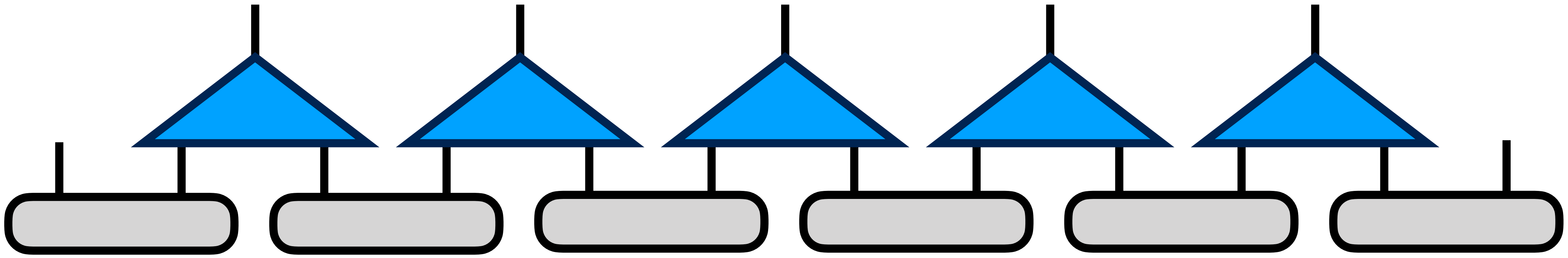
Tensor Networks and Applications



E.M. Stoudenmire

Apr 2018 - Sao Paulo

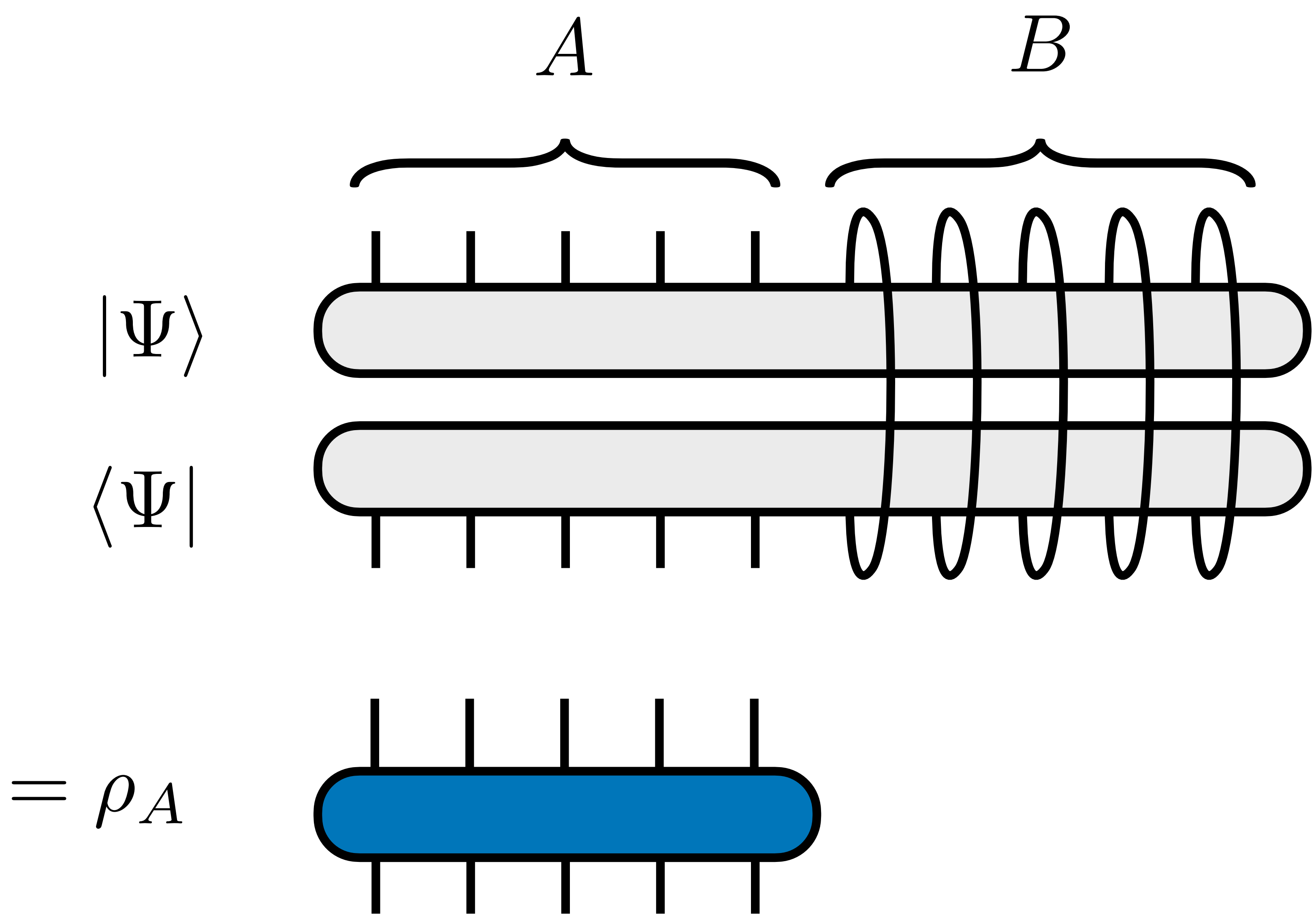
FLATIRON INSTITUTE

SIMONS FOUNDATION

Review of Previous Lecture

Reduced density matrix

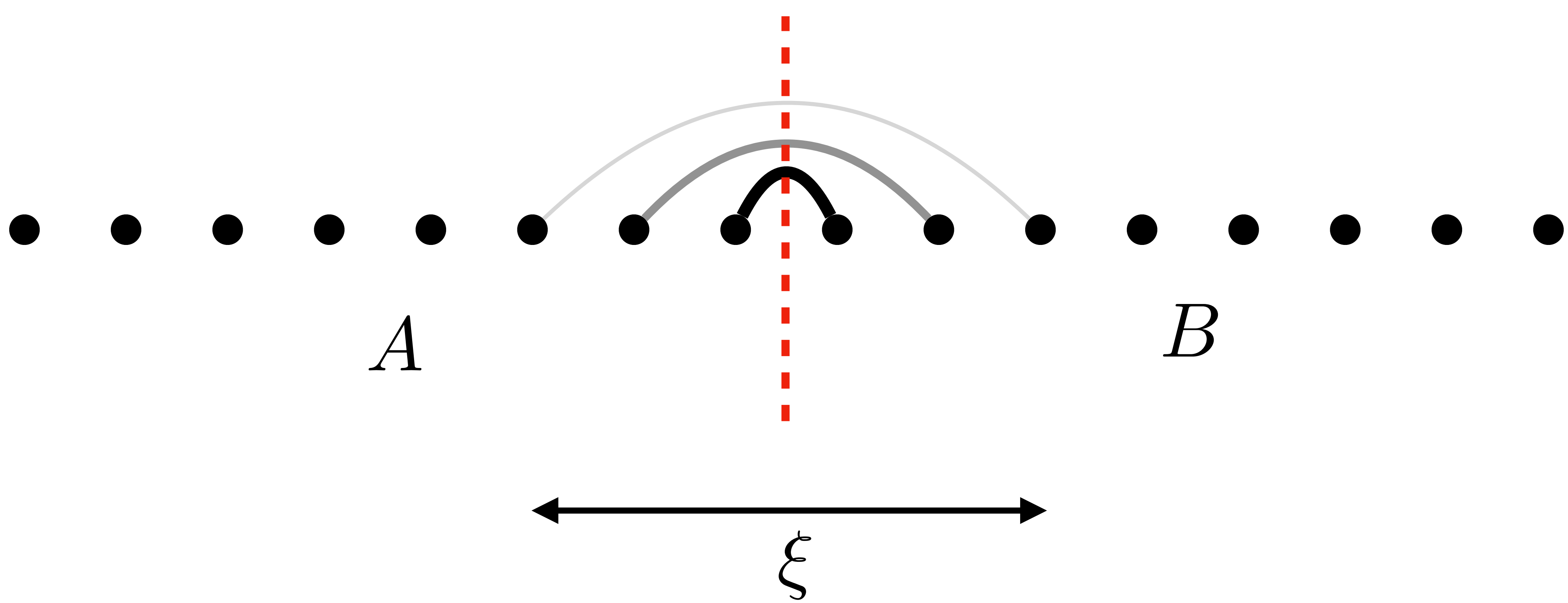
Trace over region B to get reduced density matrix of A



Eigenvalues of ρ_A define entanglement between A and B

Boundary law implies limited entanglement of ground states

Entanglement between A and B due to spins near boundary (for ground state of gapped, local Hamiltonian)



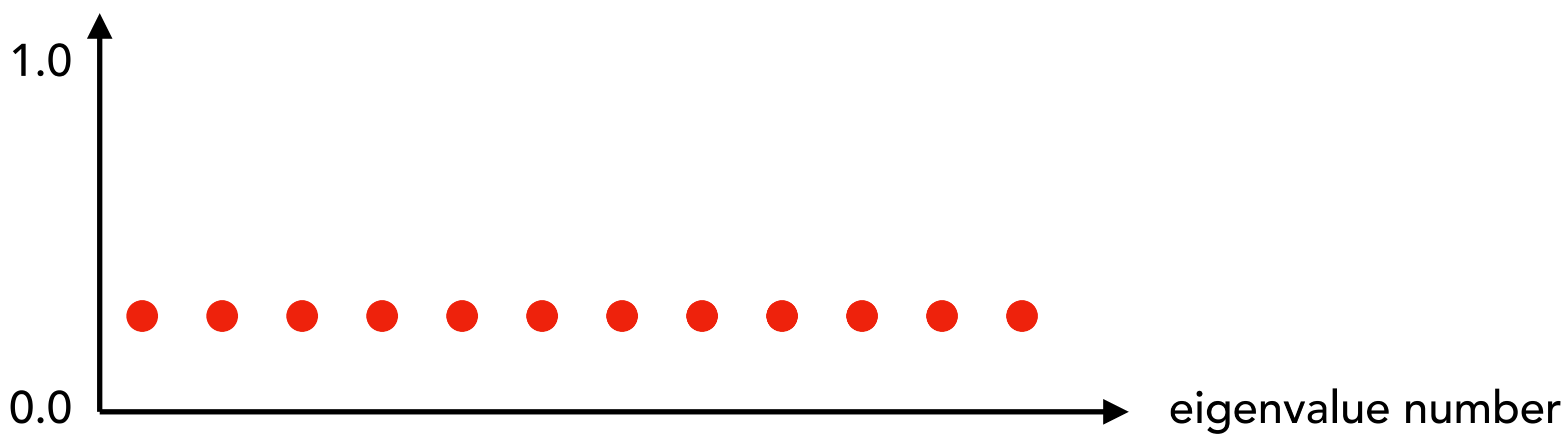
Local H and **gap** implies a correlation length ξ

Truncating Wavefunctions

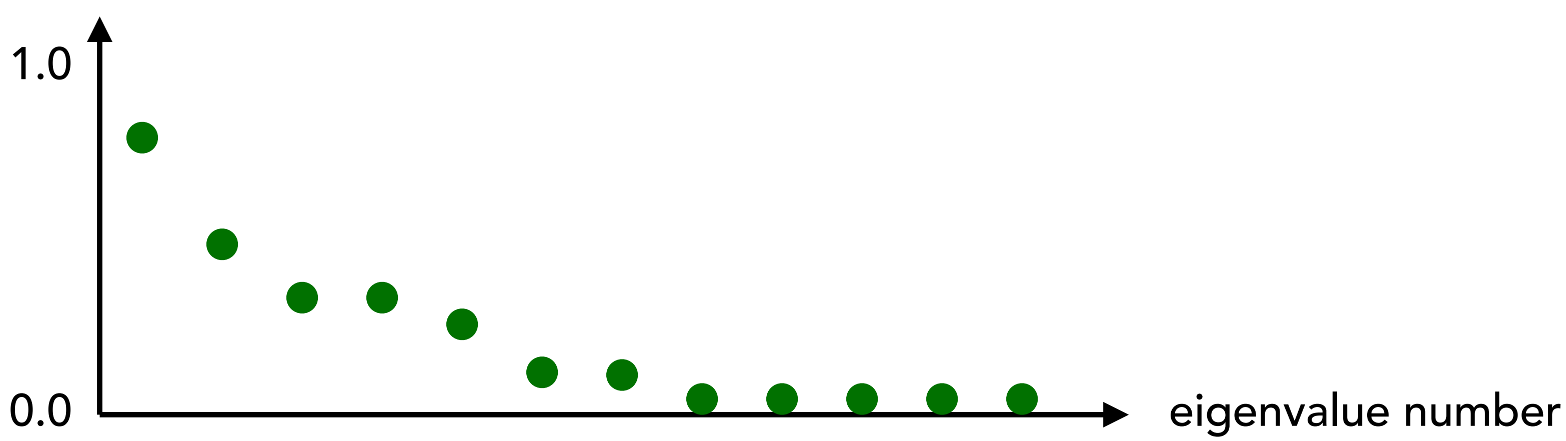
Boundary law means entanglement much less than it could be

So density matrix eigenvalues fall quickly...

Random state:

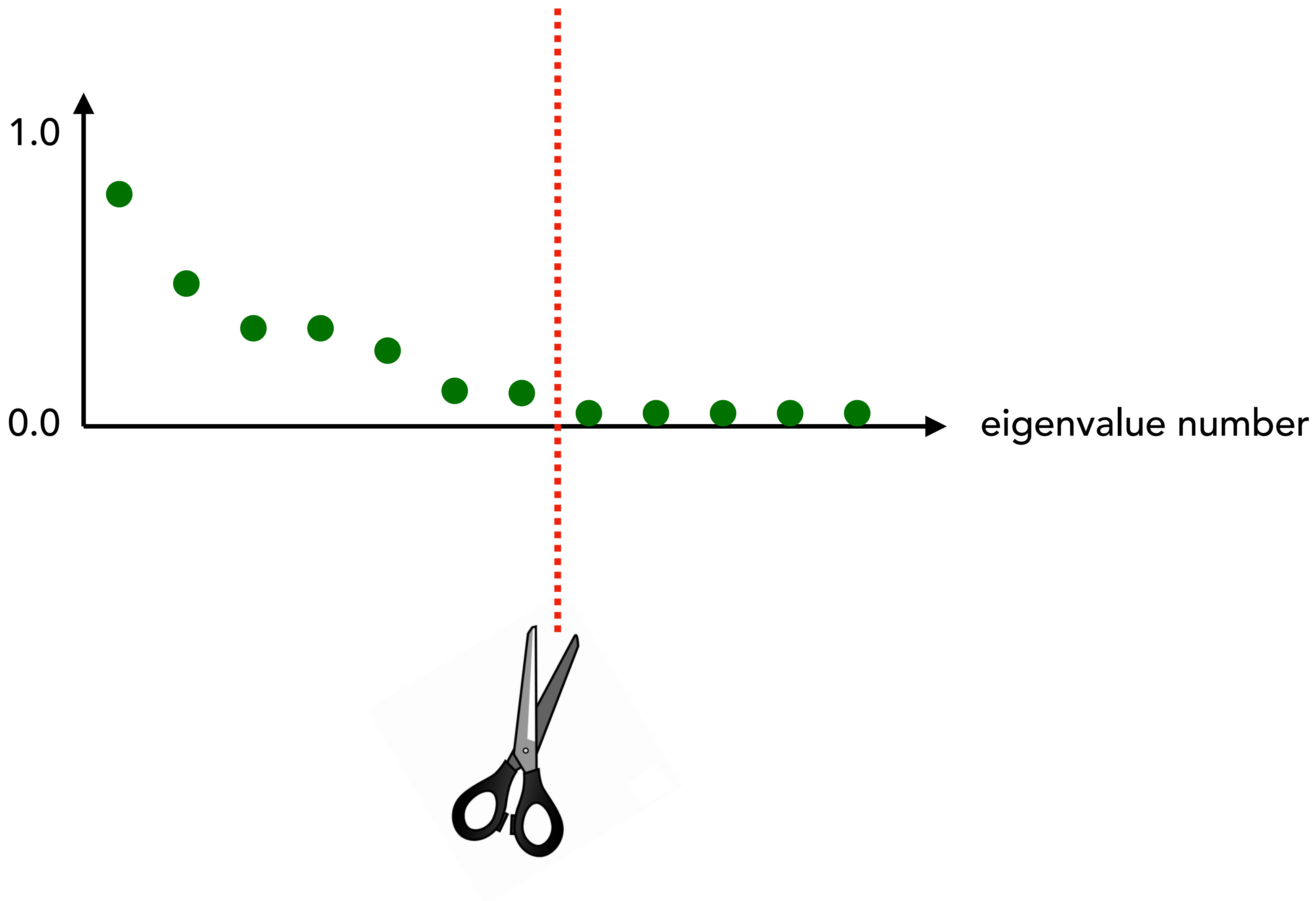


Ground state:



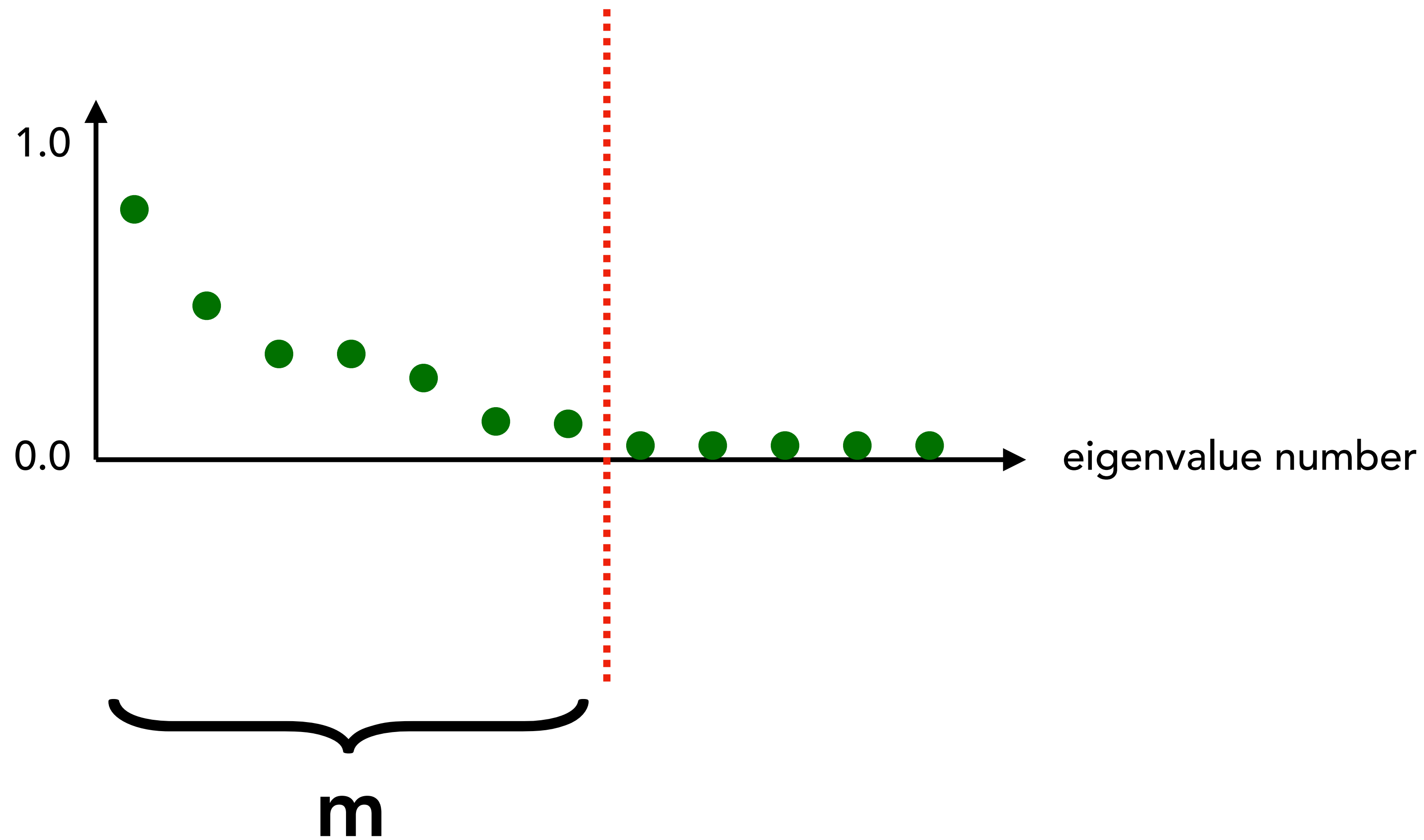
Suggests an approximation

Idea of **truncating** density matrix spectrum



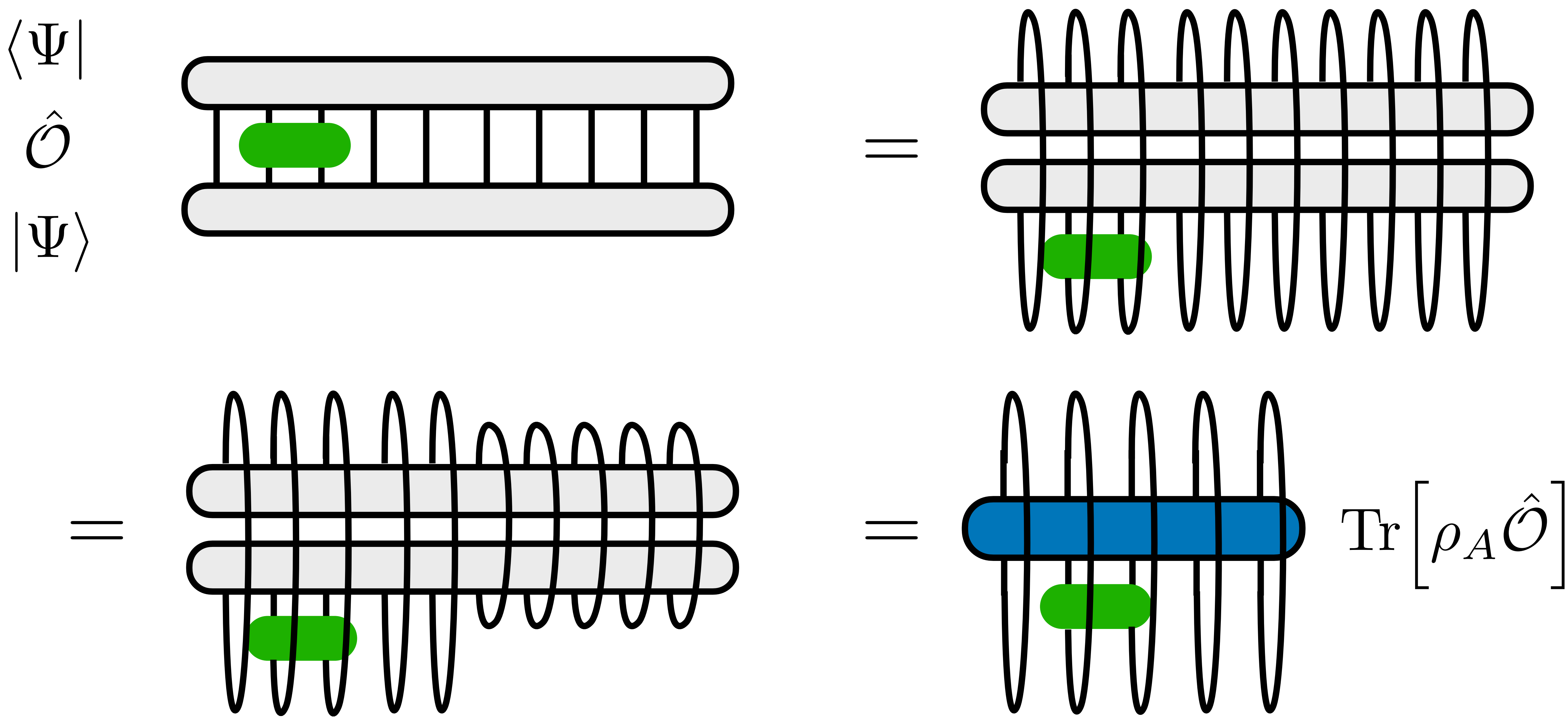
Implication of **boundary law** (= area law):

Number of eig. vals **m** needed for accuracy doesn't grow with N



Why approximate density matrix?

Reduced density matrix determines **all** observables in region A



Any observable in region A can be written as

$$\langle \hat{\mathcal{O}} \rangle = \langle \Psi | \hat{\mathcal{O}} | \Psi \rangle = \mathrm{Tr}\left[\rho_A \hat{\mathcal{O}} \right]$$

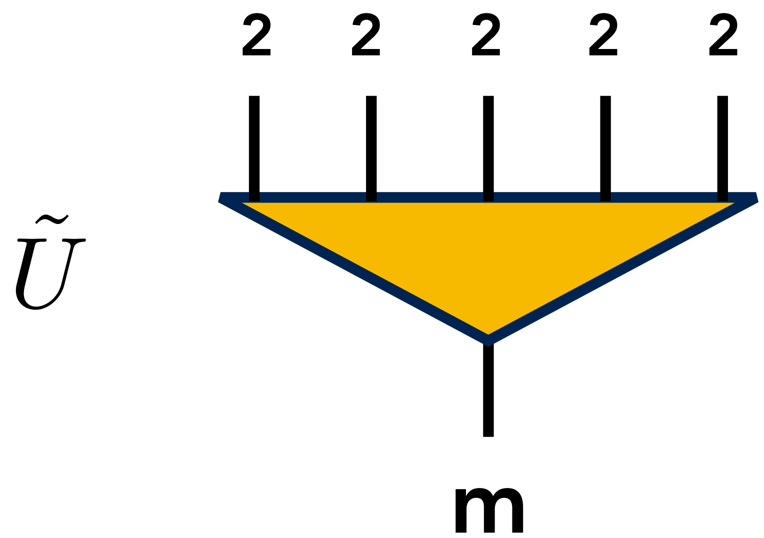
If we diagonalize $\rho_A = \sum_n p_n |n\rangle\langle n|$

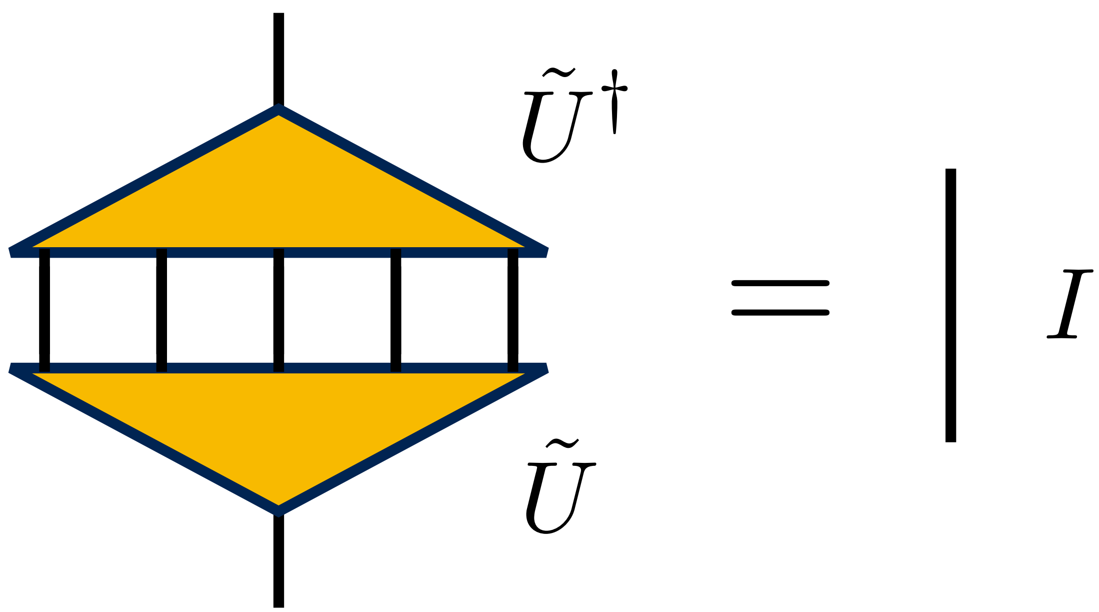
$$\langle \hat{\mathcal{O}} \rangle = \mathrm{Tr}\left[\rho_A \hat{\mathcal{O}} \right] = \sum_n p_n \langle n | \hat{\mathcal{O}} | n \rangle$$

discarding small p_n *means small error in* $\langle \hat{\mathcal{O}} \rangle$

Ability to truncate ρ_A means the following

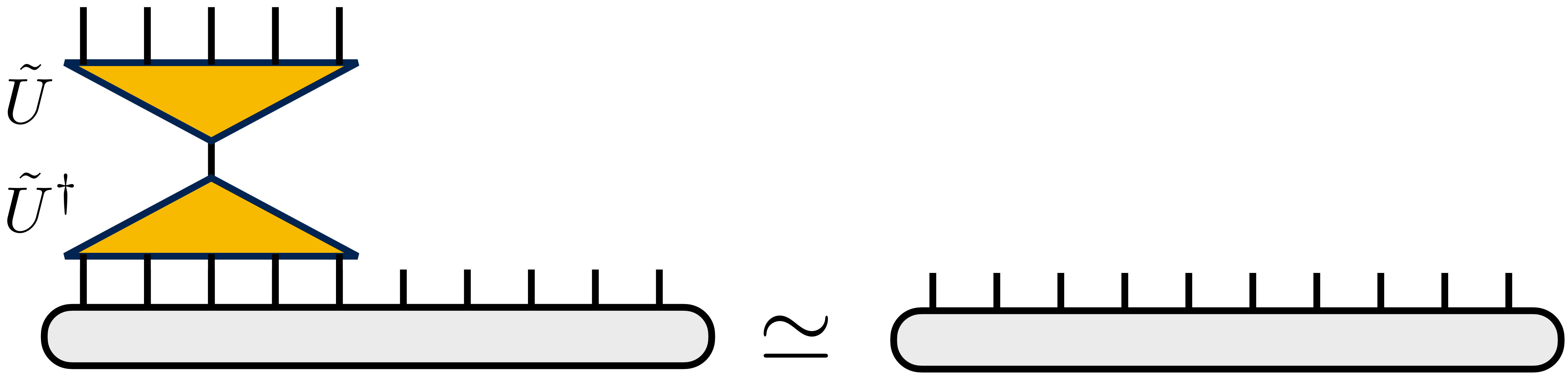
There is an isometry (rotation + projection) from sites of A to truncated basis of ρ_A





$\tilde{U}$ is first **m** columns of unitary diagonalizing ρ_A

Proposition: following approximation holds

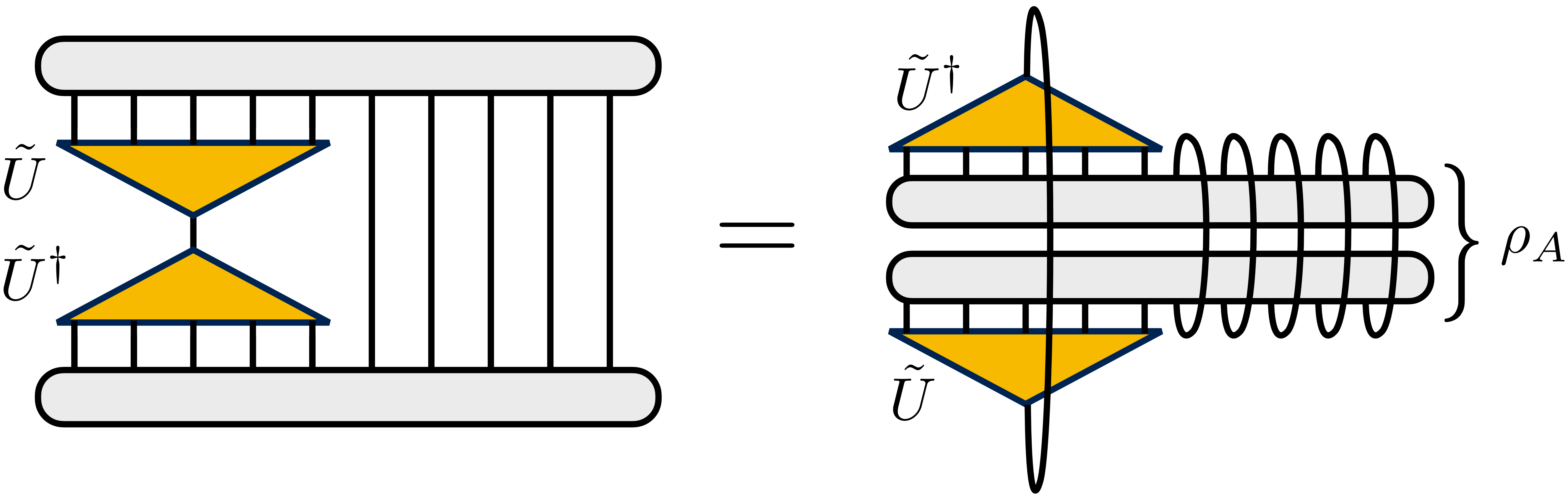


$$\tilde{U}\tilde{U}^\dagger|\Psi\rangle \simeq |\Psi\rangle + \mathcal{O}(\epsilon)$$

Where $\epsilon = \sum_{n=m+1}^{d_A} p_n$ is the sum of discarded eigenvalues of ρ_A

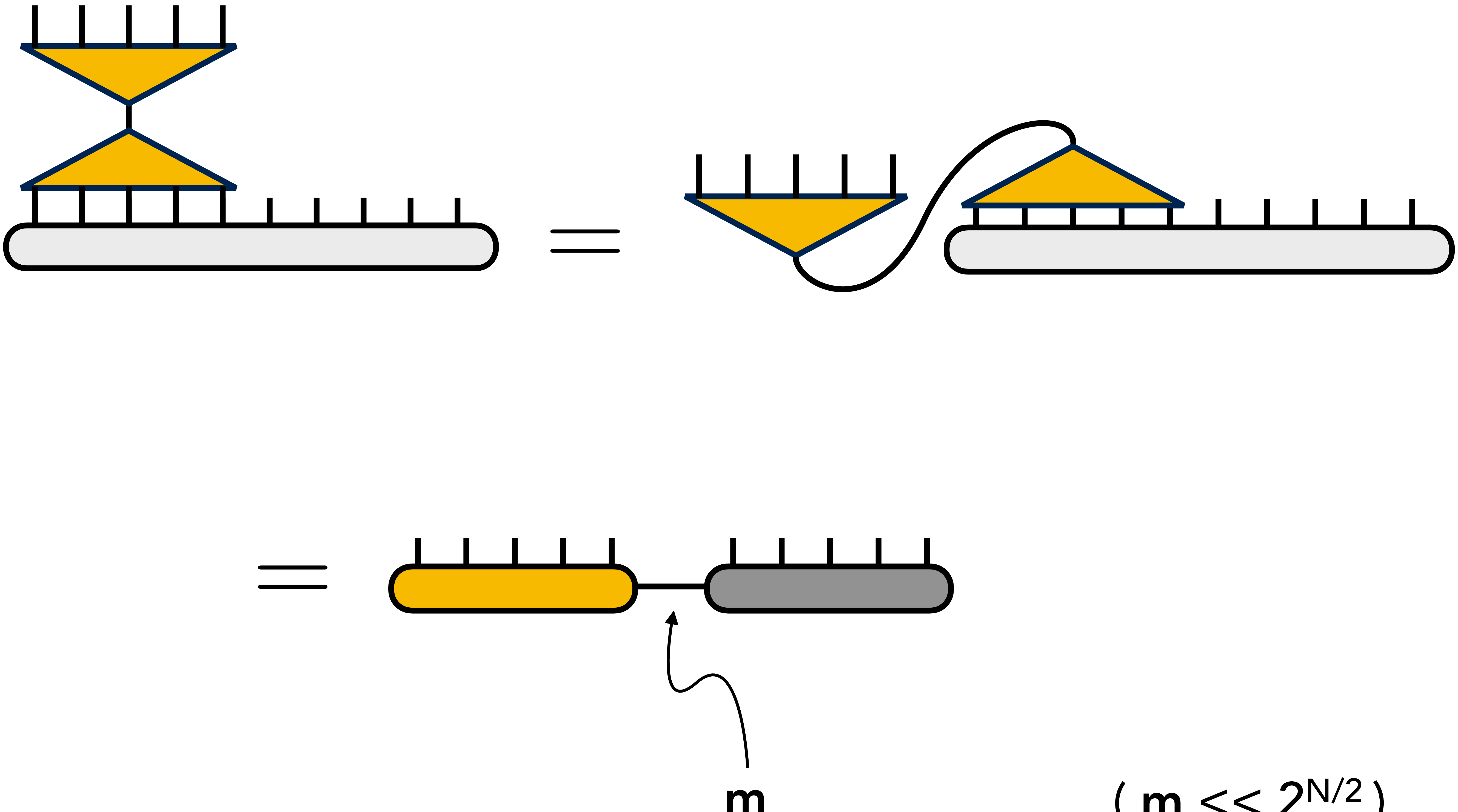
Why?

To prove, compute overlap $\langle\Psi|\left(\tilde{U}\tilde{U}^\dagger|\Psi\rangle\right)$



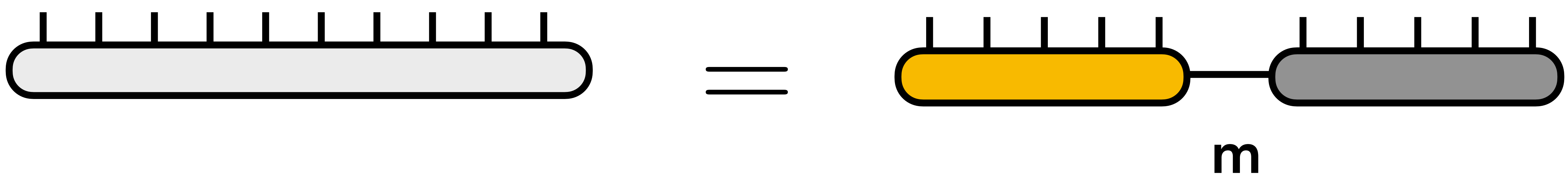
$$= \tilde{U}^\dagger \rho_A \tilde{U} = \sum_{n=1}^{m} p_n = \boxed{1-\epsilon}$$

Can approximately rewrite $|\Psi\rangle$ as follows



$(m << 2^{N/2})$

The upshot is we can factorize any ground state



2^N dimensional 'vector'

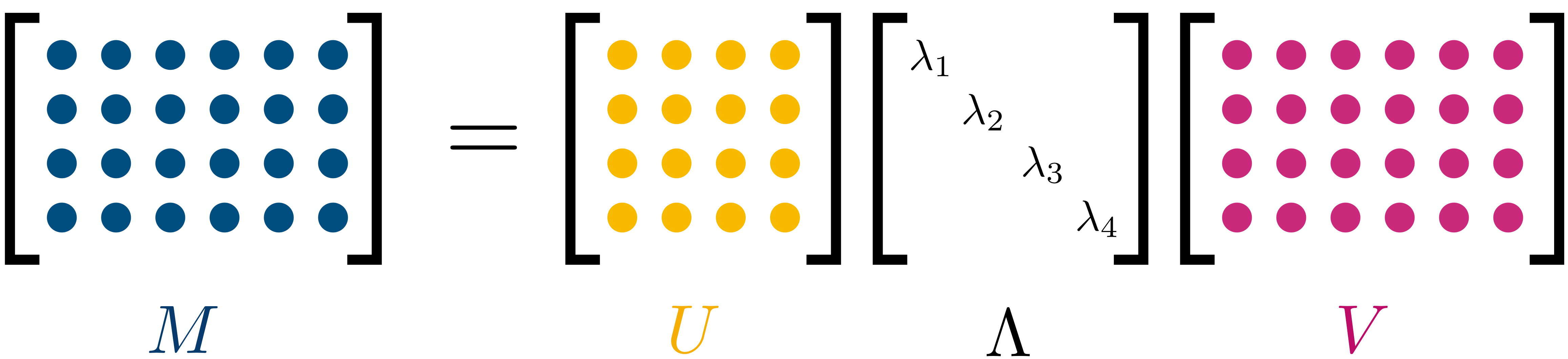
two ($2^{N/2}$ x **m**) 'matrices'

huge reduction in parameters!

Density matrix approach is "roundabout" however

Simpler approach is singular value decomposition (SVD)

Recall, for any rectangular matrix:



U and V are unitary, and singular values λ_n are real & positive

Consider a numerical SVD example:

$$M = \begin{bmatrix} 0.435839 & 0.223707 & 0.10 \\ 0.435839 & 0.223707 & -0.10 \\ 0.223707 & 0.435839 & 0.10 \\ 0.223707 & 0.435839 & -0.10 \end{bmatrix}$$

Can decompose as

$$\begin{bmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} 0.933 & 0 & 0 \\ 0 & 0.300 & 0 \\ 0 & 0 & 0.200 \end{bmatrix} \begin{bmatrix} 0.707107 & 0.707107 & 0 \\ -0.707107 & 0.707107 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Keep fewer and fewer singular values:

$$
\underset{U}{\begin{bmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \end{bmatrix}}
\underset{\Lambda}{\begin{bmatrix} 0.933 & 0 & 0 \\ 0 & 0.300 & 0 \\ 0 & 0 & 0.200 \end{bmatrix}}
\underset{V}{\begin{bmatrix} 0.707107 & 0.707107 & 0 \\ -0.707107 & 0.707107 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}
$$

$$
= M = \begin{bmatrix} 0.435839 & 0.223707 & 0.10 \\ 0.435839 & 0.223707 & -0.10 \\ 0.223707 & 0.435839 & 0.10 \\ 0.223707 & 0.435839 & -0.10 \end{bmatrix}
$$

$$
||M - M||^2 = 0
$$

Keep fewer and fewer singular values:

$$\underset{U}{\begin{bmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \end{bmatrix}} \underset{\Lambda}{\begin{bmatrix} 0.933 & 0 & 0 \\ 0 & 0.300 & 0 \\ 0 & 0 & 0 \end{bmatrix}} \underset{V}{\begin{bmatrix} 0.707107 & 0.707107 & 0 \\ -0.707107 & 0.707107 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$= M_2 = \begin{bmatrix} 0.435839 & 0.223707 & 0 \\ 0.435839 & 0.223707 & 0 \\ 0.223707 & 0.435839 & 0 \\ 0.223707 & 0.435839 & 0 \end{bmatrix}$$

$$||M_2 - M||^2 = 0.04 = (0.2)^2$$

Keep fewer and fewer singular values:

$$\underbrace{\begin{bmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 0.933 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} 0.707107 & 0.707107 & 0 \\ -0.707107 & 0.707107 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{V}$$

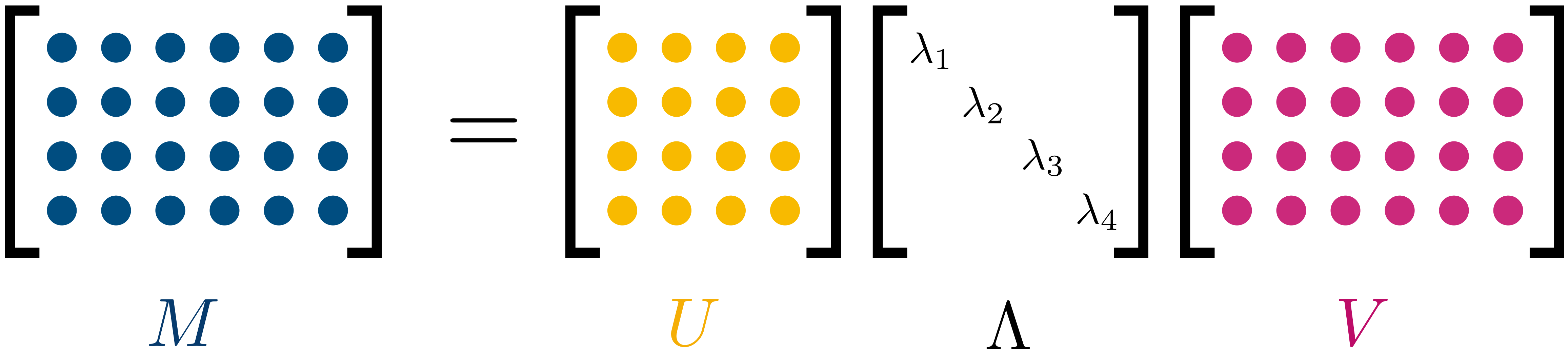
$$= M_3 = \begin{bmatrix} 0.329773 & 0.329773 & 0 \\ 0.329773 & 0.329773 & 0 \\ 0.329773 & 0.329773 & 0 \\ 0.329773 & 0.329773 & 0 \end{bmatrix}$$

Truncating SVD =

Controlled approximation for M

$$||M_3 - M||^2 = 0.13 = (0.3)^2 + (0.2)^2$$

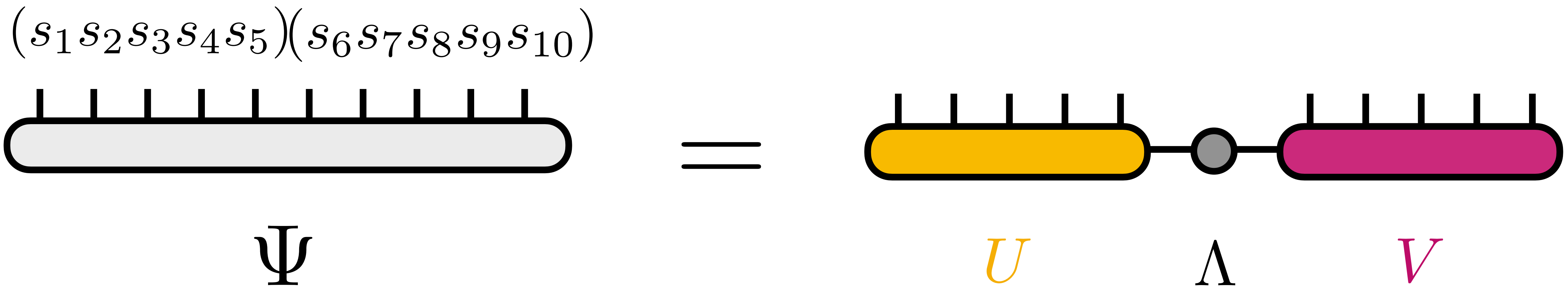
Diagrammatically, SVD looks like:



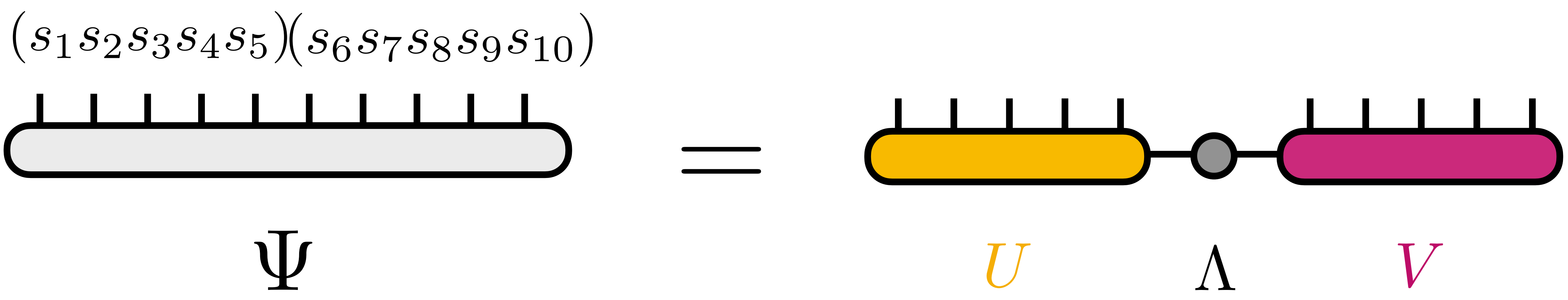
Let's apply it to our wavefunction

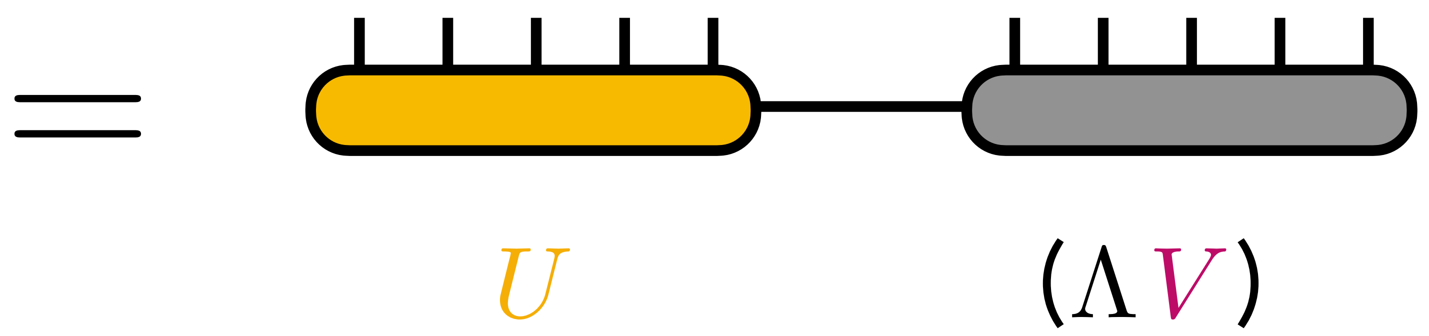
Treat wavefunction as matrix by grouping indices

$$\Psi^{(s_1 s_2 s_3 s_4 s_5)(s_6 s_7 s_8 s_9 s_{10})} = \sum_n U_n^{(s_1 s_2 s_3 s_4 s_5)} \Lambda_n V_n^{(s_6 s_7 s_8 s_9 s_{10})}$$



Multiplying singular vals into V, get same factorization as before



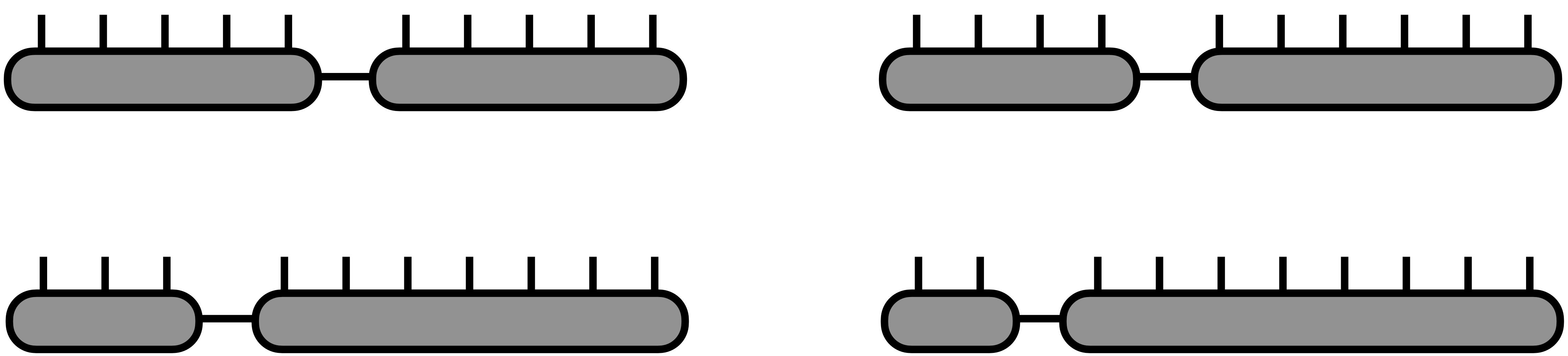


Matrix U is the *same* as from diagonalizing density matrix

Central idea of tensor networks:

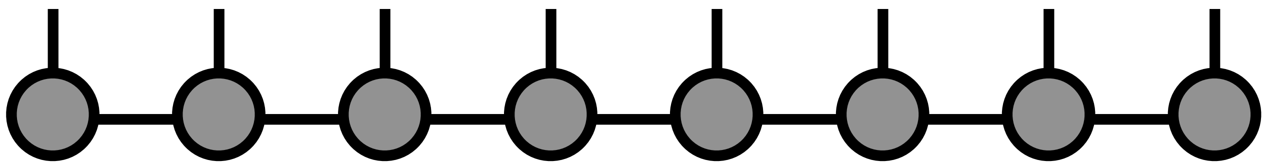
can factorize (s_1 s_2 s_3) (s_4 s_5 s_6), but what is special about this bond or partition?

why not factorize all partitions at once?



etc.

Motivates following factorization



Known as *matrix product state*

Key example of a tensor network

Matrix Product States

Wavefunction just a rule to map spin configurations to numbers $\Psi^{s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8}$

$$\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow \longrightarrow \Psi^{\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow}$$

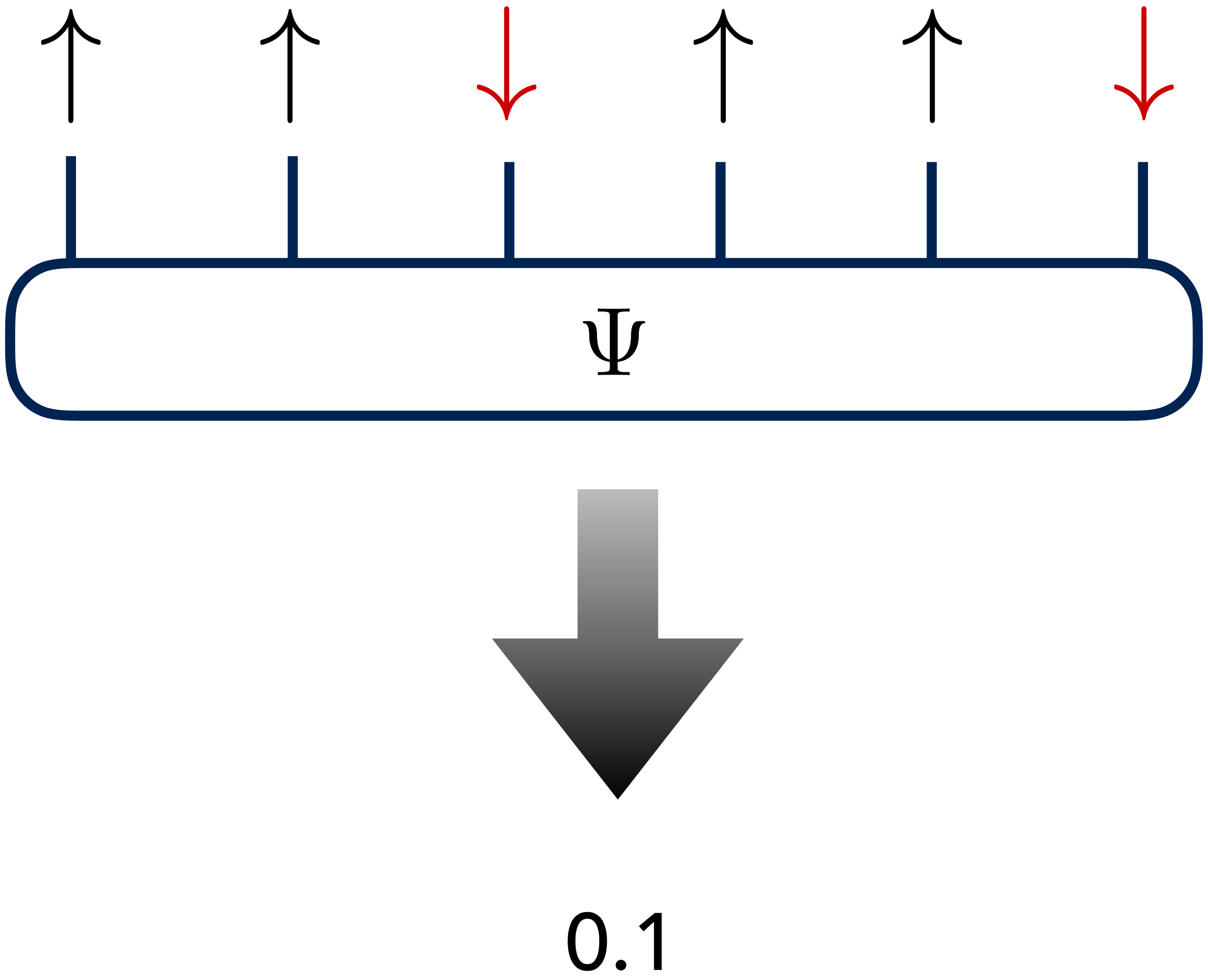
$$\uparrow\downarrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow \longrightarrow \Psi^{\uparrow\downarrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow}$$

$$\uparrow\uparrow\uparrow\uparrow\downarrow\uparrow\downarrow\uparrow \longrightarrow \Psi^{\uparrow\uparrow\uparrow\uparrow\downarrow\uparrow\downarrow\uparrow}$$

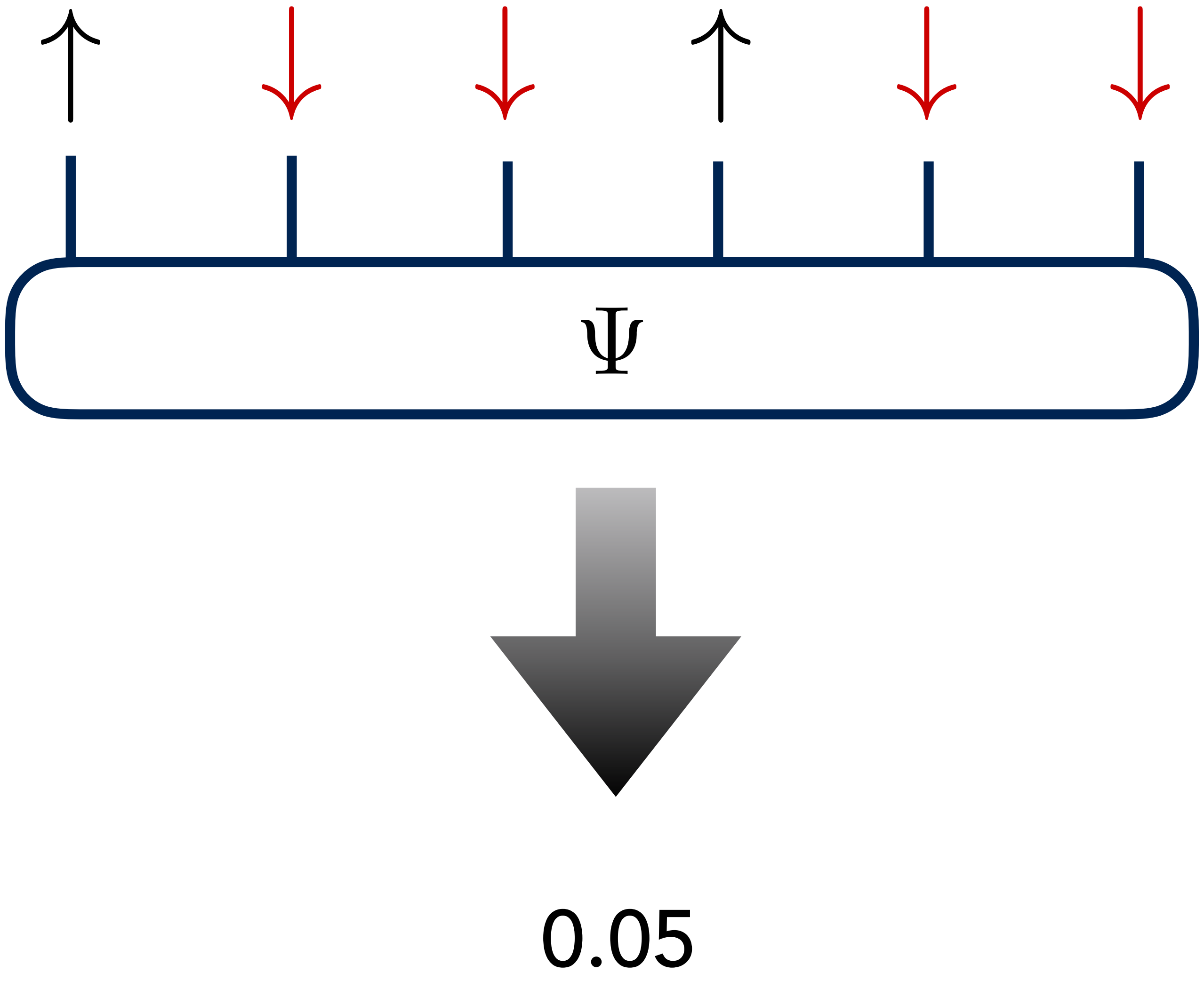
$$\uparrow\downarrow\downarrow\downarrow\downarrow\uparrow\uparrow\uparrow \longrightarrow \Psi^{\uparrow\downarrow\downarrow\downarrow\downarrow\uparrow\uparrow\uparrow}$$

Simplest rule: store every amplitude separately

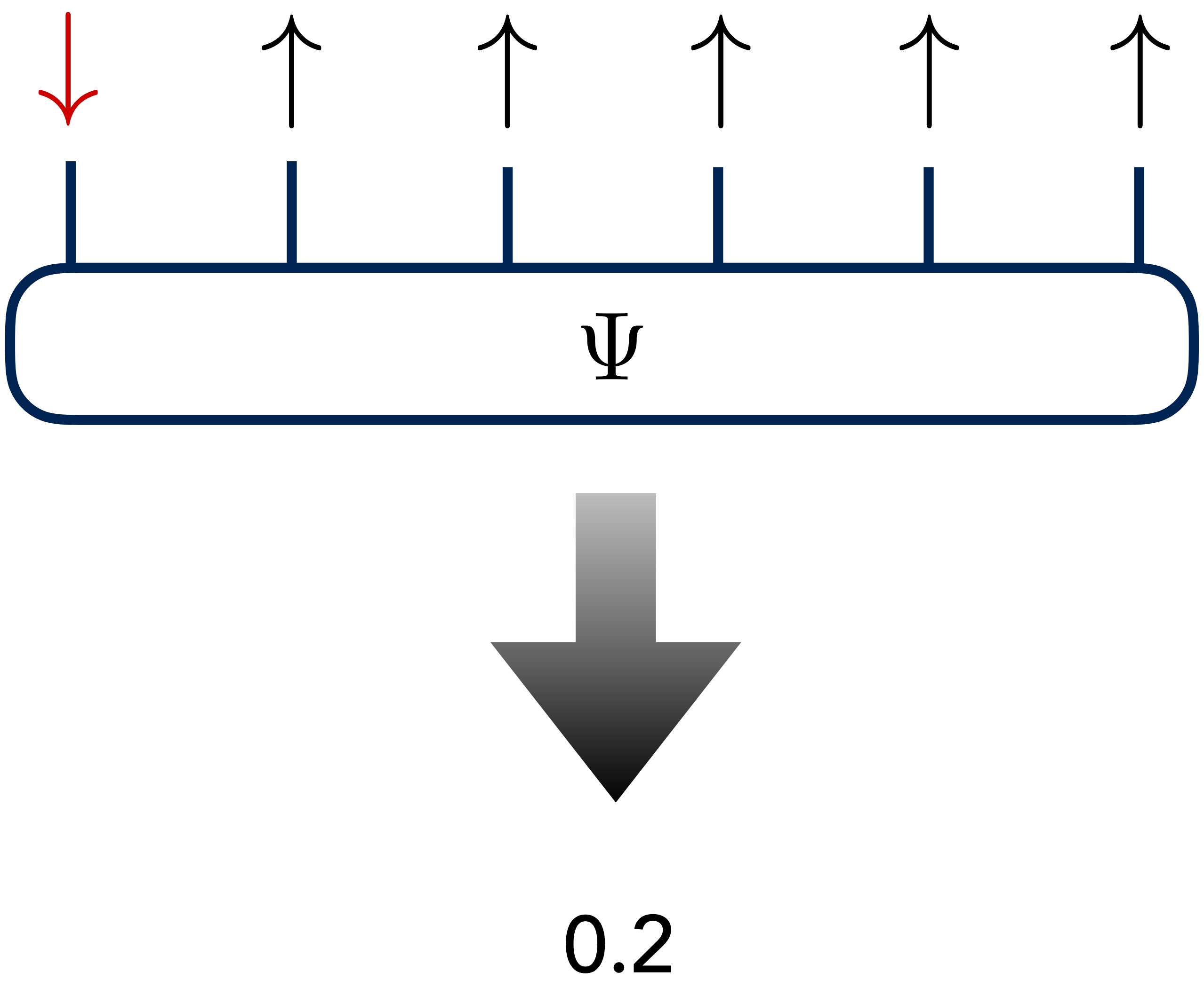
Wavefunction a "machine" mapping configurations to numbers



Wavefunction a "machine" mapping configurations to numbers



Can make up any rule assigning patterns to numbers



How about this rule:

- to each spin state (up, down) associate a matrix
- multiply matrices to get the probability

Pictorially:

$\uparrow$

$\downarrow$

Pictorially:

$$\uparrow \quad \longrightarrow \quad M^{\uparrow} = \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}$$

$$\downarrow \quad \longrightarrow \quad M^{\downarrow} = \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\uparrow}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\uparrow} M_4^{\uparrow}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\uparrow} M_4^{\uparrow} M_5^{\downarrow}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\uparrow} M_4^{\uparrow} M_5^{\downarrow}$$

$$\Psi^{\uparrow\uparrow\downarrow\downarrow\downarrow} \approx M_1^{\uparrow} M_2^{\uparrow} M_3^{\downarrow} M_4^{\downarrow} M_5^{\downarrow}$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\uparrow} M_4^{\uparrow} M_5^{\downarrow}$$

$$\Psi^{\uparrow\uparrow\downarrow\downarrow\downarrow} \approx M_1^{\uparrow} M_2^{\uparrow} M_3^{\downarrow} M_4^{\downarrow} M_5^{\downarrow}$$

$$\Psi^{\uparrow\downarrow\downarrow\uparrow\uparrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\downarrow} M_4^{\uparrow} M_5^{\uparrow}$$

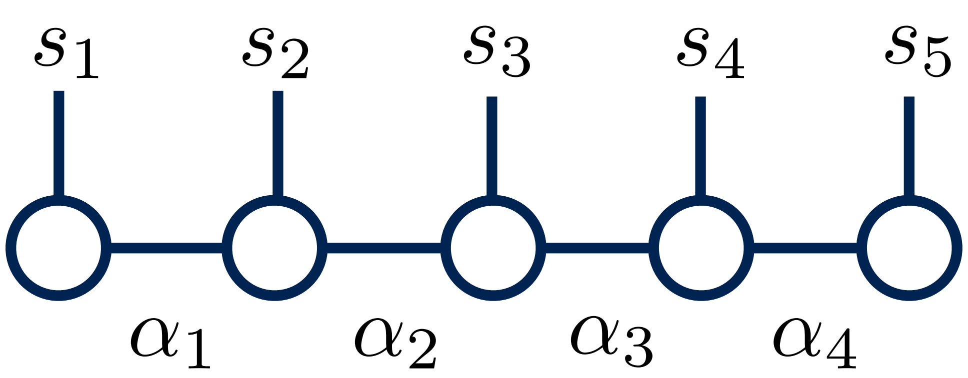
Ansatz known as ***matrix product state***

$$\Psi^{s_1 s_2 s_3 s_4 s_5} = M_1^{s_1} M_2^{s_2} M_3^{s_3} M_4^{s_4} M_5^{s_5}$$

More detail & diagrammatic form

$$\Psi^{s_1 s_2 s_3 s_4 s_5} = M_1^{s_1} M_2^{s_2} M_3^{s_3} M_4^{s_4} M_5^{s_5}$$

$$= \sum_{\{\alpha\}} M_{\alpha_1}^{s_1} M_{\alpha_1 \alpha_2}^{s_2} M_{\alpha_2 \alpha_3}^{s_3} M_{\alpha_3 \alpha_4}^{s_4} M_{\alpha_4}^{s_5}$$

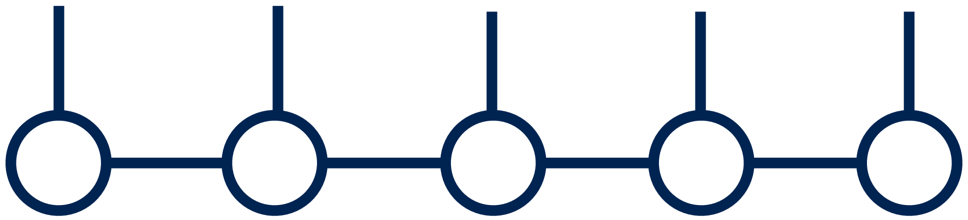
$=$ 

More detail & diagrammatic form

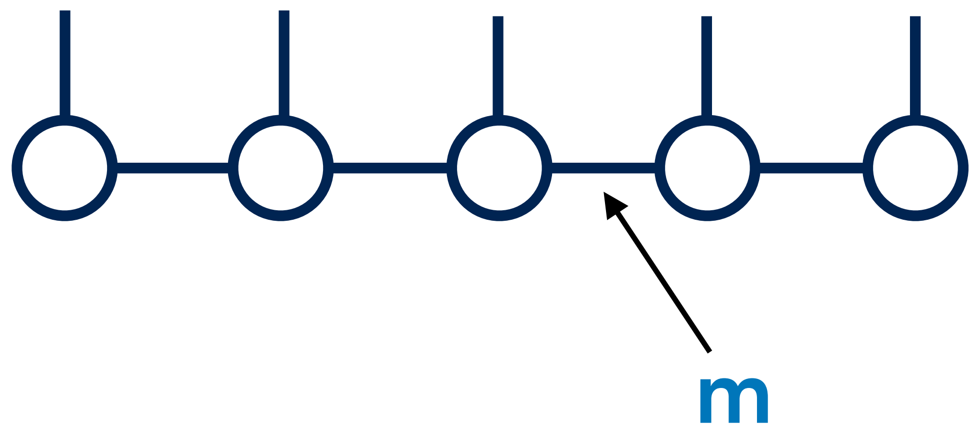
$$\Psi^{s_1 s_2 s_3 s_4 s_5} = M_1^{s_1} M_2^{s_2} M_3^{s_3} M_4^{s_4} M_5^{s_5}$$

$$= \sum_{\{\alpha\}} M_{\alpha_1}^{s_1} M_{\alpha_1 \alpha_2}^{s_2} M_{\alpha_2 \alpha_3}^{s_3} M_{\alpha_3 \alpha_4}^{s_4} M_{\alpha_4}^{s_5}$$

$$=$$



Key facts about matrix product states



- linear size of matrices (dimension of bond indices) known as the *bond dimension* **m** (sometimes χ or D)

- for large enough **m**, can represent *any state* ($m = 2^{N/2}$)

- entanglement of left-right cut bounded by log(**m**), so boundary law guaranteed

Computations with Matrix Product States

Key reason to use matrix product states (MPS) is many computations become *efficient*

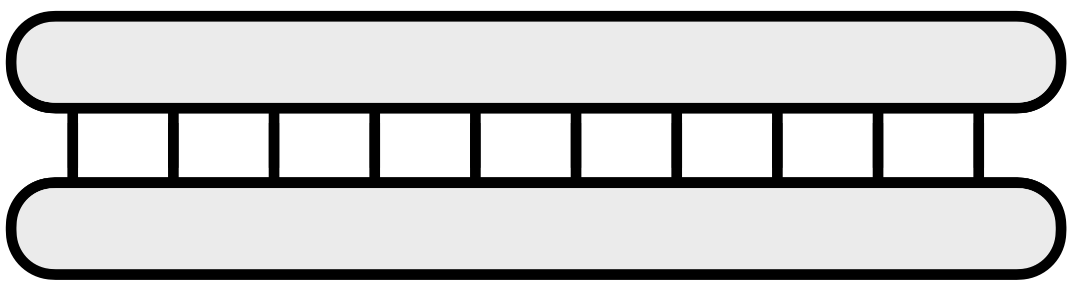
Key reason to use matrix product states (MPS) is many computations become *efficient*

By contrast, all operations with full wavefunction inefficient

Key reason to use matrix product states (MPS) is many computations become *efficient*

By contrast, all operations with full wavefunction inefficient

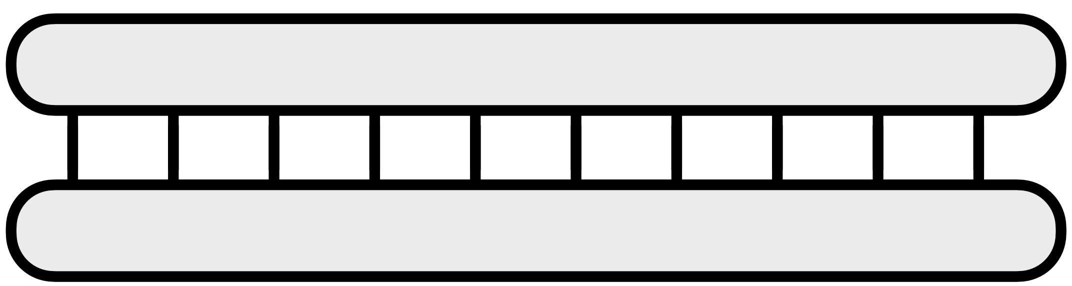
Consider computing norm of full wavefunction

$\langle \Psi | \Psi \rangle$ 

Key reason to use matrix product states (MPS) is many computations become *efficient*

By contrast, all operations with full wavefunction inefficient

Consider computing norm of full wavefunction

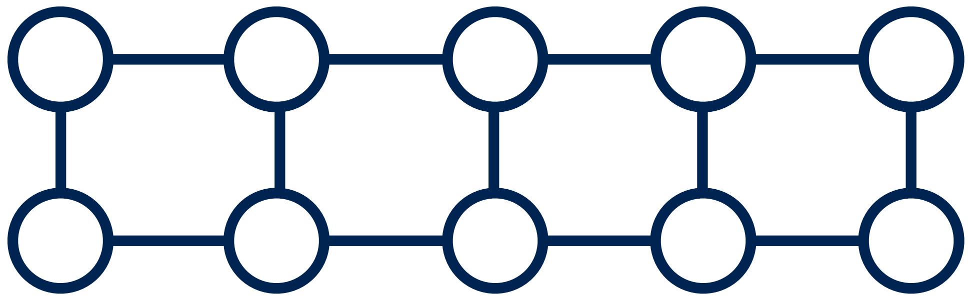
$\langle \Psi | \Psi \rangle$ 

$$= \sum_{\{s\}} \Psi^{s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10}} \bar{\Psi}_{s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10}}$$

Requires summing 2^{10} terms

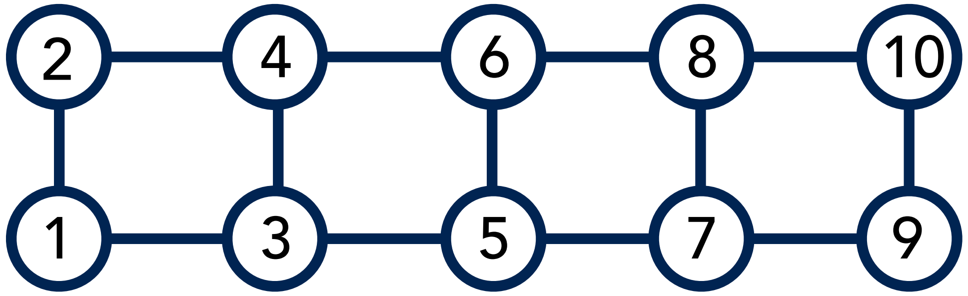
Is there a more efficient strategy to compute norm of MPS?

$$\langle \Psi | \Psi \rangle =$$



Is there a more efficient strategy to compute norm of MPS?

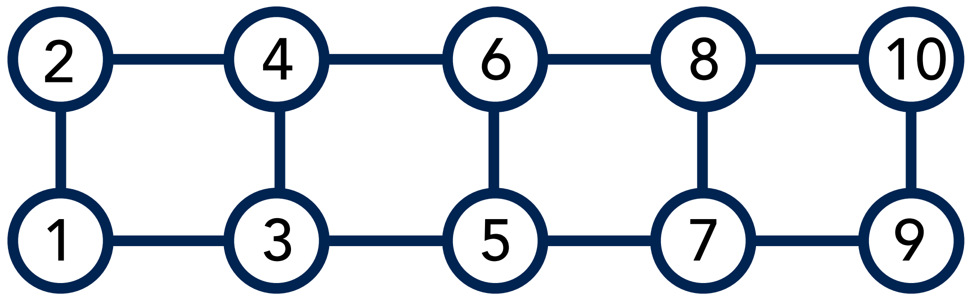
$$\langle \Psi | \Psi \rangle =$$



Yes!

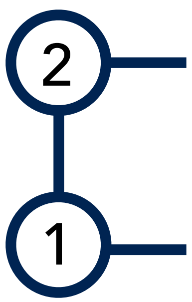
Is there a more efficient strategy to compute norm of MPS?

$$\langle \Psi | \Psi \rangle =$$

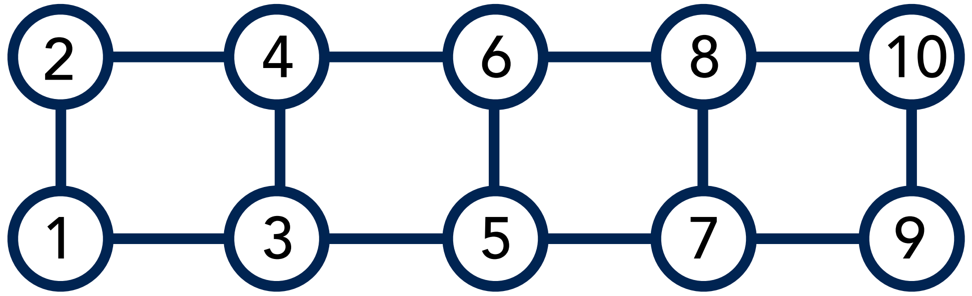


Yes!

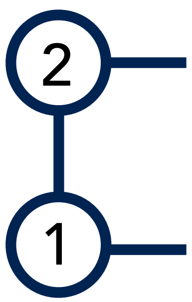
(1)

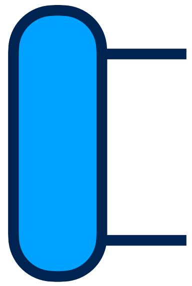


Is there a more efficient strategy to compute norm of MPS?

$\langle \Psi | \Psi \rangle =$ 

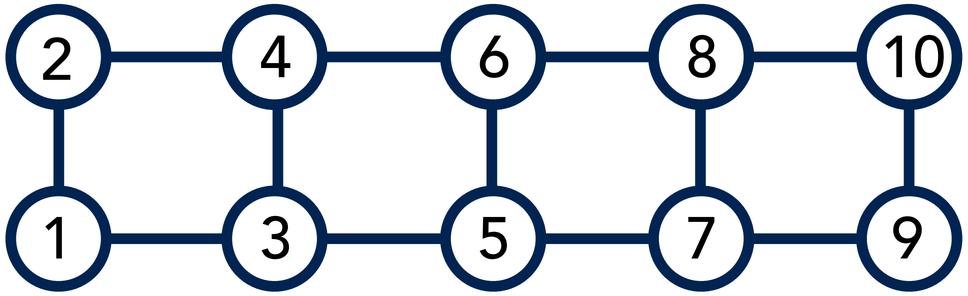
Yes!

(1) 

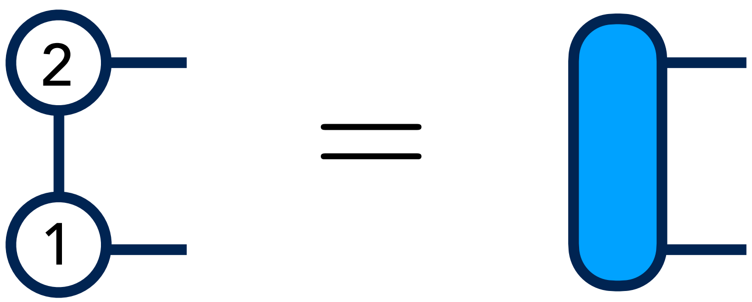
$=$ 

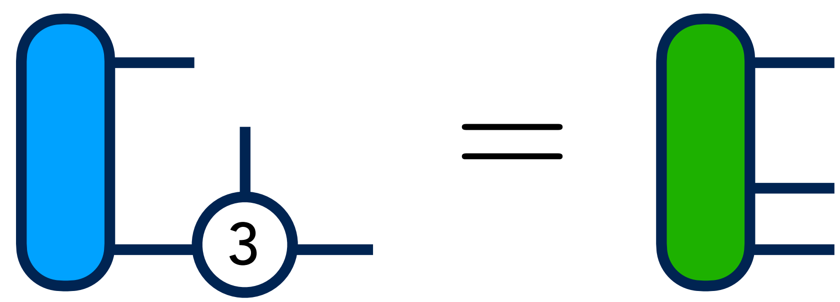
Is there a more efficient strategy to compute norm of MPS?

$$\langle \Psi | \Psi \rangle =$$

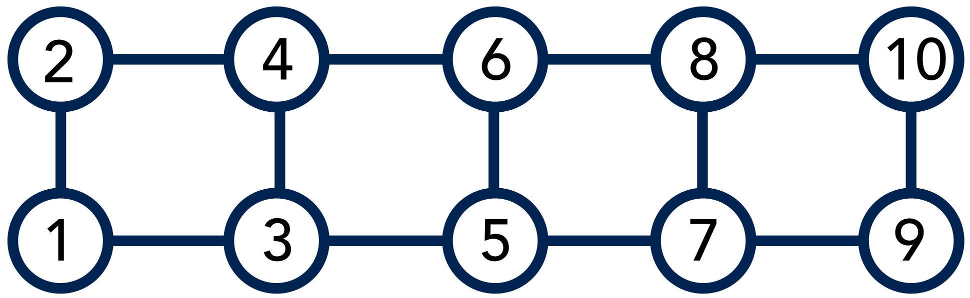


Yes!

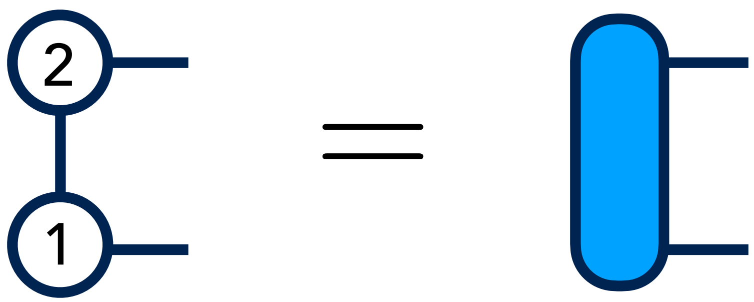
(1) 

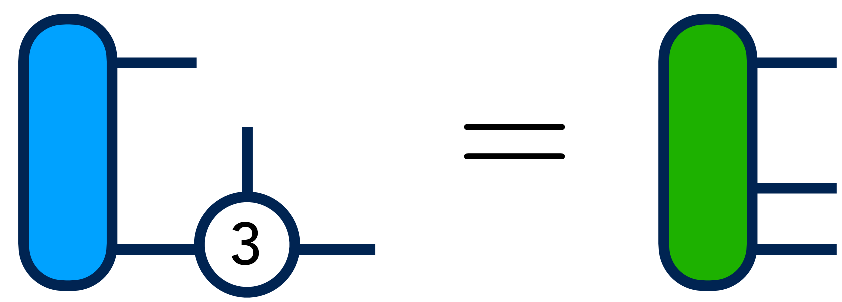
(2) 

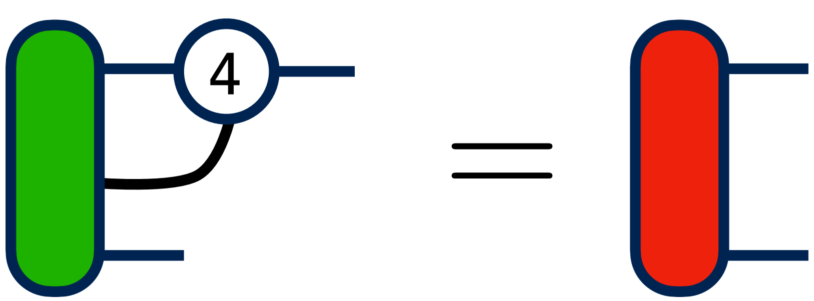
Is there a more efficient strategy to compute norm of MPS?

$\langle \Psi | \Psi \rangle =$ 

Yes!

(1) 

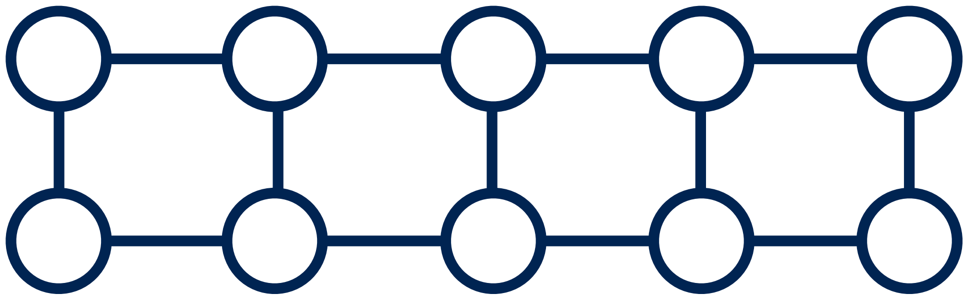
(2) 

(3) 

etc. until all tensors are contracted

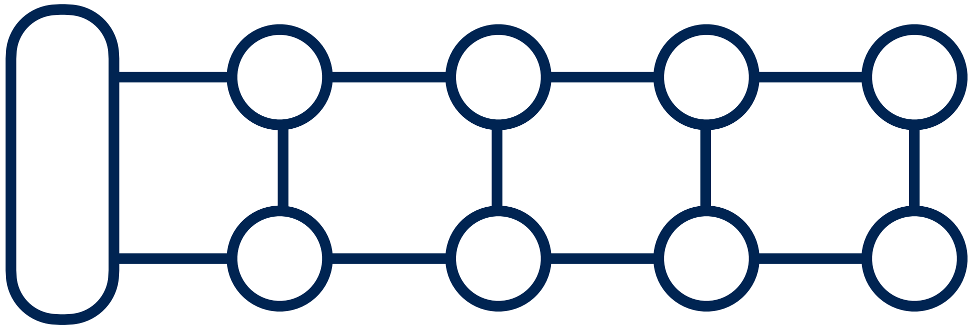
Full contraction process:

$$\langle \Psi | \Psi \rangle =$$



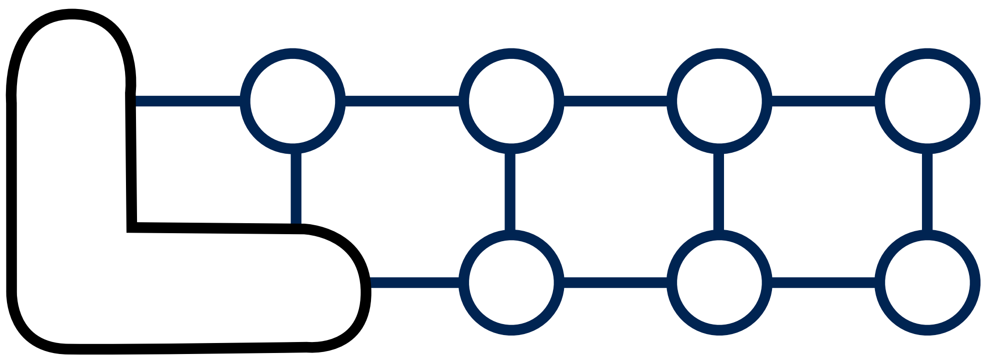
Full contraction process:

$$\langle \Psi | \Psi \rangle =$$



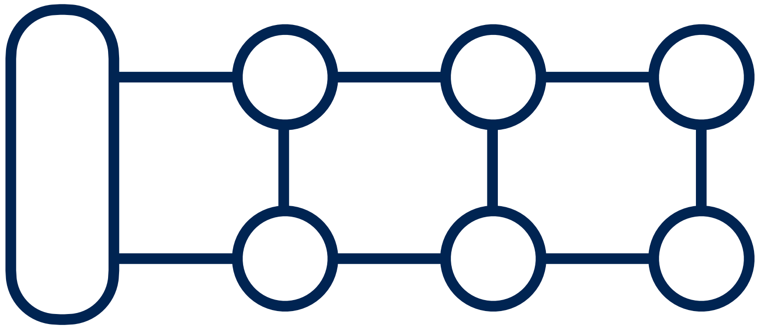
Full contraction process:

$$\langle \Psi | \Psi \rangle =$$



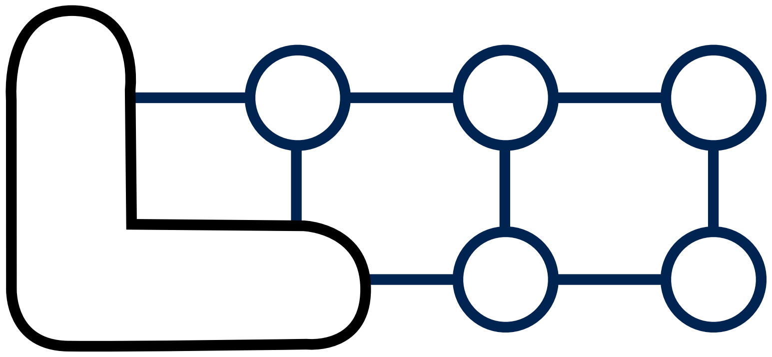
Full contraction process:

$$\langle \Psi | \Psi \rangle =$$



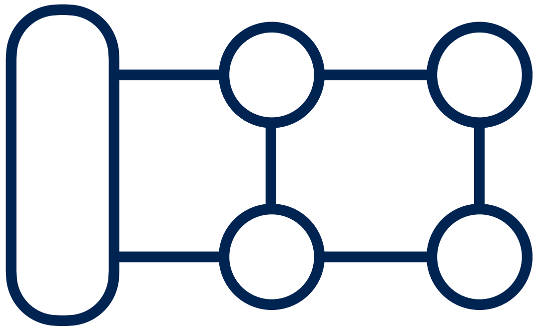
Full contraction process:

$$\langle \Psi | \Psi \rangle =$$



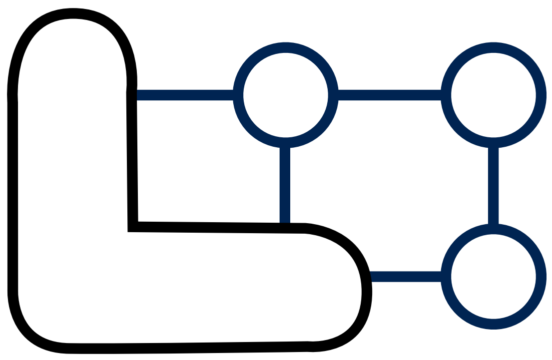
Full contraction process:

$$\langle\Psi|\Psi\rangle =$$



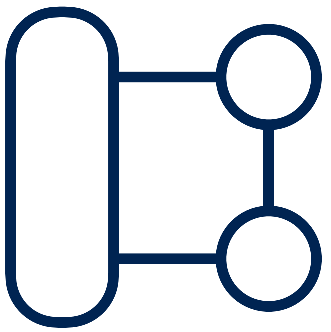
Full contraction process:

$$\langle\Psi|\Psi\rangle =$$



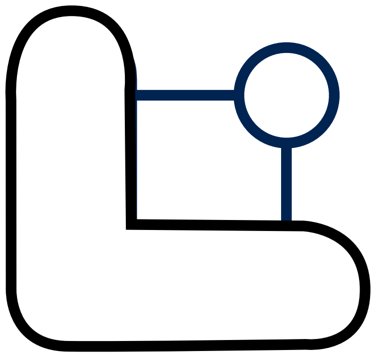
Full contraction process:

$$\langle \Psi | \Psi \rangle =$$



Full contraction process:

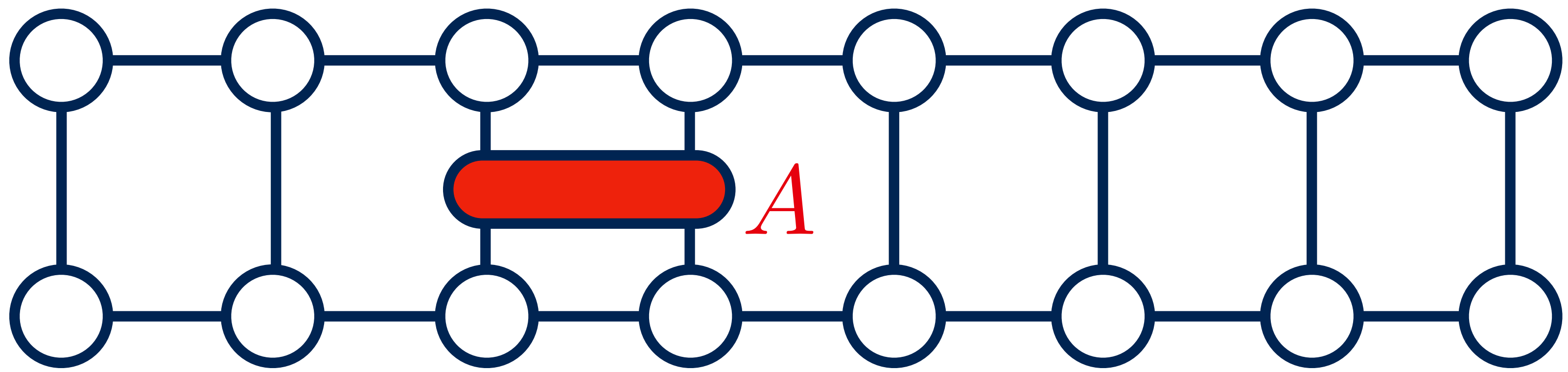
$$\langle\Psi|\Psi\rangle =$$



equals a scalar (**why?**)

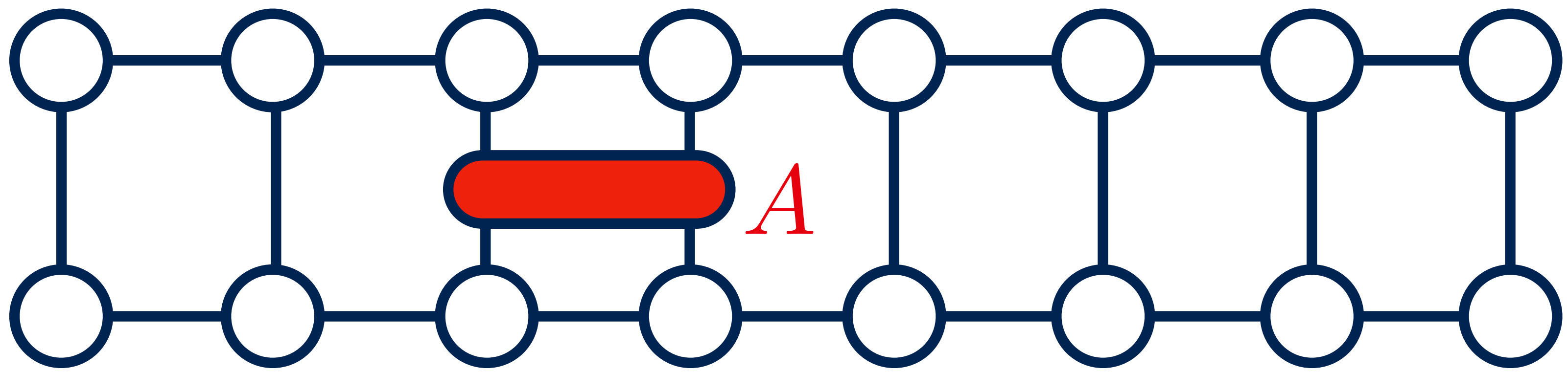
Another key computation is an **expectation value**

$$\langle\Psi|A|\Psi\rangle =$$

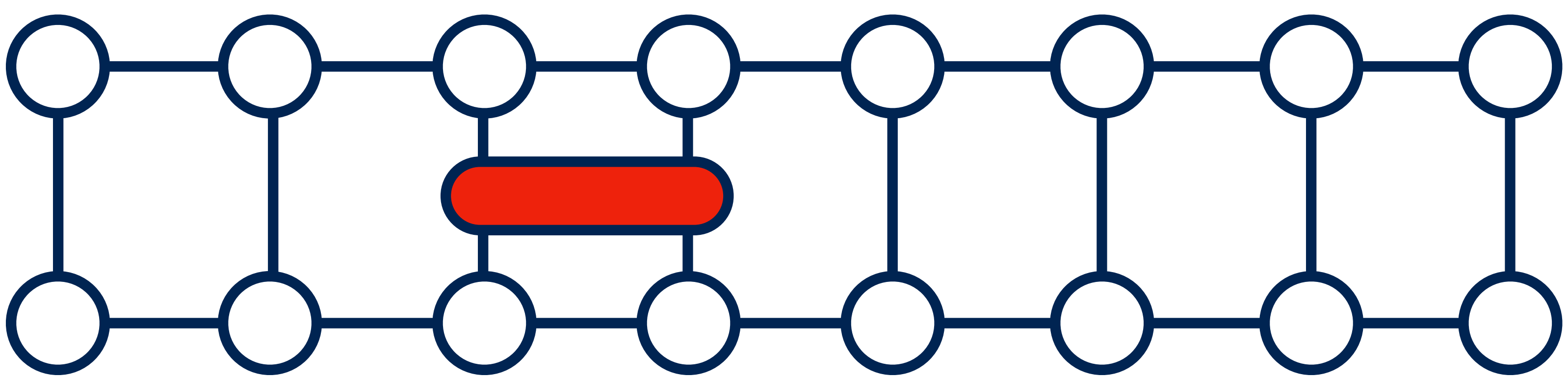


Another key computation is an **expectation value**

$$\langle \Psi | A | \Psi \rangle =$$

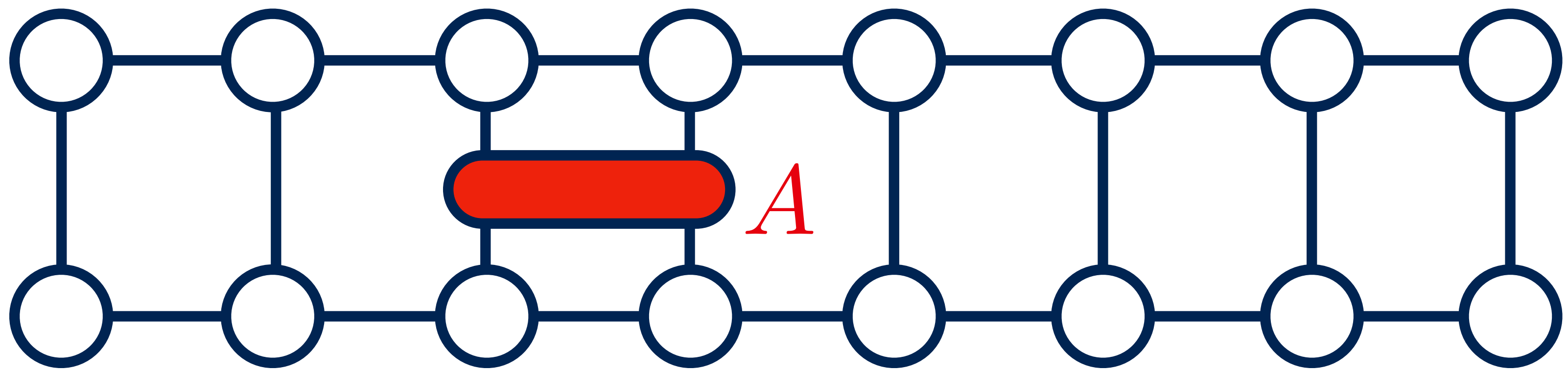


Using similar procedure as norm:

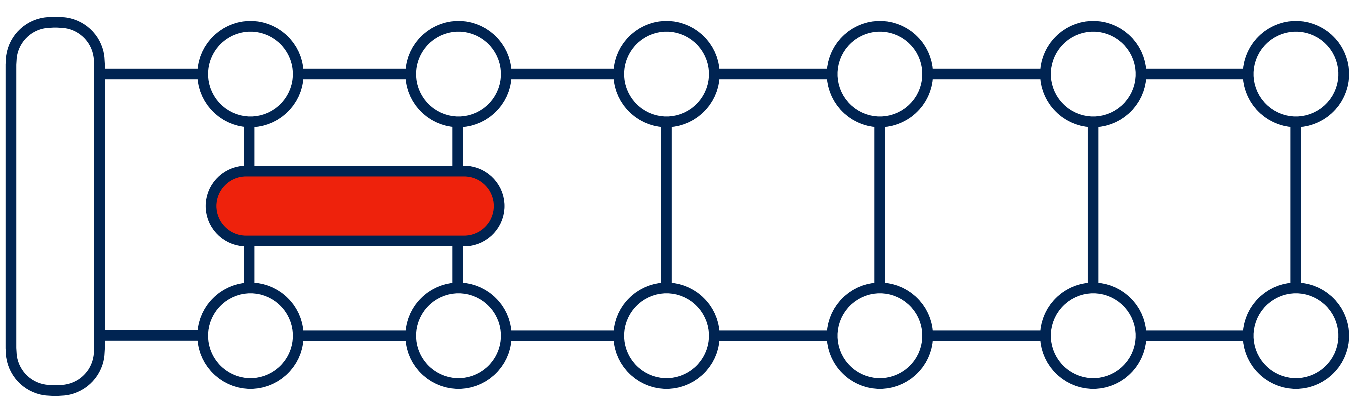


Another key computation is an **expectation value**

$$\langle \Psi | A | \Psi \rangle =$$

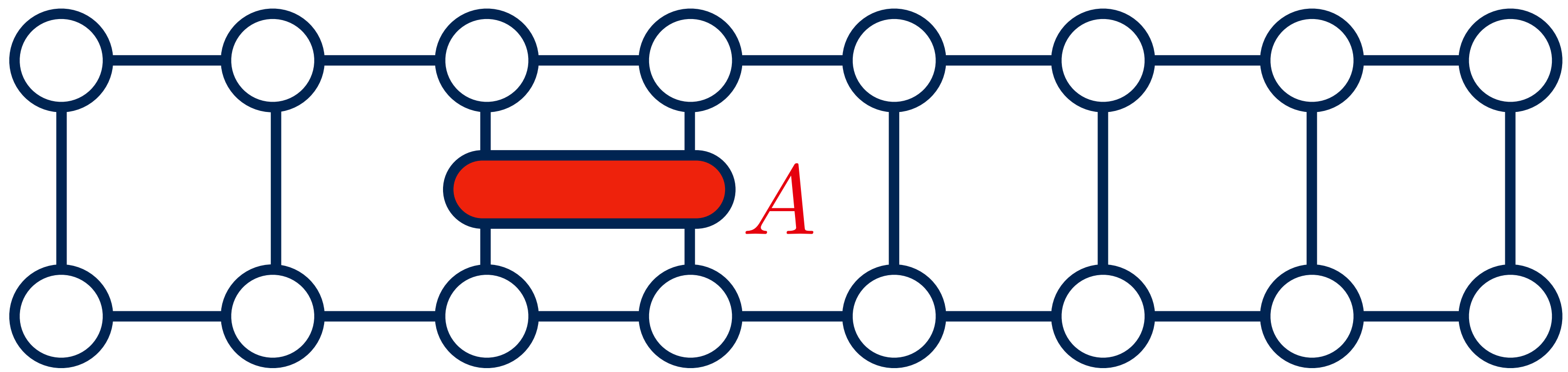


Using similar procedure as norm:

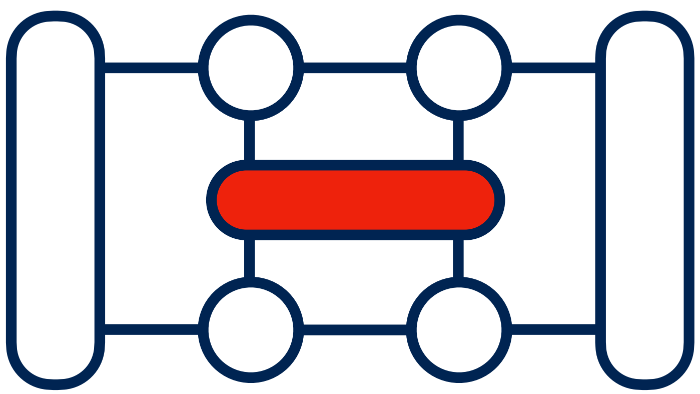


Another key computation is an **expectation value**

$$\langle \Psi | A | \Psi \rangle =$$

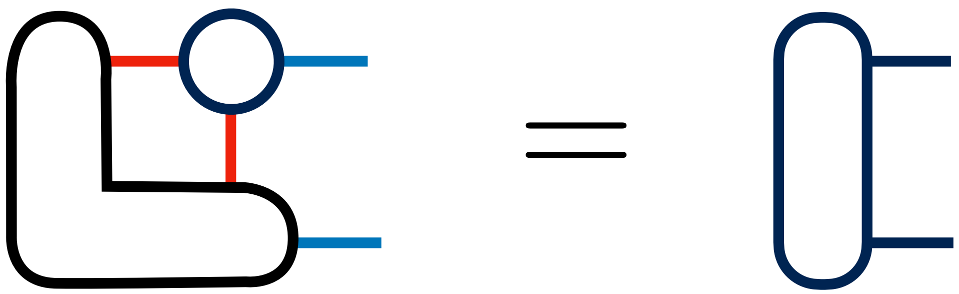


Using similar procedure as norm:



What is the **scaling** of the computational cost ?

To calculate, break computation into separate steps, such as:

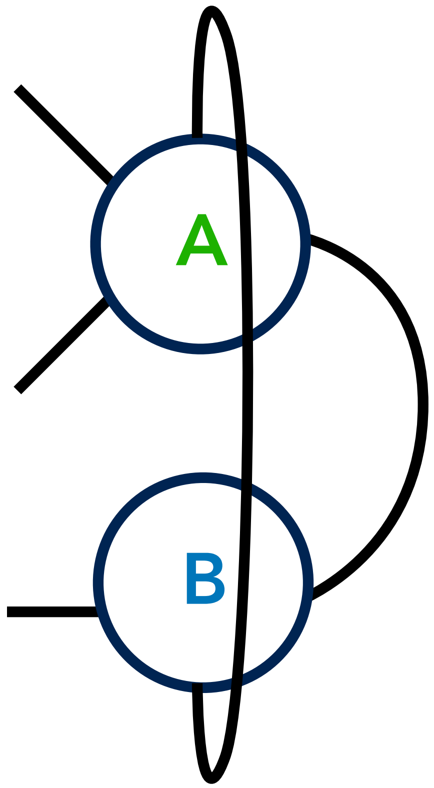


Then use rule for cost of tensor contraction:

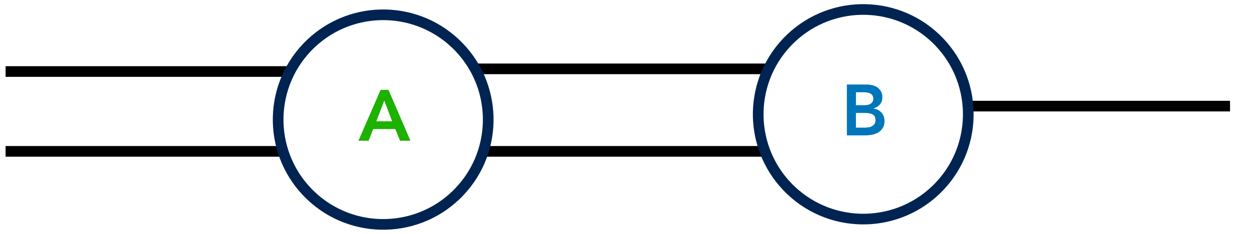
(**dimension of contracted indices**)

x (**dimension of uncontracted indices**)

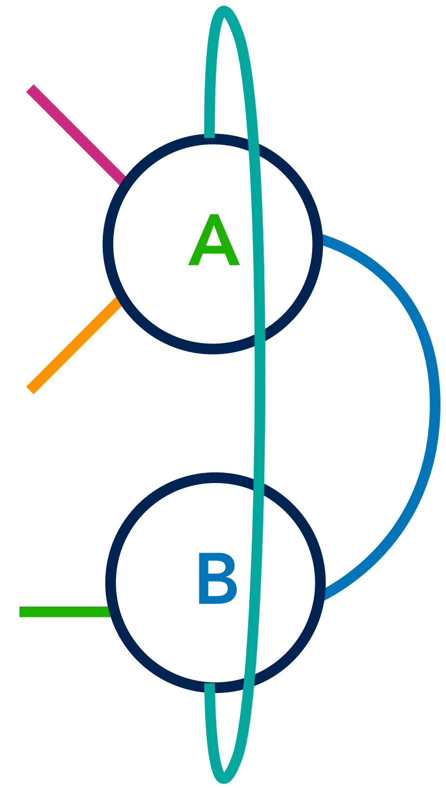
Complicated tensor contraction



Can always be written as matrix mult. with grouped indices

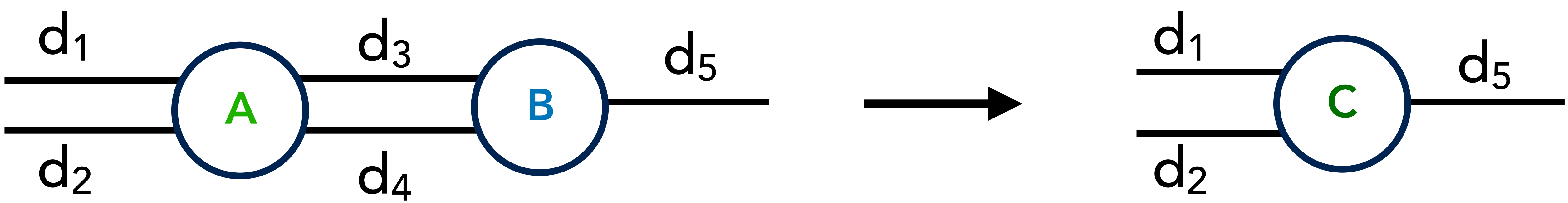


Complicated tensor contraction



Can always be written as matrix mult. with grouped indices

To compute scaling, consider index dimensions



Computation of C equivalent to:

```
for i₁=1:d₁, i₂=1:d₂, i₅=1:d₅
    for i₃=1:d₃, i₄=1:d₄
        C[i₁,i₂,i₅] = A[i₁,i₂,i₃,i₄]*B[i₃,i₄,i₅]
    end
end
```

Computation of C:

```
for i1=1:d1, i2=1:d2, i5=1:d5
    for i3=1:d3, i4=1:d4
        C[i1,i2,i5] = A[i1,i2,i3,i4]*B[i3,i4,i5]
    end
end
```

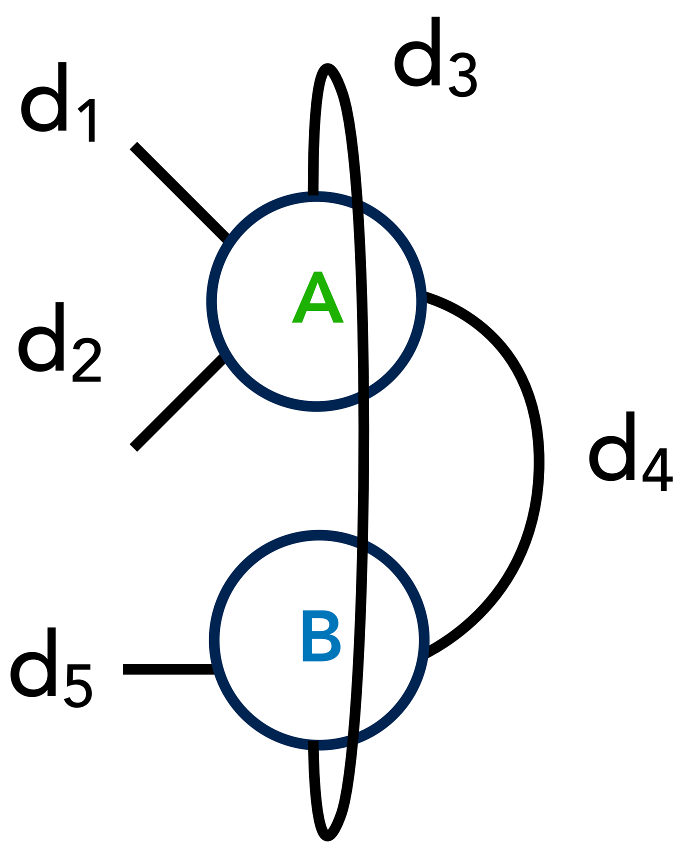
Each i_n loop takes d_n operations to complete

Overall scaling: $$(d_1 d_2 d_5) \times (d_3 d_4)$$

(dim. contracted indices) x (dim. uncontracted indices)

Summary of **scaling rule**

Whenever you see a tensor contraction



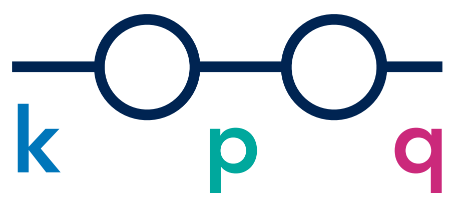
Multiply dimension of every index (just once)

$$\text{cost} = (d_1 \cdot d_2 \cdot d_3 \cdot d_4 \cdot d_5)$$

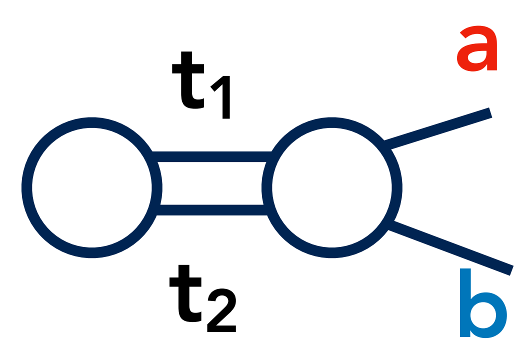
Test Your Knowledge! 🤔

Write down the cost of the following tensor contractions (letters are index dimensions):

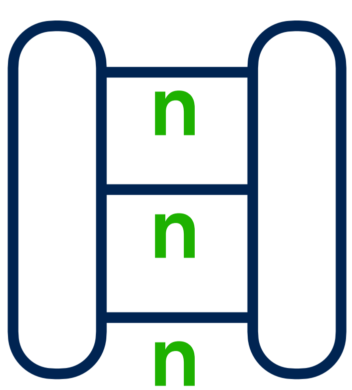
(1)



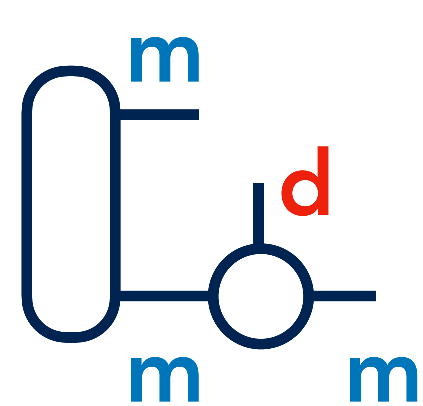
(2)



(3)



(4)

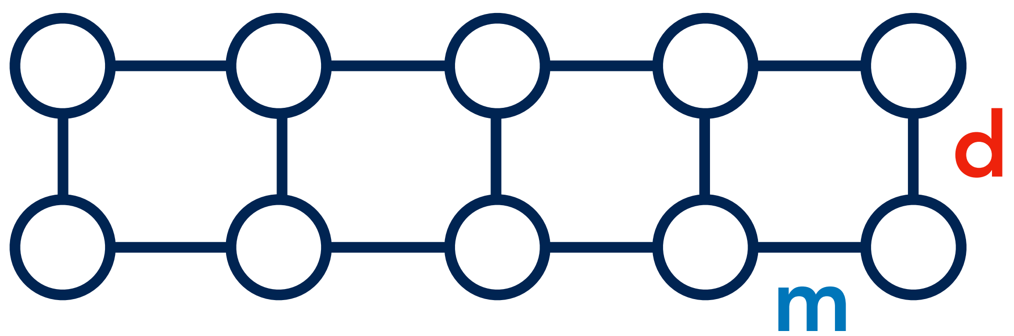


What is the **scaling** of calculations with MPS?

What is the **scaling** of calculations with MPS?

Consider norm of MPS bond dimension **m**, site dimension **d**

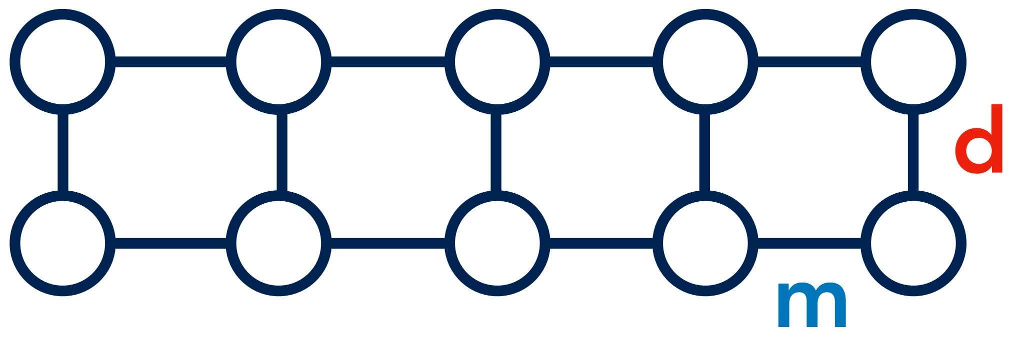
$$\langle \Psi | \Psi \rangle =$$



What is the **scaling** of calculations with MPS?

Consider norm of MPS bond dimension **m**, site dimension **d**

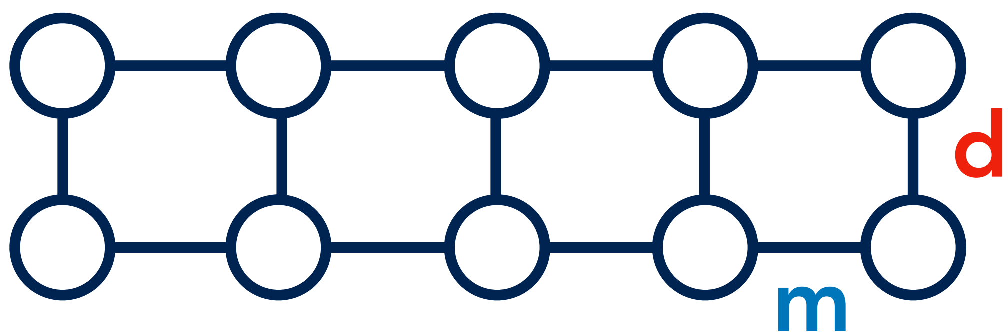
$$\langle \Psi | \Psi \rangle =$$



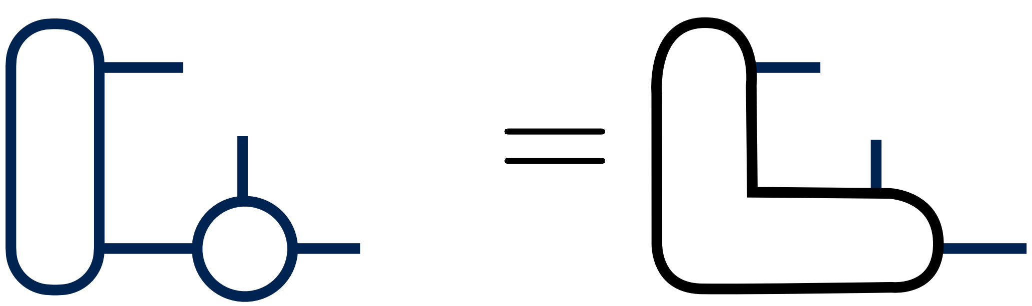
Two key operations

What is the **scaling** of calculations with MPS?

Consider norm of MPS bond dimension **m**, site dimension **d**

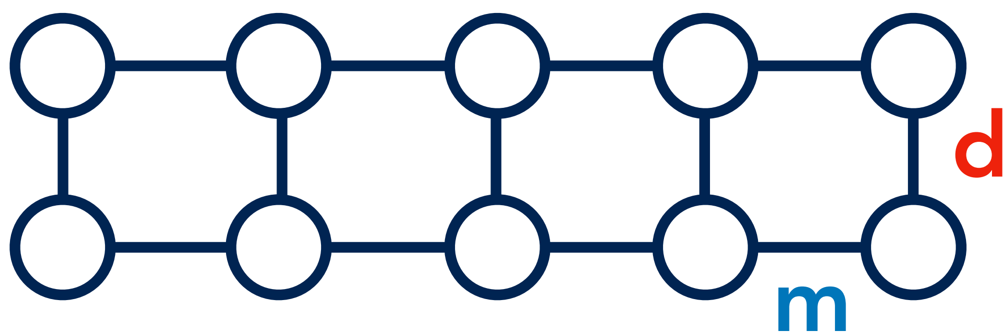
$\langle \Psi | \Psi \rangle =$ 

Two key operations

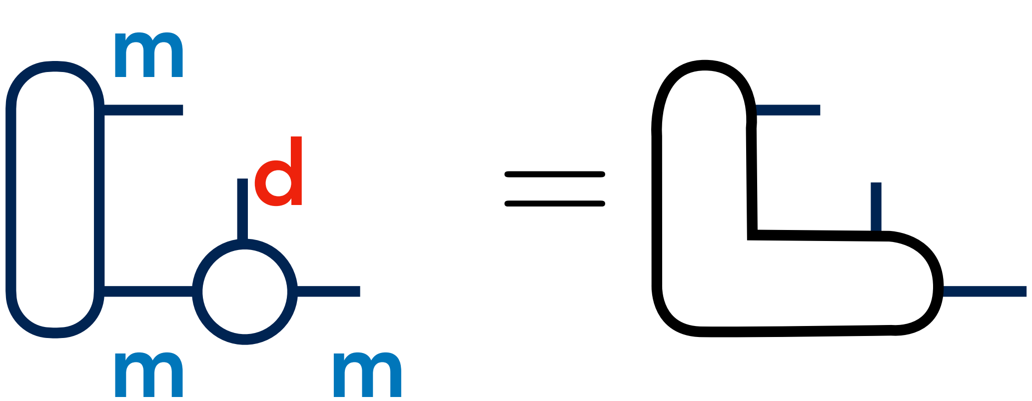
(1) 

What is the **scaling** of calculations with MPS?

Consider norm of MPS bond dimension m, site dimension d

$\langle \Psi | \Psi \rangle =$ 

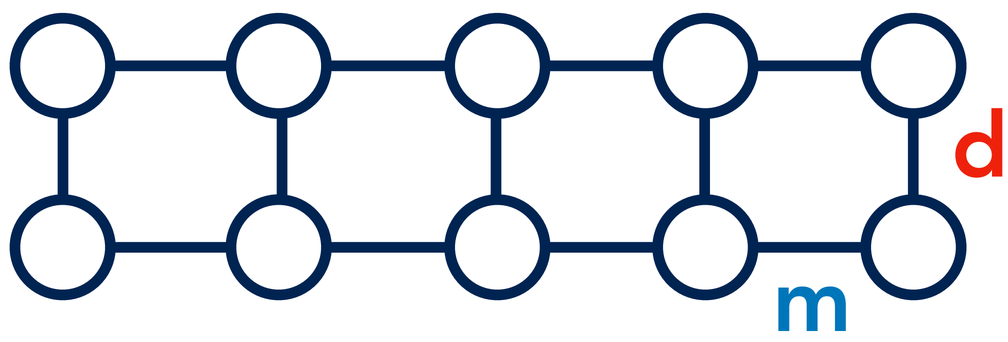
Two key operations

(1) 

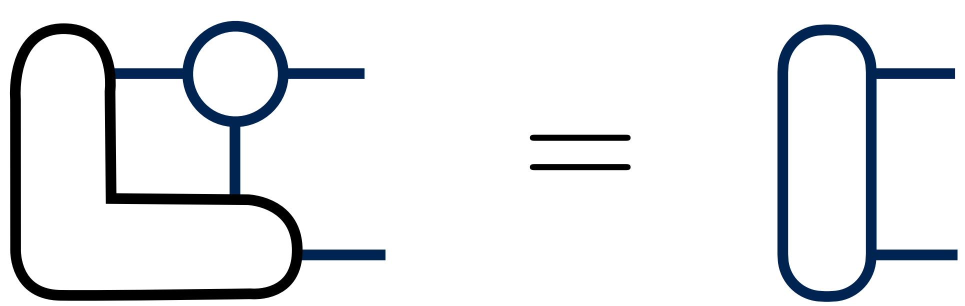
scaling is $m^3 d$

What is the **scaling** of the computational cost ?

Consider norm calculation, MPS bond dimension **m**,
site dimension **d**

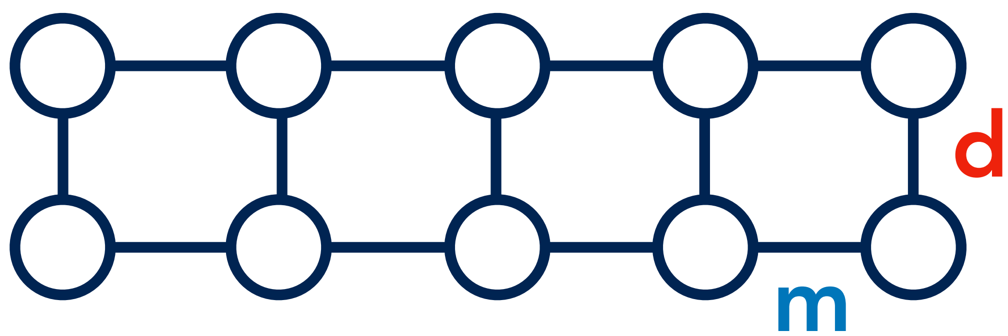
$\langle \Psi | \Psi \rangle =$ 

Two key operations

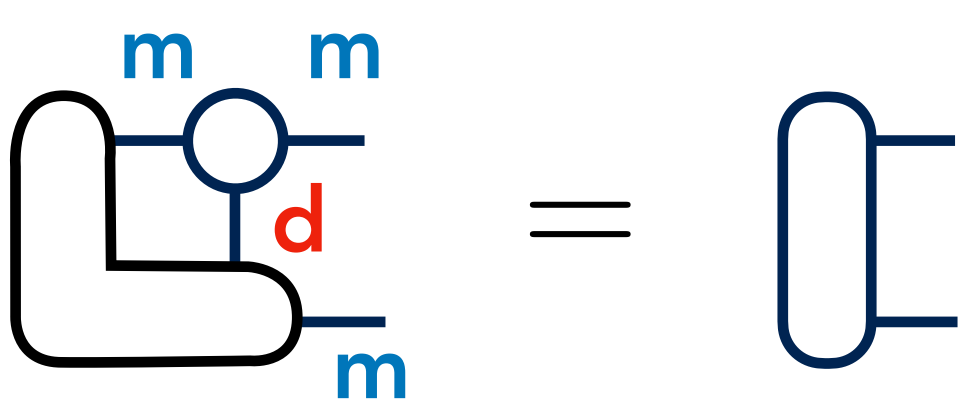
(2) 

What is the **scaling** of the computational cost ?

Consider norm calculation, MPS bond dimension **m**, site dimension **d**

$\langle \Psi | \Psi \rangle =$ 

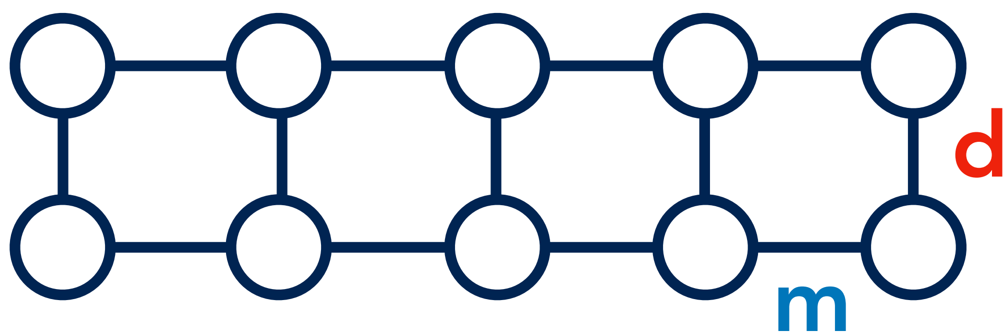
Two key operations

(2) 

scaling is $m^3 d$

What is the **scaling** of the computational cost ?

Consider norm calculation, MPS bond dimension **m**, site dimension **d**

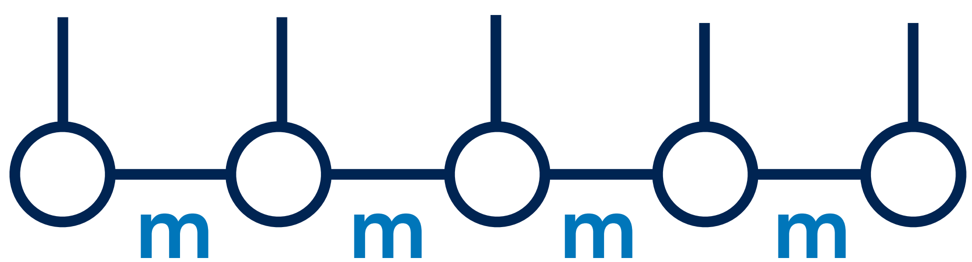
$$\langle\Psi|\Psi\rangle =$$ 

So overall scaling of norm calculation is

$$m^3\, d$$

Rule of thumb: most every operation needed to manipulate MPS can be made to scale as

$$m^3$$



Intuition: MPS involves multiplying mxm matrices

Scaling of mxm matrix multiplication is m^3

Examples of Matrix Product States

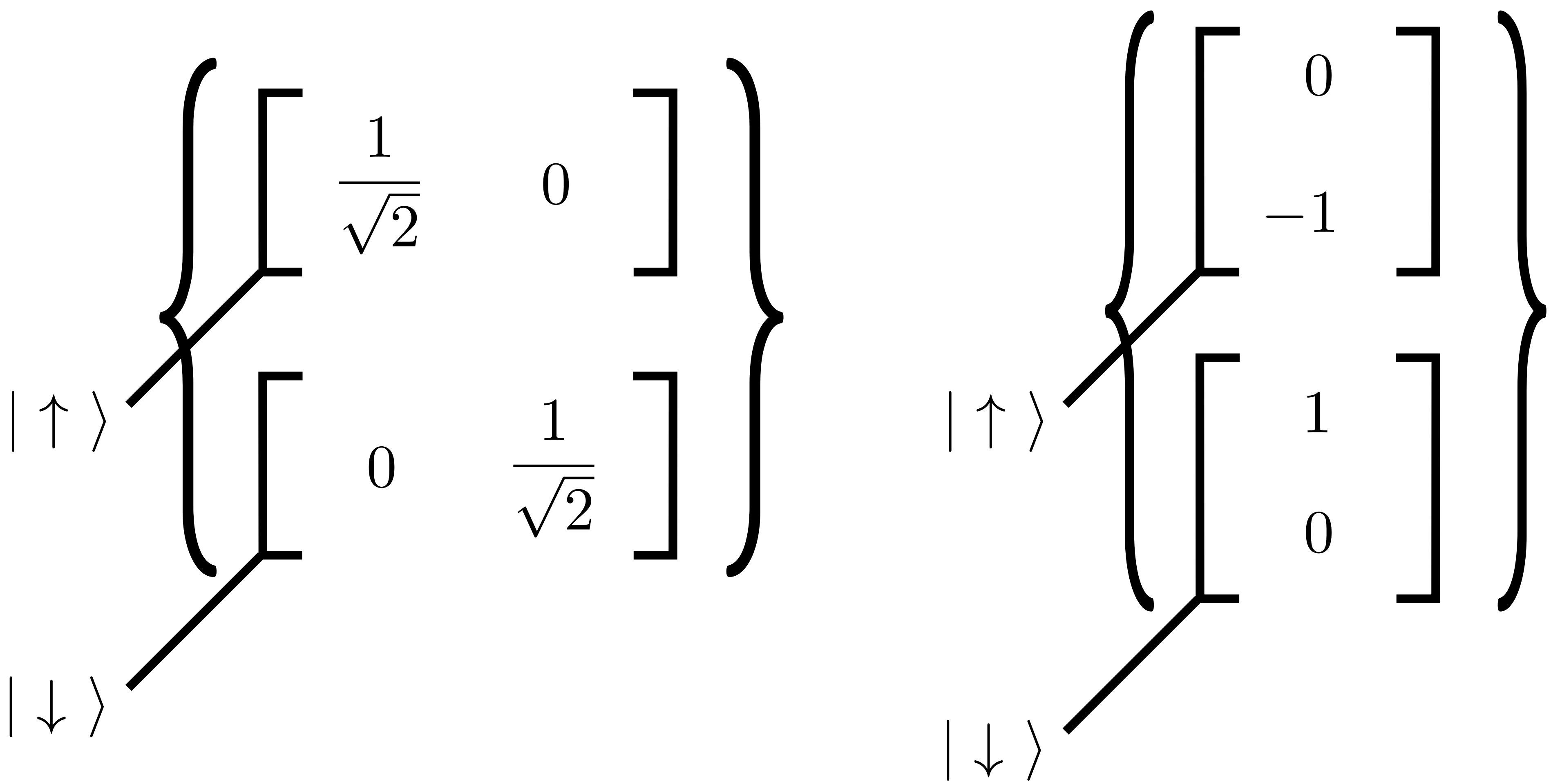
Example #1: singlet state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle|\downarrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle|\uparrow\rangle$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}}|\uparrow\rangle & \frac{1}{\sqrt{2}}|\downarrow\rangle \end{bmatrix} \begin{bmatrix} |\downarrow\rangle \\ -|\uparrow\rangle \end{bmatrix}$$

How to see this is an MPS?

$$\begin{bmatrix} \frac{1}{\sqrt{2}}|\uparrow\rangle & \frac{1}{\sqrt{2}}|\downarrow\rangle \end{bmatrix} \begin{bmatrix} |\downarrow\rangle \\ -|\uparrow\rangle \end{bmatrix} =$$



Example #2: AKLT wavefunction

The AKLT wavefunction is the exact ground state of the following S=1 Hamiltonian

$$H = \sum_j \mathbf{S}_j \cdot \mathbf{S}_{j+1} + \frac{1}{3} \sum_j (\mathbf{S}_j \cdot \mathbf{S}_{j+1})^2$$

In the same phase as S=1 Heisenberg model, plus 'small' perturbation of $(\mathbf{S}\cdot\mathbf{S})^2$ biquadratic term

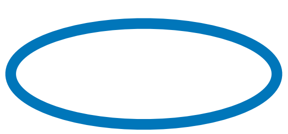
Can construct AKLT wavefunction as follows

Start with 2N spin 1/2's in singlet pairs

 $= \frac{1}{\sqrt{2}}|\uparrow\rangle|\downarrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle|\uparrow\rangle$

Can construct AKLT wavefunction as follows

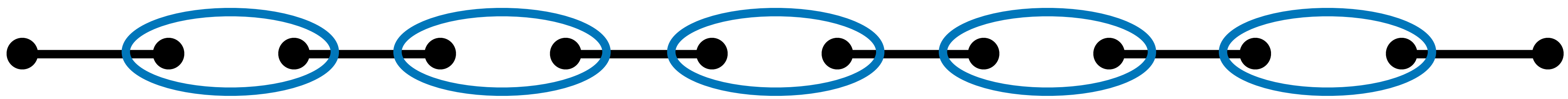
Act on pairs of S=1/2's with projection operator P

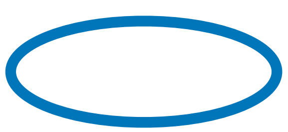


$$= \hat{P} = |+\rangle\langle\uparrow\uparrow| + |0\rangle\frac{\langle\uparrow\downarrow| + \langle\downarrow\uparrow|}{\sqrt{2}} + |-\rangle\langle\downarrow\downarrow|$$

Can construct AKLT wavefunction as follows

Act on pairs of S=1/2's with projection operator P

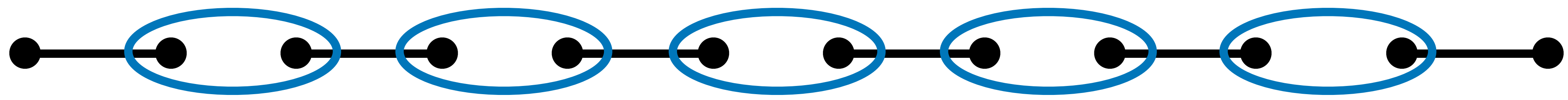




$$= \hat{P} = |+\rangle\langle\uparrow\uparrow| + |0\rangle\frac{\langle\uparrow\downarrow| + \langle\downarrow\uparrow|}{\sqrt{2}} + |-\rangle\langle\downarrow\downarrow|$$

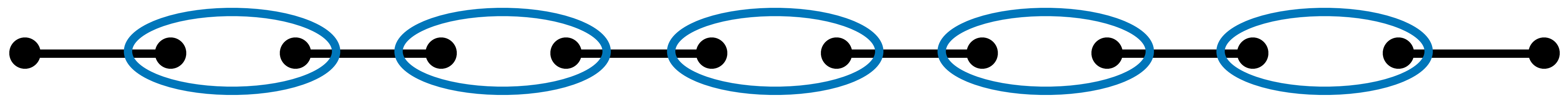
Can construct AKLT wavefunction as follows

After projection, blue ovals are S=1 spins



Can construct AKLT wavefunction as follows

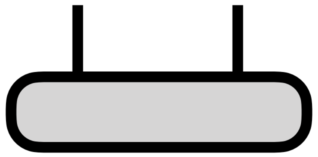
After projection, blue ovals are S=1 spins



Can predict interesting properties:

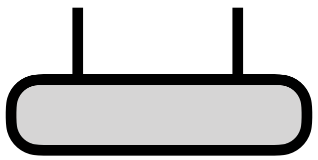
- doubly degenerate entanglement spectrum
- emergent S=1/2 edge spins

Tensor approach to AKLT

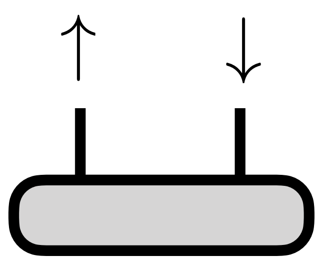


$$= \frac{1}{\sqrt{2}}|\uparrow\rangle|\downarrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle|\uparrow\rangle$$

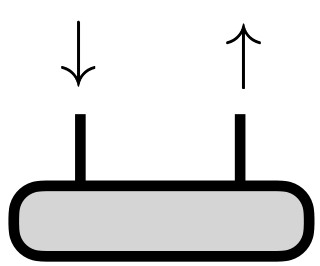
Tensor approach to AKLT



$$= \frac{1}{\sqrt{2}} |\uparrow\rangle |\downarrow\rangle - \frac{1}{\sqrt{2}} |\downarrow\rangle |\uparrow\rangle$$

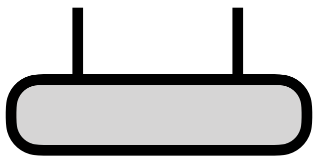


$$= \frac{1}{\sqrt{2}}$$

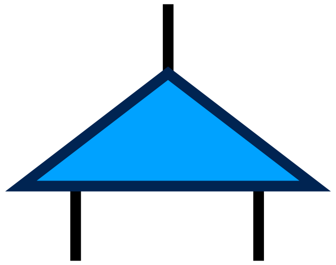


$$= -\frac{1}{\sqrt{2}}$$

Tensor approach to AKLT

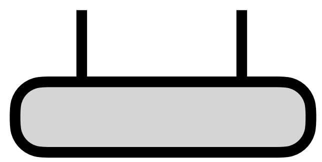


$$= \frac{1}{\sqrt{2}}|\uparrow\rangle|\downarrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle|\uparrow\rangle$$

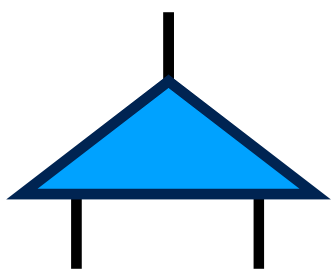


$$= |+\rangle\langle\uparrow\uparrow| + |0\rangle\frac{\langle\uparrow\downarrow| + \langle\downarrow\uparrow|}{\sqrt{2}} + |-\rangle\langle\downarrow\downarrow|$$

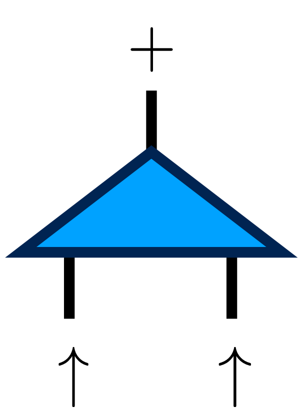
Tensor approach to AKLT



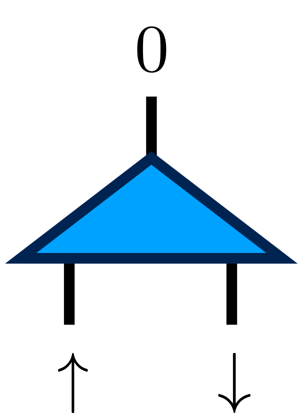
$$= \frac{1}{\sqrt{2}}|\uparrow\rangle|\downarrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle|\uparrow\rangle$$



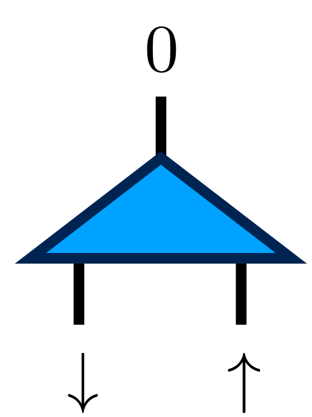
$$= |+\rangle\langle\uparrow\uparrow| + |0\rangle\frac{\langle\uparrow\downarrow| + \langle\downarrow\uparrow|}{\sqrt{2}} + |-\rangle\langle\downarrow\downarrow|$$



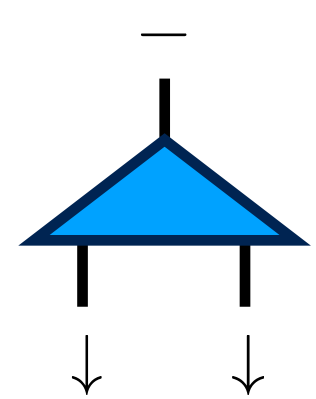
$= 1$



$= \frac{1}{\sqrt{2}}$

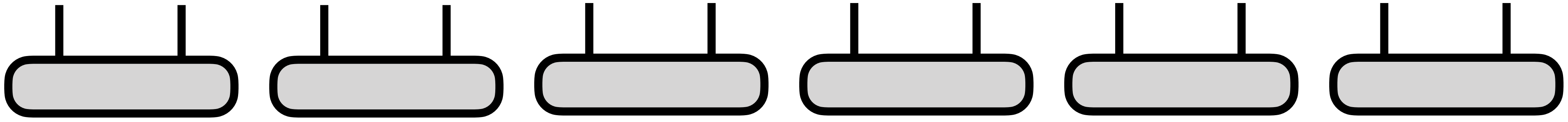


$= \frac{1}{\sqrt{2}}$

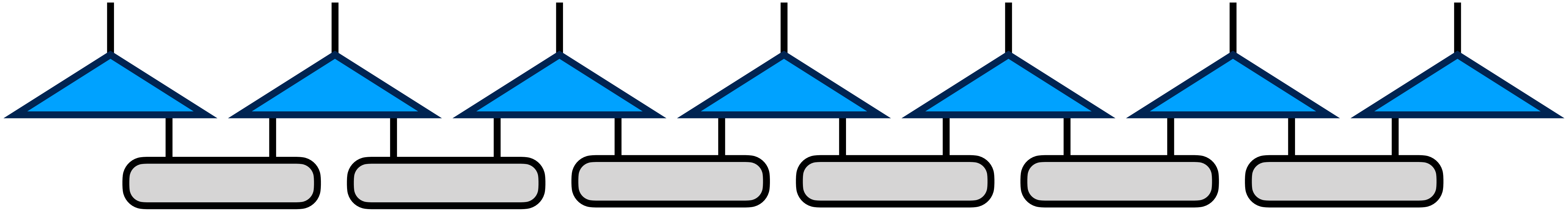


$= 1$

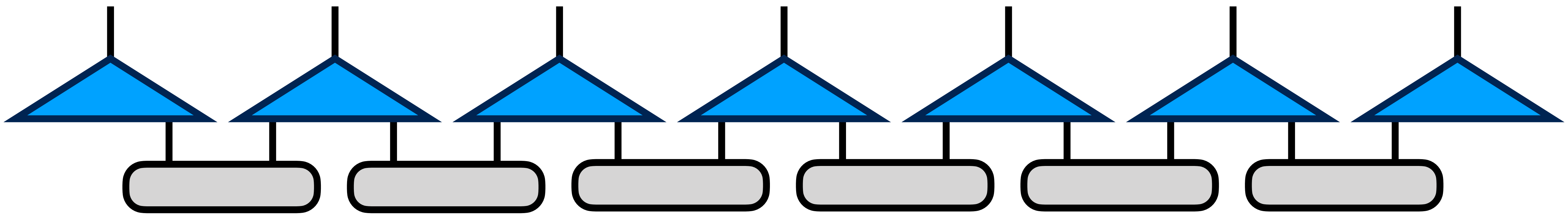
Tensor approach to AKLT



Tensor approach to AKLT

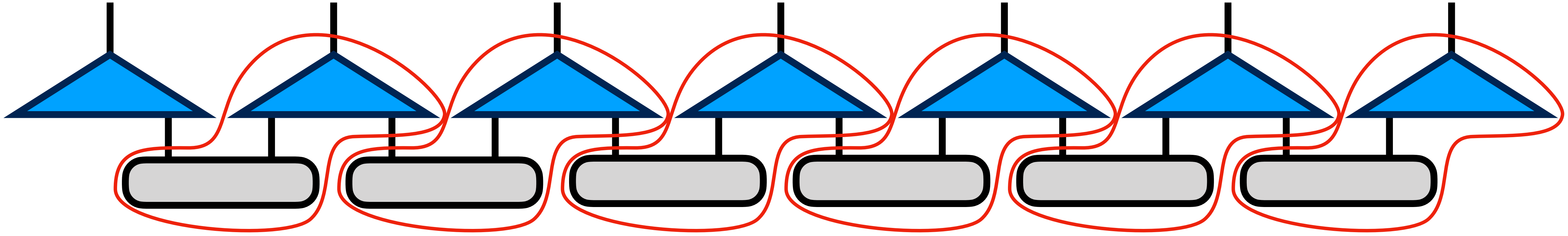


Put into MPS form



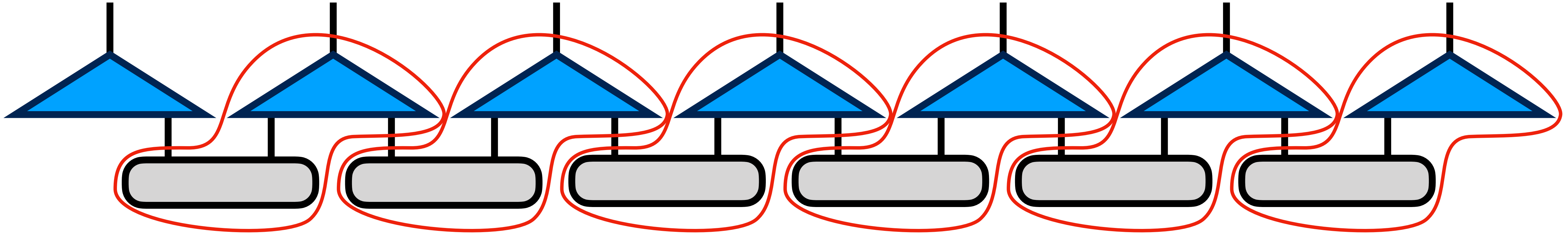
Put into MPS form

Contract pairs of tensors:

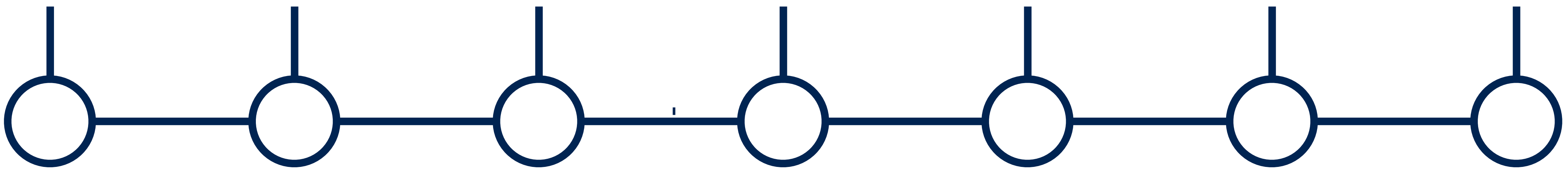


Put into MPS form

Contract pairs of tensors:



=



Nice form of AKLT matrix product state with periodic boundary conditions

Can actually show the following:

$$|\Psi_{\text{AKLT}}\rangle = \text{Tr}\left[M^{s_1} M^{s_2} M^{s_3} \cdots M^{s_N}\right] |s_1 s_2 s_3 \ldots s_N\rangle$$

where

$$M^{+} = \sqrt{\frac{2}{3}}\;\sigma^{+}$$

$$M^{0} = -\sqrt{\frac{1}{3}}\;\sigma^{z}$$

$$M^{-} = -\sqrt{\frac{2}{3}}\;\sigma^{-}$$

Takeaway

- MPS guaranteed to obey boundary law, as do all 1D ground states (of gapped, local Hamiltonians)
- MPS can capture certain interesting states exactly
- maybe they are a useful class of wavefunction to optimize!