Tensor Network Rules for $(1+1)-D$ Quantum Cellular Automata

Uddhav Sen

March 7, 2025

1 Tensor Networks- An Introduction

One of the biggest obstacles to the theoretical and numerical study of quantum many-body systems is the curse of dimensionality, the exponential growth of the Hilbert space of quantum states[1]. In general this curse prevents efficient description of states, providing a significant complexity barrier to their study. Despite this, physically relevant states often possess additional structure not found in arbitrary states, and as such do not exhibit this pathological complexity, allowing them to be efficiently described and studied. Tensor networks have proven to be an important technique in studying condensed matter systems, with much of the modern theory and numerics used to study these systems involving tensor networks. In the numerical regime, tensor networks provide variational classes of states which can be efficiently described. For example, minimising the energy over one of these classes, one can learn a great deal about the low-energy spectrum some physical system of interest. The key variational classes are: matrix product states (MPS), projected entangled pair states (PEPS), and Multiscale Entanglement Renormalisation Ansatz (MERA)[2].

1.1 What is a Tensor?

We start with a pedagogical explanation of tensor networks. They are a mathematical framework used to represent and manipulate high-dimensional data efficiently. At the heart of tensor networks are *tensors*, which are generalizations of vectors and matrices to higher dimensions.

- **Scalars** (0D): A single number. Example: $a = 5$.
- **Vectors** (1D): A list of numbers. Example: $v = [v_1, v_2, \ldots, v_n]$.
- **Matrices** (2D): A table of numbers. Example: $M = [M_{ij}]$.
- **Tensors** (3D and beyond): A multi-dimensional array of numbers. Example: $T = [T_{ijk}]$.

A tensor can have more than two indices. A 3D tensor might look like T_{ijk}, where i, j, k refer to different dimensions.

> **Example**: A 3D tensor can represent a cube of data, where each axis has values, similar to how a matrix represents a 2D grid.

1.2 Why Tensor Networks?

As the dimensions of a tensor grow, the number of elements in it increases exponentially. For example, if each dimension has 10 values, a 3D tensor would have $10^3 = 1000$ elements, and a 5D tensor would have $10^5 = 100,000$ elements. This makes direct computation on high-dimensional tensors very expensive in both memory and processing. *Tensor networks* help reduce this complexity by breaking down large tensors into a network of smaller, interconnected tensors.

1.3 Basic Idea of a Tensor Network

A tensor network consists of **nodes** and **edges**:

- **Nodes** represent tensors.
- **Edges** (lines connecting the nodes) represent contracted indices between the tensors.

The edges encode how tensors are related to each other (via index contraction), much like matrix multiplication but generalized for tensors.

1.4 Example: Matrix Multiplication as a Tensor Network

Consider the multiplication of two matrices A and B:

- Matrix A has indices A_{ij}, and matrix B has indices B_{jk}.
- When we multiply them, we sum over the shared index j to get the product matrix $C_{ik} = \sum_j A_{ij}B_{jk}$.

In a tensor network, this would be represented by two nodes (for A and B), connected by an edge corresponding to the index j that gets contracted. The resulting product is a new tensor (or matrix in this case).

1.5 Graphical Representation

Tensor networks often use a *graphical notation*:

- **Nodes** are represented by circles or squares.
- **Edges** are lines between nodes, corresponding to indices.
- **Unconnected edges** are called *open legs*, representing free indices that aren't contracted.

2 Key Operations in Tensor Networks

2.1 Tensor Contraction

Tensor contraction is the core operation in tensor networks. It's analogous to summing over a shared index in matrix multiplication. When two tensors share an index, we "contract" over that index, reducing the number of indices in the resulting tensor. For example, contracting two tensors A_{ijk} and B_{klm} over the index k gives a new tensor C_{ijlm}.

2.2 Tensor Decomposition

Decomposing a large tensor into smaller ones helps simplify computations. One common decomposition is the *singular value decomposition (SVD)*, which can factorize a matrix into simpler parts. In tensor networks, we extend these ideas to higher dimensions, breaking large tensors into simpler, smaller pieces that are easier to handle.

3 Types of Tensor Networks

Different types of tensor networks are used depending on the problem and the structure of the data. Here are a few common types:

3.1 Matrix Product States (MPS)

- One of the simplest tensor networks.
- Useful for representing 1D systems, particularly in quantum physics (e.g., spin chains).
- Each tensor in the network is connected to its neighboring tensor, forming a chain or "string."

3.2 Tree Tensor Networks (TTN)

- A hierarchical structure resembling a tree.
- Used to capture multi-scale phenomena, where higher-level nodes capture large-scale correlations.

3.3 Projected Entangled Pair States (PEPS)

- A generalization of MPS for 2D systems.
- Suitable for representing higher-dimensional systems, where tensors are laid out on a grid, capturing interactions in multiple dimensions.

3.4 Tensor Train

- This is a machine learning-inspired structure, related to MPS, used for high-dimensional data like feature spaces.

4 Applications of Tensor Networks

Tensor networks have applications across various fields due to their ability to simplify complex systems:

- **Quantum Physics**: Tensor networks are used to efficiently simulate quantum systems, where representing the full quantum state can be exponentially large in terms of the number of particles.
- **Machine Learning**: Tensor networks can be used to compress and speed up deep learning models, especially when dealing with large datasets[3].
- **Optimization**: Tensor networks are used in numerical optimization, particularly in solving large-scale linear systems and tensor factorization problems.

5 Why Tensor Networks Work

The power of tensor networks lies in their ability to exploit the structure and correlations in data. Many real-world systems, especially physical systems, have low-dimensional structures hidden within their high-dimensional data. Tensor networks allow us to capture these patterns by breaking down the data into smaller, manageable components, leading to *computational efficiency* and *memory savings*. In quantum mechanics, the *entanglement* between particles means that many-body quantum states are inherently high-dimensional. Tensor networks like MPS or PEPS provide a compact way to represent these states without needing to store the entire exponentially large wavefunction. Tensor networks are a powerful and flexible tool for representing and manipulating high-dimensional data in an efficient manner. They provide a way to break down large, complex tensors into smaller parts, making otherwise intractable computations feasible. Their applications span from quantum physics to machine learning, offering solutions to problems that involve large-scale data and complex correlations.

- *Tensors* generalize vectors and matrices.
- *Tensor networks* break down high-dimensional tensors into simpler, interconnected pieces.
- *Tensor contraction* is the core operation, reducing the complexity of manipulating large tensors.
- Different types of tensor networks (MPS, TTN, PEPS) are used in different domains, from physics to machine learning.

Understanding tensor networks allows us to work with systems that would otherwise be computationally impossible to handle directly.

6 Matrix Product States

By studying the structure and properties of different classes of tensor networks, for example MPS, one can learn a great deal about the types of states which they can describe. Tensor network states therefore provide an important analytic framework for understanding the universal properties of classes of states which possess particular properties, such as those which only support certain entanglement or correlation structures. In addition to their application to many-body physics, tensor networks can also be used to understand many of the foundational results in quantum information. We start with MPS; where quantum correlations decay exponentially with increase in lattice size. Let $|\psi\rangle = \sum_{j_1 j_2 \ldots j_N=0}^{d-1} \psi_{j_1 j_2 \ldots j_N} |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_N\rangle$ be the completely general state of N qudits . The state is completely specified by knowledge of the rank-N tensor ψ. Repeated Singular Value Decompositions on the N-tensor can give rise to a set of N connected 3 tensors as shown in;

$$
\begin{aligned}
|\psi\rangle &= \sum_{j_1 j_2 \ldots j_N=0}^{d-1} \sum_{m_0, m_1, \ldots, m_N} M^{j_1}_{1; m_0, m_1} \ldots M^{j_N}_{N; m_{N-1}, m_N} |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_N\rangle \\
&= \sum_{\sigma_1 \sigma_2 \ldots \sigma_N=0}^{1} \sum_{m_0, m_1, \ldots, m_N} M^{\sigma_1}_{1; m_0, m_1} \ldots M^{\sigma_N}_{N; m_{N-1}, m_N} |\sigma_1 \sigma_2 \cdots \sigma_N\rangle
\end{aligned} \tag{1}
$$

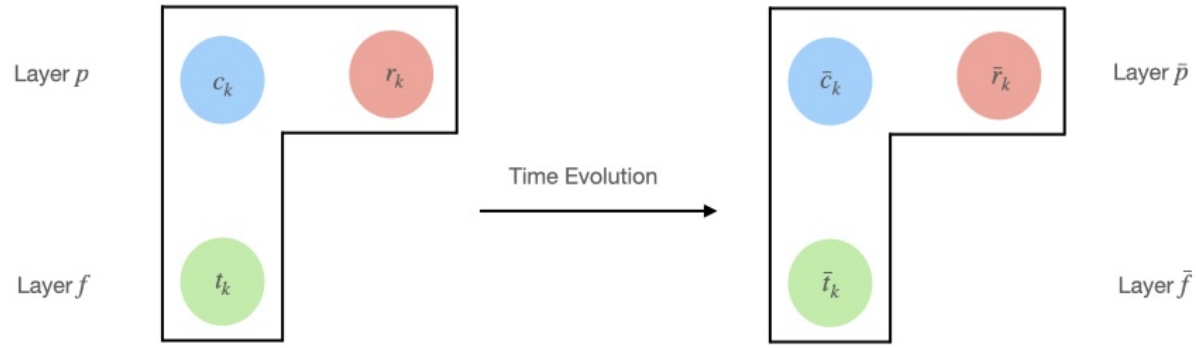

Figure 1: Our choice of perceptron is a 3−site lattice with control (c), right (r) and target (t) qubits.

We work with qubits; i.e. $d = 2, j \to \sigma$ and m_0 and m_N are 1-dimensional dummy indices introduced for consistency. The bonds dimensions denoted by m_j between lattice $(j, j+1)$ encodes quantum correlations primarily entanglement. For a specific set of local $\sigma_1, ..., \sigma_N, m_0, ..., m_N$ we use a single matrix M^{σ_j} per site. We hence evaluate a matrix product to give a single entry $\psi_{\sigma_1,...,\sigma_N}$ resulting in the name "matrix-product" states. We will use σ_j to index the local physical states on site j. Depending on the bond dimensions m_j of the tensors M_j different quantum states can be represented exactly. The required bond dimension $m = max_j(m_j)$ to represent a quantum system exactly grows exponentially with it's left/right partitions. We focus on lowly-entangled states which require small bond dimensions in $(1+1) - D$; allowing efficient representations of a quantum state.

6.1 Matrix Product Operators

We express every operator as an MPO like we represent quantum states; i.e. a contraction of N rank 4 tensors as;

$$\begin{aligned}\hat{O} &= \sum_{\sigma_1\sigma_2...\sigma_N} \sum_{\sigma'_1\sigma'_2...\sigma'_N} c_{\sigma_1\sigma_2...\sigma_N,\sigma'_1\sigma'_2...\sigma'_N} |\sigma_1\sigma_2...\sigma_N\rangle \langle\sigma'_1\sigma'_2...\sigma'_N| \\ &= \sum_{\sigma_1\sigma_2...\sigma_N} \sum_{\sigma'_1\sigma'_2...\sigma'_N} \sum_{w_0,w_1,...,w_N} W^{\sigma_1,\sigma'_1}_{1;w_0,w_1} \dots W^{\sigma_N,\sigma'_N}_{N;w_{N-1},w_N} |\sigma_1\sigma_2...\sigma_N\rangle \langle\sigma'_1\sigma'_2...\sigma'_N| \end{aligned} \tag{2}$$

We apply (2) to (1) as shown in () and the bond dimension increase with each iteration. It becomes impractical to represent wavefunctons exactly after some point and we consequently come up with various dimension trimming/truncation algorithms like Zip Up or Variational methods to keep the bond dimension in check along with the primary task of efficiently representing the wave function.

6.2 Mixed States

MPS by default can only represent pure quantum systems. We need to *purify* a system for effective representation with a MPS. It works by doubling the Hilbert Space $\mathcal{H} \to \mathcal{H}_p \otimes \mathcal{H}_a$ where a denotes the ancilla D.O.F. The density matrix we wish to represent is just the trace over the ancilla space of our purified quantum state $|\psi\rangle \in \mathcal{H}_p \otimes \mathcal{H}_a$; $\hat{\rho} = tr_a |\psi\rangle \langle\psi|$. So, in practice we take the MPS representation of $|\psi\rangle$ and contract the ancilla legs in the TN algorithm.

7 The Perceptron - A Brief Description

This section is aimed at creating a full and comprehensive review of implementing a generic $(1+1)-$ D Quantum Cellular Automata (QCA) algorithm using Matrix Product States (MPS) and Matrix Product Operators (MPO). The main operational idea is as follows; start with a many body chain in a chosen Hilbert Space; act some non/-unitary dynamics on it locally and then move on to the next lattice (generally nearest neighbour) and repeat. This leads to the idea of the perceptron and a local evolution operator being sufficient enough to explain the entire system.

We start this section by revealing our choice of the perceptron. We take two consecutive chains(where each lattice is a qubit) present p and future f at the current time-step and evolve it to layers $\bar{p}$ and $\bar{f}$ using the local evolution operator. The initial many body wave-function in the $p-$layer can be generally chosen;

$$|\psi_0\rangle = \sum_{\sigma_1\sigma_2...\sigma_L} \psi_{\sigma_1\sigma_2...\sigma_L} |\sigma_1\sigma_2...\sigma_L\rangle \tag{3}$$

We take $\psi_{\sigma_1\sigma_2...\sigma_L} = 1$ iff $\sigma_1 = 1, \sigma_{i>1} = 0$ and the rest are 0. This choice is for initial state to be all vacuum, $|\Omega\rangle\langle\Omega|$. We begin the $(1+1)-$D QCA algorithm by setting up the interactions in the system. This is a nearest neighbour interaction and localised to a 3 lattice for the construction of our present perceptron.

$$\mathcal{H} = -V_{ij}\sigma_i^z\sigma_j^z + B_j\sigma_j^x \tag{4}$$

We must note that the algorithm will work for any Hamiltonian as long as we have a nearest neighbour interaction with any choice of one body localised external fields. The evolution of the perceptron is governed by the $\mathcal{G} = \prod_k \mathcal{G}_k$ operator which acts on the whole lattice. We choose the lattice local components of $\mathcal{G}_k = S_k D_k U_k$ such that $[S_k, D_k] = [S_k, U_k] = [D_k, U_k] = 0$[4]. The two body fermionic swap operator is given as $S_k = \hat{n}_k \otimes \hat{n}_k + \hat{\bar{n}}_k \otimes \hat{\bar{n}}_k + \hat{\sigma}_k^+ \otimes \hat{\sigma}_k^- + \hat{\sigma}_k^- \otimes \hat{\sigma}_k^+$. The unitary and dissipation operators are represented as;

$$U_k = e^{-\iota dt h_{k,k+1}} \tag{5}$$

$$D_k = e^{-\iota\theta(L_k^\dagger \otimes \sigma_k^- + h.c.)} \tag{6}$$

Physically, the perceptron begins by correlating the current lattice with the adjacent lattice followed by dissipation with the same lattice in the next layer. We swap at the end to exchange information about the evolution to the future layer. Mathematically, the system dynamics looks like;

$$\rho(t+dt) = tr_{\bar{p}}(\mathcal{G}(\rho(t) \otimes |\Omega\rangle\langle\Omega|_{t+1})\mathcal{G}^\dagger) \tag{7}$$

The swap and unitary operators are obvious but there is justification for the choice of the dissipator. This is the most interesting part of the QCA algorithm as we can do a lot of things here(like learning and operator spreading). In Ref. the authors have shown that a particular QCA can be described by a simple many body Lindbladian with local particle decay. Here, we are interested to inflect this argument and study how generic (local) Lindbladians can be simulated in a QCA. First, we consider the master equation,

$$\frac{d\rho}{dt} = -i[\mathcal{H}, \rho] + \sum_{\nu=1}^{N} L_\nu \rho L_\nu^\dagger - \frac{1}{2}L_\nu^\dagger L_\nu \rho - \frac{1}{2}\rho L_\nu^\dagger L_\nu \tag{8}$$

and identify the corresponding Kraus operators. This is readily done by discretizing (8), viz.,

$$\begin{aligned}\rho(t+1) &= \rho(t) - i[\mathcal{H}, \rho(t)]\delta t + \delta t \sum_{\nu=1}^{N} L_\nu \rho L_\nu^\dagger - \frac{1}{2}L_\nu^\dagger L_\nu \rho - \frac{1}{2}\rho L_\nu^\dagger L_\nu \\ &\stackrel{!}{=} \sum_\mu M_\mu \rho(t) M_\mu^\dagger\end{aligned} \tag{9}$$

For $\mu \geq 1$ we may simply choose $M_\mu = \sqrt{\delta t} L_\mu$ and the remaining terms can be encompassed by an additional Kraus operator $M_0 = \hat{1} + \delta t G$. The choice $G = -i\mathcal{H} - \frac{1}{2}\sum_{\nu=1}^{N} L_\nu^\dagger L_\nu$ then reproduces the Lindblad master equation.

$$\begin{aligned}\sum_{\mu=0}^{N} M_\mu^\dagger M_\mu &= (\hat{1} + \delta t G^\dagger)(\hat{1} + \delta t G) + \delta t \sum_{\mu=1}^{N} L_\mu^\dagger L_\mu \\ &= \hat{1} + \delta t(G + G^\dagger) + \delta t \sum_{\mu=1}^{N} L_\mu^\dagger L_\mu + \mathcal{O}(\delta t^2) = \hat{1} + \mathcal{O}(\delta t^2)\end{aligned} \tag{10}$$

However, since the construction from Ref. already encodes the unitary time evolution and it is therefore useful to introduce the unitary time evolution operator $U_t = \prod_k \exp(-i\delta t h_{k,k+1})$. U_t is equivalent to first order to $\tilde{U}_t = \exp(-i\delta t\mathcal{H})$ with $\mathcal{H} = \sum_k h_{k,k+1} + v_k$. Hence,

$$\tilde{\rho}(t) = \tilde{U}_t \rho(t) \tilde{U}_t^\dagger = \rho(t) - i\delta t[\mathcal{H}, \rho(t)] + \mathcal{O}(\delta t^2) \tag{11}$$

Choose $K_0 = \hat{1} - \delta t/2 \sum_\nu L_\nu^\dagger L_\nu$ and $K_\nu = M_\nu$, then the claim is

$$\rho(t+1) = \sum_\mu K_\mu \tilde{\rho}(t) K_\mu^\dagger + \mathcal{O}(\delta t^2) \tag{12}$$

this coincides with results in Ref. Although, this is correct it is not exactly what we need. We need single-particle Kraus operators in order to generate a $k-$ site Dissipation operator D_k with a generic Lindbladian.

We consider a Kraus map

$$\varrho(t+1) = \sum_{j \in C} \mathcal{K}_j \varrho(t) \mathcal{K}_j^\dagger \tag{13}$$

where each Kraus operator can be written as product of local opertors, viz., $\mathcal{K}_j = \prod_j K_{m_j,j}$.

$$K_{\circ k} \simeq \hat{1} - \frac{\theta^2}{2} L_k^\dagger L_k, \quad K_{\bullet k} \simeq i\theta L_k, \quad C = \{\circ, \bullet\}^L \tag{14}$$

First, we show that the time evolution is of Lindblad form up to $\mathcal{O}(\theta^2)$. Up to second order in θ only the vacuum $m = \circ \ldots \circ$ and single excitations $m = \circ \ldots \circ \bullet \circ \ldots \circ$. We thus find

$$\begin{aligned} \varrho(t+1) &= \sum_{m \in C} \left(\prod_j K_{m_j,j} \right) \varrho(t) \left(\prod_{j'} K_{m_{j'},j'}^\dagger \right) \\ &\simeq \prod_{jj'} K_{\circ,j} \varrho(t) K_{\circ,j'}^\dagger + \sum_\ell \prod_{j,j'} K_{1_\ell,j} \varrho(t) K_{1_\ell,j'}^\dagger \end{aligned} \tag{15}$$

where we introduced the shorthand $1_\ell = \circ \ldots \circ \bullet \circ \ldots \circ$ with the excitation at site ℓ. The first term is straightforward to evaluate, neglecting higher order terms in θ. To evaluate the second term, we observe that $K_{\bullet j} = \mathcal{O}(\theta)$. Since this contribution is on the left and right of $\varrho(t)$ we need to replace $K_{\circ j} \simeq \hat{1}$ in the second term to only retain $\mathcal{O}(\theta^2)$. This yields the discretized Lindblad master equation

$$\varrho(t+1) = \varrho(t) - \theta^2 \sum_j \left(\frac{1}{2} L_j^\dagger L_j \varrho(t) + \sum_j \frac{1}{2} \varrho(t) L_j^\dagger L_j - L_j \varrho(t) L_j^\dagger \right) \tag{16}$$

Our aim is to determine an operator $D_{t,t+1} = \prod_{k=1}^L D_k$ such that

$$\langle m_j | D_j | \circ \rangle = K_{m_j,j} \tag{17}$$

This requires the operator D_j to be of the form

$$D_j = K_{\circ j} |\circ\rangle \langle\circ| + K_{\bullet j} |\bullet\rangle \langle\circ| + x_j |\circ\rangle \langle\bullet| + y_j |\bullet\rangle \langle\bullet| \tag{18}$$

Here, x_j and y_j are unspecified operators that de-facto do not contribute to the dynamics of the QCA as the next row is always in the vacuum state. However, to implement a QCA and simulate a one-dimensional open quantum system, one needs to engineer a Hamiltonian interaction between rows and thus full knowledge of D_j can help identifying a suitable Hermitian generator. From here, we drop the index j for clarity. We need to ensure that D is unitary, i.e.,

$$\begin{aligned} D^\dagger D = &\left[(\hat{1} - \frac{\theta^2}{2} L^\dagger L)^2 + \theta^2 L^\dagger L \right] |\circ\rangle \langle\circ| + \left[x^\dagger x + y^\dagger y \right] |\bullet\rangle \langle\bullet| \\ &+ \left[(\hat{1} - \frac{\theta^2}{2} L^\dagger L) x - i\theta L^\dagger y \right] |\circ\rangle \langle\bullet| + \text{h.c.} = \hat{1} \end{aligned} \tag{19}$$

First, we verify that the $|\circ\rangle \langle\circ|$ component is true to this relation up to $\mathcal{O}(\theta^2)$. This is readily verified by expanding the braket, viz.,

$$(\hat{1} - \frac{\theta^2}{2} L^\dagger L)^2 + \theta^2 L^\dagger L = \hat{1} + \mathcal{O}(\theta^2) \tag{20}$$

From the remaining components we obtain the equations

$$(\hat{1} - \frac{\theta^2}{2} L^\dagger L)x - i\theta L^\dagger y = 0$$
$$x^\dagger x + y^\dagger y = \hat{1} \tag{21}$$

It is important to note that the solutions to these equations may not be unique, however for our purposes of finding a physically realizable setup this fact is secondary. We expand

$$x = x_0 + \theta x_1 + \theta^2 x_2 + \mathcal{O}(\theta^3), \quad y = y_0 + \theta y_1 + \theta^2 y_2 + \mathcal{O}(\theta^3) \tag{22}$$

From (21) we obtain

$$x_0 + \theta\left(x_1 - iL^\dagger y_0\right) + \theta^2\left(x_2 - \frac{1}{2}x_0 L^\dagger L - iL^\dagger y_1\right) + \mathcal{O}(\theta^2) = 0 \tag{23}$$

and it is apparent that $x_0 = 0$. In order to distill the simplest solution, we admit that x is an odd function in θ and y even. This yields the ansatz $y_1 = 0$ and $x_2 = 0$ such that we readily obtain $x_1 = iL^\dagger y_0$. For a consistent $\theta \to 0$ limit we further admit $y_0 = \hat{1}$. Hence, $x_1 = iL^\dagger$ and the only operator to be determined is y_2. Eq.(21) yields

$$x^\dagger x + y^\dagger y \simeq \theta^2 x_1^\dagger x_1 + (\hat{1} + \theta^2 y_2^\dagger)(\hat{1} + \theta^2 y_2) + \mathcal{O}(\theta^2)$$
$$\simeq \hat{1} + \theta^2\left(LL^\dagger + y_2 + y_2^\dagger\right) + \mathcal{O}(\theta^3) \tag{24}$$

and we may choose $y_2 = -\frac{1}{2}LL^\dagger$. Hence,

$$D = \begin{pmatrix} \hat{1} - \frac{\theta^2}{2}L^\dagger L & i\theta L^\dagger \\ i\theta L & \hat{1} - \frac{\theta^2}{2}LL^\dagger \end{pmatrix}$$
$$= \left[\hat{1} - \frac{\theta^2}{2}L^\dagger L\right]|\circ\rangle\langle\circ| + i\theta L\,|\bullet\rangle\langle\circ| + i\theta L^\dagger\,|\circ\rangle\langle\bullet| + \left[\hat{1} - \frac{\theta^2}{2}LL^\dagger\right]|\bullet\rangle\langle\bullet| \tag{25}$$

To find the hermitian generator, we make the ansatz,

$$D = e^{i\theta\left(A\otimes\sigma^- + A^\dagger\otimes\sigma^+\right)} \simeq \hat{1} + i\theta\left(A\otimes\sigma^- + A^\dagger\otimes\sigma^+\right)$$
$$- \frac{\theta^2}{2}\left(A\otimes\sigma^- + A^\dagger\otimes\sigma^+\right)\left(A\otimes\sigma^- + A^\dagger\otimes\sigma^+\right) + \mathcal{O}(\theta^2)$$
$$= \hat{1} + i\theta\left(A\otimes\sigma^- + A^\dagger\otimes\sigma^+\right) - \frac{\theta^2}{2}\left(AA^\dagger\otimes(1-n) + A^\dagger A\otimes n\right) + \mathcal{O}(\theta^2)$$
$$= \left[\hat{1} - \frac{\theta^2}{2}AA^\dagger\right]|\circ\rangle\langle\circ| + i\theta A^\dagger\,|\bullet\rangle\langle\circ| + i\theta A\,|\circ\rangle\langle\bullet| + \left[\hat{1} - \frac{\theta^2}{2}A^\dagger A\right]|\bullet\rangle\langle\bullet| \tag{26}$$

Comparing Eqs. (25) and (26) then yields $A = L^\dagger$. Thus,

$$D_{t,t+1} = \prod_{k=1}^{L} e^{i\theta\left(L_k^\dagger\otimes\sigma_k^- + L_k\otimes\sigma_k^+\right)} = \exp\left(i\theta\sum_k L_k^\dagger\otimes\sigma_k^- + L_k\otimes\sigma_k^+\right) \tag{27}$$

We dealt with a local site Lindbladian operator in the previous section and derived the Dissipation Operator in (27). We aim to extend the definition for many body Lindbladians. We start with a simple choice of a two particle operator $L = \sigma_1^-\otimes\sigma_2^+$ and we will generalise it later. It is important to note that the subscripts denote the lattice positions. We start with the Lindbladian master equation;

$$\rho(t+1) = \rho(t) - i\;dt[H,\rho(t)] + dtL\,\rho(t)L^\dagger - \frac{dt}{2}\{L^\dagger L,\rho(t)\} + \mathcal{O}(dt^2) \tag{28}$$

When expressed in the Kraus representation;

$$\rho(t+1) = \sum_\mu K_\mu\tilde{\rho}(t)K_\mu^\dagger + \mathcal{O}(dt^2) \simeq K_0\tilde{\rho}(t)K_0^\dagger + K_1\tilde{\rho}(t)K_1^\dagger \tag{29}$$

$$K_0 = \hat{1} - \frac{dt}{2}L^\dagger L = \hat{1} - \frac{dt}{2}(\sigma_1^+\otimes\sigma_2^-)(\sigma_1^-\otimes\sigma_2^+) = \hat{1} - \frac{dt}{2}(n_1\otimes\tilde{n}_2) \tag{30}$$

$$K_1 = \sqrt{dt}L = \sqrt{dt}(\sigma_1^-\otimes\sigma_2^+) \tag{31}$$

We are interested in constructing the equivalent dissipation operator from the given two body dynamics. Since the locality is now effectively 2-body; we modify our basis for the choice of our $D_{t,t+1}$ to be $\{|\circ\circ\rangle, |\circ\bullet\rangle, |\bullet\circ\rangle, |\bullet\bullet\rangle\}$. We define D such that;

$$\langle\circ\circ| D |\circ\circ\rangle = \hat{1} - \frac{dt}{2}(n_1 \otimes \tilde{n}_2) \tag{32}$$

$$\langle\bullet\bullet| D |\circ\circ\rangle = \sqrt{dt}(\sigma_1^- \otimes \sigma_2^+) \tag{33}$$

Following the proof from [4], the generic QCA dynamics yields,

$$\begin{aligned}
\rho(t+1) &= \mathrm{tr}_{t+1}[(SDU)\rho(t) \otimes |\circ\circ\rangle \langle\circ\circ|_{t+1} \ (U^\dagger D^\dagger S)] \\
&= \mathrm{tr}_{t+1}[(DS)\tilde{\rho}(t) \otimes |\circ\circ\rangle \langle\circ\circ|_{t+1} \ (SD^\dagger)] \\
&= \mathrm{tr}_{t+1}[D |\circ\circ\rangle \langle\circ\circ|_t \otimes \tilde{\rho}(t)_{t+1} D^\dagger] \\
&= \sum_{m=\{\circ,\bullet\}^2} \langle m| D |\circ\circ\rangle_{t+1} \ \tilde{\rho}(t)_t \langle\circ\circ| D^\dagger |m\rangle_{t+1} \\
&= \langle\circ\circ| D |\circ\circ\rangle \ \tilde{\rho}(t) \langle\circ\circ| D^\dagger |\circ\circ\rangle + \langle\circ\bullet| D |\circ\circ\rangle \ \tilde{\rho}(t) \langle\circ\circ| D^\dagger |\circ\bullet\rangle + \\
&\quad \langle\bullet\circ| D |\circ\circ\rangle \ \tilde{\rho}(t) \langle\circ\circ| D^\dagger |\bullet\circ\rangle + \langle\bullet\bullet| D |\circ\circ\rangle \ \tilde{\rho}(t) \langle\circ\circ| D^\dagger |\bullet\bullet\rangle
\end{aligned} \tag{34}$$

We have condensed notation from the second last line for ease. To build our equivalence with the master equation; we define the two particle Kraus operators as $K_{m_1,m_2} = \langle m_1 \ m_2| D |\circ\circ\rangle$. This takes (34) to it's final form;

$$\begin{aligned}
\rho(t+1) &= K_{\circ\circ}\tilde{\rho}(t)K_{\circ\circ}^\dagger + K_{\circ\bullet}\tilde{\rho}(t)K_{\circ\bullet}^\dagger + K_{\bullet\circ}\tilde{\rho}(t)K_{\bullet\circ}^\dagger + K_{\bullet\bullet}\tilde{\rho}(t)K_{\bullet\bullet}^\dagger \\
&= \sum_{m=\{\circ,\bullet\}^2} K_{m_1,m_2}\tilde{\rho}(t)K_{m_1,m_2}^\dagger
\end{aligned} \tag{35}$$

This supplies our equivalence with the master equation (28). This readily gives us the two other Kraus operators $K_{\circ\bullet} = K_{\bullet\circ} = 0$ alongwith the obvious ones $K_0 = K_{\circ\circ}, K_1 = K_{\bullet\bullet}$. We claim to choose a $4-$site Dissipation operator $D = e^{(i\theta(\sigma_1^+\otimes\sigma_2^-)_t\otimes(\sigma_1^-\otimes\sigma_2^-)_{t+1} + (\sigma_1^-\otimes\sigma_2^+)_t\otimes(\sigma_1^+\otimes(\sigma_2^+)_{t+1})}$ with $\theta^2 = -dt$. to verify this choice, we start with a condensed notation by sucking up the tensor products and lattice/time components and expand in $\mathcal{O}(\theta^3)$;

$$\begin{aligned}
D &= e^{i\theta\left(\sigma^+\sigma^-\sigma^-\sigma^- + \sigma^-\sigma^+\sigma^+\sigma^+\right)} \\
&\simeq \hat{1} + i\theta(\sigma^+\sigma^-\sigma^-\sigma^- + \sigma^-\sigma^+\sigma^+\sigma^+) - \\
&\frac{\theta^2}{2}(\sigma^+\sigma^-\sigma^-\sigma^- + \sigma^-\sigma^+\sigma^+\sigma^+)(\sigma^+\sigma^-\sigma^-\sigma^- + \sigma^-\sigma^+\sigma^+\sigma^+) \\
&= \hat{1} + i\theta(\sigma^+\sigma^-\sigma^-\sigma^- + \sigma^-\sigma^+\sigma^+\sigma^+) - \frac{\theta^2}{2}(n\tilde{n}\tilde{n}\tilde{n} + \tilde{n}nnn)
\end{aligned} \tag{36}$$

Here we have $\tilde{n} = \hat{1} - n$ and we can easily verify that the choice of D gets us the desired Kraus operators.

8 Double Space Representation of the MPO

This section primarily focuses on explaining the motivation and the subsequent implementation of the Double Space Representation(denoted by $\mathcal{D}$) in the MPO and MPS respectively. This is essentially a non-unique isomorphic transformation in Fock Space where we convert the bras into kets. We start with the definition;

$$\begin{aligned}
\rho &= \sum_{n,m} \langle n| \rho |m\rangle |n\rangle \langle m| \\
|\rho\rangle\rangle &= \sum_{n,m} \langle n| \rho |m\rangle |n\rangle |m\rangle
\end{aligned} \tag{37}$$

This is the superoperator form of the density matrix and we should be mindful that it is not a quantum state. Let us start with looking at the trace operation in $\mathcal{D}$. We start with the identity as $|\mathbb{1}\rangle\rangle = \sum_n |n\rangle |n\rangle$. This leads to the definition of the inner product in $\mathcal{D}$ as well;

$$\begin{aligned}
\langle\langle\mathbb{1}|\rho\rangle\rangle &= \sum_{l,n,m} \langle l| \langle l|n\rangle |m\rangle \langle n| \rho |m\rangle \\
&= \sum_l \delta_{n,l}\delta_{m,l} \langle n| \rho |m\rangle = \sum_l \langle l| \rho |l\rangle = tr(\rho) = 1
\end{aligned} \tag{38}$$

The "norm" in $\mathcal{D}$ also leads to purity;

$$\begin{aligned}
\langle\langle\rho|\rho\rangle\rangle &= \sum_{n,m,n',m'} \langle n|\,\rho\,|m\rangle^*\,\langle n|\,\langle m|\,\langle n'|\,\rho\,|m'\rangle\,|n'\rangle\,|m'\rangle \\
&= \sum_{n,m,n',m'} \langle n|\,\rho\,|m\rangle^*\,\langle n'|\,\rho\,|m'\rangle\,\delta_{n',n}\delta m', m \\
&= \sum_{n,m} \langle n|\,\rho\,|m\rangle^*\,\langle n|\,\rho\,|m\rangle \quad (\rho = \rho^\dagger) \\
&= \sum n, m\,\langle m|\,\rho\,|n\rangle\,\langle n|\,\rho\,|m\rangle = \sum_m \langle m|\,\rho^2\,|m\rangle
\end{aligned} \tag{39}$$

This is the purity of an open quantum system. Now let us shift our focus to the real thing; system evolution for the $(1+1)-$ D QCA. From [4], we know the the density operator evolves as;

$$\tilde{\rho}(t+1) = tr_{\bar{p}}(\mathcal{G}(\rho(t)\otimes|0\rangle\,\langle 0|_f)\mathcal{G}^\dagger) \tag{40}$$

Let me take a step back and try to explain the immediate next step in the grander scheme of things. We start with Eq.(8) and see how our construction of $\mathcal{D}$ changes it. In the literature we call this isomorphism as the superoperator or vectorisation of the density matrix. Simply put, Eq.(37) vectorises a matrix. Effectively that is all that $\mathcal{D}$ does. The Master equation $\dot{\rho} = \mathcal{L}\rho$ vectorises in $\mathcal{D}$ as,

$$\begin{aligned}
|\dot{\rho}\rangle\rangle &= \mathcal{L}\,|\rho\rangle\rangle \\
|\rho(t)\rangle\rangle &= e^{\mathcal{L}t}\,|\rho(0)\rangle\rangle
\end{aligned} \tag{41}$$

We will follow a similar path of aiming to find the dynamical system chararcterised as $|\rho(t+1)\rangle\rangle = \Lambda\,|\rho(t)\rangle\rangle$ and construct the MPO of $\mathcal{G}$ in $\mathcal{D}$. For any element $|\alpha\rangle\rangle = |n\rangle\,|m\rangle\,|n'\rangle\,|m'\rangle$ in $\mathcal{D}$ with $|n'\rangle\,|m'\rangle$ as the new row and $n, m, n'm' = 0, 1, ..., 2^L - 1$.

$$\begin{aligned}
\langle\langle\alpha|\tilde{\rho}(t+1)\rangle\rangle &= \langle n|\,\langle n'|\,\tilde{\rho}(t+1)\,|m\rangle\,|m'\rangle \\
&= \langle n|\,\langle n'|\,\mathcal{G}(\rho(t)\otimes|0\rangle\,\langle 0|_f\,|m\rangle\,|m'\rangle \\
&= \sum_{k,l,k',l'} \langle n|\,\langle n'|\,\mathcal{G}\,|k\rangle\,|k'\rangle\,\langle k|\,\langle k'|\,\rho(t)\otimes|0\rangle\,\langle 0|_f\,|l\rangle\,|l'\rangle\,\langle l|\,\langle l'|\,\mathcal{G}^*\,|m\rangle\,|m'\rangle \\
&= \sum_\beta \langle n|\,\langle n'|\,\mathcal{G}\,|k\rangle\,|k'\rangle\,\langle l|\,\langle l'|\,\mathcal{G}^*\,|m\rangle\,|m'\rangle\,\langle\langle\beta|\rho(t)\otimes|0\rangle\,\langle 0|\rangle\rangle \\
&\text{(Set } |\beta\rangle\rangle = |k\rangle\,|l\rangle\,|k'\rangle\,|l'\rangle\text{)} \\
&= \sum_\beta \langle n\,n'|\,\mathcal{G}\,|k\,k'\rangle\,\langle m\,m'|\,\mathcal{G}\,|l\,l'\rangle^*\,\langle\langle\beta|\rho(t)\otimes|0\rangle\,\langle 0|\rangle\rangle \\
&= \sum_\beta \langle n\,m\,n'\,m'|\,\mathcal{G}\otimes\mathcal{G}^*\,|k\,l\,k'\,l'\rangle\,\langle\langle\beta|\rho(t)\otimes|0\rangle\,\langle 0|\rangle\rangle \\
&= \sum_\beta \langle\alpha|\,\mathcal{G}\otimes\mathcal{G}^*\,|k\,l\,k'\,l'\rangle\,\langle\langle\beta|\rho(t)\otimes|0\rangle\,\langle 0|\rangle\rangle \\
&= \langle\alpha|\,\mathcal{G}\otimes\mathcal{G}^*\,|\rho(t)\otimes|0\rangle\,\langle 0|\rangle\rangle
\end{aligned} \tag{42}$$

We expand the superoperator ket;

$$\begin{aligned}
|\rho(t)\otimes|0\rangle\,\langle 0|\rangle\rangle &= \sum_{n,m,n',m'} \langle n\,n'|\,\rho(t)\otimes|0\rangle\,\langle 0|m\,m'\rangle\,|n\,n'\rangle\rangle\,|m\,m'\rangle\rangle \\
&= \sum_{n,m,n',m'} \langle n|\,\rho(t)\,|m\rangle\,\langle n'|0\rangle\,\langle 0|m'\rangle\,|n\,n'\rangle\rangle\,|m\,m'\rangle\rangle \\
&= \sum_{n,m,n',m'} \langle n|\,\rho(t)\,|m\rangle\,\delta_{n',0}\delta_{m',0}\,|n\,n'\rangle\rangle\,|m\,m'\rangle\rangle \\
&= \sum_{n,m} \langle n|\,\rho(t)\,|m\rangle\,|n\,m\,0\,0\rangle\rangle
\end{aligned} \tag{43}$$

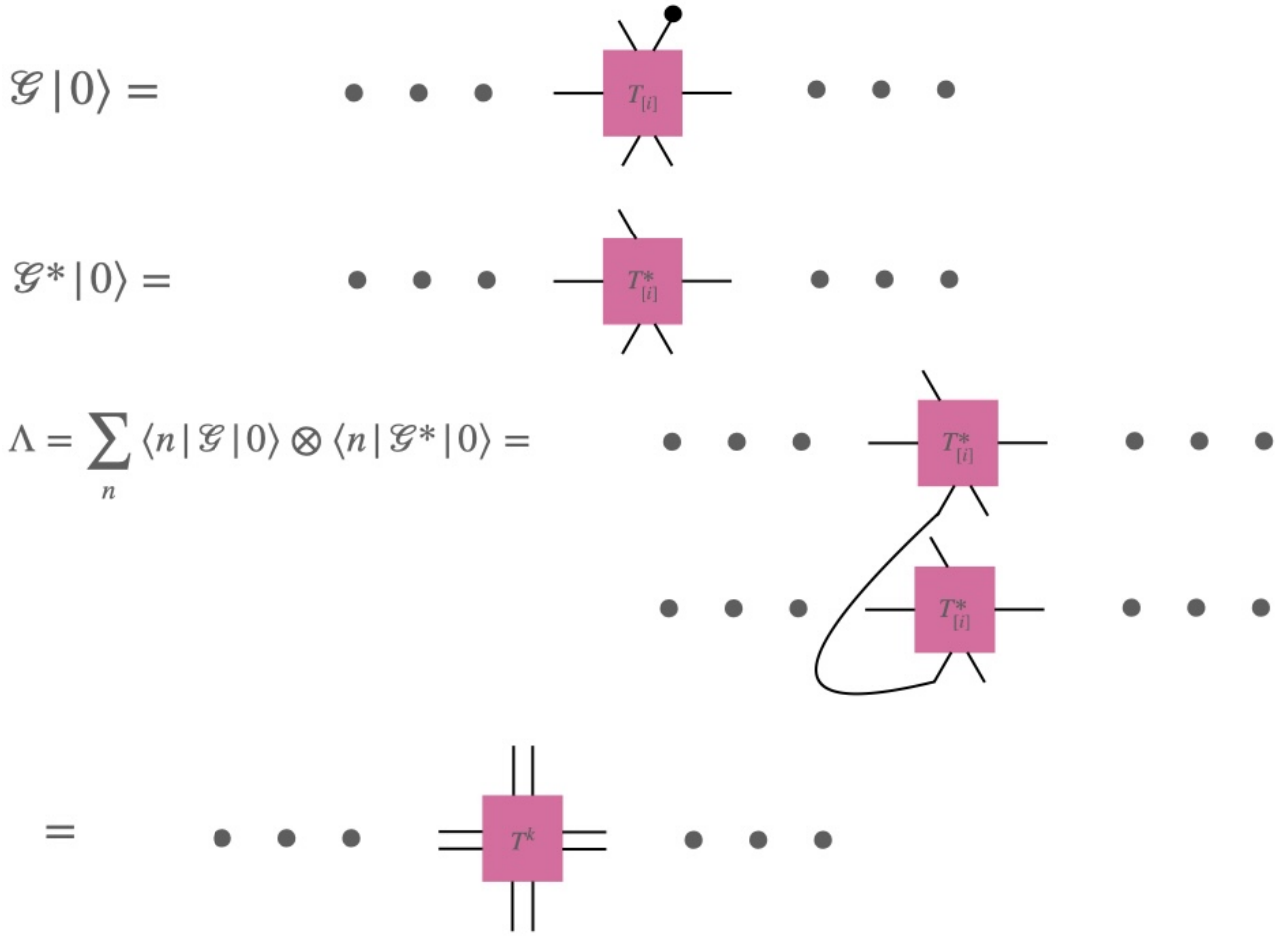

Figure 2: MPO representation of the dynamical map in $\mathcal{D}$. The tensor T^k represents all the possible choices in the perceptron (T_U, T_X, T_Y, T_V).

We substitute and expand;

$$\begin{aligned}
\langle\langle\alpha|\tilde{\rho}(t+1)\rangle\rangle &= \sum_{k,l} \langle n\,m\,n'\,m'|\,\mathcal{G}\otimes\mathcal{G}^*\,|k\,l\,0\,0\rangle\,\langle k|\,\rho(t)\,|l\rangle \\
&= \sum_{k,l} \langle n\,n'|\,\mathcal{G}\,|k\,0\rangle\,\langle m\,m'|\,\mathcal{G}^*\,|l\,0\rangle\,\langle k|\,\rho(t)\,|l\rangle \\
|\tilde{\rho}(t+1)\rangle\rangle &= \sum_{k,l} (\mathcal{G}\otimes\mathcal{G}^*)\,|k\,l\,0\,0\rangle\,\langle k|\,\rho(t)\,|l\rangle
\end{aligned} \tag{44}$$

We are interested in trying to find the Λ-map for the single lattice density operator $|\rho(t+1)\rangle\rangle$.

$$\begin{aligned}
|\tilde{\rho}(t+1)\rangle\rangle &= \langle\langle\mathbb{1}|\tilde{\rho}(t+1)\rangle\rangle \\
&= \sum_{n,k,l} \langle n|\,\langle n|\,(\mathcal{G}\otimes\mathcal{G}^*)\,|k\,l\,0\,0\rangle\,\langle k|\,\rho(t)\,|l\rangle \\
&= \sum_{n,k,l} \langle n|\,\mathcal{G}\,|k\rangle\,|0\rangle \otimes \langle n|\,\mathcal{G}^*\,|l\rangle\,|0\rangle\,\langle k|\,\rho(t)\,|l\rangle \\
&= \sum_{n} \langle n|\,\mathcal{G}\,|0\rangle \otimes \langle n|\,\mathcal{G}^*\,|0\rangle \sum_{k,l} |k\rangle\,\langle k|\,\rho(t)\,|l\rangle\,\langle l| \\
&= \sum_{n} \langle n|\,\mathcal{G}\,|0\rangle \otimes \langle n|\,\mathcal{G}^*\,|0\rangle\,\rho(t) \\
\implies \Lambda &= \sum_{n} \langle n|\,\mathcal{G}\,|0\rangle \otimes \langle n|\,\mathcal{G}^*\,|0\rangle
\end{aligned} \tag{45}$$

Fig.2 shows the MPO representation of Λ-map in $\mathcal{D}$.

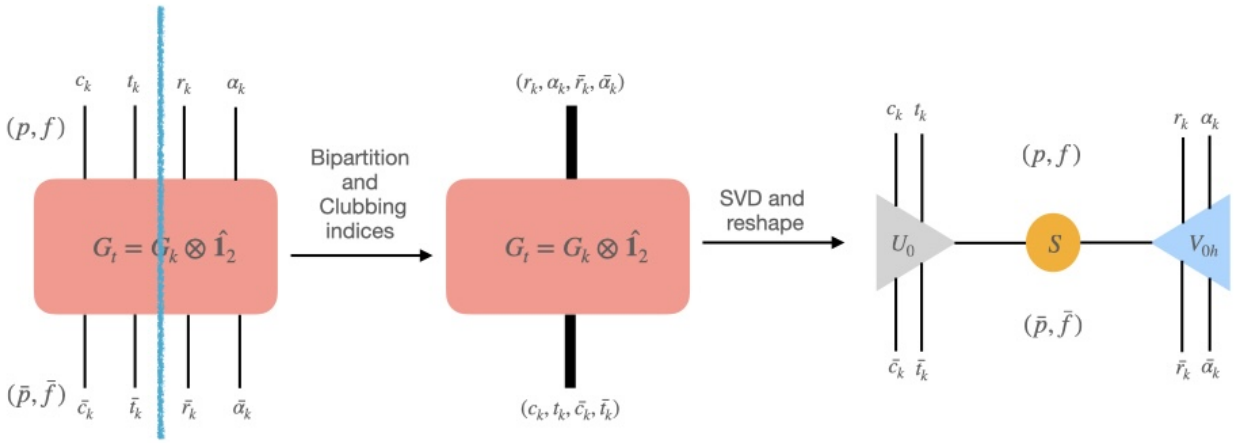

Figure 3: The blue line in the 8−tensor is the bipartition. The SVD localises the indices in our desired order.

9 Numerical Implementation of the (1+1)-D QCA

We singularly dedicate this section to explain the working mechanism of the QCA implementation with tensor networks including validation tests. We start with the algorithm in a block structure and slowly expand them with more detailed explanations.

9.1 The Full lattice MPO - $\mathcal{G}$

Algorithm 1 The Matrix Product Operator Algorithm.

Require: Hamiltonian $\mathcal{H}$, Single Particle Lindbladian $\mathcal{L}_k$.
Ensure: MPO boundary terms T_U, T_V and bulk terms T_X, T_Y.
1: Assign $\mathcal{H}$ and $\mathcal{L}_k$.
2: Calculate S_k, D_k, U_k.
3: $G_k \coloneqq S_k D_k U_k$. ▷ G_k is now a 4-body(16×16) matrix.
4: Reshape $dim(G_k) \rightarrow (2, 2, 2, 2, 2, 2, 2, 2)$. ▷ The legs are in and out qubits.
5: Transpose the legs s.t. the order is set to $(\bar{c}, \bar{t}, \bar{r}, \bar{\alpha}, c, t, r, \alpha)$.
6: Reshape $dim(G_k) \rightarrow (2^4 \times 2^4)$and perform SVD.
7: $G_k = USV^\dagger$.
8: Construct the bulk perceptron operators U, V.
9: Construct the MPO perceptron from the contraction diagrams.

After setting up the system(of our choice), we build the perceptron $G_k = S_k D_k U_k$ which is effectively a 2− tensor as all the indices are bunched up together. We should note that Qutip has arranged the qubits in our initial choice; control, target and right. We add the identity legs with G_k to symmetrise the input and output qubit indices after the SVD. The SVD decomposition takes the quantum object $G = G_k \otimes \mathbb{1}$ to an 8−dimensional numpy tensor. The transpose() function reorders the qubits in a way that we create a natural bipartition between the input and the output as shown in Fig 3. We reclub U_0 and V_{0h} inputs and outputs together to form U and V_h as it will help in buliding the localised lattice chain operator of the MPO. We soak in the diagonal S matrix into the left normalised U_0 tensor immediately. The next step is to construct the full lattice MPO $\mathcal{G}$ by placing the 2-site local MPOs(Fig.4) separable to each other for a single time step. This results in creating a perceptron like structure clearly visible in $\mathcal{G}$. Other than the two boundary terms $U = \mathcal{U}$ and $V_h = \mathcal{V}$; we get a repetitive $V_h - U$ and $U - V_h$ contraction to get $\mathcal{X}$ and $\mathcal{Y}$ respectively. The entire MPO thus has a self repeating structure in the bulk like $\mathcal{G} = \mathcal{U}\mathcal{X}\mathcal{Y}\mathcal{X}\mathcal{Y}...\mathcal{X}\mathcal{Y}\mathcal{V}$.

We have also included some sanity checks to verify the validity of the global MPO construction by force implementing $G_k \otimes \mathbb{1}$ tensors on top of each other to compare with the MPO.

9.2 MPO-MPS Contraction

This function is built to emulate time evolution (open dynamics) for the TN code.

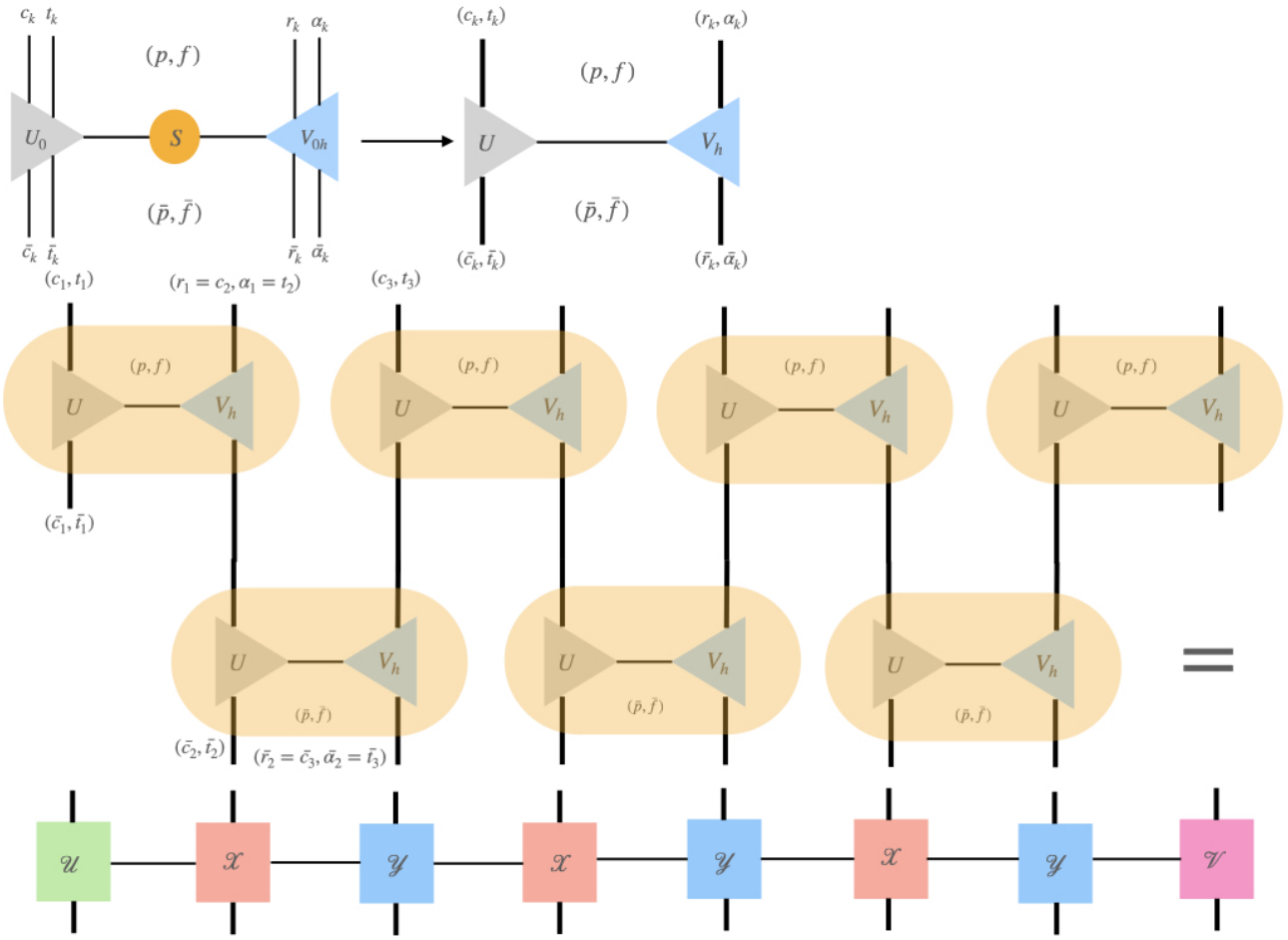

Figure 4: We soak in the Diagonal Matrix to the left and club the indices again to create the lattice MPO $\mathcal{G}$. Each of the yellow bubbles in the shown $L = 8$ case are the 2−site tensors from the first diagram.

Algorithm 2 MPO-MPS Contraction.

Require: MPO T_X, T_U, T_V, T_Y and MPS of $|\rho(t)\rangle\rangle$.
Ensure: Evolved MPS $|\rho(t+1)\rangle\rangle$.

$|\rho(t+1)\rangle\rangle_{\delta S=0} = T_X |\rho(t)\rangle\rangle_{\delta S=0}$.
$|\rho(t+1)\rangle\rangle_{\delta S=N-1} = T_Y |\rho(t)\rangle\rangle_{\delta S=N-1}$. ▷ These are the Boundary Terms.
for $k \to 1$ to $N-2$ **do**
    if k is odd **then**
        $|\rho(t+1)\rangle\rangle_k = T_U |\rho(t)\rangle\rangle_k$.
    else if k is even **then**
        $|\rho(t+1)\rangle\rangle_k = T_V |\rho(t)\rangle\rangle_k$.
    end if
end for

As typical in tensor network codes, we get a localised set of contractions that we can perform on the (MPO-MPS) system viz.($[T_U |1\rangle |1\rangle, T_X |0\rangle |0\rangle, T_Y |0\rangle |0\rangle,$
$T_V |0\rangle |0\rangle]$). We keep in mind that the contractions here are exact and we have not truncated ranks of any of the tensors. We keep that for a later section when we automate the code and run the simulation for multiple time steps and for a larger lattice.

9.3 Multipartite SVD Truncation

Algorithm 3 MPS Truncation.

Require: Analytical MPS $|\rho(t)\rangle\rangle$.
Ensure: Approximate MPS $|\rho(t)\rangle\rangle$.
1: **for** $k \to 1$ to $N-1$ **do**
2: Contract MPS of $(w(k,k+1) = |\rho(t)\rangle\rangle_k, |\rho(t)\rangle\rangle_{k+1})$ along the bond.
3: $w(k,k+1) = USV^\dagger$.
4: Reduce bond rank for $U(k)$.
5: Initialise truncated MPS for site k as $U(k)$.
6: $|\rho(t)\rangle\rangle_{k+1} \coloneqq SV^\dagger$.
7: **end for**

The analytical MPS received after one time evolution has an increase in bond dimension due to the MPO-MPS contraction. This becomes extremely impractical after a few time steps. An efficient way to bypass this issue is to take adjacent evolved MPS from sites $(k, k+1)$ and truncate their bond dimension after a Schmidt Decomposition. The function takes the analytical MPS as input and spits out an effective low rank MPS.

9.4 Automation Loop

We are ready to bring all the bits and pieces together now that we have the truncation and contraction functions. We start by constructing the initial density matrix $|\rho_0\rangle\rangle$ in $\mathcal{D}$. For our algorithm, we choose an initial $|\psi_0\rangle = |1000\rangle$ in a 4-lattice system. The primary motivation to choose 4 is to ensure that we cover the entire perceptron in our code. The rest of the lattice is self-repeating and computationally expensive; so we leave it to the HPC for later! The corresponding $|\rho_0\rangle\rangle = |\psi_0\rangle |\psi_0\rangle$ is obvious in $\mathcal{D}$. The algorithm can calculate expectation values throughout the simulation for any operators expressible as local transfer matrices; which is quite general. The

Algorithm 4 Simulation Loop.

Require: Initial MPS $|\rho(0)\rangle\rangle$, Time steps T, MPO perceptron T_U, T_X, T_Y, T_V,
Lattice size N, Observable $\hat{O} = \sum_{k=0}^{N-1} \hat{O}_k$.
Ensure: Evolved MPS $|\rho(t)\rangle\rangle$, Expectation value $\langle\hat{O}(t)\rangle$ $\forall t \in [0, T-1]$.
1: **for** $t \to 0$ to $T-1$ **do**
2: $c_{0,N-1} \coloneqq$ Contract MPS $|\rho(t)\rangle\rangle_{0,N-1}$ with the identity $\mathbb{1}_2$.
3: $d_{0,N-1} \coloneqq$ Contract MPS $|\rho(t)\rangle\rangle_{0,N-1}$ with MPO local operator $\hat{O}_{0,N-1}$.
4: **for** $n \to 1$ to $N-2$ **do**
5: Calculate c/d_n.
6: **end for**
7: Calculate the norm by contracting $\{c\}$ along the bond dimension.
8: Renormalise the truncated MPS. ▷ This is done to preserve the norm.
9: **for** $n \to 0$ to $N-1$ **do**
10: Calculate $\langle\hat{O}(t)\rangle += \frac{1}{N}$ ncon $(c_0 c_1 ... d_n ... c_{N-1})$. ▷ The ncon(network contractor) is along the bond dimension.
11: **end for**
12: $|\rho(t)\rangle\rangle \to (T_U |\rho(t)\rangle\rangle_0, T_X |\rho(t)\rangle\rangle_1, T_Y |\rho(t)\rangle\rangle_2, ..., T_V |\rho(t)\rangle\rangle_{N-1})$.
▷ MPO-MPS Contraction Function.
13: $|\rho(t)\rangle\rangle \to$ Truncate $|\rho(t)\rangle\rangle$.
14: **end for**

algorithm has been validated by reproducing plots from analytical calculations for small lattice sizes($N = 4$). The code has also matched TN simulations for the Quantum Contact process model from [4].

10 Numerical Validation for The QCA Algorithm

We show numerical plots for six different cases of the QCA simulation algorithm using MPO and MPS.

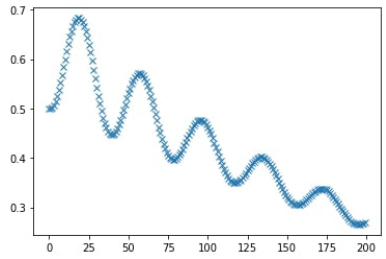

Figure 5: Quantum Contact Process $N = 4, H \neq 0$, Ancilla-V.S.

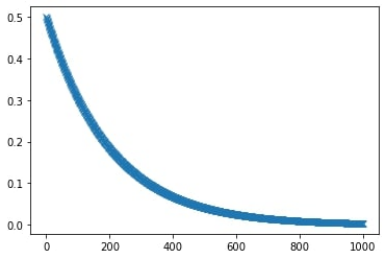

Figure 6: Quantum Contact Process $N = 4, H = 0$, Ancilla-V.S.

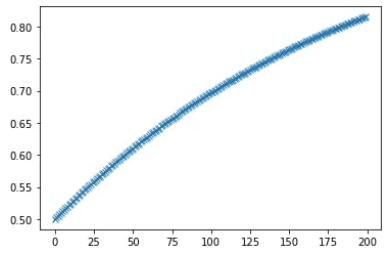

Figure 7: Quantum Contact Process $N = 4, H = 0$, Ancilla - O.S.

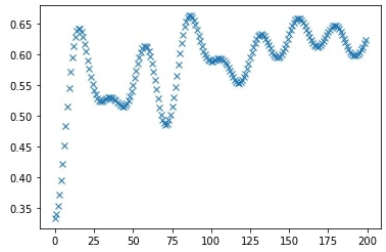

Figure 8: Quantum Contact Process $N = 6, H \neq 0$, Ancilla-O.S.

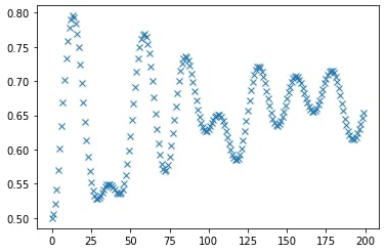

Figure 9: Quantum Contact Process $N = 6, H = 0$, Ancilla-O.S.

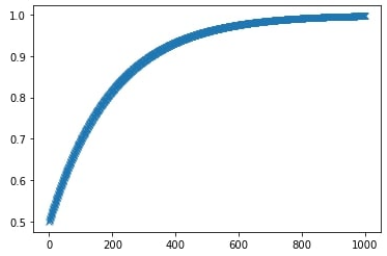

Figure 10: Quantum Contact Process $N = 6, H = 0$, Ancilla - O.S.

References

[1] Román Orús. "A practical introduction to tensor networks: Matrix product states and projected entangled pair states". In: *Annals of Physics* 349 (Oct. 2014), pp. 117–158. ISSN: 0003-4916. DOI: 10.1016/j.aop.2014.06.013. URL: http://dx.doi.org/10.1016/j.aop.2014.06.013.

[2] Jacob C Bridgeman and Christopher T Chubb. "Hand-waving and interpretive dance: an introductory course on tensor networks". In: *Journal of Physics A: Mathematical and Theoretical* 50.22 (May 2017), p. 223001. ISSN: 1751-8121. DOI: 10.1088/1751-8121/aa6dc3. URL: http://dx.doi.org/10.1088/1751-8121/aa6dc3.

[3] Sirui Lu et al. "Tensor networks and efficient descriptions of classical data". In: (2021). arXiv: 2103.06872 [quant-ph]. URL: https://arxiv.org/abs/2103.06872.

[4] Edward Gillman, Federico Carollo, and Igor Lesanovsky. "Using (1+ 1) D quantum cellular automata for exploring collective effects in large-scale quantum neural networks". In: *Physical Review E* 107.2 (2023), p. L022102.