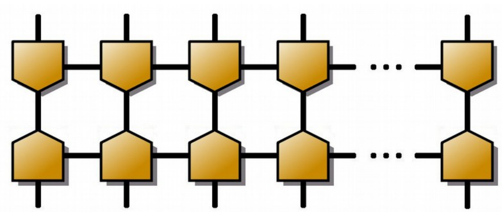
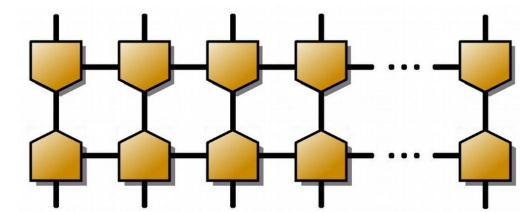
Positive Tensor Network approach for simulating open quantum many-body systems



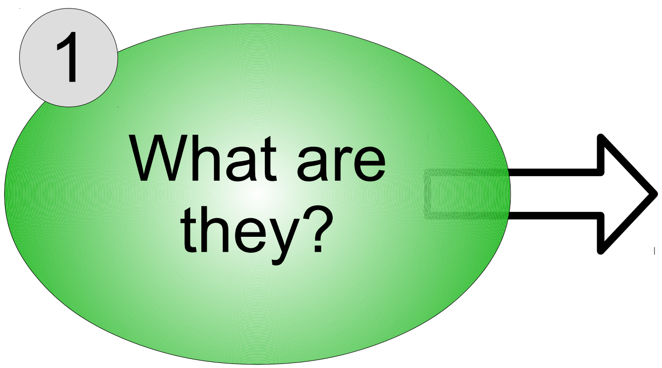
19 / 9 / 2016



A. Werner, D. Jaschke, P. Silvi, M. Kliesch,
T. Calarco, J. Eisert and S. Montangero

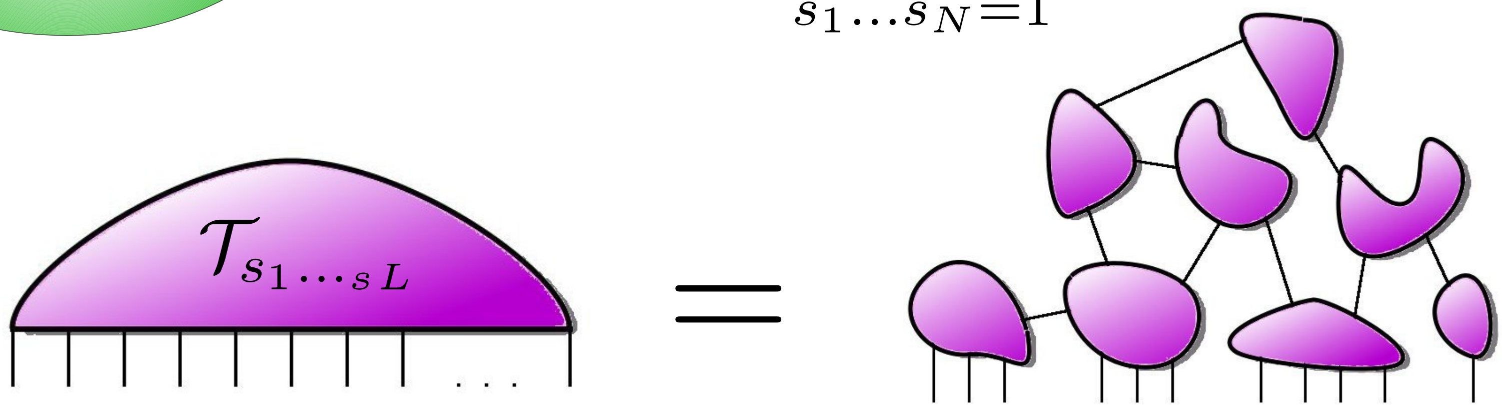
*PRL **116**, 237201 (2016)*

Tensor Network ansatz states:



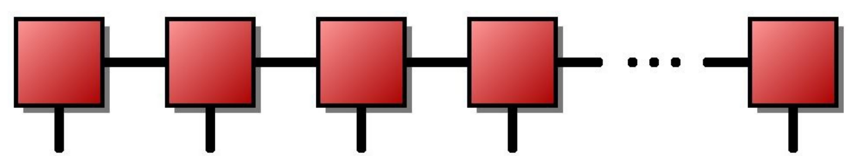
Classes of tailored variational quantum many-body wavefunctions

$$|\Psi\rangle = \sum_{s_1 \dots s_N = 1}^{d} \mathcal{T}_{s_1 \dots s_N} |s_1, \dots, s_N\rangle$$



Example[1,2]: **Matrix Product States**

$$\mathcal{T}_{s_1 \dots s_N} = \sum_{\alpha_2 \dots \alpha_N = 1}^{\chi} A^{[1]s_1}_{\alpha_2} A^{[2]s_2}_{\alpha_2,\alpha_3} A^{[3]s_3}_{\alpha_3,\alpha_4} \cdots A^{[N]s_N}_{\alpha_N} =$$



1) S. Romer, S. Ostlund; Phys. Rev. B **55**, 2164 (1997)

2) U. Schollwoeck, Annals of Physics **326**, 96 (2011)

Tensor Network ansatz states:

2

What are
They good for?

- Simulating ground states of 1D, 2D quantum lattices with OBC (DMRG[3])
- … with PBC
- … in infinite systems
- Simulating out-of equilibrium dynamics

3

Why are
They good?

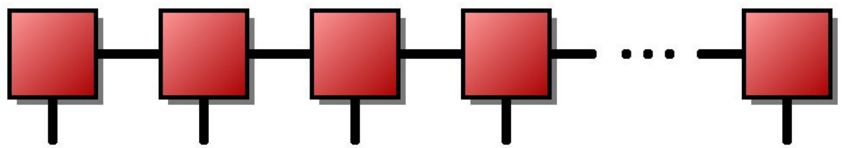
Physical quantum many-body states obey precise entanglement scaling laws

Tensor networks encode these states faithfully and efficiently

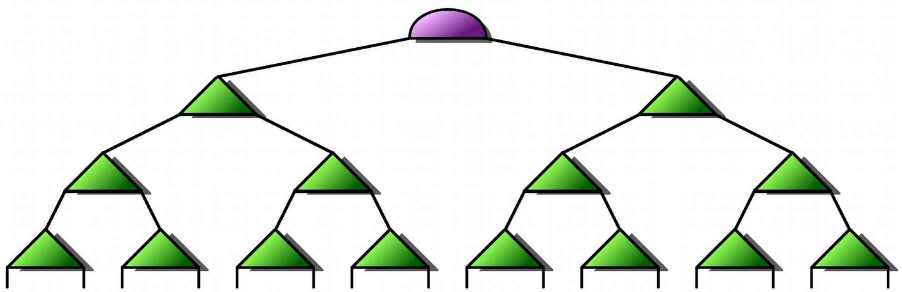
3) S. White; Phys. Rev. Lett. **69**, 2863 (1992)

Zoology of Tensor Networks:

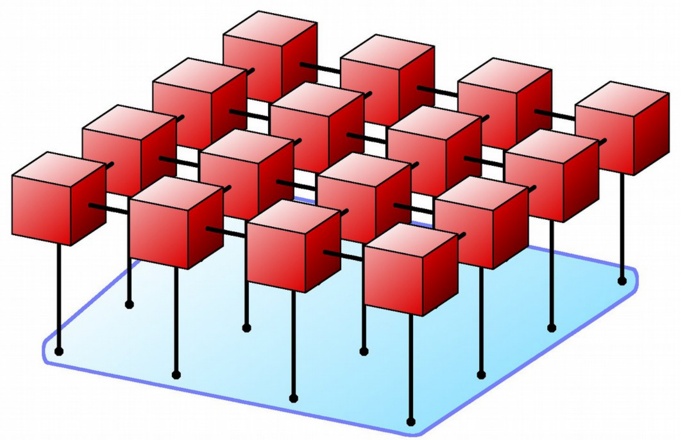
Matrix Product States (MPS)



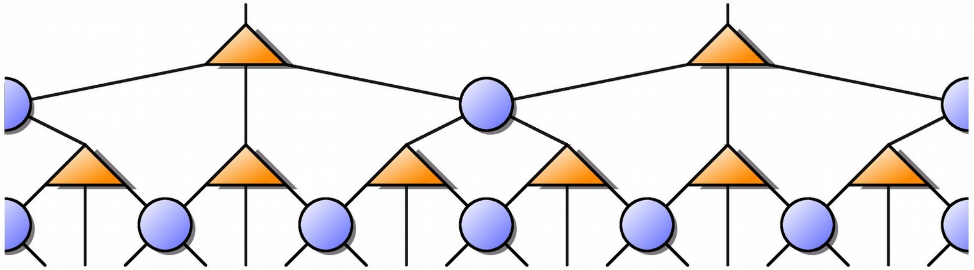
Tree Tensor Networks



Projected Entangled Pair States (PEPS)[4]



Multiscale Entanglement Renormalization Ansatz (MERA)[5]



...and many more.

4) Verstaete, Wolf, Perez-Garcia, Cirac; PRL **96**, 220601 (2006)

5) G. Vidal; PRL **99**, 220405 (2007)

Motivation:

We want to extend the known quantum many-body (QMB) dynamics algorithms to encompass Open Systems.
The focus is:

- Disregard reservoir dynamics
- Capture both transient and steady behavior
- Stay numerically efficient and control precision/errors

Pathways

1) *Stochastic unravelling* of pure TN-state dynamics (quantum jumps, quantum trajectories)

2) Extend TN-states to describe *density matrices*

Pathway A: Quantum Jumps

Sample several pure states and simulate stochastic trajectories according to (at first order)[6]:

$$|\psi(t+\delta t)\rangle = \begin{cases} \dfrac{e^{iH_{\mathrm{eff}}\delta t}|\psi(t)\rangle}{\sqrt{P}} & \text{with } P = 1 - \sum_j p_j \\[2ex] \dfrac{L_j|\psi(t)\rangle}{\sqrt{p_j/\delta t}} & \text{with } p_j = \delta t \langle\psi(t)|L_j^\dagger L_j|\psi(t)\rangle \end{cases}$$

Where $H_{\mathrm{eff}} = H - \frac{i}{2}\sum_j L_j^\dagger L_j$ and Lindbladians L_j

Perform dynamics simulations with Tensor Networks.
Reconstruct full dynamics by averaging over the samples.

6) A. J. Daley; Adv. Phys. **63**, 77 (2014)

Pathway B: QMB Density Matrices

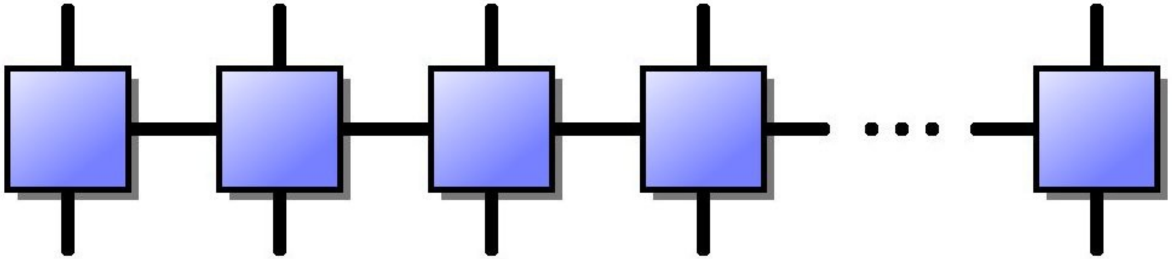
$$\rho = \sum_{s_1 \dots s_L}^{d} \sum_{r_1 \dots r_N}^{d} \mathcal{T}_{s_1 \dots s_N}^{r_1 \dots r_N} |s_1, \dots, s_N\rangle\langle r_1, \dots, r_N|$$

writing $\mathcal{T}_{s_1 \dots s_N}^{r_1 \dots r_N}$ as a tensor network.

A viable way: the MPDO

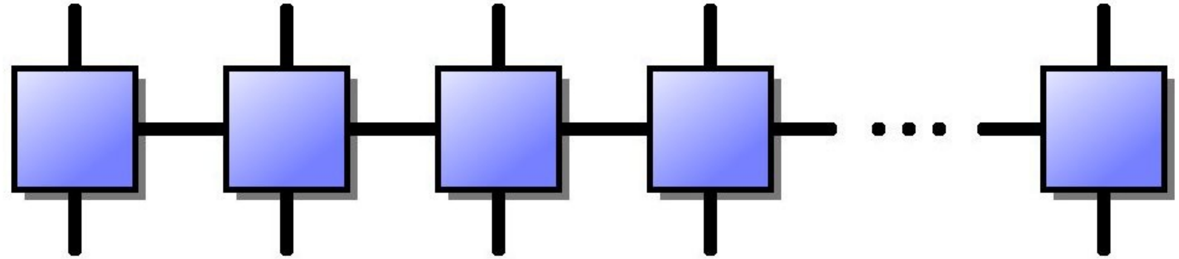
A possible path to do so is with Matrix Product (Density) Operators: MPDO[7]

$$\mathcal{T}_{s_1 \dots s_N}^{r_1 \dots r_N} = \sum_{\alpha_2 \dots \alpha_N = 1}^{\chi} A_{\alpha_2}^{[1] s_1, r_1} A_{\alpha_2, \alpha_3}^{[2] s_2, r_2} A_{\alpha_3, \alpha_4}^{[3] s_3, r_3} \dots A_{\alpha_N}^{[N] s_L, r_N} =$$



7) Verstraete, Cirac; PRL **93**, 207204 (2004). Zwolak, Vidal; PRL **93**, 207205 (2004)

The MPDO way



Features:

- Finite Temperature states of short-range Hamiltonian
- Can simulate open-system dynamics
- Direct targeting of steady states[8]

Drawback:

Positivity is not guaranteed (positivity check is NP-hard).

8) J. Cui, I. Cirac, M. C. Banuls; Phys. Rev. Lett. **114**, 220601 (2015)

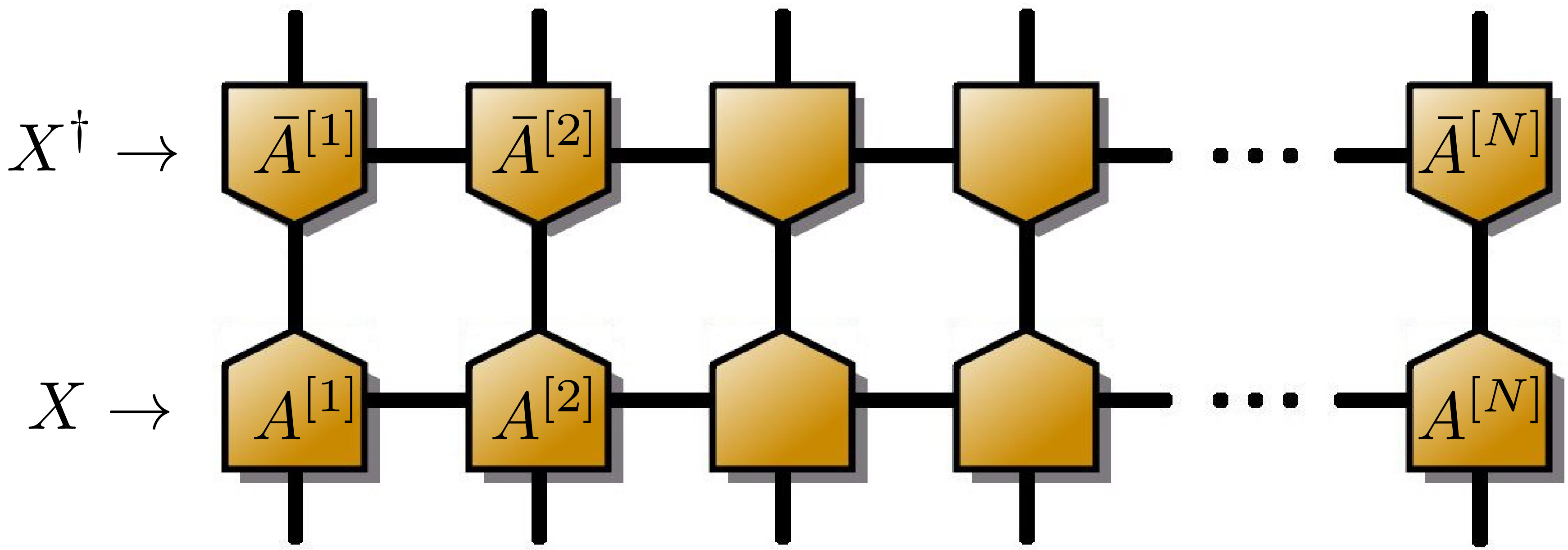
How can we impose positivity in a natural way?

Simple trick: write the density matrix as

$$\rho = X X^{\dagger}$$

where X is a many-body operator, which we can write as a matrix product operator

$\rho =$



Locally Purified Tensor Network (LPTN)

Why this name[9]?

Assume to extend the L system sites with L ancillary sites

$$|X\rangle\!\rangle = \sum_{s_1 \dots s_N}^{d} \sum_{q_1 \dots q_N}^{K} X_{s_1 \dots s_N}^{q_1 \dots q_N} |s_1 \dots s_N\rangle_{\mathrm{system}} \otimes |q_1 \dots q_N\rangle_{\mathrm{ancilla}}$$

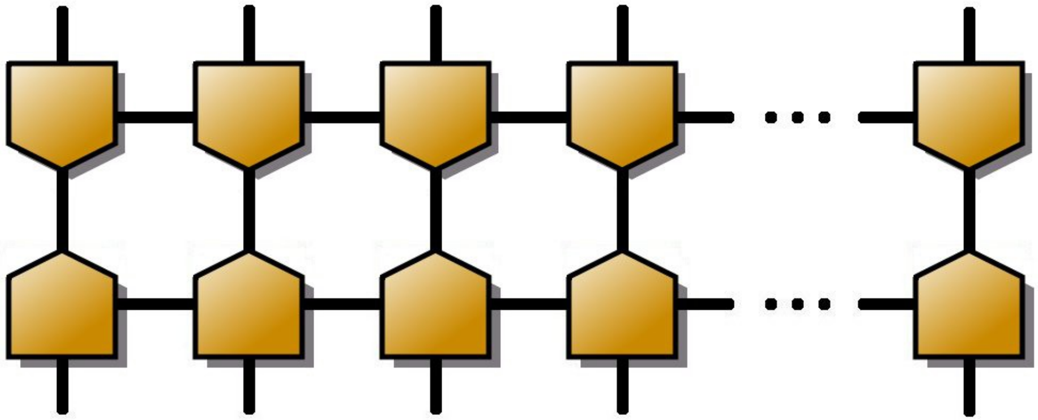
If we now disregard (trace away) the ancillas, we get

$$\rho = \mathrm{Tr}_{\mathrm{ancilla}} \left[|X\rangle\!\rangle\langle\!\langle X| \right] = X X^{\dagger}$$

We have a purification representation where every site has a dedicated bath (of dimension K).

9) G. De las Cuevas, N. Schuch, D. Perez-Garcia, I.J. Cirac; NJP **15**, 123021 (2013)

The LPTN way



Features:

- Finite Temperature states
- Can simulate open-system transient dynamics (steady states are reached dynamically)
- Positivity always guaranteed

Issues:

Variational constraint: algorithms must preserve the symmetry $X \leftrightarrow X^\dagger$ at all times.

Markovian dynamics with LPTN

And now something familiar…

$$\frac{d\rho}{dt} = i[\rho, H] + \sum_j \left(L_j \rho L_j^\dagger - \frac{1}{2}\{\rho, L_j^\dagger L_j\} \right)$$

with the following conditions:

- Hamiltonian is short-range (nearest-neighbour interactions): $H = \sum_j h_{j,j+1}$
- Lindbladians are local (single site)

Extensible to n-n two-site Lindbladians (not in this talk).

Liouville rep.
in a nutshell

$$\rho \rightarrow |\rho\rangle\rangle$$

$$A\rho B \rightarrow A \otimes B^T |\rho\rangle\rangle$$

Liouville representation will help us

$$\frac{d|\rho\rangle\rangle}{dt} = \left(-iH \otimes \mathbb{I} + i\mathbb{I} \otimes H^T + \mathcal{D}_j\right)|\rho\rangle\rangle = \mathcal{L}|\rho\rangle\rangle$$

where $\mathcal{D}_j = L_j \otimes \bar{L}_j - \frac{1}{2}L_j^\dagger L_j \otimes \mathbb{I} - \frac{1}{2}\mathbb{I} \otimes L_j^T \bar{L}_j$

Because now we discretize the time in finite small intervals δt, and solve the real-time dynamics:

$$|\rho(t+\delta t)\rangle\rangle = \exp(\delta t \mathcal{L})|\rho(t)\rangle\rangle$$

The Algorithm now focuses on how to implement $\exp(\delta t \mathcal{L})$ on the LPTN state efficiently and controlling errors.

Suzuki-Trotter decomposition

At second order, with three operators:

$$e^{\delta t(A+B+C)} = e^{\frac{\delta t}{2}A} e^{\frac{\delta t}{2}B} e^{\delta t C} e^{\frac{\delta t}{2}B} e^{\frac{\delta t}{2}A} + O(\delta t^3)$$

which follows from Baker-Hausdorff formulas.

Let us decompose $\mathcal{L}$ in 3 pieces: $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$

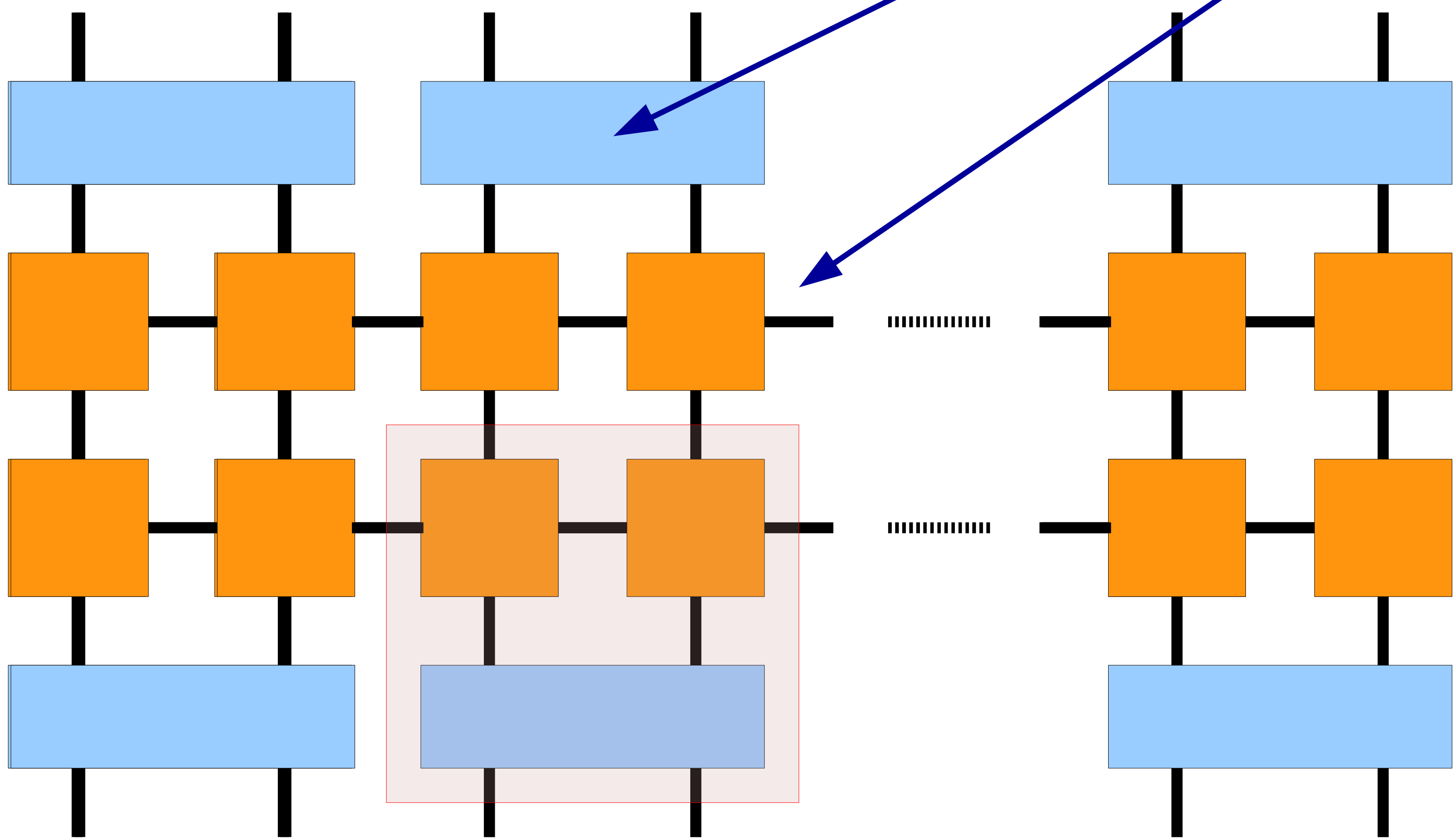
1) Odd-even Hamiltonian $\mathcal{L}_1 = i \sum_j \left(h_{2j-1,2j} \otimes \mathbb{I} - \mathbb{I} \otimes h^T_{2j-1,2j} \right)$

2) Even-odd Hamiltonian $\mathcal{L}_2 = i \sum_j \left(h_{2j,2j+1} \otimes \mathbb{I} - \mathbb{I} \otimes h^T_{2j,2j+1} \right)$

3) All dissipators $\mathcal{L}_3 = \sum_j \mathcal{D}_j$

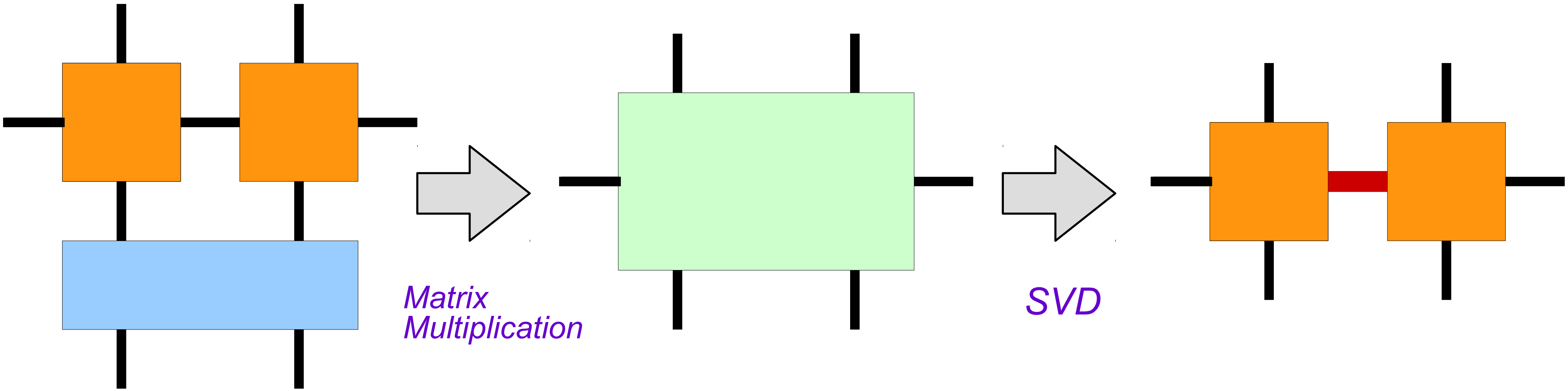
Advantage: each piece is made of commuting terms, therefore

$$e^{\delta t \mathcal{L}_1} |\rho\rangle\rangle = \bigotimes_j \left(e^{-i\delta t\, h_{2j-1,2j}} \otimes e^{i\delta t\, \bar{h}_{2j-1,2j}} \right) |\rho\rangle\rangle$$



Notice: This operation fulfills automatically the top-bottom symmetry requirement.

We need to perform the following linear algebra operation

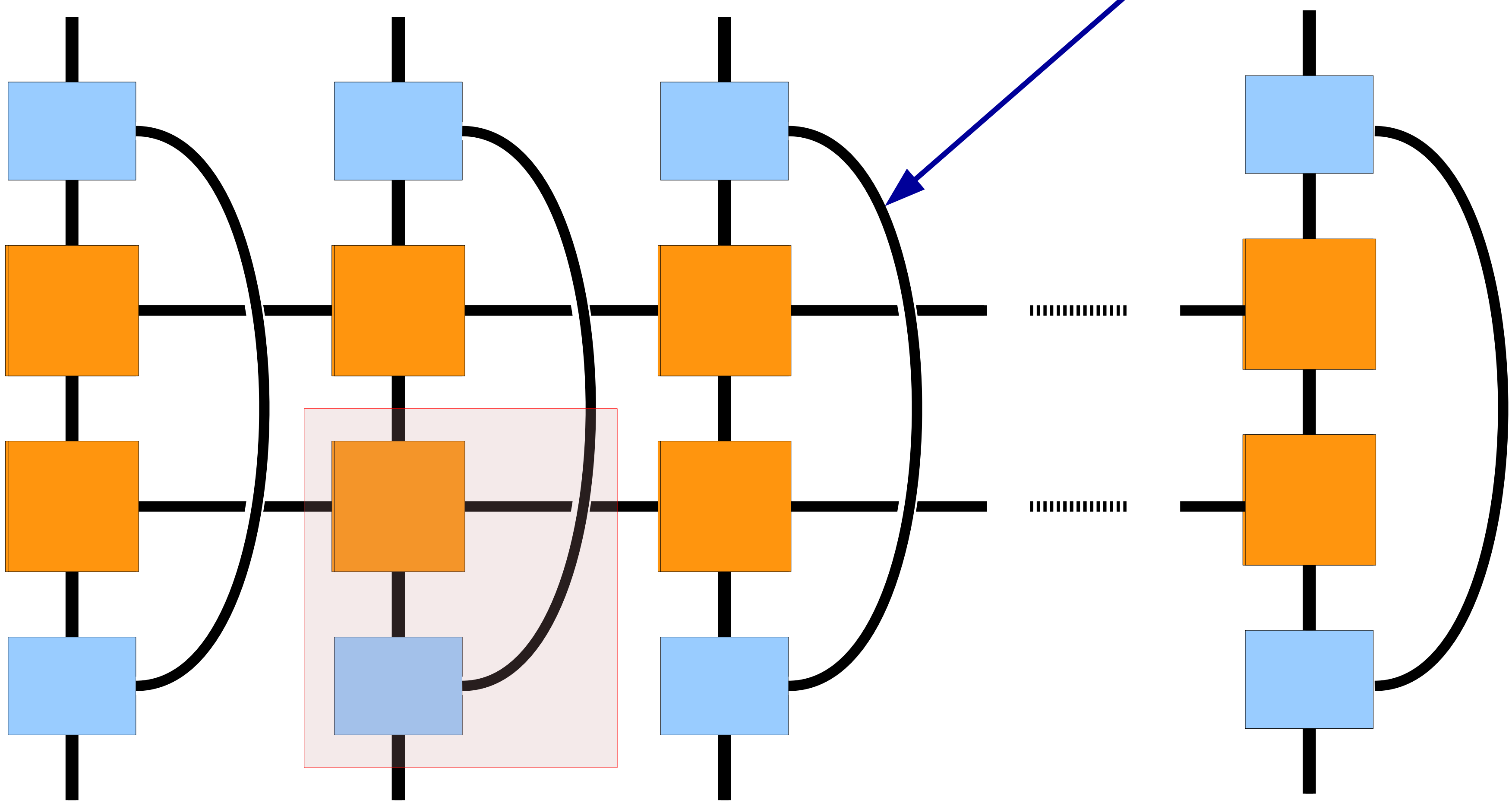


After this operation, the “correlation” bond (dimension of the tensors) is enlarged:

We “compress” it by discarding the smallest values in the singular value decomposition (second source of error).

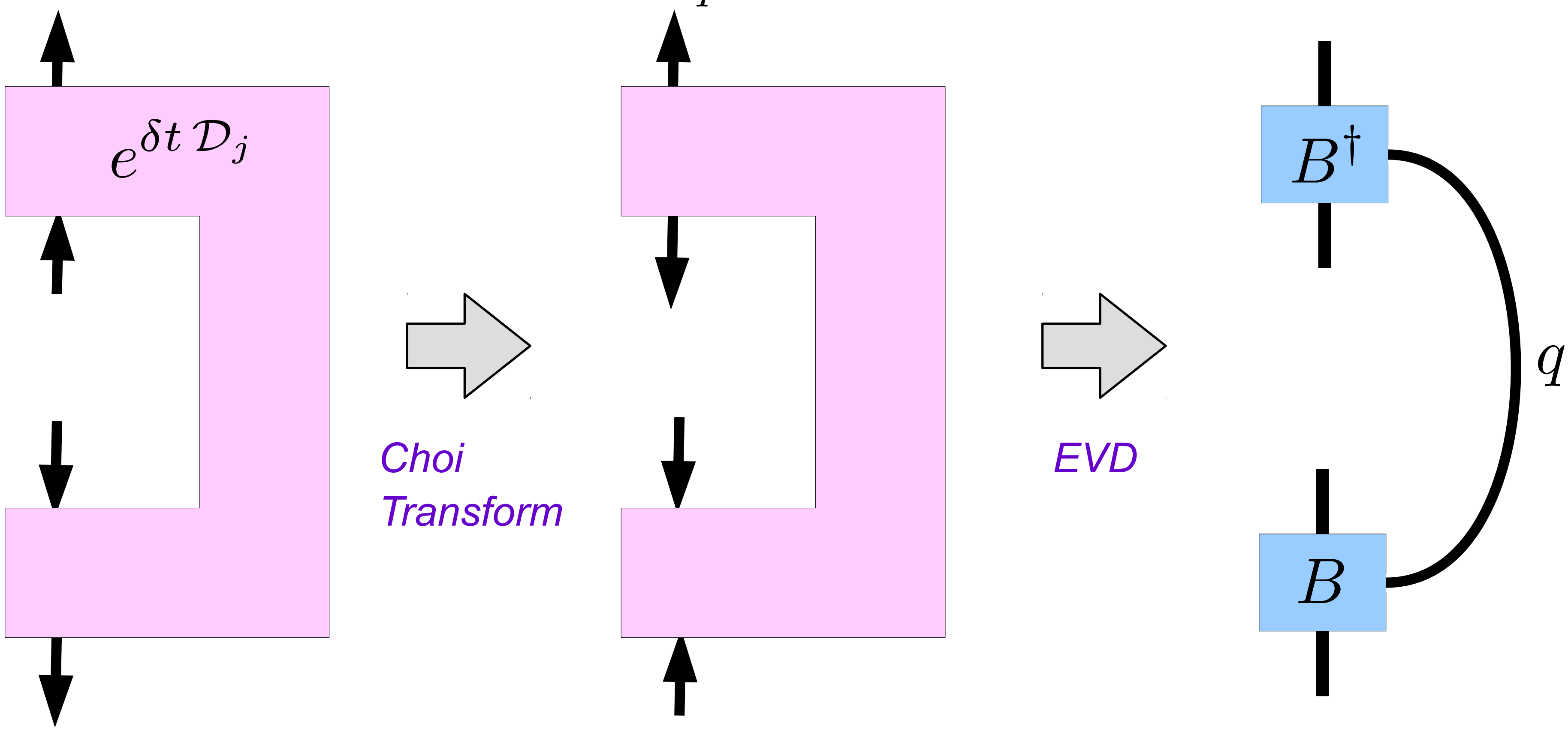
$\mathcal{L}_2$ is performed analogously to $\mathcal{L}_1$.

$\mathcal{L}_3$ requires a bit of care: $e^{\delta t \mathcal{L}_3} |\rho\rangle\rangle = \bigotimes_j \left(e^{\delta t \, \mathcal{D}_j} \right) |\rho\rangle\rangle$



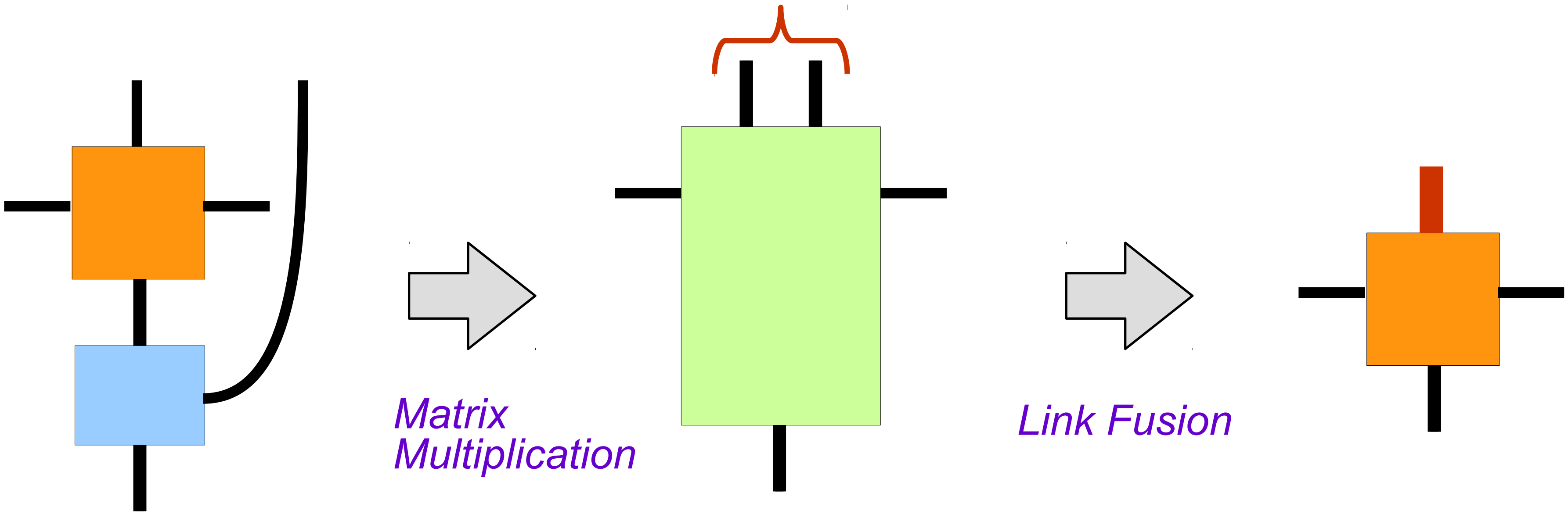
We numerically obtain the Kraus decomposition of the local dissipation quantum channel $e^{\delta t\, \mathcal{D}_j}$, which is CPT.

$$e^{\delta t\, \mathcal{D}_j} = \sum_q^{K'} B_q \otimes \bar{B}_q$$



Notice: The Kraus-decomposed map satisfies the top-bottom symmetry requirement.

The only operation left to perform is:



This time, the “bath” bond is enlarged. We can compress it again via SVD and truncation of the smallest singular values.

<u>Algorithm Complete !</u>

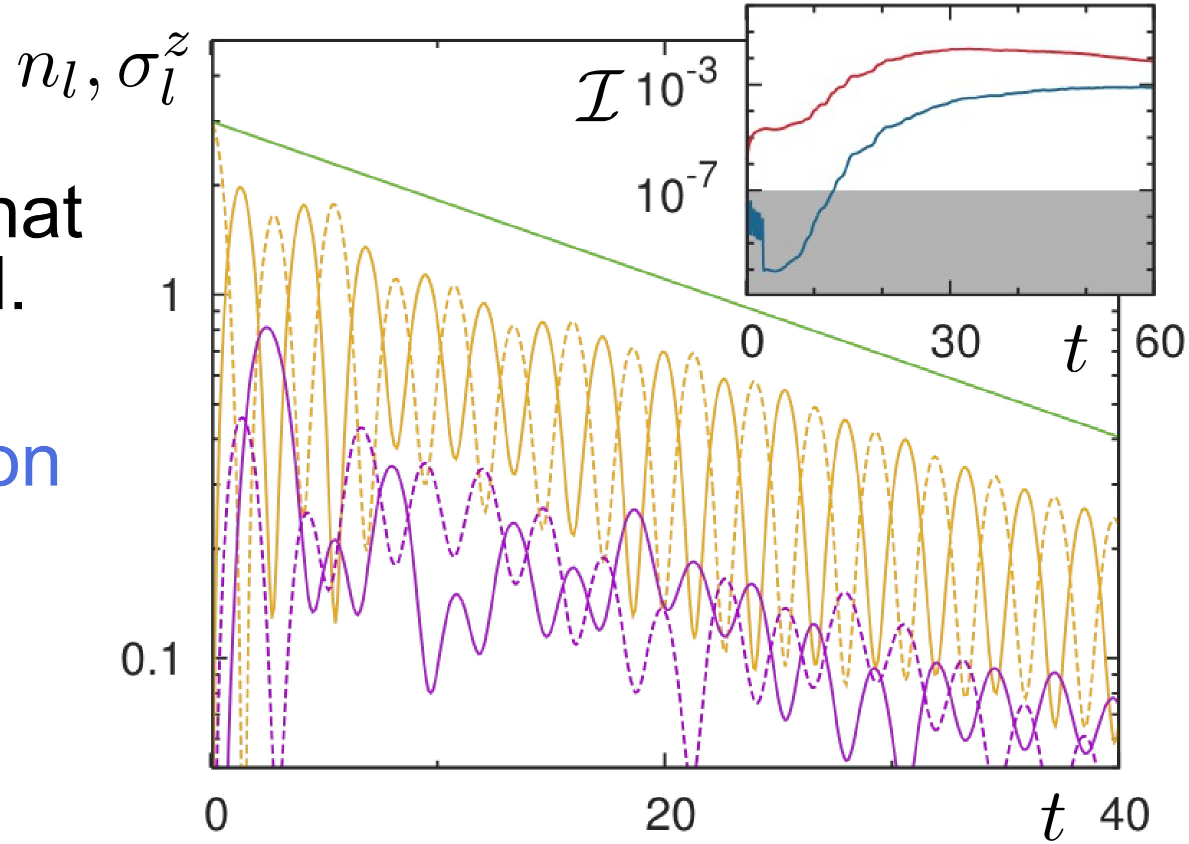
Apologies for being so technical, but that's my job.

Benchmarks

It is high time to prove that our algorithm works well.

B.1) "Photonic Josephson Junction":

System: two spins-1/2, each within an optical cavity.



$$H = \sum_{l=1,2} \left(\alpha_l(\sigma_l^+ c_l + \sigma_l^- c_l^\dagger) + \omega_C n_l + \omega_S \sigma_l^z\right) + \eta(c_1^\dagger c_2 + c_2^\dagger c_1)$$

Dissipation: spontaneous loss $L_{S_l} = \sqrt{\gamma}\,\sigma_l^-, \quad L_{C_l} = \sqrt{\gamma}\,c_l$

<u>Study transient dynamics and compare to exact results</u>

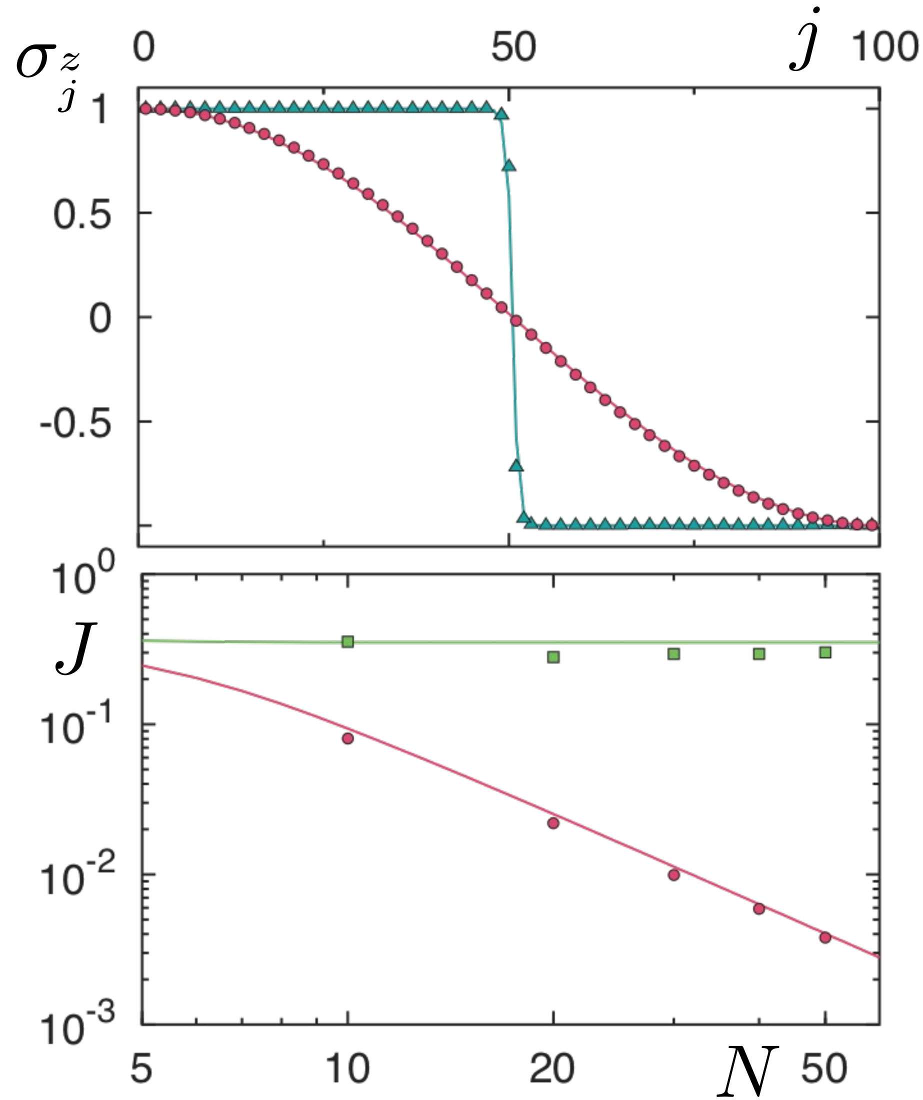
B.2) Fermionic quantum wire:

Spin-1/2 XXZ model (equivalent to Hubbard with density-density int.)

$$H = \sum_j (\sigma_j^x \sigma_{j+1}^x + \sigma_j^y \sigma_{j+1}^y + \Delta \sigma_j^z \sigma_{j+1}^z)$$

Dissipation: particle-source at left edge, particle-drain at right edge

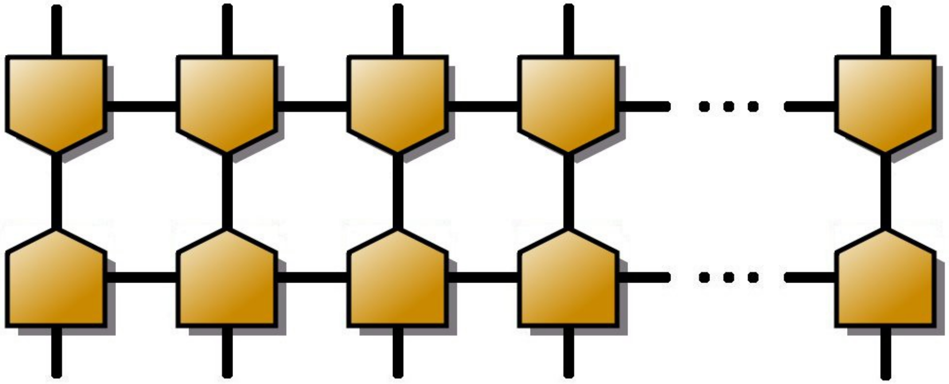
$$L_s = \sqrt{\gamma}\, \sigma_1^+, \quad L_d = \sqrt{\gamma}\, \sigma_N^-$$



We study steady dynamics and measure population and particle current $J = 2\,\mathrm{Im}\langle \sigma_j^+ \sigma_{j+1}^- \rangle$. We compare results with analytical predictions[8].

8) T. Prozen; Phys. Rev. Lett. **107**, 137201 (2011)

Conclusions



We designed an algorithm based on Locally Purified Tensor Network states, which:

- Simulates open-system Markovian dynamics, which can capture both transient and steady behavior.

- Also simulates finite temperature states

- Guarantees positivity of the variational ansatz at all times, overcoming previous limitations.

Methods Comparison

Quantum Jumps

- Excellent transient dynamics
- Challenging for highly mixed states

MPDO

- Excellent for steady states
- Can not determine positivity

LPTN

- Positive, efficient, and accurate in both regimes.
- Slightly more expensive