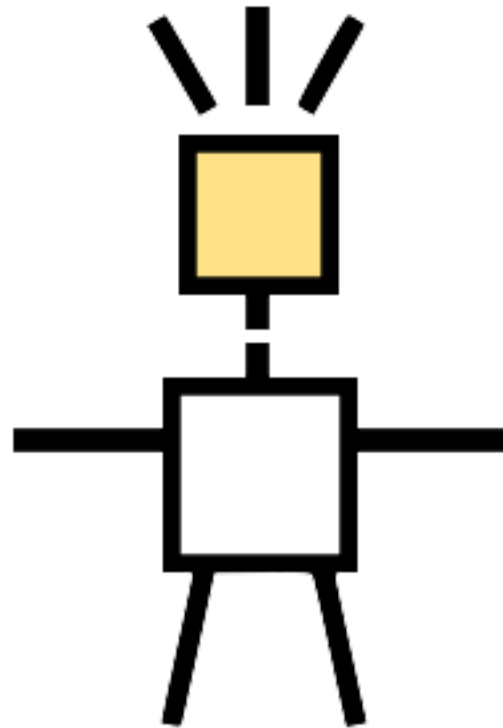


Welcome to the ITensor CCQ School



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 you!

Please ask questions if
anything is unclear...

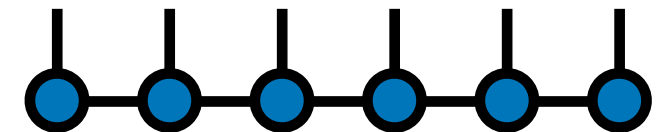
Schedule

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

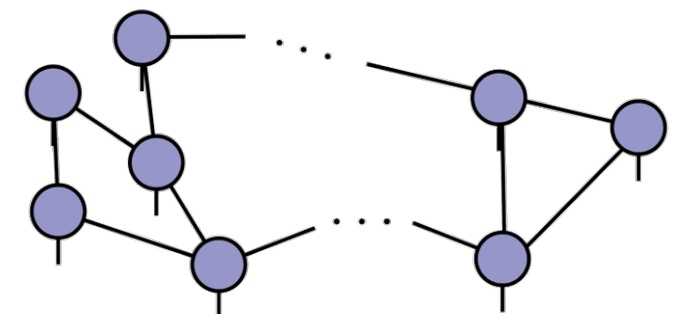
$$e^{-iHt} |\psi(0)\rangle = |\psi(t)\rangle$$

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

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

Amplitudes form a big tensor!

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


The diagram shows a blue rounded rectangle representing a tensor. Six vertical lines extend upwards from the top edge of the rectangle, each labeled with an index: s_1 , s_2 , s_3 , s_4 , s_5 , and s_6 from left to right.

Tensor Networks

What is a tensor?

vector

$$v = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$v_2 = 3$$

matrix

$$M = \begin{bmatrix} 5 & 7 \\ 8 & 9 \end{bmatrix}$$

$$M_{12} = 7$$

order-3
tensor

$$T = \begin{bmatrix} 3 & \begin{bmatrix} 5 & 4 \end{bmatrix} & 7 \\ 1 & \begin{bmatrix} 3 & 2 \end{bmatrix} & 5 \end{bmatrix}$$

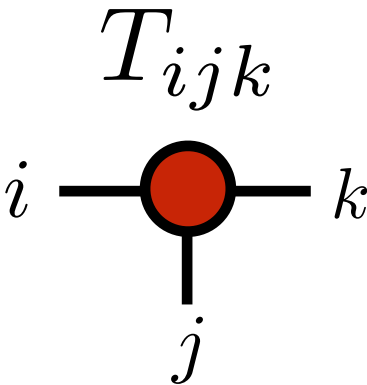
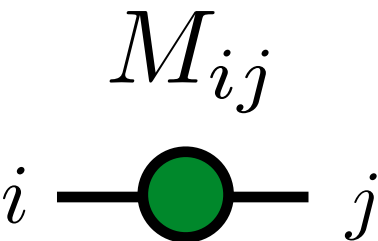
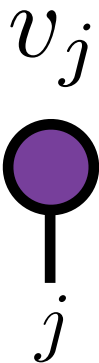
$$T_{112} = 5$$

Tensor Networks

N-index tensor = shape with N lines

$$T^{s_1 s_2 s_3 \cdots s_N} = \text{Diagram of a tensor with } N \text{ lines labeled } s_1, s_2, s_3, s_4, \dots, s_N$$

Low-order examples:




Joining wires means contraction:



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

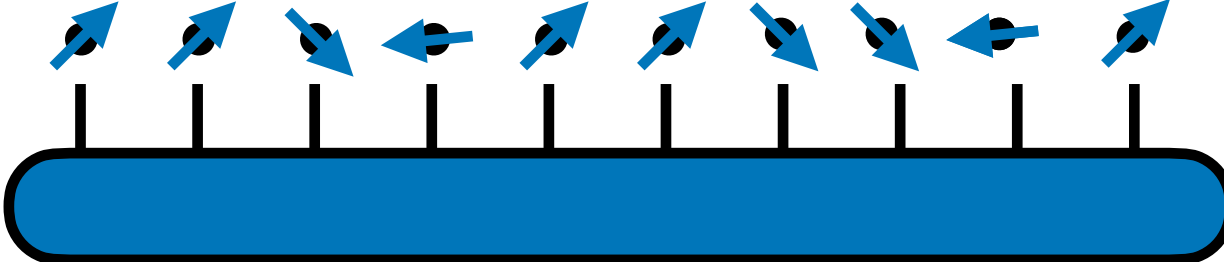
Tensor Networks

A problem: N-index tensor **exponential** to store

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


The diagram shows a light gray rounded rectangle representing a tensor. Ten vertical lines extend upwards from the top edge of the rectangle. Above the first four lines are the labels s_1 , s_2 , s_3 , and s_4 . Above the fifth line is a dot, followed by two more dots, then another dot, and finally the label s_N above the tenth line.

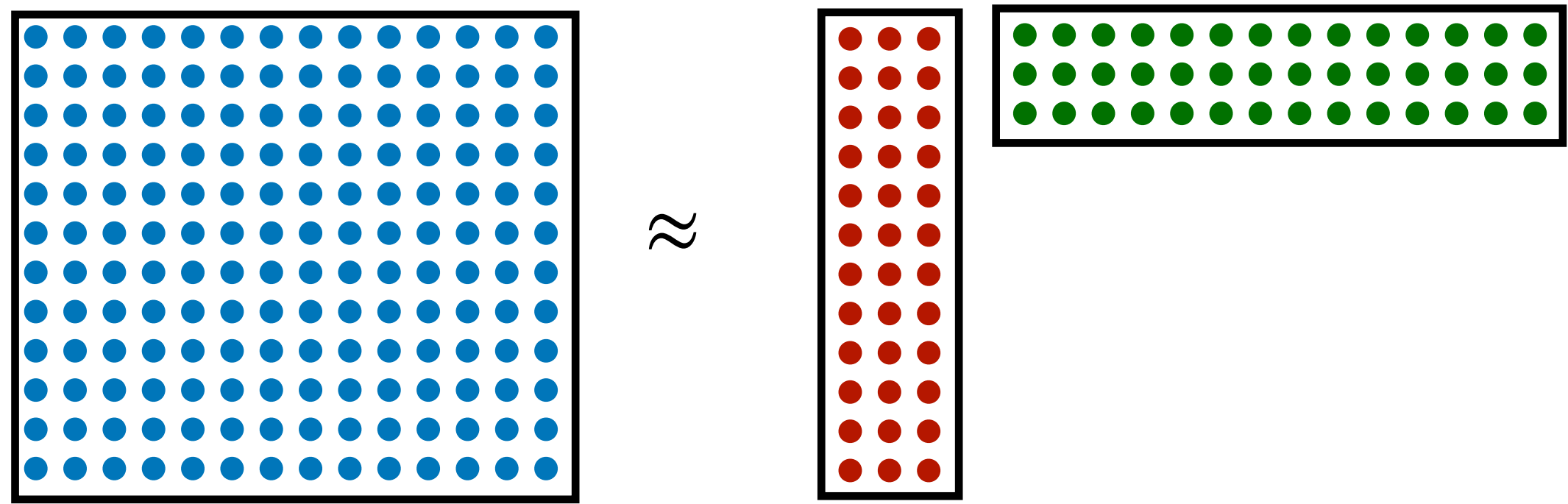
A version of the "**many-body problem**"

$$|\Psi\rangle =$$


The diagram shows a blue rounded rectangle representing a many-body state. Ten vertical lines extend upwards from the top edge of the rectangle. At the end of each line is a black dot with a blue arrow pointing in a different direction, representing a spin or state of a particle.

Tensor Networks

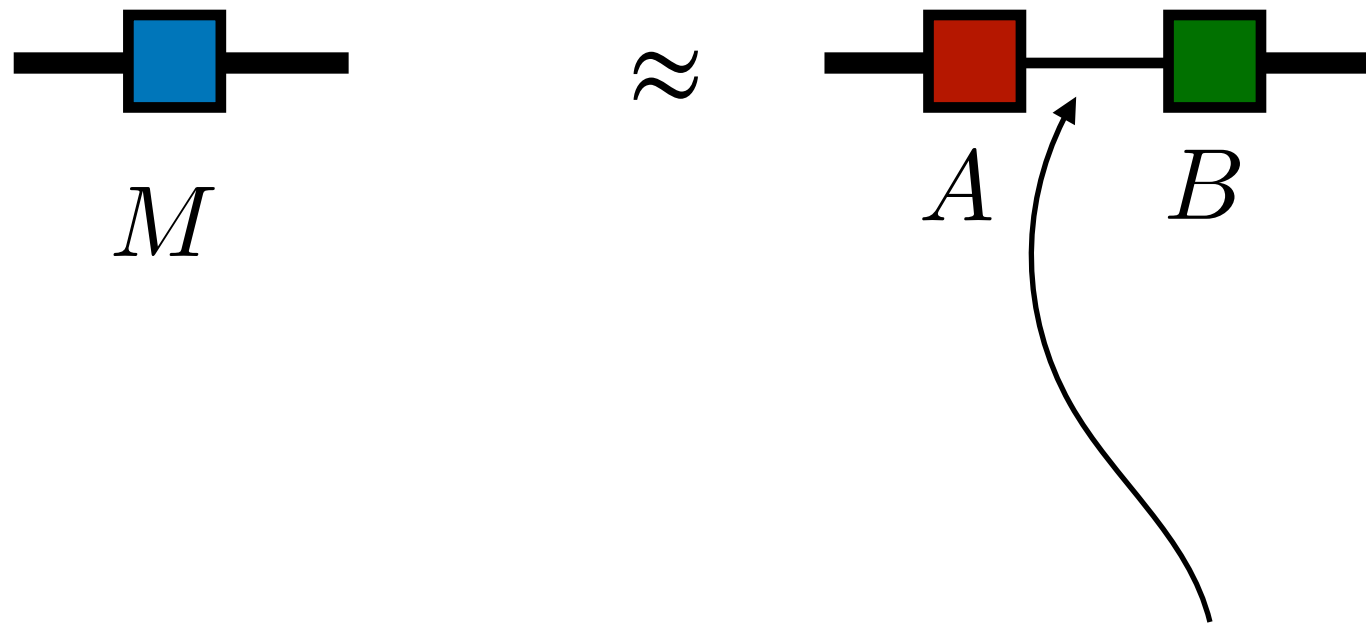
For matrices, can perform low-rank compression



Less memory and fewer operations

Tensor Networks

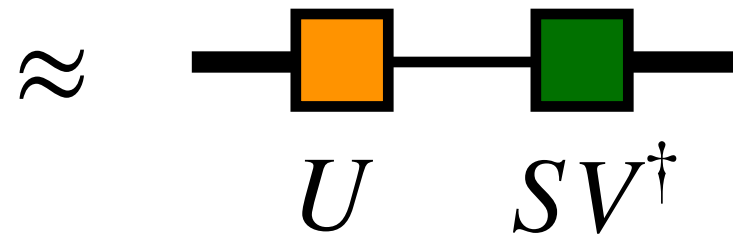
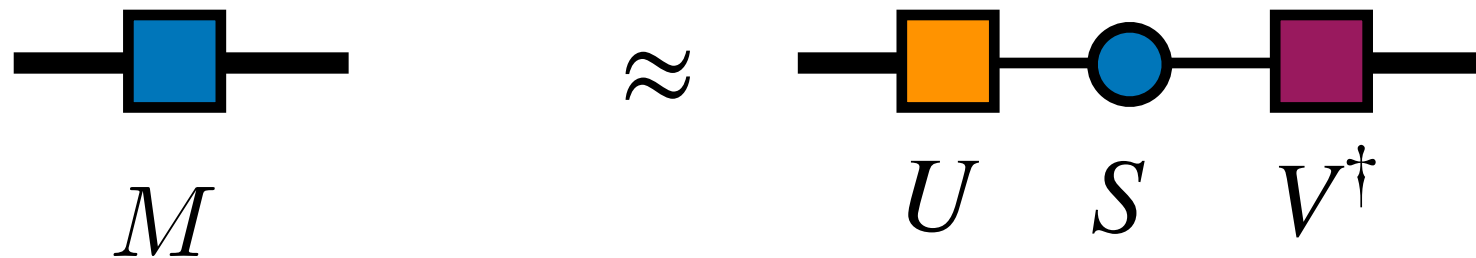
For matrices, can perform low-rank compression



*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?

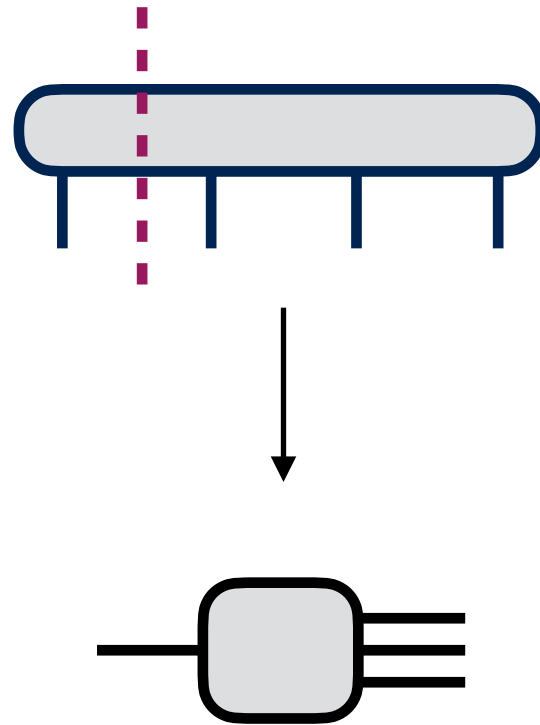
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

High rank (full rank)

All columns orthogonal / maximally linearly independent

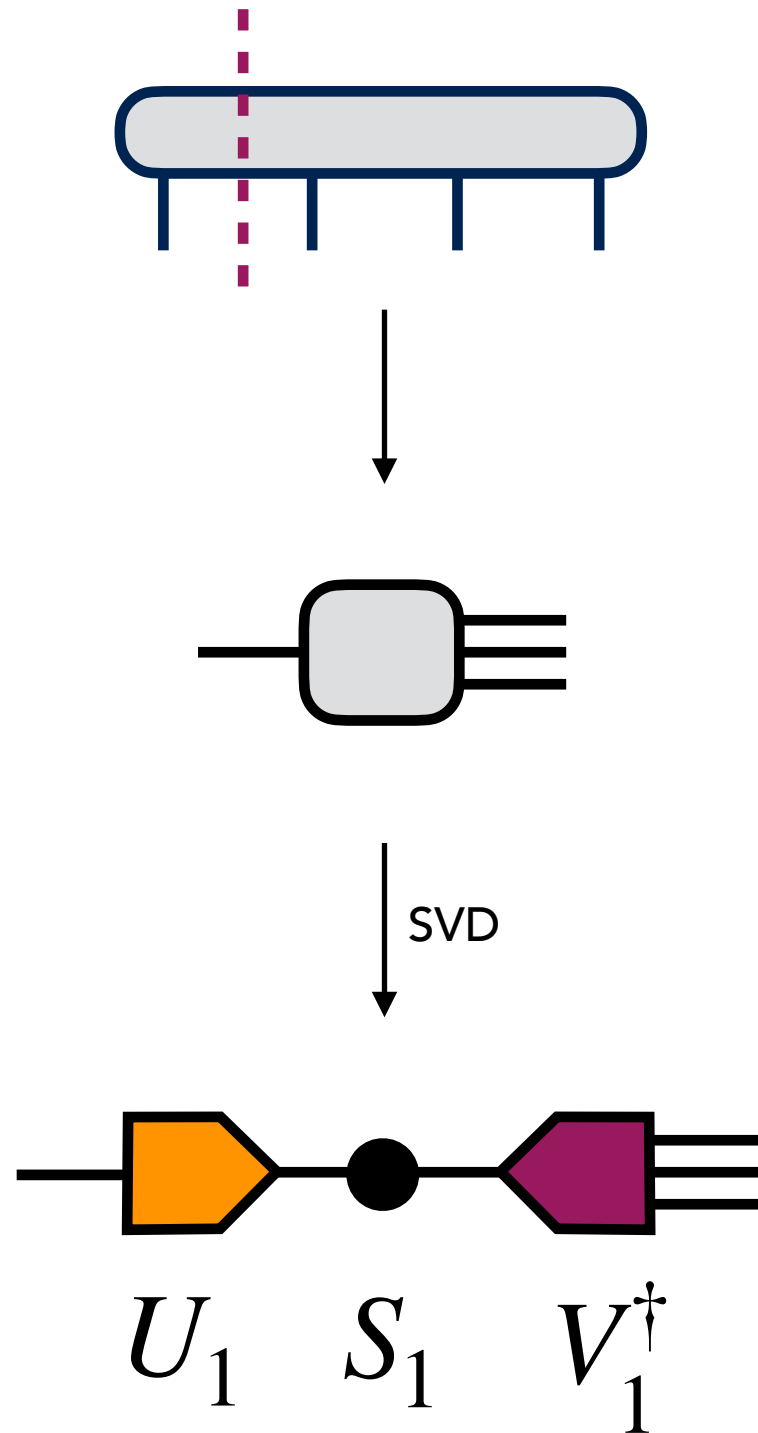
How to generalize SVD to tensors?

Reshape as a matrix:



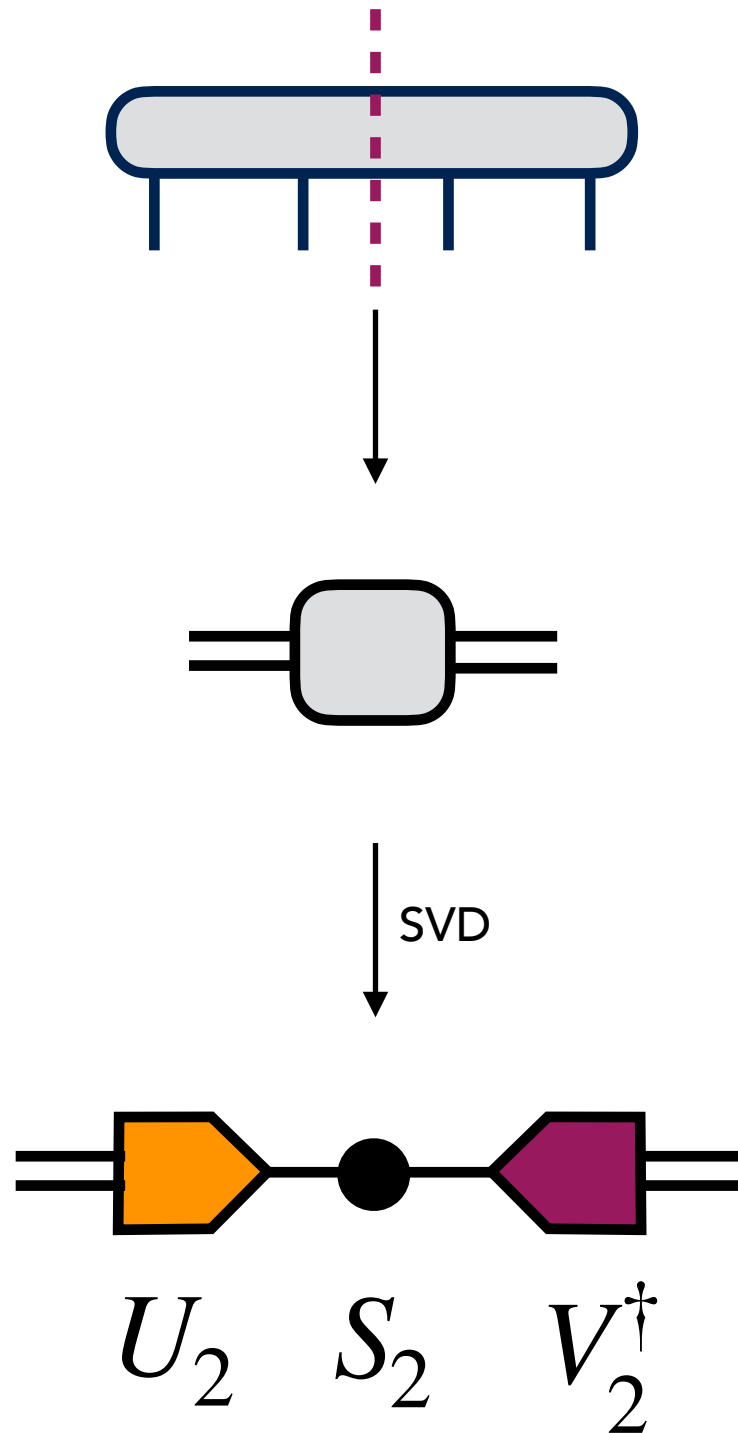
Generalizing SVD to tensors

Reshape as a matrix:



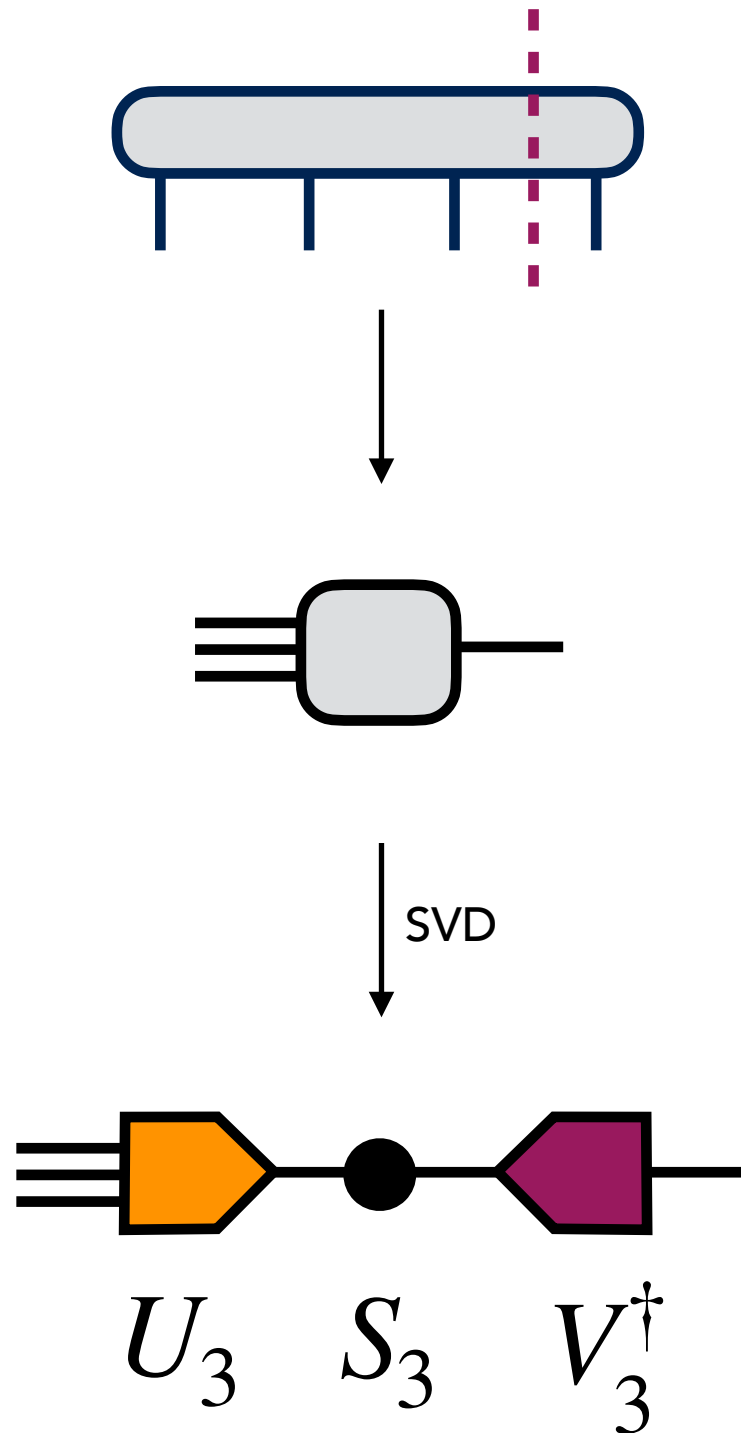
Generalizing SVD to tensors

Other partitions:

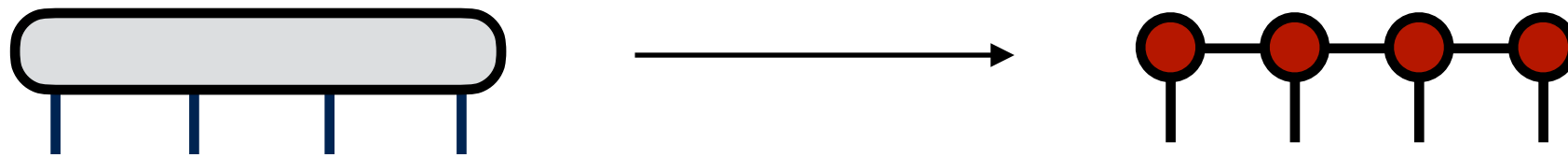


Generalizing SVD to tensors

Other partitions:

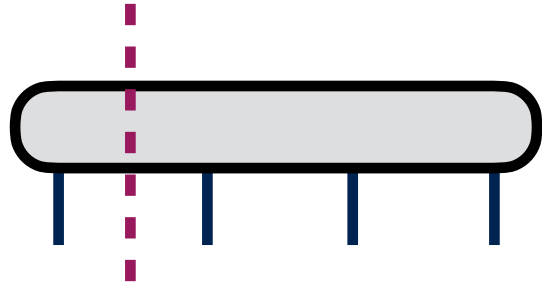


Factorizations lead to tensor networks

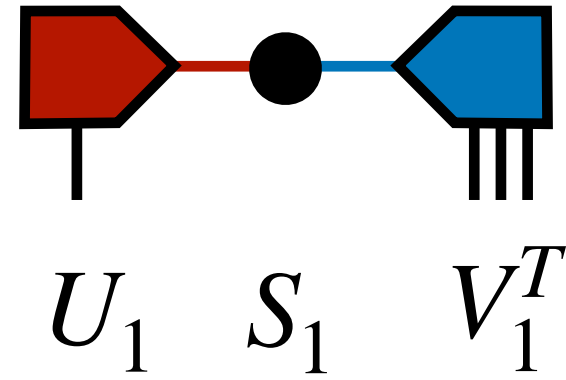


Consider sequence of SVD's...

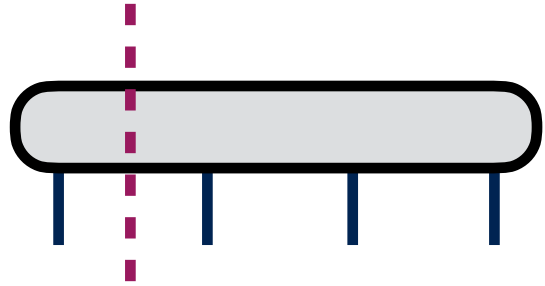
1. SVD first index from rest



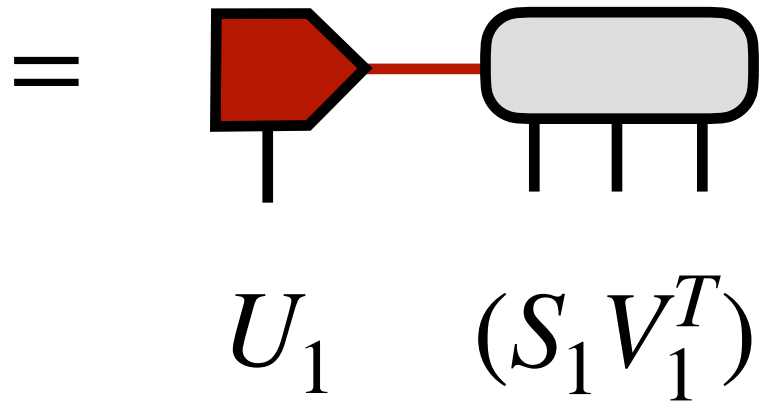
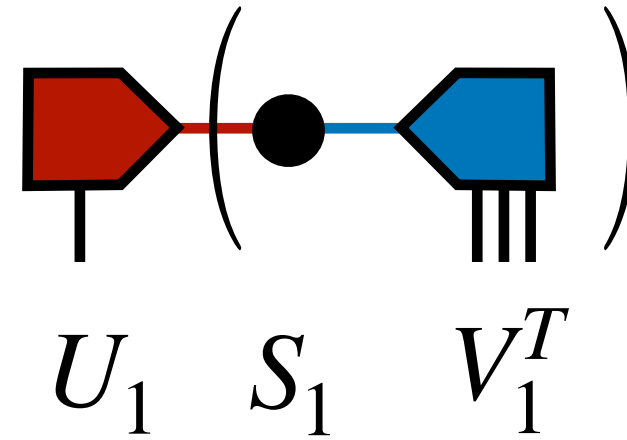
SVD
 \approx



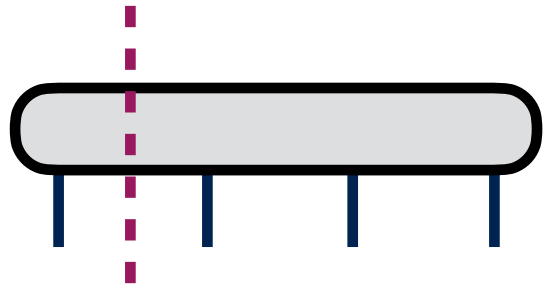
2. Multiply S_1 into V_1^T



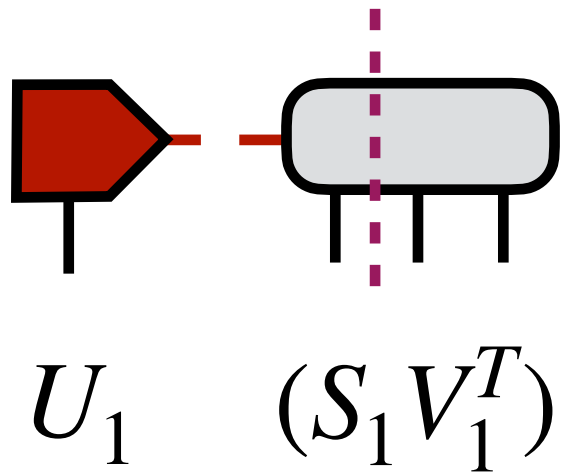
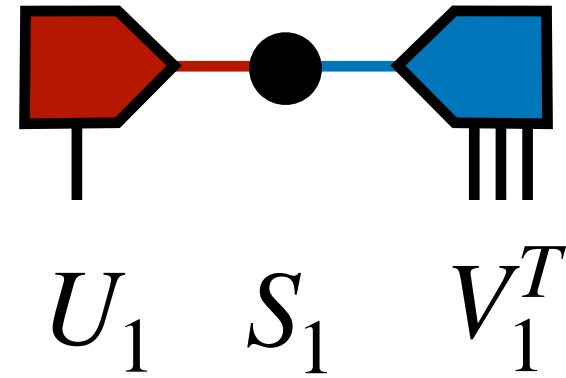
SVD
 \approx



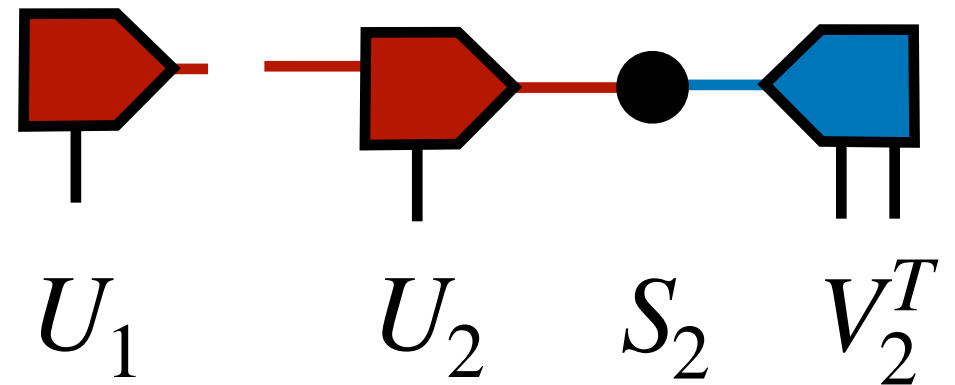
3. SVD this new tensor $(= S_1 V_1^T)$



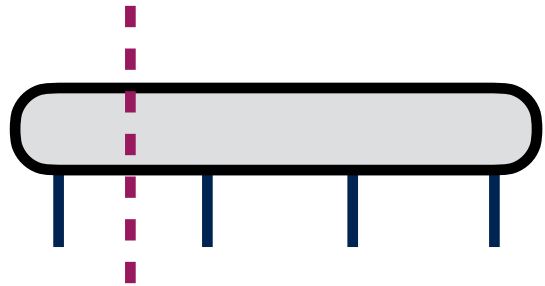
SVD
 \approx



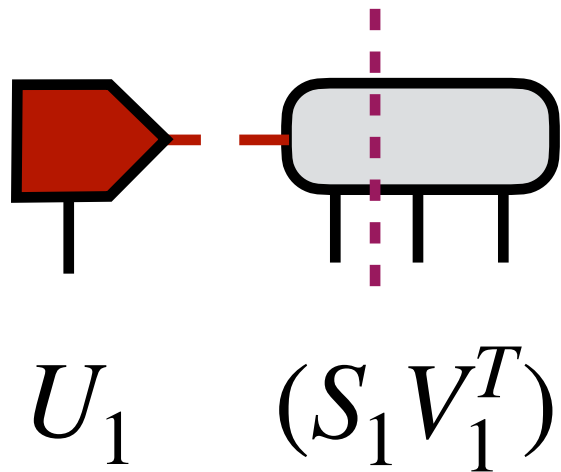
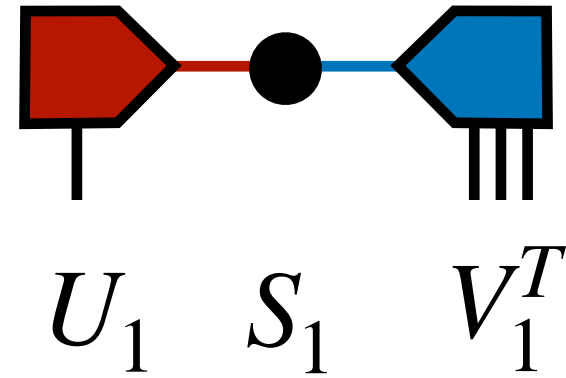
SVD
 \approx



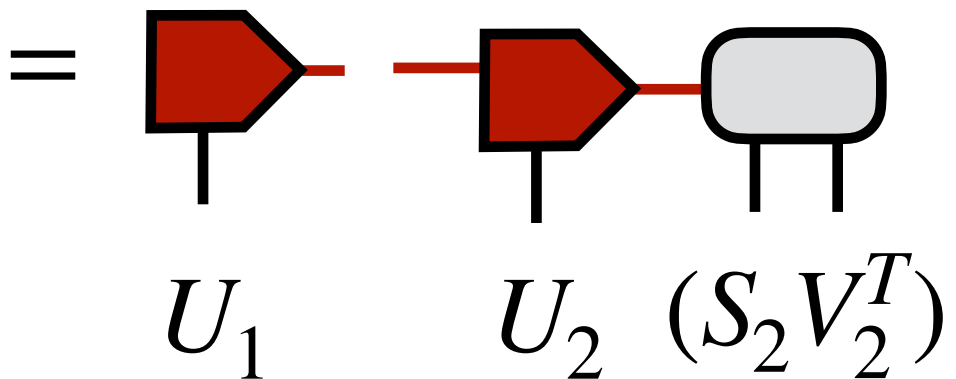
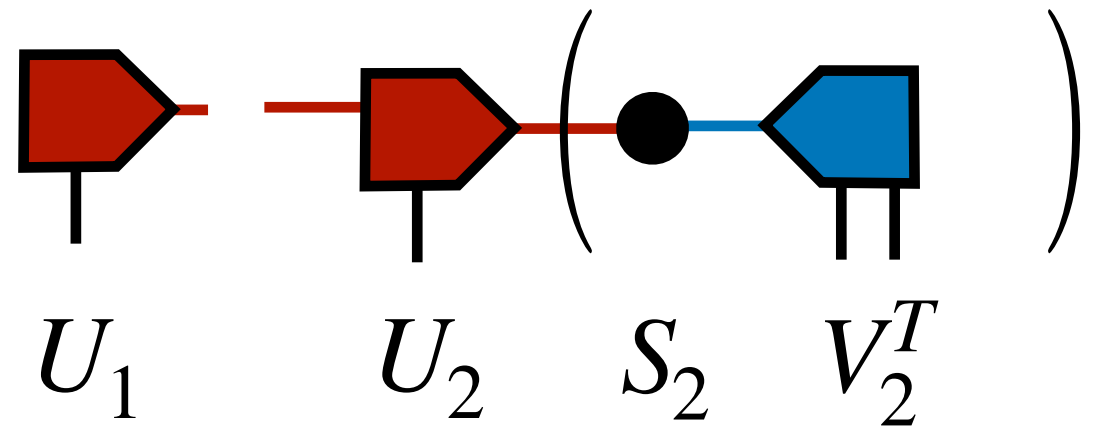
4. Multiply S_2 into V_2



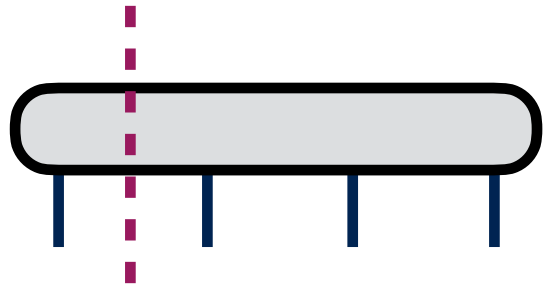
SVD
 \approx



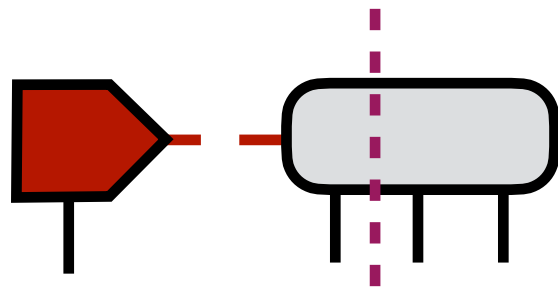
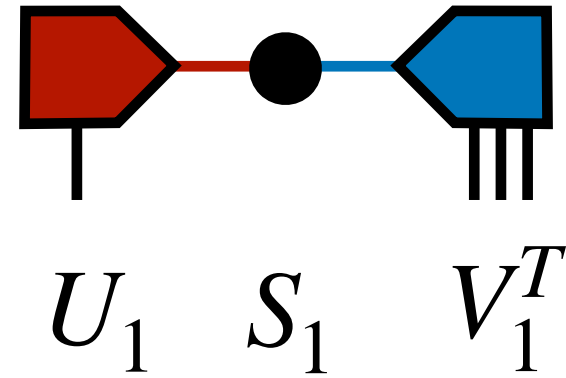
SVD
 \approx



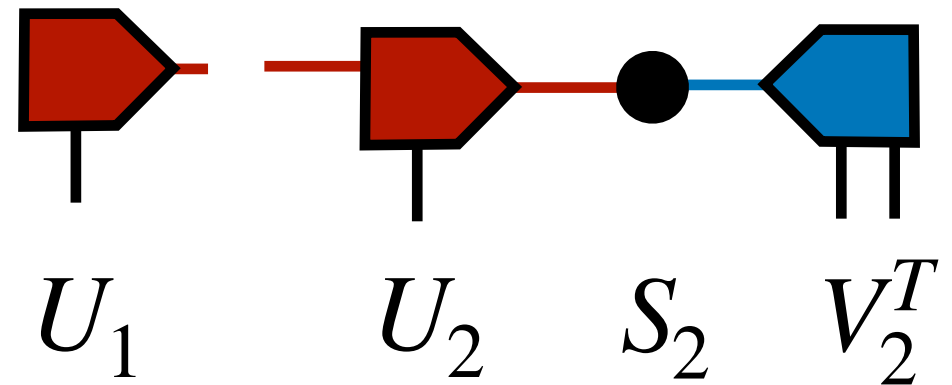
5. Finally SVD ($S_2 V_2^T$)



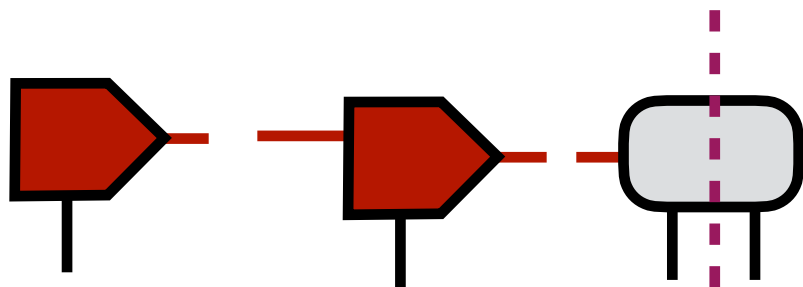
SVD
 \approx



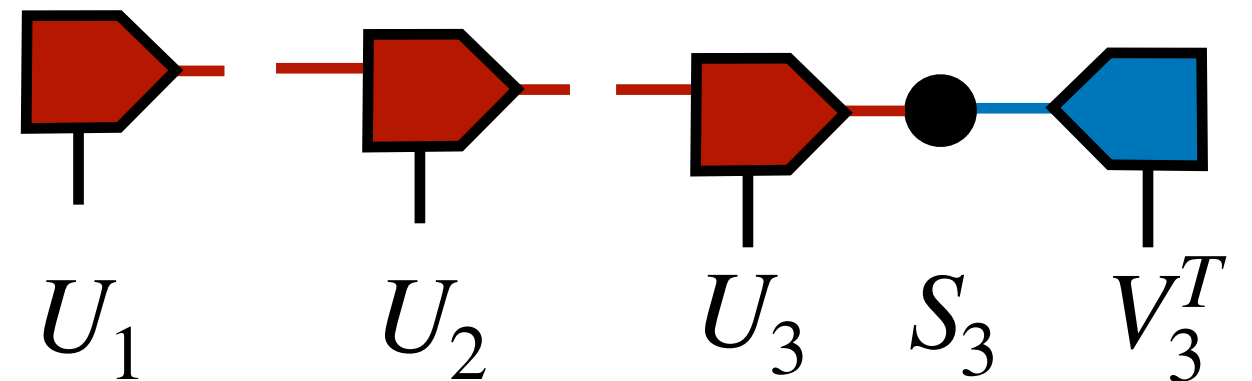
SVD
 \approx



$U_1 \quad (S_1 V_1^T)$

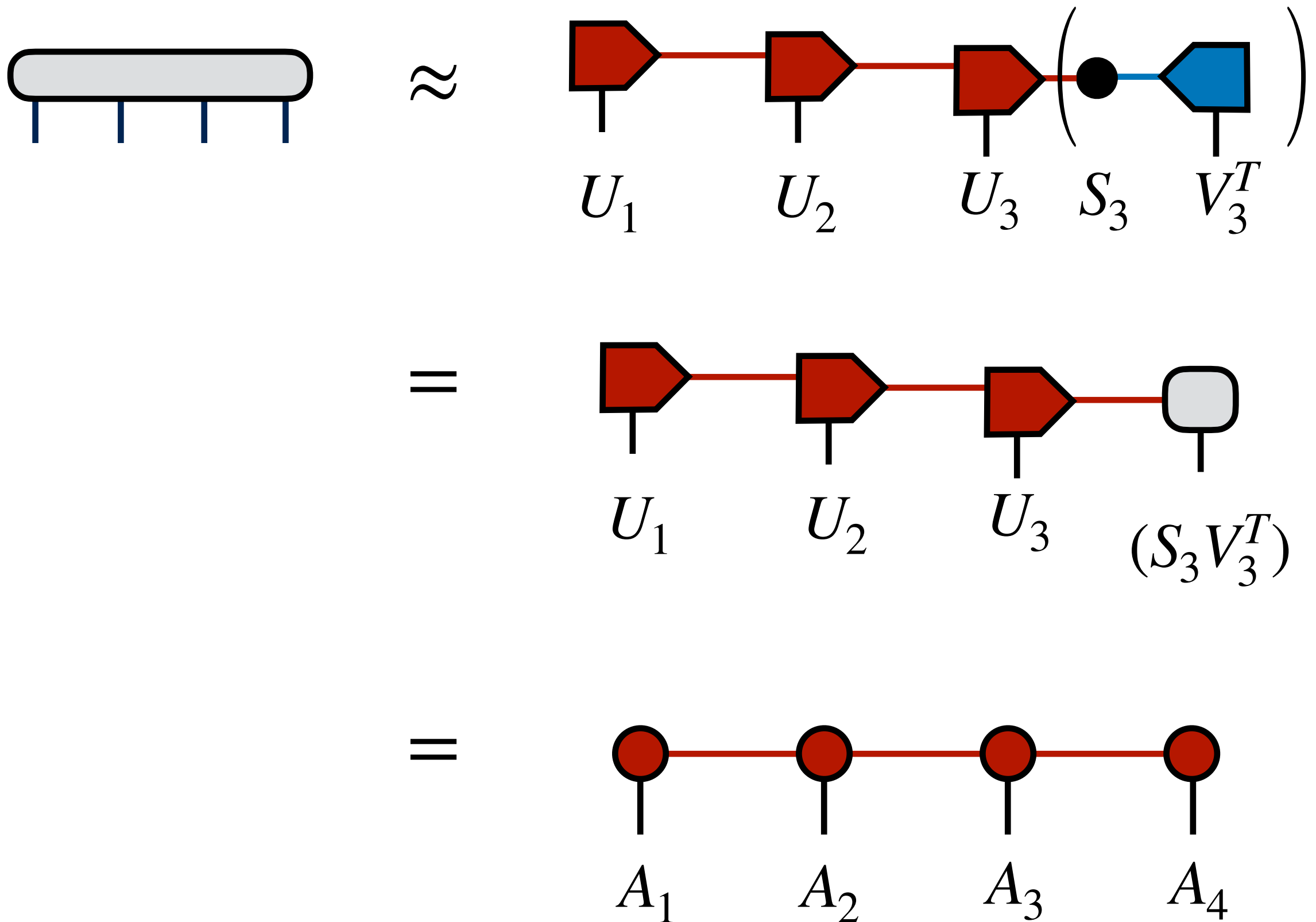


SVD
 \approx

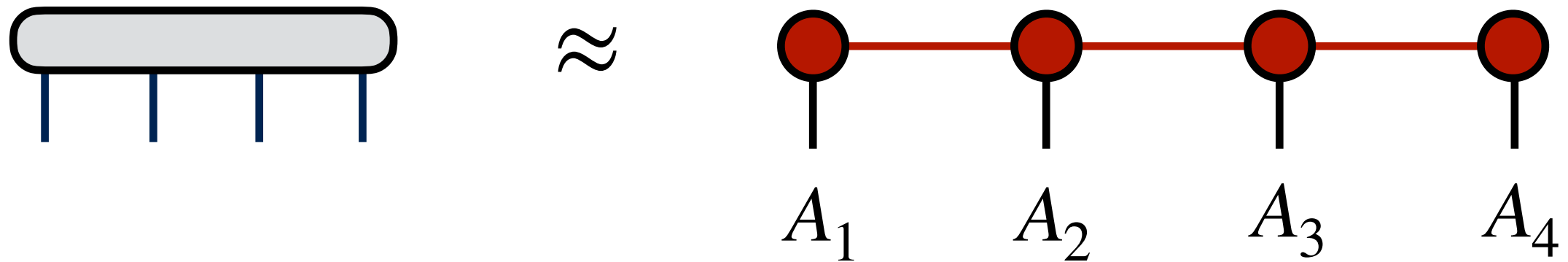


$U_1 \quad U_2 \quad (S_2 V_2^T)$

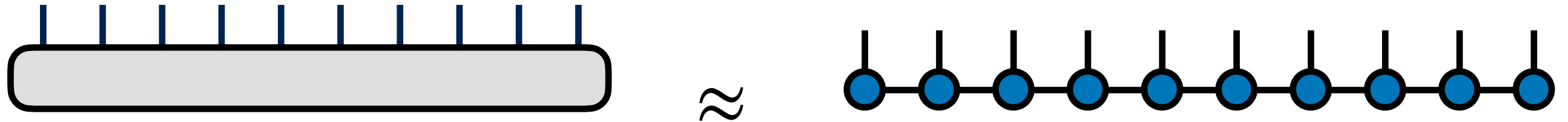
6. Interpret result as a tensor network



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



Two views of matrix product states



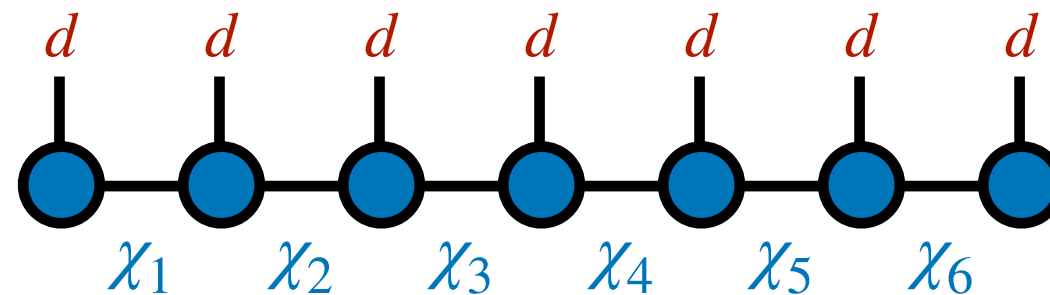
factorization of a tensor



*special subset of tensors
(low-rank tensor manifold)*

Matrix product state (MPS) tensor network

Size of a matrix product state (MPS)



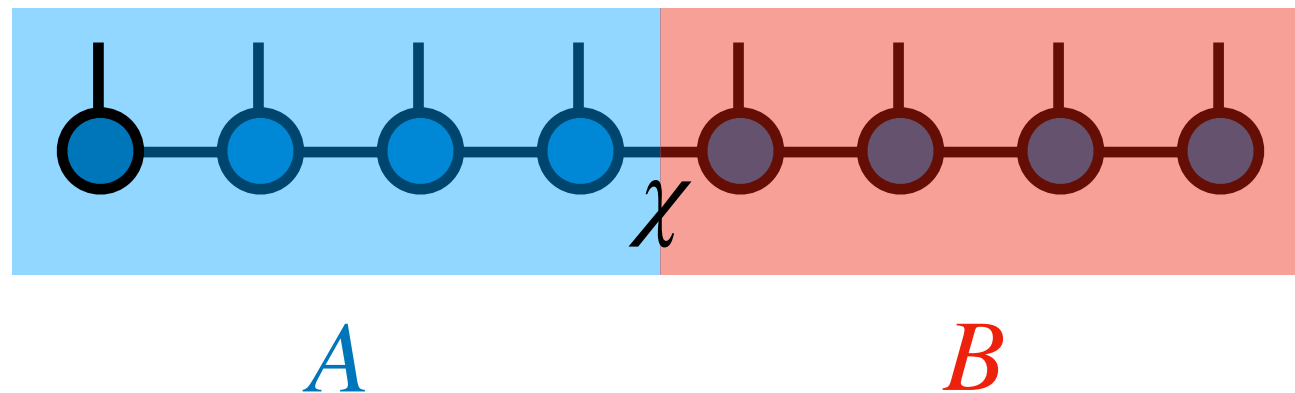
N tensors of size $\chi \cdot d \cdot \chi = d\chi^2$

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

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)

$$U|\Psi\rangle = \text{Diagram 1} \approx \text{Diagram 2}$$

The diagram shows the application of a quantum gate U to a state $|\Psi\rangle$. On the left, a horizontal chain of five blue circular tensors is shown. An orange rectangular box, representing the gate U , is positioned above the third and fourth tensors, with vertical lines connecting it to both. This is followed by an approximation symbol \approx . On the right, the resulting state is shown as a horizontal chain of five circular tensors. The first two are blue, the next two are green, and the last one is blue. The green tensors are connected by a thick black line, indicating a contraction.

"Perfect sampling" from MPS tensor networks

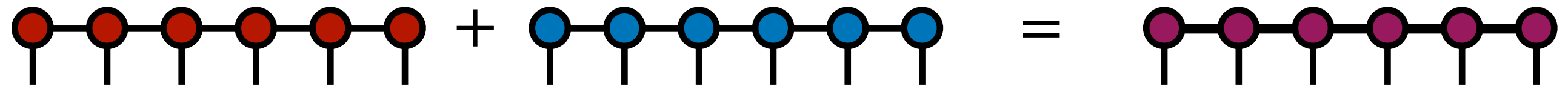
$$\text{Diagram 3} = \text{Diagram 4}$$

The diagram illustrates "Perfect sampling" from an MPS tensor network. On the left, a tensor ρ_1 is shown as two vertically stacked blue circles. A grey die is positioned above it. Two arrows point from the tensor to the right, labeled $|0\rangle$ and $|1\rangle$. This is followed by an equals sign. On the right, the resulting state is shown as a horizontal chain of five circular tensors. The first tensor is a box containing $|1\rangle$ above and $\langle 1|$ below. The second tensor is a teal circle, and the third, fourth, and fifth are blue circles. The teal tensor is connected to the first box by a thick black line, indicating a contraction.

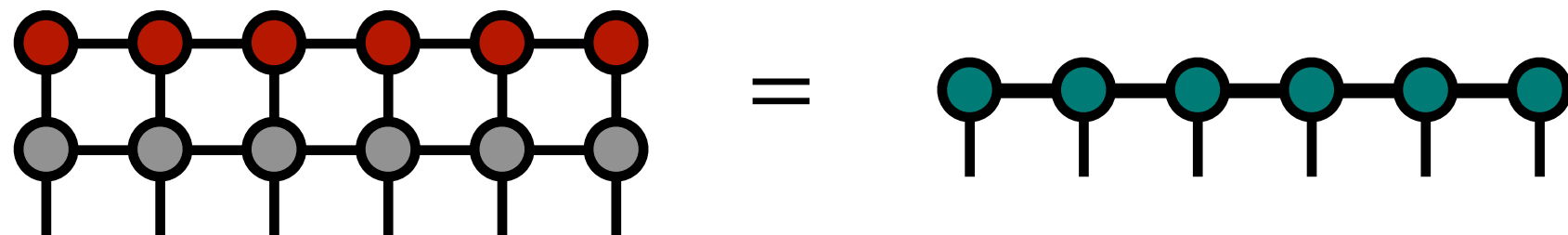
Tensor Networks

Many efficient algorithms for tensor networks

Summing MPS in compressed form



Multiply by other networks (MPO operator \times MPS)

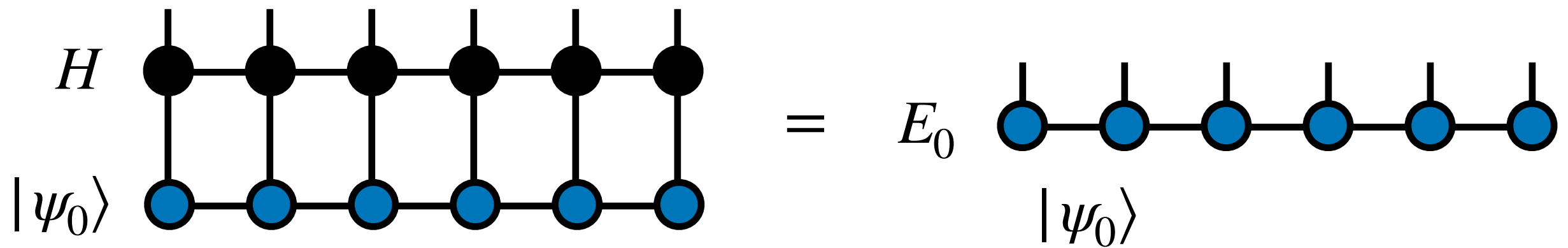


Tensor Networks

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

$$H = \begin{array}{cccccc} | & & | & & | & & | & & | & & | \\ \bullet & - & \bullet & - & \bullet & - & \bullet & - & \bullet & - & \bullet \\ | & & | & & | & & | & & | & & | \end{array}$$

A matrix product operator (MPO) network

MPO Tensor Network

Easiest to understand as
operator-valued matrices

$$H = \sum_{j=1}^{N-1} Z_j Z_{j+1} + \sum_{j=1}^N X_j = \leftarrow \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \rightarrow$$

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

This means

$$2 \text{---} \bullet \text{---} 1 = Z$$

$$3 \text{---} \bullet \text{---} 1 = X$$

etc.

MPO Tensor Network

Verify formula for Ising model + transverse field

$$H = \leftarrow \bullet \text{---} \bullet \rightarrow$$

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

MPO Tensor Network

Verify formula for Ising model + transverse field

$$H = \leftarrow \bullet \text{---} \bullet \rightarrow$$

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

MPO Tensor Network

Verify formula for Ising model + transverse field

$$H = \leftarrow \bullet \text{---} \bullet \rightarrow$$

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

MPO Tensor Network

Verify formula for Ising model + transverse field

$$H = \leftarrow \bullet \text{---} \bullet \rightarrow$$

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

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

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
```

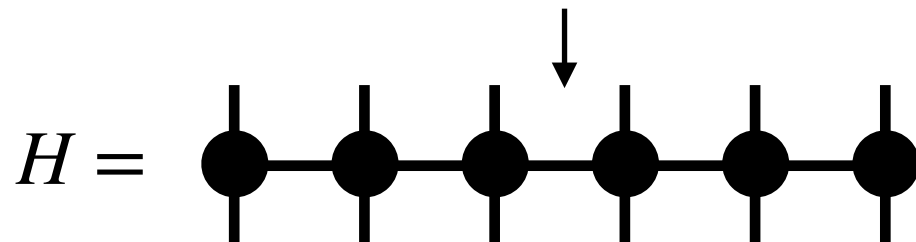
```
end
```

```
for j=1:N
```

```
    terms -= h,"X",j
```

```
end
```

```
H = MPO(terms,sites)
```



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

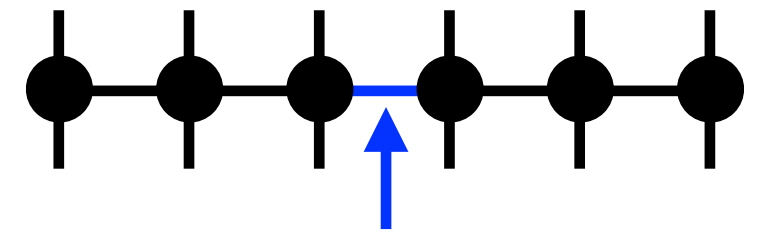
$$H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1}$$

bond dimension **5**

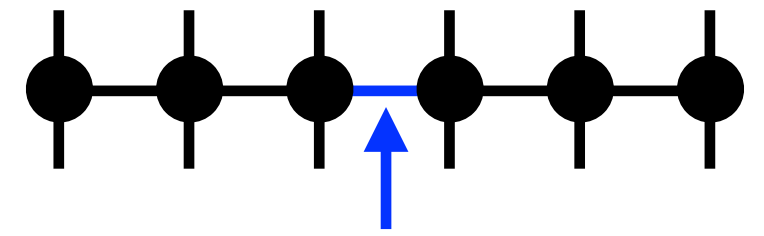
- Hubbard model

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

bond dimension **6**

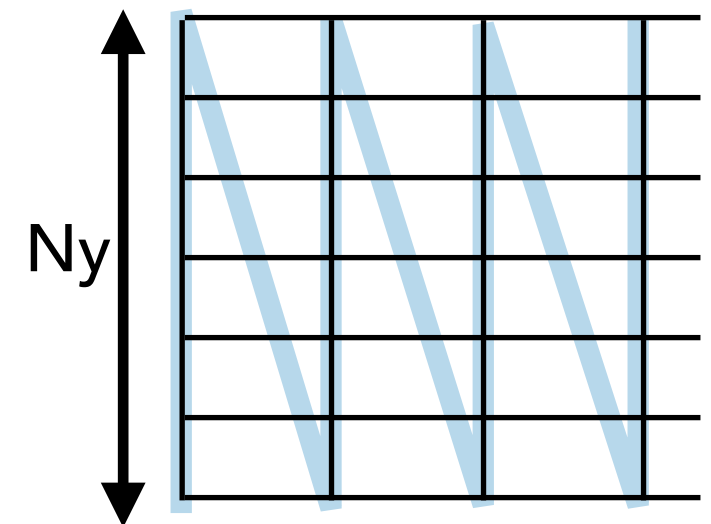
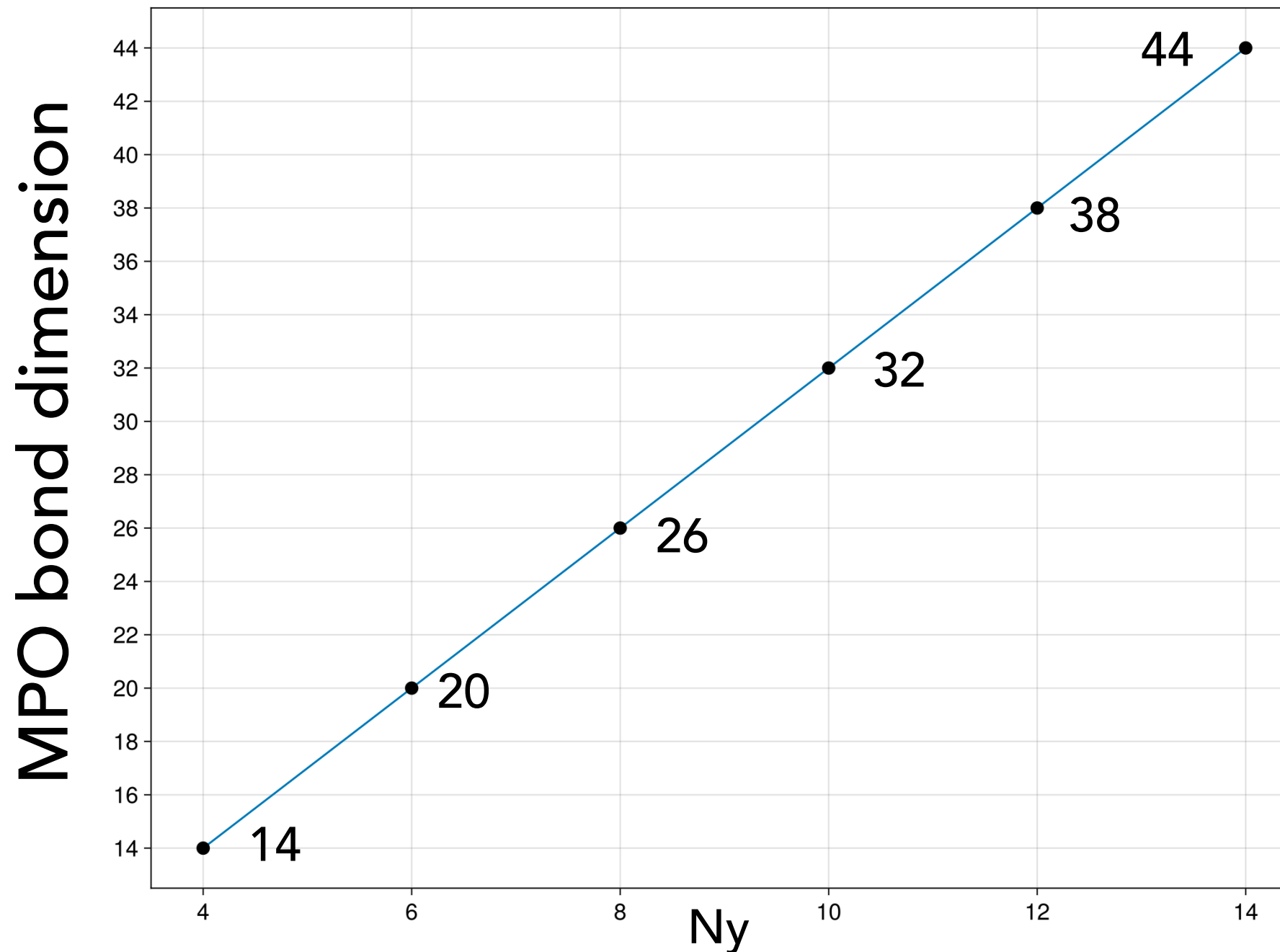


MPO Tensor Network



Common MPO bond dimensions – 2D

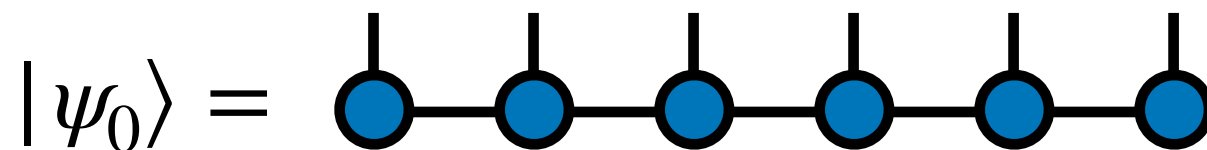
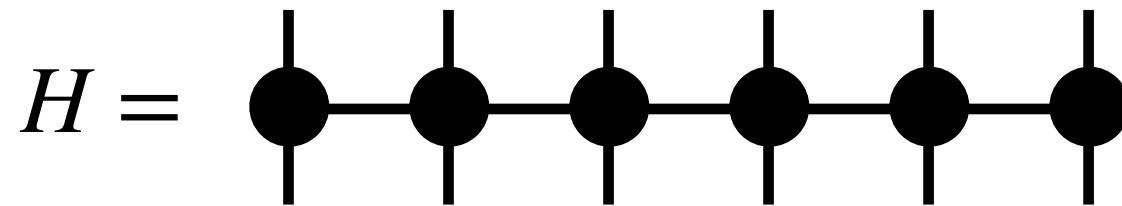
- Heisenberg model $H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1}$



Tensor Network Algorithms

DMRG algorithm

DMRG finds ground state of H as MPS tensor network



Tensor Network Algorithms

DMRG algorithm

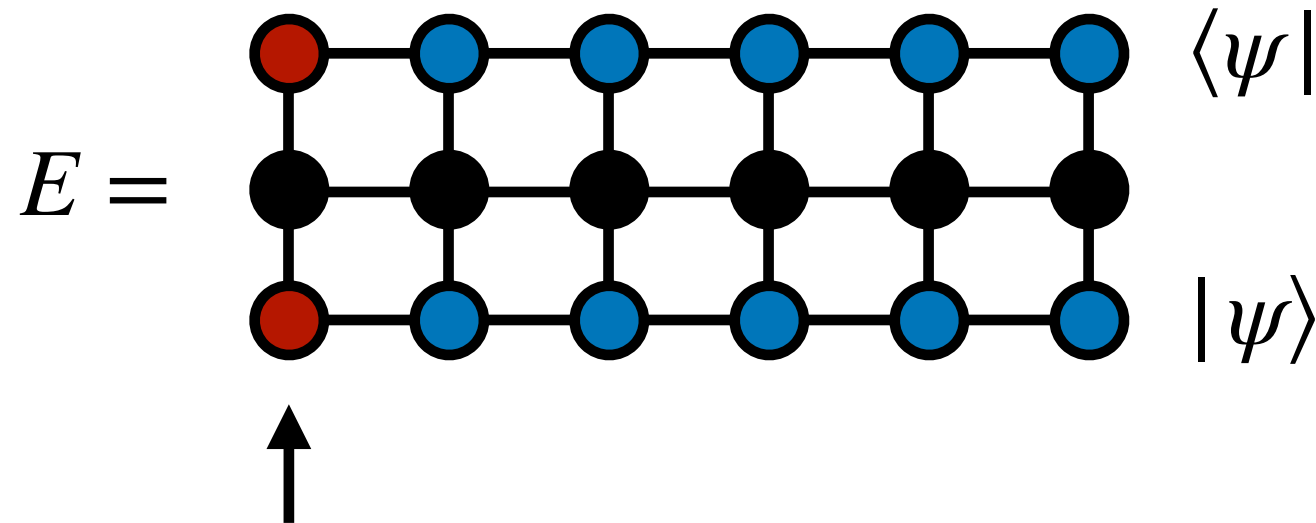
Energy is

$$E = \begin{array}{cccccc} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \langle \psi | \\ | & | & | & | & | & | & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\ | & | & | & | & | & | & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & | \psi \rangle \end{array}$$

Tensor Network Algorithms

DMRG algorithm

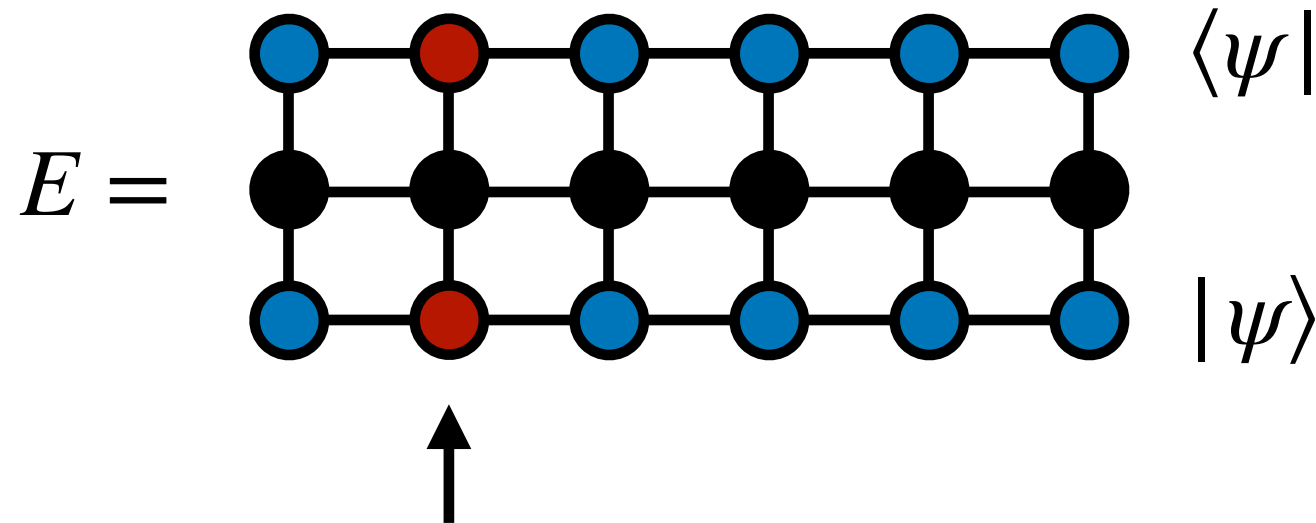
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

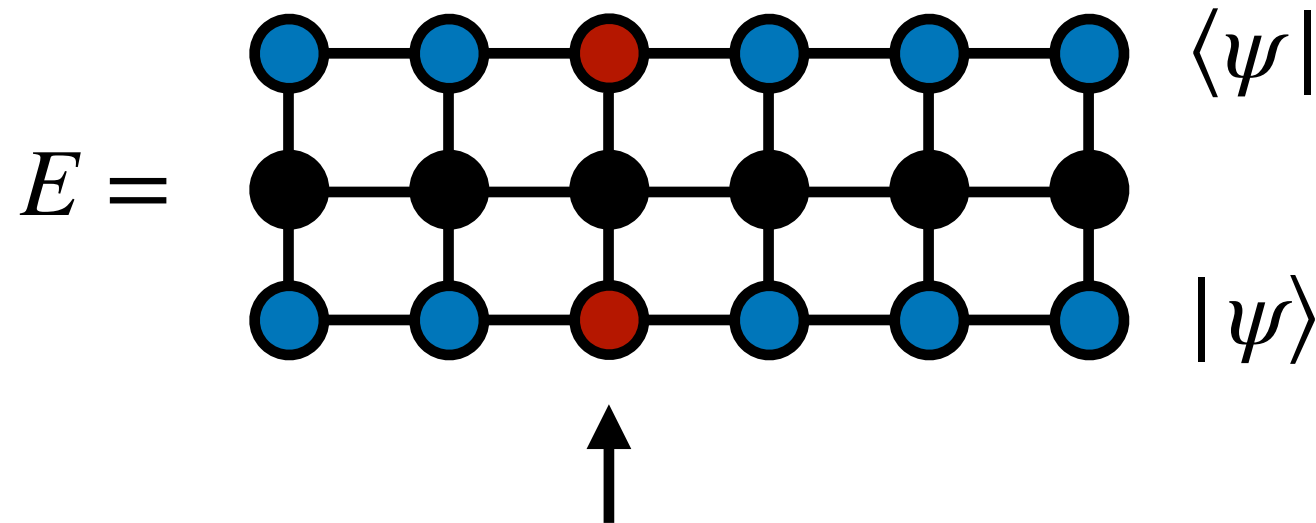
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

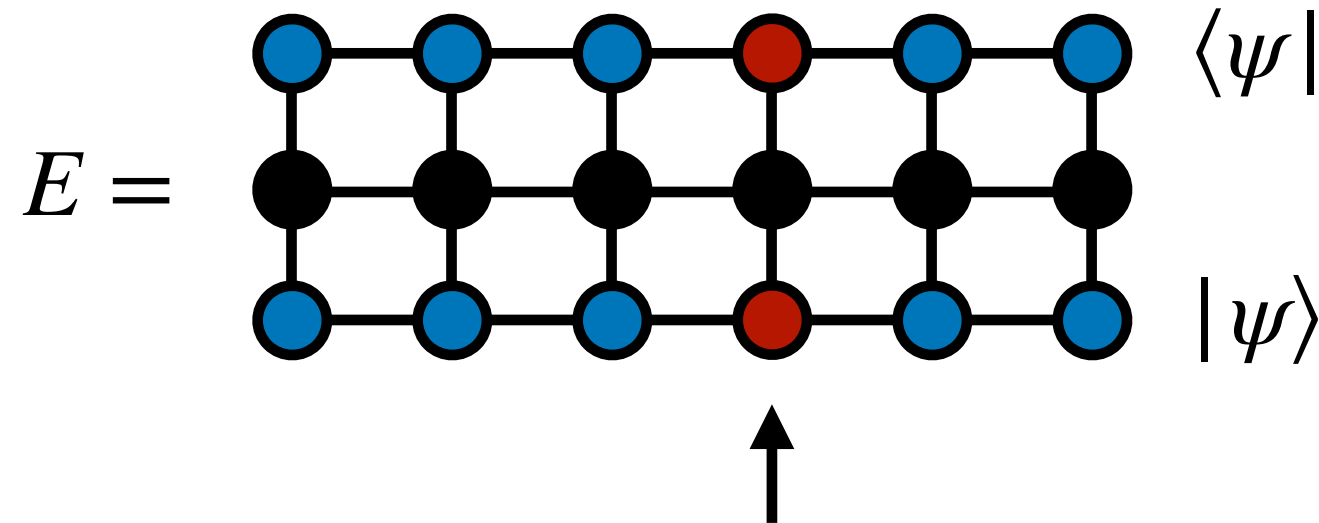
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

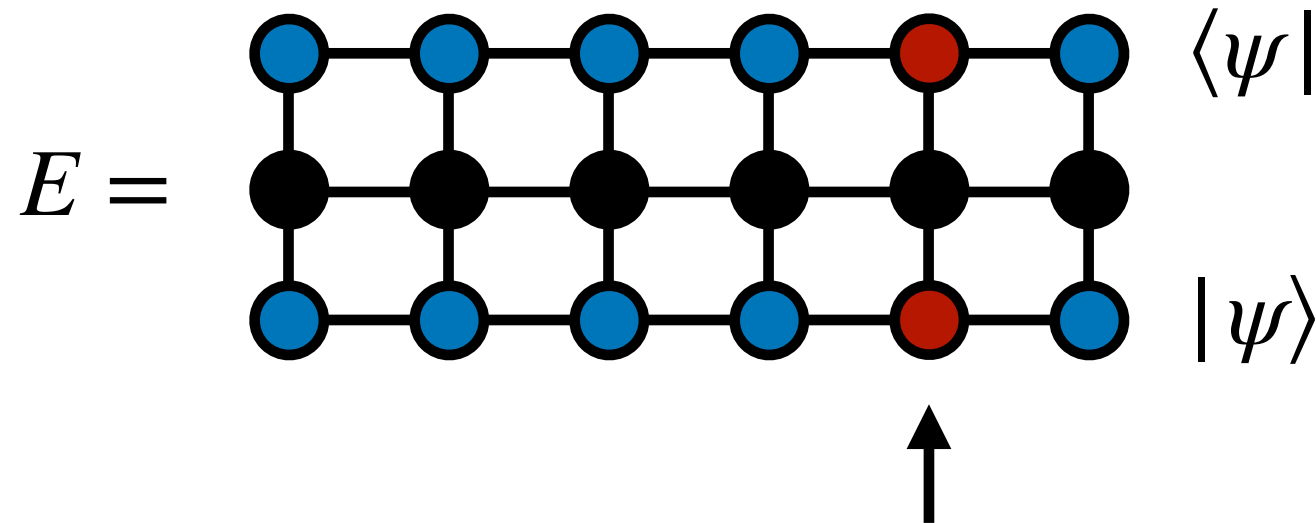
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

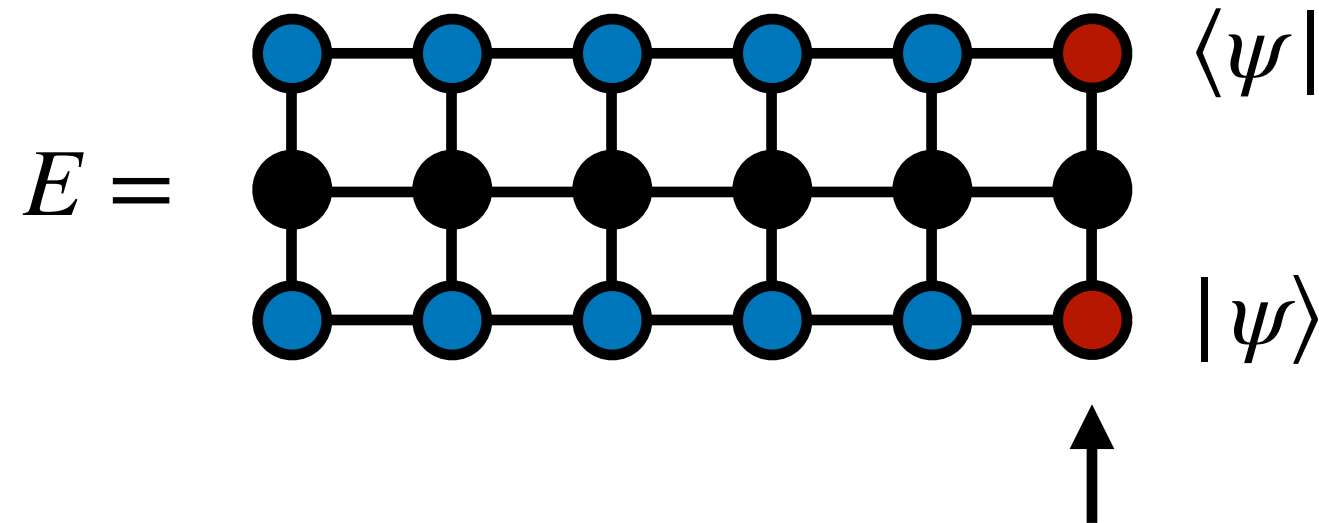
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

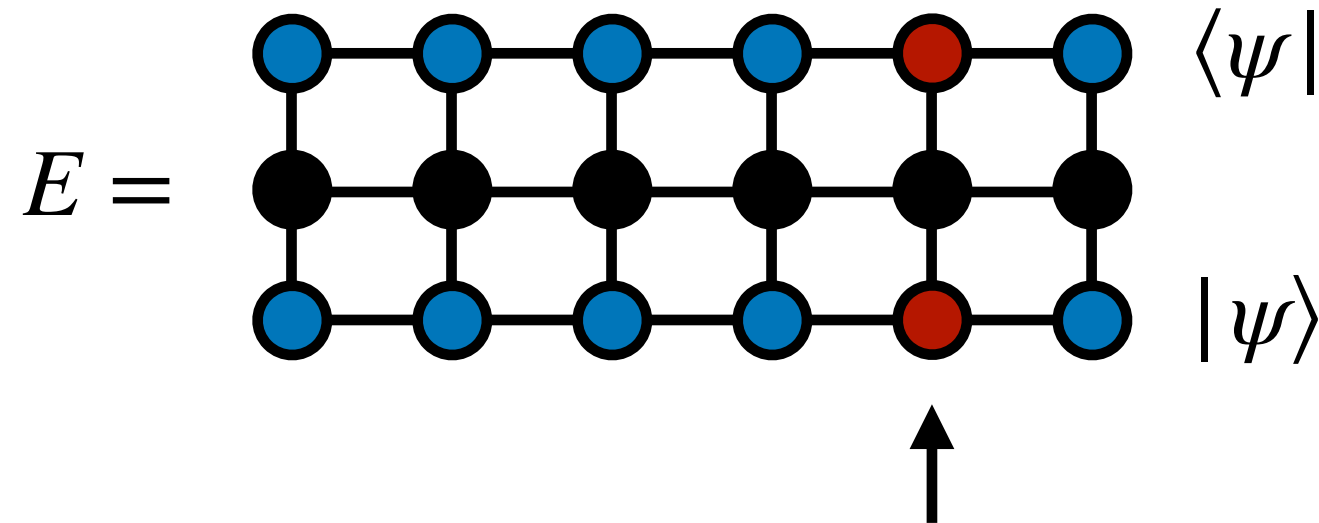
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

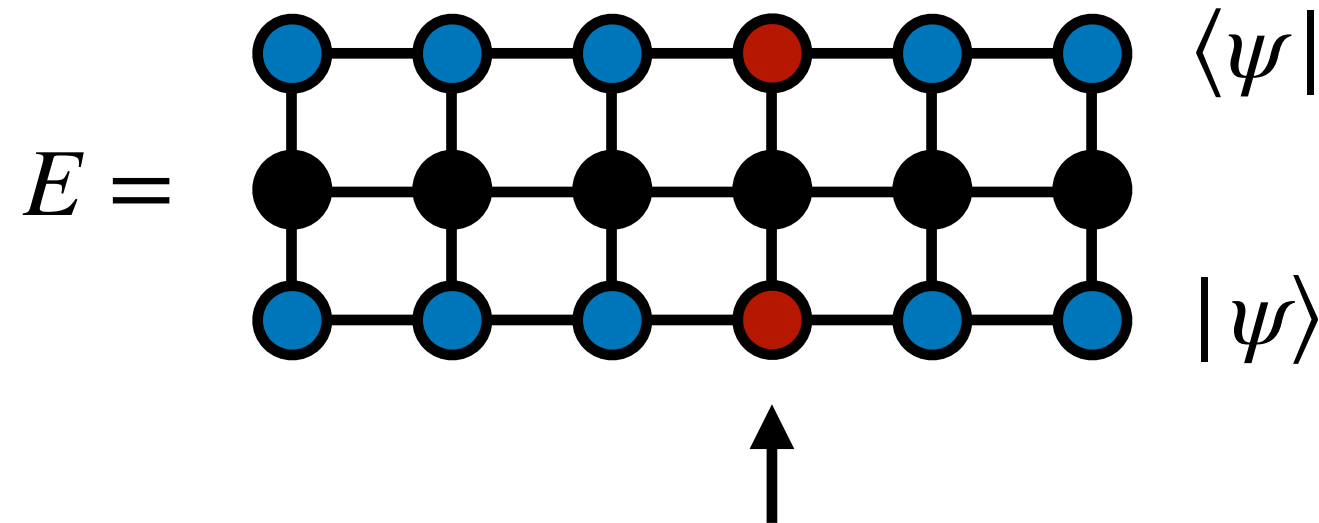
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

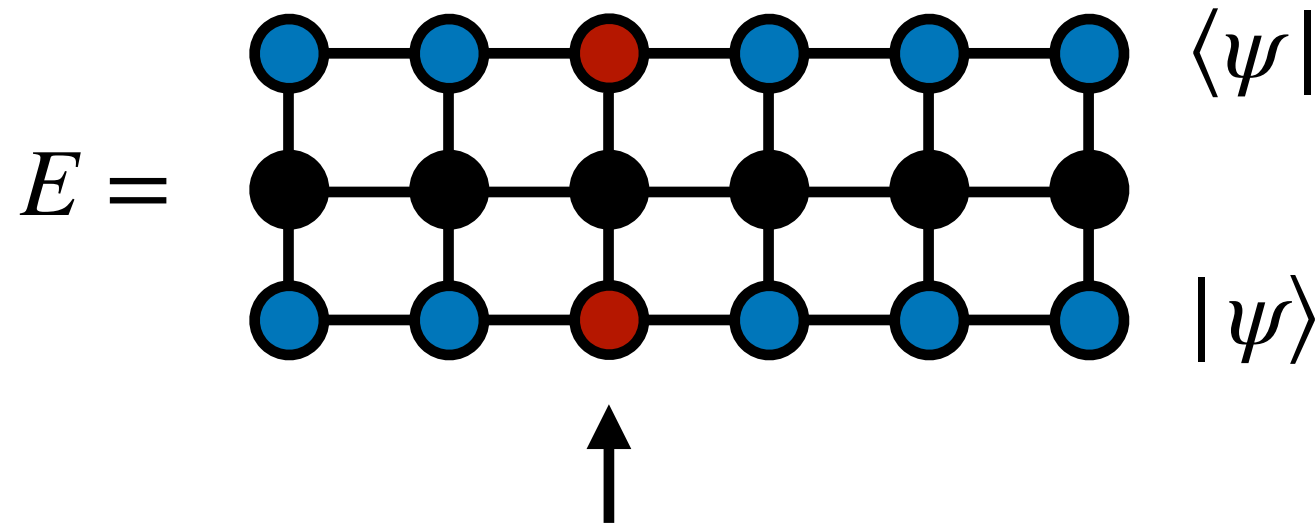
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

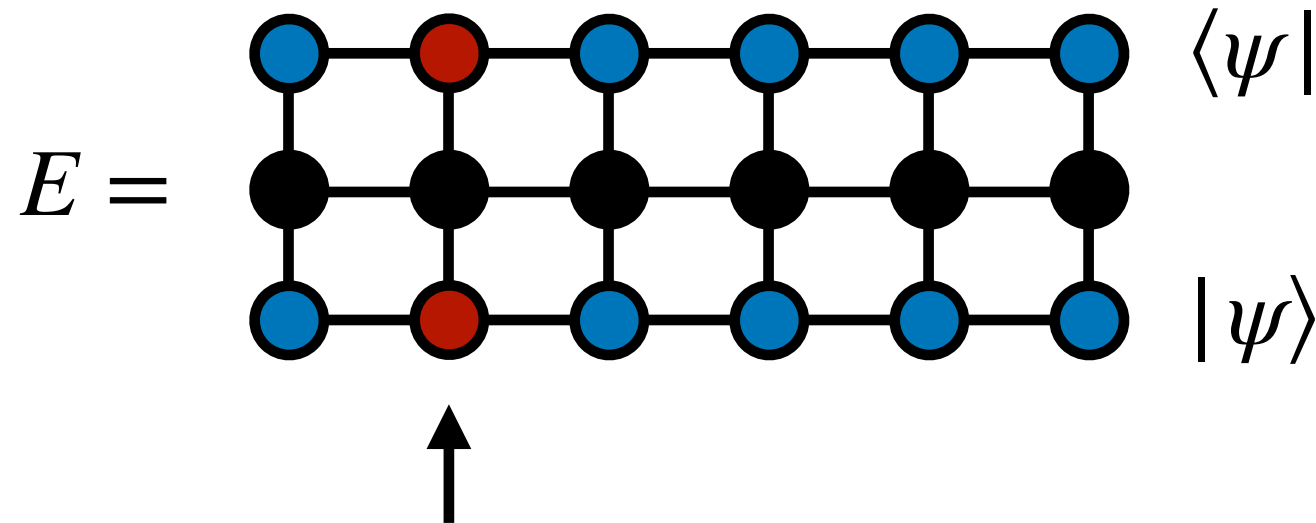
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

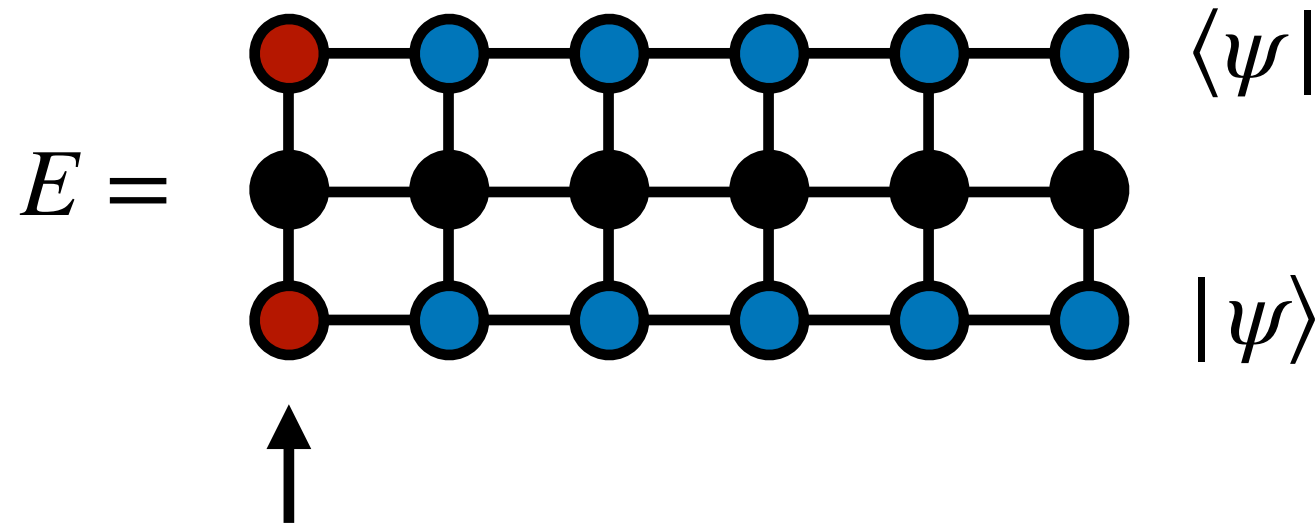
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

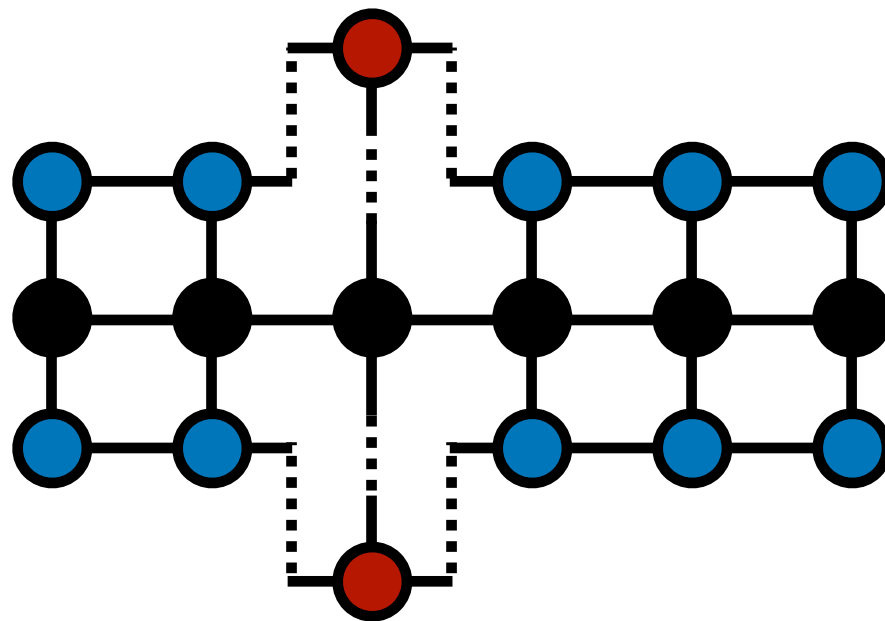
At each step, solve a reduced diagonalization problem

$$E = \begin{array}{cccccc} \text{blue} & \text{blue} & \text{red} & \text{blue} & \text{blue} & \text{blue} & \langle \psi | \\ \text{black} & \text{black} & \text{black} & \text{black} & \text{black} & \text{black} & \\ \text{blue} & \text{blue} & \text{red} & \text{blue} & \text{blue} & \text{blue} & | \psi \rangle \end{array}$$

Tensor Network Algorithms

DMRG algorithm

At each step, solve a reduced diagonalization problem

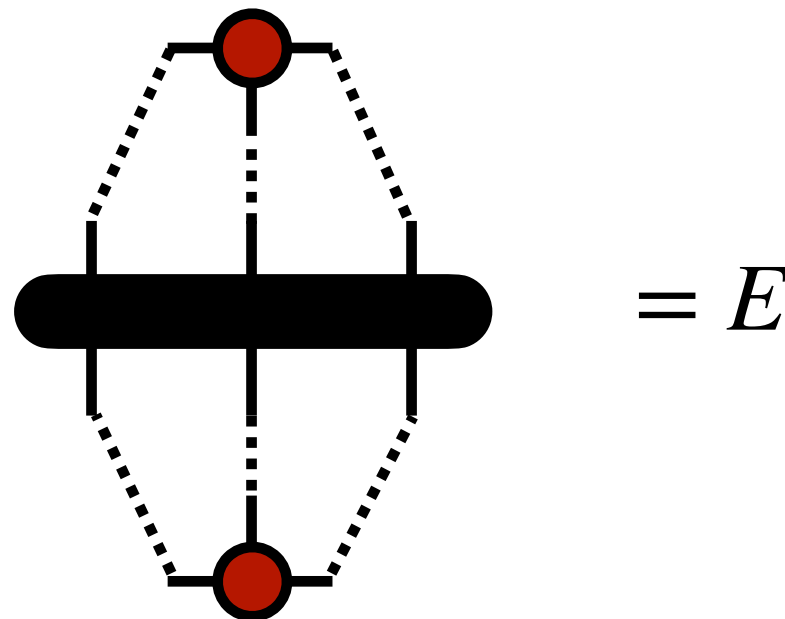


*technical note: for efficiency, frozen tensors are contracted in three groups, not exactly as shown above

Tensor Network Algorithms

DMRG algorithm

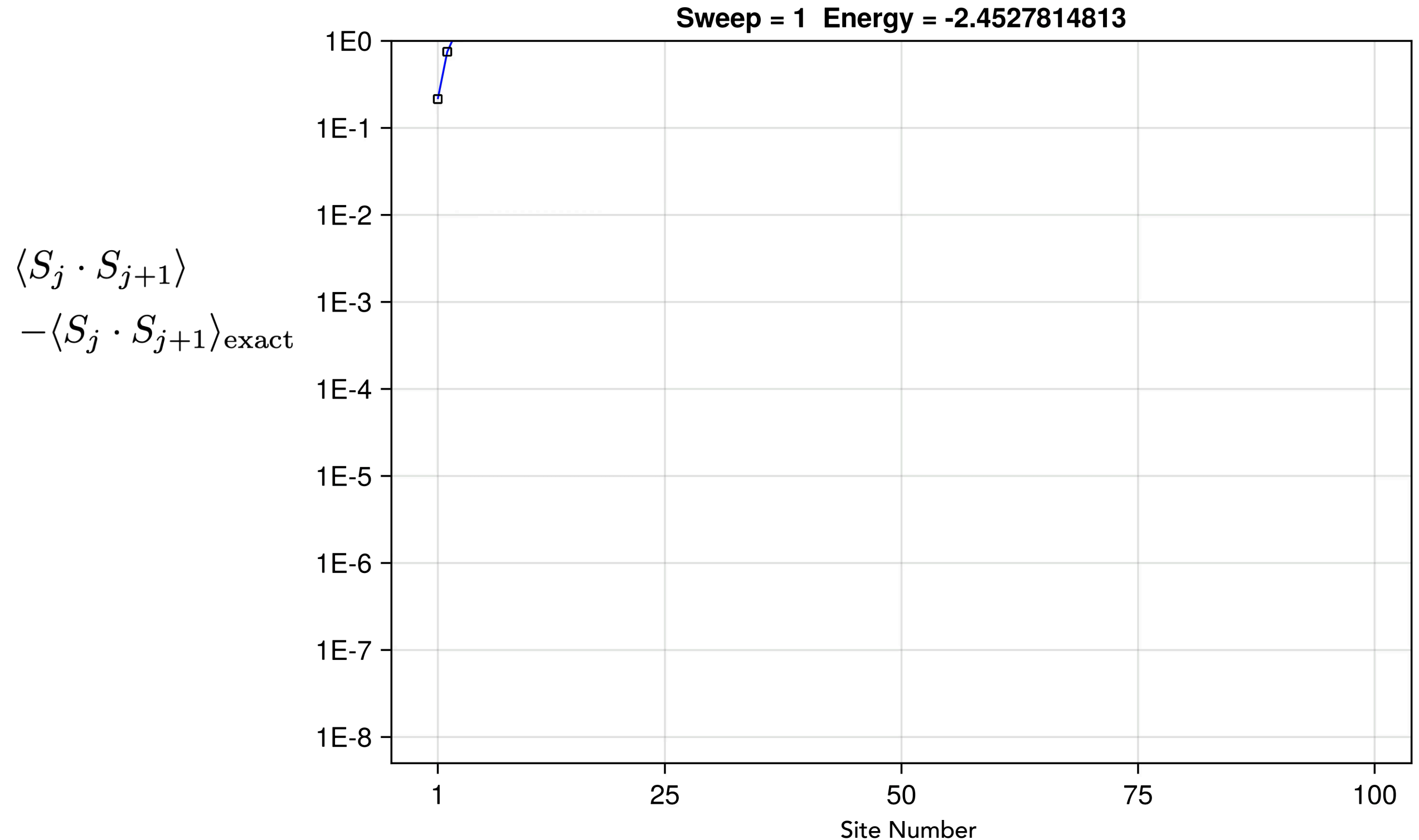
At each step, solve a reduced diagonalization problem



*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)



Tensor Network Algorithms

DMRG algorithm is extremely precise

Energy error vs number of 'sweeps' of DMRG

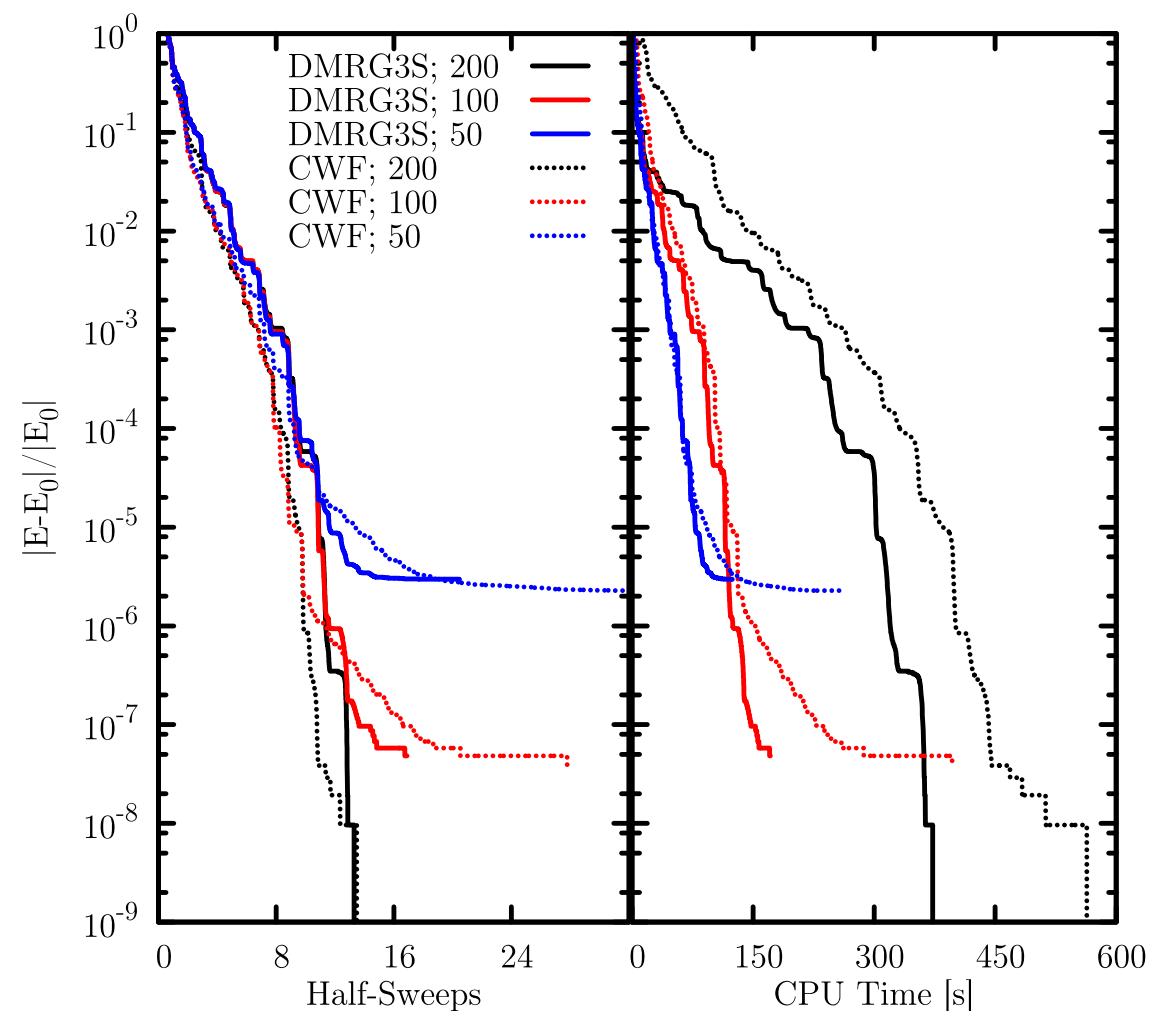
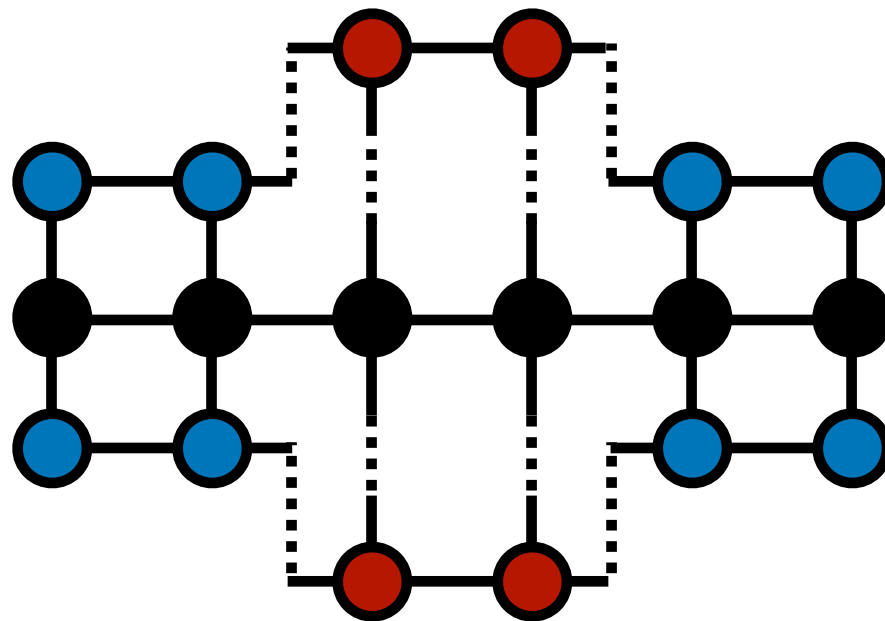


FIG. 4. (Color online) Bosonic system Eq. (41): normalized error in energy from CWF and DMRG3S as a function of sweeps (left) and CPU time used (right) for $m = 50, 100, 200$.

Tensor Network Algorithms

In practice, some important differences from above version

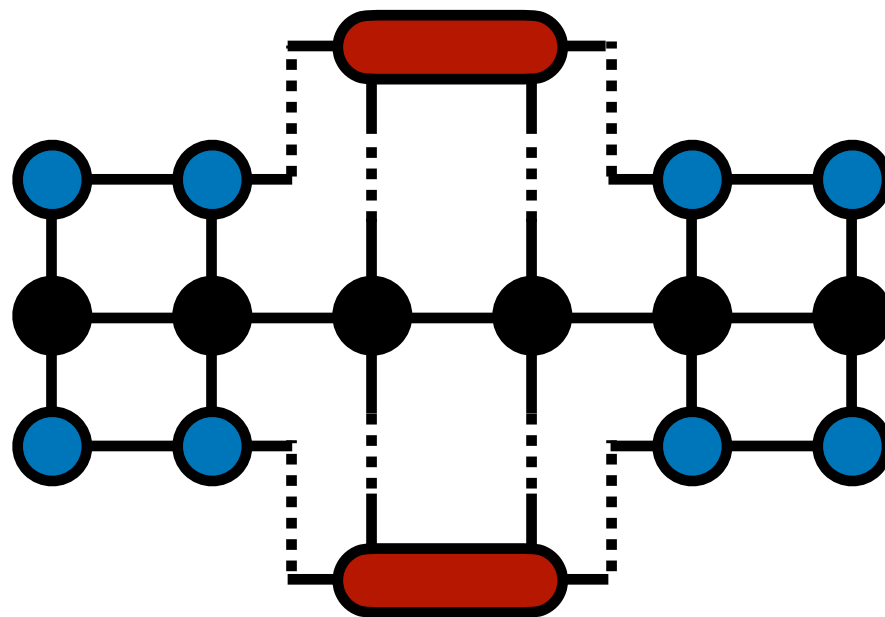
Typically two-site algorithm used:



Tensor Network Algorithms

In practice, some important differences from above version

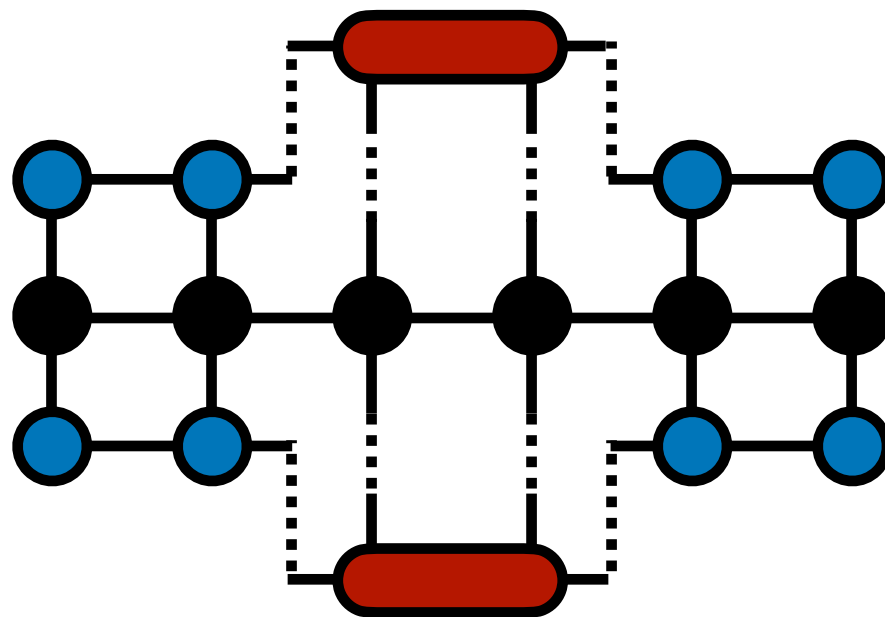
Typically two-site algorithm used:



Tensor Network Algorithms

In practice, some important differences from above version

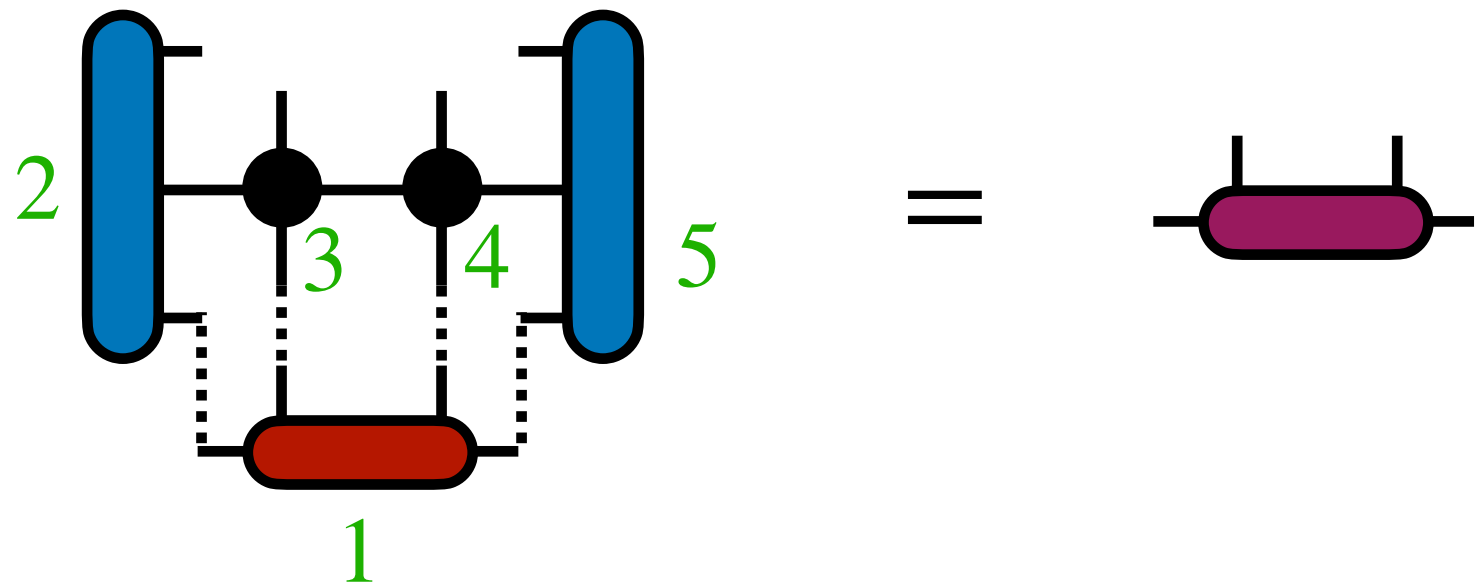
Eigen-solver step done more efficiently



Tensor Network Algorithms

In practice, some important differences from above version

Eigen-solver step done more efficiently

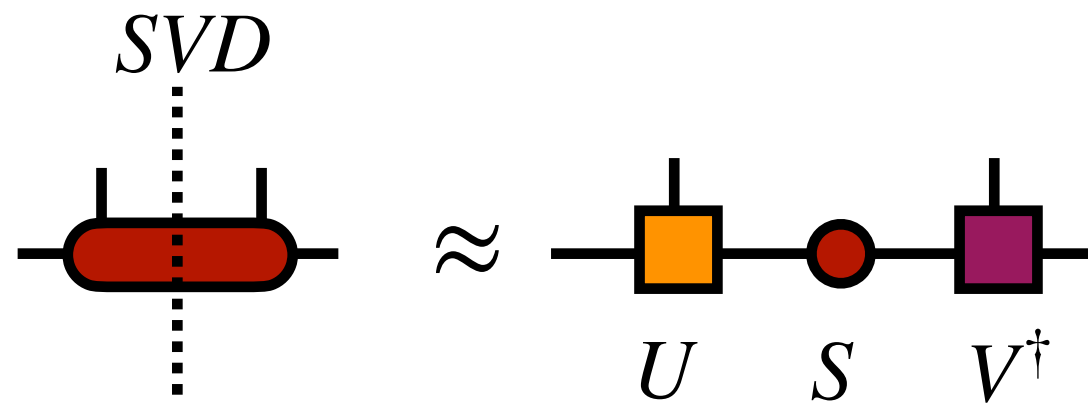


Lanczos / Krylov method to compute solution

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:

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

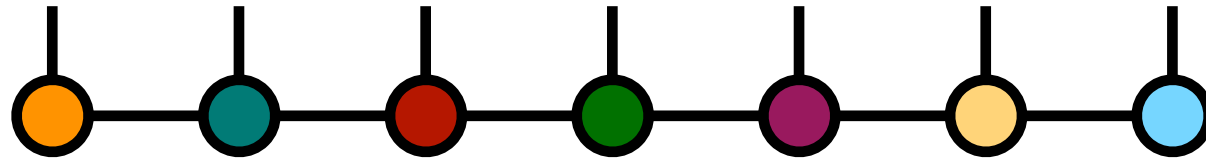
$$\varphi \rightarrow \varphi - \frac{\partial f}{\partial t}$$

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

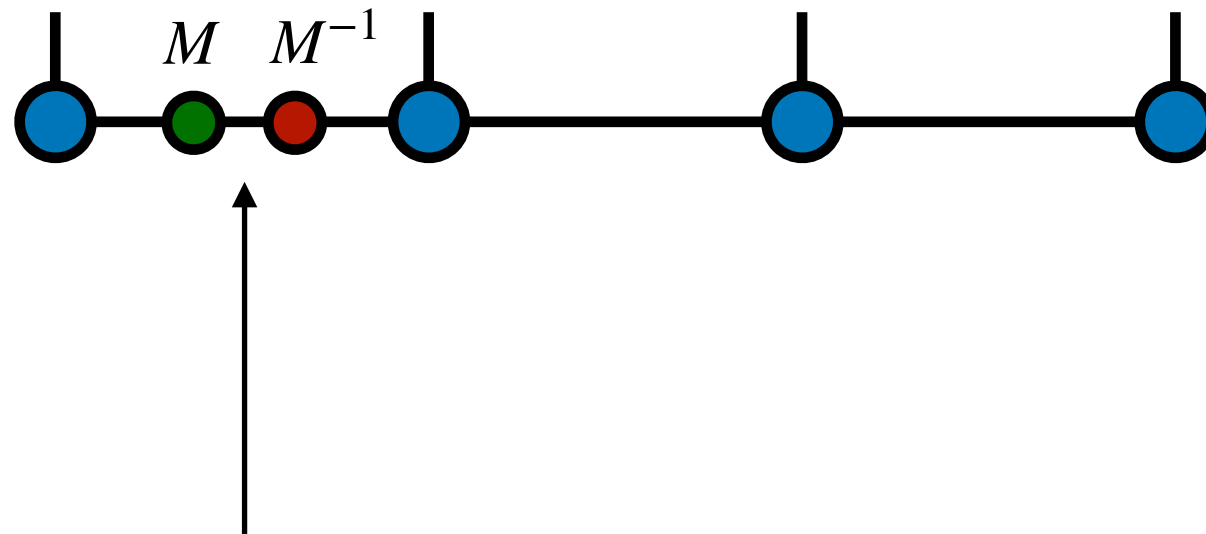
Gauging Tensor Networks

What are the gauge transformations of an MPS?



Gauging Tensor Networks

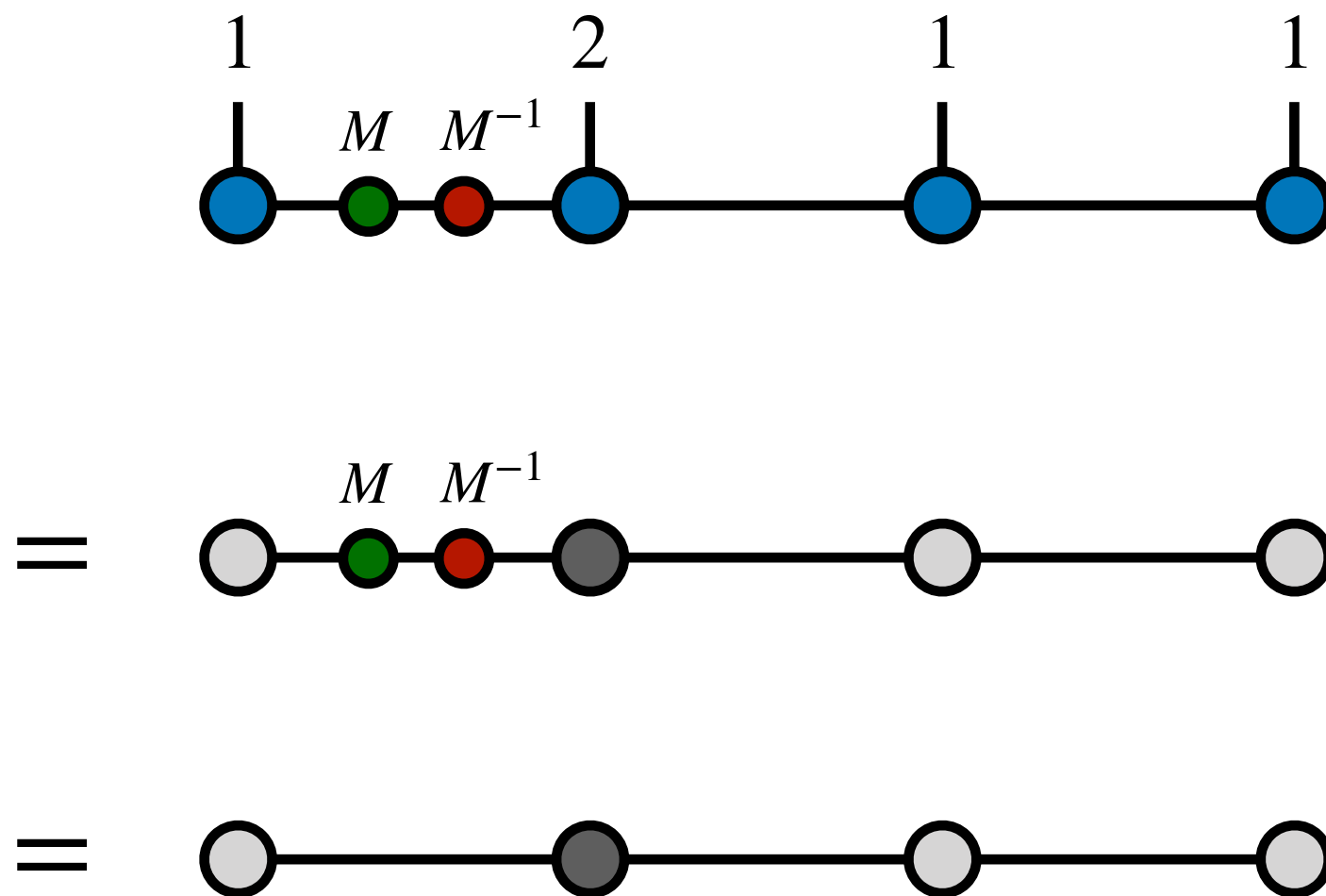
What are the gauge transformations of an MPS?



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?

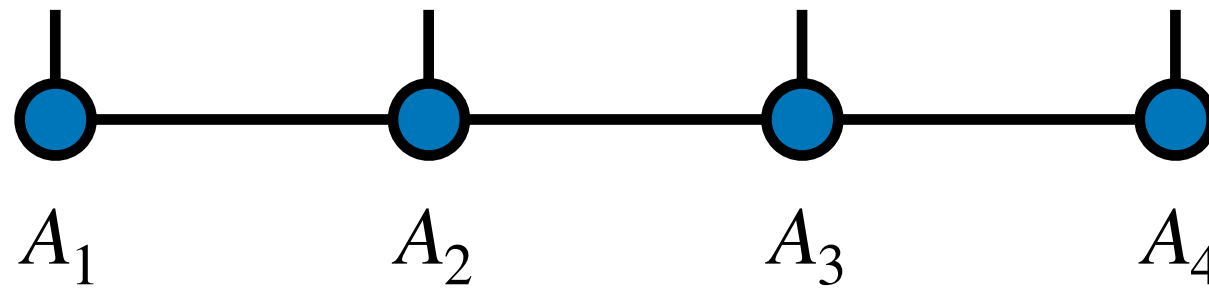
Interesting TN gauges related to:

- matrix factorizations
- quantum circuits

Orthogonal MPS Gauge

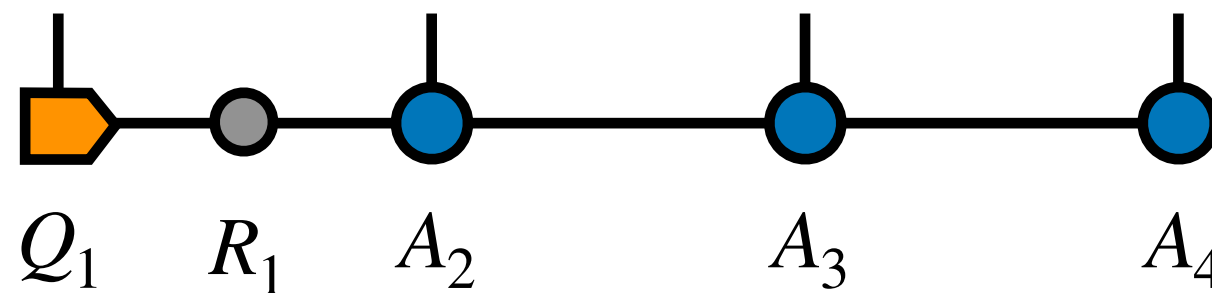
Let's find the "orthogonal" gauges of an MPS

Start from an arbitrary MPS

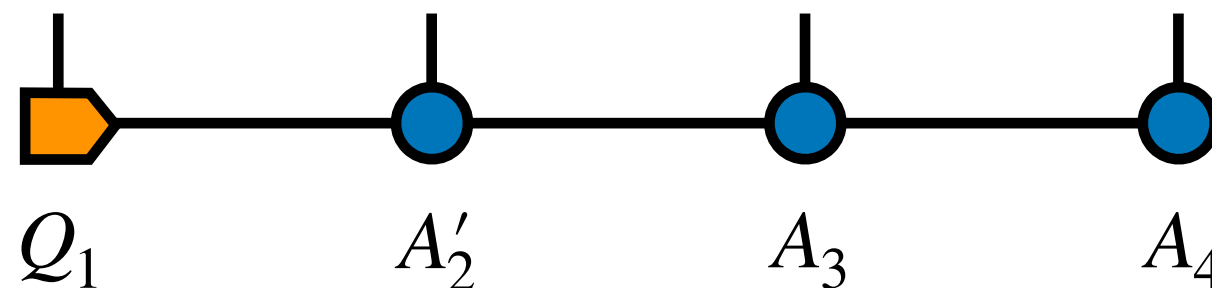


Orthogonal MPS Gauge

Now compute QR decomposition of $A_1 = Q_1 R_1$

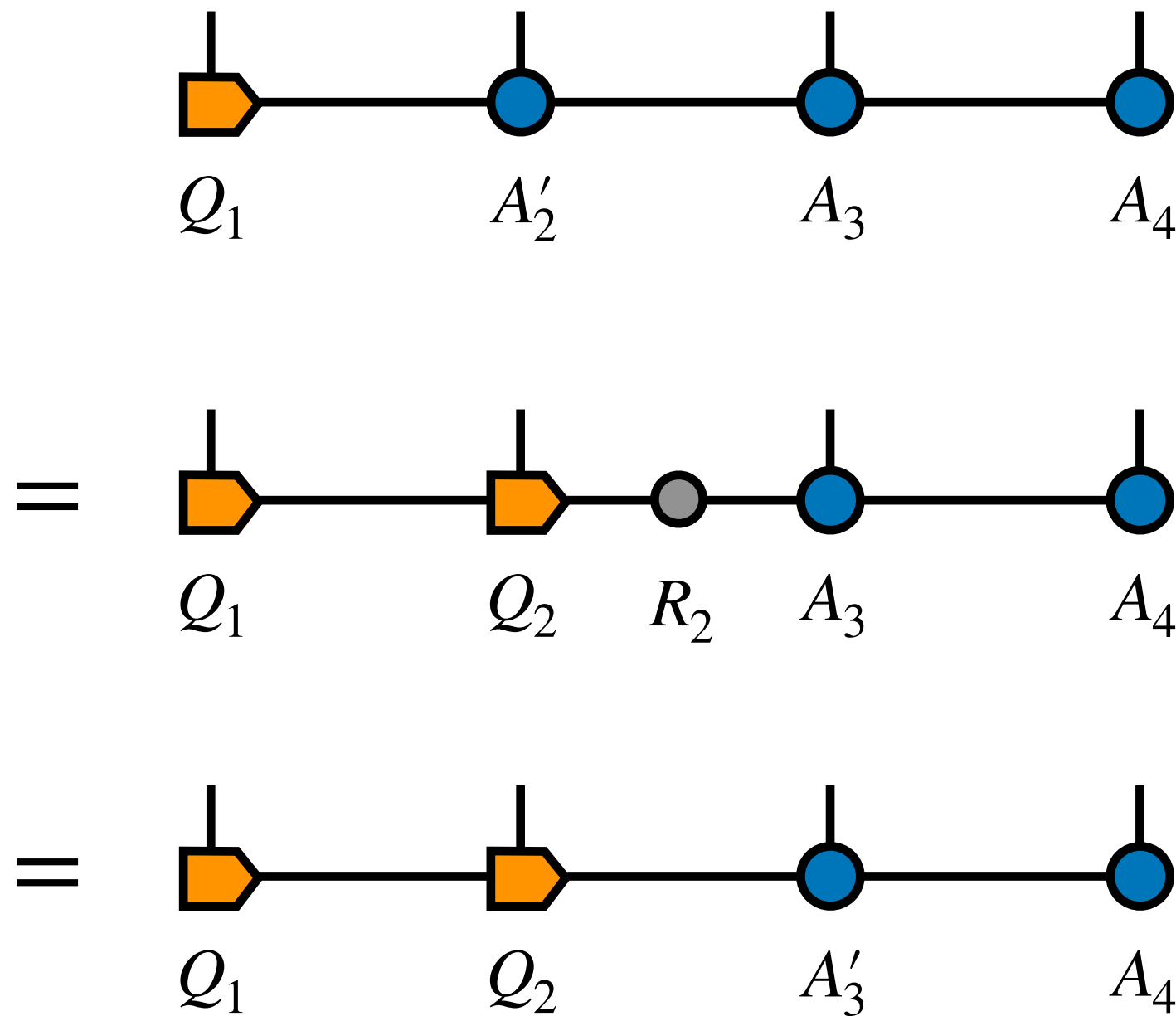


Multiply R_1 into A_2 to make A'_2



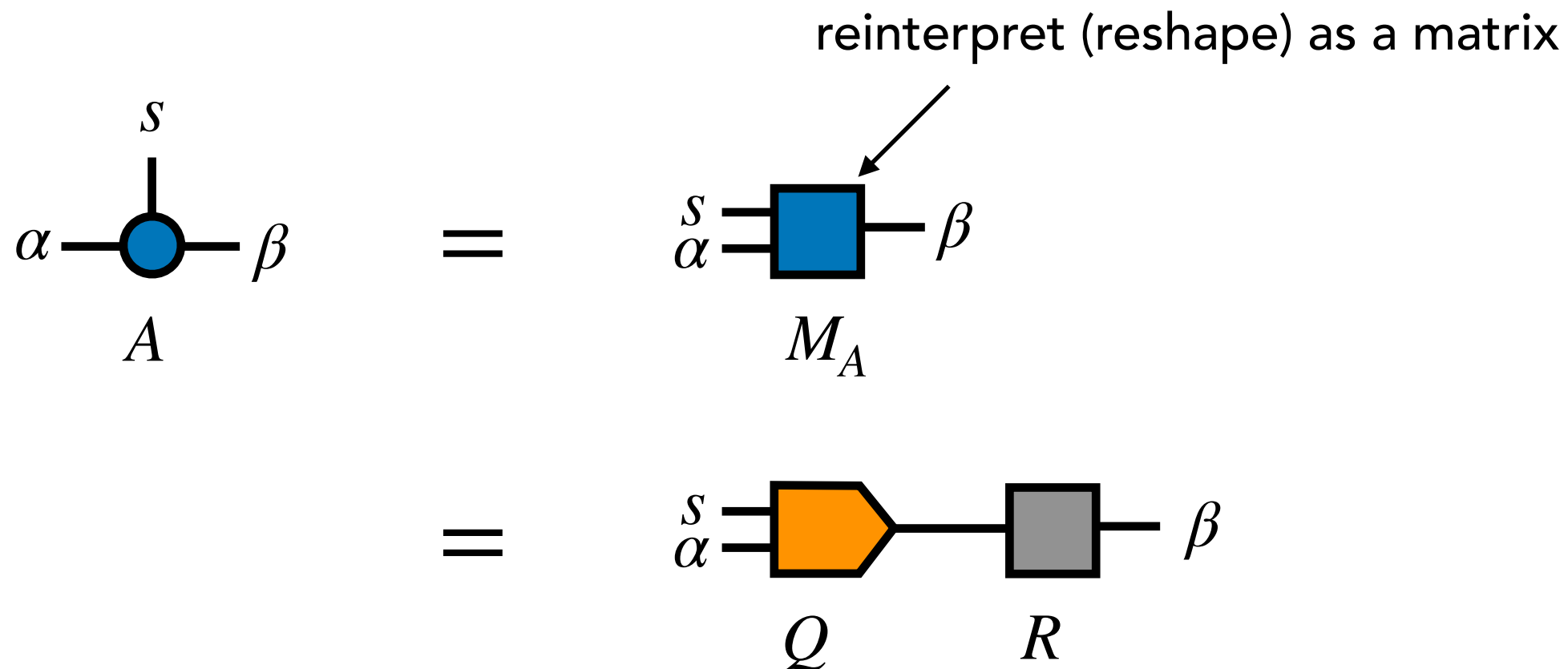
Orthogonal MPS Gauge

Next compute QR decomposition of $A'_2 = Q_2 R_2$

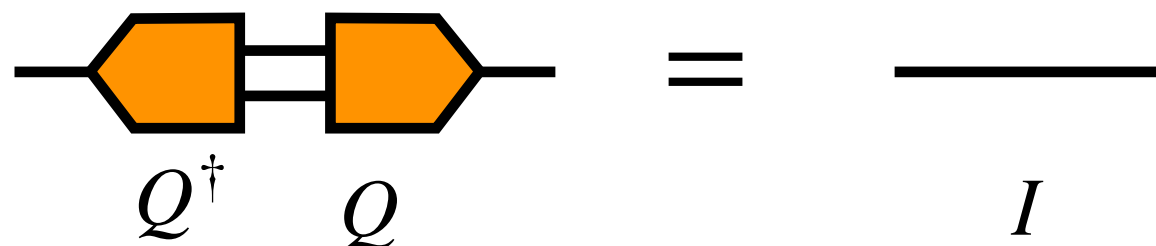


Orthogonal MPS Gauge

Recall QR of a tensor

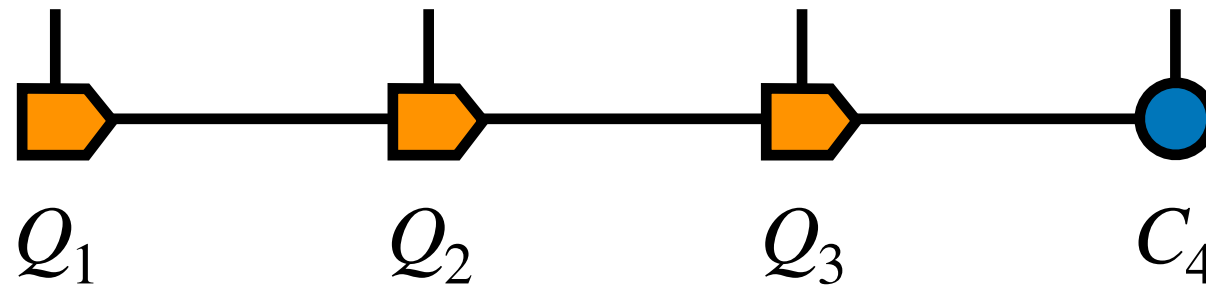


Q is an "isometric embedding" (isometry)



Orthogonal MPS Gauge

Continuing with more QR moves, one obtains



Call this "left orthogonal" gauge

Q tensors have "left orthogonality" property

A diagram illustrating the left orthogonality property of the Q tensors. On the left, two orange hexagonal tensors are stacked vertically. The top tensor is labeled Q_2^\dagger and the bottom tensor is labeled Q_2 . A curved line connects the right side of the top tensor to the right side of the bottom tensor. This is followed by an equals sign and a single curved line, representing the identity matrix.

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

Orthogonal MPS Gauge

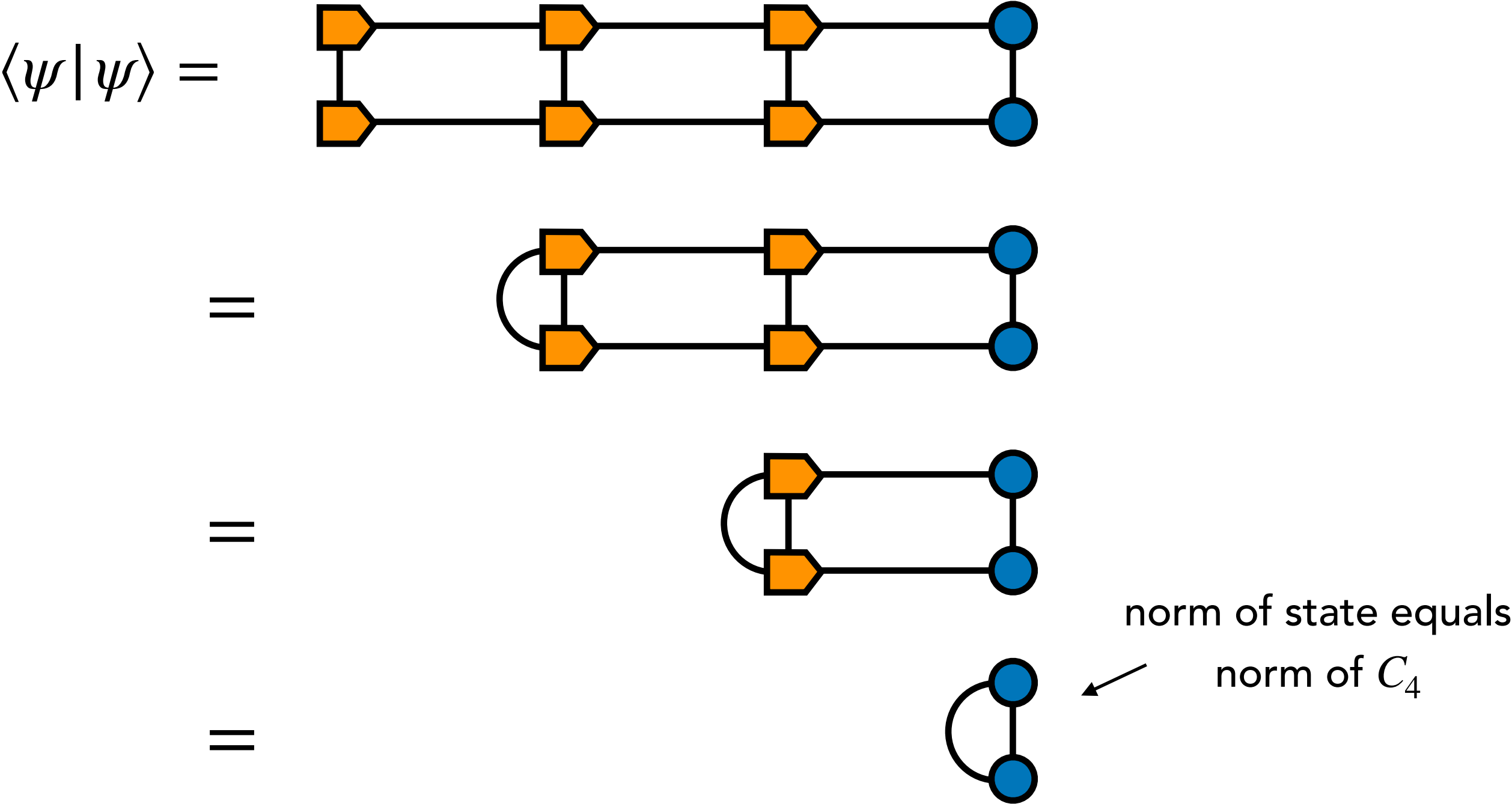
Challenge question:

Why QR and not SVD?

What advantages/disadvantages might one find?

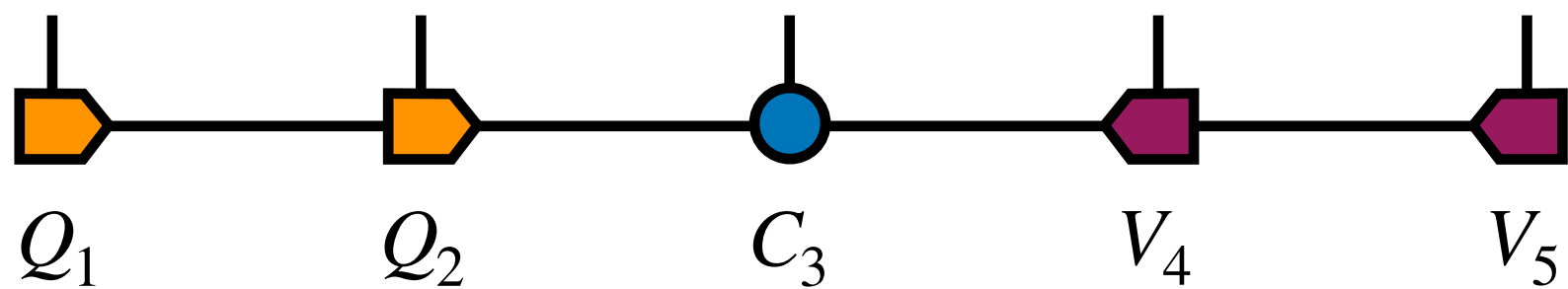
Orthogonal MPS Gauge

Can deduce certain properties



Orthogonal MPS Gauge

Can make "mixed" gauges too

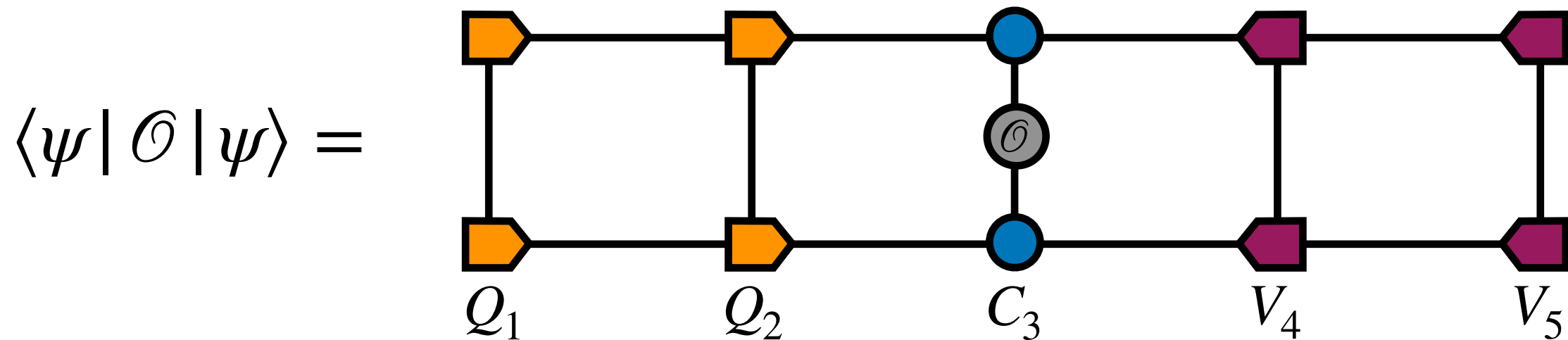


A diagrammatic equation showing the contraction of Q_j^\dagger and Q_j . On the left, two orange right-pointing arrows are stacked vertically, connected by a vertical line. A curved line connects the left side of the top arrow to the left side of the bottom arrow. This is followed by an equals sign and a large left parenthesis $($.

A diagrammatic equation showing the contraction of V_j^\dagger and V_j . On the left, two purple left-pointing arrows are stacked vertically, connected by a vertical line. A curved line connects the right side of the top arrow to the right side of the bottom arrow. This is followed by an equals sign and a large right parenthesis $)$.

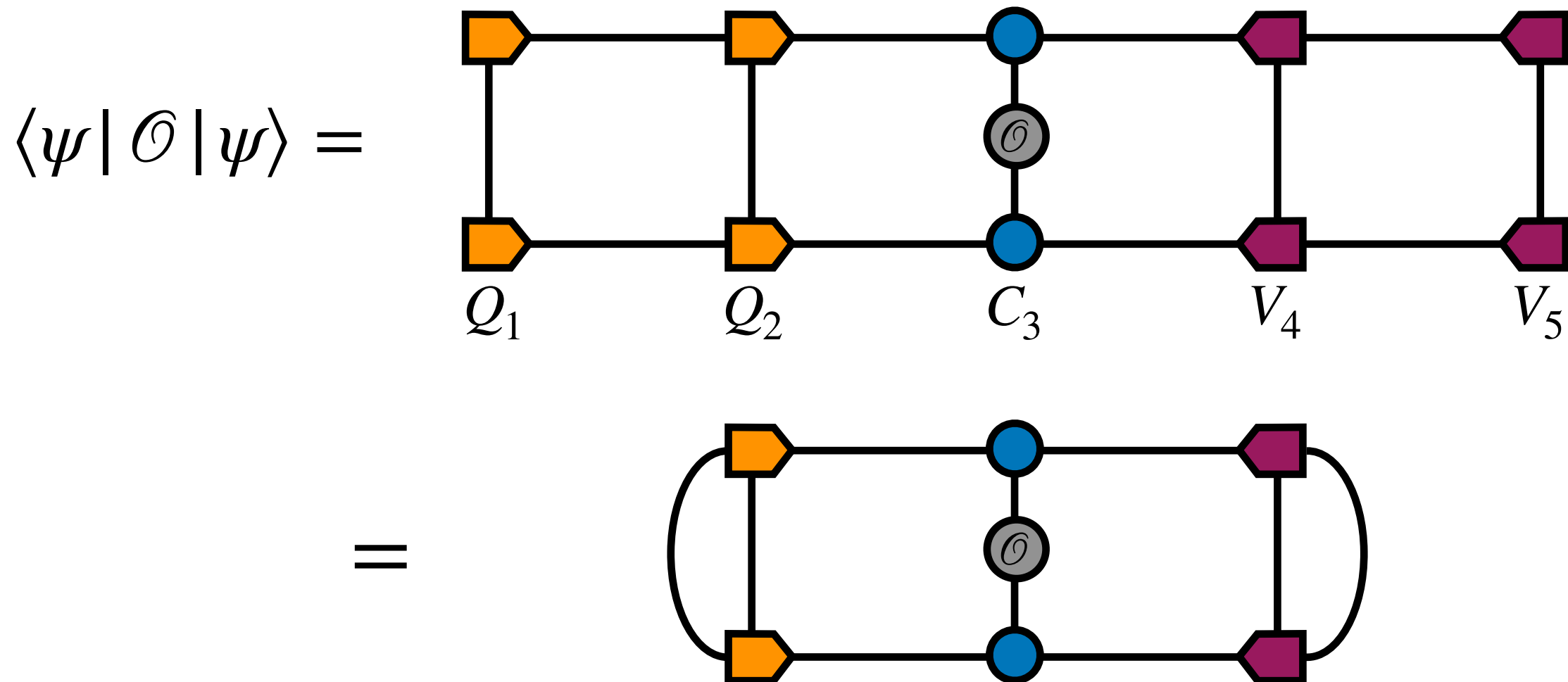
Orthogonal MPS Gauge

Mixed gauges useful for expected values



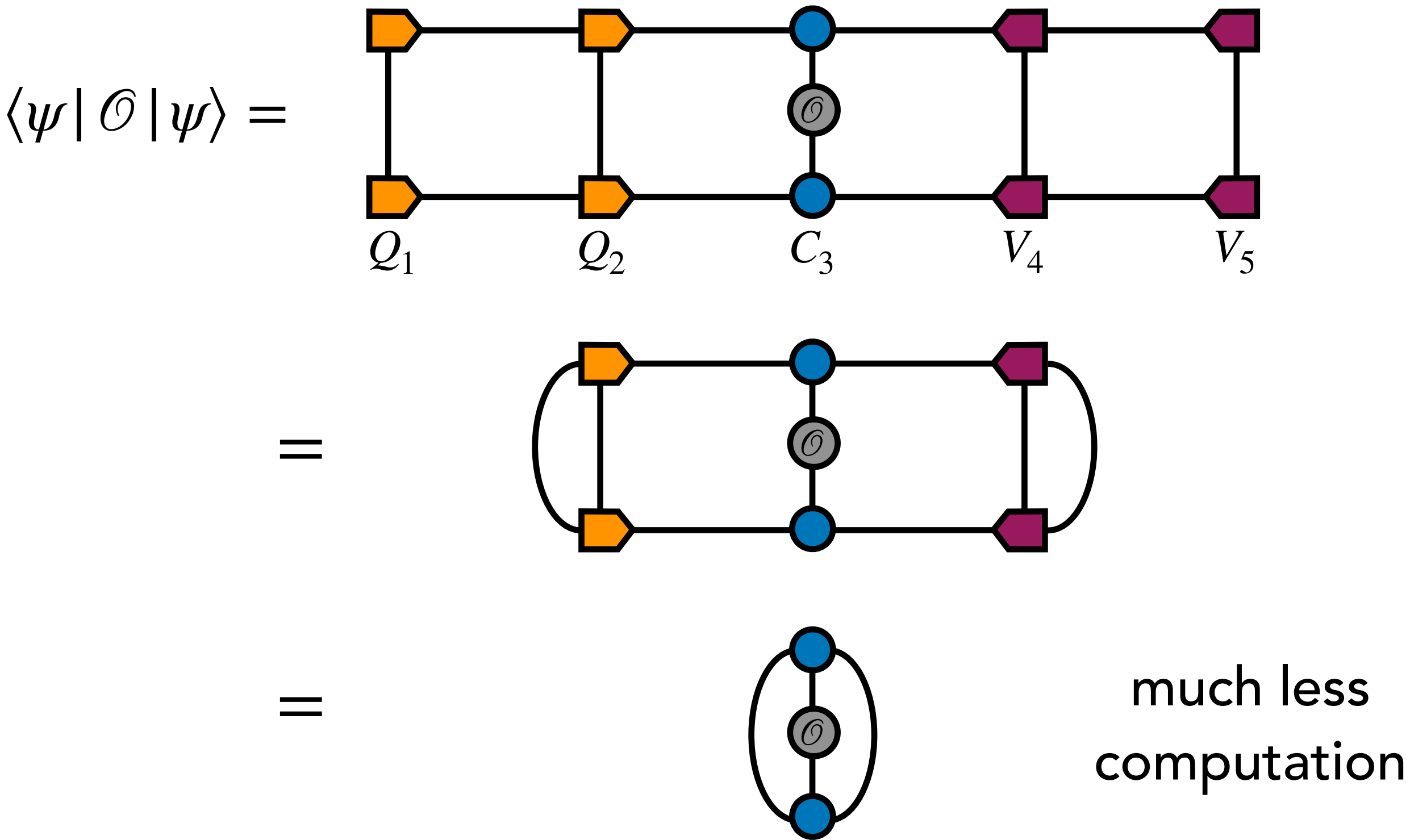
Orthogonal MPS Gauge

Mixed gauges useful for expected values



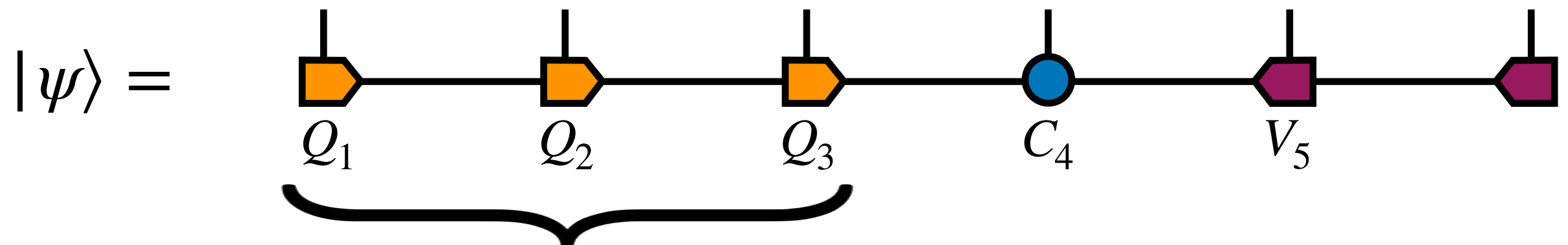
Orthogonal MPS Gauge

Mixed gauges useful for expected values



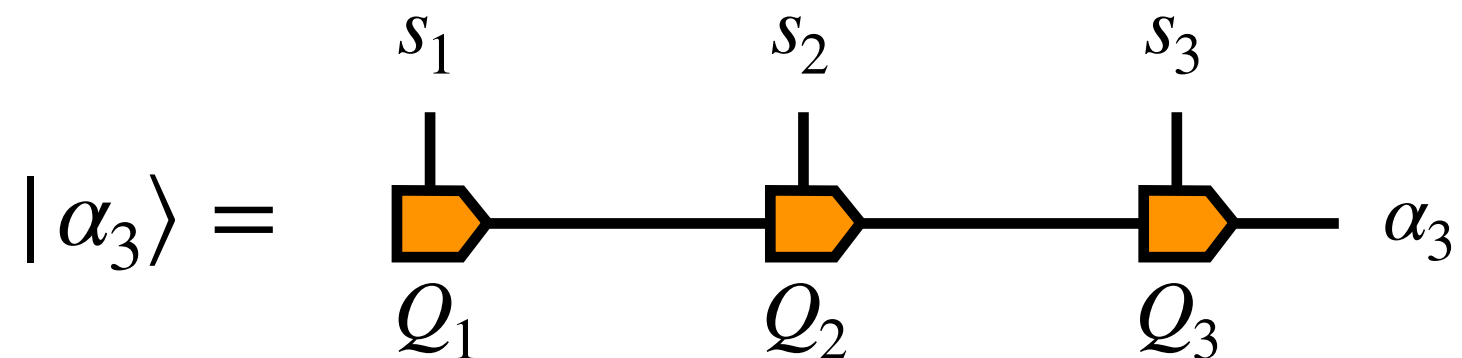
Orthogonal MPS Gauge

Mixed gauges also give interesting interpretation of MPS



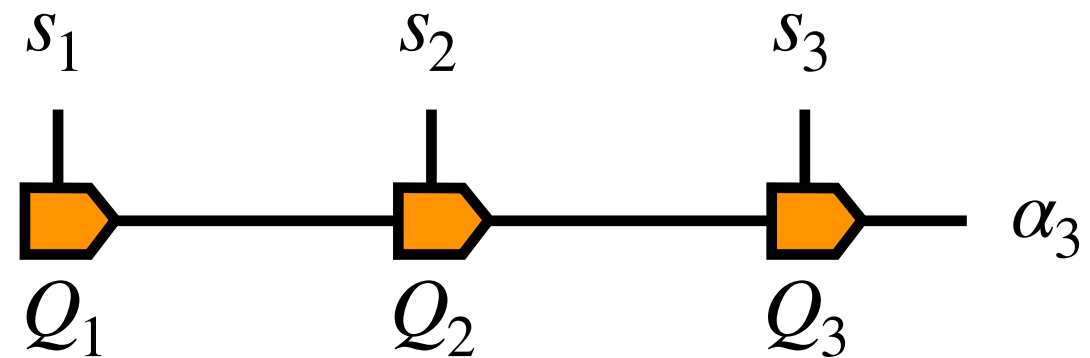
consider left part of 'gauged' MPS

It is an orthonormal basis – why?

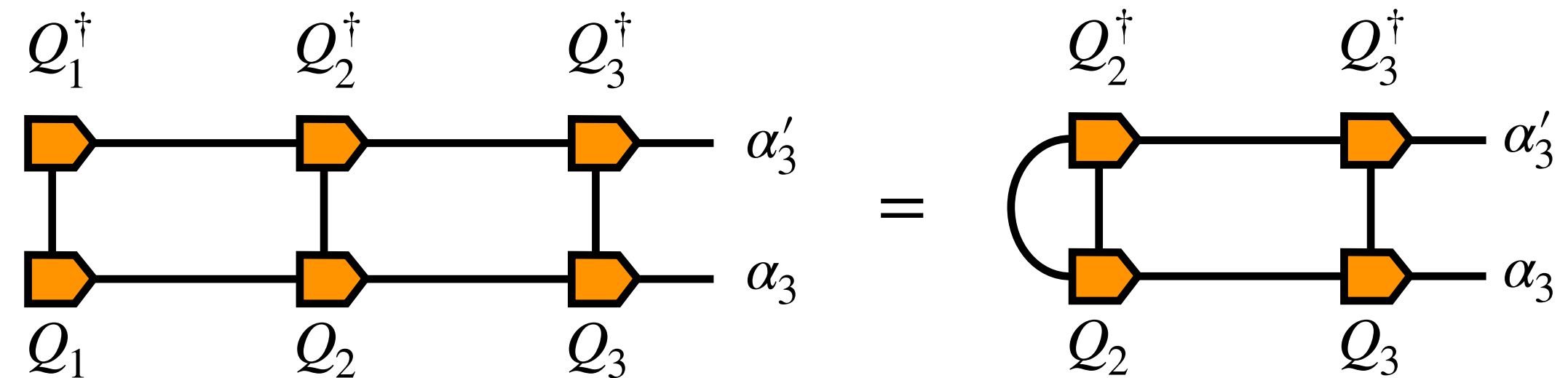


Orthogonal MPS Gauge

It is an orthonormal basis – why?

$$|\alpha_3\rangle =$$


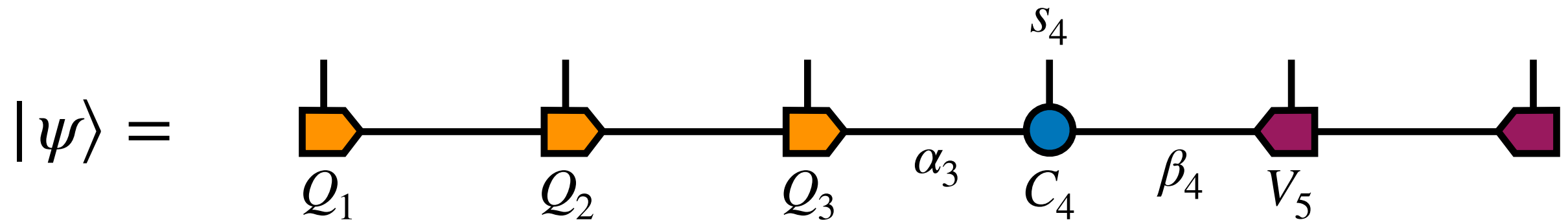
Compute overlap

$$\langle \alpha'_3 | \alpha_3 \rangle =$$


$$= \delta_{\alpha'_3 \alpha_3} \text{ orthonormal in the } \alpha'_3, \alpha_3 \text{ labels}$$

Orthogonal MPS Gauge

Thus entire MPS can be viewed as following

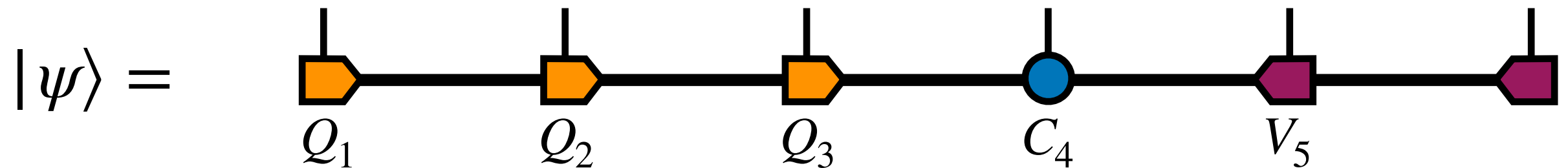


$$|\psi\rangle = \sum_{\alpha_3 \beta_4} C_{\alpha_3 \beta_4}^{s_4} |\alpha_3\rangle |s_4\rangle |\beta_4\rangle$$

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

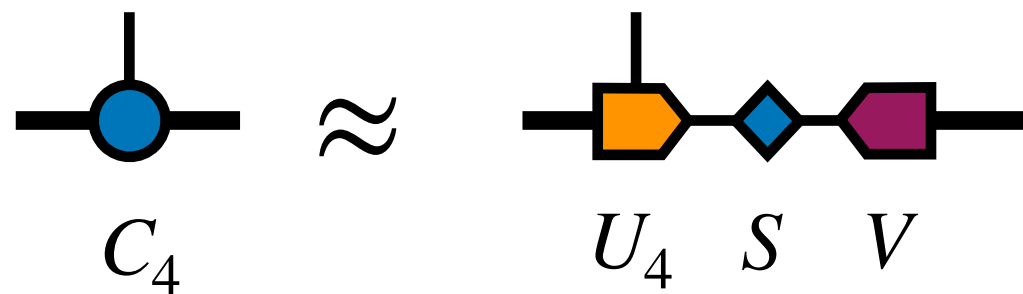
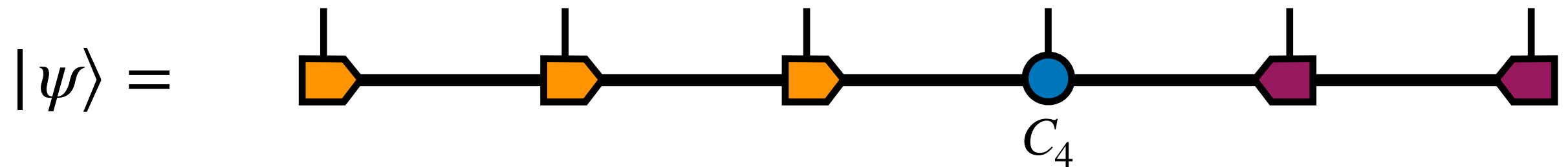
Orthogonal MPS Gauge

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



Orthogonal MPS Gauge

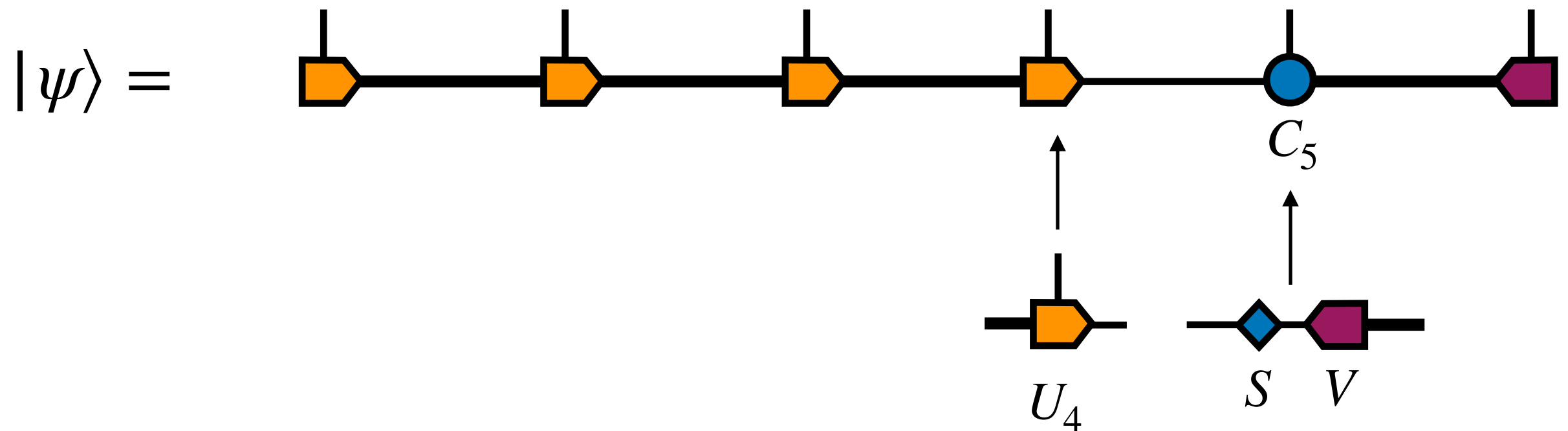
Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension



Perform SVD of C_4 tensor – **truncate** small singular values

Orthogonal MPS Gauge

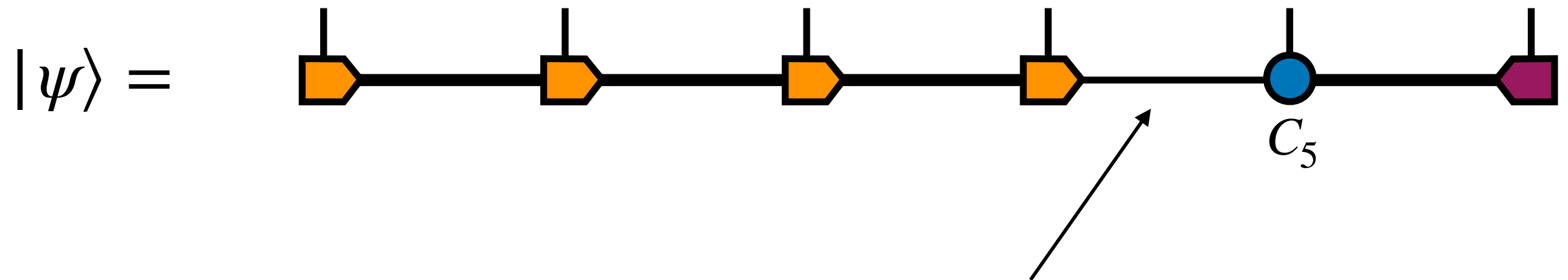
Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension



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

Orthogonal MPS Gauge

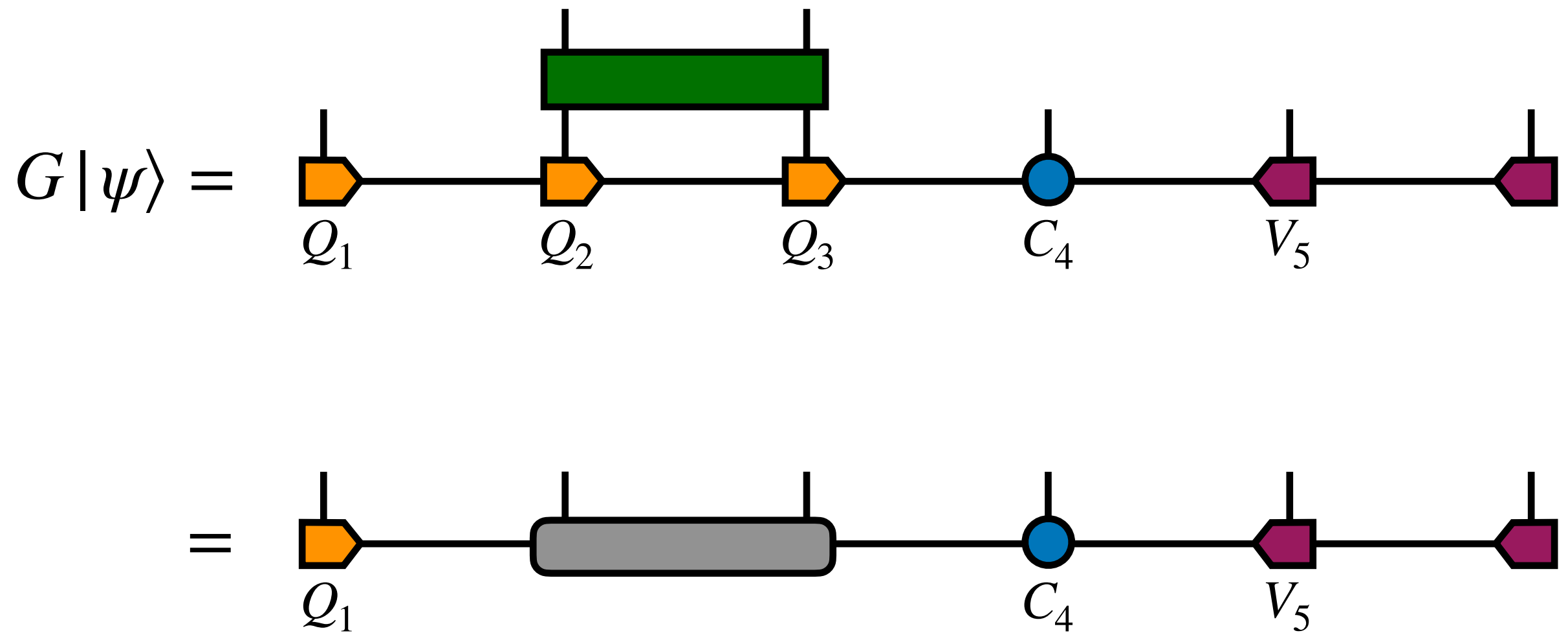
Using SVD instead of QR gives opportunity to "round" or truncate the bond dimension



Afterward, bond size has been reduced

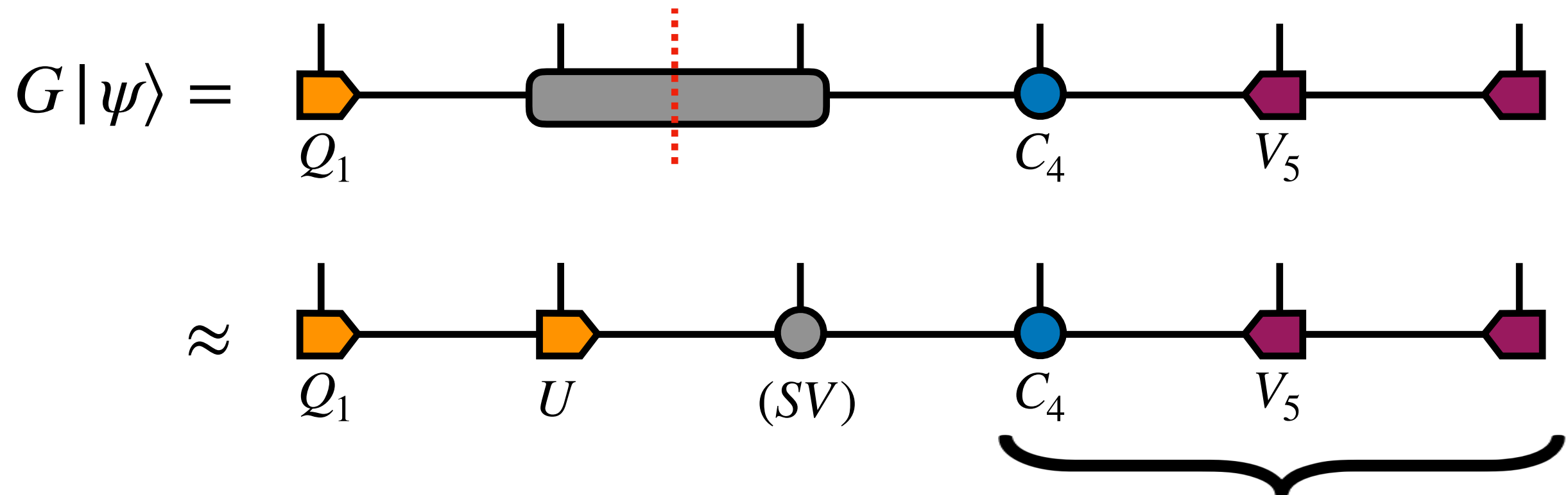
Importantly, truncation must happen within orthogonal gauge "center" – why?

Say we apply a gate away from the center



So far it is exact – no error

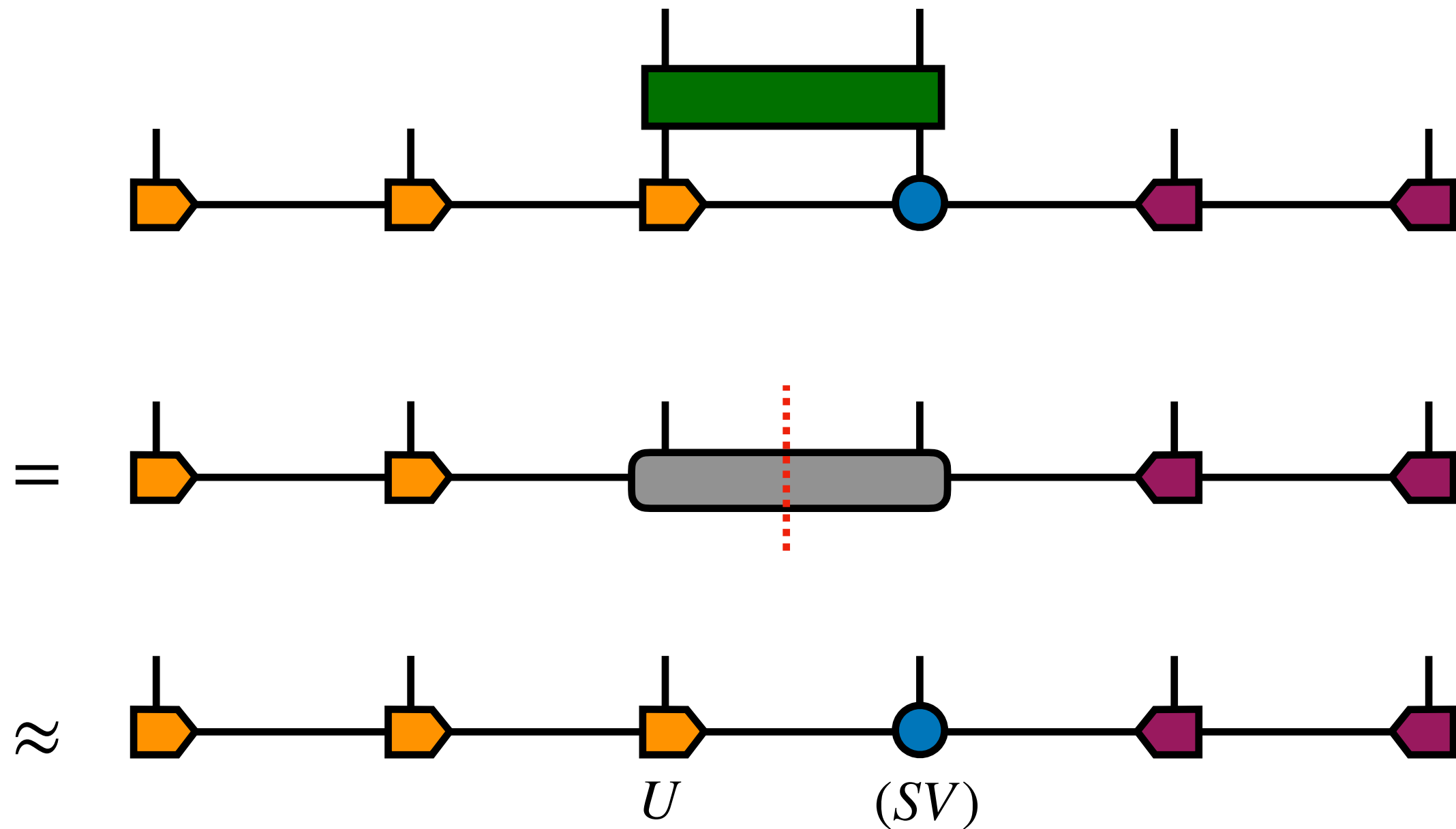
Now we can SVD & truncate small singular values



Not an orthonormal basis

Might make an **arbitrarily large** error

In contrast, truncating inside an orthonormal basis is controlled



local SVD error = global error