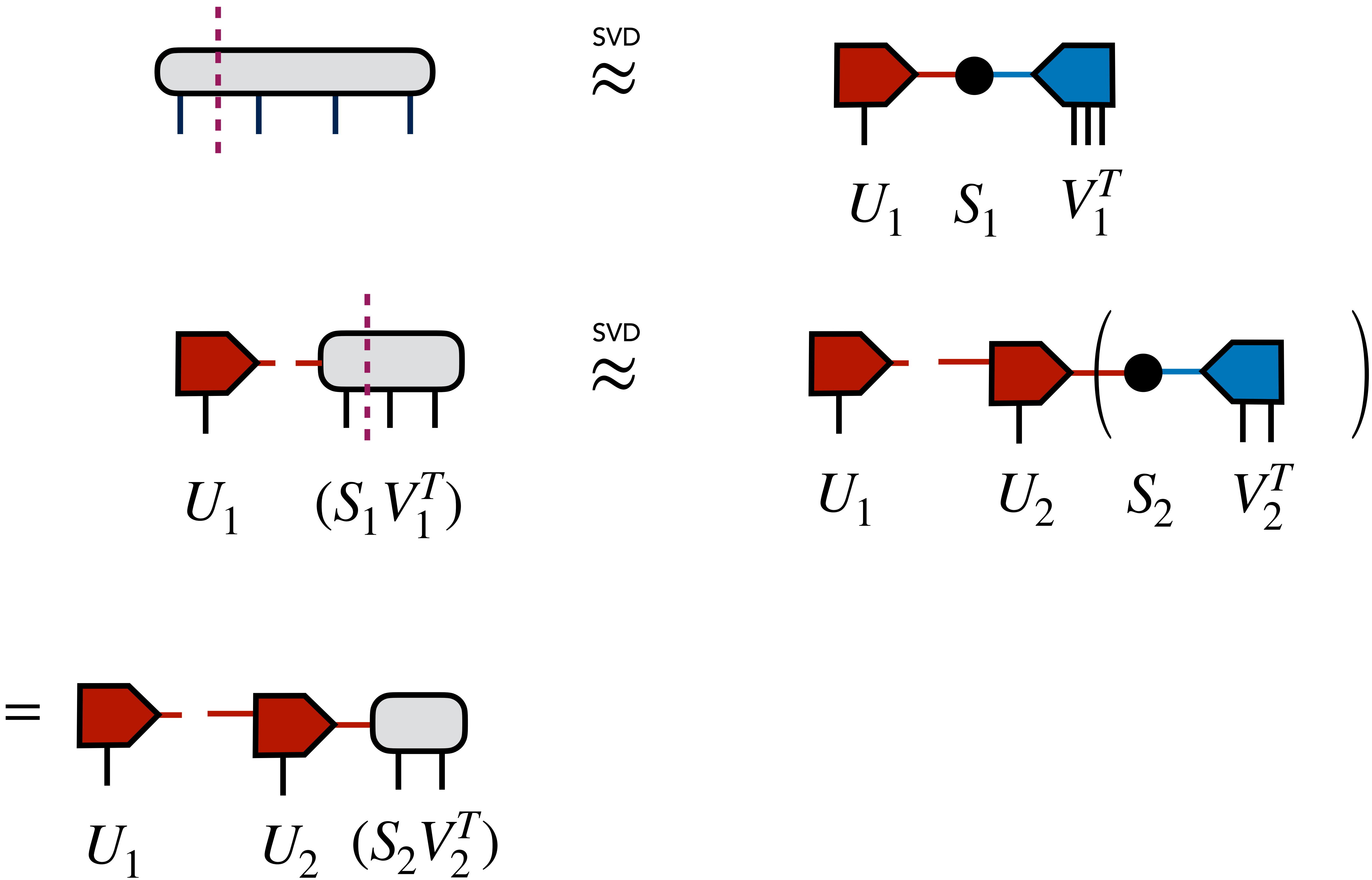
2. Multiply S_1 into V_1^T



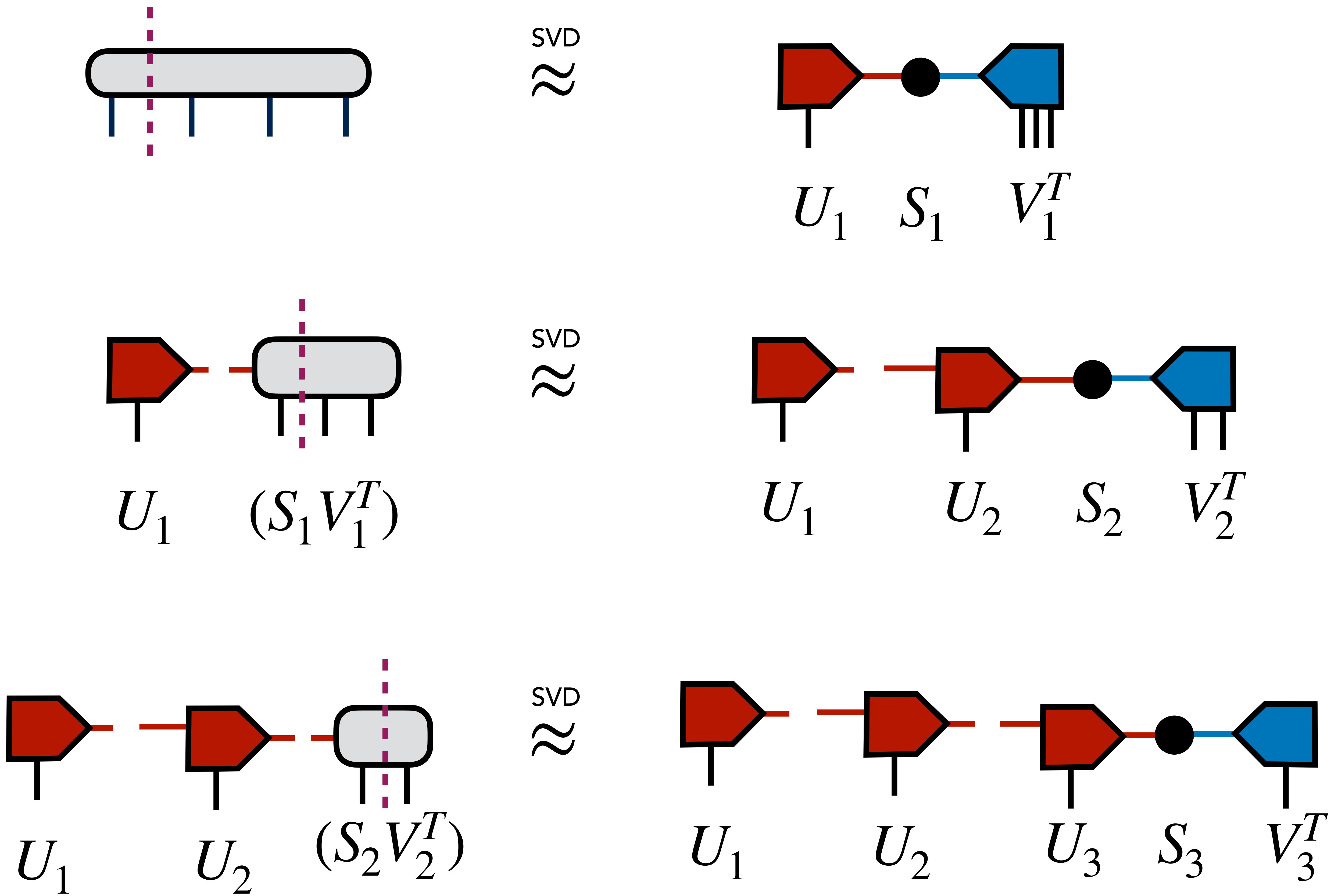
3. SVD this new tensor $(= S_1 V_1^T)$



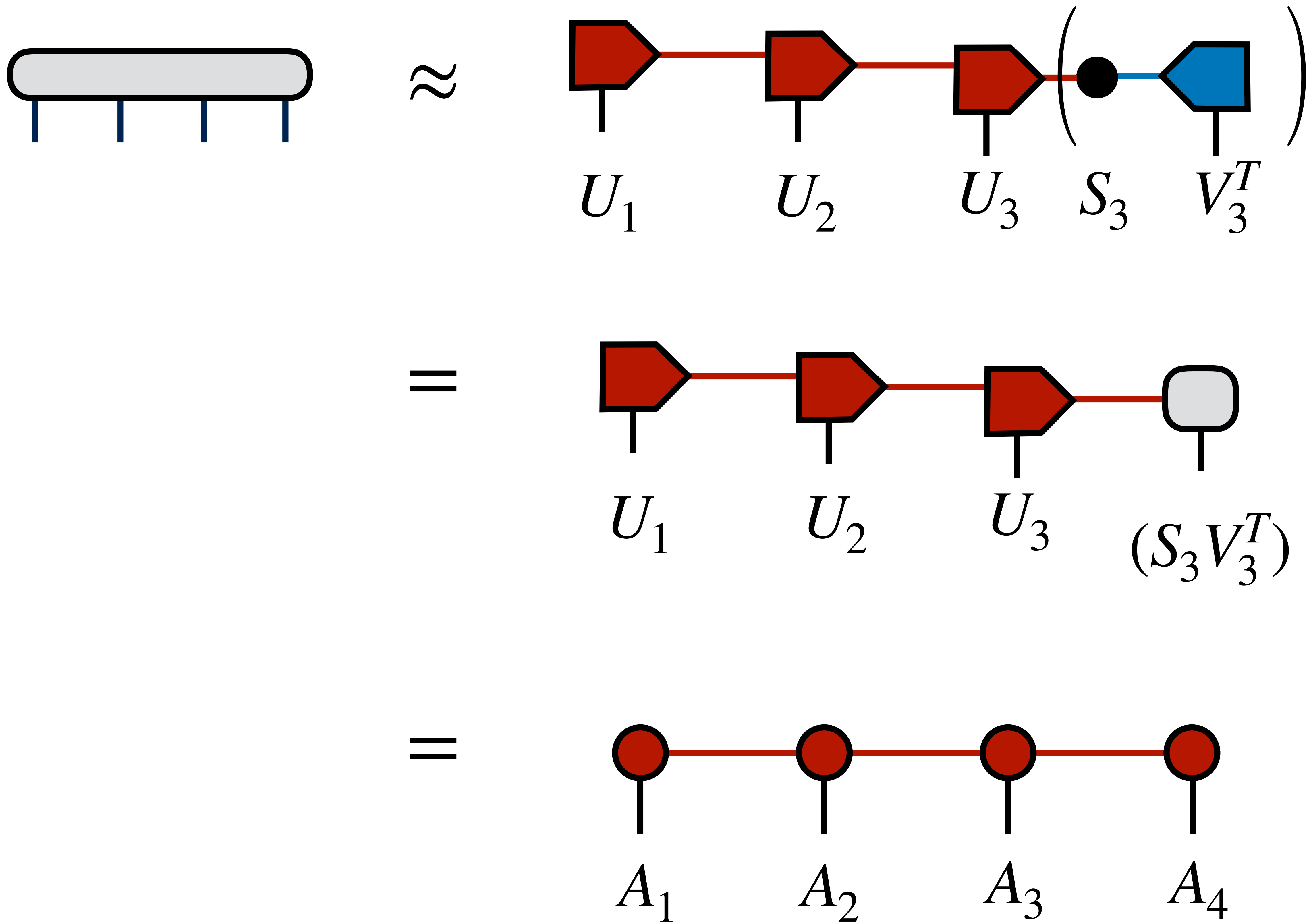
4. Multiply S_2 into V_2



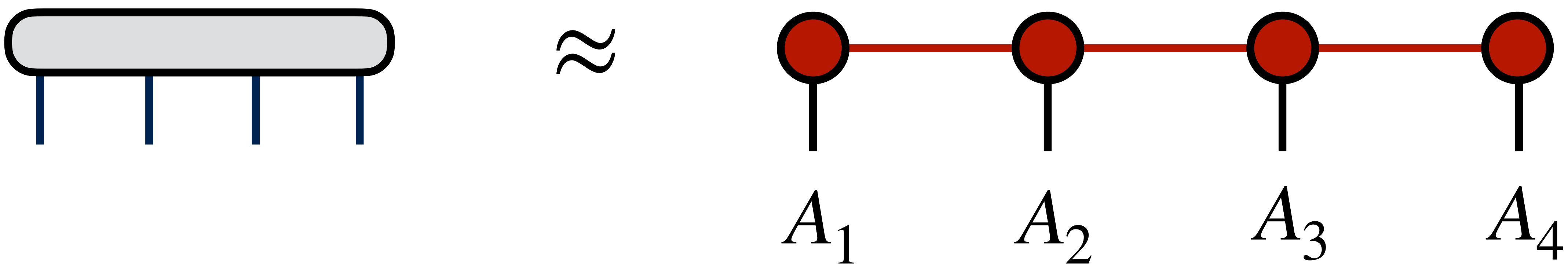
5. Finally SVD $(S_2 V_2^T)$



6. Interpret result as a tensor network



This decomposition is called a "**matrix product state**" (MPS)



Two views of matrix product states

$\longrightarrow$

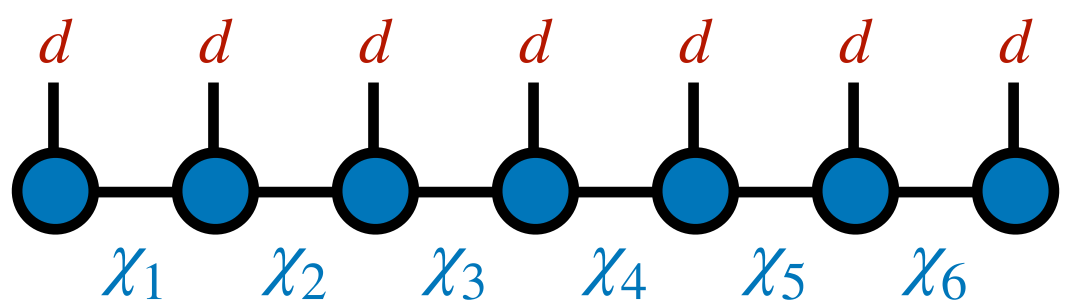
factorization of a tensor

$\longleftarrow$

special subset of tensors
(low-rank tensor manifold)

Matrix product state (MPS) tensor network

Size of a matrix product state (MPS)



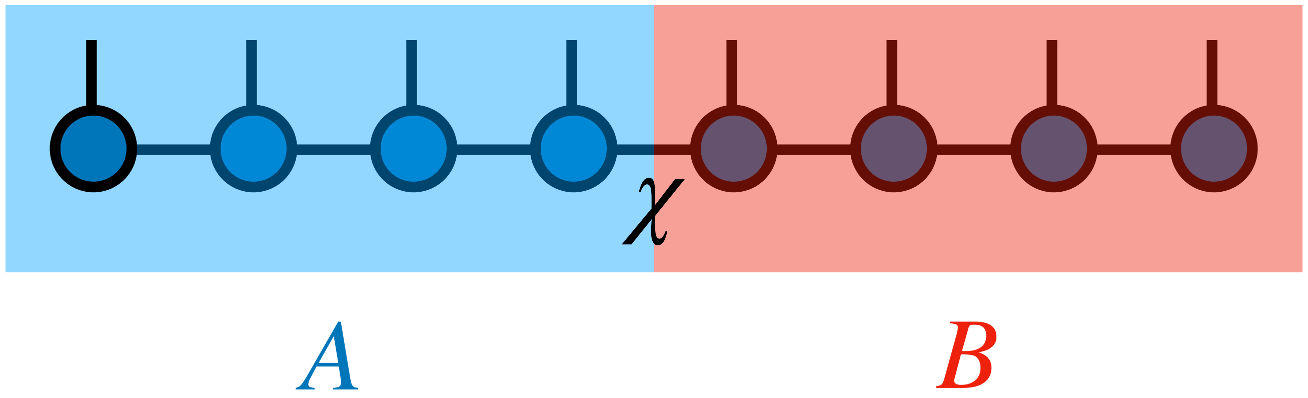
N tensors of size $\chi \cdot d \cdot \chi = d\chi^2$

Memory cost: $N\, d\, \chi^2 \quad \left(\ll d^N \right)$

Matrix product state (MPS) tensor network

What governs bond dimensions?

Divide sites into regions A and B



Can show $\chi \geq \exp(S)$

Where S is the entanglement entropy* between A and B

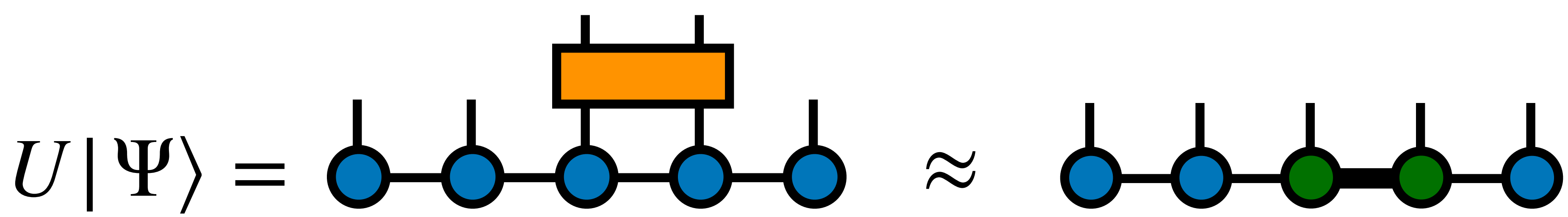
More entanglement $\implies$ larger bond dimensions

* von Neumann entanglement entropy

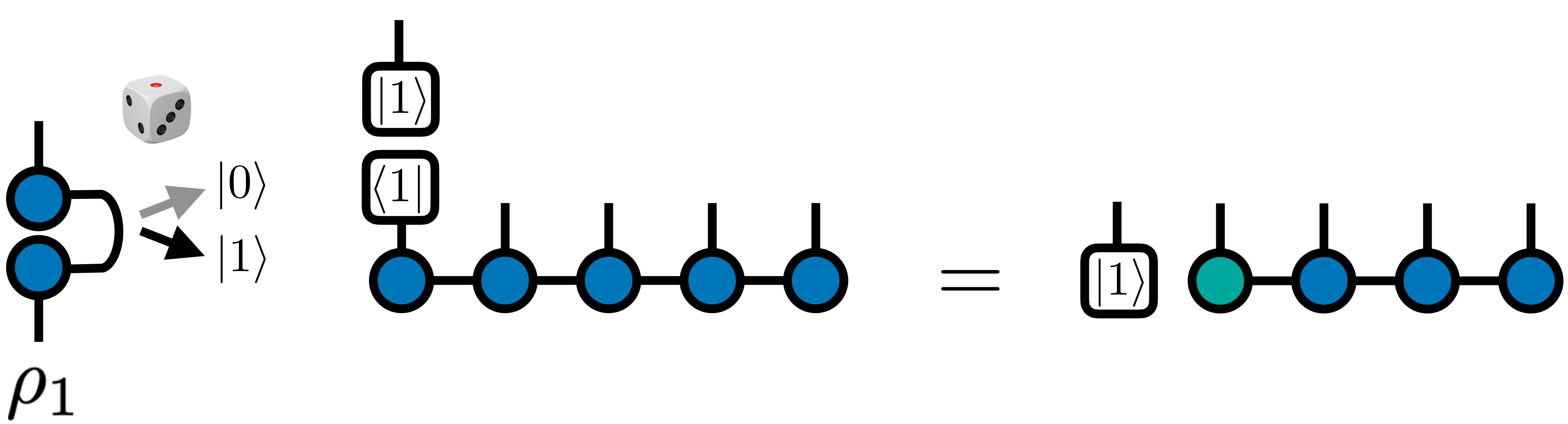
Tensor Networks

Many efficient algorithms for tensor networks

Applying quantum gates (time evolution)



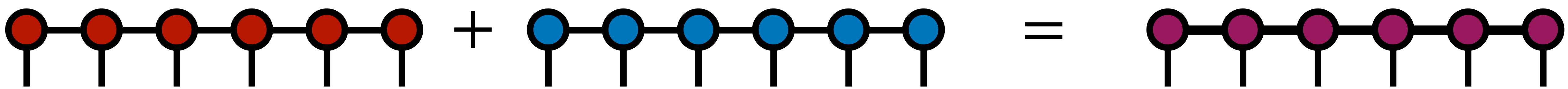
"Perfect sampling" from MPS tensor networks



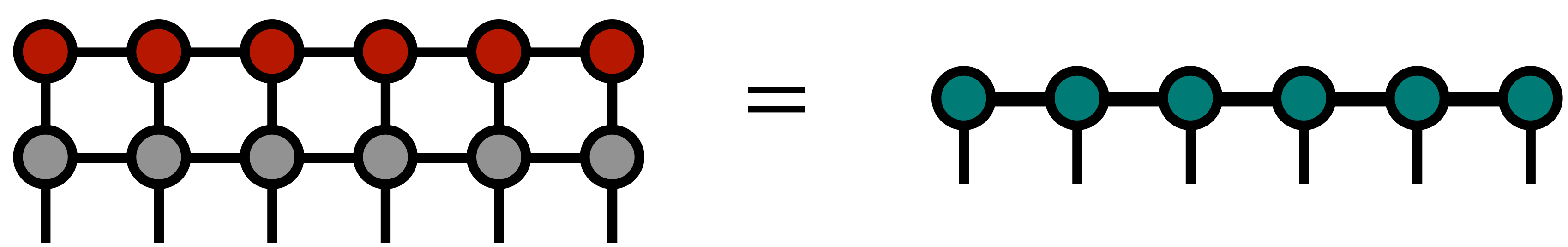
Tensor Networks

Many efficient algorithms for tensor networks

Summing MPS in compressed form



Multiply by other networks (MPO operator $\times$ MPS)

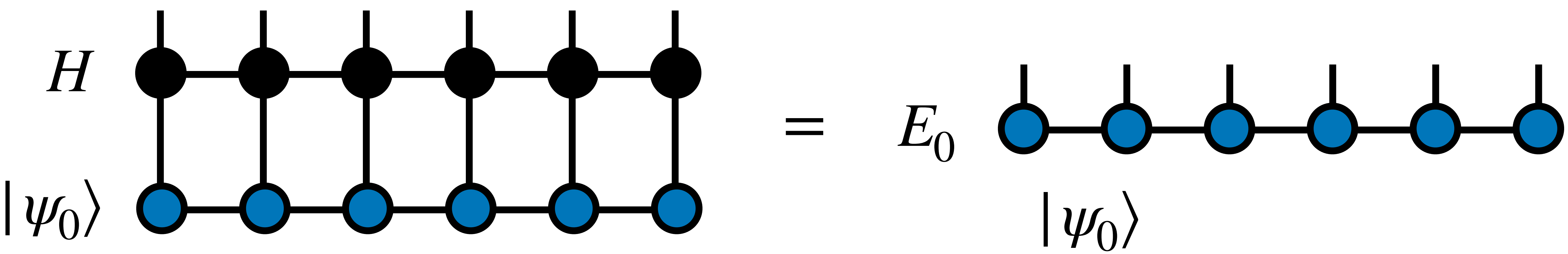


Tensor Networks

Many efficient algorithms for tensor networks

Let's study an important algorithm more fully...

DMRG algorithm



Density Matrix Renormalization Group (DMRG)

DMRG Algorithm

The seminal tensor network algorithm is DMRG [1,2] (density matrix renormalization group)

Given a Hamiltonian H

Find **ground state** and its **energy**

$$\min_{\psi} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} = E_0$$

[1] White, PRL 69, 2863 (1992)
[2] Schollwöck, Annals of Phys. 326, 96 (2011)

DMRG Algorithm

The seminal tensor network algorithm is DMRG [1,2] (density matrix renormalization group)

Given a Hamiltonian H

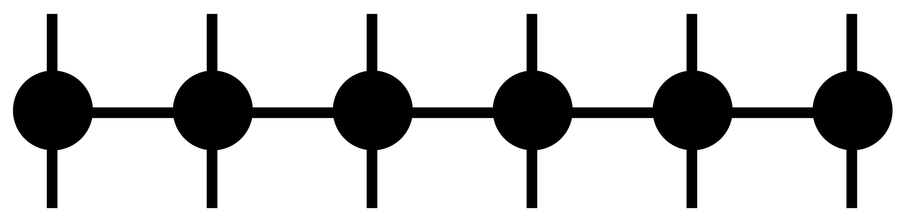
Equivalent to finding 'dominant' eigenvector

$$H|\psi\rangle = E_0|\psi\rangle$$

[1] White, PRL 69, 2863 (1992)
[2] Schollwöck, Annals of Phys. 326, 96 (2011)

DMRG Algorithm

For most common Hamiltonians H,
we can write H as a tensor network

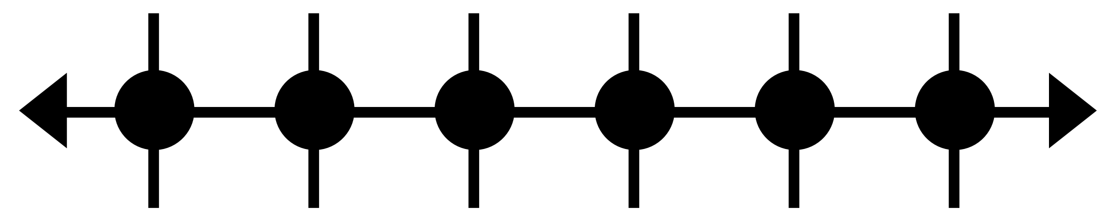
$H =$ 

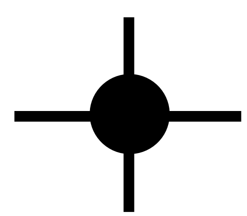
A matrix product operator (MPO) network

MPO Tensor Network

Easiest to understand as operator-valued matrices

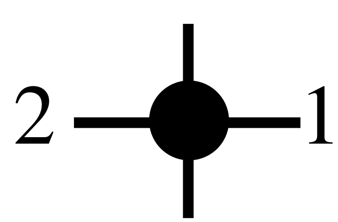
$$H = \sum_{j=1}^{N-1} Z_j Z_{j+1} + \sum_{j=1}^{N} X_j \;=$$



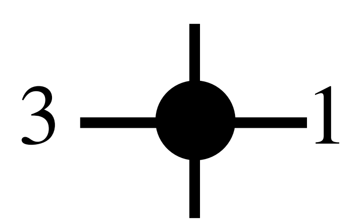


$$= \begin{bmatrix} I & 0 & 0 \\ Z & 0 & 0 \\ X & Z & I \end{bmatrix}$$

This means



$= Z$

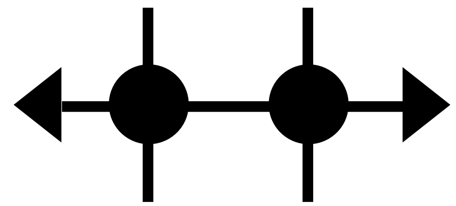


$= X$

etc.

MPO Tensor Network

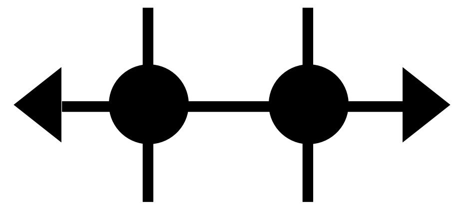
Verify formula for Ising model + transverse field

$H =$ 

$$= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_1 & 0 & 0 \\ Z_1 & 0 & 0 \\ X_1 & Z_1 & I_1 \end{bmatrix} \begin{bmatrix} I_2 & 0 & 0 \\ Z_2 & 0 & 0 \\ X_2 & Z_2 & I_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

MPO Tensor Network

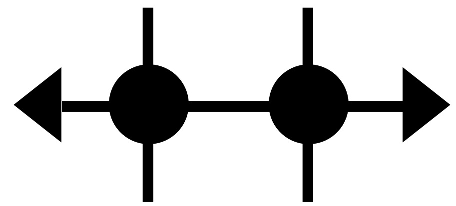
Verify formula for Ising model + transverse field

$H =$ 

$$= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_1 & 0 & 0 \\ Z_1 & 0 & 0 \\ X_1 & Z_1 & I_1 \end{bmatrix} \begin{bmatrix} I_2 \\ Z_2 \\ X_2 \end{bmatrix}$$

MPO Tensor Network

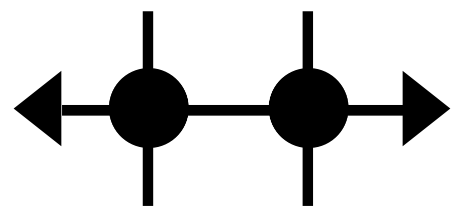
Verify formula for Ising model + transverse field

$H =$ 

$$= \qquad \begin{bmatrix} X_1 & Z_1 & I_1 \end{bmatrix} \begin{bmatrix} I_2 \\ Z_2 \\ X_2 \end{bmatrix}$$

MPO Tensor Network

Verify formula for Ising model + transverse field

$H =$ 

$$= \begin{bmatrix} X_1 & Z_1 & I_1 \end{bmatrix} \begin{bmatrix} I_2 \\ Z_2 \\ X_2 \end{bmatrix}$$

$$= \quad X_1 I_2 + Z_1 Z_2 + I_1 X_2$$

MPO Tensor Network

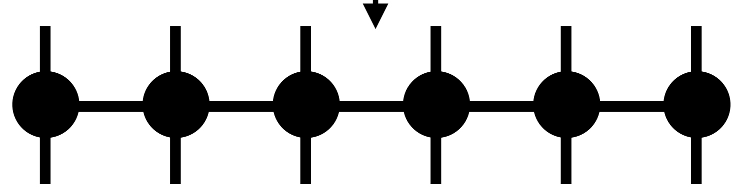
Example in **ITensor** for Ising spin chain

$$H = \sum_j Z_j Z_{j+1} - h \sum_j X_j$$

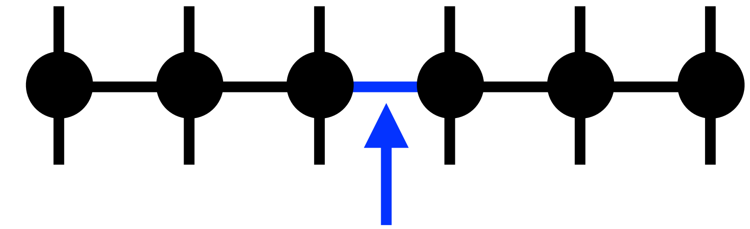
```
sites = siteinds("S=1/2",N)

terms = OpSum()
for j=1:N-1
  terms += "Z",j, "Z",j+1
end
for j=1:N
  terms -= h,"X",j
end

H = MPO(terms,sites)
```

$H =$ 

MPO Tensor Network



Common MPO bond dimensions – 1D

- Transverse-field Ising

$$H = \sum_j Z_j Z_{j+1} - h \sum_j X_j$$

bond dimension 3

- Heisenberg model

$$H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1}$$

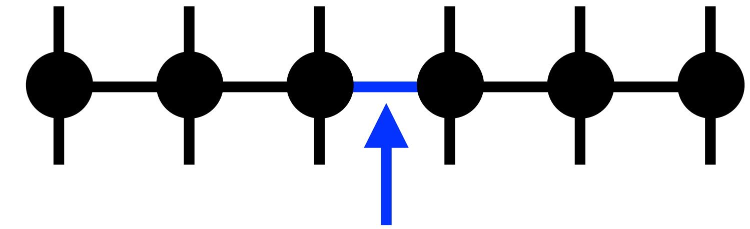
bond dimension 5

- Hubbard model

$$H = -t \sum_{j,\sigma=\uparrow,\downarrow} (c^\dagger_{j,\sigma} c_{j+1,\sigma} + c^\dagger_{j+1,\sigma} c_{j,\sigma}) + U \sum_j n_{j\uparrow} n_{j\downarrow}$$

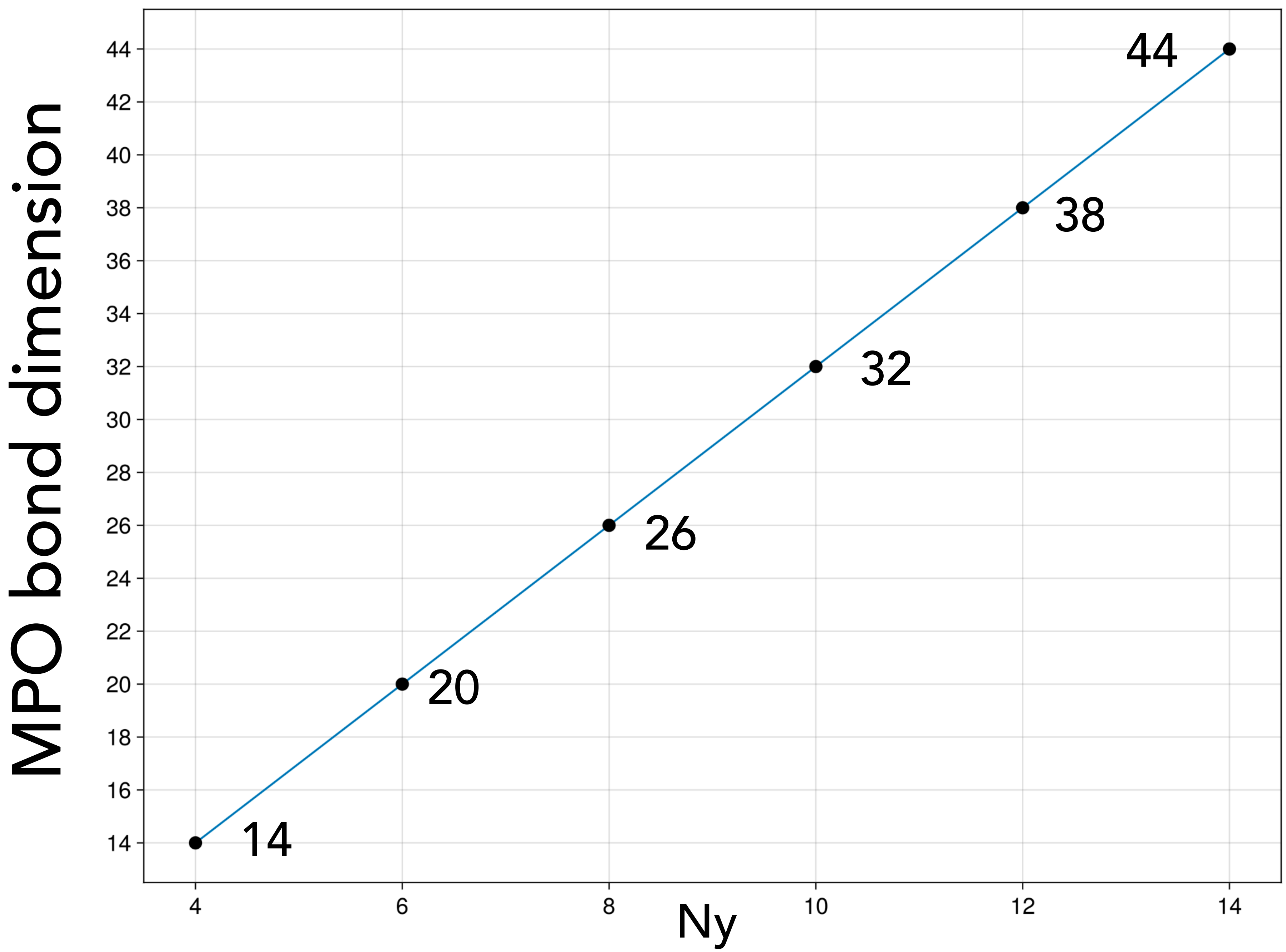
bond dimension 6

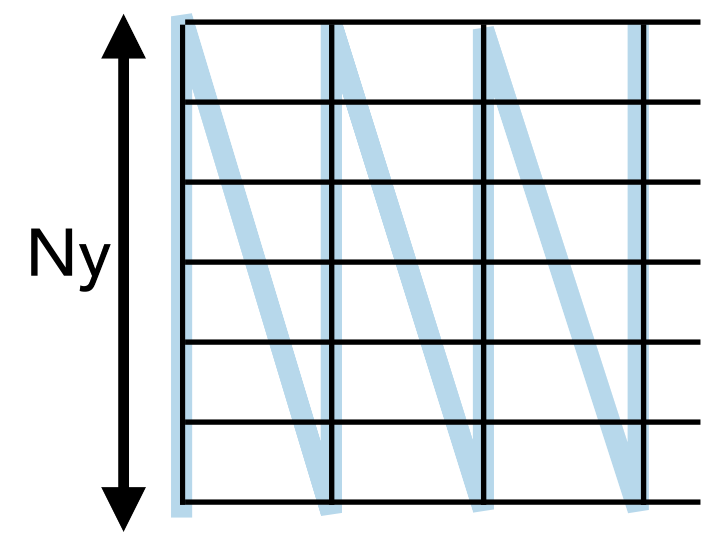
MPO Tensor Network



Common MPO bond dimensions – 2D

- Heisenberg model $H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1}$

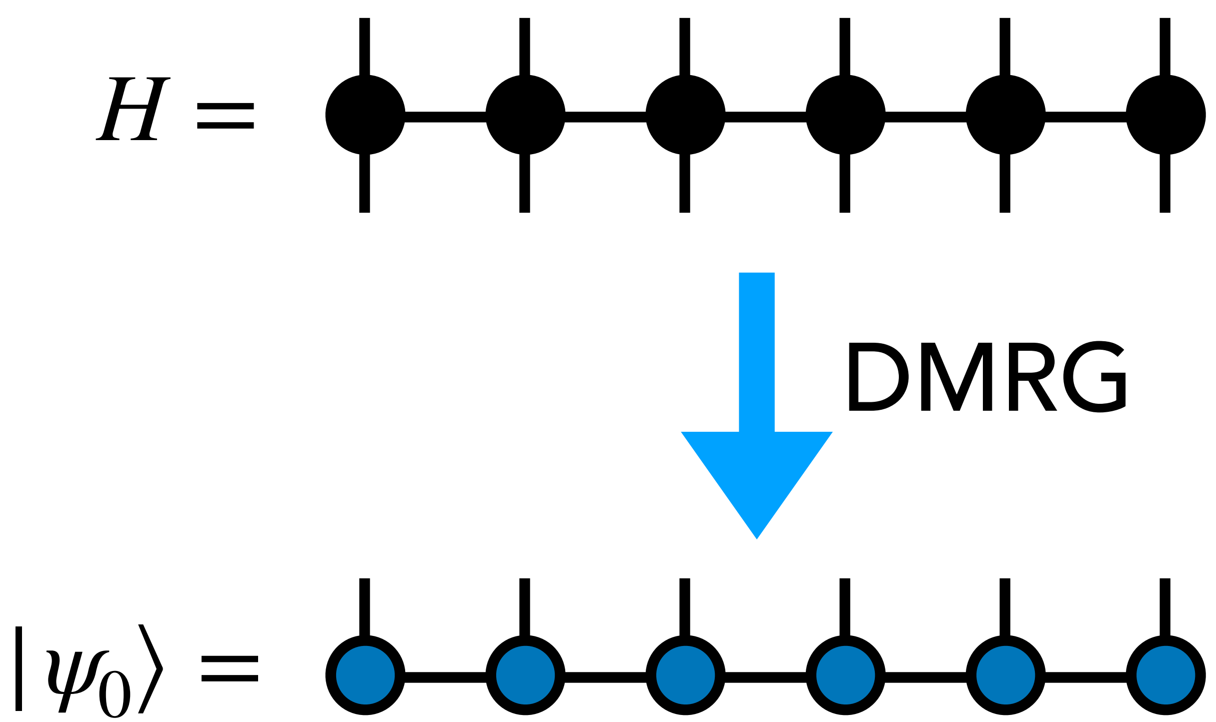




Tensor Network Algorithms

DMRG algorithm

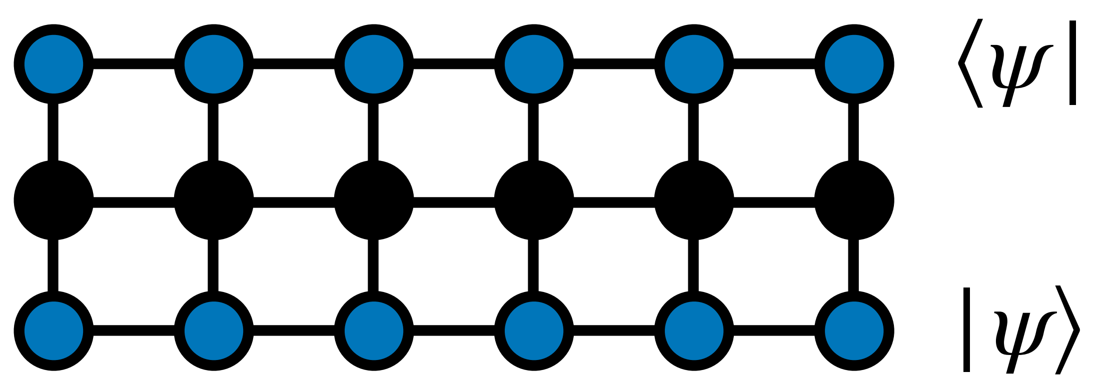
DMRG finds ground state of H as MPS tensor network



Tensor Network Algorithms

DMRG algorithm

Energy is

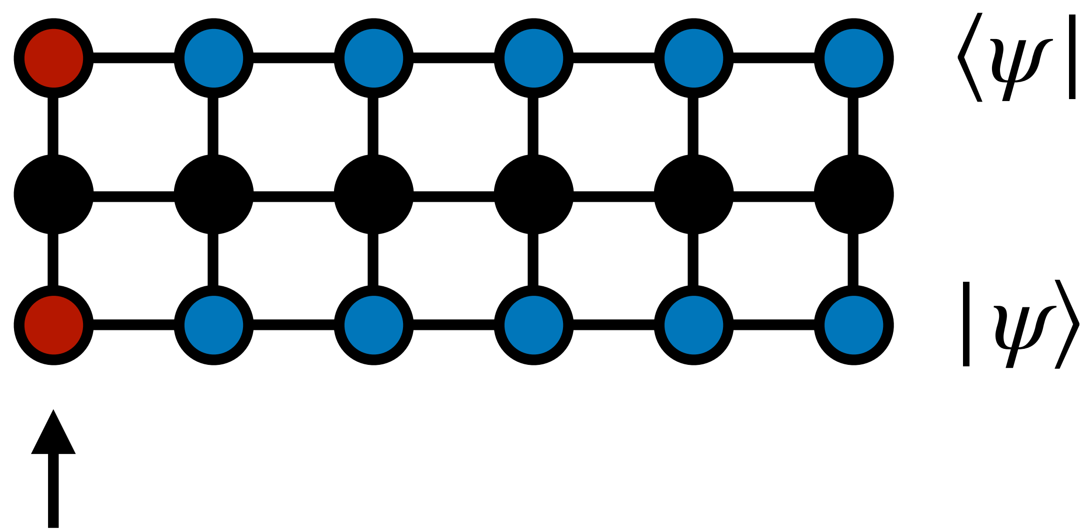
$E =$ 

Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize

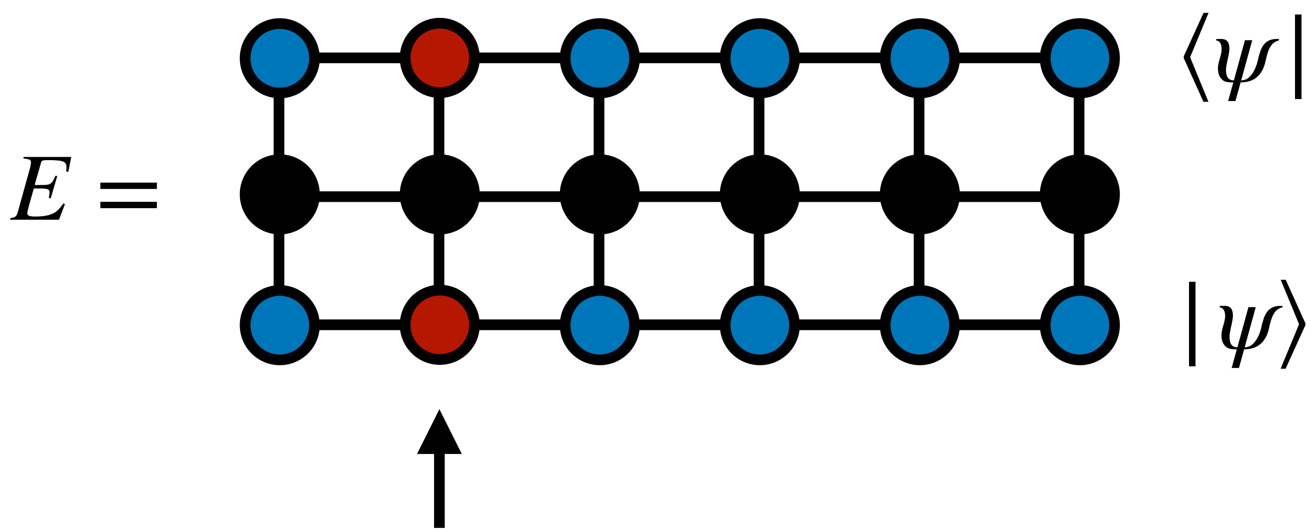
$E =$



Tensor Network Algorithms

DMRG algorithm

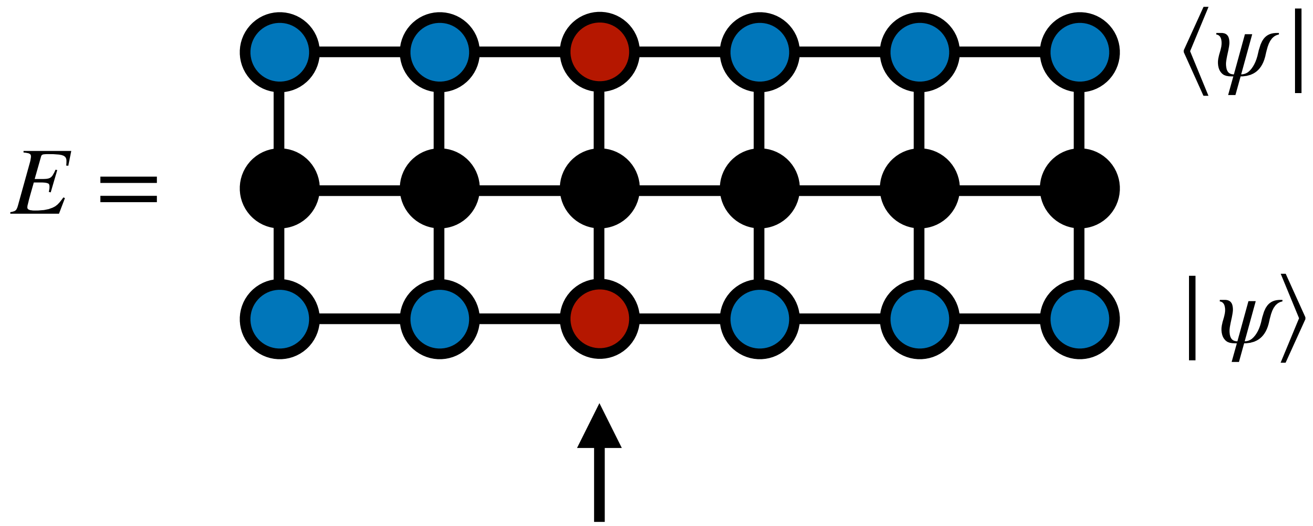
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

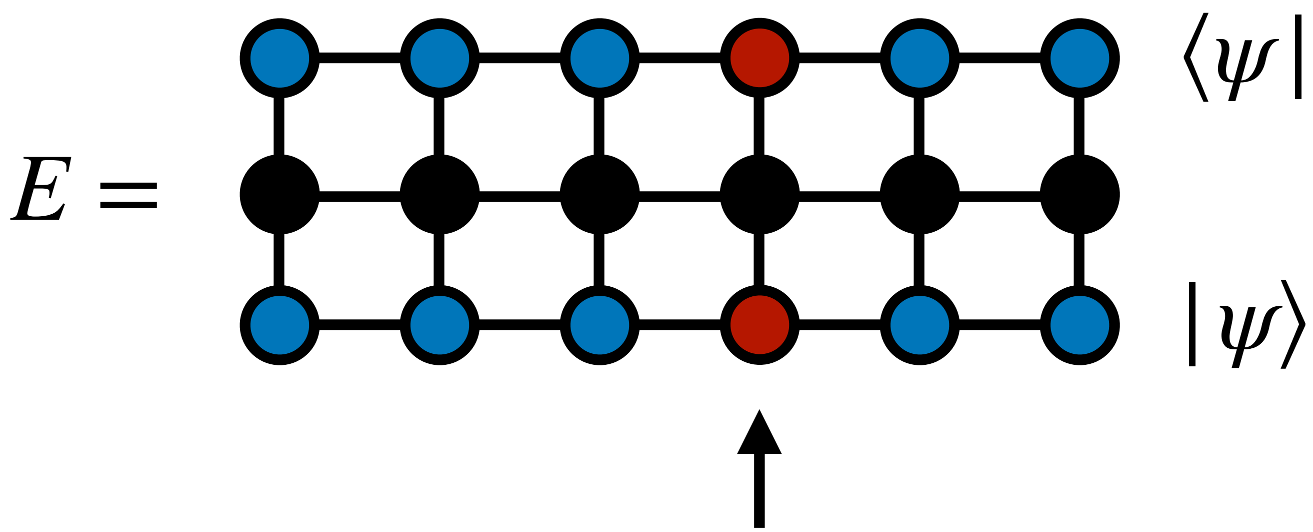
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize

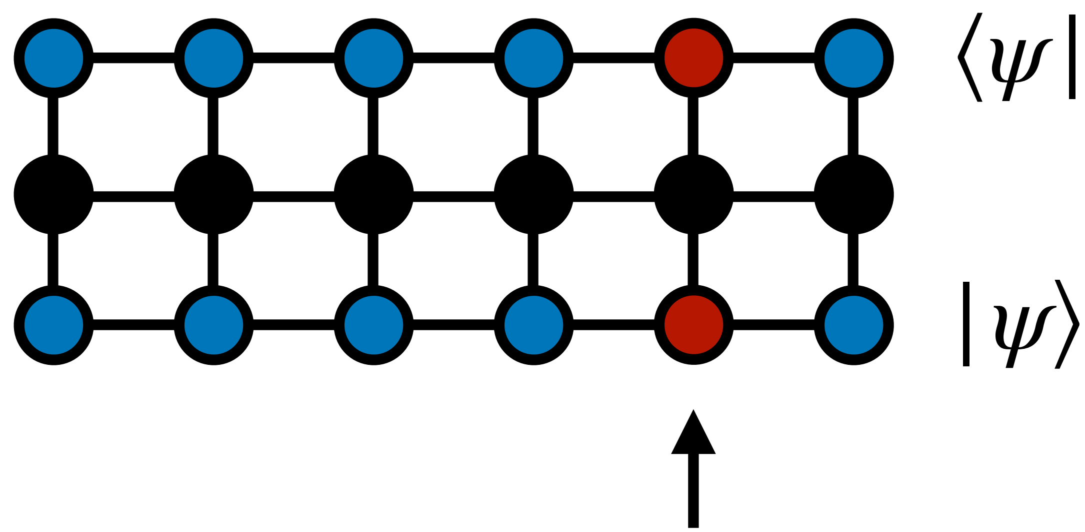


Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize

$E =$

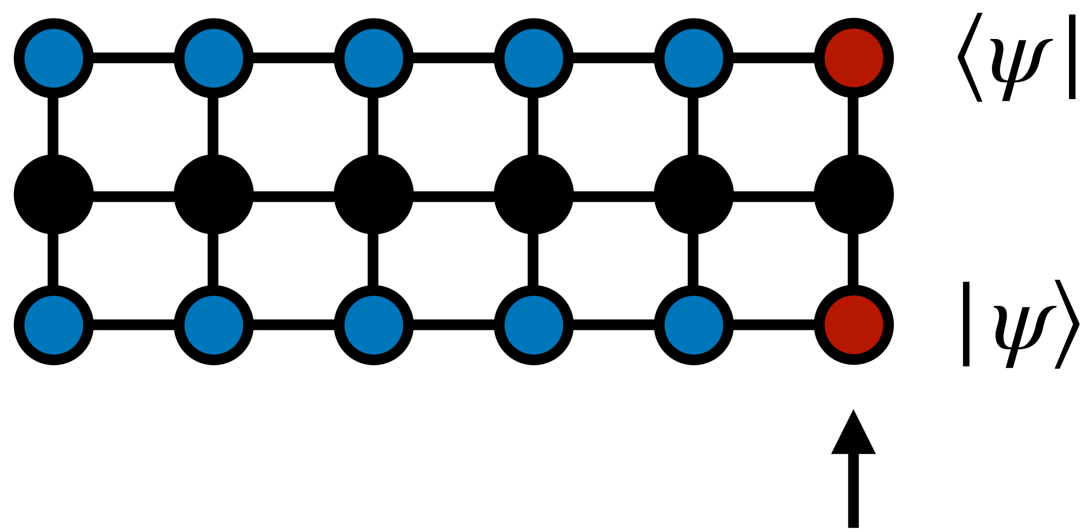


Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize

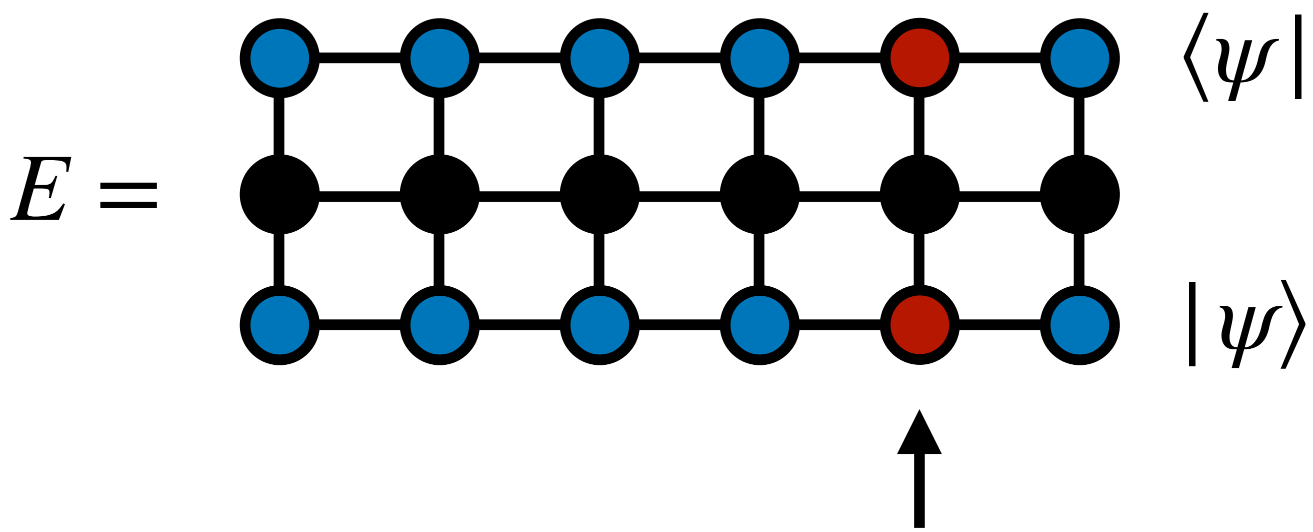
$E =$



Tensor Network Algorithms

DMRG algorithm

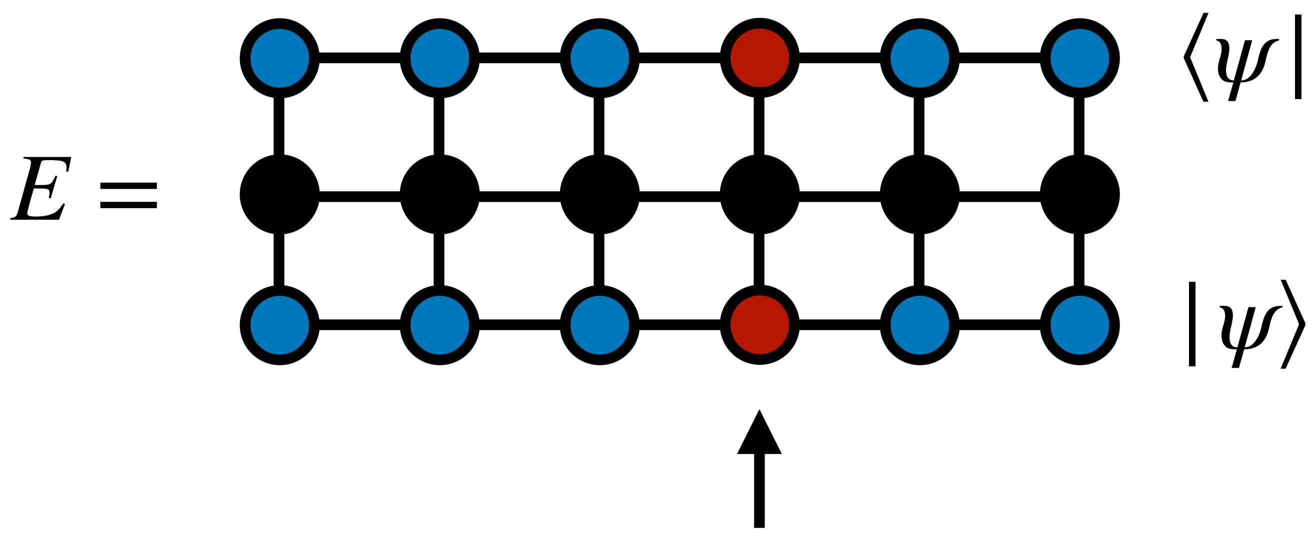
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

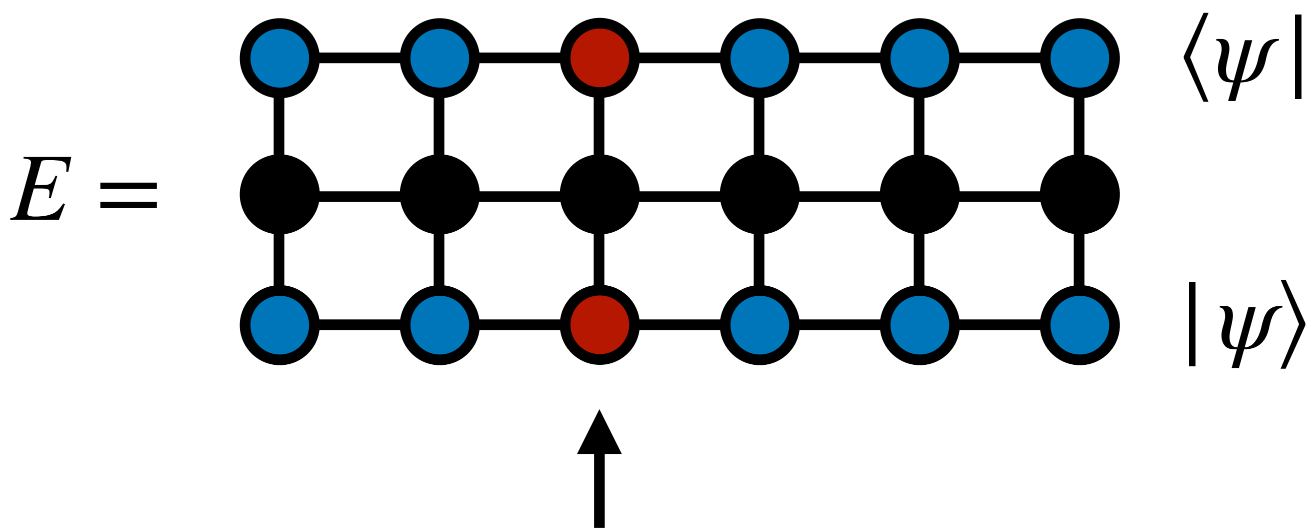
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize

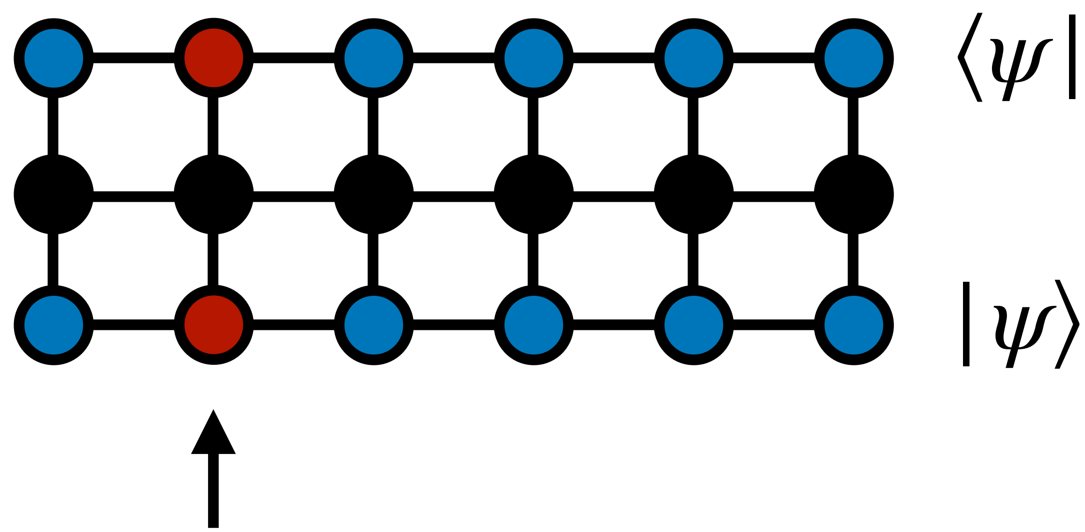


Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize

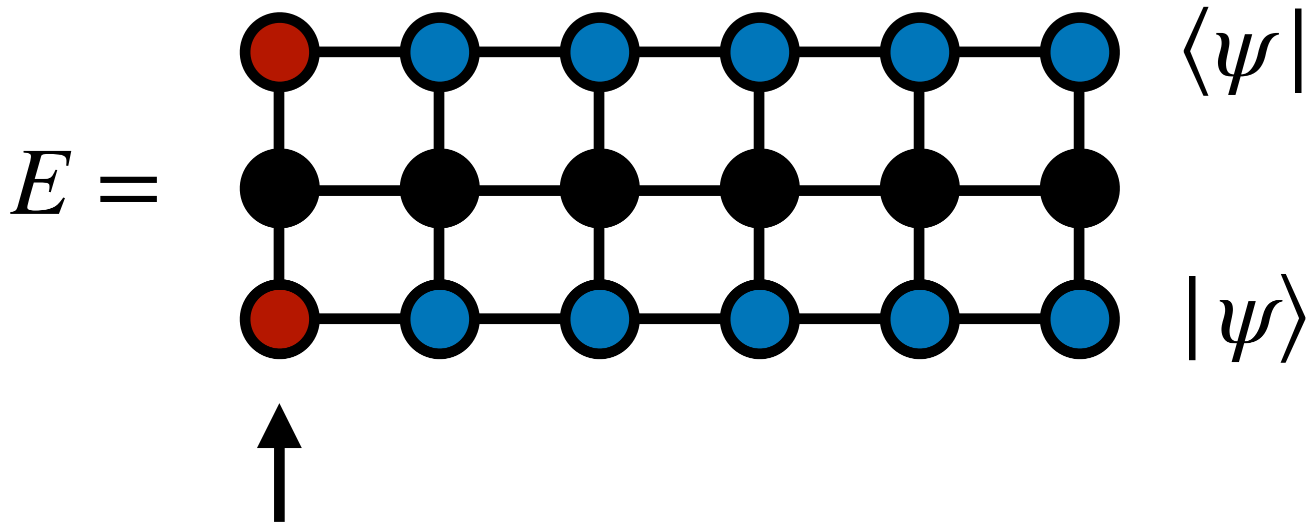
$E =$



Tensor Network Algorithms

DMRG algorithm

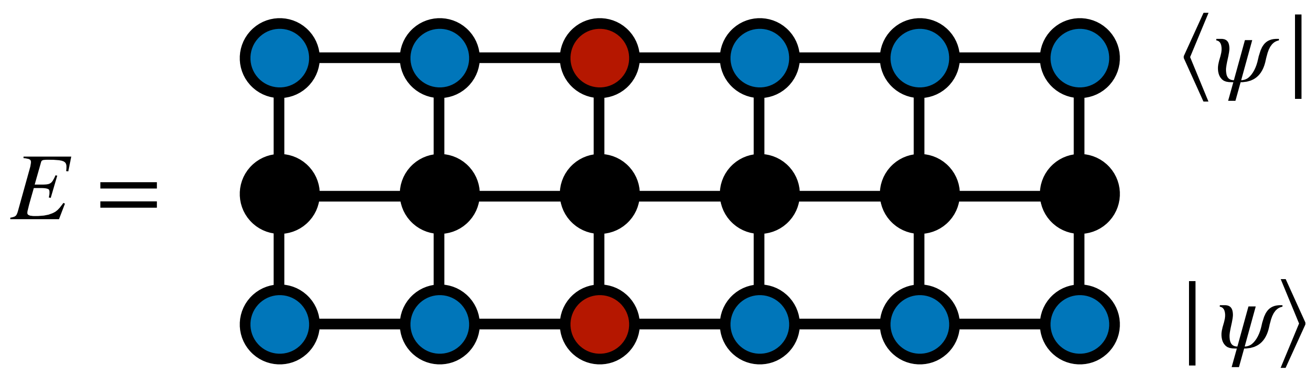
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

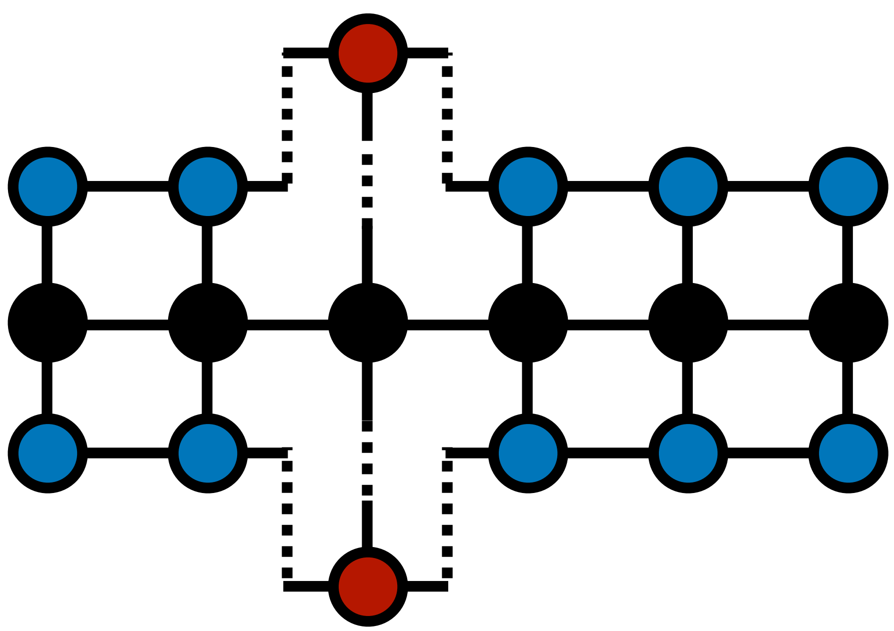
At each step, solve a reduced diagonalization problem



Tensor Network Algorithms

DMRG algorithm

At each step, solve a reduced diagonalization problem

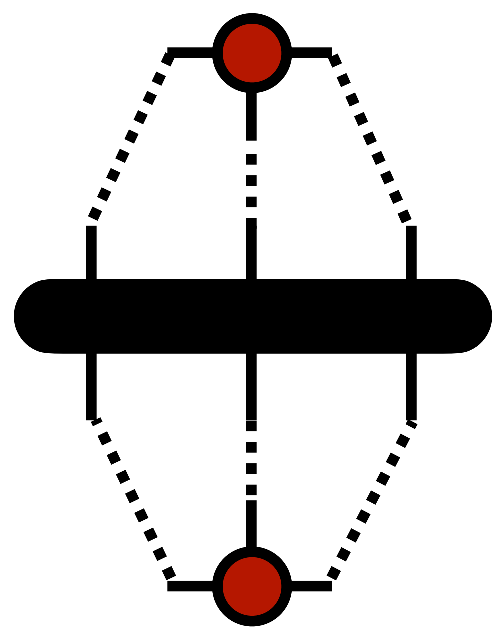


*technical note: for efficiency, frozen tensors are contracted in three groups, not exactly as shown above

Tensor Network Algorithms

DMRG algorithm

At each step, solve a reduced diagonalization problem



$$= E$$

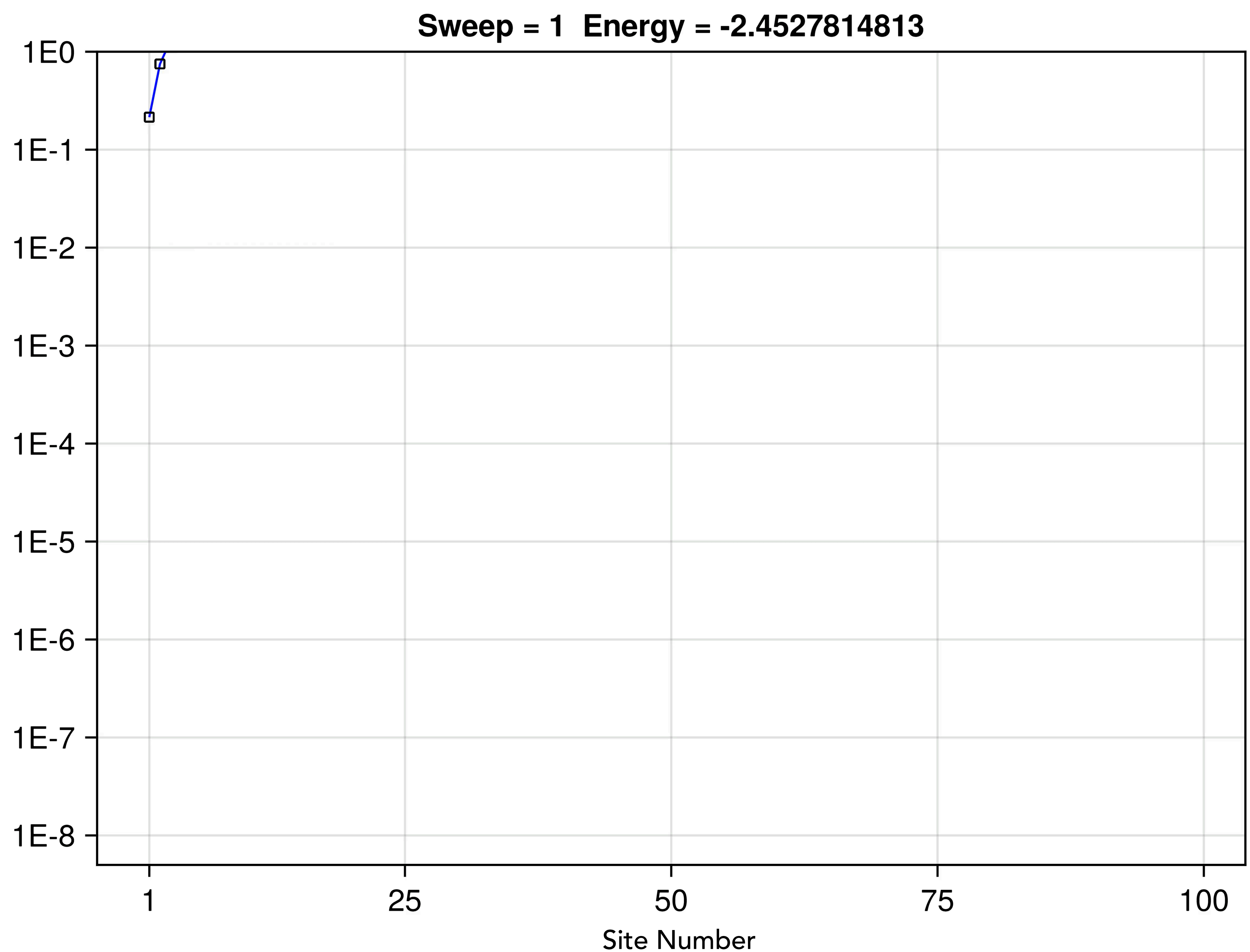
*technical note: for efficiency, frozen tensors are contracted in three groups, not exactly as shown above

Tensor Network Algorithms

DMRG solving S=1 Heisenberg chain $H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1}$

(Using error goal or cutoff = 1E-8)

$$\langle S_j \cdot S_{j+1} \rangle - \langle S_j \cdot S_{j+1} \rangle_{\text{exact}}$$



Tensor Network Algorithms

DMRG algorithm is extremely precise

Energy error vs number of 'sweeps' of DMRG

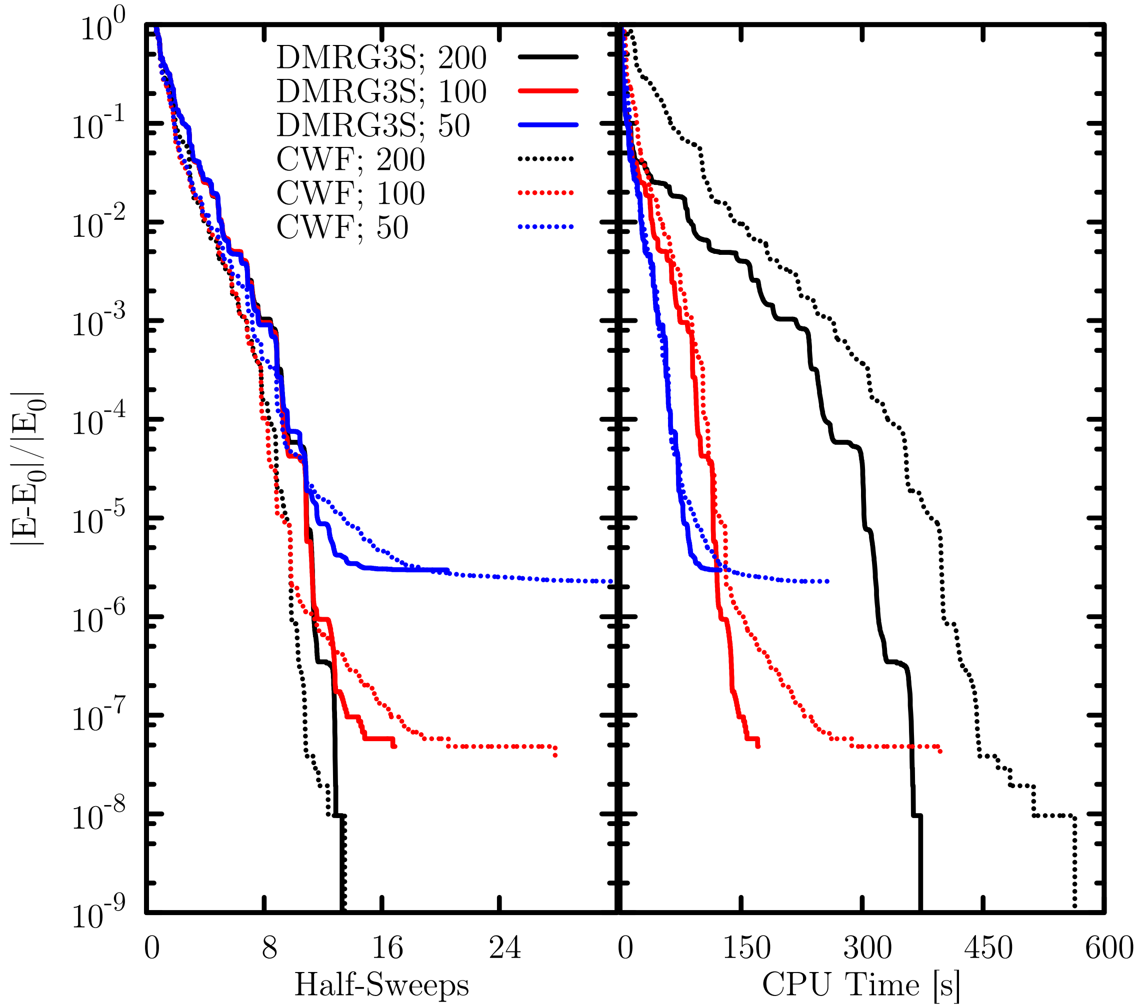


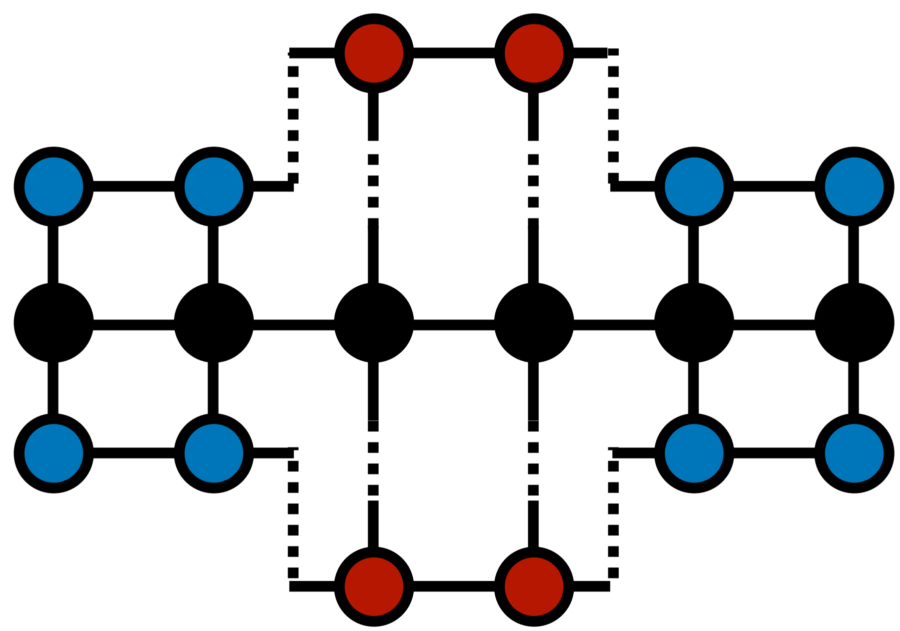
FIG. 4. (Color online) Bosonic system Eq. (41): normalized error in energy from CWF and DMRG3S as a function of sweeps (left) and CPU time used (right) for $m = 50, 100, 200$.

Hubig, McCulloch, Schollwöck, Wolf, Phys. Rev. B 91, 155115

Tensor Network Algorithms

In practice, some important differences from above version

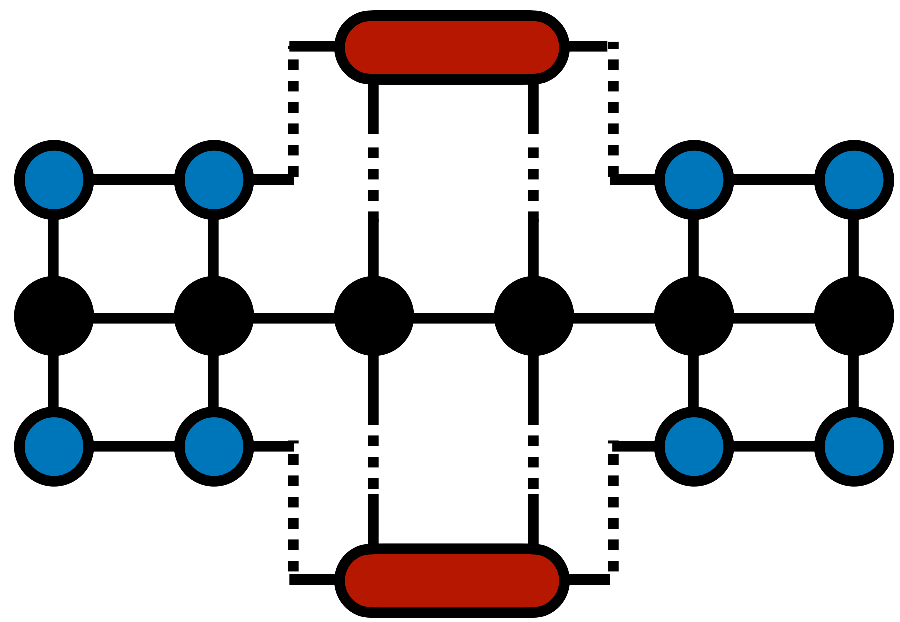
Typically <u>two-site</u> algorithm used:



Tensor Network Algorithms

In practice, some important differences from above version

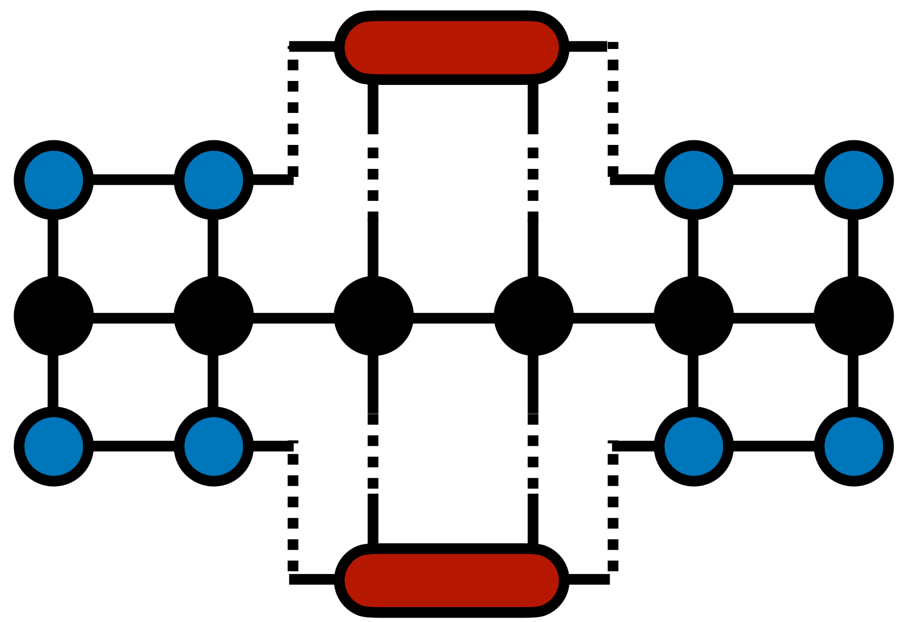
Typically <u>two-site</u> algorithm used:



Tensor Network Algorithms

In practice, some important differences from above version

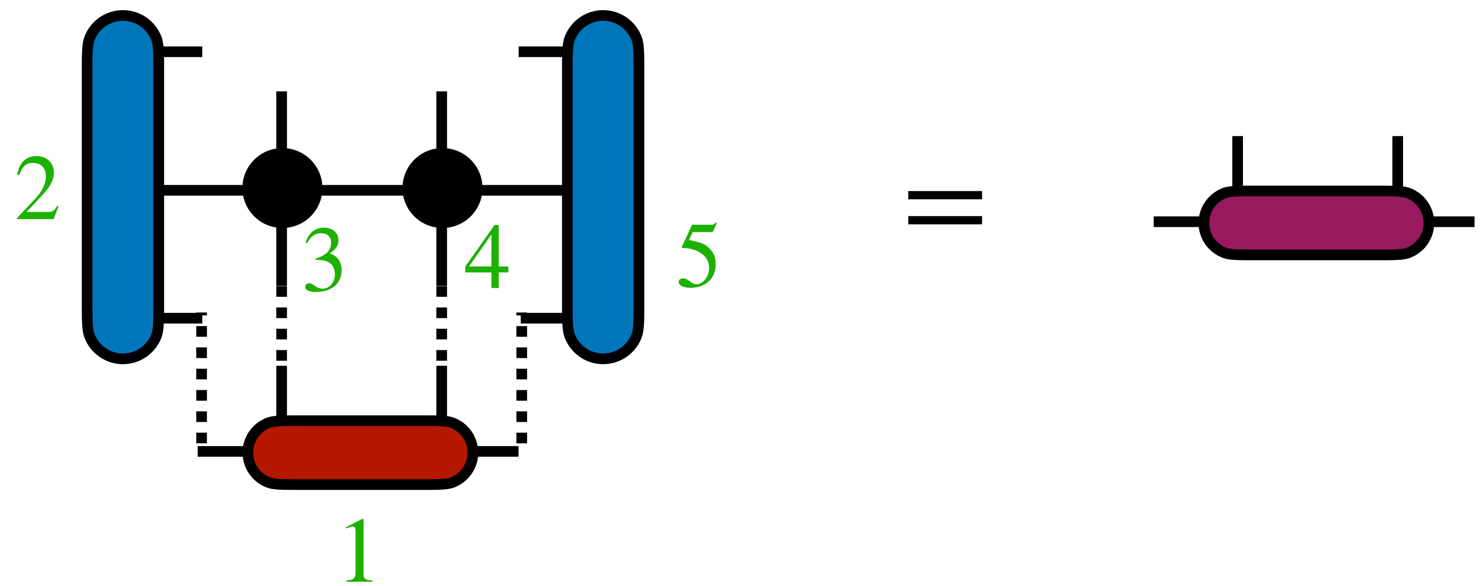
Eigen-solver step done more efficiently



Tensor Network Algorithms

In practice, some important differences from above version

Eigen-solver step done more efficiently

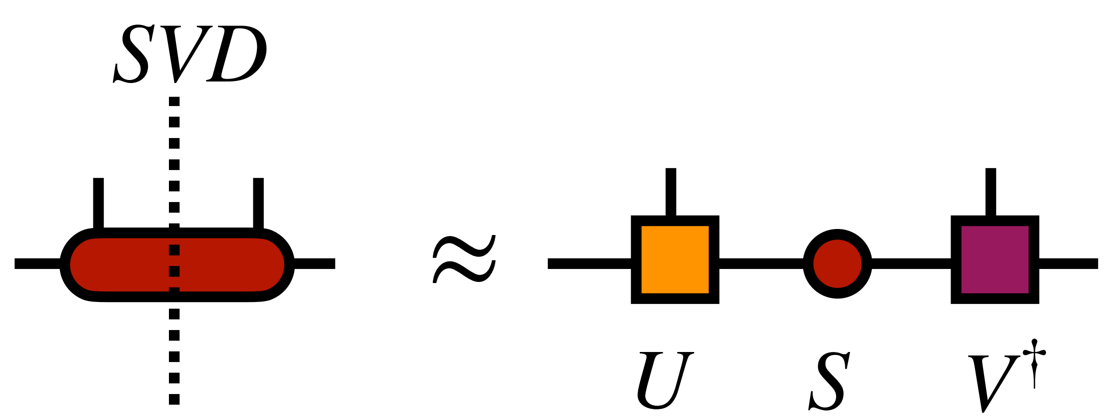


Lanczos / Krylov method to compute solution

Tensor Network Algorithms

In practice, some important differences from above version

Bond dimension "adapted" afterward



DMRG is named after this step

Summary

Tensor networks can compress many-body states by an exponential amount

DMRG efficiently finds ground states as MPS tensor networks

Up Next

After a break, we will learn how to set up and run DMRG calculations using the ITensor software

Bonus Slides

Gauging Tensor Networks

Gauging Tensor Networks

Recall gauging in electromagnetism ⚡
If one transforms:

$$\mathbf{A} \to \mathbf{A} + \nabla f$$

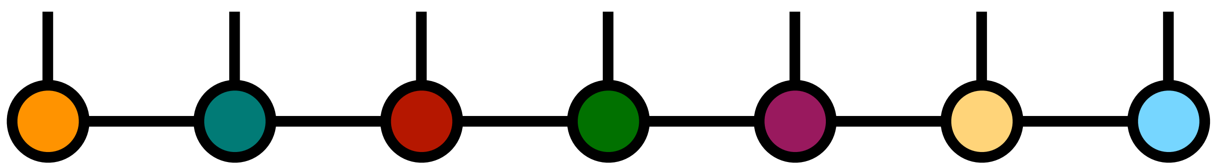
$$\varphi \to \varphi - \frac{\partial f}{\partial t}$$

Then physical observables do not change

Implies redundancy in the representation

Gauging Tensor Networks

Tensor networks likewise have a huge redundancy



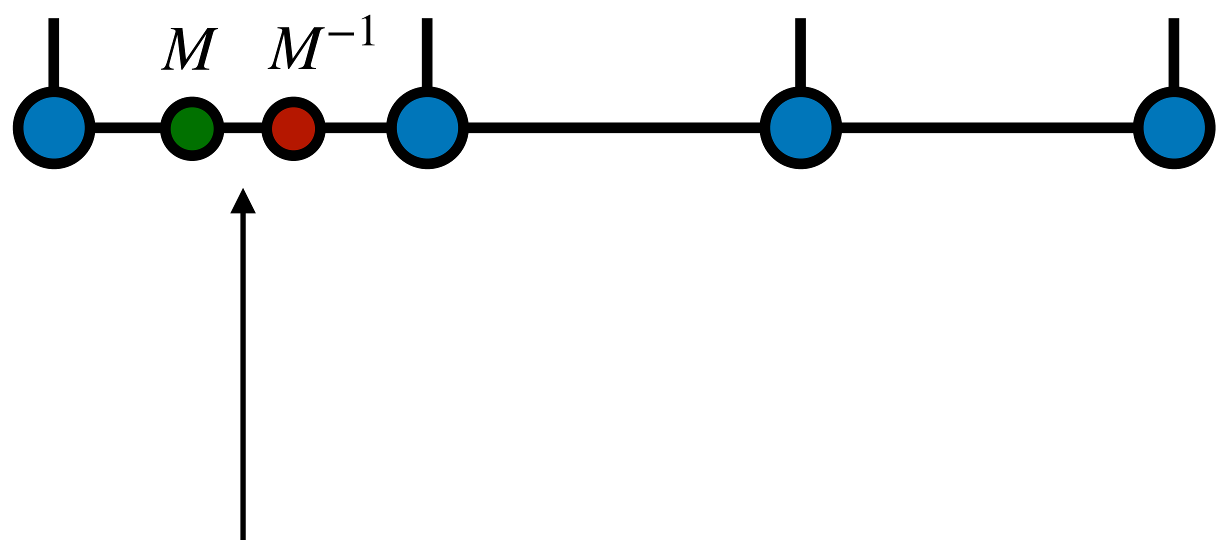
They are "gauge theories" of quantum states

Gauging Tensor Networks

What are the gauge transformations of an MPS?

Gauging Tensor Networks

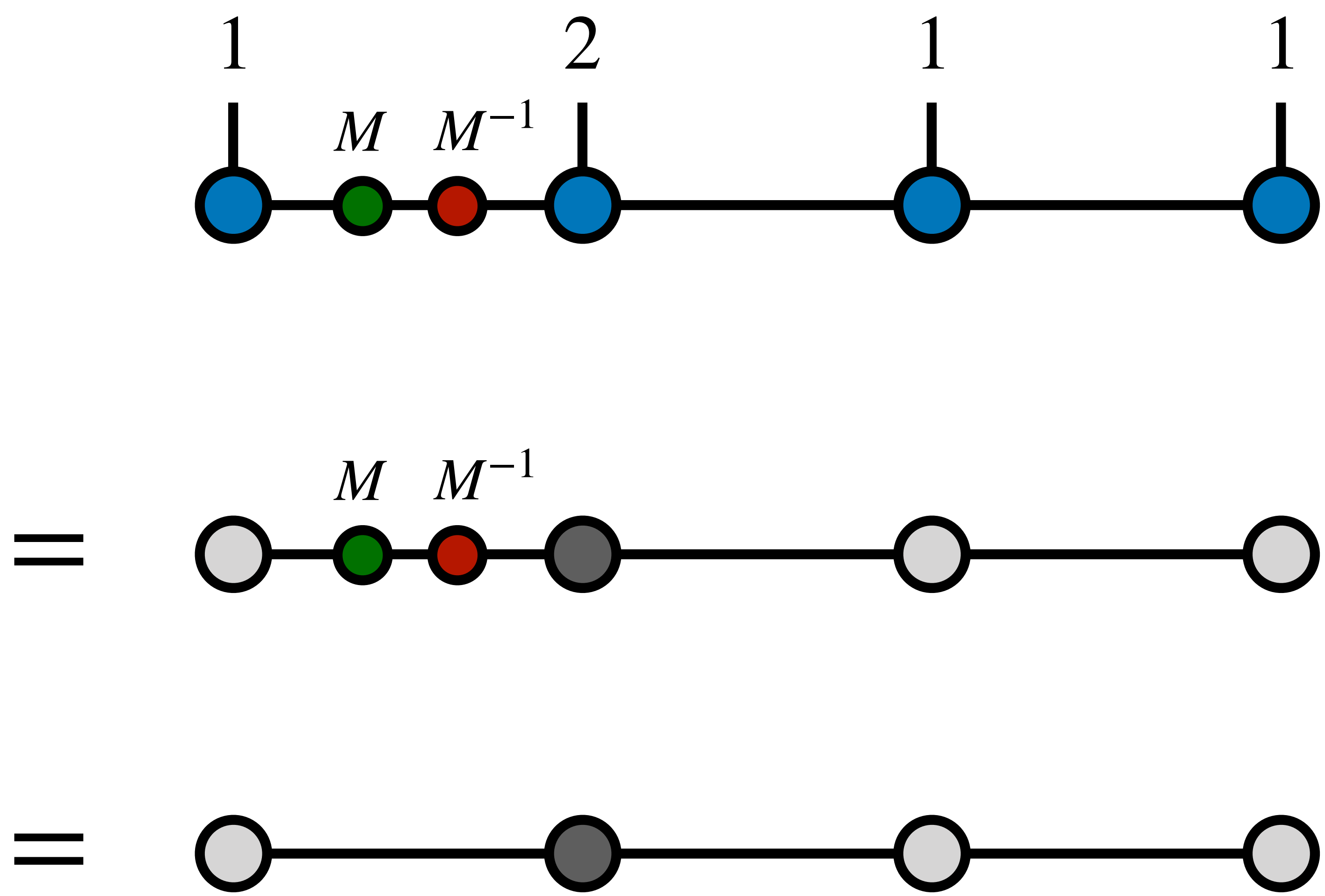
What are the gauge transformations of an MPS?



Insert resolution of identity $1 = MM^{-1}$

Gauging Tensor Networks

Check the tensor is unchanged – compute the T^{1211} element for example:



M and M^{-1} cancel out, proving each element unchanged

Gauging Tensor Networks

Can absorb "gauging" matrices into MPS

Different MPS representing the same tensor

Gauging Tensor Networks

What gauges are useful?

How does one compute useful gauges?

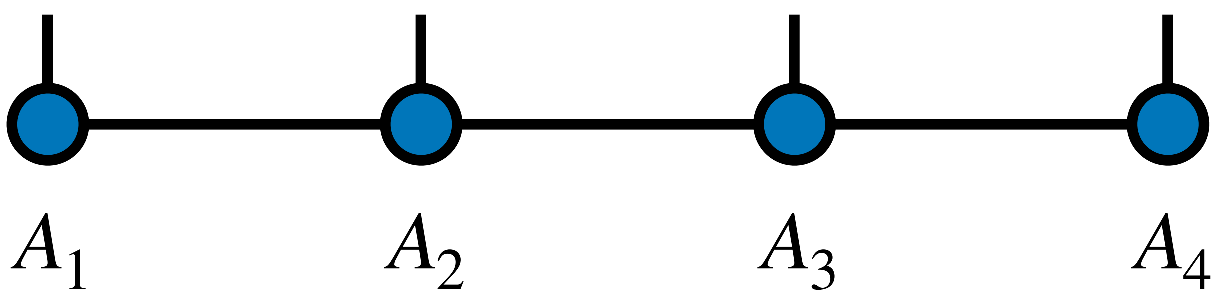
Interesting TN gauges related to:

- matrix factorizations
- quantum circuits

Orthogonal MPS Gauge

Let's find the "orthogonal" gauges of an MPS

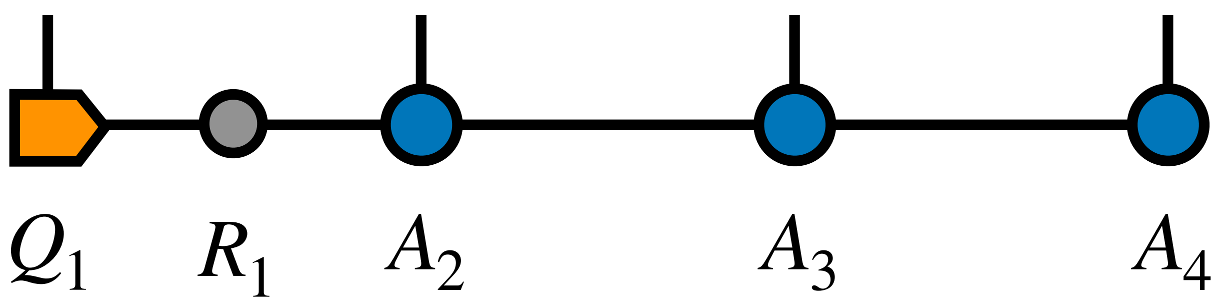
Start from an arbitrary MPS



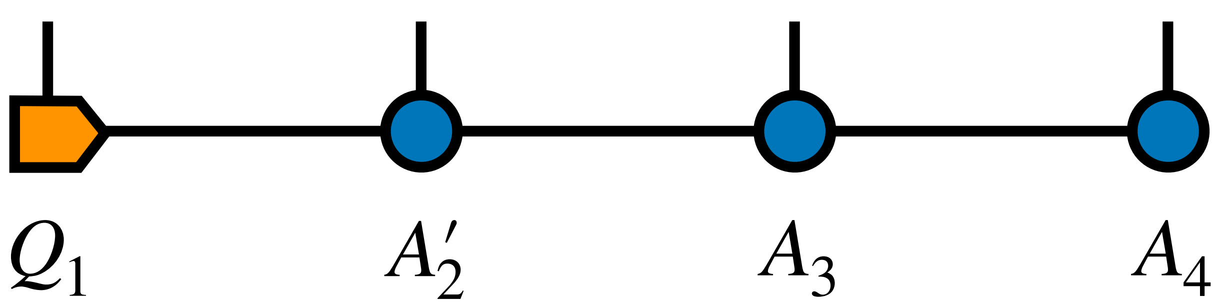
A_1 A_2 A_3 A_4

Orthogonal MPS Gauge

Now compute QR decomposition of $A_1 = Q_1 R_1$

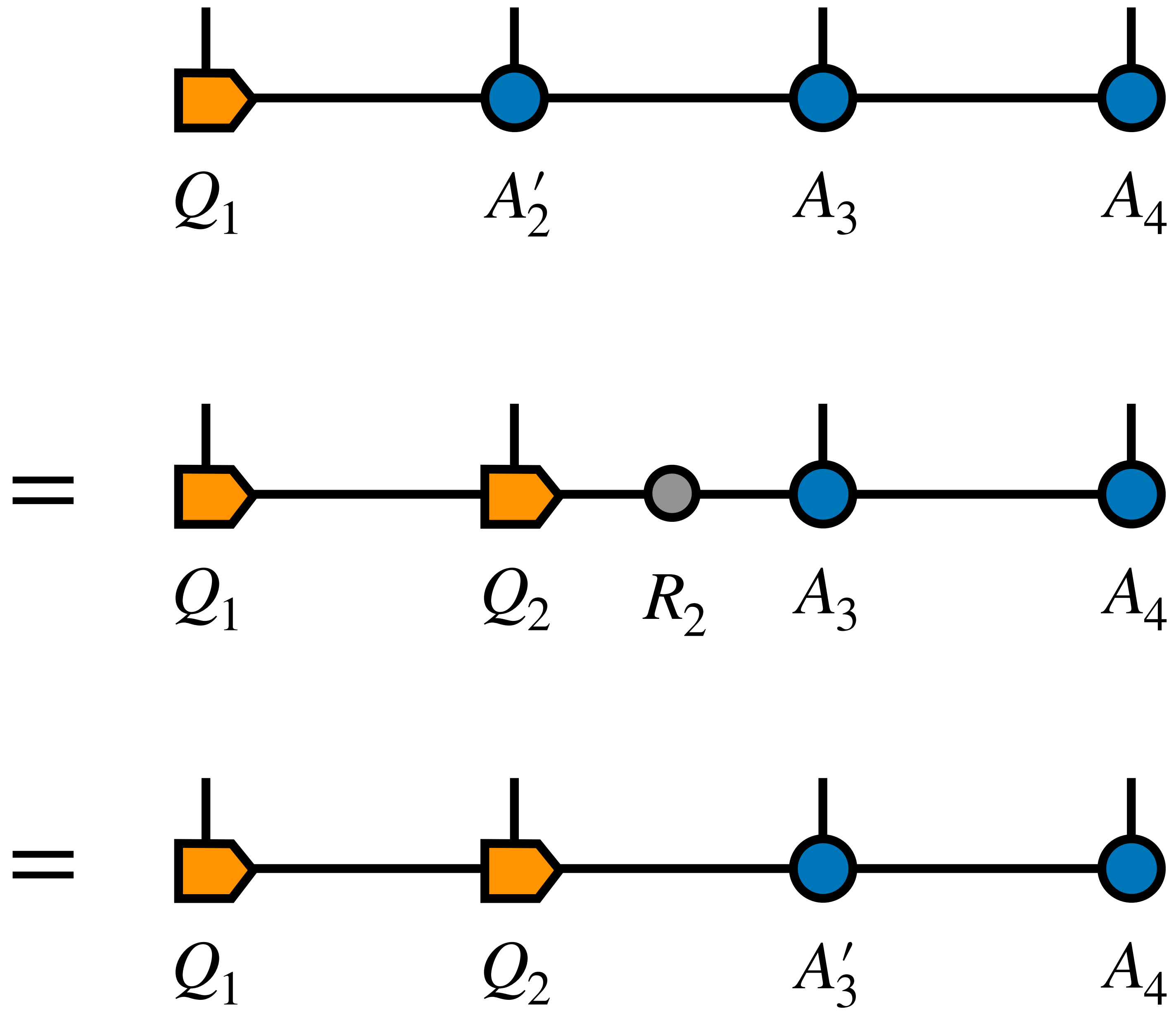


Multiply R_1 into A_2 to make A_2'



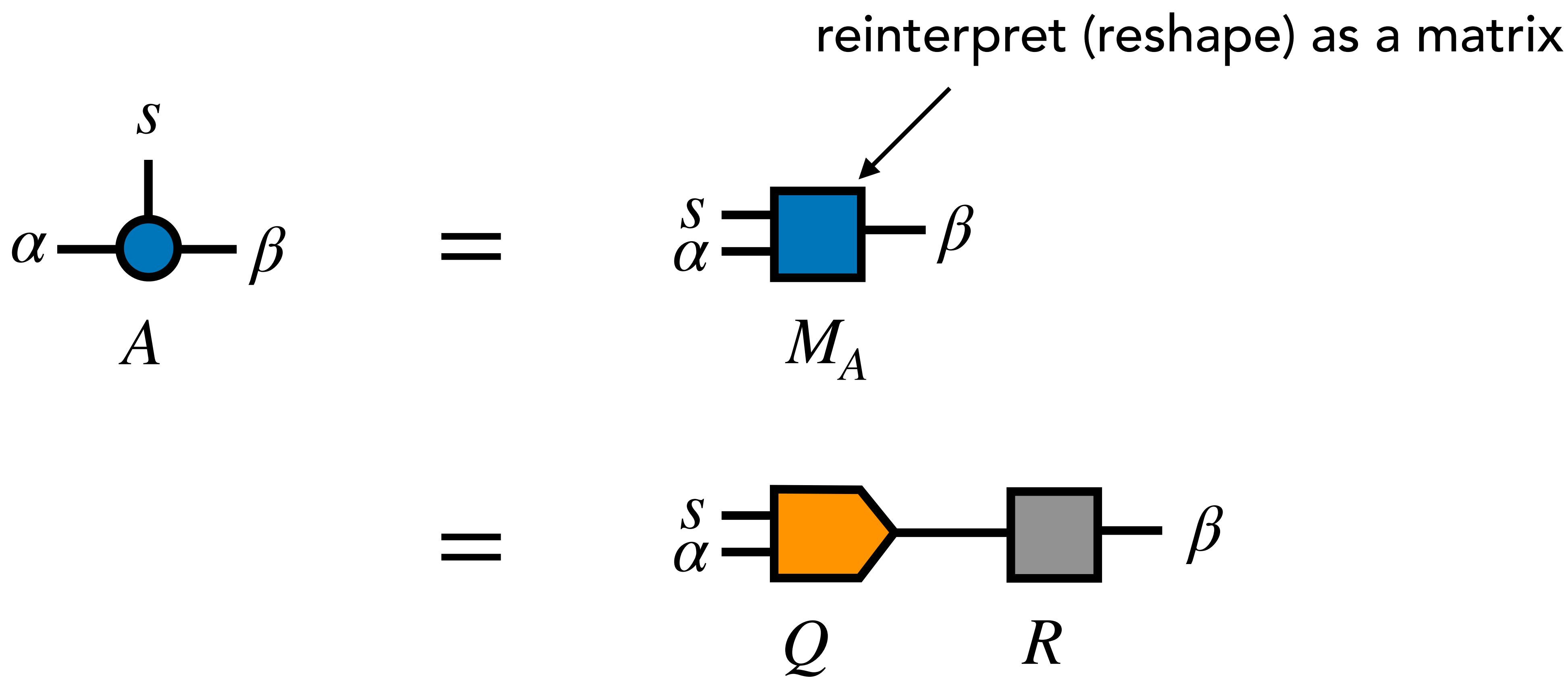
Orthogonal MPS Gauge

Next compute QR decomposition of $A_2' = Q_2 R_2$

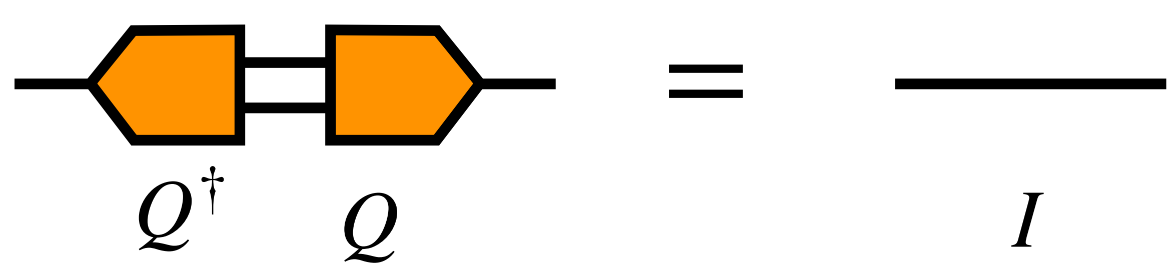


Orthogonal MPS Gauge

Recall QR of a tensor

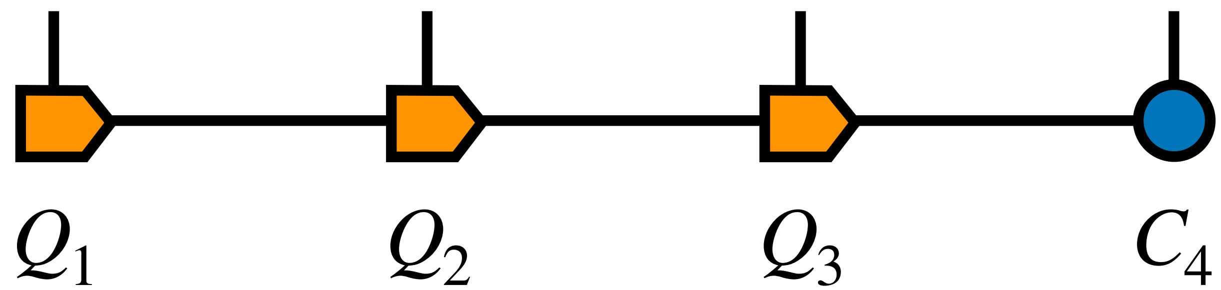


Q is an "isometric embedding" (isometry)



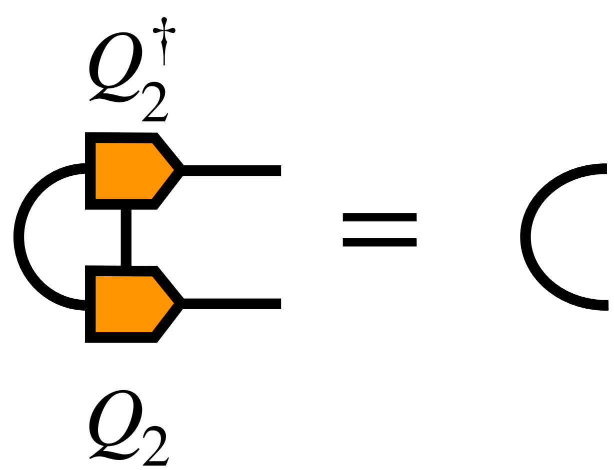
Orthogonal MPS Gauge

Continuing with more QR moves, one obtains



Call this "left orthogonal" gauge

Q tensors have "left orthogonality" property



$$\sum_{s,a} Q^{\dagger s}_{ba} Q^{s}_{ab'} = I_{bb'}$$

Orthogonal MPS Gauge

Challenge question:

Why QR and not SVD?

What advantages/disadvantages might one find?

Orthogonal MPS Gauge

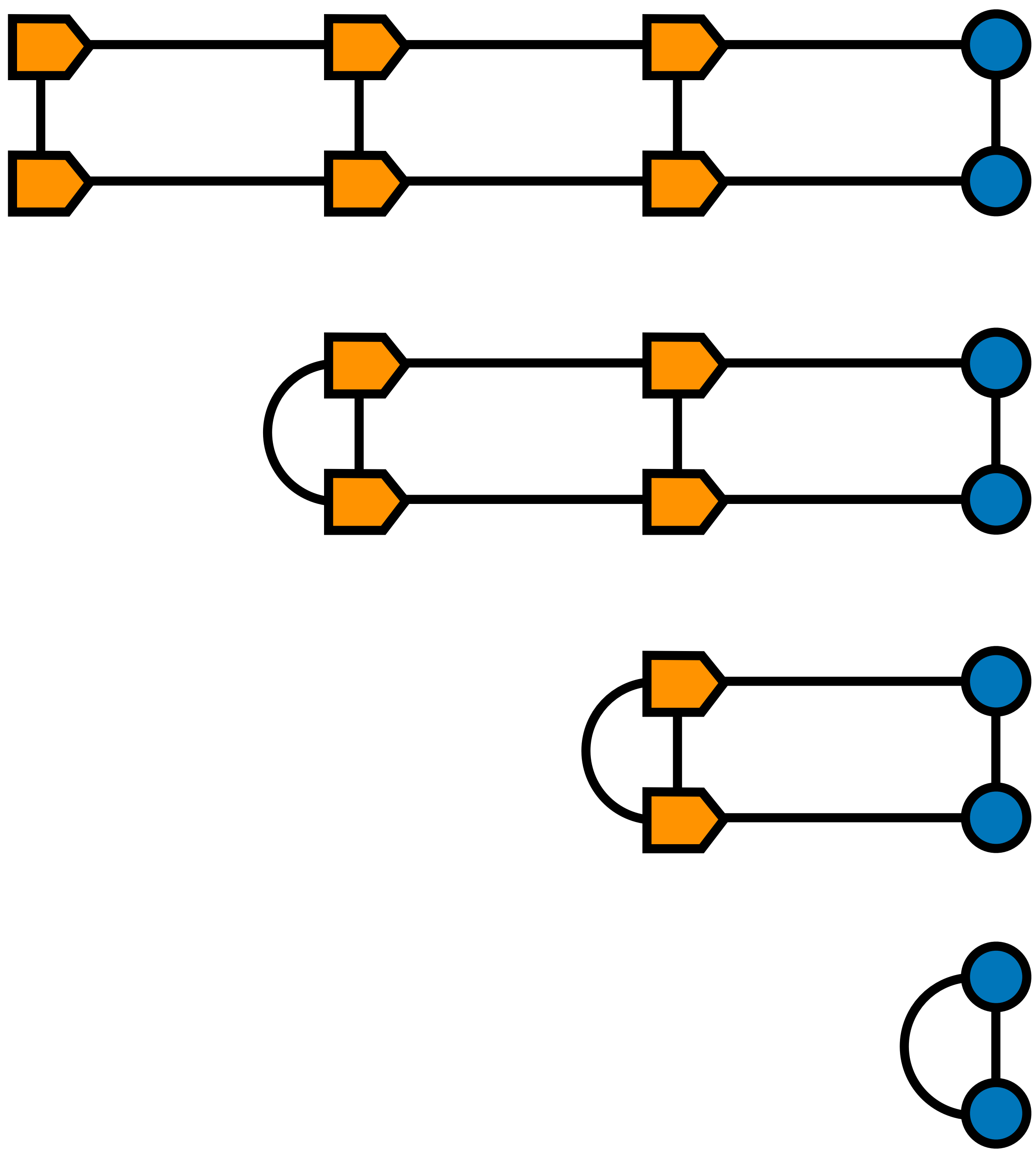
Can deduce certain properties

$\langle\psi|\psi\rangle = $

$=$

$=$

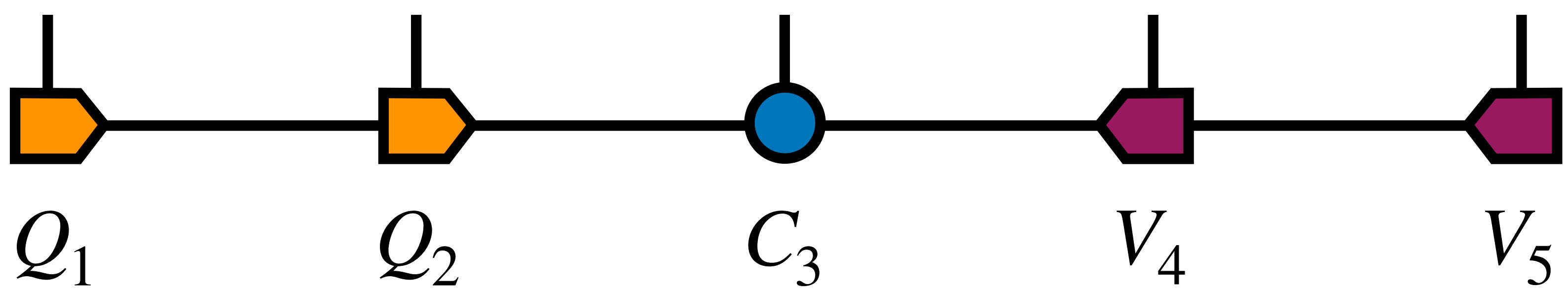
$=$

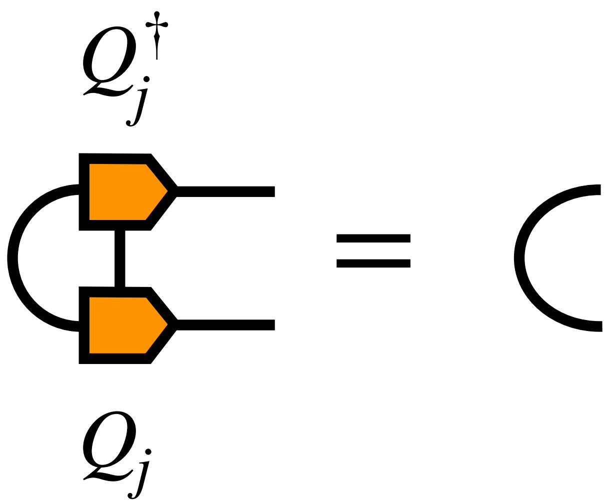


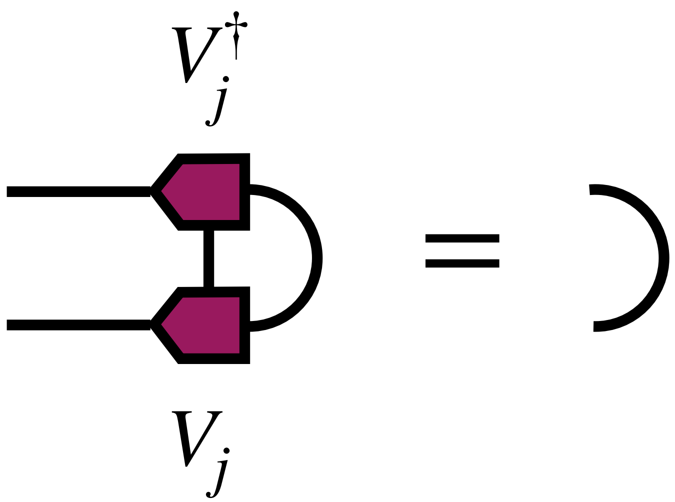
norm of state equals
norm of C_4

Orthogonal MPS Gauge

Can make "mixed" gauges too



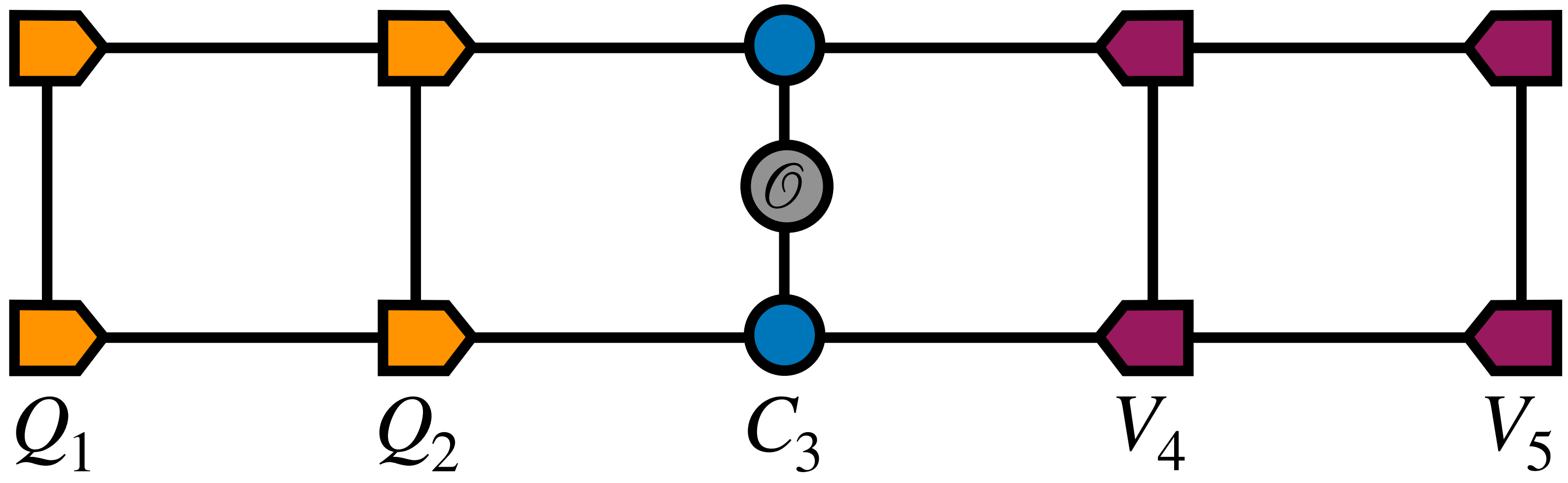




Orthogonal MPS Gauge

Mixed gauges useful for expected values

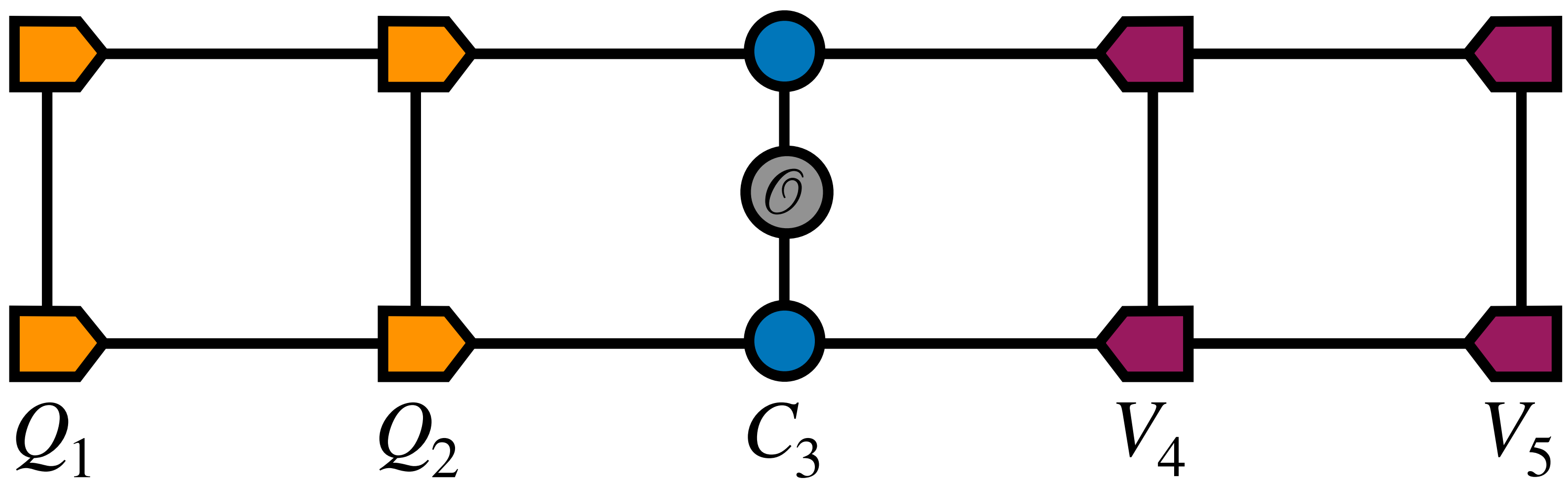
$$\langle \psi | \mathcal{O} | \psi \rangle =$$



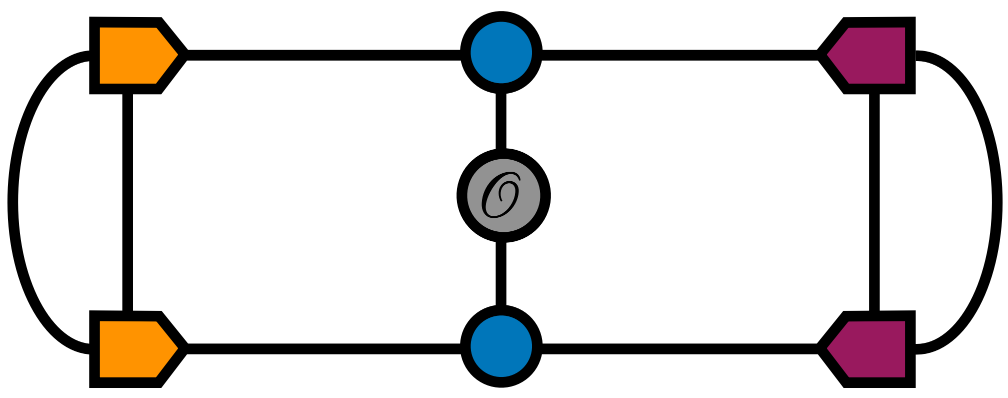
Orthogonal MPS Gauge

Mixed gauges useful for expected values

$$\langle \psi | \mathcal{O} | \psi \rangle =$$



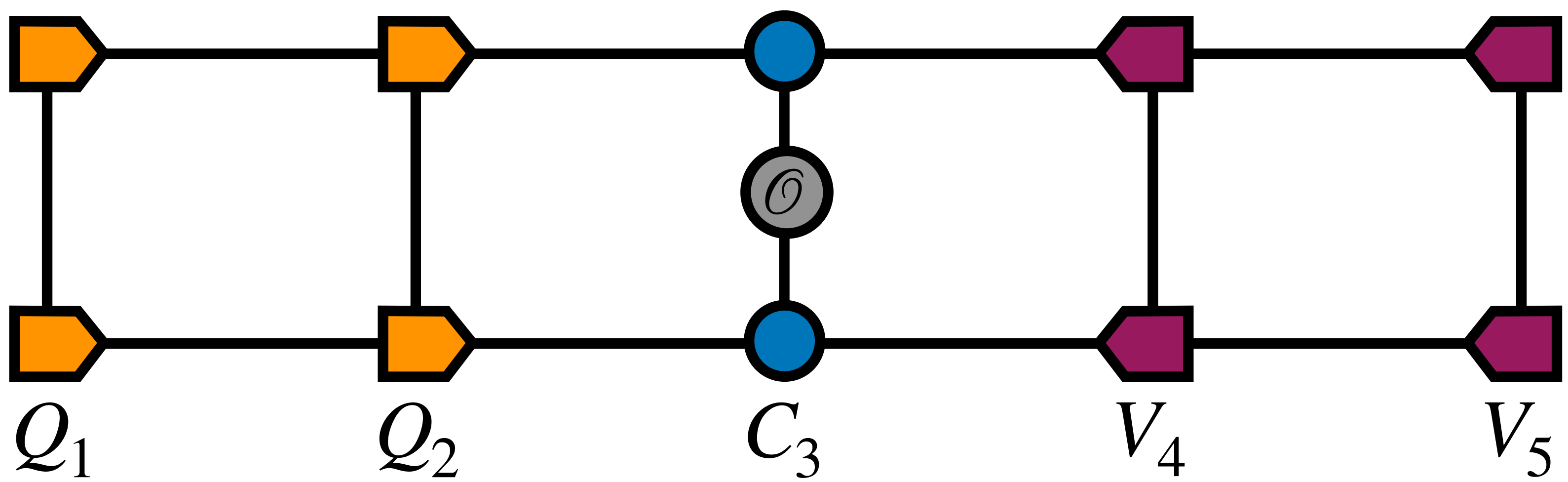
$$=$$



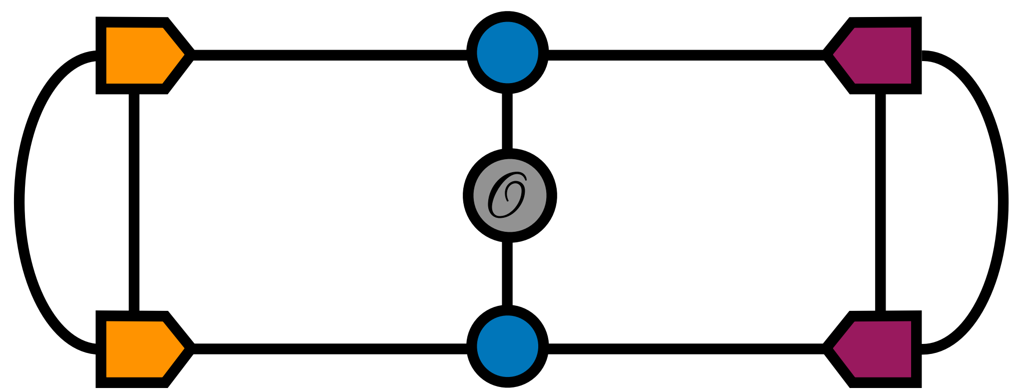
Orthogonal MPS Gauge

Mixed gauges useful for expected values

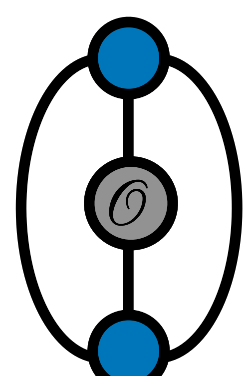
$$\langle \psi | \mathcal{O} | \psi \rangle =$$



$$=$$



$$=$$

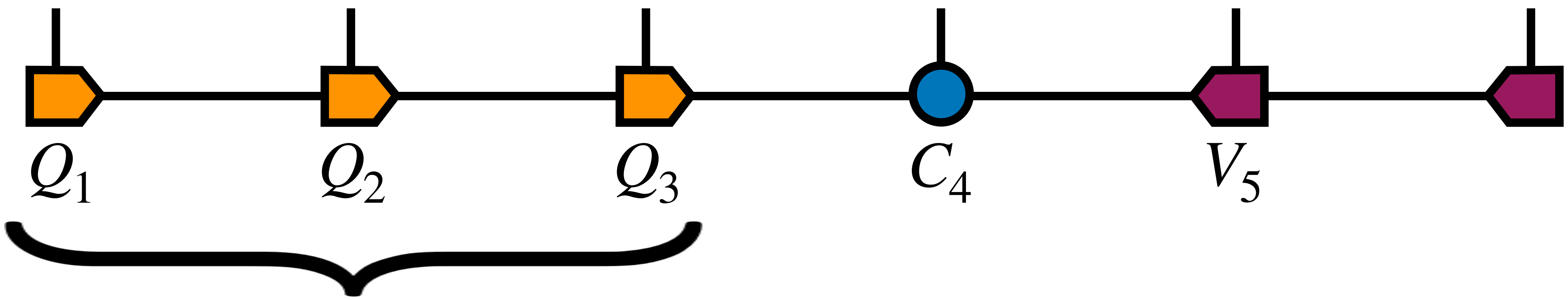


much less computation

Orthogonal MPS Gauge

Mixed gauges also give interesting interpretation of MPS

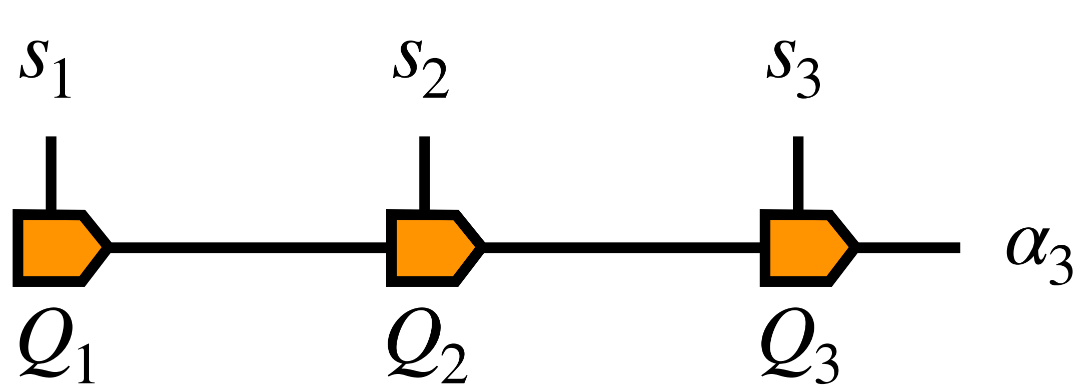
$|\psi\rangle =$



consider left part of 'gauged' MPS

It is an orthonormal basis – why?

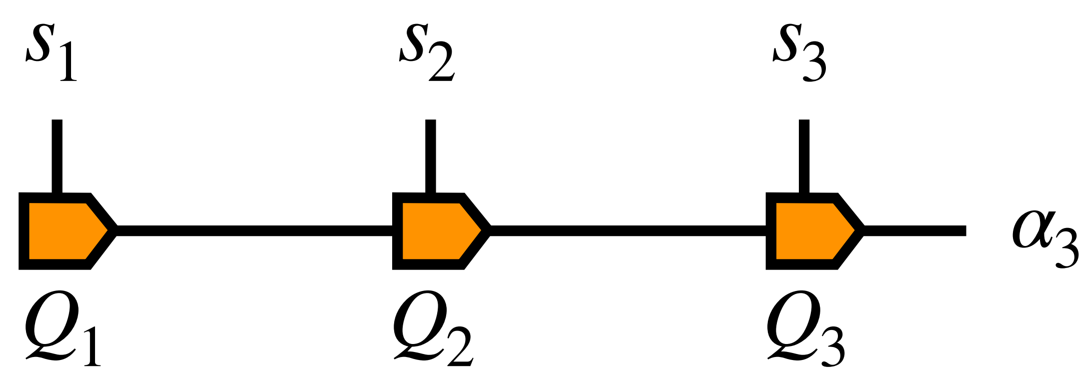
$|\alpha_3\rangle =$



Orthogonal MPS Gauge

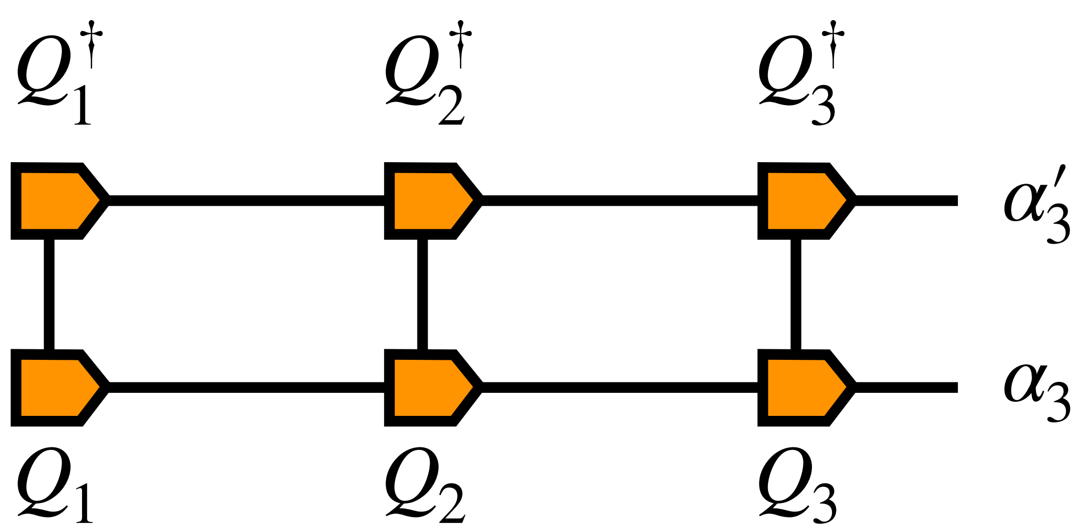
It is an orthonormal basis – why?

$|\alpha_3\rangle =$

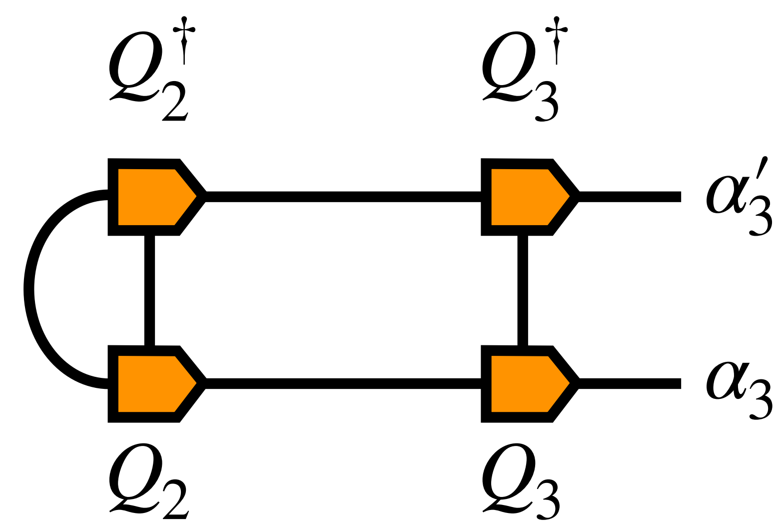


Compute overlap

$\langle \alpha_3' | \alpha_3 \rangle =$



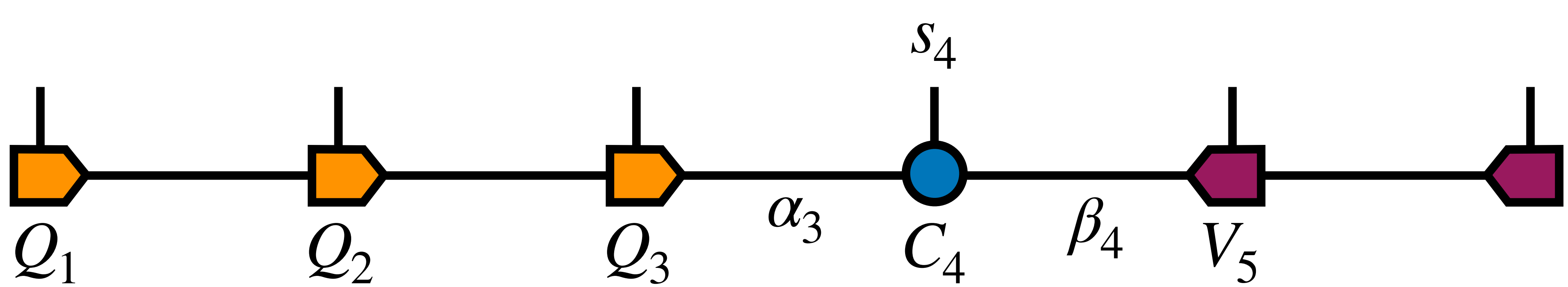
$=$



$= \delta_{\alpha_3' \alpha_3}$ orthonormal in the α_3', α_3 labels

Orthogonal MPS Gauge

Thus entire MPS can be viewed as following

$|\psi\rangle =$ 

$$|\psi\rangle = \sum_{\alpha_3\beta_4} C^{s_4}_{\alpha_3\beta_4} \, |\alpha_3\rangle \, |s_4\rangle \, |\beta_4\rangle$$

In a sense, C_4 is the whole tensor or wavefunction just written in a specific "many-body" basis

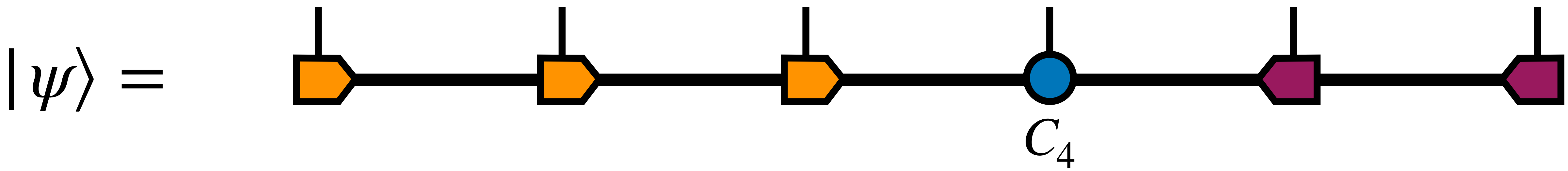
Orthogonal MPS Gauge

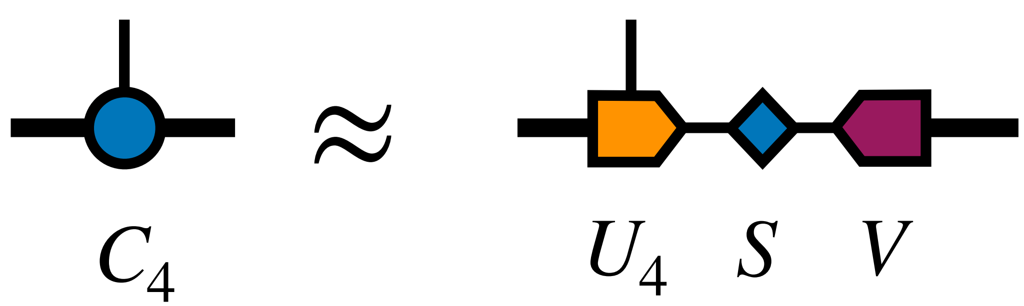
Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension

$|\psi\rangle =$ 

Orthogonal MPS Gauge

Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension

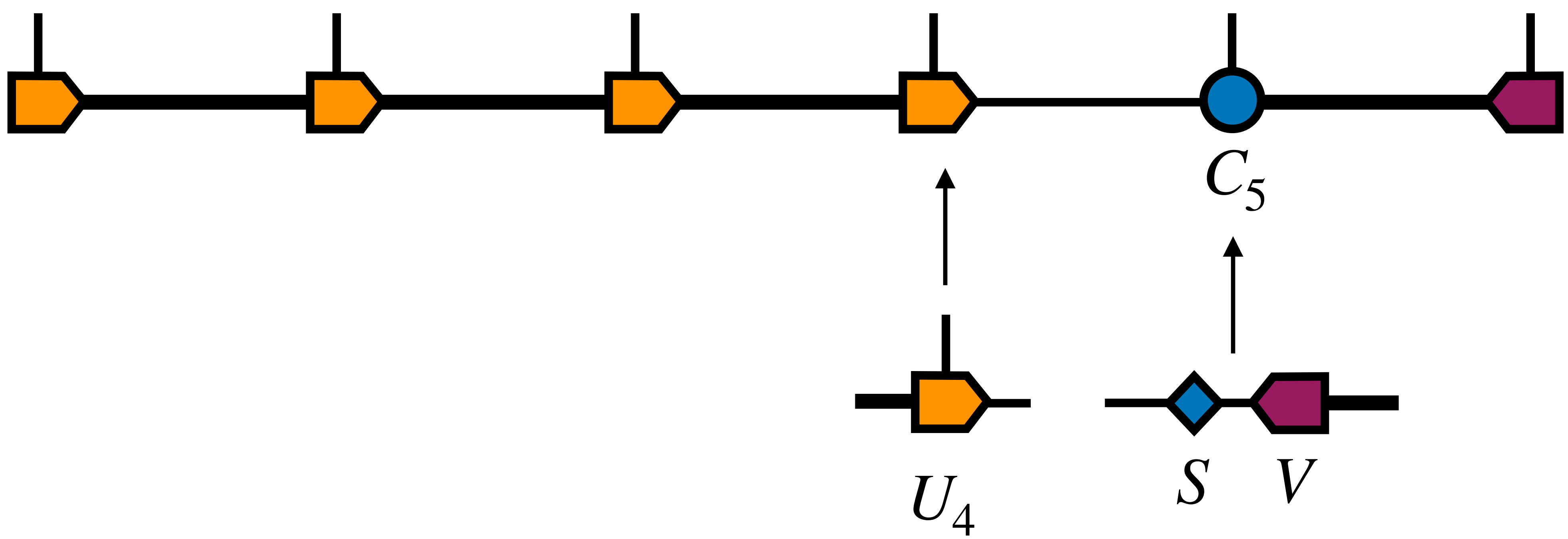




Perform SVD of C_4 tensor – truncate small singular values

Orthogonal MPS Gauge

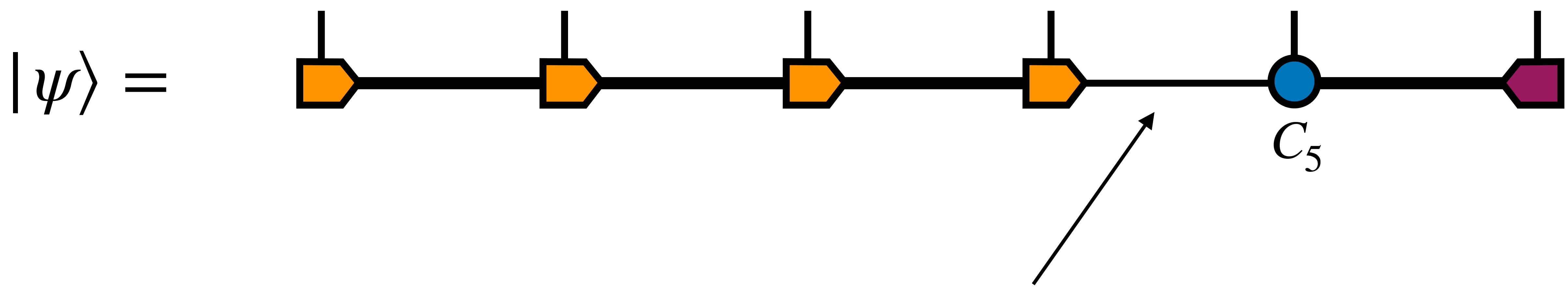
Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension

$|\psi\rangle =$ 

Leave "U" behind, merge S*V to form new "center"

Orthogonal MPS Gauge

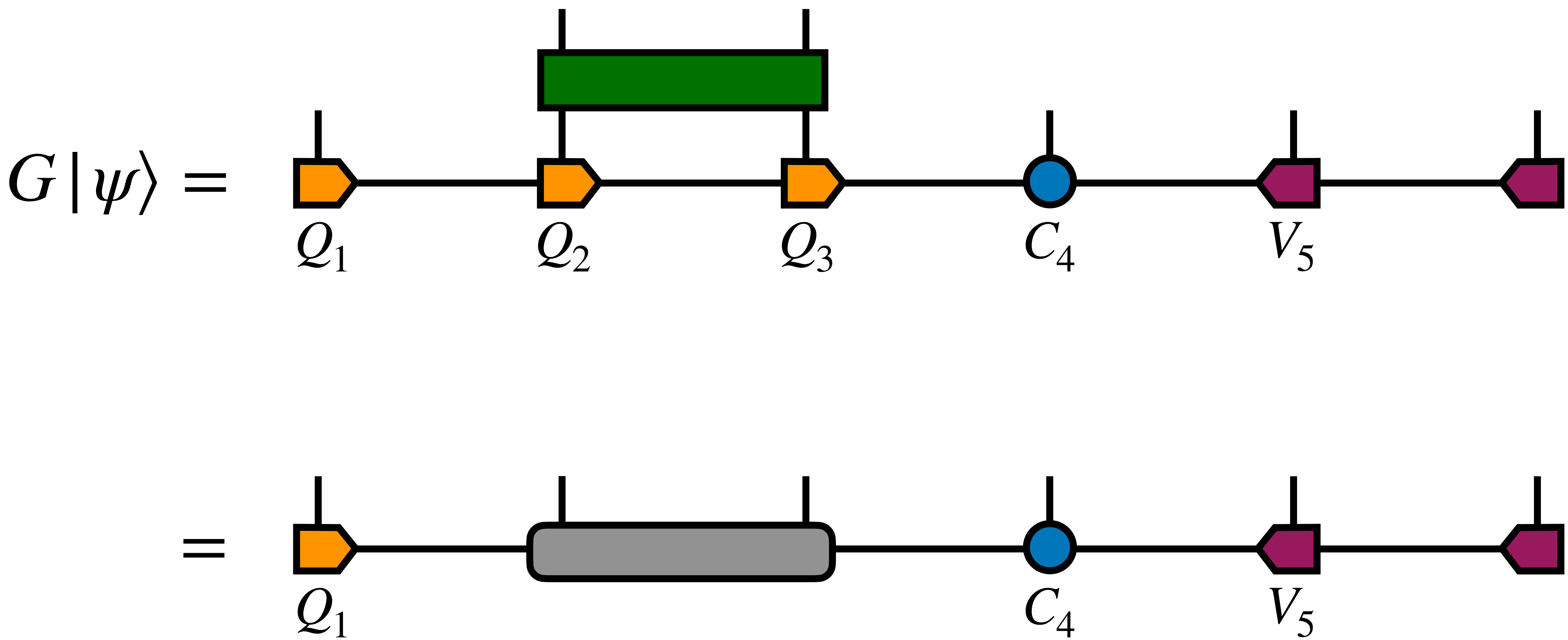
Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension



Afterward, bond size has been reduced

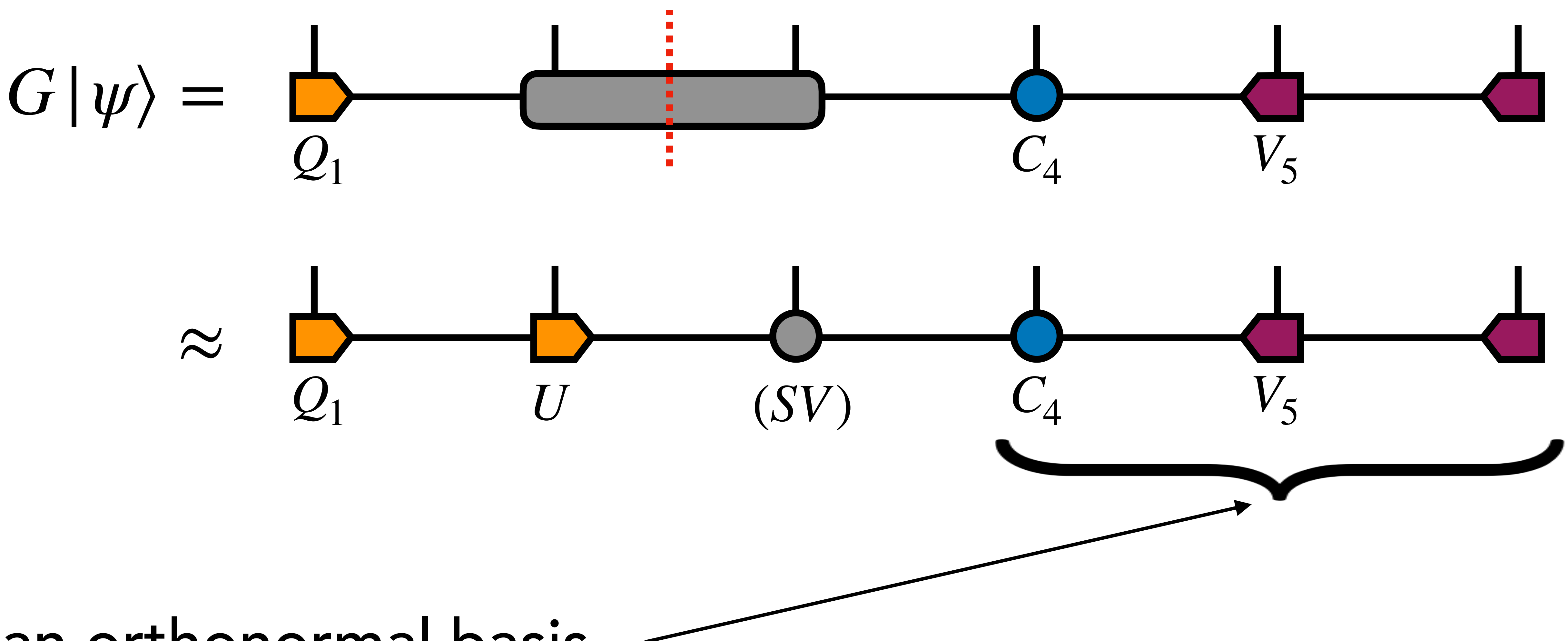
Importantly, truncation must happen within orthogonal gauge "center" – why?

Say we apply a gate away from the center



So far it is exact – no error

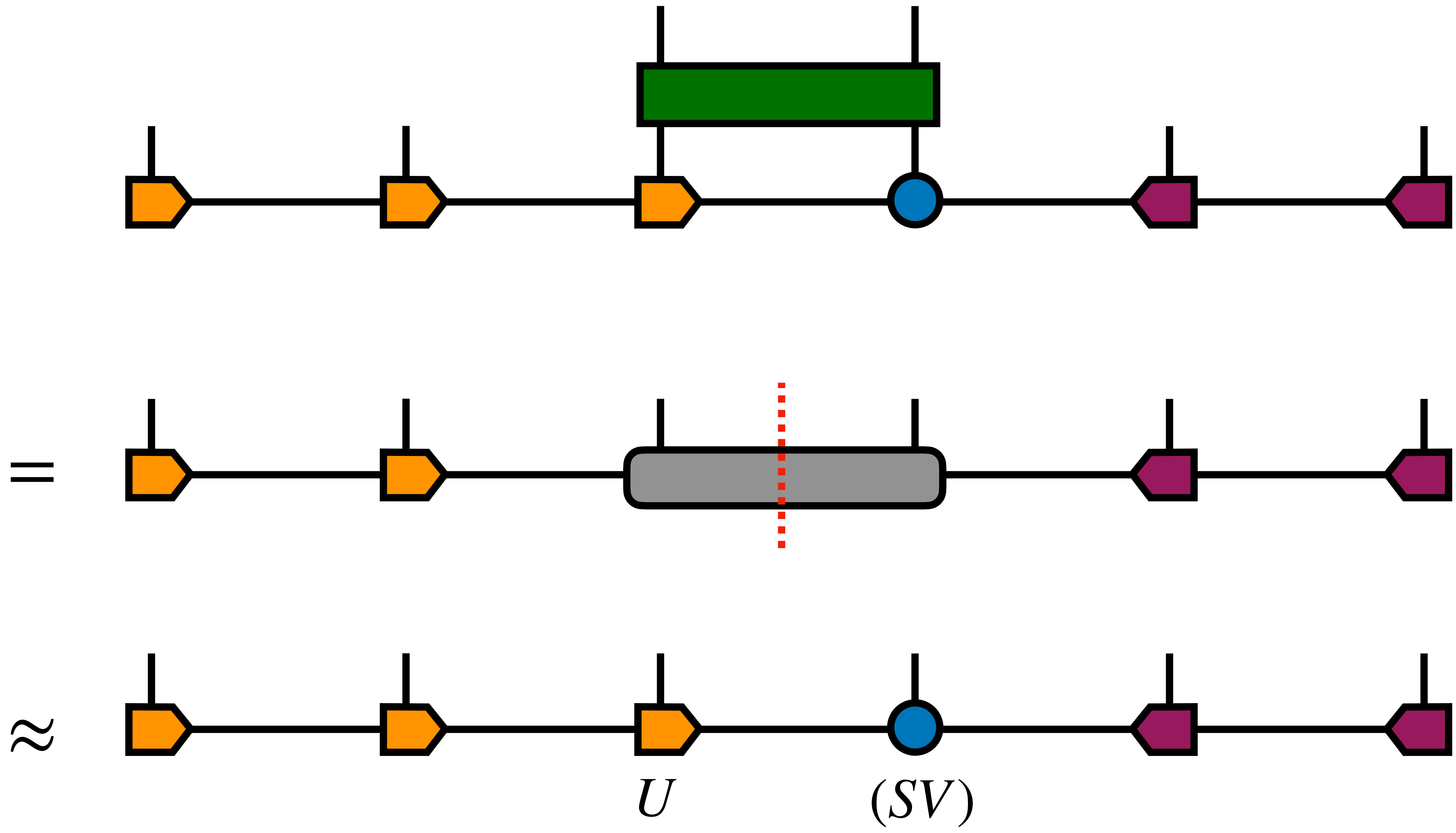
Now we can SVD & truncate small singular values



Not an orthonormal basis

Might make an arbitrarily large error

In contrast, truncating inside an orthonormal basis is controlled



local SVD error = global error