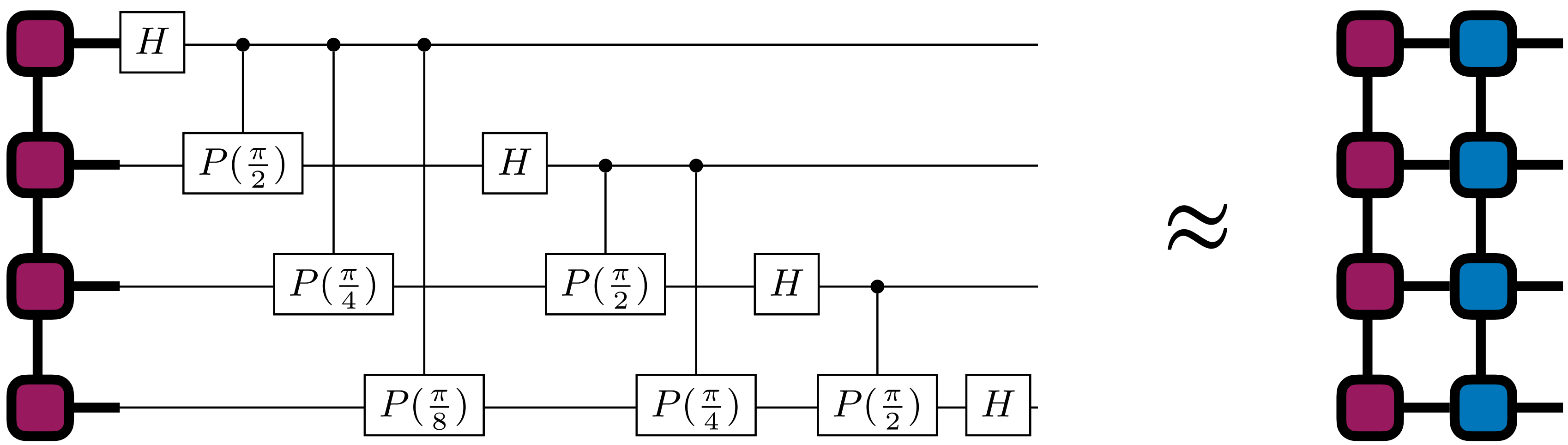
Time Evolution with Tensor Networks



Tensor Network Algorithms

An important capability is
time evolving tensor networks

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

Algorithms:

- Time evolving block decimation (TEBD)
- Time-dependent variational principle (TDVP)

Tensor Network Algorithms

TEBD algorithm $$H = \sum_j h_{j,j+1}$$

Tensor Network Algorithms

TEBD algorithm

$$H = \sum_j h_{j,j+1}$$

$$U = e^{-\mathrm{i}Ht} = \prod_{k=1}^{n} e^{-\mathrm{i}H\delta t} \qquad n\delta t = t$$

Tensor Network Algorithms

TEBD algorithm $\quad H = \sum_j h_{j,j+1}$

$$U = e^{-\mathrm{i}Ht} = \prod_{k=1}^{n} e^{-\mathrm{i}H\delta t} \qquad n\delta t = t$$

Want to decompose $e^{-\mathrm{i}H\delta t}$ to keep Trotter Error low in a given time step. Use Suzuki-Trotter

Tensor Network Algorithms

TEBD algorithm

$$H = \sum_j h_{j,j+1}$$

$$U = e^{-\mathrm{i}Ht} = \prod_{k=1}^{n} e^{-\mathrm{i}H\delta t} \qquad n\delta t = t$$

Want to decompose $e^{-\mathrm{i}H\delta t}$ to keep Trotter Error low in a given time step. Use Suzuki-Trotter

1st Order: $e^{-\mathrm{i}H\delta t} = \prod_i e^{-\mathrm{i}h_{i,i+1}\delta t} + \mathcal{O}(\delta t^2)$

Tensor Network Algorithms

TEBD algorithm

$$H = \sum_j h_{j,j+1}$$

$$U = e^{-\mathrm{i}Ht} = \prod_{k=1}^{n} e^{-\mathrm{i}H\delta t} \qquad n\delta t = t$$

Want to decompose $e^{-\mathrm{i}H\delta t}$ to keep Trotter Error low in a given time step. Use Suzuki-Trotter

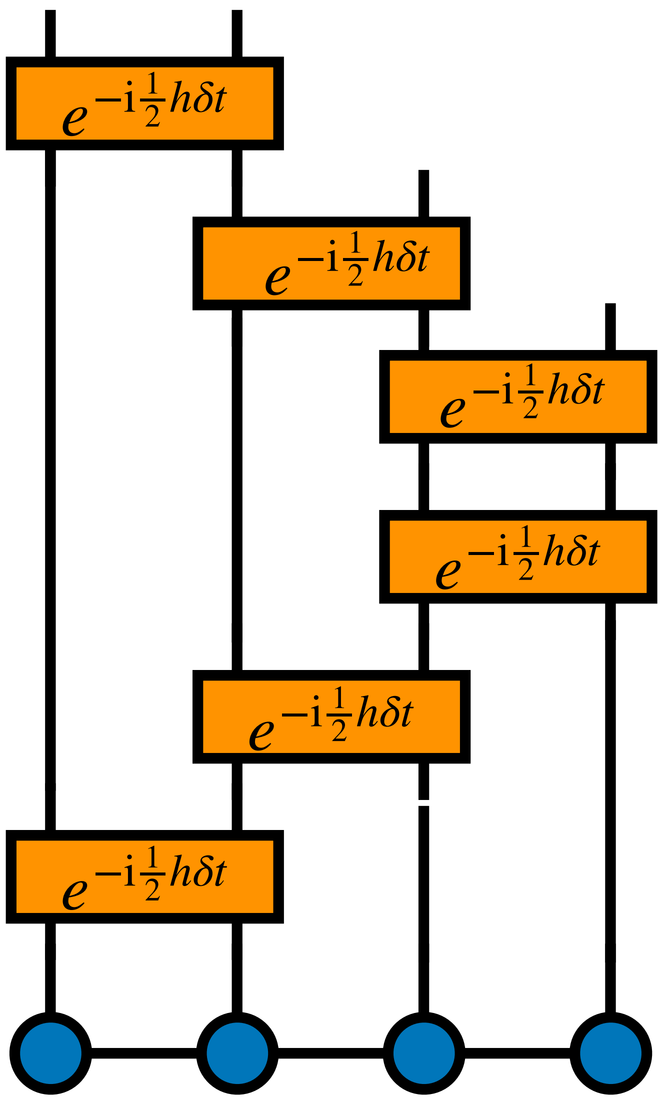
1st Order: $e^{-\mathrm{i}H\delta t} = \prod_i e^{-\mathrm{i}h_{i,i+1}\delta t} + \mathcal{O}(\delta t^2)$

2nd Order: $e^{-\mathrm{i}H\delta t} = \left(\prod_{1 \leq i \leq N} e^{-\frac{1}{2}\mathrm{i}h_{i,i+1}\delta t} \right) \left(\prod_{N \geq i \geq 1} e^{-\frac{1}{2}\mathrm{i}h_{i,i+1}\delta t} \right) + \mathcal{O}(\delta t^3)$

Tensor Network Algorithms

TEBD algorithm

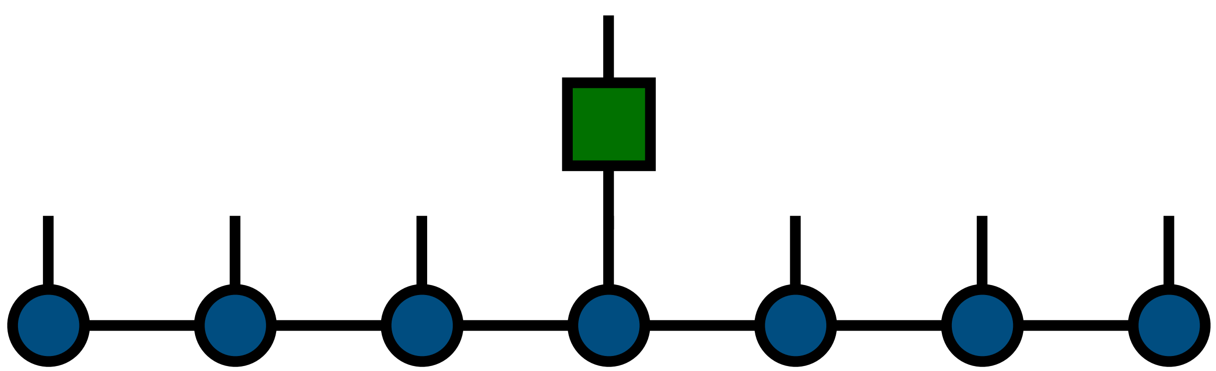
$$e^{-\mathrm{i}H\delta t} = \left(\prod_{1 \leq i \leq N} e^{-\frac{1}{2}\mathrm{i}h_{i,i+1}\delta t} \right) \left(\prod_{N \geq i \geq 1} e^{-\frac{1}{2}\mathrm{i}h_{i,i+1}\delta t} \right) + \mathcal{O}(\delta t^3)$$



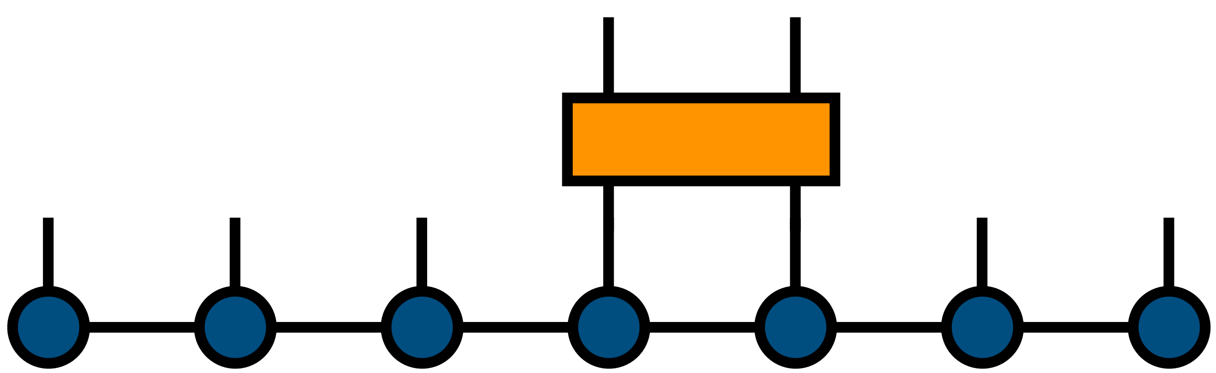
$$\approx e^{-\mathrm{i}H\delta t} \,|\, \psi(t)\rangle = |\, \psi(t + \delta t)\rangle$$

Tensor Network Algorithms

TEBD = controlled application of one-qubit and two-qubit gates to an MPS



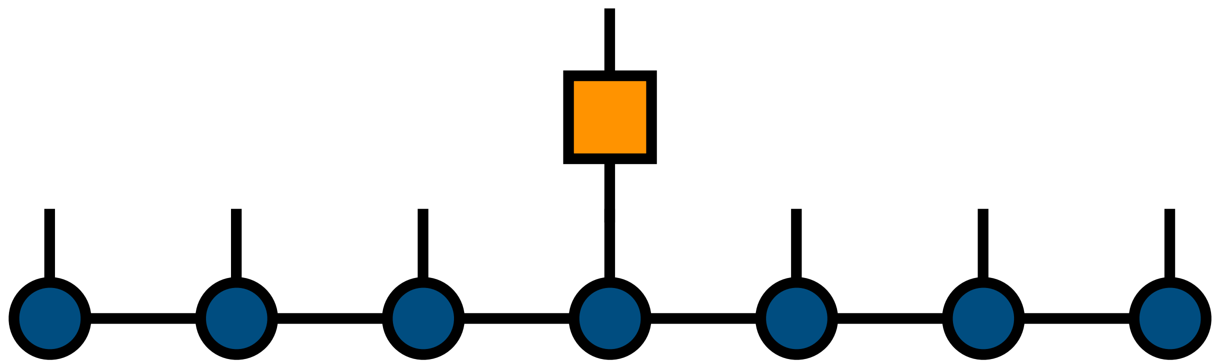
one-qubit gate is exact



two-qubit gate incurs small *error* controlled by bond-dimension χ

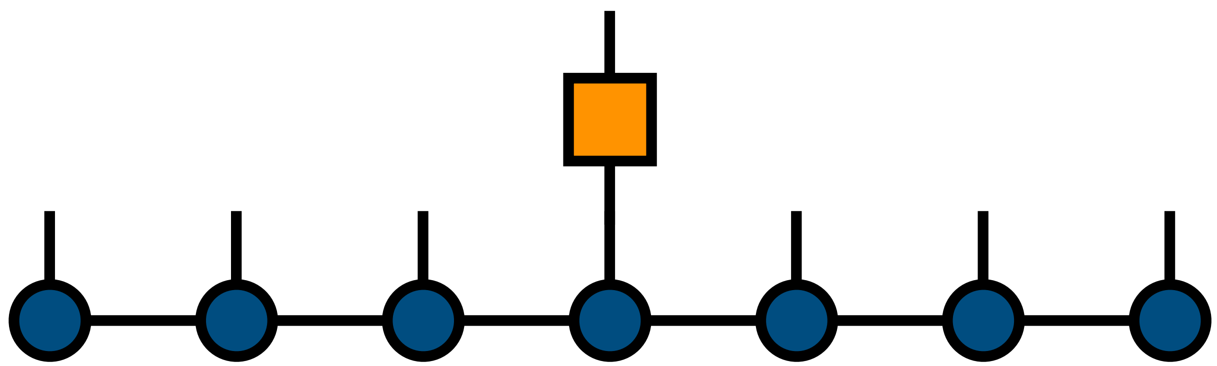
Tensor Network Algorithms

Say we want to act a <u>single-qubit gate</u> on a wavefunction in MPS form

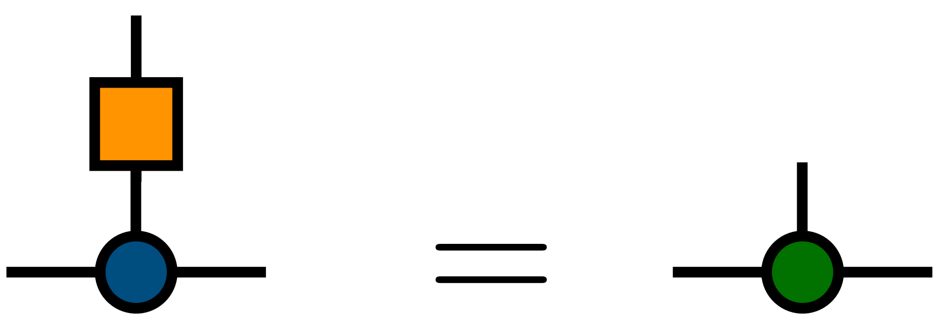


Tensor Network Algorithms

Say we want to act a single-qubit gate on a wavefunction in MPS form

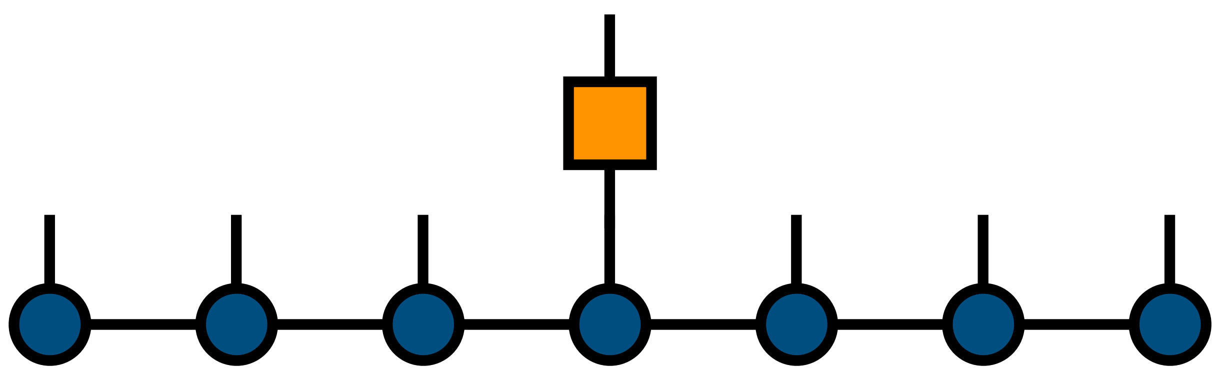


Just operate on one tensor:

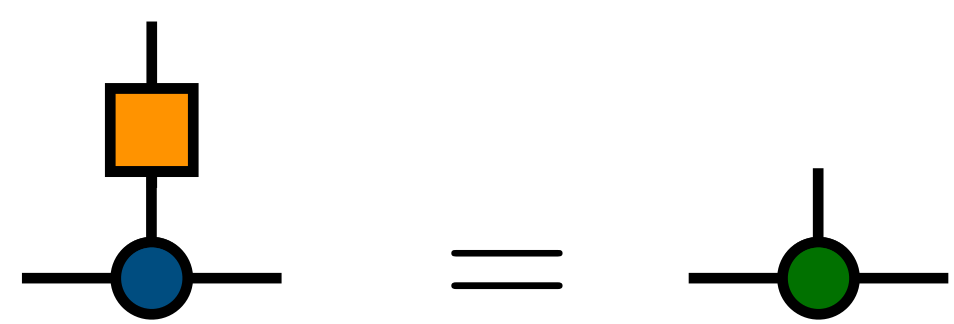


Tensor Network Algorithms

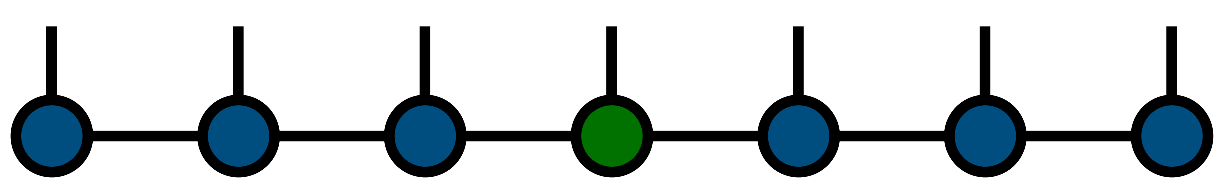
Say we want to act a single-qubit gate on a wavefunction in MPS form



Just operate on one tensor:

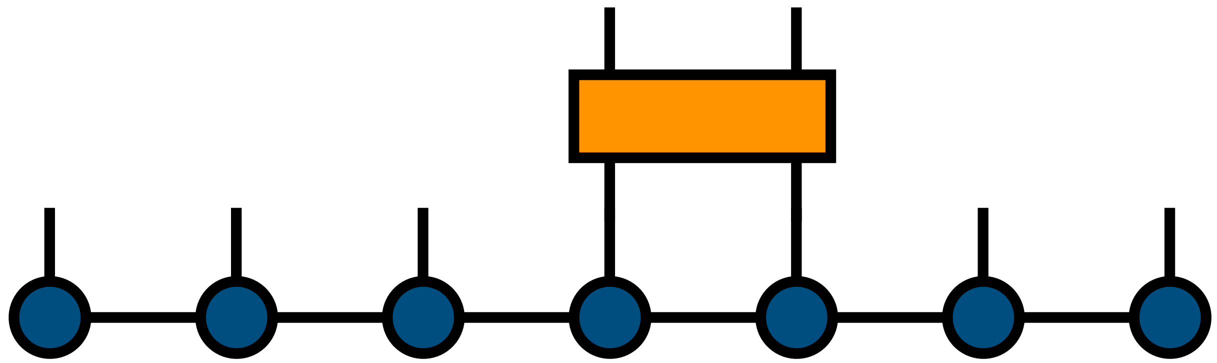


Result:

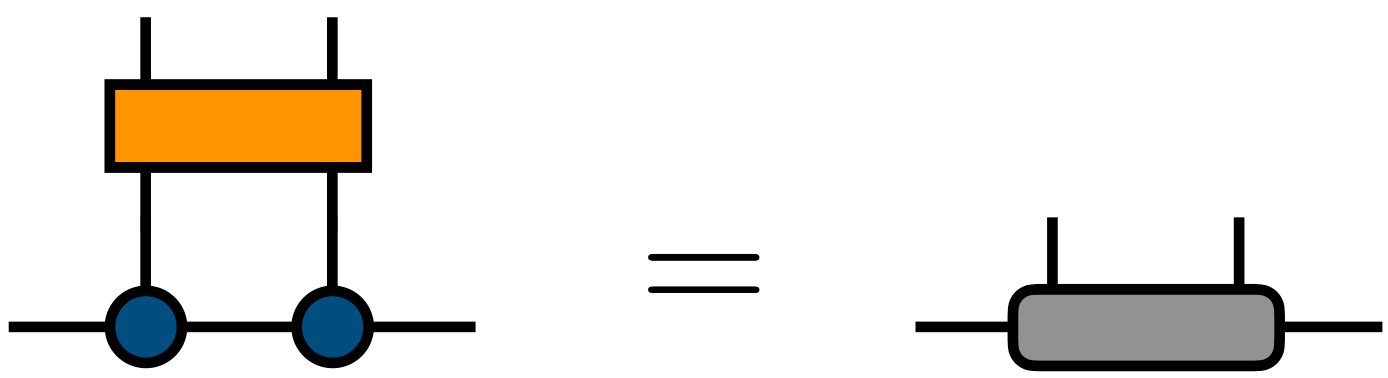


same bond dimension

Say we want to act <u>two-qubit gate</u> on a wavefunction in MPS form

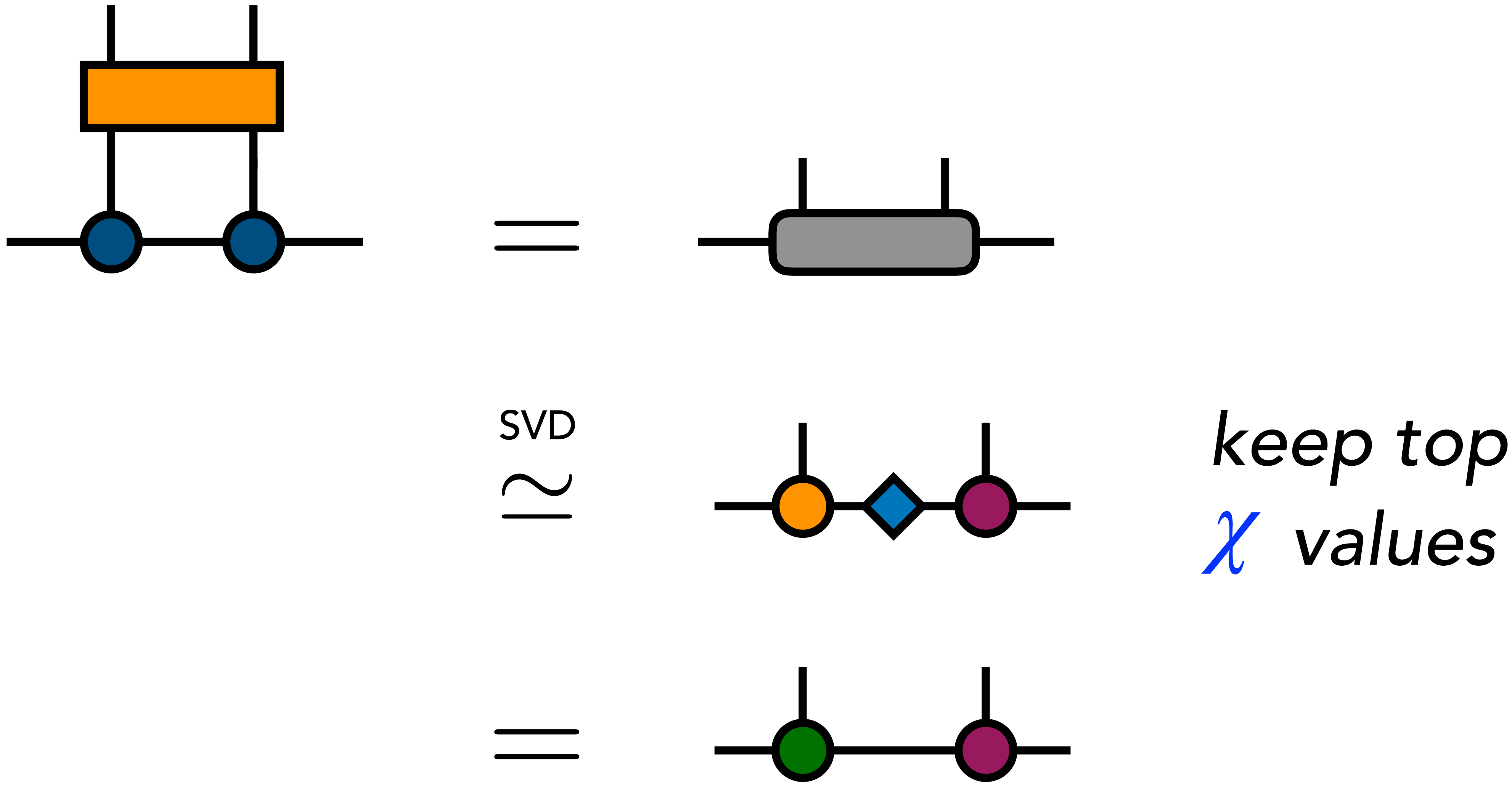


Operate on two tensors:

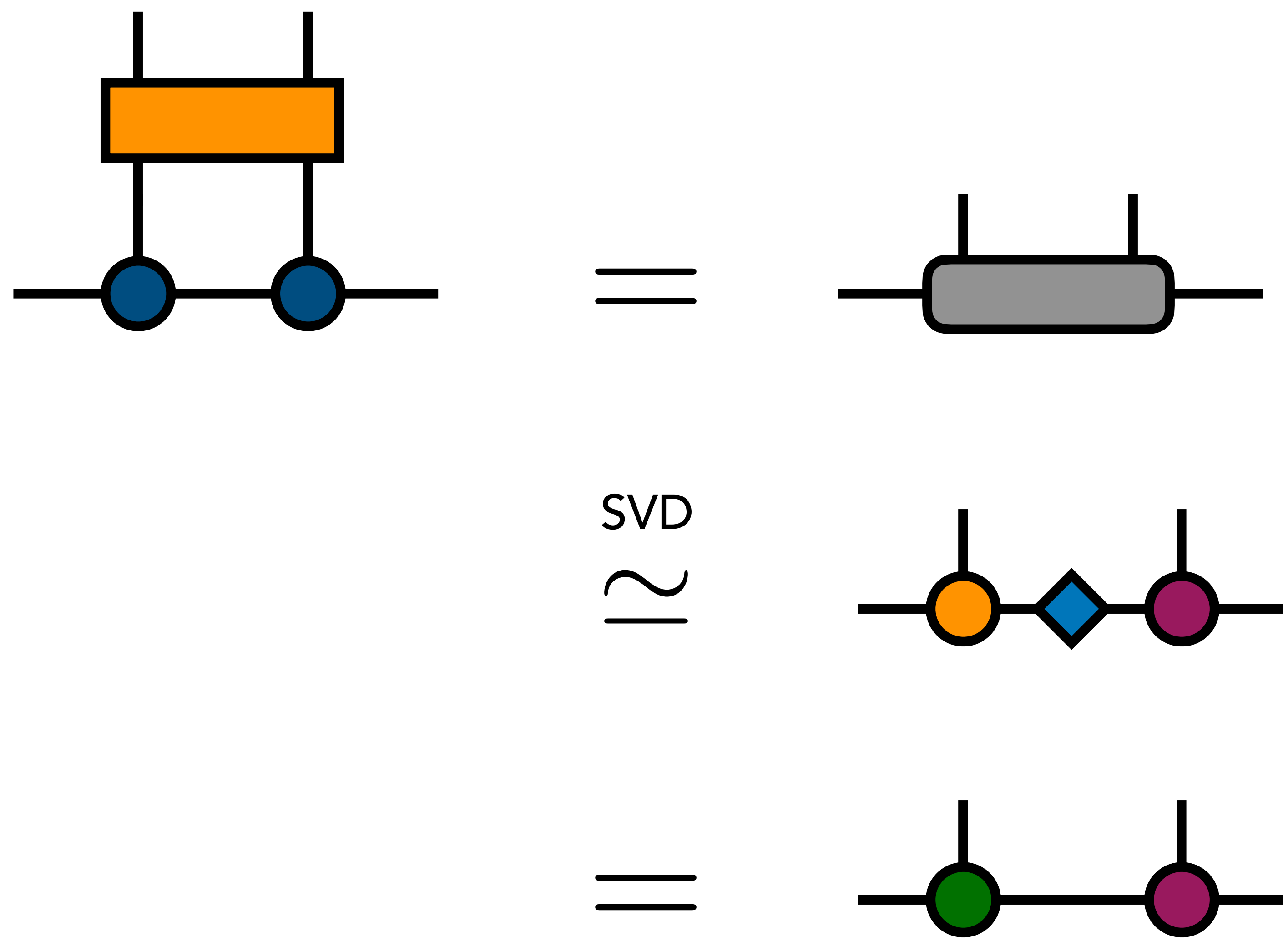


destroy MPS form locally

But can recover MPS form using truncated SVD:

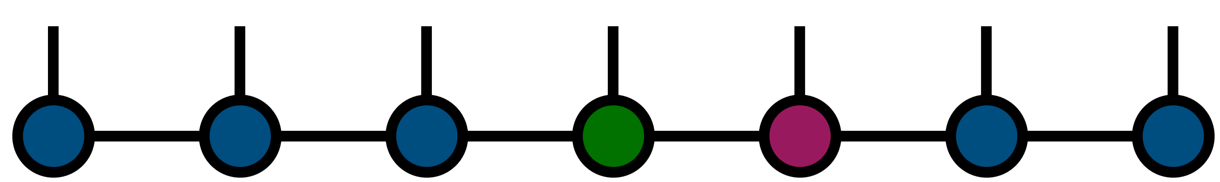


But can recover MPS form using truncated SVD:



keep top χ *values*

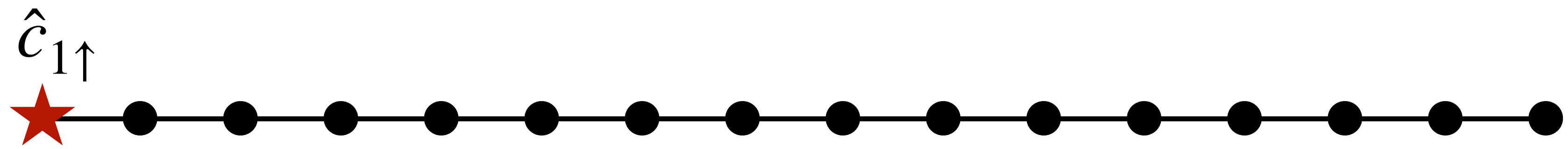
Result:

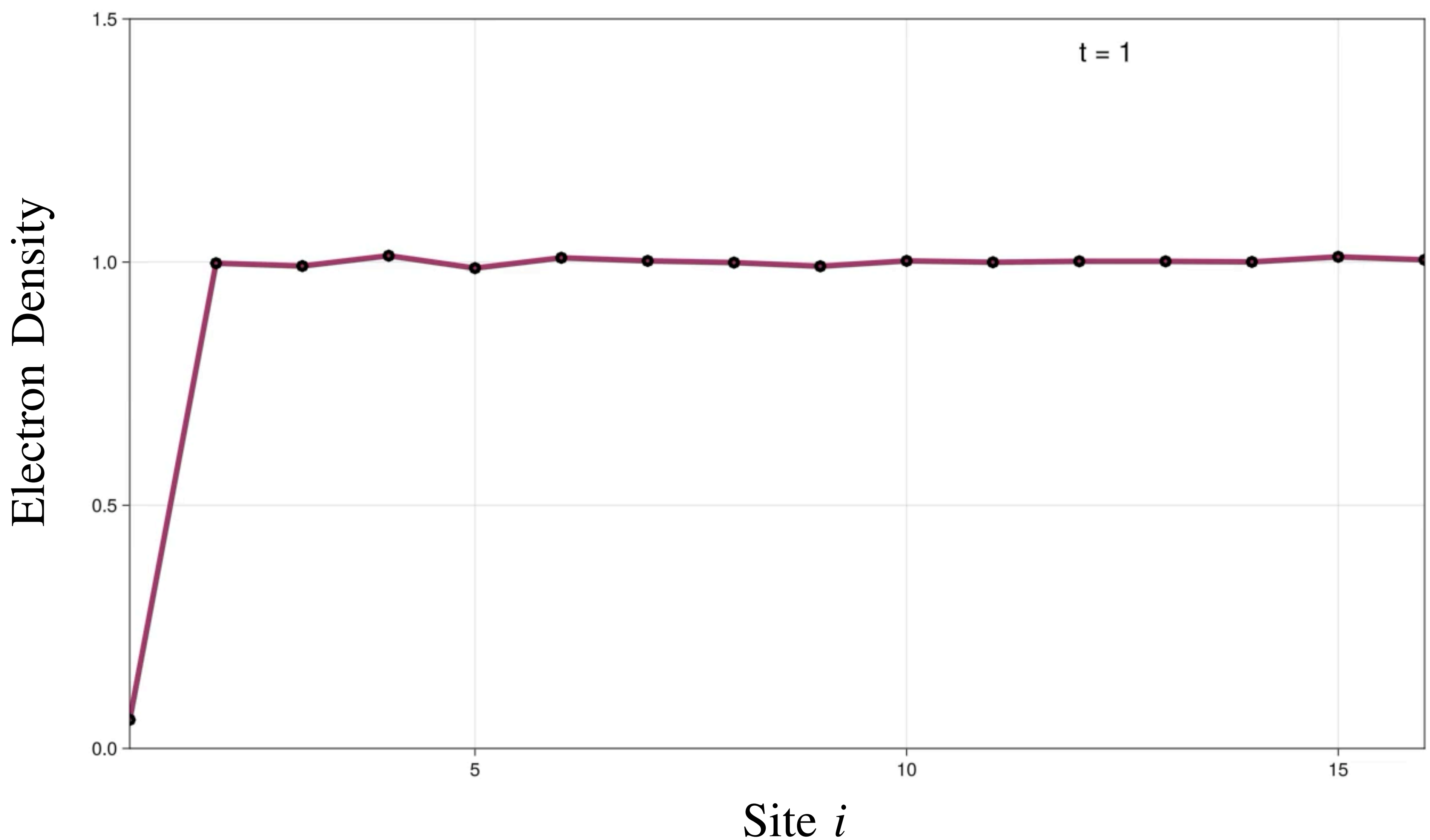


with loss of fidelity dependent on level of truncation

Tensor Network Algorithms

TEBD results: electron chain* after removing particle on site 1

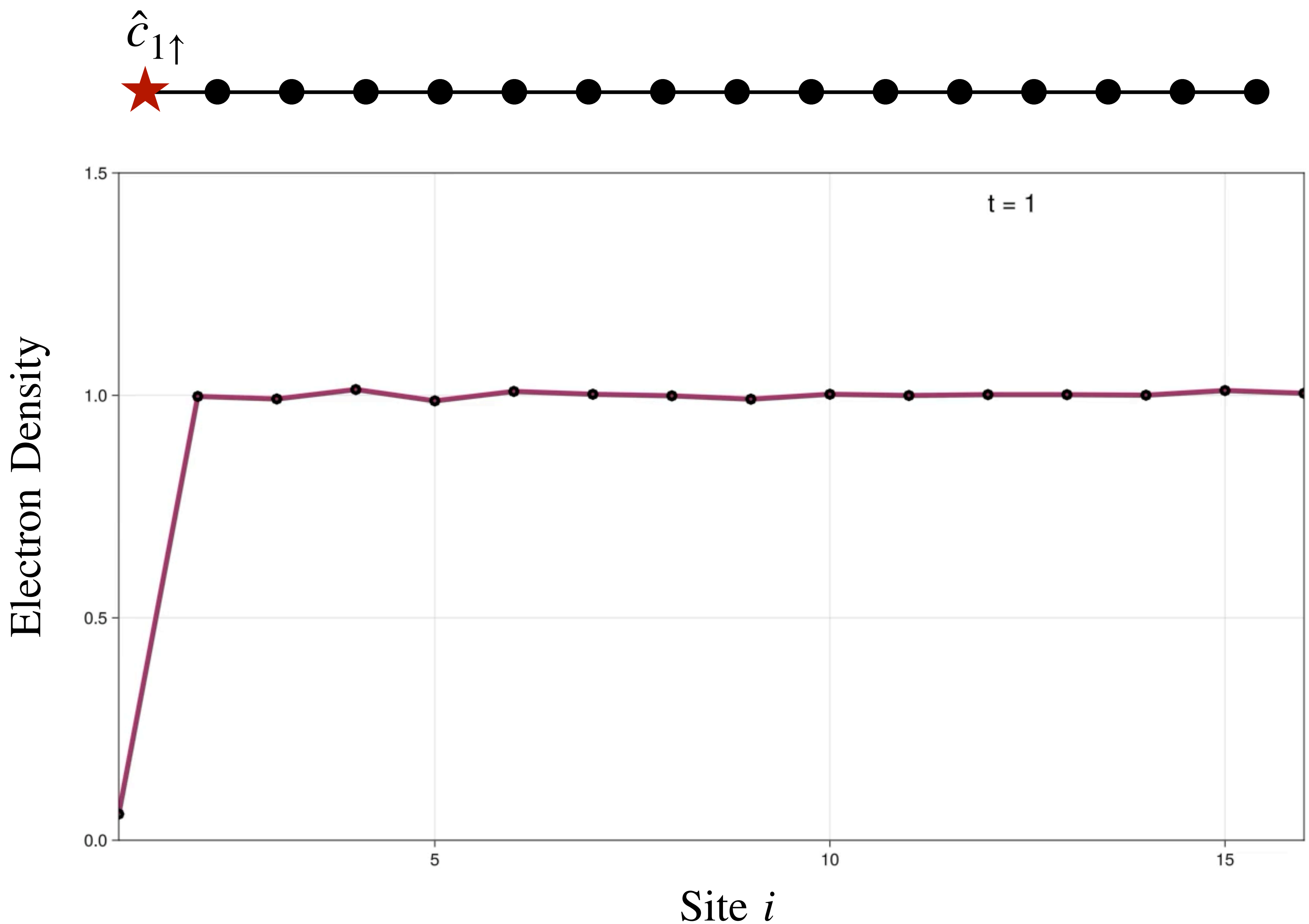




*Impurity model with uniform hopping and U=6

Tensor Network Algorithms

TEBD results: electron chain* after removing particle on site 1



*Impurity model with uniform hopping and U=6

Tensor Network Algorithms

TEBD Limitations:

Tensor Network Algorithms

TEBD Limitations:

- Entanglement of states generically **grows linearly** in time

Tensor Network Algorithms

TEBD Limitations:

- Entanglement of states generically **grows linearly** in time

- TEBD is limited to nearest-neighbor interactions in 1D. If we want to apply longer range gates we would need SWAP gates

TEBD Limitations:

- Entanglement of states generically **grows linearly** in time

- TEBD is limited to nearest-neighbor interactions in 1D. If we want to apply longer range gates we would need SWAP gates

Time-dependent variational principle (TDVP)

TEBD Limitations:

- Entanglement of states generically **grows linearly** in time

- TEBD is limited to nearest-neighbor interactions in 1D. If we want to apply longer range gates we would need SWAP gates

Time-dependent variational principle (TDVP)

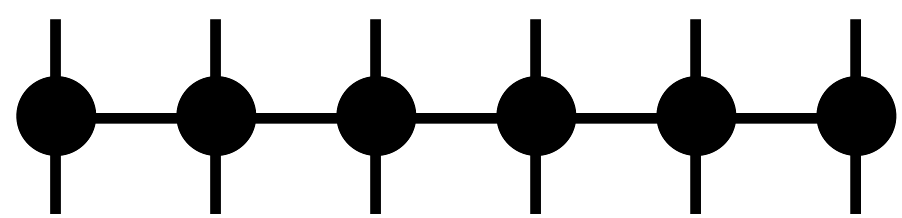
- Lets us handle long range interactions

Tensor Network Algorithms

TDVP algorithm

Time evolve via a DMRG-like sweeping procedure

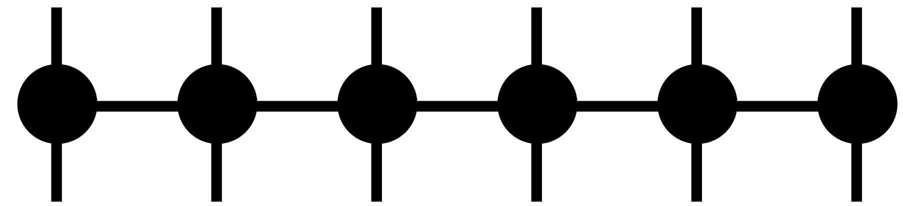
Start with the full Hamiltonian

$H =$ 

Can be a local or non-local Hamiltonian now

Tensor Network Algorithms

TDVP algorithm [1]

$H =$ 

Solve the Schrödinger equation in the manifold of MPS of fixed bond dimension χ

$$\frac{d\,|\,\psi(t)\rangle}{dt} = -\,\mathrm{i}\mathscr{P}_{|\psi(t)\rangle} H\,|\,\psi(t)\rangle$$

$\mathscr{P}_{|\psi(t)\rangle}$ is the projector onto the tangent space of this manifold for the current state

[1] J. Haegeman et al, TDVP for Quantum Lattices, PRL **107** (2011)

TDVP algorithm

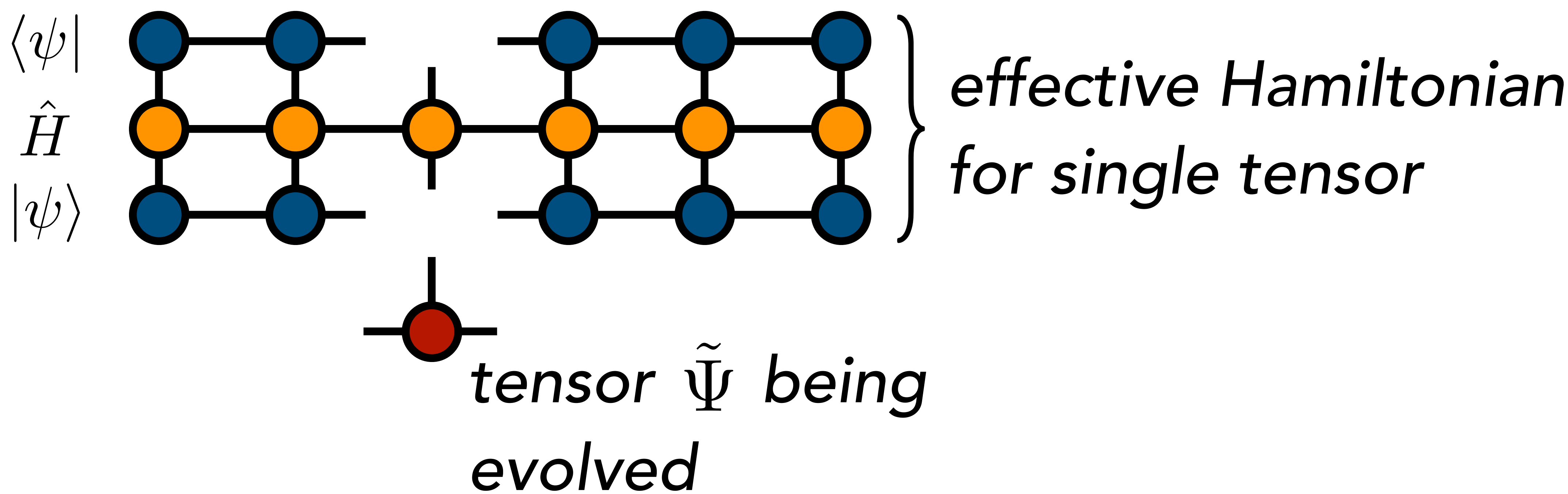
$$\frac{d\,|\psi(t)\rangle}{dt} = -\,\mathrm{i}\mathscr{P}_{|\psi(t)\rangle} H\,|\psi(t)\rangle$$

$$|\psi(t+\delta t)\rangle = e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle} H \delta t}\,|\psi(t)\rangle$$

$\mathscr{P}_{|\psi(t)\rangle}$ can be constructed exactly and we Trotterize the exponential at the level of the projector, not the Hamiltonian, solving the local integral equation that arises site by site.

Tensor Network Algorithms

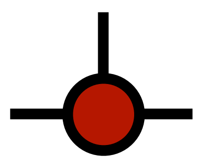
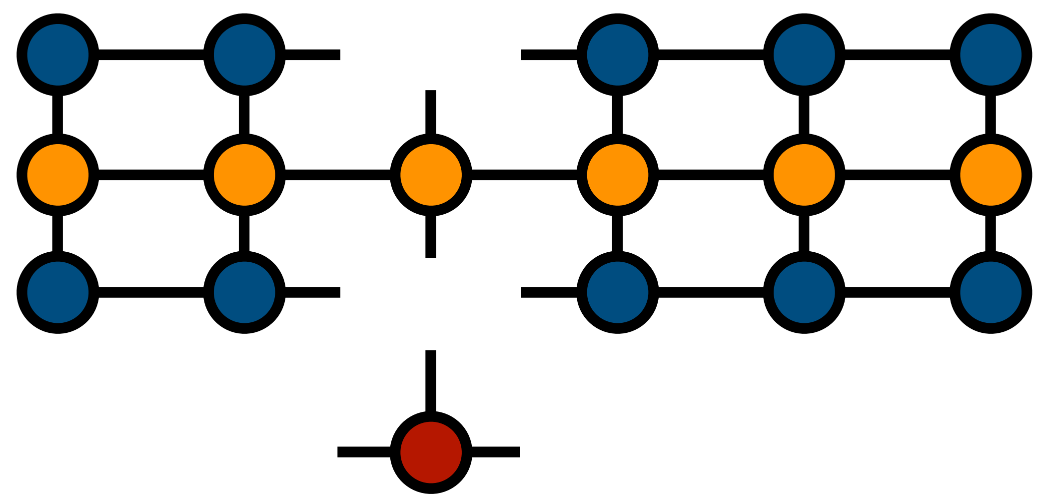
TDVP algorithm

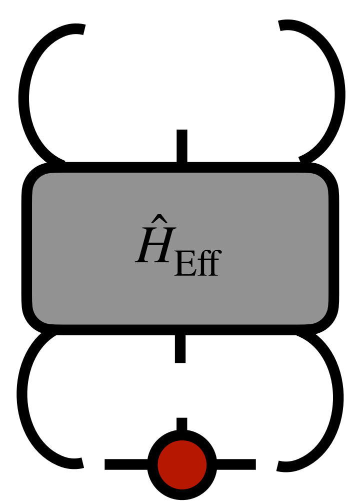


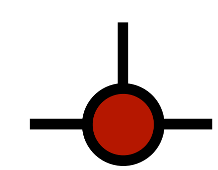
Tensor Network Algorithms

At each configuration, integrate time forward by small amount

Solving:

$i\frac{d}{dt}$  $=$ 

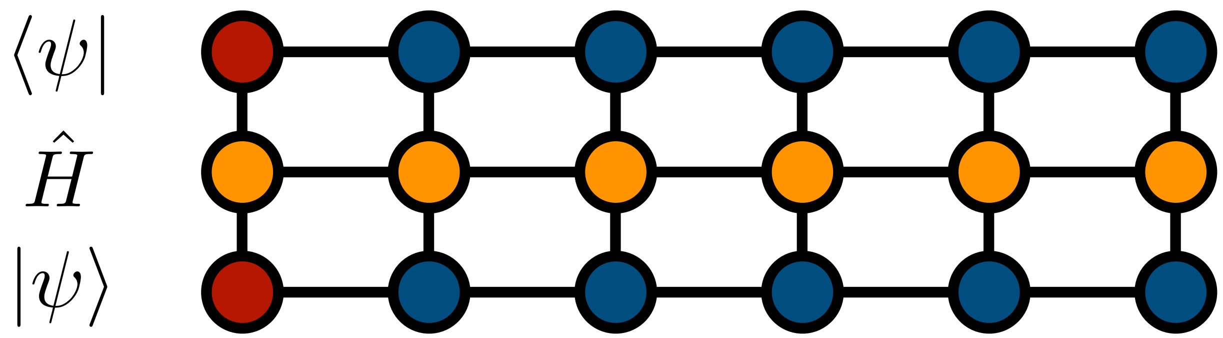
from $t \rightarrow t + dt$ $=$ 

$= \hat{H}_{\mathrm{Eff}} \cdot$ 

Tensor Network Algorithms

TDVP algorithm

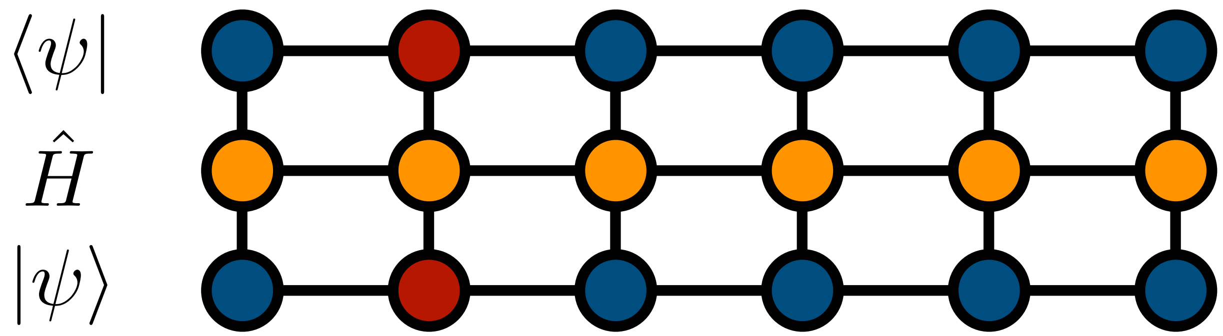
"Sweep" just like in DMRG, but solve a different local optimization problem (by integrating the local equation of motion induced by the Projected Hamiltonian $e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle}H\delta t}$)



Tensor Network Algorithms

TDVP algorithm

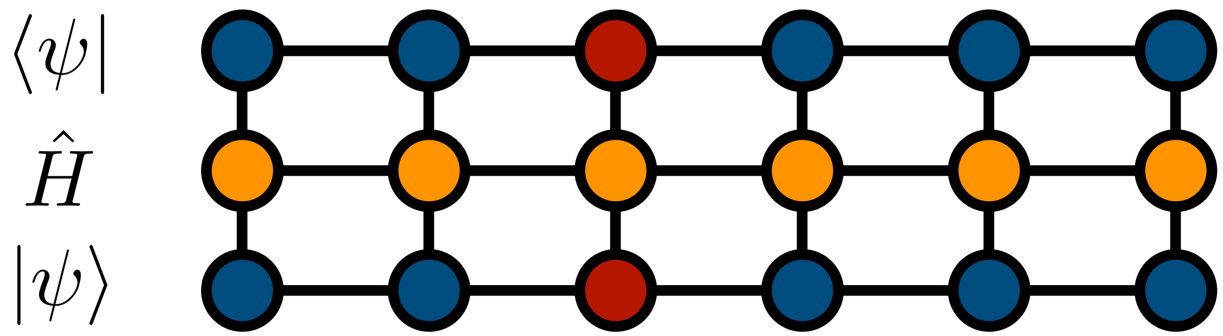
“Sweep” just like in DMRG, but solve a different local optimization problem (by integrating the local equation of motion induced by the Projected Hamiltonian $e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle}H\delta t}$)



Tensor Network Algorithms

TDVP algorithm

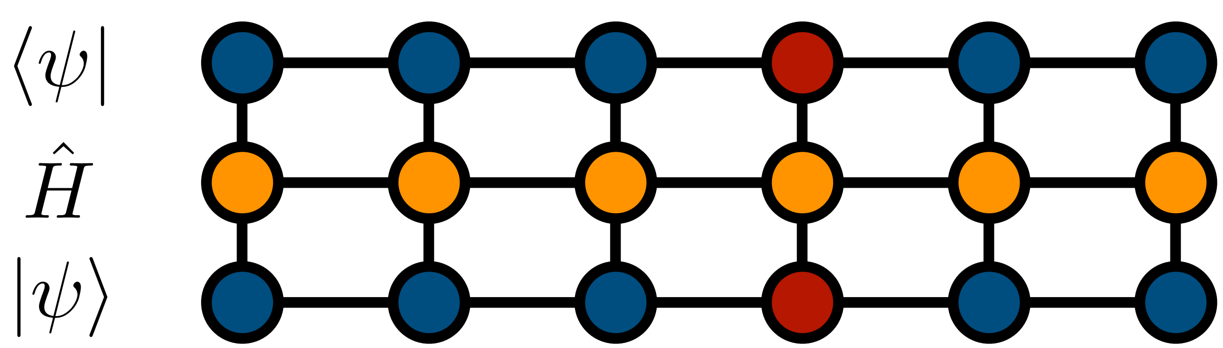
"Sweep" just like in DMRG, but solve a different local optimization problem (by integrating the local equation of motion induced by the Projected Hamiltonian $e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle}H\delta t}$)



Tensor Network Algorithms

TDVP algorithm

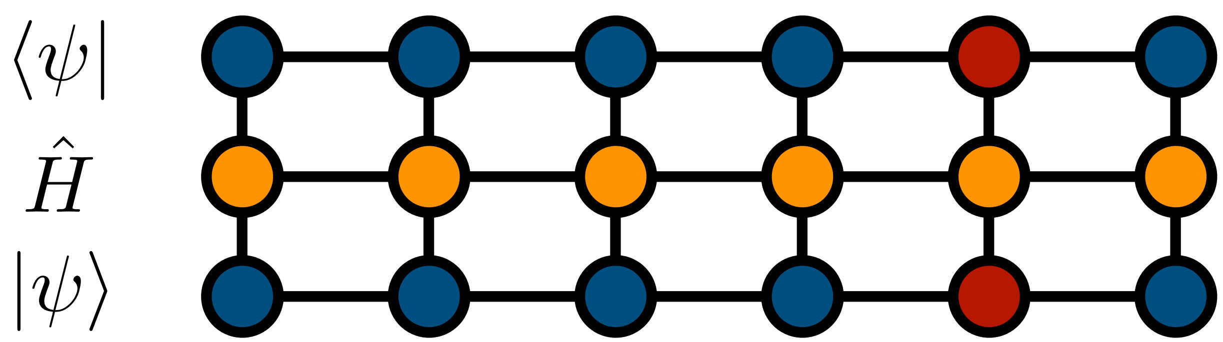
"Sweep" just like in DMRG, but solve a different local optimization problem (by integrating the local equation of motion induced by the Projected Hamiltonian $e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle}H\delta t}$)



Tensor Network Algorithms

TDVP algorithm

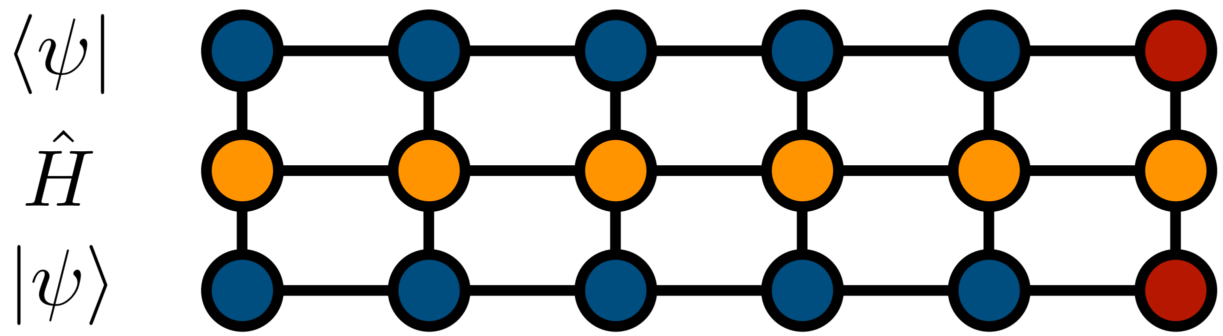
"Sweep" just like in DMRG, but solve a different local optimization problem (by integrating the local equation of motion induced by the Projected Hamiltonian $e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle}H\delta t}$)



Tensor Network Algorithms

TDVP algorithm

"Sweep" just like in DMRG, but solve a different local optimization problem (by integrating the local equation of motion induced by the Projected Hamiltonian $e^{-\mathrm{i}\mathscr{P}_{|\psi(t)\rangle}H\delta t}$)



Tensor Network Algorithms

TDVP algorithm

- Sweeping algorithm, like DMRG. One-site version (fixed bond dimension) and two-site (adaptive bond dimension) versions exist.

- Scales as $\mathcal{O}(L\chi^3 t)$ for a fixed total time t, just like TEBD.

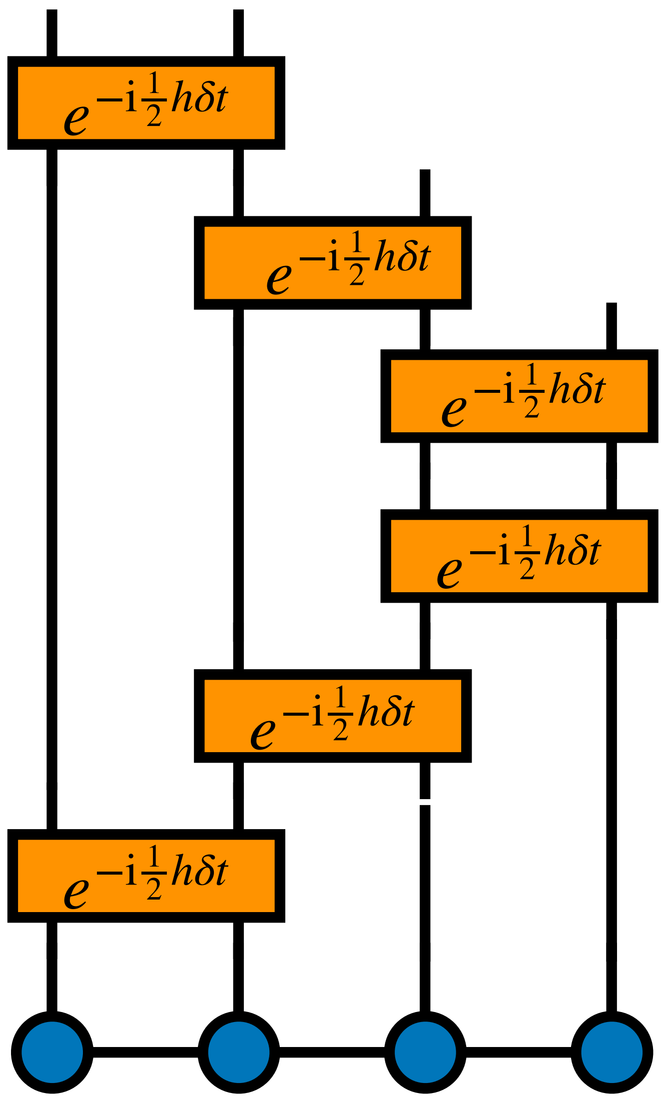
- Handles long-range interactions

- “State-of-the-art” for time evolving an MPS

TDVP and TEBD Applications

What can we do with these time-evolution algorithms?

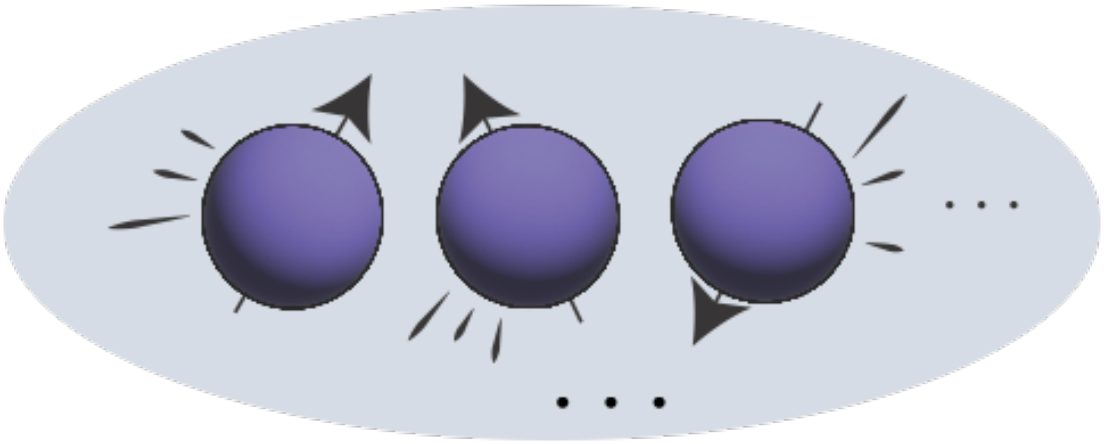
- Time evolving block decimation (TEBD)
- Time-dependent variational principle (TDVP)



TDVP and TEBD Applications

Quenches are the immediate use case of TEBD and TDVP

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

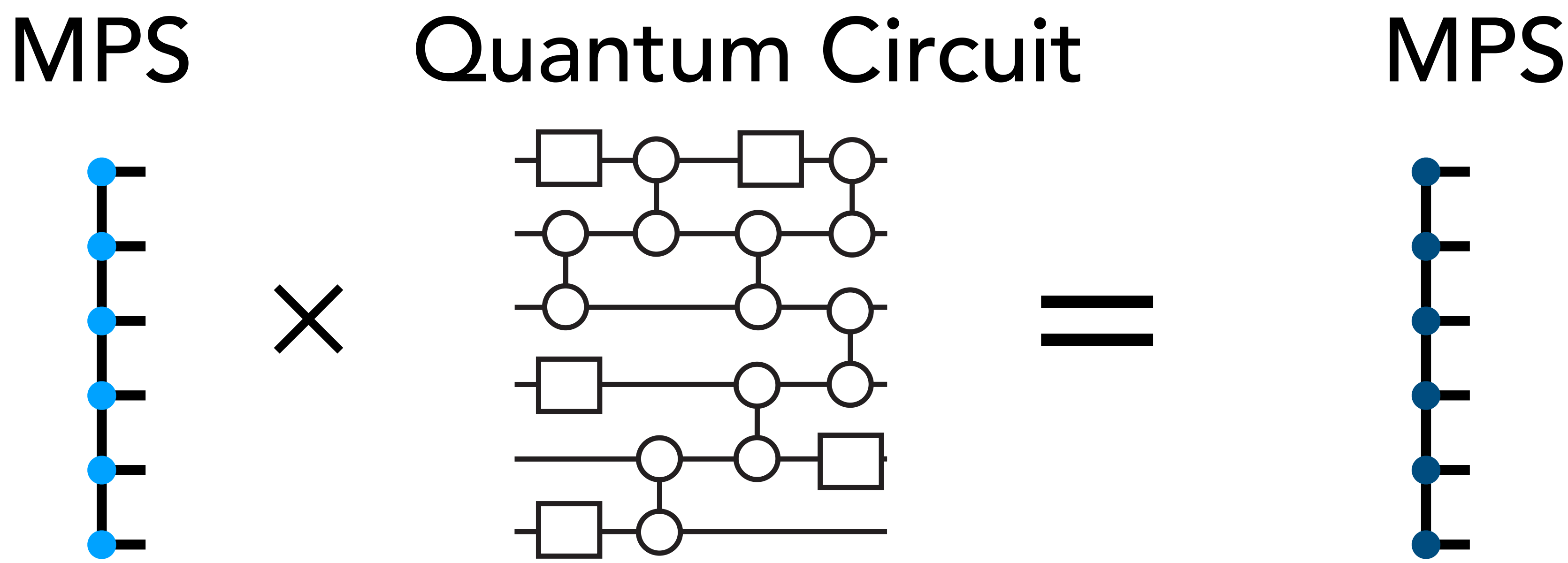


Algorithm Runtime: $\mathcal{O}(L\chi^3 t)$

Out-of-equilibrium physics is a very rich, active research topic

TDVP and TEBD Applications

Simulating quantum circuits is possible with TEBD



Can even sample perfectly $x \sim P(x) = |\langle x | \psi \rangle|^2$ from an MPS $|\psi\rangle$ in $\mathcal{O}(L\chi^3)$ time

- Zhou, EMS, Waintal, "What Limits the Simulation of Quantum Computers?", [PRX 10, 041038](#) (2020)
- Ayral et al., "DMRG Algorithm for Simulating Quantum Circuits with a Finite Fidelity", [PRX Quantum 4, 020304](#) (2023)

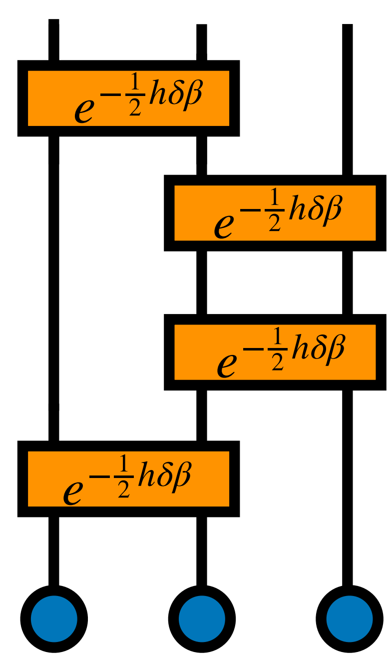
TDVP and TEBD Applications

Imaginary time evolution (TEBD or TDVP)

We can evolve in imaginary time with $t \to \mathrm{i}\beta$ to get ground states or even thermal states (by evolving an MPO)

$$\lim_{\beta\to\infty} e^{-\beta H}|\psi\rangle = \lim_{n\to\infty} \prod_{k=1}^{n} e^{-\delta\beta H}|\psi\rangle \to |\psi_{GS}\rangle$$

$$e^{-\delta\beta H}|\psi\rangle =$$



TDVP and TEBD Applications

We can even do finite temperature whilst staying in the pure state picture

First observe, for any basis, $\beta = \dfrac{1}{T}$

$$e^{-\beta \hat{H}} = \sum_i \underbrace{e^{-\frac{\beta}{2}\hat{H}}|i\rangle}_{\substack{\text{Freedom to} \\ \text{choose these}}}\langle i|e^{-\frac{\beta}{2}\hat{H}} \propto \sum_i |\phi_i\rangle\langle\phi_i|$$

Freedom to choose these $|\phi_i\rangle \propto e^{-\frac{\beta}{2}\hat{H}}|i\rangle$

We have

$$e^{-\beta \hat{H}} = \sum_i e^{-\frac{\beta}{2}\hat{H}} |i\rangle\langle i| e^{-\frac{\beta}{2}\hat{H}} \propto \sum_i |\phi_i\rangle\langle\phi_i|$$

Obtain observables as

$$\langle \hat{A} \rangle = \frac{1}{\mathcal{Z}} \mathrm{Tr}\left[e^{-\beta \hat{H}} \hat{A}\right] = \frac{1}{\mathcal{Z}} \sum_i P_i \, \langle \phi_i | \hat{A} | \phi_i \rangle$$

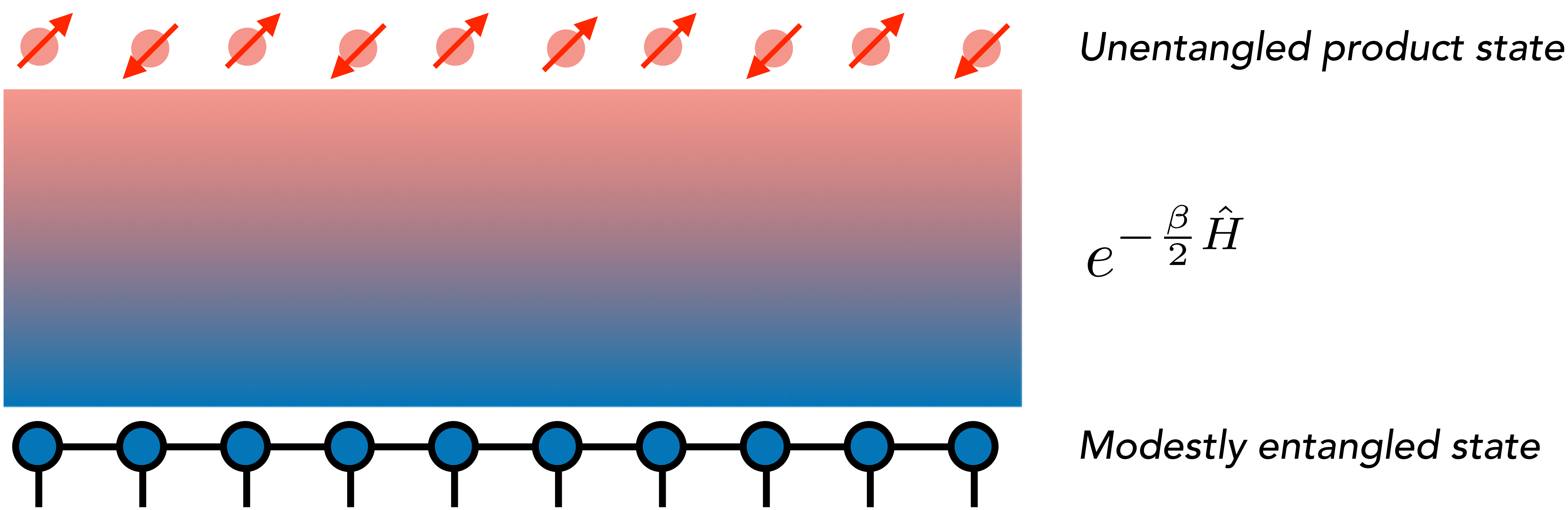
an average over
pure states

Expanding $e^{-\beta \hat{H}} \propto \sum_i |\phi_i\rangle\langle\phi_i|$

To give tensor networks their best chance

choose $|\phi_i\rangle \propto e^{-\frac{\beta}{2}\hat{H}}|i\rangle$ to "descend" from

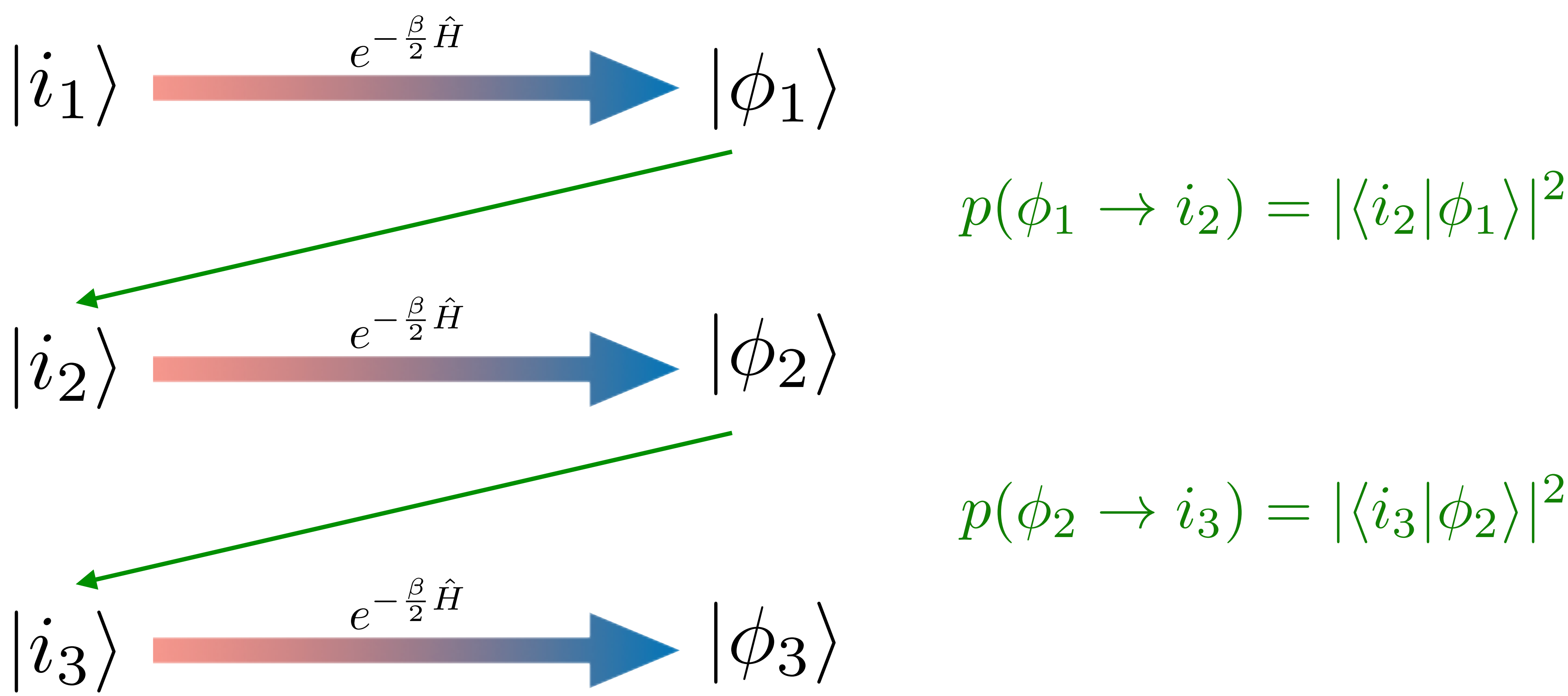
<u>untentangled</u> (zero-entanglement) states



Solved problem of representing $|\phi_i\rangle \propto e^{-\frac{\beta}{2}\hat{H}}|i\rangle$
(choose $|i\rangle$ as product states)

One more **problem**: too many states –
there are exponentially many $|i\rangle$ and thus $|\phi_i\rangle$

Solution: sample over the $|i\rangle$

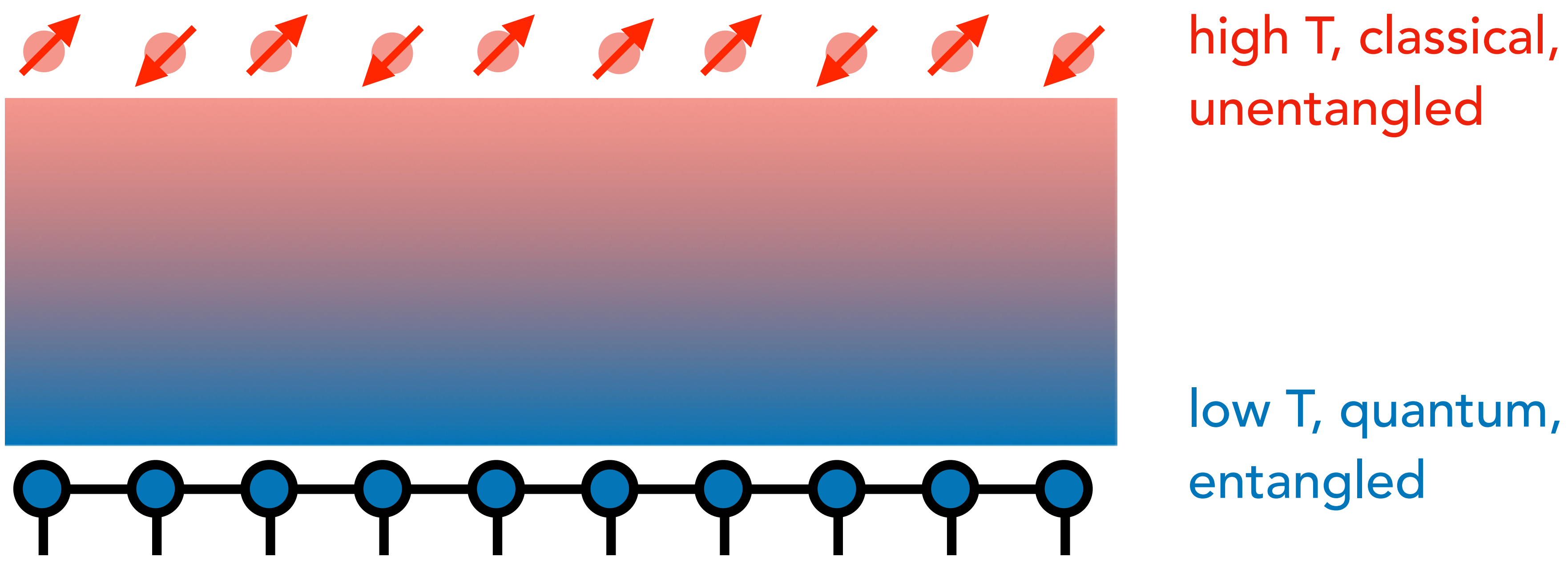


Algorithm just described named
minimally entangled typical thermal states (**METTS**)[1,2]

$$|\phi_i\rangle \propto e^{-\frac{\beta}{2}\hat{H}}|i\rangle$$ METTS wavefunction

Quantum Monte Carlo where samples are
entangled wavefunctions, not classical configurations

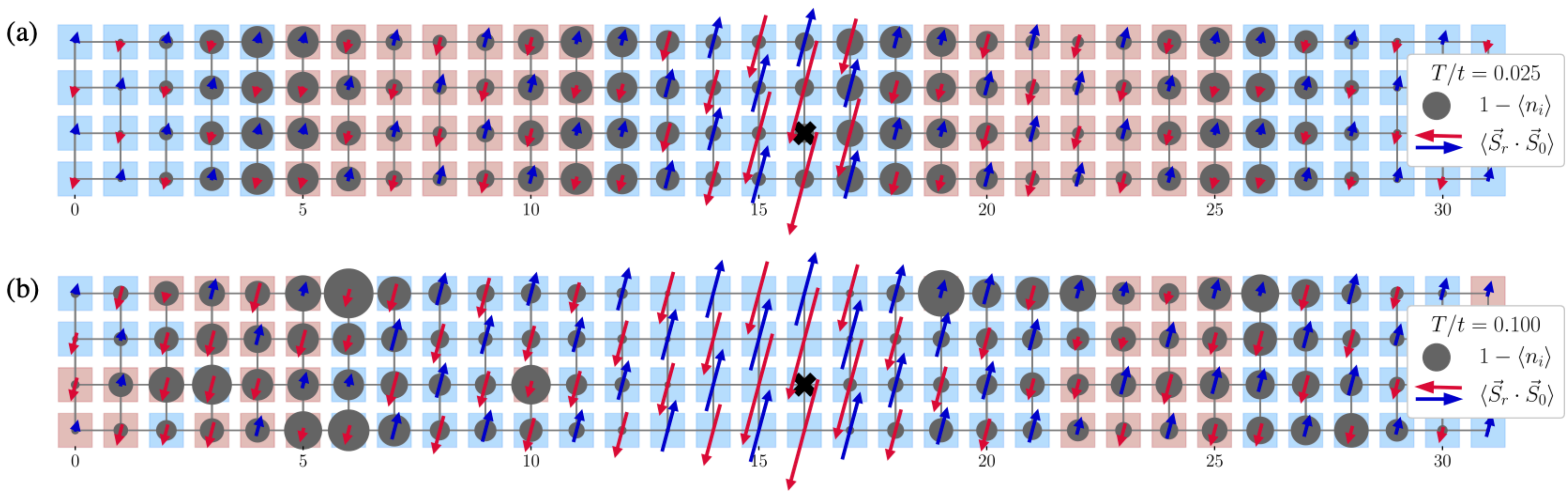
Classicality of METTS depend on T



[1] S.R. White, PRL (2009)
[2] Stoudenmire, White, NJP (2010)

Minimally entangled typical thermal states (**METTS**)

Used by researchers here at CCQ to study finite temperature electronic systems [1]:



[1] A. Wietek et al, PRX **11** (2021)

Further Applications of TDVP and TEBD

- Open Quantum Systems $\rho(t) = \exp(\mathscr{L}t)\rho(0)$

- Green's functions

- OTOCS (Out-of-time ordered correlaters)

- Many more…

- Now lets do some time evolution with ITensorMPS.jl