



## PAPER

## Quantum compression of tensor network states

Ge Bai<sup>1,6</sup> , Yuxiang Yang<sup>2</sup> and Giulio Chiribella<sup>1,3,4,5</sup> <sup>1</sup> Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong, People's Republic of China<sup>2</sup> Institute for Theoretical Physics, ETH Zürich, 8093 Zürich, Switzerland<sup>3</sup> Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, United Kingdom<sup>4</sup> HKU Shenzhen Institute of Research and Innovation, Kejizhong 2nd Road, Shenzhen, People's Republic of China<sup>5</sup> Perimeter Institute For Theoretical Physics, 31 Caroline Street North, Waterloo N2L 2Y5, Ontario, Canada<sup>6</sup> Author to whom any correspondence should be addressed.E-mail: [baige@connect.hku.hk](mailto:baige@connect.hku.hk), [yangyu@phys.ethz.ch](mailto:yangyu@phys.ethz.ch) and [giulio@cs.hku.hk](mailto:giulio@cs.hku.hk)**Keywords:** quantum data compression, tensor networks, matrix product states, quantum machine learning, quantum many-body systems

## RECEIVED

19 December 2019

## REVISED

24 February 2020

## ACCEPTED FOR PUBLICATION

26 February 2020

## PUBLISHED

21 April 2020

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

**Abstract**

We design quantum compression algorithms for parametric families of tensor network states. We first establish an upper bound on the amount of memory needed to store an arbitrary state from a given state family. The bound is determined by the minimum cut of a suitable flow network, and is related to the flow of information from the manifold of parameters that specify the states to the physical systems in which the states are embodied. For given network topology and given edge dimensions, our upper bound is tight when all edge dimensions are powers of the same integer. When this condition is not met, the bound is optimal up to a multiplicative factor smaller than 1.585. We then provide a compression algorithm for general state families, and show that the algorithm runs in polynomial time for matrix product states.

**1. Introduction**

Quantum data compression [1, 2] is one of the pillars of quantum information theory. At the foundational level, it establishes the qubit as the basic unit of quantum information. At the more practical level, it provides a blueprint for the efficient transmission of quantum data in future quantum communication networks, with applications to distributed quantum computing [3] and quantum cloud computing [4].

The ultimate limit for compressing sequences of independently prepared quantum states was initially established in the pure state case [1] and later extended to mixed states [5–7]. Universal compression protocols for the scenario where the average state of each system is unknown, except for an upper bound on its von Neumann entropy, were provided in [8]. In recent years, there has been an interest in developing compression protocols for identically prepared systems [9–13]. Such systems occur in a wide range of tasks, including quantum tomography [14, 15], quantum cloning [16, 17], estimation [18, 19], and quantum machine learning [20]. Compression protocols for identically prepared systems have found applications in quantum metrology [21] and inspired new results in quantum state estimation [22]. An instance of compression for identically prepared systems was experimentally demonstrated in [23].

Most of the existing compression protocols assume that the input systems are in a product state. However, many relevant scenarios involve correlated systems, whose state cannot be expressed as a tensor product of single-system states. The ability to store correlated states into a smaller amount of quantum bits is important for the simulation of many-body quantum systems on small and medium-size quantum computers. For example, Kraus *et al* showed that  $\log n$  qubits are enough to simulate several families of  $n$ -qubit many-body states [24–26]. In particular, the result of [24] led to an experimental simulation of a 32-spin Ising chain using only 5 qubits [27]. In addition to quantum simulations, many-body states are relevant to quantum metrology, where they can serve as probes for unknown quantum processes [28, 29]. In this context, compression protocols for many-body states could be useful to transmit such probes from one location to another, or to store them in a quantum memory until further processing is required.

In this paper we address the compression of tensor network states, a broad class that includes cluster states [30, 31], matrix product states (MPSs) [32–34], projected entangled pair states (PEPS) [35, 36], tree tensor networks [37], and multi-scale entanglement renormalization ansatz (MERA) states [38].

First, we provide an efficiently computable upper bound of the number of qubits required to compress unknown states from a given parametric family of tensor network states. The upper bound can be interpreted as a bottleneck for the information flow from the parameters specifying the states to the physical systems in which the states are embodied. For the family of all tensor network states with given network topology and given edge dimensions, this upper bound is tight whenever all the edge dimensions are powers of the same integer. In general, the upper bound is tight up to a multiplicative factor of at most  $\log 3 \approx 1.585$ .

Second, we design a quantum algorithm that implements the compression protocol, and we show that the algorithm runs in polynomial time for families of MPSs. For more general state families, we provide sufficient conditions for the algorithm to run in polynomial time. Informally, the conditions express the fact that the linear span of the state family contains a ‘sufficiently dense’, yet polynomial-size set of states that can be efficiently prepared on a quantum computer.

One of the state families considered in our paper involves translationally invariant MPSs [33], hereafter abbreviated as TIMPS. We show that a completely unknown TIMPS of  $n$  identical systems with given bond dimension can be compressed without errors into a number of logical qubits growing at most as  $O(\log n)$ . Our result enables a compressed simulation of various models of many-body quantum states, such as the one-dimensional Ising model [39] and the AKLT model [40]. The logarithmic scaling of the total memory is optimal, as the set of TIMPSs includes the set of all identically prepared states, for which the optimal compression protocol is known to require  $\Omega(\log n)$  memory qubits, both for exact [11] and approximate compression protocols [12, 13]. The same result holds for higher dimensional lattices, and for a broader class of tensor network states for which the correlation tensors are site-independent: a generic site-independent  $n$ -particle state with a given bond dimension can be perfectly stored into  $O(\log n)$  logical qubits. We also consider tensor network states with the property that all tensors except those on the boundary are constant. For every subset of systems in the bulk, we show that the exact compression protocol satisfies an area law: the number of logical qubits used to compress the systems in the chosen subset is proportional to the size of its boundary.

This article is structured as follows. In section 2 we introduce the graphical notations for tensor networks. In section 3 we state our first result on the memory usage of exact compression of tensor network states and apply it to a case-wise study of tensor network state families in section 4. We extend our results from pure states to marginal and mixed states in section 5. Section 6 provides a compression protocol for MPSs with variable boundary conditions, which can be realized by logarithmic-depth circuits explicitly constructed from the description of the MPS. In section 7 we construct a quantum algorithm realizing compression protocols for general efficiently preparable states, and discuss its applicability to tensor network states. Finally, we conclude with discussions on how our results can provide bounds for coding theory in section 8.

## 2. Preliminaries

### 2.1. Compression of parametric state families

Consider a quantum system  $P$  with Hilbert space  $\mathcal{H}_P$ , and denote by  $S(\mathcal{H}_P)$  the set of density operators on  $\mathcal{H}_P$ . Let  $\{\rho_x\}_{x \in X} \subseteq S(\mathcal{H}_P)$  be a parametric family of quantum states, labeled by a parameter  $x$  in a given manifold  $X$ . For example,  $x$  could be a parameter that determines the Hamiltonian of the system, and  $\rho_x$  could be the ground state of the Hamiltonian parametrized by  $x$ .

Given a parametric family  $\{\rho_x\}_{x \in X}$ , the goal of compression is to store the states of the family into a quantum memory  $M$ , whose dimension is smaller than the dimension of the initial system  $P$ . A compression protocol for the states  $\{\rho_x\}_{x \in X}$  is specified by an encoding channel  $\mathcal{E}: S(\mathcal{H}_P) \rightarrow S(\mathcal{H}_M)$ , and by a decoding channel  $\mathcal{D}: S(\mathcal{H}_M) \rightarrow S(\mathcal{H}_P)$ , where  $\mathcal{H}_M$  denotes the Hilbert space of the quantum memory. Mathematically, the channels are described by completely positive trace-preserving linear maps. Both channels  $\mathcal{E}$  and  $\mathcal{D}$  are required to be independent of the parameter  $x$ , meaning that the compression operations must work ‘blindly’, without any knowledge of which state is being compressed.

In the following we will consider exact compression protocols, that is, protocols satisfying the condition

$$(\mathcal{D} \circ \mathcal{E})(\rho_x) = \rho_x \quad \forall x \in X \quad (1)$$

For pure state families, with  $\rho_x = |\Psi_x\rangle\langle\Psi_x|$  for all  $x \in X$ , the simplest compression protocols are defined by isometries  $V: \mathcal{H}_{\text{in}} \rightarrow \mathcal{H}_M$  from the input subspace  $\mathcal{H}_{\text{in}} := \text{Span}\{|\Psi_x\rangle\}_{x \in X} \subseteq \mathcal{H}_P$  to the memory space  $\mathcal{H}_M$ . An optimal compression protocol is a protocol that uses a memory system whose dimension is exactly  $d_M = \dim(\mathcal{H}_{\text{in}})$ . In this case, the isometry  $V: \mathcal{H}_{\text{in}} \rightarrow \mathcal{H}_M$  is actually a unitary.

In theory, constructing compression protocols for families of pure states is straightforward: one only needs to determine the input subspace  $\mathcal{H}_{\text{in}}$ , and to define an isometry  $V$  from  $\mathcal{H}_{\text{in}}$  to a memory space  $\mathcal{H}_M \simeq \mathcal{H}_{\text{in}}$ . In

practice, the efficiency of this construction is an issue. When the input system consists of many particles, constructing the isometry  $V$  may be computationally unfeasible, because it requires manipulations of exponentially long vectors. The situation is different when the subspace  $\mathcal{H}_{\text{in}}$  exhibits some specific structure that can be used to efficiently identify it and to construct the encoding operations. An example of this situation is the totally symmetric subspace  $\mathcal{H}_{\text{sym}} = \text{Span}\{|\phi\rangle^{\otimes n}|\phi\rangle \in \mathcal{H}_p\}$ , for which an efficient compression exists [9] and is based on the Schur transform [41, 42]. In this paper we will identify other scenarios in which the compression operations can be constructed efficiently, taking advantage of the tensor network structure.

### 2.2. Graphical notation for tensors

Here we introduce the graphical notation used in the rest of the paper. Our notation coincides, up to minor changes, with other notations used in the literature on tensor networks [43–45].

*Vectors and matrices.* A vector is represented as a box connected to an open edge. A column vector has an outgoing edge, while a row vector has an ingoing edge. A matrix is represented as a box with both an ingoing edge and an outgoing one. In the following examples,  $|v\rangle$  is a  $d$ -dimensional column vector,  $\langle v|$  is the adjoint of  $|v\rangle$ ,  $\langle \bar{v}|$  is the transpose of  $|v\rangle$ , and  $A$  is a  $d \times d$  matrix

$$|v\rangle = \leftarrow \boxed{v}, \quad \langle v| = \boxed{v} \leftarrow, \quad A = \leftarrow \boxed{A} \leftarrow \tag{2}$$

For a matrix, the place where an arrow is attached to the box matters. Here we assume that the left side of  $A$  corresponds to its row index, and the right side its column index. For a vector, the attachment position is unimportant, because the vector has only one index.

*Multiplication.* An edge connecting two tensors represents a summation over the corresponding index. With this notation, one can conveniently represent multiplications between matrices and vectors

$$AB|v\rangle = \sum_{i,j,k} A_{ij}B_{jk}v_k|i\rangle = \leftarrow^i \boxed{A} \leftarrow^j \boxed{B} \leftarrow^k \boxed{v} \tag{3}$$

The outgoing open edge indicates that the result of the multiplication is a column vector.

*Tensor product.* A tensor network with several disconnected components is a tensor product of the components (or an outer product of vectors)

$$A|v\rangle \otimes B|v\rangle = \leftarrow \boxed{A} \leftarrow \boxed{v} \quad \leftarrow \boxed{B} \leftarrow \boxed{v}, \quad A|v\rangle \langle v| A^\dagger = \leftarrow \boxed{A} \leftarrow \boxed{v} \quad \boxed{\bar{v}} \leftarrow \boxed{A^\dagger} \leftarrow = \leftarrow \boxed{A} \leftarrow \boxed{v} \quad \rightarrow \boxed{\bar{A}} \rightarrow \boxed{\bar{v}} \tag{4}$$

*Trace.* The trace of a matrix is represented by connecting its two indices:

$$\text{Tr}[A] = \boxed{A} \tag{5}$$

In general, a network with no open edges evaluates to a scalar.

*Higher-order tensors.* Higher order tensors can describe states and linear operations involving multiple systems. To represent them, one uses boxes with more than two edges. For example, the following graph represents an order-3 tensor  $T$ , where  $\{|i\rangle\}$ ,  $\{|j\rangle\}$ ,  $\{|k\rangle\}$  are orthonormal bases in their corresponding spaces

$$\leftarrow^j \boxed{T} \leftarrow^k \begin{matrix} \uparrow \\ i \end{matrix} = \sum_{i,j,k} T_{i,j,k} |j\rangle \langle k| \tag{6}$$

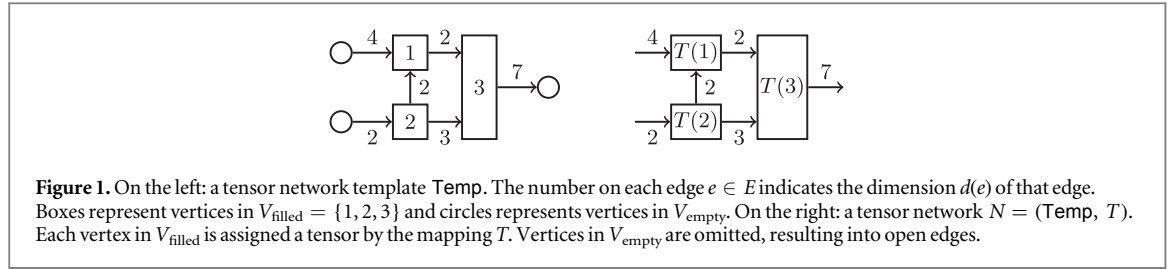
We label the edges by  $i$ ,  $j$  and  $k$  to indicate their correspondence to the first, second and third index of  $T$ , respectively. In the following, the indices will be sometimes omitted in the graphical notation.

*Reversal of edges.* Multiplication by the unnormalised maximally entangled state  $|I\rangle := \sum_i |i\rangle |i\rangle$  or its adjoint  $\langle\langle I| := \sum_i \langle i| \langle i|$  does not alter the elements of a tensor, but it converts a column index to a row index and vice versa. We represent a multiplication with  $|I\rangle$  by a reversal of the direction of the arrow involved in the multiplication

$$T|I\rangle = \leftarrow \boxed{T} \leftarrow \boxed{I} \rightarrow = \leftarrow^j \boxed{T} \leftarrow^k \begin{matrix} \uparrow \\ i \end{matrix} = \sum_{i,j,k} T_{i,j,k} |j\rangle |k\rangle \tag{7}$$

We always assume that the Hilbert space of each edge comes with a default basis, so that for each edge, the maximally entangled state is uniquely defined.

*Vectorization.* If we reverse all ingoing edges of a tensor, we obtain a tensor with only outgoing edges, which is a column vector on the tensor product of the Hilbert spaces corresponding to all the edges. For example



$$\leftarrow \boxed{B} \rightarrow = \leftarrow \boxed{B} \leftarrow \boxed{I} \rightarrow = (B \otimes I) |I\rangle = \sum_{ij} B_{ij} |i\rangle |j\rangle \quad (8)$$

In a tensor network, reversing non-open edges does not affect the values assigned to the whole network. For example,

$$\leftarrow \boxed{A} \leftarrow \boxed{B} \rightarrow |v\rangle = \leftarrow \boxed{A} \leftarrow \boxed{B} \leftarrow |v\rangle \quad (9)$$

for every  $A, B$  and  $|v\rangle$ .

### 2.3. Tensor networks

Informally, a tensor network is a set of tensors connected with each others. In the following we introduce a few formal definitions that will become useful later in the paper.

**Definition 1.** A *tensor network template* [46] is a triple  $\text{Temp} = (G, d, V_{\text{filled}})$ , where

- $G = (V, E)$  is an oriented<sup>7</sup> graph, with set of vertices  $V$  and set of edges  $E \subseteq V \times V$
- $d: E \rightarrow \mathbb{N}_+$  is a function that associates each edge  $e$  with an integer  $d(e)$ , called the *dimension of the edge*  $e$
- $V_{\text{filled}} \subset V$  is a subset of vertices, called the *filled vertices*, such that each vertex in  $V \setminus V_{\text{filled}}$  is adjacent to one and only one vertex, and that vertex is in  $V_{\text{filled}}$ .

A tensor network is obtained from a tensor network template by filling all vertices in  $V_{\text{filled}}$  with tensors:

**Definition 2.** A *tensor network* is a pair  $N = (\text{Temp}, T)$ , where  $\text{Temp}$  is a tensor network template, and  $T$  is a function mapping filled vertices  $v \in V_{\text{filled}}$  into tensors  $T(v)$ , with the order of the tensor  $T(v)$  equal to the number of edges incident on  $v$ . Each edge  $e$  incident on a filled vertex represents an index of the corresponding tensor, and the values of the index range from 1 to  $d(e)$ .

Graphically, we will represent a tensor network as a diagram where the filled vertices are represented by boxes, and the empty vertices  $V_{\text{empty}} := V \setminus V_{\text{filled}}$  are omitted. An illustration is shown in figure 1.

In the following, we will associate each edge  $e \in E$  with a Hilbert space  $\mathcal{H}_e$  of dimension  $d(e)$ . With this notation, the tensor network defines an operator from the total Hilbert space associated to the ingoing edges to the total Hilbert space associated to the outgoing ones:

**Definition 3.** Let  $E_{\text{in}} := \{(u, v) \in E: u \in V_{\text{empty}}, v \in V_{\text{filled}}\}$  ( $E_{\text{out}} := \{(u, v) \in E: u \in V_{\text{filled}}, v \in V_{\text{empty}}\}$ ) be the set of ingoing (outgoing) edges of a tensor network  $N$ , and let  $\mathcal{H}_{\text{in}} := \bigotimes_{e \in E_{\text{in}}} \mathcal{H}_e$  and  $\mathcal{H}_{\text{out}} := \bigotimes_{e \in E_{\text{out}}} \mathcal{H}_e$  be the corresponding Hilbert spaces. The *tensor network operator* of a tensor network  $N$  is a linear operator  $N_*: \mathcal{H}_{\text{in}} \rightarrow \mathcal{H}_{\text{out}}$  obtained from contracting the tensors  $\{T(v)\}_{v \in V_{\text{filled}}}$  according to the connections specified by the network template.

For example, the tensor network operator associated to the tensor network in figure 1 is

$$N_* = \sum_{i=1}^4 \sum_{j=1}^2 \sum_{k=1}^7 \left[ \sum_{l=1}^2 \sum_{m=1}^2 \sum_{n=1}^3 T(1)_{ilm} T(2)_{jln} T(3)_{mnk} \right] |k\rangle \langle i| \otimes \langle j| \quad (10)$$

When a given operator  $A$  arises from the contraction of tensors in a tensor network with template  $\text{Temp}$ , we say that the operator  $A$  is compatible with that template:

<sup>7</sup> We recall that an oriented graph is a directed graph in which no edge is bidirected, namely, for every two vertices  $u$  and  $v$ , at most one of the ordered pairs  $(u, v)$  and  $(v, u)$  is an edge in the graph.

**Definition 4.** An operator  $A: \bigotimes_{e \in E_{in}} \mathcal{H}_e \rightarrow \bigotimes_{e \in E_{out}} \mathcal{H}_e$  is *compatible* with the template  $\text{Temp}$  if there exists a tensor network  $N = (\text{Temp}, T)$  such that  $A$  is the tensor network operator associated to  $N$ , namely  $A = N_*$ .

**2.4. Pure tensor network states**

If a tensor network  $N$  has only outgoing edges but no ingoing ones, its operator  $N_*$  is a vector on the tensor product Hilbert space  $\mathcal{H}_{out}$ . A pure quantum state  $|\Psi\rangle \in \mathcal{H}_{out}$  is a *tensor network state* if its amplitudes are represented by a tensor network, namely,  $|\Psi\rangle = N_*$  for some tensor network  $N$  with no ingoing edges.

An example of tensor network states is provided by the MPSs [32–34]. MPSs can be used to represent the ground states of one-dimensional chains of particles with local interactions, including the one-dimensional Ising model [39] and the AKLT model [40]. In addition, many MPSs, including the AKLT state, are a resource for measurement-based quantum computation (MBQC) [47].

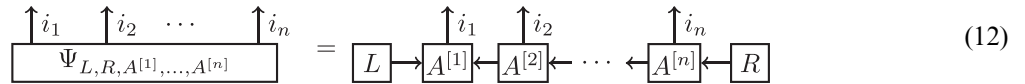
As an example, consider the class of MPSs with open boundary conditions [33] and assume for simplicity that every physical system in the MPS has the same dimension. Any such MPS is specified by

1.  $n$  physical systems, each of dimension  $d_p$
2. a correlation space  $\mathcal{H}_c$  of dimension  $d_c$ , called the bond dimension
3. a set of  $d_c \times d_c$  matrices  $\{A_i^{[k]}\}$ , labeled by an index  $k \in \{1, \dots, n\}$  and another index  $i \in \{1, \dots, d_p\}$
4. two vectors  $|L\rangle$  and  $|R\rangle$  in  $\mathcal{H}_c$ , called the boundary conditions.

Explicitly, the MPS is the  $d_p^n$ -dimensional vector

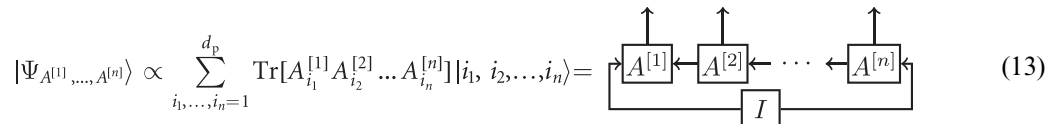
$$|\Psi_{L,R,A^{[1]}, \dots, A^{[n]}}\rangle = \sum_{i_1, \dots, i_n=1}^{d_p} \langle L | A_{i_1}^{[1]} A_{i_2}^{[2]} \dots A_{i_n}^{[n]} | R \rangle | i_1, i_2, \dots, i_n \rangle \tag{11}$$

We assume that the norms of the vectors  $|L\rangle$  and  $|R\rangle$  are chosen in such a way that the overall vector  $|\Psi_{L,R,A^{[1]}, \dots, A^{[n]}}\rangle$  is normalized. The MPS (11) is described by a tensor network of the following form:



where for each  $k$ , we regard the set  $\{A_i^{[k]}\}_{i=1}^{d_p}$  as an order-3 tensor of dimension  $d_p \times d_c \times d_c$ , denoted as  $A^{[k]}$ . The vertical arrows correspond to the physical systems, while horizontal ones correspond to the correlation spaces.

MPSs with periodic boundary conditions [33] are defined by replacing the boundary conditions with the maximally entangled state, as the following:

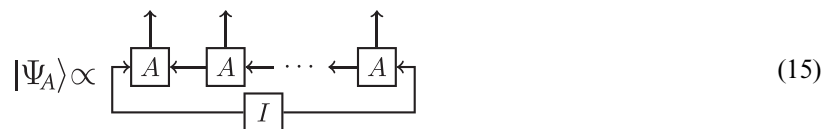


where  $|I\rangle := \sum_{i=1}^{d_c} |i\rangle |i\rangle$  represents the unnormalised maximally entangled state.

We will sometimes restrict our attention to SIMPSs [33], that is, MPSs where the matrices  $A_i^{[k]}$  are independent of  $k$ . Hence, the set of matrices will be simply denoted as  $\{A_i\}_{i=1}^{d_p}$ . In the site-independent case, equation (11) becomes

$$|\Psi_{L,R,A}\rangle = \sum_{i_1, \dots, i_n=1}^{d_p} \langle L | A_{i_1} A_{i_2} \dots A_{i_n} | R \rangle | i_1, i_2, \dots, i_n \rangle \tag{14}$$

SIMPSs with periodic boundary conditions (13) are called translationally invariant MPSs [33], and have the following form.



**3. Memory bound for the storage of tensor network states**

In this section we apply the framework of flow networks to bound the amount of memory qubits needed to compress a given family of tensor network states. We illustrate this approach for various families of tensor network states, including MPSs and PEPSs.

### 3.1. Parametric families of tensor network states

A parametric family of tensor network states is specified by a tensor network where some of the tensors depend on the values of the parameters. For example, consider the following family of MPSs with variable boundary conditions

$$\begin{array}{c} \uparrow \quad \uparrow \quad \dots \quad \uparrow \\ \boxed{\Psi_x} \end{array} = \boxed{L_x} \leftarrow \boxed{A} \leftarrow \boxed{A} \leftarrow \dots \leftarrow \boxed{A} \leftarrow \boxed{R_x} \quad (16)$$

for some parameter  $x \in X$  and some mappings  $L: x \mapsto |L_x\rangle$  and  $R: x \mapsto |R_x\rangle$ . Here, the vectors  $|L_x\rangle$  and  $|R_x\rangle$  are variable, while the tensor  $A$  is fixed.

Using vectorization, the tensor network can be rearranged in such a way that all the variable tensors are column vectors. The tensor product Hilbert space associated to the edges of the variable tensors will be called the *parameter Hilbert space* and will be denoted by  $\mathcal{H}_X$ . The tensor product Hilbert space associated to all the physical systems in the network will be called the *physical Hilbert space* and will be denoted by  $\mathcal{H}_P$ .

With the above notation, every parametric family of tensor network states can be represented as

$$|\Psi_x\rangle = N_* |v_x\rangle, \quad (17)$$

where  $|v_x\rangle$  is a vector in  $\mathcal{H}_X$ ,  $N$  is the tensor network consisting of the fixed tensors, and  $N_*: \mathcal{H}_X \rightarrow \mathcal{H}_P$  is the tensor network operator associated to  $N$ . An example of this parametrisation is provided in the following,

$$\boxed{\Psi_x} \rightarrow = \begin{array}{c} \boxed{v_x} \rightarrow \begin{array}{c} \boxed{A} \\ \boxed{B} \end{array} \rightarrow \boxed{C} \end{array} \quad (18)$$

N

where the operator  $N_*$  arises from the tensor network  $N$  inside the dashed frame.

It is clear from equation (17) that the dimension of the input subspace  $\mathcal{H}_{in} = \text{Span}\{|\Psi_x\rangle\}$  is upper bounded by the rank of  $N_*$ . Hence, the (logarithm of the) rank of  $N_*$  provides an upper bound on the number of qubits needed for the optimal compression. However, the matrix  $N_*$  may generally have an exponentially large number of columns and rows, and its rank may not be efficiently computable. One way to address this problem is to search for efficiently computable upper bounds on the rank of  $N_*$ , by inspecting the internal structure of the tensor network  $N$ . In the following subsection we will exploit a connection between tensor networks and flow networks to construct useful bounds on the rank of  $N_*$ , and therefore, on the number of qubits needed for compression.

### 3.2. Flow networks and memory bounds

We now provide a construction that associates tensor network templates with flow networks, and provide a memory bound valid for all families of tensor network states with a given template.

A flow network [48]  $\tilde{N} = (G, c, s, t)$  consists of a directed graph  $G = (V, E)$ , with set of vertices  $V$  and set of edges  $E$ , a function  $c: E \rightarrow \mathbb{R}_{0+} := \{x \in \mathbb{R} | x \geq 0\}$ , associating each edge  $e \in E$  with a non-negative number  $c(e)$ , called its capacity, and two distinguished vertices,  $s$  and  $t$ , called the source and the sink, respectively.

A flow network can be intuitively understood as a pipe system with edges being pipes and vertices being junctions. Fluid enters in the pipe system from the source  $s$  and exits at the sink  $t$ . At any time, the flux in each pipe is no more than the capacity of the pipe, while at each junction, the total amount of fluid is conserved, meaning that the total flux going into the junction equals to the total flux going out. This idea is captured by the mathematical notion of flow. A *flow*  $f: E \rightarrow \mathbb{R}_{0+}$  in a flow network is an assignment of non-negative numbers to the edges of the network, subject to the following conditions.

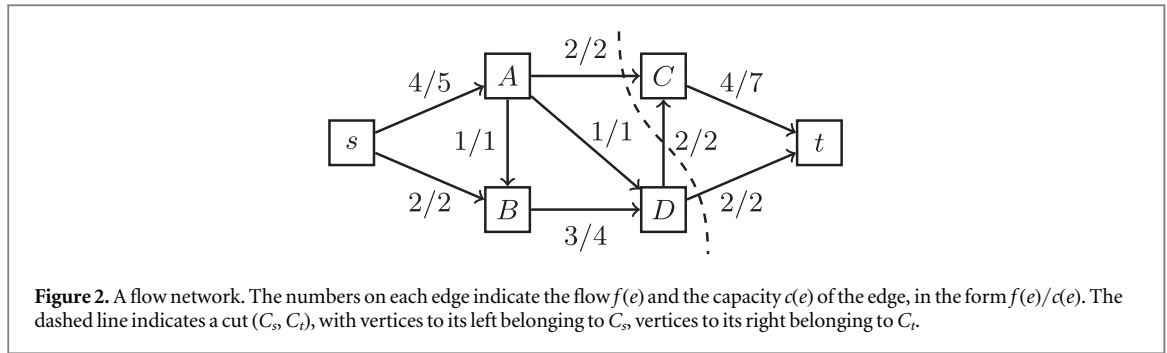
1. For every edge  $e$ , the flow is upper bounded by the capacity, namely  $f(e) \leq c(e)$ .
2. For every vertex  $v$  other than the source and the sink, the total flow entering in the vertex  $v$  is equal to the total flow exiting from it, namely  $\sum_{u \in V, (u,v) \in E} f(u, v) = \sum_{u \in V, (v,u) \in E} f(v, u)$ .

In the pipe system analogy,  $f(e)$  represents the amount of fluid that is flowing through the pipe  $e$ .

The *value of the flow*  $f$ , denoted by  $f_s$ , is the total flow exiting from the source, namely

$$f_s := \sum_{v \in V, (s,v) \in E} f(s, v) \quad (19)$$





The maximum of  $f_s$  over all possible flows is called the *max-flow* of the network, and is denoted as

$$\text{max-flow}(\tilde{N}) := \max_f f_s \quad (20)$$

Intuitively,  $f_s$  represents the maximum amount of fluid that can enter into a pipe system. This amount can be upper bound by considering that all the fluid entering from the source has to exit from the sink, and, in order to do so, it has to pass through the pipes between the source and the sink. A *cut* of the flow network is a cross-section of the pipe system that separates the source and the sink. Formally, a cut is a partition of the vertices into two disjoint subsets  $C_s \subset V$  and  $C_t \subset V$ , with  $s \in C_s$  and  $t \in C_t$ . We write the cut as  $C = (C_s, C_t)$ . An edge  $(u, v)$  is called a *cut edge* if  $u$  belongs to  $C_s$  and  $v$  belongs to  $C_t$ .

Since all the fluid entering from the source has to pass through the cut  $C$ , the total capacity of the pipes associated to cut edges poses an upper bound to the flux. Explicitly, the *capacity of the cut*  $(C_s, C_t)$ , denoted as  $c(C_s, C_t)$  is the sum of the capacities of the cut edges, namely

$$c(C_s, C_t) := \sum_{u \in C_s, v \in C_t, (u,v) \in E} c(u, v) \quad (21)$$

and one has the upper bound

$$\text{max-flow}(\tilde{N}) \leq c(C_s, C_t) \quad (22)$$

for every possible cut  $(C_s, C_t)$ . The best bound is obtained by choosing the cut with minimum capacity. The minimum of the capacity  $c(C_s, C_t)$  over all possible cuts  $(C_s, C_t)$  is called the *min-cut*, and is denoted by

$$\text{min-cut}(\tilde{N}) := \min_{(C_s, C_t)} c(C_s, C_t) \quad (23)$$

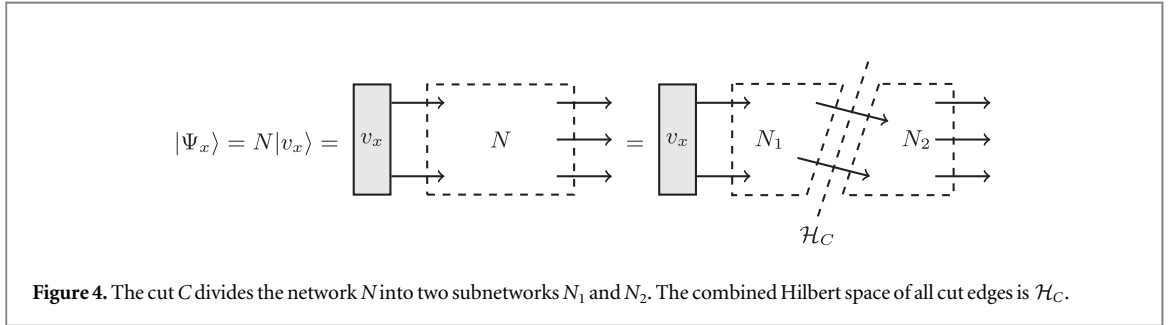
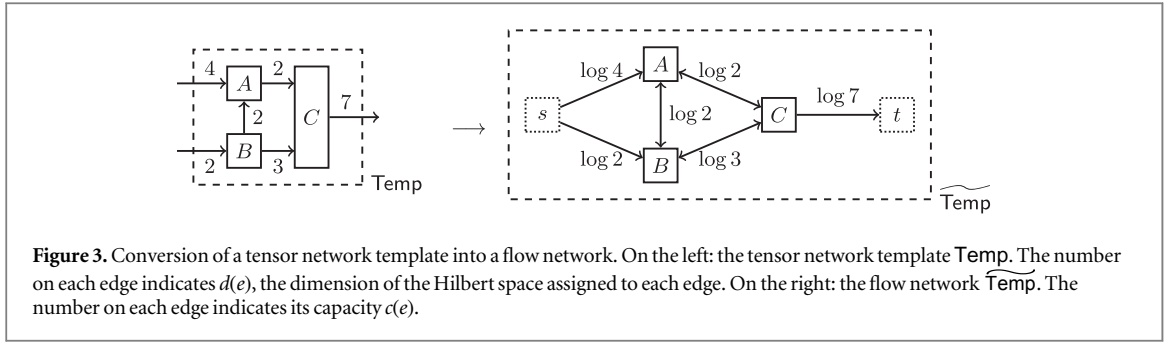
The max-flow min-cut theorem states that  $\text{max-flow}(\tilde{N}) = \text{min-cut}(\tilde{N})$  [48]. Intuitively, this shows that the maximum flux a pipe system can carry from  $s$  to  $t$  is *exactly* equal to the capacity of the minimal cross-section of the pipes.

An example of a flow network is shown in figure 2.

The analogy of the pipe system will be useful to understand the intuitive content of our results, where we use flow networks to model the ‘flow of information’, rather than the flow of a material fluid. We imagine information flowing into the ingoing edges and out of the outgoing edges of the tensor network, and each edge has a capacity equal to the maximum amount of information it can carry, which is  $\log d(e)$  qubits for an edge  $e$  with dimension  $d(e)$ . Given a tensor network template, we relate it to a flow network that can be constructed in the following way.

**Construction 1.** Let  $\text{Temp} = (G, d, V_{\text{filled}})$  be a tensor network template, and let  $V_{\text{empty}} = V \setminus V_{\text{filled}}$  the set of empty vertices. The flow network associated to the template  $\text{Temp}$ , denoted by  $\widetilde{\text{Temp}} = (\tilde{G}, c, s, t)$ , is constructed through the following prescriptions.

1. Add the vertices  $s$  and  $t$  to  $V$ .
2. Replace each ingoing edge  $(u, v)$  (with  $u \in V_{\text{empty}}$  and  $v \in V_{\text{filled}}$ ) with an edge  $(s, v)$ . Define the capacity of the edge  $(s, v)$  as  $c(s, v) := \log d(u, v)$ .
3. Replace each outgoing edge  $(v, u)$  (with  $v \in V_{\text{filled}}$  and  $u \in V_{\text{empty}}$ ) with an edge  $(v, t)$ . Define the capacity of the edge  $(v, t)$  as  $c(v, t) := \log d(v, u)$ .
4. Remove all the vertices in  $V_{\text{empty}}$ .



- For every internal edge  $(v, w)$  (with both  $v$  and  $w$  in  $V_{\text{filled}}$ ) include also  $(w, v)$  in the set of edges. Define the capacities  $c(v, w) = c(w, v) = \log d(v, w)$ .

An illustration of the above procedure is provided in figure 3.

Consider the tensor network  $N = (\text{Temp}, T)$  associated to the fixed tensors in the given state family  $\{|\Psi_x\rangle\}$ , and let  $\widetilde{\text{Temp}}$  be the flow associated to the template  $\text{Temp}$  via construction 1. Every cut in the flow network  $\widetilde{\text{Temp}}$  defines a factorization of the operator  $N_*$  as  $N_* = N_{2*}N_{1*}$ , where  $N_{1*}$  and  $N_{2*}$  are the operators of the two subnetworks  $N_1$  and  $N_2$  on the two sides of the cut, as illustrated in figure 4. Then, one has the bound

$$\text{rank}(N_*) \leq \min\{\text{rank}(N_{1*}), \text{rank}(N_{2*})\} \leq d_C \tag{24}$$

where  $d_C$  is the dimension of the Hilbert space  $\mathcal{H}_C$  associated to the edges in the cut. Hence,  $\lceil \log(d_C) \rceil$  qubits are sufficient to compress the state family. Recalling that the logarithm of  $d_C$  is the capacity of the cut, we obtain the following:

**Theorem 1.** For every cut  $C = (C_s, C_t)$  of  $\widetilde{\text{Temp}}$ , the state family  $\{|\Psi_x\rangle\}_{x \in X}$  can be compressed without errors into  $\lceil c(C) \rceil$  memory qubits. In particular, the state family can be compressed into

$$Q_{\text{cut}} = \lceil \text{min-cut}(\widetilde{\text{Temp}}) \rceil \tag{25}$$

memory qubits, where  $\text{min-cut}(\widetilde{\text{Temp}}) = \min_C \log d_C$  is the minimum cut of the flow network  $\widetilde{\text{Temp}}$ .

Intuitively, theorem 1 tells us that, since the maximum amount of information that can flow in the network is upper bounded by the capacity of the min-cut, the amount of memory needed to store this information is also upper bounded by the capacity of the min-cut. The point of theorem 1 is that, while the calculation of  $\text{rank}(N_*)$  may not be computationally feasible, the minimum cut can be found efficiently using known algorithms such as the relabel-to-front algorithm [48], which runs in  $O(|V|^3)$  time.

In section 4, we will provide explicit examples of minimum cuts for some relevant families of tensor network states. Before that, we will discuss the optimality of  $Q_{\text{cut}}$  as an upper bound on the number of memory qubits needed for compression.

### 3.3. Optimality for fixed tensor network templates

The amount of memory used by the best compression protocol is intuitively related to the flow of quantum information from the parameters specifying the quantum state to the physical systems in which the states are embodied. To make this intuition precise, suppose that we want to compress a known, but otherwise generic family of tensor network states with network template  $\text{Temp}$ , that is, a family of the form  $\{N_*|\nu_x\rangle\}_{x \in X}$ , where  $N_*$  is a tensor network operator compatible with the template  $\text{Temp}$ , and  $\{|\nu_x\rangle\}$  is a generic set of (suitably normalized) vectors in the parameter space  $\mathcal{H}_X$ . In the worst case over  $\{|\nu_x\rangle\}$  and  $N_*$ , it is easy to see that the



minimum number of memory qubits necessary for compression is  $\lceil \log \text{rank}(N_*) \rceil$ . Indeed, the vectors  $\{|v_x\rangle\}$  could form a spanning set for the parameter space  $\mathcal{H}_X$ , so that the dimension of the input space  $\mathcal{H}_{\text{in}} = \text{Span}\{N_*|v_x\rangle\}_{x \in X}$  is exactly equal to the rank of  $N_*$ . It is then immediate to conclude that every exact compression protocol will require at least  $\lceil \log \text{rank}(N_*) \rceil$  memory qubits. Taking the worst case over  $N_*$ , we obtain the following

**Proposition 1.** *The minimum number of memory qubits required for the exact compression of a generic state family of tensor network states with template  $\widetilde{\text{Temp}}$  is*

$$Q_{\min} = \lceil \text{quantum-max-flow}(\widetilde{\text{Temp}}) \rceil \quad (26)$$

where  $\text{quantum-max-flow}(\widetilde{\text{Temp}})$  is the quantum max-flow [46], defined as

$$\text{quantum-max-flow}(\widetilde{\text{Temp}}) := \max_{N_*} \log \text{rank}(N_*) \quad (27)$$

the maximum being over all tensor network operators  $N_*$  compatible with the template  $\widetilde{\text{Temp}}$ <sup>8</sup>.

Now, an important question is whether the compression protocols of theorem 1 can reach the minimum number of qubits (26), in the worst case over all state families compatible with a given network template. In other words, the question is whether the equality  $\lceil \text{quantum-max-flow}(\widetilde{\text{Temp}}) \rceil = \lceil \text{min-cut}(\widetilde{\text{Temp}}) \rceil$  holds. Such equality would follow from a quantum version of the max-flow min-cut theorem [49], which would state the equality  $\text{quantum-max-flow}(\widetilde{\text{Temp}}) = \text{min-cut}(\widetilde{\text{Temp}})$ . Remarkably, [46] shows that such quantum version does not always hold, and in general the quantum max-flow is only a lower bound on the min-cut

$$\text{quantum-max-flow}(\widetilde{\text{Temp}}) \leq \text{min-cut}(\widetilde{\text{Temp}}) \quad (28)$$

Nevertheless, the equality holds in the case where all dimensions are powers of the same integer [46]. In this case, the validity of the quantum max-flow-min-max theorem implies the following optimality property:

**Proposition 2.** *Let  $\widetilde{\text{Temp}}$  be a network template with  $d(e) = b^{n(e)}$  for some fixed integer  $b$  and for some integer-valued function  $n: E \rightarrow \mathbb{N}$ . Then,  $Q_{\text{cut}} = Q_{\min}$ , meaning that the number of qubits used in theorem 1 is minimum in the worst case over all state families with the given network template.*

Proposition 2 guarantees that, under the assumption that each dimension  $d(e)$  is an integer power of  $b$ , the amount of qubits used in theorem 1 is optimal for the least compressible family of tensor network states compatible with the given template. In the general case, we show that the amount of qubit used by the compression protocol of theorem 1 is at most  $\log 3$  times the minimum number needed for compression. This result, provided in the following proposition, is based on a general relation between the min-cut and the quantum-max-flow of a generic tensor network:

**Proposition 3.** *For every network template  $\widetilde{\text{Temp}} = (G, d, V_{\text{filled}})$ , one has the bound*

$$\text{min-cut}(\widetilde{\text{Temp}}) \leq (\log 3) \text{quantum-max-flow}(\widetilde{\text{Temp}}) \quad (29)$$

As a consequence, one has the bound

$$Q_{\text{cut}} \leq (\log 3) Q_{\min} + 1 \quad (30)$$

which implies that, asymptotically, the number of qubits used in the compression protocol of theorem 1 is at most  $\log 3$  times the minimum number  $Q_{\min}$ .

The proof is provided in appendix A. In conclusion, the number of qubits used in theorem 1 is either minimum (if all dimensions are power of the same integer), or within a factor  $\log(3) \approx 1.585$  of the minimum (if some dimensions are not power of the same integer). Equipped with this result, in the next section we will analyze the number of qubits needed to compress various families of tensor network states.

#### 4. Exponentially compressible families of tensor network states

In this section we apply the memory bound of theorem 1 to various families of tensor network states. In all these examples, the amount of memory qubits required to store the states is exponentially smaller than the original number of physical particles in which the states are embodied.

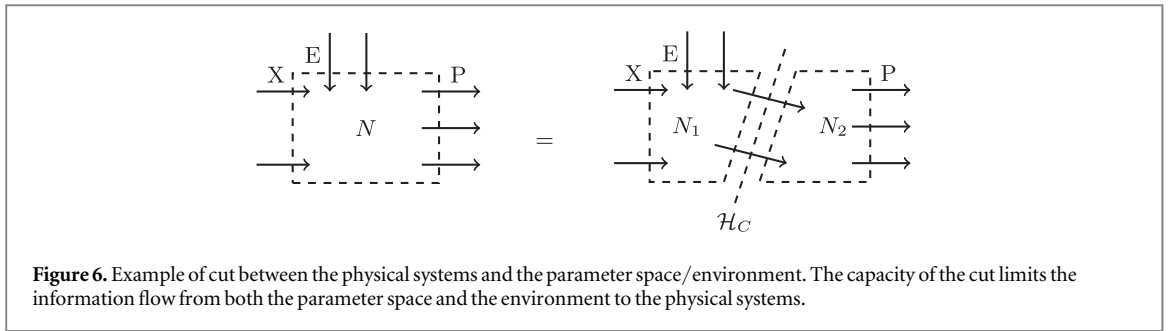
<sup>8</sup> Note that the quantum max-flow adopted here is the logarithm of the quantum max-flow defined in [46].





**Table 1.** Memory bounds for tensor network state families.

Case	State	Expression	Parameters	Memory (qubits, rounding up)
1	MPSs with variable boundary conditions	$ \Psi_{L,R}\rangle$ (31)	$L, R$	$2 \log d_c$
2	Site-independent MPSs	$ \Psi_{A,L,R}\rangle$ (32)	$A, L, R$	$(d_c^2 d_p - 1) \log(n + d_c^2 d_p - 1) + 2 \log d_c$
3	$n \times m$ PEPSs with variable boundary condition	$ \Psi_B^{(PEPS)}\rangle$ (figure B1)	$B$	$(2n + 2m) \log d_c$
4	Site-independent $n \times m$ PEPSs	$ \Psi_{A,B}^{(PEPS)}\rangle$ (figure B2)	$A, B$	$(d_c^4 d_p - 1) \log(nm + d_c^4 d_p - 1) + (2n + 2m) \log d_c$
5	Fixed $n$ -system state under $U_g^{\otimes n}$	$U_g^{\otimes n}  \Psi_0\rangle$	$g$	$\frac{d_p^2 + d_p - 2}{2} \log(n + d_p - 1)$
6	Tensor network state under $U_g^{\otimes n}$	$U_g^{\otimes n} N_*  v_x\rangle$	$g, x$	$\frac{d_p^2 + d_p - 2}{2} \log(n + d_p - 1) + \min\text{-cut}(\widetilde{\text{Temp}})$ where $\widetilde{\text{Temp}}$ is the flow network associated with $N$
7	MPSs with variable boundary conditions under $U_g^{\otimes n}$	$U_g^{\otimes n}  \Psi_{L,R}\rangle$	$L, R, g$	$\frac{d_p^2 + d_p - 2}{2} \log(n + d_p - 1) + 2 \log d_c$
8	$n \times m$ PEPSs with unknown boundary condition under $U_g^{\otimes nm}$	$U_g^{\otimes nm}  \Psi_B^{(PEPS)}\rangle$	$B, g$	$\frac{d_p^2 + d_p - 2}{2} \log(nm + d_p - 1) + (2n + 2m) \log d_c$



A local compression protocol can be constructed from a partial isometry  $V: \mathcal{H}_P \rightarrow \mathcal{H}_M$  that satisfies the following:

$$(V^\dagger V \otimes I_E)|\Psi_x\rangle = |\Psi_x\rangle, \quad \forall x \in X \tag{36}$$

where  $I_E$  is the identity operator on system E.

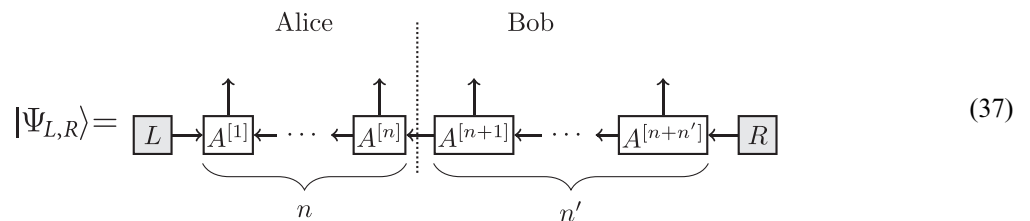
In order to generalize theorem 1 to the scenario of local compression, we need to cope with the presence of the inaccessible environment E. The key idea is to regard the environment not as an output of the tensor network, but as another source of information, in addition to the parameter Hilbert space in which the parameter  $x$  is encoded. Mathematically, this change of perspective corresponds to a reversal of the edges associated to the environment, which become inputs, instead of outputs. After the edges have been reversed, we apply theorem 1, and search for the minimum cut that separates physical systems from the parameter space and from the environment, as shown in figure 6.

This approach leads to an upper bound on the number of memory qubits needed for local compression, provided in the following proposition:

**Proposition 4.** Let  $\{|\Psi_x\rangle\}$  be a parametric family of pure states of the form  $|\Psi_x\rangle = N_*|v_x\rangle$ , where  $N_*: \mathcal{H}_X \rightarrow \mathcal{H}_P \otimes \mathcal{H}_E$  is a fixed tensor network operator and  $|v_x\rangle \in \mathcal{H}_X$  is some (not necessarily normalized) vector, parameterized by parameter  $x$ . Let  $N = (\text{Temp}, T)$  be the tensor network associated to the operator  $N_*$ , each of whose outgoing edges corresponds to either a physical system (i.e. a subsystem of P) or a part of the environment (i.e. a subsystem of E). Let  $\text{Temp}'$  be the tensor network template obtained from reversing all open edges of  $\text{Temp}$  that are associated to the environment. Let  $\widetilde{\text{Temp}'}$  be the flow network corresponding to the template  $\text{Temp}'$  via construction 1. For every cut  $C = (C_s, C_t)$  of  $\widetilde{\text{Temp}'}$ , the marginal state on system P of the state family  $\{|\Psi_x\rangle\}_{x \in X}$  can be compressed without errors into  $\lceil c(C) \rceil$  qubits.

The proof is provided in appendix C.

As an example, consider the scenario where Alice holds the leftmost  $n$  systems of an MPS and Bob holds the rest  $n'$  systems, and Alice's task is to store her systems in a quantum memory, while preserving the correlations with Bob's systems.



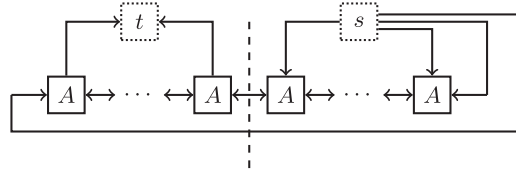
We assume Alice does not know the boundary conditions  $|L\rangle$  and  $|R\rangle$ , and therefore her goal is to find channels  $\mathcal{E}: S(\mathcal{H}_P) \rightarrow S(\mathcal{H}_M)$  and  $\mathcal{D}: S(\mathcal{H}_M) \rightarrow S(\mathcal{H}_P)$  that satisfy

$$(\mathcal{D} \circ \mathcal{E} \otimes \mathcal{I}_B)(|\Psi_{L,R}\rangle\langle\Psi_{L,R}|) = |\Psi_{L,R}\rangle\langle\Psi_{L,R}|, \quad \forall |L\rangle, |R\rangle \tag{38}$$

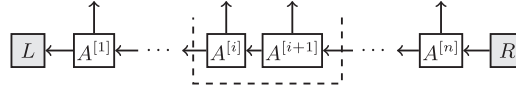
for some Hilbert space  $\mathcal{H}_M$  whose dimension should be minimized.

Using proposition 4, we convert the tensor network in equation (37) to into the flow network in figure 7, with the cut indicated by the dashed line. There are two cut edges, each has dimension  $d_c$ , and therefore Alice can still compress her state into  $\lceil \log \dim \mathcal{H}_C \rceil = \lceil 2 \log d_c \rceil$  qubits, as in the case of compression of an MPS with variable boundary conditions (section 4.1). The compression protocol will be presented explicitly in section 6.

Proposition 4 automatically provide upper bounds on the amount of memory needed to compress mixed tensor network states. Any such state  $\rho_x \in S(\mathcal{H}_P)$  can be regarded as the marginal of a pure tensor network state involving an environment, namely  $\text{Tr}_E[|\Psi_x\rangle\langle\Psi_x|] = \rho_x$  for some pure state  $|\Psi_x\rangle \in \mathcal{H}_P \otimes \mathcal{H}_E$ . Clearly, a local



**Figure 7.** Flow network for the compression of  $|\Psi_{L,R}\rangle$ . Except the edges connected to  $s$  or  $t$ , all edges are made bidirectional. The dashed line indicates the cut.



**Figure 8.** We regard the  $i$ th and the  $(i + 1)$ th systems as the physical systems to which the compression protocol will be applied. The other systems are regarded as the environment.

compression protocol for the purifications  $\{|\Psi_x\rangle\}$  is also a compression protocol for the mixed states  $\{\rho_x\}$ , as one can see by taking the partial trace over the environment on both sides of the local compression condition (35).

## 6. Compression protocol for MPSs with variable boundary conditions

Here we construct an explicit compression protocol for the family of all MPSs with variable boundary conditions. The idea is to perform a local compression on each pair of adjacent physical systems locally, and to iterate the protocol until we cannot reach a smaller memory size. For simplicity of presentation, we assume the number of physical systems  $n$  is a power of 2. We consider two adjacent physical systems, the  $i$ th and the  $(i + 1)$ th, and regard the others as the environment, as shown in figure 8.

By proposition 4, there exists a partial isometry  $V_{i,i+1}$  that faithfully encodes the  $i$ th and the  $(i + 1)$ th physical systems into a single system of dimension  $d_c^2$ . Note that, *per se*,  $V_{i,i+1}$  may not be a useful compression operation, because the dimension  $d_c^2$  may be larger than the dimension  $d_p^2$  of the two input systems. Nevertheless, we now show that a concatenation of partial isometries like  $V_{i,i+1}$  can squeeze the initial state into the minimum number of qubits, equal to  $\lceil 2 \log d_c \rceil$ .

Explicitly, the partial isometry  $V_{i,i+1}$  satisfies the local compression condition (36), which reads

$$(V_{i,i+1}^\dagger V_{i,i+1} \otimes I_{1,\dots,i-1,i+2,\dots,n})|\Psi_{L,R}\rangle = |\Psi_{L,R}\rangle \quad (39)$$

where  $I_{1,\dots,i-1,i+2,\dots,n} := \bigotimes_{k=1}^{i-1} I_k \otimes \bigotimes_{k=i+2}^n I_k$  is the identity operator on all systems except the  $i$ th and the  $(i + 1)$ th. Applying the partial isometries  $V_{1,2}, V_{3,4}, \dots, V_{n-1,n}$  in parallel, we obtain the relation

$$(V_{1,2}^\dagger V_{1,2} \otimes V_{3,4}^\dagger V_{3,4} \otimes \dots \otimes V_{n-1,n}^\dagger V_{n-1,n})|\Psi_{L,R}\rangle = |\Psi_{L,R}\rangle \quad \forall |L\rangle, \forall |R\rangle \quad (40)$$

This condition means that the product isometry  $V^{(1)} := V_{1,2} \otimes V_{3,4} \otimes \dots \otimes V_{n-1,n}$  defines an exact compression protocol that stores  $n$  systems (each of dimension  $d_p$ ) into  $n/2$  systems (each of dimension  $d_c^2$ ).

The construction can be iterated, because the output of the isometry  $V^{(1)}$  is itself an MPS. This can be verified by defining the tensors

$$\begin{array}{c} \uparrow \\ \leftarrow A^{[i,i+1]} \leftarrow \\ \uparrow \\ \leftarrow A^{[i]} \leftarrow A^{[i+1]} \leftarrow \\ \uparrow \\ V_{i,i+1}^\dagger \end{array} \quad := \quad \begin{array}{c} \uparrow \\ V_{i,i+1}^\dagger \\ \uparrow \uparrow \\ \leftarrow A^{[i]} \leftarrow A^{[i+1]} \leftarrow \\ \uparrow \\ \leftarrow A^{[i,i+1]} \leftarrow \end{array} \quad (41)$$

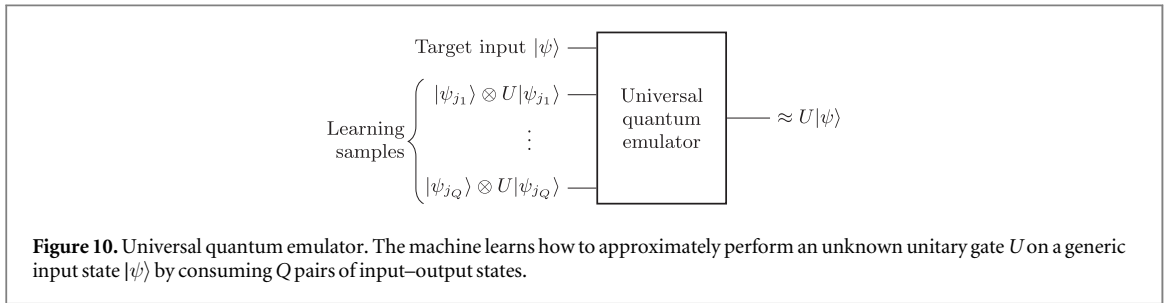
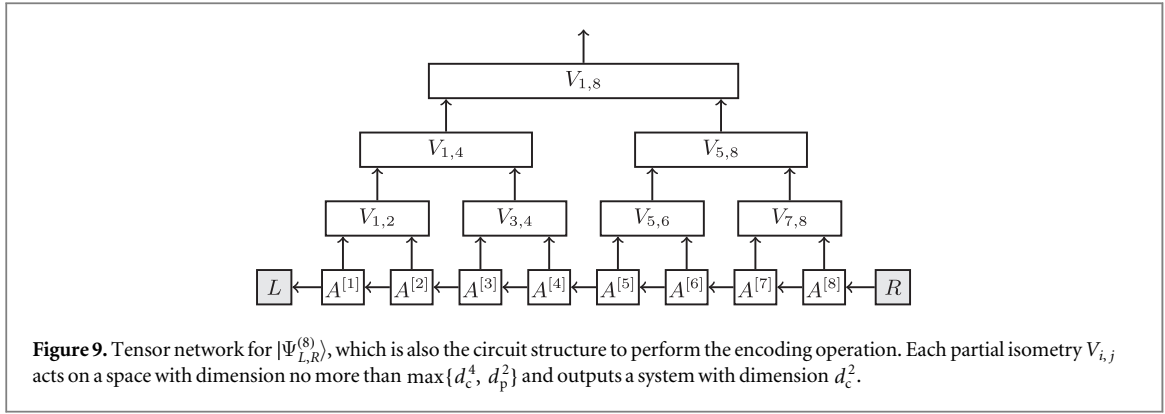
so that the output state  $|\Psi_{L,R}^{(2)}\rangle := V^{(1)}|\Psi_{L,R}\rangle$  can be expressed in the MPS form

$$|\Psi_{L,R}^{(2)}\rangle = \begin{array}{c} \uparrow \\ L \leftarrow A^{[1,2]} \leftarrow A^{[3,4]} \leftarrow \dots \leftarrow A^{[n-1,n]} \leftarrow R \\ \uparrow \uparrow \uparrow \end{array} \quad (42)$$

Crucially, the bond dimension is still  $d_c$ .

Now, we can again apply proposition 4 to each adjacent pair of  $d_c^2$ -dimensional systems, and compress them into a single  $d_c^2$ -dimensional system, using partial isometries  $V_{1,4}, V_{5,8}, \dots, V_{n-3,n}$ . Also in this case, proposition 4





guarantees that the  $(n/2)$ -particle  $|\Psi_{L,R}^{(2)}\rangle$  is encoded faithfully into the  $(n/4)$ -particle state  $|\Psi_{L,R}^{(4)}\rangle := V^{(2)}|\Psi_{L,R}^{(2)}\rangle$ ,  $V^{(2)} := V_{1,4} \otimes V_{5,8} \otimes \cdots \otimes V_{n-3,n}$ .

Iterating this pairwise encoding for a total of  $\log n$  times, we can faithfully compress the input state into  $\lceil 2 \log d_c \rceil$  qubits. An illustration of the compression protocol for  $n = 8$  is provided in figure 9.

The encoding can be realized by a quantum circuit of depth  $O(\log n)$ , implementing partial isometries  $V_{i,j}$  shown above. Since each partial isometry has size no larger than  $d_c^2 \times \max\{d_c^4, d_p^2\}$  and the circuit uses  $n - 1$  such partial isometries in total, the overall complexity of the encoding operations is  $O(\text{poly}(n))$  (assuming  $d_c$  is polynomial in  $n$ ), meaning that this construction is efficient in the number of physical systems. The same argument applies to the decoding circuit, which can be obtained from the encoding circuit by reversing each gate.

Note that the above technique also applies to the local compression of MPSs, corresponding to the scenario where only a subset of the physical system is accessible.

## 7. Compression algorithm for pure states in low-dimensional subspaces

### 7.1. The algorithm

Here we outline a general quantum algorithm for compressing families of pure states lying in a low-dimensional subspace of a high-dimensional quantum system. The idea of the algorithm is to train a quantum machine to perform the desired compression operations, by showing to the machine how such operations should act on a fiducial set of input states.

The algorithm is based on the *universal quantum emulator* of Marvian and Lloyd [53], a quantum circuit that ‘learns’ how to implement a completely unknown unitary gate  $U$  from a set of examples, as illustrated in figure 10. To implement the gate  $U$  on a state  $|\psi\rangle$ , the emulator consumes  $Q$  pairs of input–output states, of the form  $(|\psi_{j_k}\rangle, U|\psi_{j_k}\rangle)$  with  $k \in \{1, \dots, Q\}$ . Each input state  $|\psi_{j_k}\rangle$  is taken from a set of  $m$  possible inputs  $\{|\psi_j\rangle\}_{j=1}^m$ , with  $m \leq Q$ .

For large  $Q$ , the output of the emulator converges to the desired output state  $U|\psi\rangle$  provided that:

1. The input state  $|\psi\rangle$  belongs to the subspace  $\mathcal{K} := \text{Span}\{|\psi_j\rangle\}_{j=1}^m$ .
2. The quantum channel  $\mathcal{R}: S(\mathcal{K}) \rightarrow S(\mathcal{K})$  defined by

$$\mathcal{R}(\rho) := \frac{1}{m} \sum_{j=1}^m (I_{\mathcal{K}} - 2|\psi_j\rangle\langle\psi_j|) \rho (I_{\mathcal{K}} - 2|\psi_j\rangle\langle\psi_j|), \quad \forall \rho \in S(\mathcal{K}) \quad (43)$$

where  $I_{\mathcal{K}}$  is the identity on  $\mathcal{K}$ , is *mixing* [54], meaning that every input state  $\rho$  converges to a fixed state  $\rho_0$  (in this particular case  $\rho_0 = I/d$ ) under a large number of repeated applications of the channel; in formula,  $\lim_{k \rightarrow \infty} \mathcal{R}^k(\rho) = \rho_0$ .

The second condition is equivalent to the statement that the channel  $\mathcal{R}$  has one and only one eigenvalue on the unit circle [55]. Since all the eigenvalues of a quantum channel are inside the unit circle, this implies that the eigenvalue  $\lambda$  with the second largest modulus satisfies the condition  $|\lambda| < 1$ . Equivalently, this means that the spectral gap  $\gamma_{\mathcal{R}} := 1 - |\lambda|$  is non-zero.

The number of input–output pairs needed to approximate the gate  $U$  depends on the error tolerance  $\varepsilon$ , on the dimension of the subspace spanned by the input states  $\{|\psi_j\rangle\}$ , denoted by  $r$ , and on the spectral gap  $\gamma_{\mathcal{R}}$ . Specifically, Marvian and Lloyd [53] show that  $Q$  grows as

$$Q = O(r^2 \varepsilon^{-1} \gamma_{\mathcal{R}}^{-2} \log^2(r \varepsilon^{-1})) \quad (44)$$

To emulate the encoding, which is an isometry instead of a unitary, we choose a unitary  $U$  acting on the composite system  $\mathcal{P} \otimes \mathcal{M}$ , and satisfying the condition

$$U(|\Psi_x\rangle \otimes |W_0\rangle) = |\Psi_0\rangle \otimes V|\Psi_x\rangle \quad (45)$$

where  $|\Psi_0\rangle$  ( $|W_0\rangle$ ) is a fixed state of the physical (memory) system, and  $V: \mathcal{H}_{\mathcal{P}} \rightarrow \mathcal{H}_{\mathcal{M}}$  is an isometry that encodes the input states into the memory system. To train the emulator, we use input (output) states of the form

$$|\psi_j\rangle = |\Phi_j\rangle \otimes |W_0\rangle, \quad (U|\psi_j\rangle = |\Psi_0\rangle \otimes V|\Phi_j\rangle) \quad (46)$$

where the states  $\{|\Phi_j\rangle\}_{j=1}^m$ , hereafter called the *fiducial states*, span the input subspace  $\mathcal{H}_{\text{in}} := \text{Span}\{|\Psi_x\rangle\}_{x \in \mathcal{X}}$ . In general, the states  $\{|\Phi_j\rangle\}_{j=1}^m$  may or may not be a subset of the state family we are trying to compress.

The isometry  $V$  is constructed from the Gram matrix  $G_{jk} = \langle \Phi_j | \Phi_k \rangle$  via the following procedure:

1. Compute the rank of  $G$ , denoted by  $r$ , and set  $\mathcal{H}_{\mathcal{M}} = \mathbb{C}^r$ . The calculation of the rank can be done, e.g. by diagonalising  $G$  and counting the non-zero eigenvalues. Note that  $r$  is equal to the dimension of  $\text{Span}\{|\Phi_j\rangle\}_{j=1}^m$ , which, by construction, is equal to the dimension of the input subspace  $\mathcal{H}_{\text{in}}$ .
2. Construct an  $r \times m$  matrix  $W$  such that  $W^\dagger W = G$ . This can be done by diagonalising  $G$  as  $G = X \Lambda X^\dagger$ , setting  $W = \sqrt{\Lambda} X^\dagger$ , and removing the zero rows from  $W$ .
3. For  $j \in \{1, \dots, m\}$ , define  $V|\Phi_j\rangle := W|j\rangle$ .

The above definition is well-posed and uniquely determines the linear operator  $V$  within the subspace spanned by the fiducial states  $\{|\Phi_j\rangle\}_{j=1}^m$ . Note that  $V$  is an isometry: indeed, for every vector  $|\Psi\rangle = \sum_j c_j |\Phi_j\rangle \in \mathcal{H}_{\text{in}}$ , one has

$$\|V|\Psi\rangle\|^2 = \sum_{j,k} \bar{c}_j c_k \langle \Phi_j | V^\dagger V | \Phi_k \rangle = \sum_{j,k} \bar{c}_j c_k \langle j | W^\dagger W | k \rangle = \sum_{j,k} \bar{c}_j c_k \langle j | G | k \rangle = \sum_{j,k} \bar{c}_j c_k \langle \Phi_j | \Phi_k \rangle = \| |\Psi\rangle \|^2 \quad (47)$$

To train the emulator, we will use input–output pairs of the form  $(|\Phi_j\rangle \otimes |W_0\rangle, |\Psi_0\rangle \otimes |W_j\rangle)$ , with  $|W_j\rangle := W|j\rangle$ . Now, recall that the number of input–output pairs needed by the emulator depends on the spectral gap of the channel  $\mathcal{R}$  in equation (43), with states  $|\psi_j\rangle = |\Psi_j\rangle \otimes |W_0\rangle$ . Since the states  $|\Psi_j\rangle \otimes |W_0\rangle$  are unitarily equivalent to the states  $|\Psi_0\rangle \otimes |W_j\rangle$ , the spectral gap of the channel  $\mathcal{R}: S(\mathcal{H}_{\mathcal{P}}) \rightarrow S(\mathcal{H}_{\mathcal{P}})$  is equal to the spectral gap of the channel  $\mathcal{R}': S(\mathcal{H}_{\mathcal{M}}) \rightarrow S(\mathcal{H}_{\mathcal{M}})$  defined by

$$R'(\rho) := \frac{1}{m} \sum_{j=1}^m (I_r - 2|W_j\rangle\langle W_j|) \rho (I_r - 2|W_j\rangle\langle W_j|), \quad \forall \rho \in S(\mathbb{C}^r) \quad (48)$$

where  $I_r$  is the identity operator on  $\mathbb{C}^r$ . This observation is important because channel  $\mathcal{R}'$  acts only on the memory space, and therefore its spectral gap involves the diagonalisation of a low-dimensional matrix. With the knowledge of the spectral gap, we can keep under control the error in the emulator protocol, and determine how many input–output pairs are needed to attain the desired level of accuracy in the implementation of the gate  $U$ .

All together, the algorithm can be summarized as follows:

**Algorithm 1.** Encoding operation for state family  $\{|\Psi_x\rangle\}$ .

**Input:** Quantum state to be compressed  $|\Psi\rangle \in \mathcal{H}_{\mathcal{P}}$  and classical description of fiducial set  $\{|\Phi_j\rangle\}_{j=1}^m$

**Output:** Approximation of the compressed state  $V|\Psi\rangle \in \mathcal{H}_{\mathcal{M}}$

**Preprocessing:**

1 Compute the Gram matrix  $G_{jk} = \langle \Phi_j | \Phi_k \rangle$ ;

2 Compute the rank  $r = \text{rank}(G)$  and set  $\mathcal{H}_{\mathcal{M}} = \mathbb{C}^r$ ;

(Continued.)

- 3 Find an  $r \times m$  matrix  $W$  satisfying  $W^\dagger W = G$ ;
  - 4 Compute the spectral gap of the channel  $\mathcal{R}$  in equation (43).
 

**Emulation:**

    - 5 Run the universal quantum emulator with input state  $|\Psi\rangle \otimes |W_0\rangle$ ;
    - 6 **foreach** emulator's request for the  $j$ -th input-output pair **do**
    - 7     Prepare  $|\Phi_j\rangle$ ;
    - 8     Prepare  $|W_j\rangle := W|j\rangle$ ;
    - 9     Prepare  $|\Psi_0\rangle$  and  $|W_0\rangle$ ;
    - 10    Input the pair  $(|\Phi_j\rangle \otimes |W_0\rangle, |\Psi_0\rangle \otimes |W_j\rangle)$  into the emulator;
    - 11 **end**
    - 12 Discard the first system of the emulator's output.
- 

The above algorithm implements an approximation of the encoding channel  $\mathcal{E}(\rho) = \text{Tr}_P[U(\rho \otimes |W_0\rangle\langle W_0|)U^\dagger]$  to any desired accuracy. The same construction applies to the decoding channel  $\mathcal{D}(\rho) = \text{Tr}_M[U^\dagger(|\Psi_0\rangle\langle\Psi_0| \otimes \rho)U]$ , by simply exchanging the role of the input and output of the quantum emulator. The algorithm reaches the optimal memory size for exact compression, because the memory space has dimension  $r$ , which is exactly equal to the dimension of the input subspace  $\mathcal{H}_{\text{in}}$ .

## 7.2. Running time

Here we analyze the running time of the general quantum compression algorithm, providing sufficient conditions for its efficient implementation. We will measure the size of the input physical system  $P$  in terms of the number of logical qubits needed to represent it, namely  $n := \lceil \log \dim \mathcal{H}_P \rceil$ .

The running time of Step 1 (calculation of the Gram matrix) depends of the structure of the fiducial states. The calculation of the matrix element  $G_{jk} = \langle \Phi_j | \Phi_k \rangle$  can be implemented efficiently for various families of tensor network states, such as MPSs [43, 56–58] and MERAs [38]. The number of matrix elements is  $O(m^2)$ , where  $m$  is the size of the fiducial set. Hence, the efficient implementation of Step 1 requires  $m$  to be at most polynomial in  $n$ . In the following, we will always assume  $m = O(\text{poly}(n))$ . Of course, this implies that the subspace containing the input states has polynomial dimension  $d_{\text{in}} = O(\text{poly}(n))$ , namely it is exponentially smaller than the total Hilbert space  $\mathcal{H}_P$ .

Under the assumption  $m = O(\text{poly}(n))$ , Steps 2 and 3 (calculation of the rank and construction of the matrix  $W$ ) can be implemented in polynomial time by diagonalising the Gram matrix  $G$ , e.g. with the QR algorithm [59]. Note that one has  $r = d_{\text{in}} = O(\text{poly}(n))$ , meaning that the memory system has polynomial dimension.

Step 4, the calculation of the spectral gap, can be implemented in polynomial time by diagonalising the  $r^2 \times r^2$  matrix describing the channel  $\mathcal{R}'$ .

The emulation part has running time  $T_{\text{tot}} = Q T_{\text{prep}} + T_{\text{emulator}}$ , where  $Q$  is the number of input–output pairs used by the emulator,  $T_{\text{prep}}$  is the time complexity of preparing each input–output pair, and  $T_{\text{emulator}}$  is the running time of the emulator.

The running time of the emulator can be bounded as  $T_{\text{emulator}} = O(n Q \log Q)$  [53]. The complexity of preparing the input–output pair  $(|\Phi_j\rangle \otimes |W_0\rangle, |\Psi_0\rangle \otimes |W_j\rangle)$  is essentially the complexity of preparing the fiducial state  $|\Phi_j\rangle$ . Indeed,  $|\Psi_0\rangle$  can be chosen to be any efficiently preparable state, i.e. any state preparable in  $O(\text{poly}(n))$  time. The states  $|W_j\rangle$  and  $|W_0\rangle$  are efficiently preparable by construction, because they are vectors with a polynomial number of efficiently computable entries. Hence, the preparation time  $T_{\text{prep}}$  is polynomial if and only if each fiducial state  $|\Phi_j\rangle$  can be prepared in polynomial time. This condition is satisfied whenever the fiducial states  $\{|\Phi_j\rangle\}_{j=1}^m$  are MPSs [43, 58] or MERAs [38].

By equation (44), under the assumption  $m = O(\text{poly}(n))$ , the number of input–output examples required by the emulator is polynomial in  $n$  and  $\varepsilon^{-1}$  if the inverse spectral gap  $\gamma_{\mathcal{R}}^{-1}$  is at most polynomial in  $n$ .

In summary, the compression operations can be implemented in polynomial time if:

1. The number of fiducial states is at most polynomial,  $m = O(\text{poly}(n))$ .
2. The overlap of any two fiducial states can be computed in polynomial time.
3. Each fiducial state can be prepared in polynomial time.
4. The inverse spectral gap is at most polynomial,  $\gamma_{\mathcal{R}}^{-1} = O(\text{poly}(n))$ .

Conditions 1 and 2 are relatively straightforward for tensor network states. As we have seen in section 4, many families of tensor network states are contained in subspaces of dimension  $O(\text{poly}(n))$ , making it easy to

satisfy Condition 1. In addition, the overlap between two tensor network states can be efficiently computed in many physically relevant cases (e.g. MPSs and MERAs). In those cases, if the input subspace  $\mathcal{H}_{\text{in}}$  has polynomial dimension (as it must be in order to satisfy Condition 1), then every state in  $\mathcal{H}_{\text{in}}$  is a linear combination of polynomial number of tensor network states, and the overlap between any two states in  $\mathcal{H}_{\text{in}}$  can be computed in polynomial time.

Condition 3 is satisfied when the fiducial states are efficiently preparable tensor network states, such as MPSs or MERAs, which can be prepared through sequences of isometries [38, 43, 58]. However, it is not automatically satisfied when the fiducial states are generic vectors in  $\mathcal{H}_{\text{in}}$ . The problem is that, in general, a linear combination of efficiently preparable states may not be an efficiently preparable state. For MPSs, however, this condition is satisfied:

**Lemma 1.** *Let  $\{|\Sigma_k\rangle\}_{k=1}^t$  be a polynomial-size set of MPSs that span the input space  $\mathcal{H}_{\text{in}}$ , and let  $\{c_k\}_{k=1}^t$  be a set of coefficients such that the linear combination  $|\Psi\rangle = \sum_k c_k |\Sigma_k\rangle$  is a unit vector. If the initial states have bond dimension  $d_c$ , then the state  $|\Psi\rangle$  is an MPS with bond dimension  $t d_c$  and can be prepared in polynomial time.*

The proof is provided in appendix D. For more general families of tensor network states, other than MPSs, a sufficient condition for the efficient preparability of the fiducial states will be given in the next section.

Finally, Condition 4 can be satisfied by a suitable choice of fiducial states, as we show in the following.

In general, Condition 4 is satisfied by choosing the fiducial set to be ‘sufficiently dense’ in the input subspace. An example of such choice is provided in the following. Let  $\{|1\rangle, \dots, |r\rangle\}$  be a fixed basis for the input subspace  $\mathcal{H}_{\text{in}}$ . For every  $l \in \{1, \dots, r\}$ , we define

$$|\Psi_{l,x,\pm}\rangle := \frac{|l\rangle \pm |l \oplus 1\rangle}{\sqrt{2}}, \quad |\Psi_{l,y,\pm}\rangle := \frac{|l\rangle \pm i|l \oplus 1\rangle}{\sqrt{2}}, \quad |\Psi_{l,z,+}\rangle := |l\rangle, \quad |\Psi_{l,z,-}\rangle := |l \oplus 1\rangle \quad (49)$$

where  $\oplus$  denotes addition modulo  $r$ , and we adopt the convention  $|0\rangle := |r\rangle$ .

**Lemma 2.** *The spectral gap of the channel  $\mathcal{R}$  in equation (43) associated to the states  $\{|\Psi_{l,\alpha,s}\rangle\}_{l \in \{1, \dots, r\}, \alpha \in \{x,y,z\}, s \in \{+,-\}}$  in equation (49) is  $\gamma_{\mathcal{R}} = 8[\sin(\pi/r)]^2 / (3r)$ .*

The proof is provided in appendix E. Lemma 2 guarantees that the fiducial set  $\{|\Psi_{l,\alpha,s}\rangle\}_{l \in \{1, \dots, r\}, \alpha \in \{x,y,z\}, s \in \{+,-\}}$  gives rise to a channel with inverse spectral gap growing at most as  $O(r^3)$ , where  $r$  is the dimension of the input subspace. Since the input subspace is assumed to be of polynomial dimension (Condition 1), this result guarantees that the inverse spectral gap is at most polynomial.

Lemmas 1 and 2 imply that every family of MPSs that can be compressed into a logarithmic number of qubits can be compressed in polynomial time on a quantum computer.

**Theorem 2.** *Let  $\{|\Psi_x\rangle\}_{x \in \mathcal{X}}$  be a parametric family of  $n$ -particle MPSs with fixed bond dimension  $d_c$ . If the input subspace  $\mathcal{H}_{\text{in}} = \text{Span}\{|\Psi_x\rangle\}_{x \in \mathcal{X}}$  has polynomial dimension  $r = \text{poly}(n)$ , then the states  $\{|\Psi_x\rangle\}_{x \in \mathcal{X}}$  can be compressed into  $\lceil \log r \rceil$  qubits with error  $\varepsilon$  in polynomial time  $\text{poly}(n, \varepsilon^{-1})$ .*

**Proof.** Let  $\{|\Sigma_k\rangle\}_{k=1}^t$  be a subset of the states  $\{|\Psi_x\rangle\}_{x \in \mathcal{X}}$ , with the properties that (i)  $\{|\Sigma_k\rangle\}_{k=1}^t$  spans the input subspace, and (ii) the number of states  $t$  is polynomial in  $n$ . Such a set exists because, by hypothesis,  $\mathcal{H}_{\text{in}}$  has polynomial dimension. Then, let  $\{|l\rangle\}_{l=1}^r$  be the orthonormal basis of  $\mathcal{H}_{\text{in}}$  obtained by applying the Gram–Schmidt procedure to the set  $\{|\Sigma_k\rangle\}_{k=1}^t$ . Explicitly,  $|1\rangle := |\Sigma_1\rangle, |2\rangle := (|\Sigma_2\rangle - \langle \Sigma_1 | \Sigma_2 \rangle |\Sigma_1\rangle) / \sqrt{1 - |\langle \Sigma_1 | \Sigma_2 \rangle|^2}, \dots, |t\rangle := (|\Sigma_t\rangle - \sum_{k < t} \langle k | \Sigma_t \rangle |k\rangle) / \sqrt{1 - \sum_{k < t} |\langle k | \Sigma_t \rangle|^2}$ . By construction, each vector  $|l\rangle$  is a linear combination of MPSs, and the expansion coefficients can be computed from the scalar products  $\langle \Sigma_k | \Sigma_l \rangle$ . Since the states  $\{|\Sigma_k\rangle\}_{k=1}^t$  are MPSs, the scalar products can be computed efficiently, and lemma 1 implies that the linear combinations  $\{|l\rangle\}_{l=1}^r$  can be prepared in polynomial time. From the basis  $\{|l\rangle\}_{l=1}^r$ , one can then construct the fiducial states defined in equation (49). Since the fiducial states are linear combinations of at most 2 basis vectors, they can all be prepared in polynomial time (again, due to lemma 1). Moreover, lemma 2 guarantees that the channel  $\mathcal{R}$  associated to the states (49) has inverse spectral gap of polynomial size. Hence, all Conditions 1, 2, 3, and 4 are satisfied, implying that the compression algorithm 1 runs in polynomial time on the states  $\{|\Sigma_k\rangle\}_{k=1}^t$ .  $\square$

Theorem 2 guarantees that most relevant families of MPSs can be compressed efficiently on a quantum computer. For other state families, a sufficient condition for compressibility in polynomial time is given by the following proposition:

**Proposition 5.** Let  $\{|\Psi_x\rangle\}_{x\in\mathcal{X}}$  be a parametric family of  $n$ -particle tensor network states with a given network template. If the input subspace  $\mathcal{H}_{\text{in}} = \text{Span}\{|\Psi_x\rangle\}_{x\in\mathcal{X}}$  contains a spanning set  $\{|\Sigma_k\rangle\}_{k=1}^l$  with the following properties

- (i) the number of states is at most polynomial in  $n$ ,
- (ii) each state  $|\Sigma_k\rangle$  is efficiently preparable by a coherent process  $|k\rangle \mapsto |k\rangle \otimes |\Sigma_k\rangle$ ,
- (iii) the Gram matrix  $S_{kl} := \langle \Sigma_k | \Sigma_l \rangle$  is efficiently computable,
- (iv) the minimum non-zero eigenvalue of the Gram matrix  $S$  is at least inverse-polynomial in  $n$ ,

then the states  $\{|\Psi_x\rangle\}_{x\in\mathcal{X}}$  can be compressed into  $\lceil \log r \rceil$  qubits with error  $\varepsilon$ , using a quantum algorithm that runs in polynomial time  $O(\text{poly}(n, \varepsilon^{-1}))$ .

The proof is provided in appendix F. Note that proposition 5 is not specific to tensor network states, and applies broadly to every parametric family of states confined in a low-dimensional subspace of the total Hilbert space.

## 8. Conclusions

We designed compression protocols for parametric families of tensor network states, in which some of the tensors depend on the parameters, while some others are constant. Physically, the variable tensors can be associated to systems that carry unknown parameters, or to inaccessible degrees of freedom of the environment. Given a tensor network with constant and variable tensors, one can construct a flow network, where the variable tensors are connected to the source, and the physical systems are connected to the sink. In such a network, every cut identifies an exact deterministic compression protocol that compresses every state in the parametric family into a quantum memory of dimension equal to the size of the cut. In addition to quantifying the amount of memory needed to store tensor network states, we provided a general quantum compression algorithm, and we identified sufficient conditions for the algorithm to run in polynomial time, showing that they are satisfied by all families of MPSs.

Our results can be applied to site-independent tensor network states of  $n$  quantum systems, showing that every such state can be compressed without error into a memory of  $O(\log n)$  qubits. This scaling is optimal, because the set of SIMPSs contains as a subset the set of all identically-prepared states, which is known to require  $O(\log n)$  qubits, both for exact [11] and approximate compression protocols [12, 13]. The optimal prefactor in the logarithmic scaling of the memory for general site-independent tensor network states remains to be determined.

Our results can also be used to provide upper bounds on the amount of information one can encode into multipartite system using tensor network codes, such as the toric code [60–62] and holographic codes [63, 64]. For example, our method shows that a toric code with circumference  $L$  and with variable boundary conditions can be faithfully compressed into  $L$  qubits (a toric code is a PEPS with  $d_c = 2$  [36]). As a consequence, we can deduce that the number of qubits one can encode with the toric code is at most  $L$ . This result is consistent with the construction by Bravyi and Kitaev [62], which shows that one can encode up to  $L/2 - 1$  qubits. The discrepancy between this value and our bound is mostly due to that in [62], the boundary condition of a toric code is not arbitrary, while we consider arbitrary boundary conditions. More generally, we showed that tensor network states with variable boundary and constant interior satisfy an area law, according to which the number of qubits needed to compress these states is proportional to the size of the boundary.

In this work we mainly focused on exact compression. Since noise and imperfections are unavoidable in every realistic implementation, an important avenue of future research is to extend our results to approximate compression protocols. Tolerating a small compression error could offer great savings in terms of the amount of memory needed to store families of tensor network states. In the case of uncorrelated systems, it was observed that tolerating any non-zero error decreases the memory size discontinuously [11, 12]. Extending the study of this phenomenon to correlated systems is an interesting open question for future research.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China through grant 11675136, the Hong Kong Research Grant Council through Grant No. 17300317 and 17300918, the HKU Seed Funding for Basic Research, the Foundational Questions Institute through grant FQXi-RFP3-1325, Quantum Causal Structures,

the Croucher Foundation, the Swiss National Science Foundation via the National Center for Competence in Research ‘QSIT’ as well as project No. 200020\_165843, and the ETH Pauli Center for Theoretical Studies. This publication was made possible through the support of the ID# 61466 grant from the John Templeton Foundation, as part of the ‘‘The Quantum Information Structure of Spacetime (QISS)’’ Project (qiss.fr). The opinions expressed in this publication are those of the author(s) and do not necessarily reflect the views of the John Templeton Foundation.

## Appendix A. Proof of proposition 3

For a generic tensor network template  $\text{Temp} = (G, d, V_{\text{filled}})$ , define the new template  $\text{Temp}_2 = (G, d_2, V_{\text{filled}})$  by setting  $d_2(e) := 2^{\lfloor \log d(e) \rfloor}$ , and let  $C^* = (C_s^*, C_t^*)$  be a minimum cut of the flow network  $\widetilde{\text{Temp}}_2$ .

Then, we have the following chain of inequalities:

$$\begin{aligned} \min\text{-cut}(\widetilde{\text{Temp}}) &\leq c(C^*) = \sum_{u \in C_s^*, v \in C_t^*} \log d(u, v) \leq \left( \max_e \frac{\log d(u, v)}{\lfloor \log d(u, v) \rfloor} \right) \sum_{u \in C_s^*, v \in C_t^*} \left\lfloor \log d(u, v) \right\rfloor \\ &\leq \left( \max_{n \in \mathbb{N}} \frac{\log(n)}{\lfloor \log(n) \rfloor} \right) \sum_{u \in C_s^*, v \in C_t^*} \left\lfloor \log d(u, v) \right\rfloor = \log 3 \cdot \min\text{-cut}(\widetilde{\text{Temp}}_2) \\ &= \log 3 \cdot \text{quantum-max-flow}(\widetilde{\text{Temp}}_2) \leq \log 3 \cdot \text{quantum-max-flow}(\widetilde{\text{Temp}}) \end{aligned} \quad (\text{A1})$$

the last inequality following from the fact that the set of tensor network operators with template  $\widetilde{\text{Temp}}_2$  is included in the set of tensor network operators with template  $\widetilde{\text{Temp}}$ . This proves the upper bound  $\min\text{-cut}(\widetilde{\text{Temp}}) \leq \log 3 \cdot \text{quantum-max-flow}(\widetilde{\text{Temp}})$ .  $\square$

## Appendix B. Memory bounds for various tensor network families

### B.1. PEPS with variable boundary conditions

PEPSs are a higher dimensional analog of MPSs [35, 36]. A PEPS is defined by a lattice of tensors, where each tensor has edges connected to its neighbors. Consider a set of 2-d PEPSs in which all the tensors are fixed except those on the boundary, as in figure B1.

Here, each tensor  $A^{[i,j]}$  in the figure is a fixed order-5 tensor, and the shaded loop is a tensor  $B$  describing a variable boundary condition.

As in the MPS case, we regard the tensors  $A^{[i,j]}$  on the lattice as the tensor network  $N$ , and by properly choosing the edge directions,  $N$  defines a linear operator from the systems on the boundary to the physical systems. We call this linear operator  $N_*$ , and write the PEPS as  $|\Psi_B^{(\text{PEPS})}\rangle = N_*|B\rangle$ , where  $|B\rangle$  is a vectorised version of the tensor  $B$  describing the boundary condition. Then, we convert  $N$  to a flow network and look for its minimum cut. Assuming that the bond dimension  $d_c$  is a constant, while the lattice has size  $n \times m$  for large  $n$  and  $m$ , the optimal cut consists of the source (which replaces the tensor  $B$ ) on one side, and of the sink and the tensors  $A^{[i,j]}$  on the other side, as shown in figure B1. The cut edges contain  $2n + 2m$  number of  $d_c$ -dimensional systems, with combined dimension  $d_c^{2n+2m}$ . Using theorem 1, we conclude that the states  $\{|\Psi_B^{(\text{PEPS})}\rangle\}$  can be compressed into  $\lceil (2n + 2m)\log d_c \rceil$  qubits.

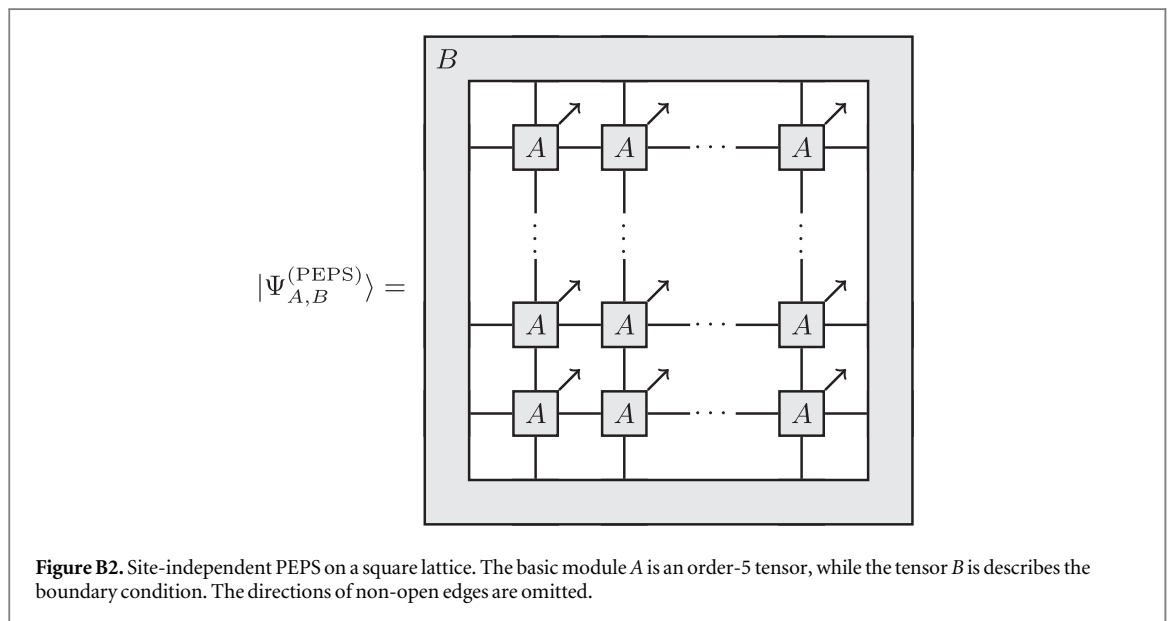
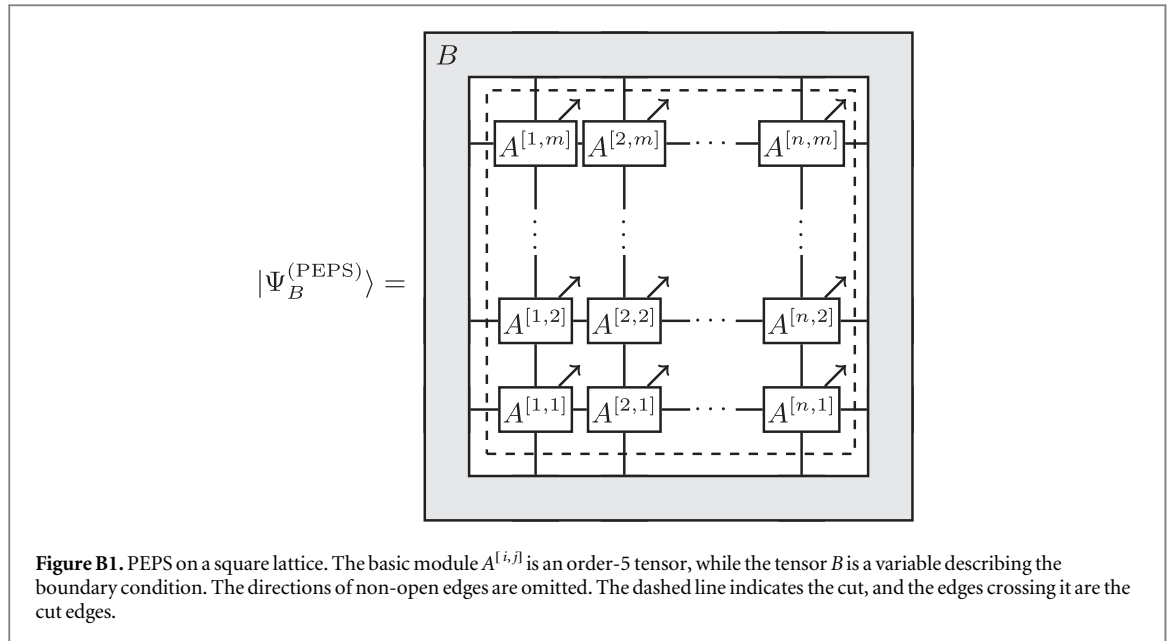
This result is consistent with the area law for PEPSs, which indicates that the amount of information contained in a two-dimensional region is upper bounded by a term proportional to its perimeter, in this case  $2n + 2m$ . More generally, this result is an instance of the bulk-boundary correspondence in [65], which shows that the bulk (namely physical systems) and boundary of a PEPS are related by an isometry. Our result can be seen as a special case of the ‘holographic compression’ of [66], which states that a state with area law can be approximately compressed into a memory proportional to the boundary size. In the special case of PEPSs, our construction shows that the compression is exact.

### B.2. Site-independent PEPSs

The method in section 4.2 for MPSs can be generalized to arbitrary systems that are ‘finitely correlated’, in the sense that they have a finite bond dimension. For example, figure B2 shows a site-independent PEPS defined on a square lattice. Each  $A$  in the figure is an order-5 tensor, and the loop is a tensor  $B$  describing the boundary condition.

Now, suppose that the state  $|\Psi_{A,B}^{(\text{PEPS})}\rangle$  is a generic site-independent PEPS, defined on a  $n \times m$  square lattice, with fixed bond dimension  $d_c$  and physical dimension  $d_p$ . By vectorization, the tensor  $A$  is transformed into a vector  $|A\rangle$  in a vector space of dimension  $d_c^4 d_p$ . The dimension of the symmetric subspace of  $nm$  copies of





$d_c^4 d_p$ -dimensional vectors is  $\begin{pmatrix} nm + d_c^4 d_p - 1 \\ d_c^4 d_p - 1 \end{pmatrix}$ . With the same argument as in the SIMPS case,

$$|A\rangle^{\otimes nm} = V_S |S_A\rangle, \text{ where } V_S \text{ is an isometry, and } |S_A\rangle \text{ is a vector in space } S \text{ with dimension } \begin{pmatrix} nm + d_c^4 d_p - 1 \\ d_c^4 d_p - 1 \end{pmatrix}.$$

We obtain the following tensor network representation for  $|\Psi_{A,B}^{(PEPS)}\rangle$ :

$$|\Psi_{A,B}^{(PEPS)}\rangle = \begin{array}{c} \begin{array}{c} I^{\otimes(2n+2m)} \\ \vdots \\ \vdots \\ B \end{array} \\ \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ S_A \end{array} \end{array} \begin{array}{c} \begin{array}{c} I^{\otimes(2nm-n-m)} \\ \vdots \\ \vdots \\ \vdots \\ V_S \end{array} \\ \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{array} \end{array} = \langle\langle I |^{\otimes(2nm+n+m)} (|B\rangle \otimes V_S |S_A\rangle) \end{array} \quad (B1)$$

where we have rearranged the outgoing edges of  $V_S$  such that the tensor labeled  $I^{\otimes(2n+2m)}$  denotes the connections between  $V_S$  and the boundary condition  $B$ , and  $I^{\otimes(2nm-n-m)}$  corresponds to the connections between neighboring copies of  $A$ . The open edges corresponds to the  $nm$  number of physical systems.

$$|S_A\rangle \text{ has dimension } \binom{nm + d_c^4 d_p - 1}{d_c^4 d_p - 1} \leq (nm + d_c^4 d_p - 1)^{d_c^4 d_p - 1}, \text{ and } |B\rangle \text{ has dimension } d_c^{2n+2m}.$$

Applying theorem 1 to the cut illustrated with the dashed line in equation (B1), we obtain that the states  $|\Psi_{A,B}^{\text{(PEPS)}}\rangle$  can be compressed into  $[(d_c^4 d_p - 1)\log(nm + d_c^4 d_p - 1) + (2n + 2m)\log d_c]$  qubits.

The same argument can be applied to a lattice of  $n$  site-independent correlated systems, each of which has physical dimension  $d_p$  and interacts with  $k$  neighbors. In this case, a generic state on the lattice can be compressed into  $[d_c^k d_p \log(n + d_c^k d_p - 1) + b \log d_c]$  qubits, where  $b$  is the boundary size, namely the number of correlation systems across the boundary ( $b = 0$  for closed lattices, like e.g. the torus).

### B.3. Multipartite states generated from a fixed state with the action of identical local unitary transformations

In this section and the next, we study the compression of state families generated from the action of identical local unitary transformations, namely a state in the form  $U_g^{\otimes n}|\Psi_0\rangle$ . We first consider the case where the initial state is fixed and known, and generalize it in the next section to the case where the initial state is a tensor network state family. Consider a parametric family of states of the form

$$|\Psi_g\rangle = U_g^{\otimes n}|\Psi_0\rangle \quad (\text{B2})$$

where  $|\Psi_0\rangle \in \mathcal{H}^{\otimes n}$  is a fixed pure state on  $n$  identical systems,  $g$  is an element of a group  $G$ , and  $U_g$  is a unitary operator belonging to a unitary representation of the group  $G$ . For example, the above states could describe the ground states of a system of  $n$  spins immersed in a uniform magnetic field of known intensity and unknown direction. All these states can be obtained from a fixed state (say, corresponding to a magnetic field oriented in the  $z$  direction) by rotating the direction of each spin by the same amount. A compression protocol would give a way to store the state of the spins in a quantum memory without knowing the direction.

To better characterize the structure of the transform  $U_g^{\otimes n}$ , we use the Schur–Weyl duality [67]. The Schur–Weyl duality decomposes the Hilbert space  $\mathcal{H}^{\otimes n}$  into the following form:

$$H^{\otimes n} \simeq \bigoplus_{\lambda \in Y_{n,d}} (\mathcal{R}_\lambda \otimes \mathcal{M}_\lambda) \quad (\text{B3})$$

where  $Y_{n,d}$  is the set of Young diagrams with  $n$  boxes and at most  $d$  rows, and  $\mathcal{R}_\lambda$  and  $\mathcal{M}_\lambda$  are certain subspaces indexed by  $\lambda$ . We denote the unitary transformation from the original  $n$ -tensor space to the decomposition as  $U_{\text{sch}}: \mathcal{H}^{\otimes n} \rightarrow \bigoplus_{\lambda \in Y_{n,d}} (\mathcal{R}_\lambda \otimes \mathcal{M}_\lambda)$ , which is known as the Schur transform. One property of the decomposition (B3) is that,  $U_g^{\otimes n}$  acts trivially on each subspace  $\mathcal{M}_\lambda$ . Therefore we can decompose  $U_g^{\otimes n}$  with respect to this decomposition as

$$U_{\text{sch}} U_g^{\otimes n} U_{\text{sch}}^\dagger = \sum_{\lambda \in Y_{n,d}} |\lambda\rangle\langle\lambda| \otimes U_{g,\lambda} \otimes I_{\mathcal{M}_\lambda} \quad (\text{B4})$$

where  $\{|\lambda\rangle\}_{\lambda \in Y_{n,d}}$  is an orthonormal basis that indexes the direct sum,  $U_{g,\lambda}$  is a unitary on  $\mathcal{R}_\lambda$ , and  $I_{\mathcal{M}_\lambda}$  is the identity on  $\mathcal{M}_\lambda$ . To match the decomposition of  $U_g^{\otimes n}$ , we also decompose  $|\Psi_0\rangle$  as

$$U_{\text{sch}}|\Psi_0\rangle = \sum_{\lambda \in Y_{n,d}} \xi_\lambda |\lambda\rangle \otimes |r_\lambda\rangle \otimes |\mu_\lambda\rangle \quad (\text{B5})$$

where  $|r_\lambda\rangle$  and  $|\mu_\lambda\rangle$  are states in  $\mathcal{R}_\lambda$  and  $\mathcal{M}_\lambda$ , respectively, and  $\sum_\lambda |\xi_\lambda|^2 = 1$ . Multiplying equation (B4) with equation (B5), we have

$$U_{\text{sch}} U_g^{\otimes n} |\Psi_0\rangle = \sum_{\lambda \in Y_{n,d}} \xi_\lambda |\lambda\rangle \otimes U_{g,\lambda} |r_\lambda\rangle \otimes |\mu_\lambda\rangle \quad (\text{B6})$$

Note that  $|\Psi_0\rangle$  is a known fixed state, and therefore  $|\mu_\lambda\rangle$  is known and fixed. We can then construct an isometry  $V_\mu: \text{Span}\{|\lambda\rangle\} \rightarrow \bigoplus_\lambda \mathcal{M}_\lambda$  that encodes the states  $\{|\mu_\lambda\rangle\}_{\lambda \in Y_{n,d}}$  as

$$V_\mu |\lambda\rangle = |\lambda\rangle \otimes |\mu_\lambda\rangle, \quad \forall \lambda \in Y_{n,d} \quad (\text{B7})$$

Defining  $|\psi_g\rangle = \sum_{\lambda \in Y_{n,d}} \xi_\lambda |\lambda\rangle \otimes U_{g,\lambda} |r_\lambda\rangle \in \bigoplus_\lambda \mathcal{R}_\lambda$ , we have  $U_{\text{sch}}^\dagger V_\mu |\psi_g\rangle = U_g^{\otimes n} |\Psi_0\rangle$ . We draw the tensor network generating the states  $\{U_g^{\otimes n} |\Psi_0\rangle\}$  in figure B3.

In figure B3,  $\tilde{U}_g = \sum_\lambda |\lambda\rangle\langle\lambda| \otimes U_{g,\lambda}$  is a compressed version of  $U_g^{\otimes n}$ . The  $T$ -shape intersection is a copying operation on the index system  $|\lambda\rangle$ , defined as

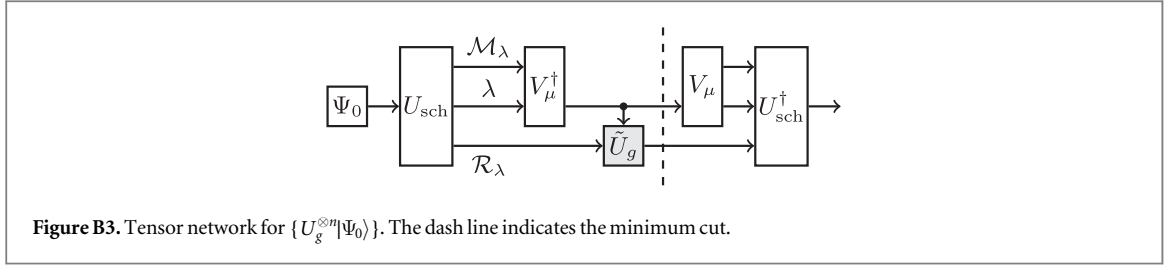


Figure B3. Tensor network for  $\{U_g^{\otimes n}|\Psi_0\rangle\}$ . The dash line indicates the minimum cut.

$$\begin{array}{c} \top \\ \downarrow \end{array} := \sum_{\lambda \in \mathcal{Y}_{n,d}} |\lambda\rangle |\lambda\rangle \langle \lambda| \quad (\text{B8})$$

The region to the left of the dashed line reads  $\tilde{U}_g(\sum_{\lambda} |\lambda\rangle |\lambda\rangle \langle \lambda|) V_\mu^\dagger U_{\text{sch}} |\Psi_0\rangle$ , which equals to  $|\psi_g\rangle$ . The partial isometry  $V_\mu^\dagger$  serves as a ‘coherent erasure’ of the system  $\mathcal{M}_\lambda$ . We first perform a coherent erasure on the multiplicity system  $\mathcal{M}_\lambda$  using  $V_\mu^\dagger$  and reprepare the system later using  $V_\mu$ . This effectively reduces the minimum cut (dashed line) of the tensor network. The cut edges constitute the Hilbert space  $\bigoplus_{\lambda \in \mathcal{Y}_{n,d}} \mathcal{R}_\lambda$ .

Note that we are slightly abusing the notations for tensor network, as the two cut edges constitute a space that is not the tensor product space of each edge. The dimension of the edge labeled by  $\mathcal{R}_\lambda$  depends on  $\lambda$ . To avoid ambiguity, we have to restrict the upper edge to take values only in the basis  $\{|\lambda\rangle\}$ .

We vectorise the variable tensor  $\tilde{U}_g$  as  $|\tilde{U}_g\rangle$ , and regard all other tensors as the tensor network  $N$ . Then we have  $|\Psi_g\rangle = N_* |\tilde{U}_g\rangle$ . Choosing the cut as the dashed line in figure B3, according to theorem 1, the states  $\{U_g^{\otimes n}|\Psi_0\rangle\}$  can be compressed into a memory of dimension equal to the dimension of  $\bigoplus_{\lambda \in \mathcal{Y}_{n,d}} \mathcal{R}_\lambda$ , which is

$$\dim \bigoplus_{\lambda \in \mathcal{Y}_{n,d}} \mathcal{R}_\lambda = \sum_{\lambda \in \mathcal{Y}_{n,d}} \dim \mathcal{R}_\lambda \leq (n+d-1)^{(d^2+d-2)/2} \quad (\text{B9})$$

namely a memory of no more than  $\left\lceil \frac{d^2+d-2}{2} \log(n+d-1) \right\rceil$  qubits. The last inequality comes from lemma 3 in [11] shown below, with  $r = d$ .

**Lemma 3.** *The total dimension of all the representation spaces corresponding to Young diagrams with no more than  $r$  rows is upper bounded as*

$$\sum_{\lambda \in \mathcal{Y}_{n,r}} \dim \mathcal{R}_\lambda \leq (n+d-1)^{(2dr-r^2+r-2)/2} \quad (\text{B10})$$

#### B.4. Parametric tensor network state family under identical local unitary transformations

In the previous section, we have discussed about the compression of states obtained from a fixed multipartite state under unknown local unitary transformations. Here we consider the generalization in which the unknown transformations are applied to a parametric family of tensor network states. This generalization could be used to treat the case of  $n$  of interacting spins with unknown couplings immersed in a uniform magnetic field of unknown direction.

Explicitly, we consider tensor network states of the form

$$|\Psi_{x,g}\rangle = U_g^{\otimes n} N_* |\nu_x\rangle \quad (\text{B11})$$

where  $|\nu_x\rangle \in \mathcal{H}_X$  is a vector in a suitable parameter space  $\mathcal{H}_X$ , and  $U_g$  is an unknown unitary transformation, representing the action of a group element  $g \in G$  on each physical system.

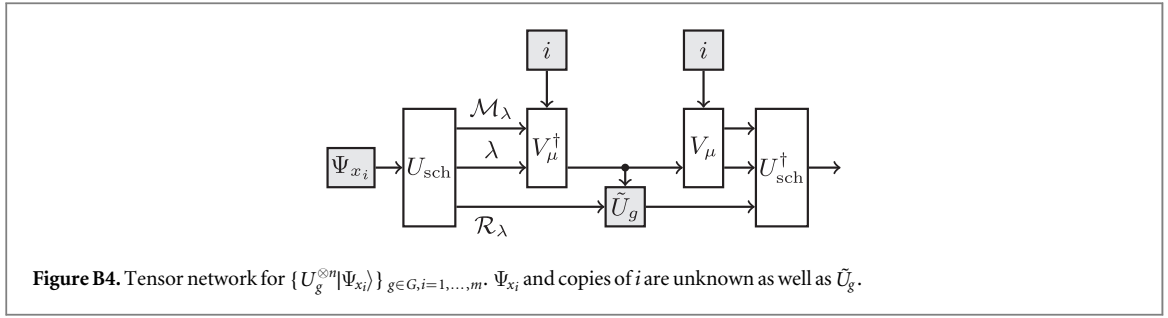
To use theorem 1, our goal is to construct a tensor network that generates the family  $\{U_g^{\otimes n} N_* |\nu_x\rangle\}_{g \in G, x \in X}$  with the property that the corresponding flow network has small minimum cut. We do the construction in two steps: we first consider a smaller state family and construct its corresponding tensor network, and then extend the network so that it generates our target state family  $\{U_g^{\otimes n} N_* |\nu_x\rangle\}_{g \in G, x \in X}$ .

Let  $m = \dim \mathcal{H}_X$ . Choose  $m$  values of the parameters  $x_1, \dots, x_m$  such that  $\{|\nu_{x_1}\rangle, \dots, |\nu_{x_m}\rangle\}$  is a basis of  $\mathcal{H}_X$ . The smaller family we consider is  $\{U_g^{\otimes n} |\Psi_{x_i}\rangle\}_{g \in G, i=1, \dots, m}$  with  $|\Psi_{x_i}\rangle := N_* |\nu_{x_i}\rangle$ . This family is an extension of the family in the previous section, where instead of fixing the initial state, the initial state is chosen from  $m$  alternatives.

For any  $x_i$ , using the Schur transform, we can decompose  $|\Psi_{x_i}\rangle$  as

$$U_{\text{sch}} |\Psi_{x_i}\rangle = \sum_{\lambda \in \mathcal{Y}_{n,d,p}} \xi_\lambda^{(x_i)} |\lambda\rangle \otimes |\tau_\lambda^{(x_i)}\rangle \otimes |\mu_\lambda^{(x_i)}\rangle \quad (\text{B12})$$

If  $|\mu_\lambda^{(x_i)}\rangle$  is known, as in the previous section, we can construct an isometry  $V_\mu$  (B7). However, in this case  $|\mu_\lambda^{(x_i)}\rangle$  is unknown and depends on the value of  $i$ . Thanks to the fact that  $i$  takes a finite number of values (1 to  $m$ ), we can construct one isometry for every value of  $i$ , in other words, an isometry controlled by  $i$ . As a result, we



**Figure B4.** Tensor network for  $\{U_g^{\otimes n}|\Psi_{x_i}\}_{g \in G, i=1, \dots, m}$ .  $\Psi_{x_i}$  and copies of  $|i\rangle$  are unknown as well as  $\tilde{U}_g$ .

redefine  $V_\mu$  as

$$V_\mu = \sum_{i=1}^m \left( \sum_{\lambda \in Y_{n,d_p}} |\lambda\rangle\langle\lambda| \otimes |\mu_\lambda^{(x_i)}\rangle \right) \langle i| \tag{B13}$$

where  $\{|i\rangle\}$  is a basis of an  $m$ -dimensional control system. And  $V_\mu^\dagger$  is defined as

$$V_\mu^\dagger = \sum_{i=1}^m \left( \sum_{\lambda \in Y_{n,d_p}} |\lambda\rangle\langle\lambda| \otimes \langle\mu_\lambda^{(x_i)}| \right) \langle i| \tag{B14}$$

where the control system is not transposed.

We then draw the tensor network that generates  $\{U_g^{\otimes n}|\Psi_{x_i}\}_{g \in G, i=1, \dots, m}$ , as shown in figure B4, which is similar to figure B3 with additional control systems for  $V_\mu$  and  $V_\mu^\dagger$ .

For any fixed  $i$ , this network reduces to figure B3, and generates the state  $U_g^{\otimes n}|\Psi_{x_i}\rangle$ . Therefore any state in the family  $\{U_g^{\otimes n}|\Psi_{x_i}\}_{g \in G, i=1, \dots, m}$  can be generated by this network. We regard  $|\Psi_{x_i}\rangle$ , two copies of  $|i\rangle$  and  $\tilde{U}_g$  as the parameters, and write  $U_g^{\otimes n}|\Psi_{x_i}\rangle = M_*|\Psi_{x_i}\rangle|i\rangle|\tilde{U}_g\rangle$ , where  $M_*$  is the linear operator represented by the constant tensors in network, and  $|\tilde{U}_g\rangle$  is the vectorised version of  $\tilde{U}_g$ .

Now we consider the original state family  $\{U_g^{\otimes n}|\Psi_x\rangle\}_{g \in G, x \in X}$ . Take any state  $U_g^{\otimes n}|\Psi_x\rangle = U_g^{\otimes n}N_*|v_x\rangle$  from the family, we can decompose  $|v_x\rangle$  in the basis  $\{|v_{x_i}\rangle\}$  as  $|v_x\rangle = \sum_i \alpha_i |v_{x_i}\rangle$ . Then the state  $U_g^{\otimes n}|\Psi_x\rangle$  can be written as a superposition of states in the smaller family:

$$U_g^{\otimes n}|\Psi_x\rangle = \sum_i \alpha_i U_g^{\otimes n}|\Psi_{x_i}\rangle \tag{B15}$$

This indicates that we can generate  $U_g^{\otimes n}|\Psi_x\rangle$  via the linear operator  $M_*$  with a superposition of the parameters. Defining

$$|\Phi_x\rangle := \sum_i \alpha_i |\Psi_{x_i}\rangle|i\rangle \tag{B16}$$

we have

$$U_g^{\otimes n}|\Psi_x\rangle = M_*(|\Phi_x\rangle \otimes |\tilde{U}_g\rangle) \tag{B17}$$

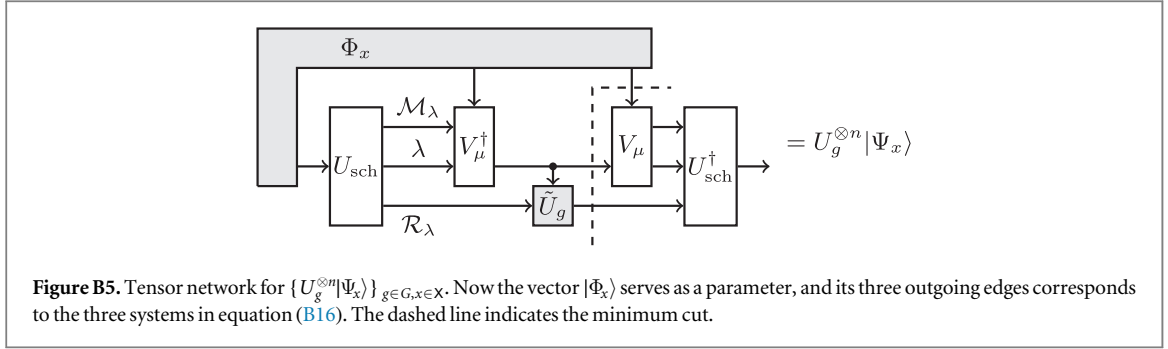
which shows that the family  $\{U_g^{\otimes n}|\Psi_x\rangle\}_{g \in G, x \in X}$  can be generated by the following tensor network in figure B5.

The lower two cut edges constitutes the space  $\bigoplus_{\lambda \in \mathcal{Y}_{n,d}} \mathcal{R}_\lambda$ , which has dimension no more than  $(n + d - 1)^{(d^2+d-2)/2}$  (lemma 3). The uppermost cut edge corresponds to the control system  $\text{Span}\{|i\rangle\}_{i=1}^m$  with dimension equal to  $\dim \mathcal{H}_X$ , and the combined dimension of all cut edges is  $(n + d - 1)^{(d^2+d-2)/2} \dim \mathcal{H}_X$ . Using theorem 1, we then obtain the memory size for compression the states  $\{U_g^{\otimes n}|\Psi_x\rangle\}_{g \in G, x \in X}$ , as stated in the following proposition.

**Proposition 6.** Consider a family of tensor network states  $\{|\Psi_x\rangle\}_{x \in X} \subset \mathcal{H}_d^{\otimes n}$  with parameter space  $\mathcal{H}_X$  (17). The state family generated by applying an unknown unitary transformation on all physical systems simultaneously, namely  $\{U_g^{\otimes n}|\Psi_x\rangle\}_{g \in G, x \in X}$ , can be compressed without error into a memory of no more than

$$\left\lceil \frac{d^2 + d - 2}{2} \log(n + d - 1) + \log \dim \mathcal{H}_X \right\rceil \text{ qubits.}$$

Note that to minimize the memory usage in proposition 6, one may choose an alternative parametrisation of the original family:  $|\Psi_x\rangle = N_*'|v_x'\rangle$ , where  $|v_x'\rangle$  resides in a space  $\mathcal{H}'_X$  that is smaller than  $\mathcal{H}_X$ . Specifically, the minimum cut  $C$  divides  $N$  into two subnetworks  $N_1$  and  $N_2$ , so that  $N_*$  is a concatenation of the corresponding



linear operators, namely  $N_* = N_{2*}N_{1*}$ , as shown in figure 4. Then by defining  $|v'_x\rangle = N_{1*}|v_x\rangle$ , we have  $|\Psi_x\rangle = N_{2*}|v'_x\rangle$ , and the new parameter space is  $\mathcal{H}_C$ , the combined Hilbert space of all cut edges. The dimension of  $\mathcal{H}_C$  can be smaller than the original parameter space  $\mathcal{H}_X$ . With the new parametrisation, when we apply proposition 6 for the states  $\{|\Psi_x\rangle\}_{g \in G, x \in X}$ , we use a memory of  $\left\lceil \frac{d^2 + d - 2}{2} \log(n + d - 1) + c(C) \right\rceil$  qubits, where  $c(C) = \log \dim \mathcal{H}_C$  is the capacity of the minimum cut.

We now consider MPSs and PEPSs. In section 4.1 and appendix B.1, we showed that an MPS or PEPS with unknown boundary condition can be written as  $|\Psi_B\rangle = N_*|B\rangle$ , where  $|B\rangle \in \mathcal{H}_X$  is a vector describing the boundary condition, and  $N_*$  is a linear operator.  $\mathcal{H}_X$  has dimension  $d_c^2$  for MPSs and  $d_c^{2n+2m}$  for PEPSs. We can then directly apply proposition 6 to the states  $\{U_g^{\otimes n}|\Psi_B\rangle\}$ . An MPS with variable boundary conditions under unknown transformation  $U_g^{\otimes n}$  can be compressed into a memory of  $\left\lceil \frac{d_p^2 + d_p - 2}{2} \log(n + d_p - 1) + 2 \log d_c \right\rceil$  qubits, while a PEPS on a square lattice with variable boundary condition under unknown transformation  $U_g^{\otimes nm}$  can be compressed into a memory of  $\left\lceil \frac{d_p^2 + d_p - 2}{2} \log(nm + d_p - 1) + (2n + 2m) \log d_c \right\rceil$  qubits.

### Appendix C. Proof of proposition 4

As illustrated in figure 6, let  $N_1$  and  $N_2$  be the subnetworks of  $N$  induced by the cut  $C$ . Then  $N_1$  defines a linear operator  $N_1: \mathcal{H}_X \rightarrow \mathcal{H}_E \otimes \mathcal{H}_C$ ,  $N_2$  defines a linear operator  $N_2: \mathcal{H}_C \rightarrow \mathcal{H}_P$ , and we have  $N = (N_2 \otimes I_E)N_1$ , where  $I_E$  is the identity operator on  $\mathcal{H}_E$ .

Let  $d_E = \dim \mathcal{H}_E$ , and take the computational basis  $\{|e_i\rangle\}_{i=1}^{d_E}$  of  $\mathcal{H}_E$ . Being the computational basis means  $|e_i\rangle = |\bar{e}_i\rangle$ . Define  $|\phi_{x,i}\rangle := (I_P \otimes \langle e_i|) |\Psi_x\rangle = (I_P \otimes \langle e_i|) N_* |v_x\rangle \in \mathcal{H}_P$ , so that  $|\Psi_x\rangle = \sum_{i=1}^{d_E} |\phi_{x,i}\rangle |e_i\rangle$ . Now we consider the compression for the (unnormalised) states  $\{|\phi_{x,i}\rangle\}_{x \in X, i=1, \dots, d_E}$ . In fact, this set of states is generated by the same network  $N$  by reversing the edges for the environment. This results in a network  $N'$ , compatible with the template  $\text{Temp}'$ . This is shown in figure C1.

We regard the cut  $C$  for network  $N$  also as a cut for  $N'$ . Let  $|\psi_{x,i}\rangle = N'_{1*} |v_x\rangle |e_i\rangle$ , and then  $|\phi_{x,i}\rangle = N_{2*} |\psi_{x,i}\rangle$ . Applying theorem 1 for  $\{|\phi_{x,i}\rangle\}$ , there exists a partial isometry  $V: \mathcal{H}_P \rightarrow \mathcal{H}_C$  such that

$$V^\dagger V |\phi_{x,i}\rangle = |\phi_{x,i}\rangle, \quad \forall x \in X, i = 1, \dots, d_E \quad (\text{C1})$$

This  $V$  is what we want. For all  $x \in X$ ,

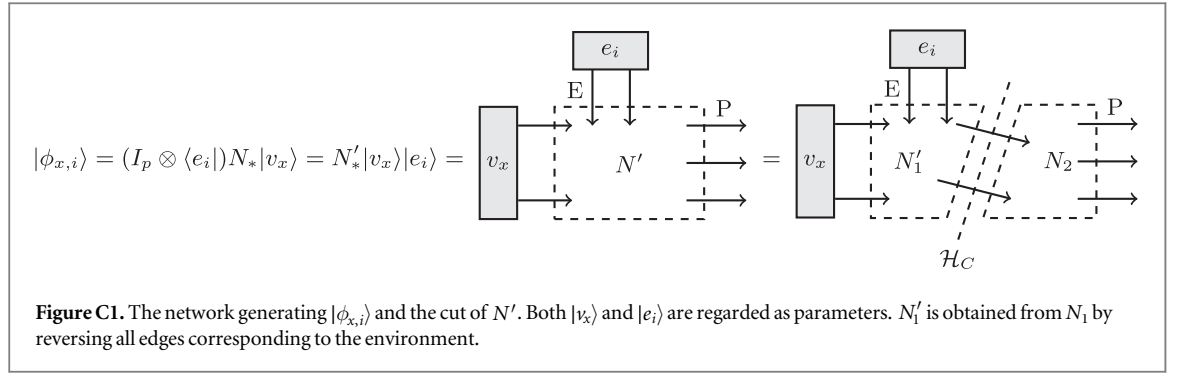
$$(V^\dagger V \otimes I_E) |\Psi_x\rangle = (V^\dagger V \otimes I_E) \sum_{i=1}^{d_E} |\phi_{x,i}\rangle |e_i\rangle \quad (\text{C2})$$

$$= \sum_{i=1}^{d_E} (V^\dagger V |\phi_{x,i}\rangle) |e_i\rangle \quad (\text{C3})$$

$$= \sum_{i=1}^{d_E} |\phi_{x,i}\rangle |e_i\rangle \quad (\text{C4})$$

$$= |\Psi_x\rangle \quad (\text{C5})$$

□



## Appendix D. Proof of lemma 1

**Proof.** For concreteness, we first prove the lemma for MPSs with open boundary conditions, and then show how to generalize the proof to arbitrary boundary conditions. For simplicity, we present the proof with SIMPs, while the proof works for general MPSs.

We encode the coefficients into a vector  $|c\rangle := \sum_{k=1}^t c_k |k\rangle$ . We assume  $|c\rangle$  is normalized such that the linear combination  $|\Psi\rangle = \sum_k c_k |\Sigma_k\rangle$  has unit length. Since  $\{|\Sigma_k\rangle\}$  are MPSs, they have the following form:

$$|\Sigma_k\rangle = \begin{array}{c} \uparrow \\ \boxed{L_k} \leftarrow \boxed{A_k^{[1]}} \leftarrow \boxed{A_k^{[2]}} \leftarrow \dots \leftarrow \boxed{A_k^{[n]}} \leftarrow \boxed{R_k} \\ \uparrow \end{array} \quad (\text{D1})$$

To represent their linear combination, we define tensors  $\Lambda$ ,  $B^{[1]}, \dots, B^{[n]}$ , and  $P$  such that

$$\begin{array}{c} \uparrow \\ \boxed{\Lambda} \leftarrow \boxed{L_k} \\ \uparrow \\ k \end{array} = \begin{array}{c} \uparrow \\ \boxed{B^{[i]}} \leftarrow \boxed{A_k^{[i]}} \\ \uparrow \\ k \end{array}, \quad \begin{array}{c} \uparrow \\ \boxed{P} \leftarrow \boxed{R_k} \\ \uparrow \\ k \end{array} \quad (\text{D2})$$

for every  $k \in \{1, \dots, t\}$  and  $i \in \{1, \dots, n\}$ . Then  $|\Sigma\rangle$  can be represented as:

$$|\Psi\rangle = \begin{array}{c} \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \\ \boxed{\Lambda} \leftarrow \boxed{B^{[1]}} \leftarrow \boxed{B^{[2]}} \leftarrow \dots \leftarrow \boxed{B^{[n]}} \leftarrow \boxed{P} \\ \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \\ \boxed{c} \end{array} \quad (\text{D3})$$

where  $\begin{array}{c} \uparrow \\ \text{---} \end{array}$  is the tensor defined by

$$\begin{array}{c} \uparrow \\ \text{---} \end{array} := \sum_{k=1}^t |k\rangle |k\rangle \langle k| \quad (\text{D4})$$

Equation (D3) shows that  $|\Psi\rangle$  is an MPS: the left boundary condition is to the left of the first dotted line, the right boundary condition is to the right of the second dotted line, and the tensor on each physical system is  $B^{[i]}$  connected with a  $T$ -intersection (D4). There are two edges connecting consecutive physical systems, one has dimension  $d_c$ , and the other one has dimension  $t$ . Therefore the bond dimension of  $|\Psi\rangle$  is  $td_c$ .

For MPSs with general boundary conditions, one just needs to replace the boundary conditions by a suitable tensor connecting  $A_k^{[1]}$  and  $A_k^{[n]}$  ( $B^{[1]}$  and  $B^{[n]}$ ), and the rest of the proof is identical.

To conclude the proof, we invoke the fact that all MPSs with polynomial-size bond dimension can be prepared in polynomial time on a quantum computer [43, 58].  $\square$

## Appendix E. Proof of lemma 2

The channel  $\mathcal{R}$  can be expressed as

$$R = \frac{1}{r} \sum_{l=0}^{r-1} R_l, \quad R_l(\rho) := \frac{1}{6} \sum_{\alpha \in \{x,y,z\}} \sum_{s \in \{+,-\}} U_{l,\alpha,s} \rho U_{l,\alpha,s}^\dagger \quad U_{l,\alpha,\pm} = I_{\text{in}} - 2|\Psi_{l,\alpha,\pm}\rangle \langle \Psi_{l,\alpha,\pm}| \quad (\text{E1})$$



Let  $\mathcal{S}_l := \text{Span}\{|l\rangle, |l \oplus 1\rangle\}$  and  $\mathcal{S}_l^\perp$  be its orthogonal complement. For each  $l$ , both  $\mathcal{S}_l$  and  $\mathcal{S}_l^\perp$  are invariant subspaces of channel  $\mathcal{R}_l$ . For  $\rho$  with support only in  $\mathcal{S}_l$ , one can explicitly determine  $\mathcal{R}_l$  as

$$R_l(\rho) = \frac{4}{3} \frac{P_l}{2} \text{Tr}[\rho] - \frac{1}{3} \rho, \quad \forall \rho, \text{Supp}(\rho) \subseteq \mathcal{S}_l \quad (\text{E2})$$

where  $P_l$  is the projector on  $\mathcal{S}_l$ . For  $\rho$  with support only in  $\mathcal{S}_l^\perp$ , the channel  $\mathcal{R}_l$  is just the identity. For a general  $\rho$ , one can decompose it as  $\rho = P_l \rho P_l + P_l^\perp \rho P_l^\perp + \rho'$ , where  $P_l^\perp$  is the projector on  $\mathcal{S}_l^\perp$ , and  $\rho' := P_l \rho P_l^\perp + P_l^\perp \rho P_l$  contains the off-block-diagonal terms. We further observe that  $\mathcal{R}_l(\rho') = 0$ , because for any off-block-diagonal element  $\sigma = |j\rangle\langle k|$  (or  $\sigma = |k\rangle\langle j|$ ) with  $j \in \{l, l+1\}$  and  $k \notin \{l, l+1\}$ , we have  $U_{l,\alpha,\sigma} U_{l,\alpha,\sigma}^\dagger + U_{l,\alpha,-\sigma} U_{l,\alpha,-\sigma}^\dagger = 0$ ,  $\forall \alpha \in \{x, y, z\}$ , and thus  $\mathcal{R}_l(\sigma) = 0$ . We conclude that  $\mathcal{R}_l(\rho) = \mathcal{R}_l(P_l \rho P_l) + \mathcal{R}_l(P_l^\perp \rho P_l^\perp)$ , and therefore for general  $\rho \in \mathcal{H}_{\text{in}}$ ,

$$R_l(\rho) = \frac{4}{3} \frac{P_l}{2} \text{Tr}[P_l \rho] - \frac{1}{3} P_l \rho P_l + P_l^\perp \rho P_l^\perp \quad (\text{E3})$$

To find the eigenvalues of the channel  $\mathcal{R}$ , we use the double-ket notation  $|A\rangle\rangle := \sum_{j,k} A_{jk} |j\rangle \otimes |k\rangle$ , representing linear operators on  $\mathcal{H}_{\text{in}}$  as vectors in the tensor product space  $\mathcal{H}_{\text{in}} \otimes \mathcal{H}_{\text{in}}$ . Using this notation, the eigenvalue equation  $\mathcal{R}(A) = \lambda A$  becomes  $\check{R}|A\rangle\rangle = \lambda |A\rangle\rangle$ , with

$$\check{R} := \frac{1}{r} \sum_{l=0}^{r-1} \check{R}_l, \quad \check{R}_l := \frac{4}{3} \frac{|P_l\rangle\rangle\langle\langle P_l|}{2} - \frac{1}{3} P_l \otimes P_l + P_l^\perp \otimes P_l^\perp \quad (\text{E4})$$

Averaging over  $l$ , we finally obtain

$$\check{R} = \check{R}_1 + \check{R}_2 + \check{R}_3 \quad (\text{E5})$$

where

$$\check{R}_1 := \frac{8}{3r} \sum_{l=0}^{r-1} |l\rangle\langle l| \otimes |l\rangle\langle l| + \frac{2}{3r} \sum_{l=0}^{r-1} (|l\rangle\langle l \oplus 1| \otimes |l\rangle\langle l \oplus 1| + |l \oplus 1\rangle\langle l| \otimes |l \oplus 1\rangle\langle l|) \quad (\text{E6})$$

$$\check{R}_2 := \frac{2}{3r} \sum_{l=0}^{r-1} |l\rangle\langle l| \otimes |l \oplus 1\rangle\langle l \oplus 1| + \frac{2}{3r} \sum_{l=0}^{r-1} |l \oplus 1\rangle\langle l \oplus 1| \otimes |l\rangle\langle l| \quad (\text{E7})$$

$$\check{R}_3 := \left(1 - \frac{4}{r}\right) (I_{\text{in}} \otimes I_{\text{in}}) \quad (\text{E8})$$

$\check{R}_3$  is proportional to the identity and does not contribute to the spectral gap. From  $\check{R}_1 \check{R}_2 = \check{R}_2 \check{R}_1 = 0$ , the supports of  $\check{R}_1$  and  $\check{R}_2$  are orthogonal subspaces, so we can consider the eigendecompositions of  $\check{R}_1$  and  $\check{R}_2$  separately. The spectral gap would then equals to the difference between the largest and second largest among the union of eigenvalues of  $\check{R}_1$  and  $\check{R}_2$ . Notice that the support of  $\check{R}_1$  is in the subspace  $\text{Span}\{|l\rangle \otimes |l\rangle\}_{l=0}^{r-1}$ . Under the basis  $\{|l\rangle \otimes |l\rangle\}_{l=0}^{r-1}$ ,  $\check{R}_1$  is a Toeplitz matrix whose eigendecomposition has a simple form [68]:

$$\check{R}_1 = \frac{4}{r} \sum_{k=0}^{r-1} \frac{2 + \cos \frac{2\pi k}{r}}{3} |\Phi_k\rangle\langle\Phi_k| \quad (\text{E9})$$

where each eigenvector  $|\Phi_k\rangle$  is the Fourier vector defined as  $|\Phi_k\rangle := \sum_l e^{2\pi i k l / d_{\text{in}}} |l\rangle \otimes |l\rangle / \sqrt{d_{\text{in}}}$ . Among the eigenvalues in equation (E9), the largest eigenvalue is  $\frac{4}{r}$ , with eigenvector  $|\Phi_0\rangle$ . The second largest is  $\frac{4}{r} \frac{2 + \cos \frac{2\pi k}{r}}{3}$  with eigenvectors  $|\Phi_1\rangle$  and  $|\Phi_{r-1}\rangle$ . Now we turn to  $\check{R}_2$ , and observe that its only eigenvalue is  $\frac{2}{3r}$ , which is smaller than the second largest eigenvalue of  $\check{R}_1$ . We therefore conclude that the spectral gap equals to the difference between the two largest eigenvalues of  $\check{R}_1$ , which is

$$\gamma_R = \frac{4}{r} - \frac{4}{r} \left( \frac{2 + \cos \frac{2\pi}{3}}{3} \right) = \frac{8 \left( \sin \frac{\pi}{r} \right)^2}{3r} \quad (\text{E10})$$

□

## Appendix F. Proof of proposition 5

The key of the proof is to show that Conditions (i)–(iv) guarantee that every linear combination  $\sum_k c_k |\Sigma_k\rangle$  with efficiently computable coefficients  $\{c_k\}$  is efficiently preparable. Once this is done, we can simply construct an orthonormal basis from the states  $\{|\Sigma_k\rangle\}$ , and use the fiducial states in equation (49).

Any vector  $|\Psi\rangle \in \mathcal{H}_{\text{in}}$  can be decomposed as  $|\Psi\rangle = \sum_k c_k |\Sigma_k\rangle$ , with  $c_k = \langle \Sigma_k | F^{-1} |\Psi\rangle$  and  $F = \sum_k |\Sigma_k\rangle\langle \Sigma_k|$  [69]. Now, let  $A$  be an auxiliary quantum system of dimension  $s$ , and consider the state

$$|\psi\rangle = \frac{\sum_k c_k |k\rangle}{\sqrt{\sum_k |c_k|^2}} \quad (\text{F1})$$

For given coefficients  $\{c_k\}$ , the state  $|\psi\rangle$  is efficiently preparable, because it is a state in a Hilbert space of polynomial dimension.

Now, consider a coherent control mechanism that prepares the state  $|\Sigma_k\rangle$  by a coherent process  $|k\rangle \mapsto |k\rangle \otimes |\Sigma_k\rangle$ , where  $\{|k\rangle\}$  is a basis of an auxiliary system. Setting the auxiliary system to  $|\psi\rangle$ , the resulting state is

$$|\Gamma\rangle = \frac{\sum_k c_k |k\rangle \otimes |\Sigma_k\rangle}{\sqrt{\sum_k |c_k|^2}} \quad (\text{F2})$$

Finally, projecting the auxiliary system on the vector  $\sum_k |k\rangle / \sqrt{s}$ , one obtains the state  $|\Psi\rangle$ . The probability that the projection takes place is  $p_\Psi = (s \sum_k |c_k|^2)^{-1} = (s \langle \Psi | F^{-1} | \Psi \rangle)^{-1} \geq 1/(s \lambda_{\min})$ , where  $\lambda_{\min}$  is the minimum non-zero eigenvalue of  $F$ . Note that the eigenvalues of  $F$  are the same as the eigenvalues of the Gram matrix  $G = \langle \Sigma_k | \Sigma_l \rangle$ . This is the case because one has  $F = XX^\dagger$  and  $G = X^\dagger X$ , with  $X = \sum_k |\Sigma_k\rangle \langle k|$ . Since  $\lambda_{\min}$  is assumed to be at least inverse polynomial, the probability is guaranteed to be at least inverse polynomial. This means that a polynomial number of repetitions of the above procedure are sufficient to generate the state  $|\Psi\rangle$  with probability close to 1.  $\square$

## ORCID iDs

Ge Bai  <https://orcid.org/0000-0002-6814-8840>

Yuxiang Yang  <https://orcid.org/0000-0002-0531-8929>

Giulio Chiribella  <https://orcid.org/0000-0002-1339-0656>

## References

- [1] Schumacher B 1995 *Phys. Rev. A* **51** 2738
- [2] Jozsa R and Schumacher B 1994 *J. Mod. Opt.* **41** 2343
- [3] Beals R, Brierley S, Gray O, Harrow A W, Kutin S, Linden N, Shepherd D and Stather M 2013 *Proc. R. Soc. A* **469** 20120686
- [4] Barz S, Kashefi E, Broadbent A, Fitzsimons J F, Zeilinger A and Walther P 2012 *Science* **335** 303
- [5] Lo H-K 1995 *Opt. Commun.* **119** 552
- [6] Horodecki M 1998 *Phys. Rev. A* **57** 3364
- [7] Barnum H, Caves C M, Fuchs C A, Jozsa R and Schumacher B 2001 *J. Phys. A: Math. Gen.* **34** 6767
- [8] Jozsa R, Horodecki M, Horodecki P and Horodecki R 1998 *Phys. Rev. Lett.* **81** 1714
- [9] Plesch M and Bužek V 2010 *Phys. Rev. A* **81** 032317
- [10] Chiribella G, Yang Y and Huang C 2015 *Phys. Rev. Lett.* **114** 120504
- [11] Yang Y, Chiribella G and Ebler D 2016 *Phys. Rev. Lett.* **116** 080501
- [12] Yang Y, Chiribella G and Hayashi M 2016 *Phys. Rev. Lett.* **117** 090502
- [13] Yang Y, Bai G, Chiribella G and Hayashi M 2018 *IEEE Trans. Inf. Theory* **64** 4766–83
- [14] D'Ariano G and Presti P L 2001 *Phys. Rev. Lett.* **86** 4195
- [15] D'Ariano G M, Paris M G and Sacchi M F 2003 *Adv. Imaging Electron Phys.* **128** 205–308
- [16] Gisin N and Massar S 1997 *Phys. Rev. Lett.* **79** 2153
- [17] Bruß D, DiVincenzo D P, Ekert A, Fuchs C A, Macchiavello C and Smolin J A 1998 *Phys. Rev. A* **57** 2368
- [18] Helstrom C W 1969 *J. Stat. Phys.* **1** 231
- [19] Holevo A S 2011 *Probabilistic and Statistical Aspects of Quantum Theory* vol 1 (Amsterdam: North Holland)
- [20] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631
- [21] Yang Y, Chiribella G and Hayashi M 2018 *Proc. R. Soc. A* **474** 20170773
- [22] Yang Y, Chiribella G and Hayashi M 2019 *Commun. Math. Phys.* **368** 223
- [23] Rozema L A, Mahler D H, Hayat A, Turner P S and Steinberg A M 2014 *Phys. Rev. Lett.* **113** 160504
- [24] Kraus B 2011 *Phys. Rev. Lett.* **107** 250503
- [25] Boyajian W L, Murg V and Kraus B 2013 *Phys. Rev. A* **88** 052329
- [26] Boyajian W L and Kraus B 2015 *Phys. Rev. A* **92** 032323
- [27] Li Z, Zhou H, Ju C, Chen H, Zheng W, Lu D, Rong X, Duan C, Peng X and Du J 2014 *Phys. Rev. Lett.* **112** 220501
- [28] Beau M and del Campo A 2017 *Phys. Rev. Lett.* **119** 010403
- [29] Czajkowski J, Pawłowski K and Demkowicz-Dobrzański R 2019 *New J. Phys.* **21** 053031
- [30] Briegel H J and Raussendorf R 2001 *Phys. Rev. Lett.* **86** 910
- [31] Raussendorf R and Briegel H J 2001 *Phys. Rev. Lett.* **86** 5188
- [32] Fannes M, Nachtergaele B and Werner R F 1992 *Commun. Math. Phys.* **144** 443
- [33] Perez-Garcia D, Verstraete F, Wolf M M and Cirac J I 2007 *Quantum Inf. Comp.* **7** 401–30
- [34] Verstraete F and Cirac J I 2006 *Phys. Rev. B* **73** 094423
- [35] Verstraete F and Cirac J I 2004 arXiv:cond-mat/0407066
- [36] Verstraete F, Wolf M M, Perez-Garcia D and Cirac J I 2006 *Phys. Rev. Lett.* **96** 220601
- [37] Shi Y-Y, Duan L-M and Vidal G 2006 *Phys. Rev. A* **74** 022320
- [38] Vidal G 2008 *Phys. Rev. Lett.* **101** 110501
- [39] Ising E 1925 *Z. Phys.* **31** 253
- [40] Affleck I, Kennedy T, Lieb E H and Tasaki H 2004 *Condensed Matter Physics and Exactly Soluble Models* (Berlin: Springer) pp 249–52

- [41] Bacon D, Chuang I L and Harrow A W 2006 *Phys. Rev. Lett.* **97** 170502
- [42] Bacon D, Chuang I L and Harrow A W 2007 *Proc. 18th Annual ACM-SIAM Symp. on Discrete Algorithms* (Philadelphia, PA: SIAM) pp 1235–44
- [43] Cirac J I and Verstraete F 2009 *J. Phys. A: Math. Theor.* **42** 504004
- [44] Singh S, Pfeifer R N and Vidal G 2010 *Phys. Rev. A* **82** 050301
- [45] Singh S, Pfeifer R N and Vidal G 2011 *Phys. Rev. B* **83** 115125
- [46] Cui S X, Freedman M H, Sattath O, Stong R and Minton G 2016 *J. Math. Phys.* **57** 062206
- [47] Gross D, Eisert J, Schuch N and Perez-Garcia D 2007 *Phys. Rev. A* **76** 052315
- [48] Cormen T H, Leiserson C E, Rivest R L and Stein C 2009 *Introduction to Algorithms* (Cambridge, MA: MIT Press)
- [49] Ford L R Jr and Fulkerson D R 2015 *Flows in Networks* (Princeton, NJ: Princeton University Press)
- [50] Hastings M B 2007 *J. Stat. Mech.* **2007** P08024
- [51] Arad I, Kitaev A, Landau Z and Vazirani U 2013 arXiv:1301.1162
- [52] Huang Y 2014 arXiv:1403.0327
- [53] Marvian I and Lloyd S 2016 arXiv:1606.02734
- [54] Burgarth D and Giovannetti V 2007 *New J. Phys.* **9** 150
- [55] Burgarth D, Chiribella G, Giovannetti V, Perinotti P and Yuasa K 2013 *New J. Phys.* **15** 073045
- [56] Cozzini M, Ionicioiu R and Zanardi P 2007 *Phys. Rev. B* **76** 104420
- [57] Verstraete F, Murg V and Cirac J I 2008 *Adv. Phys.* **57** 143
- [58] Eisert J 2013 arXiv:1308.3318
- [59] Demmel J W 1997 *Applied Numerical Linear Algebra* vol 56 (Philadelphia, PA: SIAM)
- [60] Kitaev A Y 1997 *Russ. Math. Surv.* **52** 1191
- [61] Kitaev A Y 2003 *Ann. Phys.* **303** 2
- [62] Bravyi S B and Kitaev A Y 1998 arXiv:quant-ph/9811052
- [63] Pastawski F, Yoshida B, Harlow D and Preskill J 2015 *J. High Energy Phys.* **JHEP06(2015)149**
- [64] Latorre J I and Sierra G 2015 arXiv:1502.06618
- [65] Cirac J I, Poilblanc D, Schuch N and Verstraete F 2011 *Phys. Rev. B* **83** 245134
- [66] Wilming H and Eisert J 2019 *Phys. Rev. Lett.* **122** 190501
- [67] Fulton W and Harris J 2013 *Representation Theory: A First Course* vol 129 (New York: Springer)
- [68] Gray R M et al 2006 *Found. Trends® Commun. Inf. Theory* **2** 155
- [69] Casazza P G and Christensen O 1998 *SIAM J. Math. Anal.* **29** 266