Tensor network representation of many-body quantum states and unitary operators

Thesis by

Sujeet Kumar Shukla

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2018
Defended July 21, 2017

To my parents

ACKNOWLEDGEMENTS

I am deeply thankful to my adviser John Preskill for his continuous support and guidance. I am especially indebted for the academic freedom he granted me in pursuing my interests. I could not have hoped for a better research environment and a better graduate experience.

None of the work in this thesis would have been possible without Xie Chen, who I had the honor to closely collaborate with in the later years of my graduate life. I am deeply thankful to her for her guidance and numerous helpful discussions we had over the years. She inspired and helped me formulate and shape my own research style, for which I would be forever grateful.

I am very grateful to Liang Jiang, who I had the honor to collaborate with in my earlier graduate years. He inspired me to face challenging problems with patience and rigor, which had helped me ever since.

A special thanks goes to my friend and collaborator M. Burak Şahinoğlu. A significant part of this thesis is based on this collaboration.

I also had the honor to have numerous helpful discussions with former and current Caltech students, postdocs and faculty members over the years who constantly challenged my ideas and concepts while introducing me to new ones. In particular I would like to thank Alexei Kitaev, Beni Yoshida,Fernando Pastawski, Evgeny Mozgunov (Jenia) , Alex Kubica, Michael Beverland, Jeongwan Haah, David Aasen, Feng Bi, Matthew Fishman, Anna Komar, Paraj Titum, Gaurav Sinha,Tejas Deshpande and Spyridon Michalakis.

No journey can be completed without people who keep us inspired along the way. For that, I would like to thank, Pinaky Bhattacharyya, Nisha Mohan and Vikas Trivedi (the *Winchell group*). Pinaky also helped me develop my understanding of complex numerical calculations over the years, and many parts this thesis would not have been possible without that skill.

I would also like to thank many friends I made along the way– Lucy Zhang, Tom Che, Ramya Vinayak, Priya Choudhary, Parul Kudtarkar, Tri Vu and Brian Skinner– who made this journey not only meaningful but also enjoyable.

And last, but not the least, I would like to thank my family and my wife, Vani, for their unwavering and unconditional support throughout my graduate years, without

which none of this would have been possible.

ABSTRACT

Understanding many-body quantum systems is one of the most challenging problems in contemporary condensed-matter physics. Tensor network representation of quantum states and operators are taking central stage in this pursuit and beyond. They prove to be a powerful numerical and conceptual tool, and indeed a new language altogether. This thesis investigates various aspects of these representations by focusing on two specific problems: the first half of the thesis is devoted to examining how ‘stable’ a tensor network representation is for two-dimensional quantum states with topological order, and the second half explores the representability of various unitary loop operators with tensor networks.

In the numerical usage of the tensor networks, the tensor is varied as to find the representation of the ground states of the given Hamiltonian. In chapter two and three of this thesis we show that such a numerical program for topological phases can be ‘ill-posed’. We show that tensor network can be an unstable representation for a topological phase: even an infinitesimal variation in the representation results in the loss of topological order, completely or partially. We diagnose this problem by identifying the exact causes of this instability, and find that it is only tensor variations in certain directions that result in instability, because they result in the condensation of bosonic quasi-particles of the phase. Such unstable variations are characterized by two properties: (1) they can replace a tensor in the tensor network without making the network collapse, and (2) their presence in the network represents the presence of a non-trivial topological charge. We prove that the general tensor representation of all string-net models suffer with such instabilities. We propose an exact mathematical operator to project out all such unstable variations and show its efficacy for a few models by direct calculations. Such an operator can be useful in numerical programs involving such tensor representations. We also point out that such variations play a crucial role in simulating topological phase transitions and their presence can be vital in an accurate simulation.

In chapter four and five of this thesis we focus on the representability of unitary loop operators by tensor networks. Such operators not only provide an important tool in the study of dynamical process in one-dimensional systems, but also in understanding and classification of symmetry protected topological phases in two dimensions. To characterize all such operators, we find a necessary and sufficient condition for any loop tensor network operator of a given length to represent a unitary operator. In particular, it is shown that all unitary operators that map local operators to local operators (locality-preserving) can always be represented by a tensor network. Locality-preserving unitary loop operators are classified by a rational index called the GNVW index defined in Ref. [1] which measures how much information ‘flows’ along the loop. We define Rank-Ratio index for tensor network operators and show that it is completely equivalent to the GNVW index. Therefore, GNVW index of a unitary operator can be easily extracted from its tensor network representation. We find that, other than representing locality-preserving unitary maps, tensor networks can also represent unitary operators that map local operators to global (non-local) operators. These tensor network operators are found to have a long-ranged order similar to tensors that represent topological tensor network states in two dimensions.

PUBLISHED CONTENT AND CONTRIBUTIONS

1. S. K. Shukla, M. Burak Şahinoğlu, F. Pollmann, and X. Chen. *Boson condensation and instability in the tensor network representation of string-net states.* In: ArXiv:1610.00608

Chapter 2 and 3 are based on this work, of which I was the main contributor.

2. Burak Şahinoğlu, S. K. Shukla, F. Bi, and X. Chen. M*atrix Product Representation of Locality Preserving Unitaries.* In: ArXiv: 1704.01943

Chapter 4 of this thesis is based on this work. I defined the central index of paper, and contributed to different mathematical proofs to varying degree. I discovered the counter-intuitive example discussed in the last section. Chapter 5 of this thesis is inspired by this example.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

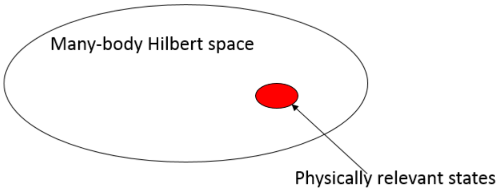
Understanding of quantum many-body systems is one of the central challenges of modern low-energy physics. Quantum many-body systems provide a rich source of unique and exotic physical phenomena. We will enlist here just a few of such systems which have seen a lot of theoretical, numerical, and experimental activities in recent years. This list, though by no means exhaustive, is enough to put forward the motivations underlying the work in this thesis.

One of the most famous such phenomena is the existence of high-temperature superconductivity, a complete understanding of which remains somewhat elusive to this day[2]. Chiral-spin liquids were proposed[3, 4] in an attempt to explain high-temperature superconductivity, but it was later realized that they actually exhibit a new kind of order, beyond Landau's symmetry breaking paradigm. It was called a 'topological order'. These phases exhibit a ground state degeneracy[5] which only depends on the topology of the underlying system. The experimental discovery of Fractional Quantum Hall (FQH) systems[6, 7] became a well known experimental realizations of such an order. They were called *anyons*. From the study of these phases the idea of a 'topological order' in quantum many-body systems emerged[8]. Other than the FQH, some spin liquids were also found to exhibit topological order[9]. These systems were shown to have quasi-particle excitations that were neither bosonic nor fermionic, but had fractional statistics[10, 11]. Much of the formulation of these systems was proposed in field theoretic terms. However, Kitaev[12], later proposed a class of exactly solvable topological lattice models, now known as Quantum Double models. These models allowed for a deeper investigation of topological phases both conceptually and numerically. Later on Quantum Doubles model were generalized into the so-called String-Net models by Levin and Wen [13].

After the discovery of topological phases the notion of Symmetry Protected Topological (SPT) phases [14–18] emerged with the discovery of Topological Insulators[19–21]. These phases were characterized by the presence of a symmetry in the system that protected the topological order. That is, different phases of an SPT could be smoothly connected if the underlying symmetry was allowed to be broken. For example, topological insulators were found to be protected by time-reversal symmetry. SPT phases were later generalized into many directions, such as Symmetry Enriched Topological phases[22], SPT with n-form symmetries etc.

What underlies all these different phenomena in a quantum-many body system is the pattern of entanglement individual constituents of the system have with each other. So, naturally, a paradigm of representation was needed which could encode the entanglement information of the system in a more explicit and efficient way than the traditional representation did. Traditionally, the quantum states have been represented as a superposition of conveniently chosen basis states in the Hilbert space of the system. Though it works well for small quantum systems such as an atom or small spin systems, it is not particularly useful for large quantum-systems with exponentially large Hilbert space. Broadly speaking, there are two problems with basis-superposition representation. First, the number of basis vectors needed for this representation scales exponentially with the system size. For example, for a system with N qubits, one needs 2^N basis states to represent a generic quantum wave function. To get an idea, a system with 1000 qubits would need $2^{1000} \approx 10^{300}$ basis for description. This number is much larger than the number of atoms in the known universe. So we could not possibly hope to work with these many basis states in any present day computation system. Second, and more fundamentally, it is very difficult to read off entanglement patterns in a state from its basis-superposition representation. It just happens to be not the natural representation as far as describing entanglement in a system is concerned. From this need to extract information about entanglement patters, and various limitation of basis-superposition representations, emerged the development of a new paradigm of representation of quantum systems: the Tensor Network (TN) representation.

But before we introduce what a TN representation is, we ask, why do we expect to find any paradigm to be able to represent many-body states efficiently? After all, any representation does not take away the underlying complexity problem: the many-body Hilbert is hopelessly big! A system of the size of Avogrado number ($\sim 10^{23}$) has a Hilbert space of $\sim 10^{10^{23}}$. This is certainly beyond the reach of any representation possible. What rescues a physicist from this problem is the fact that not all states in the Hilbert space are physically relevant. Physics imposes many constraint on the mathematical description of the system. One such key constraint is the notion of locality. Many relevant Hamiltonians that describe dynamics of a system found in nature tend to only involve local interactions. Mathematically speaking, it translates to the fact that we are mostly, at least as far as low-energy physics is concerned, interested in Hamiltonians which can be written as a sum of local terms, where each term only acts on a local subsystem. If we also restrict ourselves to 'gapped' systems (which roughly means that the excitations have certain minimum mass) then one can prove that low-energy states of such a system all belong to a very small subspace of the whole Hilbert space: the states that follow the so-called *area-law*. Area-law dictates that, for low-energy eigenstates of gapped, local Hamiltonians, the entanglement entropy of a region of space tends to scale, as the size of the boundary of the region and not as the volume. For example, in 1D it means that a line segment of the space has a constant entanglement entropy. Area-law states reside in a small subspace (exponentially small) of the whole Hilbert space. It can also be shown that not only do physically relevant states belong to an exponentially small subspace of the whole Hilbert space, any state that can be reached under Hamiltonian time evolution in a reasonable time (log(N), to be more specific where N is system size) also belongs to such an exponentially small subspace. Overall one can say that we can expect to find an efficient representation of many-body states in many cases simply because the physically relevant states for gapped phases of matter belong to an exponentially small subspace of the whole Hilbert space,



.

So it is desirable that we invent a representation paradigm that is more restricted to the physically relevant subspace, rather than allowing it to represent any arbitrary state. Sure, such a representation would be 'weaker' in the mathematical sense, but what we lose in generality, we gain in computational and conceptual simplification. In fact it is this trade off between generality and efficiency of a representation that partially motivates the work in this thesis.

As described above, the states that follow area-law are among the most relevant physical states. And this is one of the main reasons for the success of TN representation: a tensor network naturally represents the states that follow area-law.

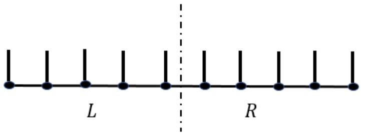
1.1 Entanglement and Tensor Network (TN)

Now that we have described the physical and conceptual need to find a better representation of quantum states, we will now show, from first principles, how a TN representation naturally emerges out of the basis-superposition representation if we try to extract entanglement information from a the state that follows area-law.

Consider a 1D spin chain of some length N. There are N sites, and on each site sits a quantum spin which takes value in some Hilbert space of dimension d. The total Hilbert space is a tensor product of local Hilbert spaces on each site. Lets say we are given some wave function on this system, ψ,

$$|\psi\rangle = \sum_{i_1,i_2,\ldots,i_N=1}^{d} C_{i_1,i_2,\ldots,i_N} |i_1, i_2, \ldots, i_N\rangle. \tag{1.1}$$

This is the usual representation of many body states in terms of computational basis. Lets say we are interested in the entanglement information of this state. Specifically, we want to know how much entanglement exists between the left half, L, and right half, R, of this wave function.



. (1.2)

It cannot be read off the above representation of ψ, so we need to represent it differently. In fact, as we know, we need to perform a Schmidt decomposition along the entanglement cut. In the present context Schmidt decomposition can be thought of as an alternative representation of the wave function in which a wave function in Hilbert space $H_L \otimes H_R$ is represented using relevant basis states from the the two constituting Hilbert spaces.

$$|\psi\rangle = \sum_{a=1}^{D} \lambda_a |\psi_a^L\rangle \otimes |\psi_a^R\rangle, \tag{1.3}$$

where ψ_a^L and ψ_a^R are states in H_L and H_R respectively, and λ_a are the Schmidt coefficients. The Schmidt index a takes D values. The number D is the Schmidt rank. D itself is a crude measure of entanglement between L and R, though a better measure would be the Von-Neumann entropy defined as $S = -\sum_a \frac{\lambda_a^2}{\sum_a \lambda_a^2} \log \frac{\lambda_a^2}{\sum_a \lambda_a^2}$. For an arbitrary state in the total Hilbert space, the Schmidt rank D can be very large (exponentially large in $N/2$). But as we discussed above, if the state follows area-law, D can be kept small by ignoring small Schmidt coefficients. So it means for area-law states it is possible to choose a finite number of basis states from the two constituent Hilbert spaces and represent the overall state as their tensor-product sum. And of course this representation is better than that in Eq. (1.1) because the entanglement between the two subsystems can be directly read off from the description.

If we want to gain more information about the entanglement structure, we can repeat this process with the states $\psi^{L/R}_j$ and further decompose them in smaller parts. Again using area-law we know that in each decomposition we only need to retain a finite number of Schmidt coefficients. We can repeat this process until the wave function is written completely in terms of Schmidt basis on each site. Since we needed $\sim D$ new Schmidt coefficients upon each decomposition, we will end up with $\sim D^{\log_2 N} = poly(N)$ coefficients that contain the entanglement information of the wave function under consideration. This is much more efficient representation of the state compared to basis-superposition in (1.1).

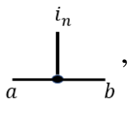
Lets say we Schmidt decompose ψ^L_a further for every a in two halves, left, L_1, and right,R_1. We might expect to get something like $|\psi^L_a\rangle = \sum_b \lambda_{a,b}|\psi^{L_1}_{a,b}\rangle \otimes |\psi^{R_1}_{b,a}\rangle$. But if the states in L_1 also carry a index, then roughly speaking, it would imply the region L_1 is is entangled with the region R as well, which should not happen under area-law. So we would instead have something like $|\psi^L_a\rangle = \sum_b \lambda_b|\psi^{L_1}_b\rangle \otimes |\psi^{R_1}_{b,a}\rangle$. The same argument can be applied at every stage of the decomposition, that the Schmidt basis appearing in a region should only carry Schmidt indices that are shared only with regions nearby. Finally when we have Schmidt decomposed down to the individual sites, we would get something like

$$|\psi\rangle = \sum_{a_{12},a_{23},\ldots,a_{N-1,N}} |\psi^{(1)}_{a_{12}}\rangle \otimes |\psi^{(2)}_{a_{12},a_{23}}\rangle \otimes \ldots |\psi^{(N)}_{a_{N-1,N}}\rangle, \tag{1.4}$$

where Schmidt coefficients have been absorbed into wave functions on individual sites. Here $a_{k,k+1}$ is a Schmidt index that runs between site k and $k+1$. Due to area-law, $a_{k,k+1}$ take maximum D values for some finite D ($O(1)$ in system size).

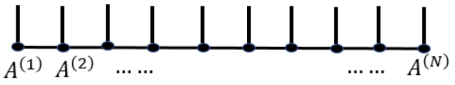
We can represent this wave function diagrammatically as follows. Represent $\psi^{(n)}_{a,b}$ (which is a state at nth site for a fixed a and b) appearing in the above as an object with 3 indices,

$$|\psi^{(n)}_{a,b}\rangle = \sum_{i_n} A^{(n)}_{i_n;a,b}$$



$$, \tag{1.5}$$

where $A^{(n)}_{i_n}$ is a matrix with elements $A^{(n)}_{i_n;a,b}$. (Note that $A^{(1)}$ and $A^{(N)}$ will have to be vectors rather than matrices.) Define a convention for summing over horizontal indices ((a,b) in above): if two such tri-leg diagrams are placed next to each other such that they share a horizontal index, then we project them to be equal to each other and sum over them, and we also assume that the diagram is always summed over all physical indices. With this convention one can see that the above state can simply be written as

$|\psi\rangle =$  , (1.6)

or more explicitly

$$|\psi\rangle = \sum_{i_1,i_2,\ldots,i_N} A^{(1)}_{i_1} A^{(2)}_{i_2} \ldots A^{(N)}_{i_N} |i_1, i_2, \ldots, i_N\rangle, \tag{1.7}$$

We have arrived at the simplest tensor network states, what is known as the open boundary Matrix Product State (MPS). The terminology comes from the fact that the coefficients in the computational basis come from products of matrices. The 3-index object we defined is called a rank-3 tensor. Mathematically, it is a diagrammatic representation of a multi-linear map of rank 3.

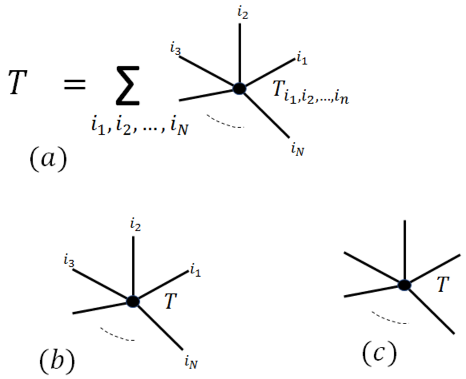
So we see that the desire to extract the entanglement information of a 1D state, facilitated by the assumption of area-law, led us to a TN representation of the state under consideration. This happens to be true generally in higher dimensions as well. In fact, most exotic many-body phenomena we mentioned can be discussed and represented efficiently through TN. It should be noted that, in principle, the TN states can represent *any* state in the Hilbert space, but the value D in the above would have to be arbitrarily high (exponential in system size). So the benefits of using tensors is mostly lost in such cases.

1.2 TN formalism

After having shown the physical motivation behind TN state construction, we introduce the general formalism of tensor networks very briefly. A general tensor T of rank N can be thought of as an element of tensor product of n Hilbert space, $T \in H_1 \otimes H_2 \otimes \ldots \otimes H_N$. H_j, in general, can have different dimensions d_j for each $j = 1, \ldots, N$. If we choose a basis in each Hilbert space, then we can write T in terms of a basis

$$T = \sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} \cdots \sum_{i_N=1}^{d_N} T_{i_1,i_2,\ldots,i_N} |i_1\rangle \otimes |i_2\rangle \ldots \otimes |i_N\rangle. \tag{1.8}$$

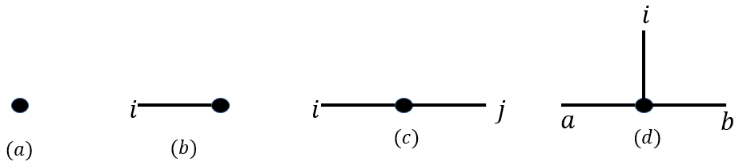
$i_1, \ldots, i_N$ are called the indices of the tensor T, and the corresponding dimensions $d_1, \ldots, d_N$ are called *bond-dimension* corresponding to the respective indices. Indices are also often called ‘legs’ or ‘edges’ when we represent T in a diagrammatic way. Depending on the context, one can represent the tensor T in various ways,



(1.9)

Representation in (a) is the mose explicit one. It shows a tensor in terms of its components. Though often, the sum over indices is implicit, in which case we represent it as in (b). In fact sometimes we do not even need to label the indices explicitly, so we simply represent it as a node with legs as in (c).

Generally, any object of interest in linear algebra can have a tensor representation



. (1.10)

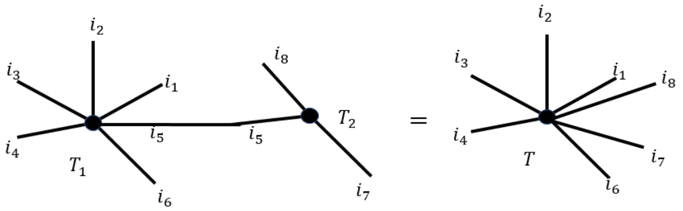
(a) is rank-0 tensor, which is nothing but a scalar. (b) is a rank-1 tensor which is nothing but a vector v, where i runs over the vector components v^i. Similarly, (c) is rank-2 tensor, which represents a matrix M with elements $M_{i,j}$. Last example is a rank-3 tensor, for example the one we encountered in MPS representation of 1D state above.

From tensors to tensor networks

There is a fundamental operation defined on single tensors, and a set of tensors called *index contraction* , or simply, *contraction*. Contraction is an operation on a set of tensors defined by summing over all possible values of the repeated indices in the set. Contraction produces another tensor typically of smaller total rank than the set it acted on. Contraction allows us to represent some of the basic operations of linear algebra. For example, multiplying a vector v with matrix M is nothing but index contraction of their common index,

$$u_b = \sum_a M_{b,a} v_a. \tag{1.11}$$

Similarly, matrix multiplication is an index contraction between two rank-2 tensors, $C_{a,c} = \sum_b A_{a,b}B_{b,c}$. Contraction is represented graphically by simply connecting the 'legs' of two tensors being contracted.

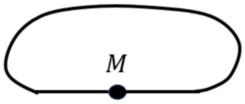


. (1.12)

It shows contraction of one index between a rank 5 tensor T_1 and a rank 3 tensor T_2 which produces a rank-6 tensor T. Algebraically, the resulting tensor will have components,

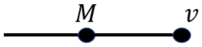
$$T_{i_1,i_2,i_3,i_4,i_6,i_7,i_8} = \sum_{i_5}(T_1)_{i_1,i_2,i_3,i_4,i_5,i_6}(T_2)_{i_5,i_7,i_8}. \tag{1.13}$$

Though in this example we deleted the contracted leg, to highlight the fact that we get another tensors, usually, the contracted legs are shown explicitly in the diagram for convenience of representation. One can see that a matrix trace operation is nothing but contraction of the two indices of a rank-2 tensor,



, (1.14)

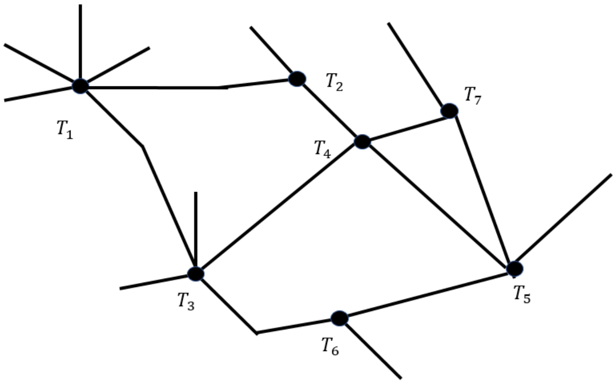
and matrix multiplication with a vector can be represented as follows,



. (1.15)

After understanding the general notions of tensors and contraction operation on them, now we can understand tensor networks

A tensor network (TN) is a set of tensors where some or all of its indices have been contracted according to some fixed graphical structure on the set. To give it a more concrete meaning, we can think of a general graph where each node represents a tensor, and the edges connecting to that node are the legs of the tensor. Then if a leg is shared by two nodes, we contract them. If it is open ended, then we do not contract them. With these contractions, the resulting tensor can be visualized as a network created out of the original set. Following is an example of a tensor network made out of 7 individual tensors,



$$. \qquad (1.16)$$

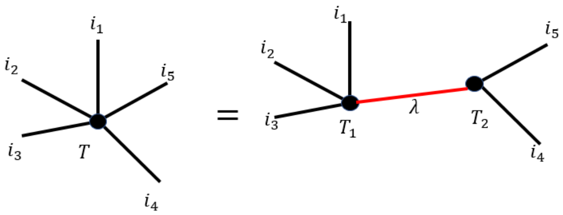
All edges of this graph that have both ends at nodes should be understood as having summed over. While the edges that are open ended should be understood as the indices of the resulting tensor (or the tensor network).

Breaking tensor netwrok into smaller pieces: tensor decomposition

Tensor decomposition is another fundamental operation on the tensor which is opposite to that of contraction. Every tensor of rank N can be decomposed into two tensors of rank $n + 1$ and $N - n + 1$. To do that, we can think of the given tensor as a linear map from n of its indices to the other $N - n$ indices. With this, it becomes a matrix, which can be SVD decomposed. For concreteness lets consider a rank-5 tensor T. Lets say we want to decompose into two tensors T_1 and T_2 of rank 4 and 3 respectively. We write

$$T = \sum_{i_1,\ldots,i_5} T_{i_1,i_2,i_3;i_4,i_5} |i_1, i_2, i_3\rangle\langle i_4, i_5|$$

. As a matrix it will have a SVD decomposition, $T = USV^\dagger$. Write $T_1 = U\sqrt{S}$ and $T_2 = \sqrt{S}V^\dagger$. It is easy to see that T_1 has 4 indices, $(T_1)_{i_1,i_2,i_3,\lambda}$ where lambda is the index coming from the singular value matrix S. Similarly T_2 will have 3 indices, $(T_2)_{\lambda,i_4,i_5}$. Diagrammatically it can be represented as



$$, \qquad (1.17)$$

where the SVD index has been shown in red to highlight. Decomposition is conceptually opposite to that of contraction. Contraction maps a set of tensor into a single tensor network, while decomposition can be used to break a tensor network into pieces. However decomposition should not be confused as an inverse map to that of contraction: if T_1 and T_2 contract to give tensor T, it is not necessarily true that decomposition of T will give back tensor T_1 and T_2.

Tensor decomposition, as we will see, plays a key role in application of tensor networks in quantum many-body physics. It allows for us to cut down and approximate inherently non-local objects, such as a quantum many-body state, into smaller, local pieces, which are easier to handle and manipulate. The reason for this is that tensor decomposition is related to a fundamental quantum mechanical quantity, the entanglement, as we will discuss now.

Tensor decomposition and entanglement

Any quantum mechanical wave function can be thought of as a vector in the tensor product of Hilbert spaces of its degrees of freedom. Therefore, it is a tensor of rank N where N is number of degrees of freedom in a particular description. To find out how entangled two degrees of freedoms are, we have to Schmidt decompose the wave function in terms of wave functions on individual degrees of freedoms. This is nothing but the tensor decomposition described above. Roughly speaking, the number values the emergent SVD leg in the decomposition has to take measures the amount of entanglement between the constituent degrees of freedom. For example, if it takes only one value, then it implies that the degrees of freedom under consideration are not entangled at all. If it takes he maximum possible values, it implies they are maximally entangled.

In the context of many-body quantum states, the degree of freedom of interest are often just the local Hilbert spaces. If we go back and analyze the way we derived the MPS state above, we can interpret it in the following way. We started with a quantum many-body state, which can be thought of as a tensor of rank N, where N is the system size. When we performed an Schmidt decomposition in the middle, it was basically decomposing the rank N tensors into two two tensors of rank $N/2+1$ each. Schmidt decomposition is nothing but the SVD decomposition described above. The Schmidt coefficients are the singular values and the basis states $\psi_a^{R/L}$ were the tensor $N/2+1$ rank tensors. This tensor decomposition was useful because of the area-law, which, roughly speaking, puts an upper limit on the dimension of the extra index that appears during decomposition. We repeated this process and broke the tensor into smaller and smaller pieces, until we reached tensors corresponding to single-sites. Again, area-law facilitated us along this decomposition by keeping the bond-dimension finite at each stage.

We would work out many such tensor networks in detail in the course of this thesis, and hence we would make extensive use of tensor decomposition in various ways.

1.3 TN representation of 1D states

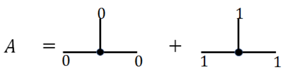
Till now we have discussed tensor networks abstractly. Now we will discuss some concrete examples to appreciate their application to quantum many-body physics.

As mentioned earlier, the TN states in 1D are often called Matrix Product States (MPS). The MPS described earlier had, in general, a different rank-3 tensor sitting at each site. But when we are not interested so much in microscopic detail of a system but rather in the class (often called a *phase*) of a universal behavior that the state belongs to, then it is usually possibly to find a representative state of that class that is translation-invariant. In this case we can put the same tensor everywhere to construct the state. Translation-invariant MPS can simply be written as,

$$|\psi\rangle \sum_{i_1,\dots,i_N} A_{i_1}\dots A_{i_N}|i_1,\dots,i_N\rangle. \tag{1.18}$$

It means that the state, and the phase it belongs to, can roughly be represented by a single rank-3 tensor, $A_{i;a,b}$. This is indeed a remarkable reduction in complexity of the problem! In the context of many-body states, the indices that are contracted to produce the tensor network are often called 'inner' or 'virtual' indices/legs. And the uncontracted legs are called 'outer' or 'physical' indices/legs. So in MPS description $A_{i;a,b}$, i is the physical index, while a, b are virtual/inner indices. Now we will look some examples of some important 1D states represented by such a rank-3 tensor. As mentioned before, Tensor network states in 1D are called Matrix Product States (MPS). MPS has found a wide range of application in 1D many-body physics. One can prove that the low-energy states of all gapped, local 1D Hamiltonian can be described efficiently by an MPS. Here are a few examples,

1- GHZ states: The simplest example of an MPS is a GHZ state. A GHZ state can simply be written as, $\psi = |00\dots0\rangle + |111\rangle$. Its MPS representation has the same rank-3 tensor A sitting at each site, where A is given by



(1.19)

2- AKLT states: The AKLT[23] Hamiltonian is, $H = \sum_i \vec{S}_i\vec{S}_{i+1} + \frac{1}{3}(\vec{S}_i\vec{S}_{i+1})^2$ where $\vec{S}_i$ is the vector of spin-1 operators at site i. It is a gapped Hamiltonian with ground state admitting a MPS representation. The local rank-3 tensor can be written as $A_0 = \sigma_z, A_1 = \sqrt{2}\sigma^+, A_2 = \sqrt{2}\sigma^-$, where σ's are Pauli-operators.

3- Cluster states: Cluster states[24] are defined as the 1D state that is a +1 eigenstate of a set of mutually commuting operators given by $O_i = \sigma_z^{i-1}\sigma_x^i\sigma_z^{i+1}$. These states can be constrcuted by applying the product of associated projector $\frac{1}{2}(1 + O_i)$ for each site to the vacuum state. It admits an MPS representation given by $A_0 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ and $A_1 = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$.

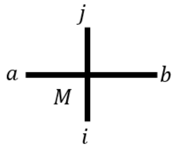
4- Symmetry Protected Topological (SPT) phases: the cluster states described above are actually a particular examples of what is known as the SPT phases of quantum matter[14–18]. These phases are characterized by their central property that these systems can actually be deformed into each other smoothly (that is, without a phase transition) if we do not protect certain global symmetries of the system. However, different phases cannot be connected if try to maintain the same global symmetry along the deformation path. It has been shown that all SPT phases in 1D can be represented efficiently by MPS [25].

Before moving on to describing TN states in two dimensions, we would first look another important application of tensor networks: representation of operators.

1.4 TN representation of 1D operators

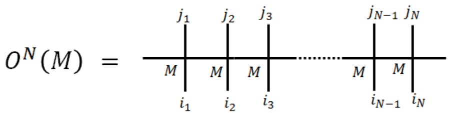
Till now we have only talked about how tensor networks can often represent physically relevant many-body quantum states. But as we remarked earlier, tensor networks can, in general, be used for any object in linear algebra. Of course, just as we need to find a good representation of states in many-body physics, we may also need to find good representation of the many-body operators that act on these states. In fact, if we are working with the TN representation of states, it is desirable to work with TN representation of operators as well.

How can we represent operators using tensors? Since we are mainly focusing on translation-invariant systems, we are mainly concerned with translation-invariant operators as well. To construct an operator, consider a rank-4 tensor as follows:



. (1.20)

We think of the indices i and j as the input and output physical index, and a and b as the left and right virtual indices. Lets say i, j take values in some local Hilbert space, H_d. Now we juxtapose these tensors side by side and contract the shared virtual indices. We get a tensor network that looks something like this:



. (1.21)

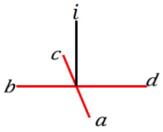
This can easily be seen to be a linear operator from Hilbert space $H_d^{\otimes N}$ to $H_d^{\otimes N}$. Such operators are called Matrix Product Operators (MPO), because they are described as a product of matrices $M^{i,j}$. We will come across various applications of MPOs in the course of this thesis. As one can see, MPO representation can produce highly entangled, non-local operator very easily. As we will see, MPOs provide a very important formalism for the study of TN representation of topological states in 2D.

The matrix product formalism [26, 27] has played a significant role in the study of one dimensional systems. In particular, the matrix product representation of 1D quantum states underlies successful numerical algorithms like the Density Matrix Renormalization Group algorithm [28] and the Time-Evolving Block Decimation algorithm [29]. Moreover, the matrix product representation provides a deep insight into the structure of the ground states in 1D [27], which enables rigorous proofs of the efficiency of 1D variational algorithms in search for the ground states [30, 31] and also a complete classification of 1D gapped phases [25, 32–34].

Operators can also be represented in a matrix product form [35–37], which provides a useful tool in the simulation of one dimensional mixed states and real / imaginary time evolutions (see for example Ref. Mascarenhas2015,Wall2012). In particular, matrix product operators which are unitary play an important role in not only the simulation of dynamical processes in 1D, but also the understanding and classification of (symmetry protected) topological phases in 2D [14–18]. One of the main objectives of this thesis is to understand the structure of Unitary matrix product operators.

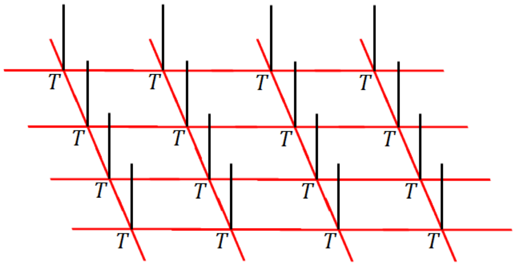
1.5 TN representation of 2D quantum states

Tensor network representation of 2D quantum states is often called Projected Entangled Pair States (PEPS). PEPS are immediate generalization of MPS to two dimensions. As before, we are mainly interested in translation-invariant states. Consider the following rank-5 tensor with components $T^i_{a,b,c,d}$,



, (1.22)

where i is the physical leg and a, b, c, d are the virtual legs. Physical leg should be visualized as vertical while the virtual legs should be visualized to be in 2D plane. To better distinguish physical and virtual legs in 2D we will follow this color convention through the thesis: physical legs would be represented by vertical black lines, while virtual legs would be represented by horizontal red lines. Now we can put these tensors next to each other on a 2D square lattice and contract the virtual legs. This tensor network will represent a 2D quantum many-body states with some dangling virtual legs on the boundary,



. (1.23)

In different context, the boundary can be closed or used differently. Note that though we chose a square lattice to represent a 2D TN, similar TN can be described on any tringulation of a 2D manifold. Of course one can see that one has to have a minimum rank-4 tensor to represent a 2D state. PEPS also satisfy area-law like MPS.

PEPS are able to represent a wide range of gapped 2D quantum states. They find applicability in Tensor Renormalizaion Groups (TRG), Second Renormalization Group (SRG), Higher-Order Tensor Renormalization Group (HOTRG) Corner Transfer Matrices (CTM) etc. They are widely used in simulation of phase transitions of various kinds.

One particular application of PEPS that we are interested in, is in representation of 2D topological phases, and simulation of topological phase transitions. The first half of this thesis would be centered around investigating these representations.

1.6 Summary of chapters and guide to read the thesis

The thesis is naturally divided in two parts: chapter 2 and 3 are focused on TN representation of 2D topological states, while chapter 4 and 5 are focused on understanding TN representation of 1D unitary operators. These two parts are mostly independent of each other. So a reader can choose to first focus on either of the part as per their interest. Both chapter 2 and 3 have the common theme: they investigate how 'stable' a TN representation of a 2D topological state is. But it is advisable to read chapter 2 before chapter 3.

Chapter 2: Chapter 2 starts with first giving an algorithm to calculate the topological entanglement entropy in section 2.2, which readers can skip if they are not interested in the details of the algorithm, as the rest of chapter does not use it as a prerequisite. Then we start with analyzing concrete, and relatively simple topological model of toric code and its TN representations. We observe that the TN representations are unstable. The next few sections delve into defining some intuitive subspaces of the tensor to explain this instability. These are the key definitions. Then the central mathematical conjecture explaining the instability of this chapter is presented in section 2.6 along with the conjectured physical explanation. Next few sections provide the physical reasons of instability in sequential logical steps. We conclude this chapter with noting that TN instability has important consequences for topological phase transition simulations.

Chapter 3: The goal of this chapter to prove the conjecture put forward in chapter 2 for the general string-net models and their general TN representation. It does so in a series of logical steps, first by calculating the required subspaces of the conjecture in section 3.4 and section 3.5. After generalizing discussion from chapter 2 to the general string-nets in next few sections, we discuss a non-abelian example, Double-Fibonacci model in section 3.9. Finally we give an analytical proof of the conjecture for all string-nets.

Chapter 4: Chapter 4 and 5 also have a common theme: characterizing MPOs that are unitary. But chapter 4 is almost entirely focused on a particular subset of these operators: operators that also map local operators to local operators. We start with assuming a general form for the MPU, based on their unitarity, which is subsequently used to prove that these tensor satisfy certain simple properties. It is shown easily in section 4.2 that all locality-preserving unitaries can be represented by MPOs. As one of the key results of this chapter, it is then shown in section 4.4 that the GNVW index, that measures the flow of information under such unitaries, can be extracted by defining a Rank-Ratio index, which can be locally calculated.

Chapter 5: In chapter 5 we go beyond the locality-preserving unitaries and find a necessary sufficient condition for all periodic 1D unitary MPOs in section 5.2 as the key result of this chapter. To discuss these MPOs a general notion of action on local operator is discussed and quantified in section 5.4. With these new tools, we first revisit the locality-preserving unitary MPOs and find a sufficient condition for locality-preservation in section 5.5 which is parallel to condition for short-ranged correlation of MPS in a certain sense. We try to improve upon some of the results from chapter 4 and show a direct relation between GNVW index and Rank-Ratio index. We conclude by discussing some interesting examples of MPUs which map local operators to non-local operators in section 5.6. These MPUs are shown to have MPO symmetries similar to PEPS representing topological order in 2D.

Chapter 2

TN REPRESENTATION OF 2D TOPOLOGICAL QUANTUM STATES AND ITS INSTABILITIES

2.1 Background and motivation

The tensor network representation of quantum states (including the matrix product states in 1D) [38–41] provides a generic tool for the numerical study of strongly interacting systems. As variational wave functions, the tensor network states can be used to find the ground state wave function of local Hamiltonians and identify the phase at zero temperature. In particular, it has become a powerful approach in the study of topological phases, whose long range entanglement is hard to capture with conventional methods. It has been shown that a large class of topological states, the string-net condensed states [13], can be represented exactly with simple tensors [42, 43]. Moreover, numerical studies applied to realistic models have identified nontrivial topological features in the ground state wave function (see e.g. Ref.[44–46]).

In the numerical program, the parameters in the tensors are varied so as to find the representation of the lowest energy state. After that, topological properties are extracted from these tensors in order to determine the topological phase diagram at zero temperature. However, this problem might not be numerically 'well-posed'. That is, arbitrarily small variations in the local tensor may lead to a completely different result as to what topological order it represents. In particular, Ref. [47] demonstrates that this happens in the case of Z_2 toric code topological order. While this presents a serious problem for the tensor network approach to study topological phases, Ref. [47] also showed that such instabilities can be avoided if certain Z_2 symmetry is preserved in the local tensor. It has been shown that the topological order in the toric code model is stable against arbitrary local perturbation to the Hamiltonian of the system [48]. The fact that a certain variation direction of the tensor network representation may induce an immediate change in the topological order indicates that such a variation corresponds to highly nonlocal changes in the ground state wave function.

Does a similar problem occur for other tensor representaion of the toric code, or for other topological states as well? This is the question we address in this chapter and the next chapter. In particular, we ask:

1. Does the tensor network representation of other topological states also have such unstable directions of variation?
2. If so, can we predict which variaions cause instabilities and which do not? In particular, can they be avoided by restricting the variations to a certain subpace?
3. What is the physical reason behind such instabilities and their prevention?

While the Z_2 symmetry requirement for toric code is naturally related to the Z_2 gauge symmetry of the theory, for more general string-nets which are not related to gauge theory, it is not clear whether similar symmetry requirement is necessary and if so what they are.

In this chapter and the next chapter, we answer the above questions as follows:

1. All string-net tensors have unstable directions of variation.
2. Instabilities are caused if a variation respects the ‘stand-alone’ symmetry but violates the Matrix-Product-Operator(MPO) symmetries introduced in Ref.[15, 49]. We give a mathematical construction of the projector onto the space of all stable variations that can be used to in numerical calculations to protect against instabilities.
3. The physical reason for the instability is that ‘stand-alone’ variations which violate these symmetries induce condensation of bosonic quasi-particles and hence destroys (totally or partially) the topological order.

To support the above claims, we calculate the topological entanglement entropy S_{topo}[50, 51] from the representing tensor and (partially) characterize the encoded topological order. In particular, consider a tensor network state represented by a local tensor T. We are interested in varying the local tensor T everywhere on the lattice, in such a way that $T \longrightarrow T + \epsilon T'$, where $\epsilon \ll 1$. In order to study whether topological order is lost or still present after a variation in the direction T', we calculate topological entanglement entropy of the original and the modified state as a function of ϵ, $S_{\text{topo}}(\epsilon)$. We say the variation is unstable in T' direction if

$$\lim_{\epsilon \longrightarrow 0} S_{\text{topo}}(\epsilon) \neq S_{\text{topo}}(0). \tag{2.1}$$

If $\lim_{\epsilon\to 0} S_{\text{topo}}(\epsilon)$ is smaller than $S_{\text{topo}}(0)$, we say that topological order is (partially) lost. If $\lim_{\epsilon\to 0} S_{\text{topo}}(\epsilon) = S_{\text{topo}}(0)$ we call that direction stable meaning that topological order is still present and remains the same. This understanding of tensor instability is important not only for the identification of topological order for a particular model, but also for the numerical study of phase transitions between topological phases. In particular, if one is to use the tensor network approach to study phase transition due to boson condensation, then the corresponding variation direction must be *allowed* in order for the simulation to give correct results. For example, in Ref. [52], it was shown that if such variation directions are not included as variational parameters, then we see a first order transition even though in fact it is second order. We are going to elaborate more on this point later at the end of this chapter.

The problem of general string-net model would be discussed at length in the next chapter. In this chapter we would focus on particular examples and a general formalism for understanding instabilities. This chapter is organized as follows. We start with showing in section 2.2 how we calculate the topological entanglement entropy with a given tensor network representation of the topological state. Then in section 2.3 we discuss the simplest topological model, the toric code [12], and study two types of tensor network representation, single-line and double-line representation, and their instabilities. We find that while single-line indeed has instability with respect to certain Z_2 symmetry breaking variations (as shown in Ref [47]), the double-line instabilities are more complicated. The double-line examples show that different virtual symmetries of a tensor interact in a complicated way in determining which variations are unstable. To understand the relation of instabilities with virtual symmetries we first define two subspaces of the virtual space: 'stand-alone' subspace in section 2.4 and then MPO subspace in section 2.5. Then we present the main conjecture of this work, the *tensor-instability conjecture* in section 2.6, where we put forward the conjecture that variations that are in stand-alone subspace but are outside the MPO subspace are the ones that cause instability. We then present the physical understanding of this conjecture in the form of a physical conjecture that says that these instabilities are caused by boson condensation. To understand this physical conjecture, we first explain the physical significance of the stand-alone space in section 2.7 followed by an understanding of the physical significance of the MPO subspace explained in section 2.8. Then in section 2.11, combining the understanding of these two subspaces, we finally explain how unstable variations physically correspond to condensation of bosons and offer this as the reason for topological phase transition seen in numerical calculations. We test the mathematical and physical conjecture on the double-semion model in section 2.12 and conclude the chapter with a discussion on what this results implies about the phase transition simulations using tensor network ansätz and finally giving an outlook in section 2.14.

2.2 Calculating topological entanglement entropy with tensor network

Here we explain the algorithm we use to calculate the topological entanglement entropy of any translation invariant tensor network state. We use the idea presented by Cirac *et al.* [53] to calculate reduced density matrix on a region and hence its entanglement entropy. We consider honeycomb lattice, though it can easily be extended to other lattices. By translation invariant we mean that all vertices on the sublattice A and sublattice B are attached with the same tensors, T_A and T_B, respectively. First we define certain notations for convenience of later discussion. The starting objects are given tensors T^I_α, where I and α denote the set of physical and virtual indices, respectively: $I = (i_1, i_2, ..)$, $\alpha = (\alpha_1, \alpha_2, ..)$. The state represented by these tensors can be written as

$$|\Psi\rangle = \sum_{I_1, I_2, ..} \mathrm{Tr}(T^{I_1} T^{I_2} \ldots) \, |I_1, I_2, \ldots\rangle \,. \tag{2.2}$$

We denote the tensor resulting from contracting the virtual indices of tensors T on a region R as $T(R)$. $\mathbb{T}$ denotes the ‘double tensor’ resulting from contracting the physical indices of T with those of $T^\dagger$, that is, $\mathbb{T} = TT^\dagger = \sum_I T^I_\alpha \left(T^I_{\alpha'}\right)^*$. Similar to $T(R)$, we denote the double tensor contracted on a region R as $\mathbb{T}(R)$.

Now let us consider putting this tensor network state on a cylinder. We denote the left half of the cylinder as L and the right half as R. The honeycomb lattice is placed in a way so that L and R divide it into exact halves. So the line between the two halves goes through the middle of the plaqeuttes as shown in the Fig. 2.1(a). We denote the tensors on the left and right boundaries as T_l and T_r.

When we contract bulk double tensors with the boundary double tensors, we get a density matrix operator on the virtual indices,

$$\sigma_L = \mathbb{T}_l(\partial L)\mathbb{T}(L), \quad \sigma_R = \mathbb{T}(R)\mathbb{T}_r(\partial R). \tag{2.3}$$

Cirac *et al.* [53] showed that the physical reduced density matrix on one of these halves, let’s say the left one, is related to the density operator on the virtual indices as,

$$\rho_L = U\sqrt{\sigma_L^T}\sigma_R\sqrt{\sigma_L^T}U^\dagger, \tag{2.4}$$

where U is an isometry. Hence ρ_L and$\sqrt{\sigma_L^T}\sigma_R\sqrt{\sigma_L^T}$ have the same spectrum. In addition, under right symmetry conditions, $\sigma_L^T = \sigma_R = \sigma_b$. When this is true, up to change of basis, we find that $\rho_L \propto \sigma_b^2$. The normalized reduced density matrix is

$$\rho_L = \frac{\sigma_b^2}{\mathrm{Tr}(\sigma_b^2)}. \tag{2.5}$$

It is known that the Rényi entropy with any Rényi index gives the same topological entanglement entropy[54]. So we calculate Rényi entropy with Rényi index $\frac{1}{2}$,

$$\begin{aligned} S_{1/2}(\rho_L) &= \frac{1}{1-1/2}\log \mathrm{Tr}(\rho_L^{1/2}) \\ &= 2\log \mathrm{Tr}(\sigma_b) - \log \mathrm{Tr}(\sigma_b^2). \end{aligned} \tag{2.6}$$

In the limit of large cylinder, it should behave like

$$S_{1/2}(\rho_L) = \alpha_0|C| - S_{\mathrm{topo}}, \tag{2.7}$$

where $|C|$ is the circumference of the cylinder. This is how we calculate S_{topo} starting with a tensor network state.

Before we move on to the next step, we would like to mention an important subtlety regarding computation of S_{topo} on a cylinder. In Ref.[55, 56] it has been shown that S_{topo} calculated this way on a cylinder, in general, might depend on the boundary conditions. We choose a particular boundary condition for all our calculations and examine the dependence of S_{topo} on boundary condition in the appendix A.1. Our findings are consistent with the conclusion in Ref.[56].

We first have to calculate $\mathbb{T}(R)\mathbb{T}_r(\partial R)$ for the above setup. The problem is, the computational complexity of exact tensor contraction grows exponentially with the size of R, so we need to use some approximate renormalization algorithm. We use an algorithm which is a slight modification of known tensor renormalization algorithms [41, 52, 57]. Consider double tensors contracted along a thin strip on the cylinder giving us a *transfer matrix operator*, $\mathbb{S}$. If R includes n of such strips, we have $\mathbb{T}(R) = \mathbb{S}^n$. Since the tensor network state under consideration are short range correlated along the cylinder, the spectrum of $\mathbb{S}$ is gapped. Consequently, for large n, only the highest eigenvalue and the corresponding eigenvector of $\mathbb{S}$ dominates. That is, in thermodynamic limit, $\mathbb{T}(R)$ only depends on the highest eigenvalue/eigenvector of the transfer matrix operator, $\mathbb{S}$. Moreover, we expect to approximate the eigenvector of highest eigenvalue with a *Matrix Product State (MPS)*

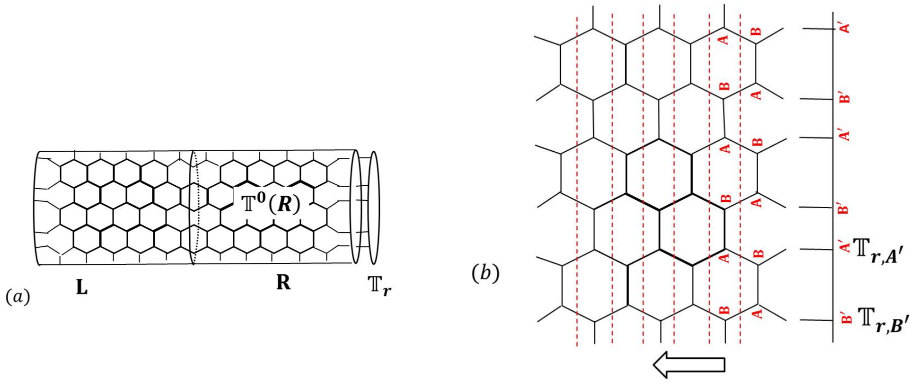


Figure 2.1: (a) The honeycomb lattice is put on a cylinder with some boundary tensors, T_r. We calculate the topological entanglement entropy by calculating the entanglement entropy of the right half of the cylinder. (b) We contract the bulk tensors with the boundary ones layer by layer (from right to left) in a recursive way. In each recursion step, a layer between the two red dotted lines is contracted with the boundary tensors.

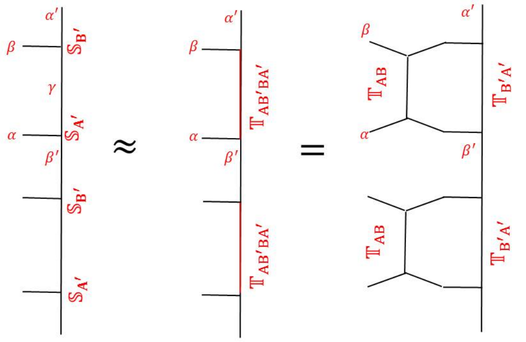


Figure 2.2: Each recursion step is shown in detail. We first contract the two bulk tensors $\mathbb{T}_A$, $\mathbb{T}_B$ and two boundary tensors $\mathbb{T}_{A'}$, $\mathbb{T}_{B'}$ to produce $\mathbb{T}_{AB'BA'}$. Then we use singular value decomposition to approximate $\mathbb{T}_{AB'BA'}$ as a contraction of two tensors $\mathbb{S}_{A'}$ and $\mathbb{S}_{B'}$.

with finite bond dimensions, since the tensor network state is short range correlated along the circumference of the cylinder. So we can start with a boundary MPS, apply the transfer matrix operator, and approximate the resulting state as an MPS with a fixed, finite bond dimensions. With each step, approximation to the eigenvector with highest eigenvalue improves and we do this recursively until we reach the fixed point giving us the desired eigenvector. Note that we require transfer matrix operators to be reflection symmetric for the condition $\sigma_L^T = \sigma_R = \sigma_b \Rightarrow \rho_L \propto \sigma_b^2$ to hold true.

The recursive algorithm is as following:

1. Initiate the boundary double tensor $\mathbb{T}_{A'} = \mathbb{T}_{r,A'}$ and $\mathbb{T}_{B'} = \mathbb{T}_{r,B'}$ (Fig. 2.1(b)).

2. Contract the bulk double tensors, $\mathbb{T}_A$ and $\mathbb{T}_B$ with each other giving a 4 leg tensor $\mathbb{T}_{AB}$. Contract $\mathbb{T}_{B'}$ and $\mathbb{T}_{A'}$ with each other giving a 4 leg tensor $\mathbb{T}_{B'A'}$. Contract $\mathbb{T}_{AB}$ and $\mathbb{T}_{B'A'}$ with one another giving 4 leg tensor $\mathbb{T}_{AB'BA'}$ (first equality in Fig. 2.2).

3. Reshape the tensor $\mathbb{T}_{AB'BA'}$ into a matrix M where $M_{\alpha\beta',\beta\alpha'} = (\mathbb{T}_{AB'BA'})_{\alpha\beta'\beta\alpha'}$ [57]. Now we perform an SVD decomposition of M, $M = U\Lambda V^\dagger$ and the approximation step: we keep only the highest D_{cut} singular values, and define the new tensors $\mathbb{S}_{A'}$ and $\mathbb{S}_{B'}$ as $(S_{A'})_{\alpha\beta'\gamma} = U_{\alpha\beta',\gamma}\sqrt{\Lambda_{\gamma,\gamma}}$ and $(S_{B'})_{\gamma\beta\alpha'} = \sqrt{\Lambda_{\gamma,\gamma}}V^\dagger_{\gamma,\beta\alpha'}$ where γ takes values $1, 2, \ldots, D_{\text{cut}}$. $\mathbb{S}_{A'}$ and $\mathbb{S}_{B'}$ form an approximate decomposition of $\mathbb{T}_{AB'BA'}$,

$$\sum_{\gamma=1}^{D_{\text{cut}}}(S_{A'})_{\alpha\beta'\gamma}(S^\dagger_{B'})_{\gamma\beta\alpha'} \approx (\mathbb{T}_{AB'BA'})_{\alpha\beta'\beta\alpha'} \tag{2.8}$$

   (approximating step in Fig. 2.2).

4. Check convergence of Λ. $\eta \ll 1$ is the precison tolerance. Let n denote the nth recursion step. If $||\Lambda_n - \Lambda_{n-1}||_1 < \eta$ exit algorithm.

5. Put $\mathbb{T}_{A'} = \mathbb{S}_{A'}$ and $\mathbb{T}_{B'} = \mathbb{S}_{B'}$ and go to step 2.

2.3 TN representations of Toric Code state and their instabilities

We start from the simplest illustrative example of nonchiral intrinsic topological order: the toric code [12]. We work on a hexagonal lattice and assign local degrees of freedom, i.e. 0-spin down- or 1-spin up, on the edges of the lattice. The Hamiltonian is a sum of local commuting projectors, given as

$$\begin{aligned} H &= -\sum_v A_v - \sum_p B_p \\ &= -\sum_v \prod_{l\in v} Z_l - \sum_p \prod_{l\in p} X_l, \end{aligned} \tag{2.9}$$

where v denotes the vertices, and p denotes the plaquettes. $l \in v$ denotes the edges attached to v and $l \in p$ denotes the edges on the boundary of plaquette p. Vertex terms restrict the ground states to closed strings of 1s and plaquette terms make all possible loop configurations of equal weight. Therefore, the toric code ground state (up to normalization) can be written as

$$\left|\Psi_{\text{gs}}\right\rangle = \sum_{X\in\text{closed}} |X\rangle\,, \tag{2.10}$$

where X denotes the string configurations on the lattice. So, the ground state of toric code hamiltonian is an equal weight superposition of all closed string configurations. It has topological order and has topological entanglement entropy $S_{\text{topo}} = \log 2$.

Now we look at tensor network representations (TNR) of the above toric code ground state. Specifically, we first explain the *Single-line* tensor representation, and then the *Double-line* tensor representation. We see that different TNR have different kinds of instabilities, which come from different self-bosons that can condense in each TNR. Specifically, the unstable direction in the single-line TNR condenses e-particles, while in the double-line and triple-line it condenses m-particles.

Single-line TNR of the toric code

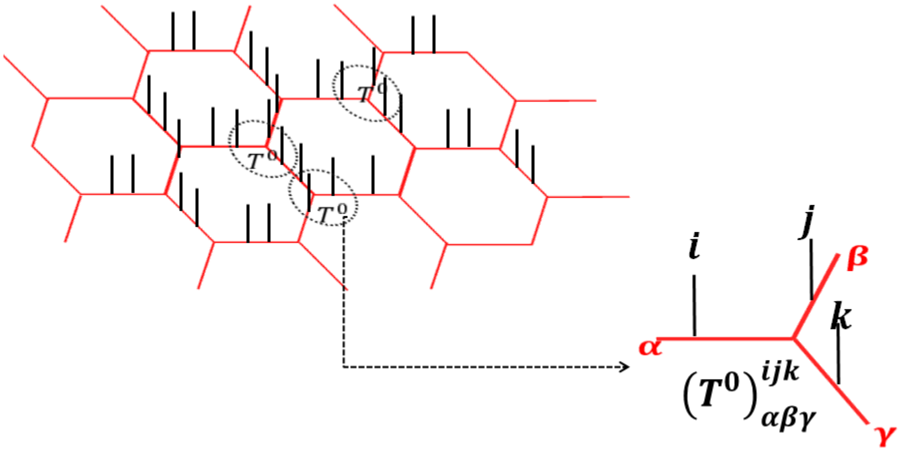


Figure 2.3: Single-line TNR of the toric code state.

This is the simplest TNR of the toric code state. We first split each qubit on the edges into two, as shown in the Fig. 2.3(a). That is, the labels 0 and 1 on every edge become 00 and 11 on the same edge. Now the local Hilbert space neighbouring each vertex is made out of three qubits. We associate a tensor with three physical indices/legs (throughout the paper we will use “indices” and “legs” interchangeably), and three virtual indices/legs to each vertex, represented algebraically as $(T^0)^{ijk}_{\alpha\beta\gamma}$ where i, j, and k are the three physical indices and α, β, γ are the three virtual indices, as shown in the Fig. 2.3(b). The components of the tensor are

$$(T^0)^{ijk}_{\alpha\beta\gamma} = \begin{cases} \delta_{i\alpha}\delta_{j\beta}\delta_{k\gamma} & \text{if } \alpha+\beta+\gamma = \text{even} \\ 0 & \text{otherwise} \end{cases}, \tag{2.11}$$

where δ is the kronecker delta function. So, physical and virtual legs are identified and an even number of indices carry label 1 out of every three edges neighbouring a vertex, i.e., we satisfy the vertex condition. The plaquette condition is also satisfied since every configuration is of equal weight. Therefore, the tensor network state constructed using the above local tensor leads to the toric code ground state given in Eq. (2.10).

Double-line TNR of the toric code state

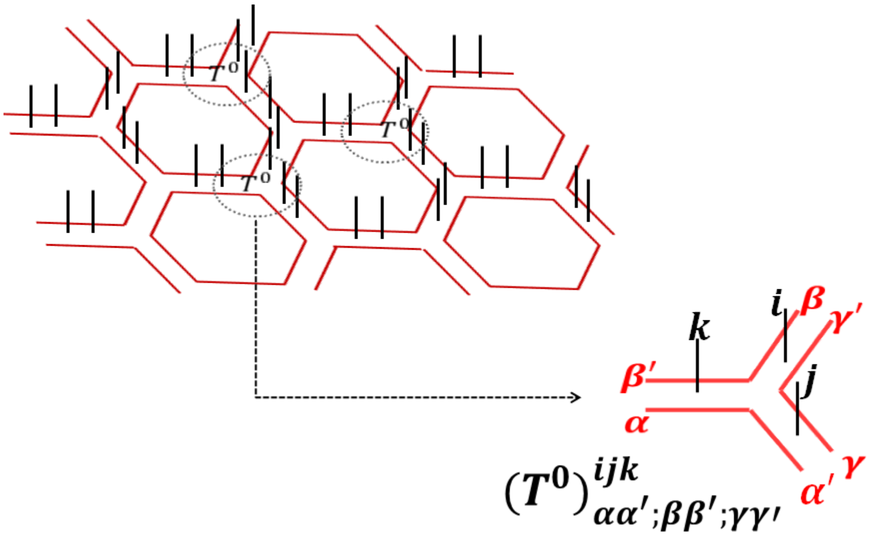


Figure 2.4: Double-line TNR of the toric code state. We again split the qubit on each edge into two, and assign each to the two nearby vertices. The local physical Hilbert space consists of 3 qubits. We associate to each vertex a tensor $T^{i,j,k}_{\alpha,\alpha';\beta,\beta';\gamma,\gamma'}$ (shown in (b)), where out of plane legs, i, j, k, correspond to the 3 physical indices, and in-plane legs $\alpha, \alpha', \beta, \beta', \gamma, \gamma'$ are the virtual indices. Virtual indices of the tensors contract along the shared edges to produce the toric code state on the physical indices.

In the double-line TNR of the toric code state, we associate with each vertex a tensor with 3 physical legs and 6 virtual legs, $T^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'}$, (see Fig. 2.4). We will refer to these virtual indices as 'plaquette indices' or 'plaquette legs' sometimes, because they carry the plaquette degree of freedom that comes from the local Hamiltonian term. All indices take values 0 and 1. We denote the TNR corresponding to the RG fixed point state as T^0. (We use the same notation for different fixed point tensors, but it should be clear from the context which fixed point tensor we are discussing.) First property of T^0 is that $(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'} \propto \delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{\gamma\gamma'}$, that is, indices on the same plaquette assume the same values. The second property is that the physical indices can be considered as labeling the domain wall between the virtual indices. If the two virtual indices in the same direction have the same values (both either 00 or 11) then the physical index in the middle has value 0, otherwise it is 1. That is, $i = \beta + \gamma, j = \gamma + \alpha, k = \alpha + \beta$ (all additions are modulo 2). So we can write T^0 as

$$(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'} = S^{ijk}_{\alpha\beta\gamma}\delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{\gamma\gamma'},$$

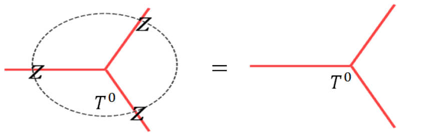
$$S^{ijk}_{\alpha\beta\gamma} = \begin{cases} 1 & \text{if } i = \beta + \gamma, j = \gamma + \alpha, k = \alpha + \beta \\ 0 & \text{otherwise} \end{cases}.$$

We can write all non-zero components explicitly,

$$\begin{array}{ll} T^{000}_{00;00;00} = T^{000}_{11;11;11} = 1, & T^{011}_{00;11;11} = T^{011}_{11;00;00} = 1, \\ T^{101}_{11;00;11} = T^{101}_{00;11;00} = 1, & T^{110}_{11;11;00} = T^{110}_{00;00;11} = 1. \end{array} \tag{2.12}$$

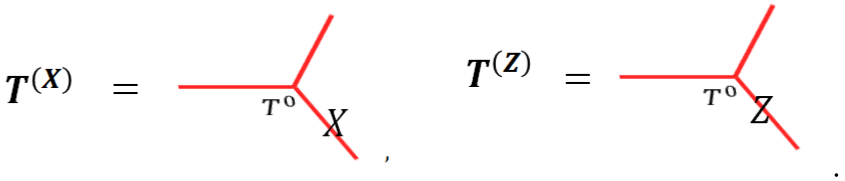
Instability in TNRs of toric code states

It was shown by Chen *et al.* [47] that single-line TNR of the toric code state is not stable in certain directions of variation. Before we explain what these unstable directions of variation are, we first note that the single-line TNR explained above has a *virtual symmetry*. If an operation on the virtual indices leaves the tensor invariant, we will call it a virtual symmetry of the tensor. Because the single-line tensor is non-zero only when virtual legs have even number of 1s, it has a natural $Z \otimes Z \otimes Z$ virtual symmetry (Fig. 2.13)(See Schuch *et al.* [58] for TNR virtual symmetries of the quantum double models). That is, the tensor in (2.11) satisfies the relation,



. (2.13)

It is a Z_2 symmetry with group elements $1 \otimes 1 \otimes 1$ and $Z \otimes Z \otimes Z$ acting on the virtual legs of the local tensor. Chen *et al.* [47] showed that topological order is stable with *any* Z_2 respecting variations and unstable with any Z_2 violating variation. To illustrate this, we can consider two different directions of variation in single-line TNR. We can add an X or Z variation on one of the virtual indices of the tensor,



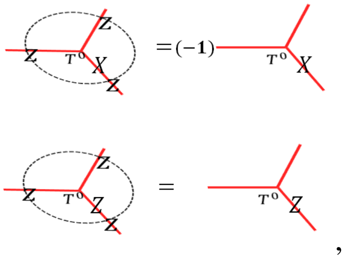
(2.14)

More explicitly, these tensor components are given by,

$$T^{(X)}_{\alpha\beta\gamma} = \sum_{\gamma'} X_{\gamma,\gamma'} T^0_{\alpha\beta\gamma'}, \tag{2.15}$$

$$T^{(Z)}_{\alpha\beta\gamma} = \sum_{\gamma'} Z_{\gamma,\gamma'} T^0_{\alpha\beta\gamma'}. \tag{2.16}$$

$T^{(X)}$ variation violates the Z_2 symmetry while $T^{(Z)}$ does not. That is,



(2.17)

and it was shown that $T^{(X)}$ type variations cause an instability and while $T^{(Z)}$ type variations do not. Note that, though we chose variations only on the virtual indices for simple illustration, the same conclusion applies for any random variation including those on the physical indices. However, if a variation acts *only* on the physical indices, it cannot break the Z_2 virtual symmetry, and hence would be stable.

We use the algorithm described in the section 2.2 to calculate S_{topo} of the tensor network state constructed by a local tensor with random variations added to the fixed point tensor given in Eq. (2.11). $I_V = I^{\otimes 3}$ is projector onto the full virtual space. $\mathbb{M} = \frac{1}{2}(I^{\otimes} + Z^{\otimes 3})$ is a projector on to the space of variations that respect the $Z^{\otimes 3}$ symmetries. So, $I_V - \mathbb{M}$ is a projector on to the space of variations that break $Z^{\otimes 3}$ symmetries. We first calculate S_{topo} in the state constructed by the fixed point tensor, T^0. Then we generate a random tensor T^r on the full space, project it on to the subspace $I_V - \mathbb{M}$, add it to the fixed point value, $T^0 \to T^0 + \epsilon(I_V - \mathbb{M})T^r$, and calculate $S_{\text{topo}}(\epsilon)$. Similarly, we generate a random tensor T^r on the full space, project it on to $Z^{\otimes 3}$ respecting subspace $\mathbb{M}$, add it to the fixed point value, $T^0 \to T^0 + \epsilon\mathbb{M}T^r$, and

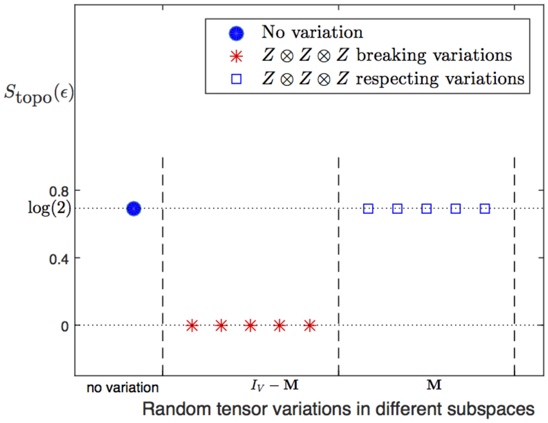


Figure 2.5: Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by toric code fixed point single-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. $I_V = I^{\otimes 3}$ is the projector on to the full virtual space. $\mathbb{M} = \frac{1}{2}(I^{\otimes} + Z^{\otimes 3})$ is the projector on to the space of variations that respect the $Z^{\otimes 3}$ symmetries. So, $I_V - \mathbb{M}$ is a projector on to the space of variations that break $Z^{\otimes 3}$ symmetries. We see that variations in $I_V - \mathbb{M}$ subspace are unstable while variations in $\mathbb{M}$ are stable. Details of this numerical calculation are given in the appendix A.2

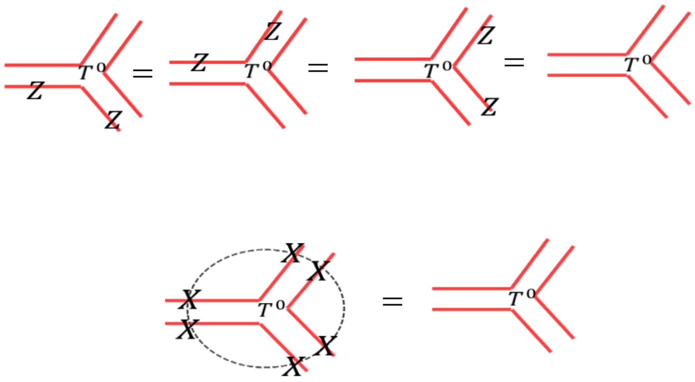
.

calculate $S_{\text{topo}}(\epsilon)$. We keep the value of variation strength $\epsilon = 0.01$ (low enough) to make sure it is not near any phase transition point. The results are shown in Fig. 2.5.

We see that $Z^{\otimes 3}$ respecting variations lead to the same topological entanglement entropy as the fixed point state, while $Z^{\otimes 3}$ violating variations lead to zero topological entanglement entropy. This reproduces the result by Chen *et al.* [47].

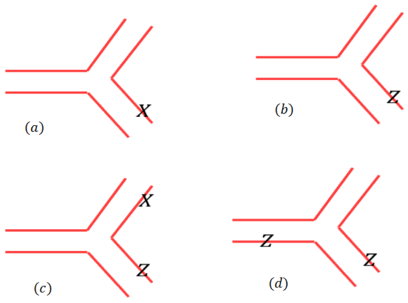
Is double-line TNR unstable too? We find that it is unstable too in certain directions of variations. Similar to the single-line case, a variation is stable or unstable depending on whether or not it violates certain virtual symmetries. So let's first look at the symmetries of the double-line TNR. It has 6 virtual indices, so the virtual space dimension is $2^6 = 64$, while the physical space dimension is again 4. So we need a symmetry group with $|G| = 64/4 = 2^4$. Indeed the tensor has a $Z_2 \times Z_2 \times Z_2 \times Z_2$ virtual symmetries. First it has a X^6 symmetry. That is, if we flip all the six virtual indices, the tensor remains the same. Second, it has 3 $Z \otimes Z$ symmetry, where $Z \otimes Z$ are applied to the two virtual indices on the same plaquette.

So the double-line tensor in (2.12) satisfies these symmetry equations:



. (2.18)

Single-line TNR had only one such Z_2 symmetry and it turned out that breaking it results in phase transition. For double-line we have four Z_2 symmetries. So the question is, are all of them important? That is, is it the case that breaking any of them with a variation leads to instability? Indeed many different possible kinds of variations are possible, for example:



. (2.19)

A variation can violate $Z \otimes Z$ but not $X^{\otimes 6}$ (for example, 2.19(a)), or it can violate $X^{\otimes 6}$ but not $Z \otimes Z$ (for example, 2.19(b)), or it can violate both (for example, 2.19(c)), or it can violate neither(for example, 2.19(d)), etc. So to find out, we need to look at the unstable directions of variations of the fixed point tensor.

Our numerical calculation reveals an interesting result. We find that (see Fig. 2.6)

1. Variations that break $Z \otimes Z$ but not $X^{\otimes 6}$ (an X variation, for example) are stable.

2. Variations that break $X^{\otimes 6}$ but not $Z \otimes Z$ (a Z variation, for example) are unstable.

3. Variations that break both $X^{\otimes 6}$ and $Z \otimes Z$ (a ZX variation, for example) are stable.

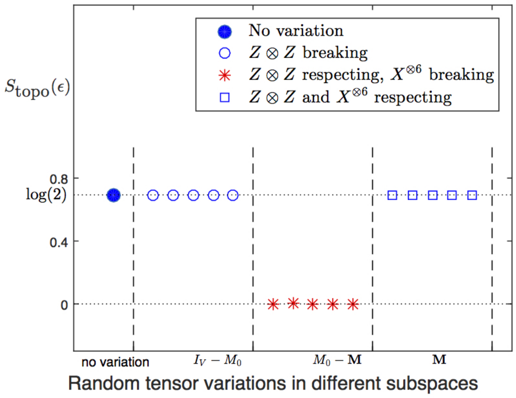


Figure 2.6: Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of the states represented by toric code fixed point double-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Variations in $I_V - M_0$ violate $Z \otimes Z$ symmetry. Variations in $M_0 - \mathbb{M}$ violate $X^{\otimes 6}$ but not $Z \otimes Z$. Variations in $\mathbb{M}$ violate no virtual symmetry. The details of this numerical calculation are given in appendix A.2.

To understand it physically, we first have to understand the notion of *'stand-alone'* variations of a TNR. So we see that the relation between unstable variations and virtual symmetries of the double-line TNR is more complicated than that for single-line TNR. The $X^{\otimes 6}$ symmetry looks like the $Z^{\otimes 3}$ symmetry of the single-line TNR, as they both operate as a loop operators, and unstable variations in both TNR violates these loop symmetries. However, the crucial difference in double-line is that then there are extra symmetries (the 3 $Z^{\otimes 2}$ symmetries) which have an exact opposite relation with the unstable variations: a variation is actually *stable* when it violates these symmetries (irrespective of whether or not it violated the loop symmetry). It indicates that the sources of these two kinds of symmetries must be different. How can we understand this phenomena? We will now show that the sources of these symmetries are indeed different, and it is the interplay between these two symmetries that determines the tensor instability phenomena. The symmetries whose violation causes instability comes from the so-called *MPO subspace* of the virtual space, while the symmetries whose violation causes stability comes from what we define to be *stand-alone subspace.* We will define these subspaces and calculate them for the two TNR in the next two sections.

2.4 Virtual subspaces of a TNR: 'Stand-alone' subspace

A generic tensor is can be thought of as a linear map from virtual vector space to physical vector space. The virtual space has two important subspaces and play the defining role in determining tensor instabilities. We will explain them now.

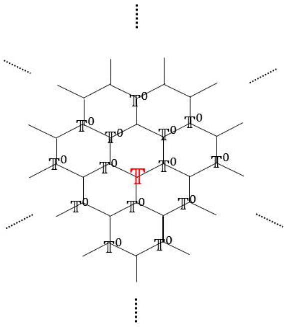


Figure 2.7: Replacing a fixed point double tensor $\mathbb{T}^0$ in the tensor network with a tensor variation $\mathbb{T}$. A varied tensor network is a superposition of wave functions where fixed point tensors have been replaced by the variation. But for most variations, this tensor network collapses (i.e. becomes zero).

Double tensor $\mathbb{T}$ of a tensor T is defined as $\mathbb{T} = \sum_I T^I_\alpha (T^*)^I_{\alpha'}$. It can be interpreted as a density matrix on the virtual space. Now consider the double tensor of a RG fixed point TNR, $\mathbb{T}^0$, contracted over some large region R. Let's say we remove $\mathbb{T}^0$ from one site and replace it with some other double tensor, $\mathbb{T}$. What do we get? In particular, are there tensors $\mathbb{T}$ such that this replacement collapses the whole tensor network? By collapse, we mean that we simply get zero upon contraction. The answer turns out to be, yes. In fact, as we see later, *most* tensors $\mathbb{T}$ will collapse the fixed point tensor network upon replacement. It turns out that only the tensors supported on a particular subspace of the full virtual space can replace the fix point tensor without collapsing the whole tensor network. We will call this space *the stand-alone subspace of the TNR*. Now we will give a systematic way of calculating this subspace for a given fixed-point TNR.

Consider contracting the fixed-point double tensors $\mathbb{T}^0$ on a large disc with an open boundary. Now we remove the tensor at the origin. This tensor network will have dangling virtual indices at the origin and at the boundary of the disc. We want to find

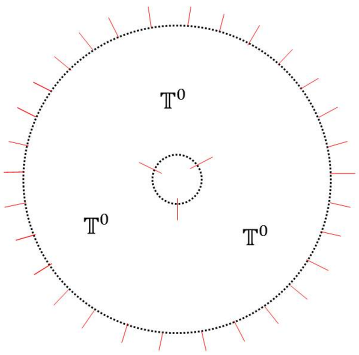


Figure 2.8: Calculation of stand-alone space. We put the fixed point double tensor network on a large disc with a hole at the origin (one double tensor removed). This tensor network has dangling virtual indices (red legs) at the outer and inner boundaries. We trace out the virtual indices at the outer boundary, and the support space of the remaining tensor at the inner boundary gives us the stand-alone space.

out the space of tensors that can be put on the origin without collapsing the tensor network. We do not care what tensor at the boundary we get. So we trace out the indices at the outer boundary (i.e. contract α and α' with each other). This leaves us with a tensor at the origin. The support space of this tensor will be precisely the stand-alone space. Any tensor supported on this subspace can stand alone with the surrounding tensor being the fixed-point tensors.

Now let's first calculate the stand-alone subspace of double-line TNR of toric code as it is more interesting than that of single-line TNR. The double tensor of T^0 in (2.12) can be written as (ignoring an overall normalization factor)

$$\begin{aligned}\mathbb{T}^0 &= \sum_I (T^0)^I_\alpha (T^{0;*})^I_{\alpha'} \\ &= (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + X^{\otimes 6}),\end{aligned} \tag{2.20}$$

where the double tensor is written as an operator between the lower virtual indices and upper virtual indices. The $Z^{\otimes 2}$ and $X^{\otimes 6}$ act in the way it is shown in Fig. 2.18. We need to contract this tensor on a disc with a hole at the origin. To contract two tensors given in an operator form, we need to multiply them and take a trace on the shared indices. A cumbersome but straight-forward calculation shows that double tensor contracted on a region R give (ignoring an overall normalization factor)

$$\begin{aligned}\mathbb{T}^0(R) &= \left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes m}(\partial R) \\ &\quad \left(I^{\otimes 2m}(\partial R) + X^{\otimes 2m}(\partial R)\right),\end{aligned} \tag{2.21}$$

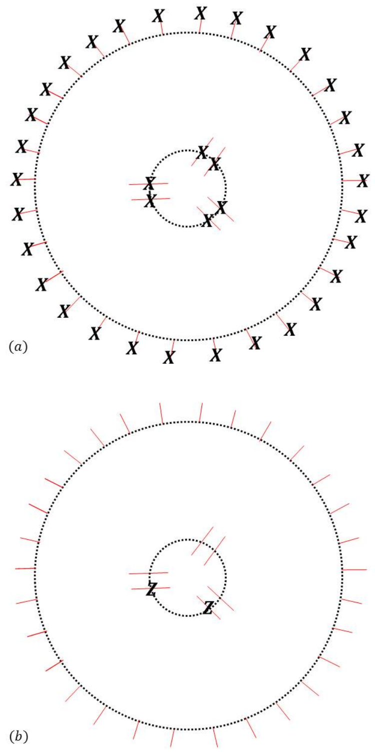


Figure 2.9: X and Z operators appear differently in the toric code double-line double tensor contracted on a disc with a hole. X operators on inner boundary only appears with X operators on the outer boundary, which vanishes upon taking the trace. But Z operators on the inner boundary appear with identity on the outer boundary. So these terms survive the trace. That is why $Z^{\otimes 2}$ symmetry is imposed on the stand-alone space but not the $X^{\otimes 6}$ symmetry.

where ∂R denote the boundary of R, and $m = |\partial R|$ is the length of the boundary. $O(\partial R)$ means the operator O is applied on the virtual legs along the boundary ∂R. We will omit this when it is clear from the context which leg the operator is being applied on. The region we want is a disc with a vertex removed, $R = D_{2m} - D_6$. D_n denotes the disc with n virtual legs at the boundary. It has two disconnected boundaries, one the boundary of D_{2m} and other the boundary of D_6 (with opposite orientation).

$$\begin{aligned}\mathbb{T}(D_{2m} - D_6) &= \left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes m} \otimes \left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes 3} \\ &\quad \left(I^{\otimes 2m} \otimes I^{\otimes 6} + X^{\otimes 2m} \otimes X^{\otimes 6}\right). \qquad (2.22)\end{aligned}$$

As explained in Fig. 2.9, X operators act on the two boundaries simultaneously, but

Z operators act independently. Now to get the stand-alone space at the origin, we need to trace out the virtual legs at the boundary of D_{2m}. If we expand the expression for $\mathbb{T}(D_{2m} - D_6)$ above and apply trace on the operators on the outer boundary, only the terms with identity on the outer boundary survive. X operator does not have such a term, but Z does. So finally, tracing out the outer boundary leaves only $Z^{\otimes 2}$ on the inner boundary. That is, we get the following tensor on the 6 virtual indices incident on a singe vertex

$$B_0 = \left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes 3}. \tag{2.23}$$

$B_0^2 = 8B_0$, so

$$M_0 = \frac{1}{2}B_0 = \frac{1}{8}\left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes 3} \tag{2.24}$$

is a projector on to the support space of B_0. M_0 defines the stand-alone space of double-line TNR of toric code. Any tensor T that satisfies $M_0 T \neq 0$ can 'stand alone'. M_0 will be used to denote the projector on to stand-alone space throughout the paper. So we see that *only* the tensors that respect the $Z \otimes Z$ symmetry can stand alone. The $X^{\otimes 6}$ symmetry, however, is not required to define the stand-alone space.

We can also calculate the stand-alone subspace of single-line TNR of toric code. One can calculate the double tensor of single-line TNR in Fig. 2.11 to be

$$\mathbb{T}^0 \;=\; \frac{1}{2}(I^{\otimes 3} + Z^{\otimes 3}). \tag{2.25}$$

This double tensor upon contraction on the disc with a hole at the origin $(D_m - D_3)$ gives (up to an overall normalization)

$$\mathbb{T}^0(D_m - D_3) \;=\; I^{\otimes m} \otimes I^{\otimes 3} + Z^{\otimes m} \otimes Z^{\otimes 3}. \tag{2.26}$$

Now contracting the outer circle gives us

$$B_0 = M_0 = I^{\otimes 3}. \tag{2.27}$$

So we see that, for single-line TNR, the stand-alone subspace is actually *all* of the virtual space. That is, there are no tensors that cannot stand alone.

2.5 Virtual subspaces of a TNR: MPO subspace

As explained above, stand-alone subspace is the maximal virtual subspace such that any tensor supported on this subspace can be inserted into the tensor network without collapsing it. Therefore, the virtual space of the RG fixed point tensor, T^0 itself must be inside the stand-alone subspace. This virtual space of T^0, which is by definition of subspace of stand-alone space, is what we would call the MPO (Matrix Product Operator) subspace. The reason for calling it an MPO space is that, as it turns out, this space is protected by symmetry operators which are matrix product operators. This was shown for general string-net models in [16], but it can be extended to other tensor networks as well. Let's make the notion of virtual space of $(T^0)^I_\alpha$ precise, whereas before, I represents the collection of physical indices and α represents the collection of virtual indices. We can think of it as a matrix with indices α and I and perform an SVD decomposition,

$$(T^0)^I_\alpha = \sum_{\alpha',I'} V_{\alpha,\alpha'}\Lambda_{\alpha',I'}P_{I',I}, \tag{2.28}$$

where V and P are unitary matrices in virtual and physical space, and Λ is the diagonal matrix containing the singular values.

Definition 1. *The MPO space, defined as the virtual support space of T^0,is the virtual subspace spanned by columns of V for which corresponding singular value is non-zero.*

Another way to think about this is to again consider the double tensor $\mathbb{T}^0_{\alpha,\alpha'} = \sum_I (T^0)^I_\alpha (T^{0;*}))^I_\alpha$ which is a matrix in the virtual space. An equivalent but more useful definition is,

Definition 2. *The MPO space is the space spanned by eigenvectors of $\mathbb{T}^0$ with nonzero eigenvalues.*

Using Eq. 2.28 we can write

$$T^0 = \sum_{j;\lambda_j\neq 0} \lambda_j \left|v_j\right\rangle \left\langle p_j\right| \tag{2.29}$$

$$\Rightarrow \mathbb{T}^0 = \sum_{j;\lambda_j\neq 0} \lambda_j^2 \left|v_j\right\rangle \left\langle v_j\right|, \tag{2.30}$$

where λ_j are the singular values and v_j and p_j are the corresponding vectors in virtual and physical space. MPO space is the space spanned by v_j, so the projector on this space is

$$\mathbb{M} = \sum_{j;\lambda_j\neq 0} \left|v_j\right\rangle \left\langle v_j\right|, \tag{2.31}$$

The mathematical understanding of the MPO space is that it is the virtual subspace which is isomorphic to the ground state physical subspace and T^0 is the isomorphism. MPO subspace by definition nested inside the stand-alone subspace, which by definition is nested inside the full virtual space. Similarly, the ground-state physical space is by definition a subspace of the full physical space. These spaces are represented visually in Fig. 2.6 for clarity.

MPO subspace of single-line TNR

We can write the single-line tensor in Eq. 2.11 in the Eq. 2.28 form,

$$T^0 = |000\rangle \langle 000| + |011\rangle \langle 011| + |101\rangle \langle 101| + |110\rangle \langle 110| . \tag{2.32}$$

Of course, this happened to be already written in SVD decomposed form. So the MPO space is spanned by vectors $\{|000\rangle, |011\rangle, |101\rangle, |110\rangle\}$. So the MPO projector is

$$\begin{aligned} \mathbb{M} &= |000\rangle \langle 000| + |011\rangle \langle 011| + |101\rangle \langle 101| + |110\rangle \langle 110| && (2.33) \\ &= \frac{1}{2}\left(I^{\otimes 3} + Z^{\otimes 3}\right). && (2.34) \end{aligned}$$

We see that this projector can be written as a translation invariant superposition of tensor product of matrices. That's why we call it the MPO subspace. Remember that stand-alone space of single-line TNR was determined to be all of virtual space, $M_0 = I_V$. So as expected, $\mathbb{M} \subset M_0$.

MPO subspace of double-line TNR

For calculation of the MPO subspace of double-line tensor in Eq. (2.12) we use the second definition in 2 above to avoid a cumbersome but straight forward calculation. We already calculated the double tensor of double-line TNR in Eq. 2.20. We ignored the normalization factor there. If we use a normalization factor of $\frac{1}{16}$ and write

$$\begin{aligned} \mathbb{M} = \frac{1}{8}\mathbb{T}^0 &= \frac{1}{16}\sum_I (T^0)^I_\alpha (T^{0;*})^I_{\alpha'} \\ &= \frac{1}{16}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + X^{\otimes 6}). && (2.35) \end{aligned}$$

Then $\mathbb{M}$ is a projector, that is, it satisfies $\mathbb{M}^2 = \mathbb{M}$, and has the same support as $\mathbb{T}^0$ hence this is the desired MPO projector. Remember that stano bed-alone projector was calculated to be $M_0 = \frac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3})$, and hence $\mathbb{M} = M_0\frac{1}{2}(I^{\otimes} + X^{\otimes 6}) \subset M_0$, as expected.

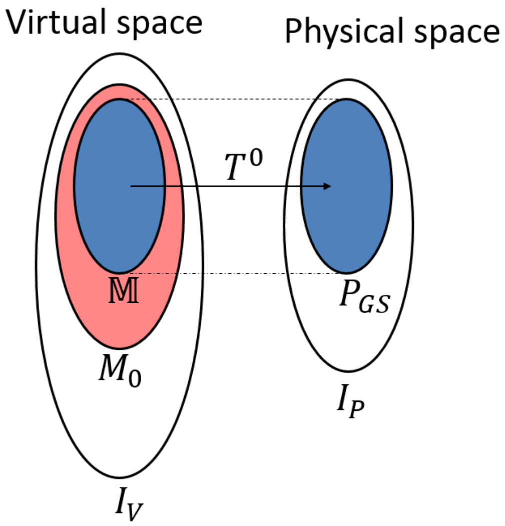


Figure 2.10: A pictorial representation of relevant vector spaces. A tensor, T^0, is a linear map from the virtual space to the physical space. We denote the full virtual space as I_V and the full physical space as I_P. M_0 (region in red and blue) is the stand-alone subspace of the virtual space. $\mathbb{M}$ (region in blue) is the MPO subspace of the stand-alone space. MPO subspace is isomorphic to the local ground-state physical subspace (also in blue), denoted as P_{GS}, which is a subspace of the full physical space.

2.6 TNR instability conjecture

Now that we have defined the stand-alone and MPO subspaces precisely, we are ready to state the central conjecture of this work.

Conjecture 1. *If, for a given RG fixed point TNR, T^0, of a topological state, M_0 and $\mathbb{M}$ are the projectors onto the stand-alone and MPO subspaces as defined above, then an infinitesimal tensor variation $T^0 \to T^0 + \epsilon T$ changes the topological phase of the state if and only if $(M_0 - \mathbb{M})T \neq 0$.*

It implies that the projector onto the stable space is $P_S = I_V - (M_0 - \mathbb{M})$.

Corollary 1. *An infinitesimal variation T does not change the topological phase if and only if $P_S T = T$.*

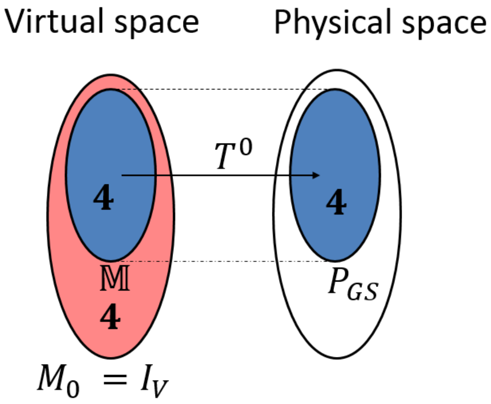
Corollary 2. *For any tensors T, the variation $T^0 \to T^0 + \epsilon P_S T$, with $\epsilon \ll 1$ does not change the topological phase.*

Or in simple words, *a variation is unstable if and only if it has a component in the stand-alone space that is outside the MPO subspace.* A Venn diagram of the decomposition of the virtual space through these projectors is shown in Fig. 2.6. We will denote $(M_0 - \mathbb{M})$ as P_U for convenience. Note that P_U shouldn't be thought of as 'the projector onto unstable subspace' because unstable variations do not form a vector space, as opposed to stable variations that do form a vector space. It is because $P_U T^1 \neq 0$ and $P_U T^2 \neq 0$ does not imply $P_U(T^1 + T^2) \neq 0$.

Let's first see how this conjecture is true for the single-line and double-line TNR of the toric code. For single-line we have already calculated the stand-alone and MPO subspaces and found $M_0 = I_V$ and $\mathbb{M} = \frac{1}{2}\left(I^{\otimes 3} + Z^{\otimes 3}\right)$. So,

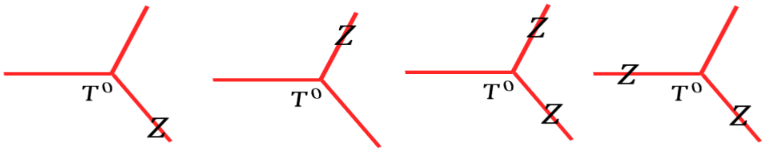
$$P_U = M_0 - \mathbb{M} = \frac{1}{2}\left(I^{\otimes 3} - Z^{\otimes 3}\right). \tag{2.36}$$

So for a tensor $P_U T \neq 0$ if and only if it violates the $Z^{\otimes 3}$ symmetry. Indeed, this is exactly what we saw numerically in Fig 2.3. All variations can be summarized visually using Fig. 2.6 as follows:



(2.37)

Variations supported on the red region are unstable, while those on blue and white are stable. The dimension of each space is indicated. So we see that the space $(M_0 - \mathbb{M})$ is 4 dimensional space spanned by basis



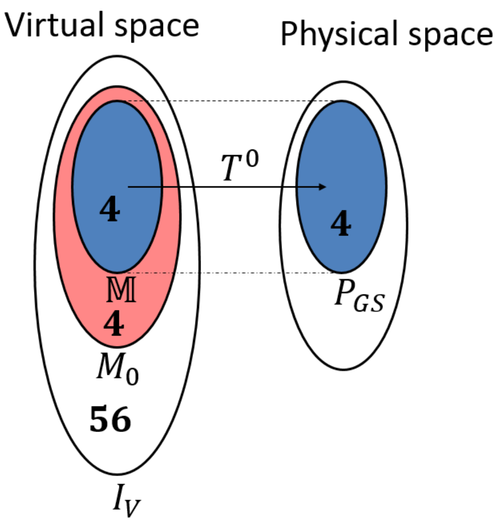
(2.38)

It is wrong to think that these basis set span the space of unstable variations, because unstable variations do not form a vector space. All we can say is if a variations has overlap with any of these basis, it would cause instabilty.

For double-line TNR we found, $M_0 = \frac{1}{8}\left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes 3}$ and $\mathbb{M} = M_0\frac{1}{2}(I^{\otimes} + X^{\otimes 6})$. So,

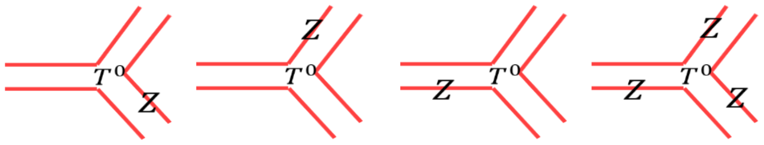
$$P_U = M_0 - \mathbb{M} = \frac{1}{8}\left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes 3}\frac{1}{2}(I^{\otimes} - X^{\otimes 6}) \tag{2.39}$$

So $P_U T \neq 0$ if and only if T satisfies the three $Z^{\otimes 2}$ symmetries but violates the $X^{\otimes 6}$ symmetry. Indeed this is precisely what we found numerically as shown in Fig. 2.4. All variations can be summarized visually as follows



(2.40)

Variations supported on the red region are unstable, while those on blue and white are stable. The dimension of each space is indicated. So we see that the space $(M_0 - \mathbb{M})$ is 4 dimensional space spanned by basis



(2.41)

The above conjecture provides a precise mathematical prescription of which tensor variations cause instability. But what does this instability correspond to physically? Why is it that a tensor causes instability if and only if it has a component in the virtual subspace $M_0 - \mathbb{M}$? To answer these questions we need to understand the physical significance of stand-alone and MPO subspaces. We present the physical interpretation of the above conjecture that we will explain in detail in the next few sections:

Conjecture 2. *Variations in stand-alone subspace M_0 correspond to 'bosonic excitations' that can condense in the given TNR. Variations in the MPO subspace $\mathbb{M}$ are the subset of these condensable 'bosons' that are trivial (belong to trivial superselection sector). Hence the variations in $M_0 - \mathbb{M}$ are the non-trivial condensable bosons. So such a variation results in boson condensation and causes a topological phase transition of the state.*

By excitation we mean any point-like variation to the ground state, or its TNR. It should be carefully noted that the word 'boson' here refers to any point like excitation (not necessary an irreducible excitation) that has trivial topological spin. For example, if a is an anyon of the given model, then composite particle $a\bar{a}$ where a and $\bar{a}$ are sitting next to each other is included in this definition of boson. Of course it is a topologically trivial boson. Similarly, if we apply any local operation on the topological state, we would call the resulting state containing a boson, though of course, it is again a trivial boson.

Now we turn to the first part of the claim, which is basically the physical significance of stand-alone space

2.7 Physical significance of stand-alone space M_0

As claimed, the physical significance of stand-alone space is that it contains proliferatable bosonic excitations.

Proliferatable variations of a TNR

First we explain what we mean by 'Proliferatable variations/excitations'. (We use the term 'variation' for any mathematical variation to the ground state tensors. 'Excitation' should be used for a quasi-particle excitation. But in slight abuse of the nomenclature we would often use them interchangeably. It is justified as we are only working with the wave functions and not Hamiltonians.) Let's say T^0 is the RG fixed point tensor of some topological ground state wave function $|\Psi_0\rangle$. Let's say we add a variation, $T^0 \longrightarrow T^0 + \epsilon T$ and the resulting wave function is $|\Psi\rangle$.

$$\begin{aligned}
|\Psi_0\rangle &= \sum_{\{i_j\}} (T^0)^{i_1} (T^0)^{i_2} \dots (T^0)^{i_n} \, |i_1 i_2 \dots i_n\rangle \,, \\
|\Psi\rangle &= \sum_{\{i_j\}} (T^0 + \epsilon T)^{i_1} (T^0 + \epsilon T)^{i_2} \dots \times \\
& \quad (T^0 + \epsilon T)^{i_n} \, |i_1 i_2 \dots i_n\rangle \\
&= |\Psi_0\rangle + \epsilon \sum_{s_1} \epsilon \left|\Psi_{s_1}\right\rangle + \epsilon^2 \sum_{s_1, s_2} \left|\Psi_{s_1, s_2}\right\rangle + \dots, \qquad (2.42)
\end{aligned}$$

where $|\psi_{s_1}\rangle$ denotes the tensor network state similar to $|\Psi_0\rangle$ except T^0 has been replaced with T at site s_1. Similarly, $\left|\Psi_{s_1,s_2}\right\rangle$ denotes the tensor network state similar to $|\Psi_0\rangle$ except T^0 is replaced with T at site s_1 and s_2. Higher order terms can be understood in a similar manner. Physically, $|\psi_{s_1}\rangle$ can be interpreted as 'excitation' T (which may be trivial) sitting at site s_1 with probability ϵ^2. Similarly, $\left|\Psi_{s_1,s_2}\right\rangle$ can be interpreted as excitation T sitting at sites s_1 and s_2 with probability ϵ^4. Higher order terms can be interpreted in a similar fashion. Though ϵ^2 looks small compared to the weight of $|\Psi_0\rangle$, one has to bear in mind there are $\sim N$ such terms in the expansion, where N is the number of sites. So after normalization they can have comparable weights.

When T is in the stand-alone space then it can appear anywhere in the tensor network state, independent of each other, even at large scales. However when T is outside of the stand-alone space, then it can at most appear next to other Ts. But then the distance between excitations is exponentially suppressed since each T appears with an ϵ weight. So such excitations do not appear at large scale and would vanish under RG process. Tensors within the stand-alone space, on the other hand, can appear at any scale and would not vanish under RG process. So we can call the new wave function as a 'proliferation/condensate of T', since the variation/excitation T proliferates and each site is in superposition of T appearing and not appearing at all length scales. (We caution that we use the term 'proliferation' to denote the mathematical fact that the wave function is a superposition of a variation appearing everywhere. While the term 'condensate' in physics means something more specific. But, again, we would use these terms interchangeably. It is justified as we are not dealing with the Hamiltonians, rather looking at the changes in the wave functions as we vary the tensors. So the 'condensation of variations' doesn't necessarily mean a phase transition. It just means a particular mathematical variations, which can be interpreted as an excitation, proliferates and the resulting wave function is a superposition of this variation appearing everywhere.)

A key point here is that v_1 and v_2 can be at arbitrary distance from each other but the contribution of this term in the superposition remains ϵ^2. Let's compare this with how the ground state changes with respect to a perturbation on the Hamiltonian level. Let's perturb the toric code Hamiltonian in (2.9) with X perturbations on every link,

$$H = H_0 + \epsilon \sum_l X_l. \tag{2.43}$$

The ground state of this perturbed Hamiltonian is also a superposition of $|\Psi_0\rangle$ and terms like $|\Psi_{v_1,v_2}\rangle$. But the weight that appears with $|\Psi_{v_1,v_2}\rangle$ is of the order of $\epsilon^{\text{distance}(v_1,v_2)}$, that is, the separation between two e particles is exponentially suppressed. So, in thermodynamic limit, these excitations disappear. But this is not the case with state in Eq. (2.42). That is why the state in Eq. (2.42) cannot be produced by infinitesimal small local perturbation of the parent Hamiltonian.

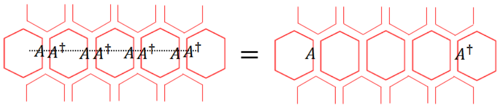
So we have argued that stand-alone space, by definition, is the space of variations that can condense. But how do we know they are 'bosonic excitations', that is, they have a trivial topological spin? We will show it now.

Condensable excitations are 'bosons'

Consider the tensor network state which has the fixed point tensor T^0 everywhere except at sites s_1 and s_2, where T^0 has been replaced by stand-alone tensors T. We denote this wave function as $|\Psi_{s_1,s_2}\rangle$, as above. Topological spins of quasi-particles in topological models are calculated using the string-operators that create them. So we need to first define a string-operators that create these variations. The anyonic string operators in topological models have the property that they commute with the Hamiltonian everywhere except possibly at its ends. But we are working directly with the quantum wave function and are not really concerned with the underlying Hamiltonian, whose form can change going away from the RG fixed point. We see that we can define an appropriate string-operator for tensor network states without referring to a Hamiltonian. To do that, first notice that every tensor network state has underlying gauge symmetries at the virtual level. That is, if we apply operators A and B on the two contracting virtual legs, such that $AB = I$, the tensor network state doesn't change (though the individual tensors may change). That is,

$$. \tag{2.44}$$

It means that if we apply a string of A, B on virtual levels along a path, the tensor network state would not change along the path but only at the ends. For example, on the double-line TNR we can create a stand-alone excitation A in the following way,



. (2.45)

The A and $A^\dagger$ cancel each other on each plaquette as all the 6 virtual legs are contracted. We chose double-line tensor network for illustration but of course it can be done for any tensor network. So wave functions like $\left|\Psi_{s_1,s_2}\right\rangle$ can be created by such string operators. Note that since the tensor network didn't change along the path, $\left|\Psi_{s_1,s_2}\right\rangle$ is still in the ground state along the path. So this string operator can only possibly create excitations at the ends, which is what we wanted. We will call such string operators *gauge-string-operators* to distinguish them from the usual string operators on the physical level. Note that gauge-string-operators can only create stand-alone variations/excitations, and they are deformabale on the ground state subspace, like the physical string operators. We know that physical string-operators might not be deformed through a site that has an anyonic excitation present. Gauge-string operators also may not be be deformed through excitations. For example, there may be another operator C present at the virtual legs such that $ACB \neq C$. But the interesting thing to note is that they can always be deformed through a stand-alone excitation. The reason for this is simple. A stand-alone tensor is surrounded by fixed point tensor T^0. So if we consider a Wilson-loop of gauge-string operator around it, $AB = I$ is still true, so A and B will simply cancel each other. So the Wilson-loop will simply disappear irrespective of what stand-alone excitation was there. So not only gauge-string operators create stand-alone excitations, they also always commute with the other stand-alone excitations. This suggests that all excitations in the stand-alone space have trivial mutual and self stastics. But to prove they are bosons, we need to do the topological spin calculation. Though again it can readily be seen that the topological spin of stand-alone excitation is 1, as explained in Fig. 2.11.

We create two pairs of $a - \bar{a}$, where a is some excitation of the model. For convenience, to keep track of the excitations we have shown them in different colors, red and blue. Let's say the blue a and red a sit at sites 1 and 2 respectively. Now we apply the following procedure in this specific order: move red a from 2 to 3 (Fig. 2.11a), move blue a from 1 to 2 (Fig. 2.11b), move red a from 3 to 1 a (Fig. 2.11c). And finally annihilate blue a with red $\bar{a}$ and *vice-versa*. In general, the order in which each process is done is important. However, when the string operator is an

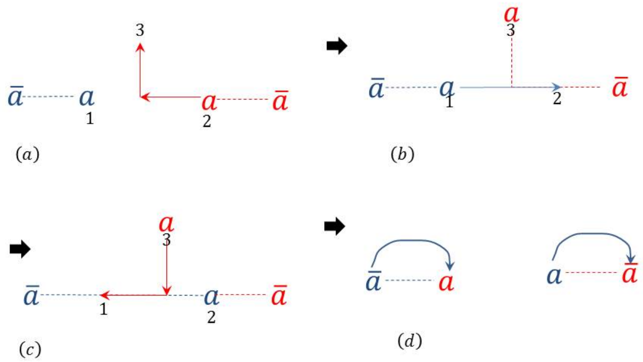


Figure 2.11: Calculation of topological spin. We create two pairs (shown as red and blue) of particle, anti-particle pairs $a-\bar{a}$, with a situated at site 1 and 2. We apply the following procedure in this order: (a) Move first a (red) from 2 to 3, (b) move second a (blue) from 1 to 2, (c) move first a (red) from 3 to 1. Finally, (d) we annihilate each a with the anti-particles of the other anyon (i.e. red a with blue $\bar{a}$ and *vice versa*). When the propagation of a happens through a gauge-string operator, which disappears along the path, this order of process becomes irrelevant, as the second string-operator does not interact with the first one, and the whole process is equivalent to creating and annihilating two pairs of $a-\bar{a}$, which has amplitude 1. It implies a has a trivial topological spin.

guage-string operator, it disappears along the path completely, and the second string operator does not interact with it. So the whole process simply becomes creating and annihilating two pairs of $a-\bar{a}$ which has an amplitude 1, giving us a topological spin equal to 1. So a has to be a boson of the model.

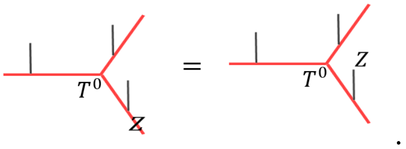
We have determined that the variations in the stand-alone space are condensable bosons. So any such variation results in a wave function which is a condensate of the boson the variation corresponds to. But this alone does not necessarily cause a phase transition, because if the boson was topologically trivial, there should be not topological phase transition. Or, mathematically speaking, the stand-alone projector projects out variations that cannot proliferate, but it doesn't project out those stable variations that can proliferate. For example, the double-line stand-alone projector does not project out variation $X \otimes X$ though it is not unstable. So to find the unstable variations, we need an additional projector to project out condensable but stable variations. We will argue now MPO subspace is precisely this projector.

To find out whether a virtual variation would cause the phase transition we need to first determine what this virtual variation corresponds to on the physical level. That is, we need to 'lift' the variation from the virtual level to the physical level. When we do that we discover that there are two kinds of variations: The first kind is where a local virtual variation is lifted to a local physical variation, and the second kind is where the local virtual variation is lifted to a *non-local* physical variation. We know that a local physical variation can only correspond to a topologically trivial boson since it can be removed by a local operation. This distinction between variations further decomposes the stand-alone into two subspaces: the MPO space $\mathbb{M}$, which corresponds to the first kind of variation, and the unstable subspace $M_0 - \mathbb{M}$, which corresponds to the second kind of variation. Let's first focus on the first kind of variations.

2.8 Physical significance of MPO subspace $\mathbb{M}$

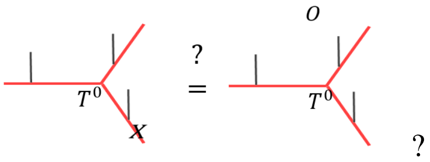
As claimed above, the physical significance of MPO subspace is that the variations in this subspace are lifted to local physical variations, which have to be topologically trivial bosons since they can be removed by a local operation. Hence the physical significance of MPO subspace is that it contains all the topologically trivial excitations.

To understand it better, let us look at concrete examples of variations that are lifted to local physical variation. Consider a Z variation on the virtual leg of the fixed point single-line TNR. If we lift it to the physical level, what do we get? Since the virtual legs are just copies of the physical legs, we get



(2.46)

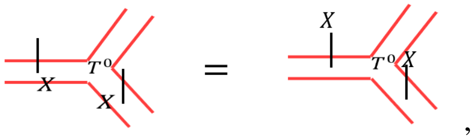
So Z virtual variation is lifted to a Z physical variation, which is local. According to our claim, it should be in the MPO subspace. And indeed it is, since it respects the MPO symmetry of the single-line TNR, $Z^{\otimes 3}$. Also note that a Z physical variation corresponds to a pair of m particles sitting next to each other, not a single m particle. It is a trivial excitation and can be removed by applying one Z operation on the state, so it matches our claim. Contrast this with the X variation on the virtual level. Can we find any local physical operator O such that



(2.47)

One can try and see that there is no such local operator O for which this equation holds. (We will later show that X can be lifted to the physical level, but it results in a non-local operator.)

A similar phenomena occurs in double-line TNR. The $X \otimes X$ variation can be lifted to a local physical operator,



(2.48)

but a Z variation cannot be. (We will later sVhow that Z can be lifted to the physical level, but it results in a non-local operator.) It is again consistent with the claim as $X \otimes X$ variation respects the double-line MPO symmetry ($X^{\otimes 6}$) but Z variation breaks it. Note that $X \otimes X$ variation on the physical level corresponds to a pair of e particles sitting across a plaquette. It is a topologically trivial excitation and can be removed by an $X \otimes X$ operation on the state. So again, this matches our claim.

Now we prove that these examples are no coincidence, and in fact any variation in the MPO subspace is a local physical variation.

Any variation in MPO subspace is a local physical variation

Let us repeat the definition of the MPO subspace here for convenience. We SVD decomps the fixed point RG tensor T^0 as a matrix between virtual and physical legs

$$T^0 = \sum_j \lambda_j \left|v_j\right\rangle \left\langle p_j\right|, \tag{2.49}$$

where λ_j are the singular values, and v_j and p_j are orthonormal vectors in the virtual and ground-state physical spaces respectively. Then the MPO subspace is the virtual subspace spanned by vectors v_j such that corresponding singular values $\lambda \neq 0$. So the MPO projector is

$$\mathbb{M} = \sum_{j;\lambda_j \neq 0} \left|v_j\right\rangle \left\langle v_j\right|. \tag{2.50}$$

A mathematically inclined reader would note that MPO subspace is nothing but the virtual subspace which is isomorphic to the image of the tensor as a map from virtual to ground state physical space. That is, if we restrict the domain of the tensor to this subspace, then tensor is an injective map from the virtual to the physical space, and a bijective map from MPO subspace to ground-state physical subspace. *Since* *these spaces are isomorphic, any operator in MPO subspace can be mapped to an operator in the ground-state physical space and vice-versa, and this mapping would be bijective (one-to-one) as well.* Let us make it precise,

Lemma 1. *If A is any operator on the virtual space completely supported on $\mathbb{M}$ ($\mathbb{M}A = A\mathbb{M} = A$) then there exists an operator B on the ground state physical space such that $AT^0 = T^0B$ and vice-versa. That is, any variation in the subspace $\mathbb{M}$ is equivalently a variation on the ground state physical space and vice-versa.*

Proof. Let us say an virtual operator A is given which is completely supported on subspace $\mathbb{M}$. Define pseudo-inverse of T^0 as

$$T^{0;+} = \sum_{j;\lambda_j \neq 0} \frac{1}{\lambda_j} \left|p_j\right\rangle \left\langle v_j\right| . \tag{2.51}$$

It is a pseudo-inverse since

$$\begin{aligned} T^0 T^{0;+} &= \mathbb{M} && (2.52)\\ T^{0;+} T^0 &= P_{GS}, && (2.53) \end{aligned}$$

where $P_{GS} \subset I_P$ denotes projector on the ground-state physical subspace of the full physical space. I_P is the projector onto the full physical space. Now define a physical operator B as

$$B = T^{0;+} A T^0, \tag{2.54}$$

then

$$T^0 B = T^0 T^{0;+} A T^0 = \mathbb{M} A T^0 = A T^0. \tag{2.55}$$

The last equality follows from the assumption that A is completely supported on MPO subspace. Similarly, given physical operator B on the ground-state physical space, define $A = T^0 B T^{0;+}$. So we have $AT^0 = T^0 B T^{0;+} T^0 = T^0 B I_P = T^0 B$. And of course these maps are injective. So A and B have a one-to-one correspondence. □

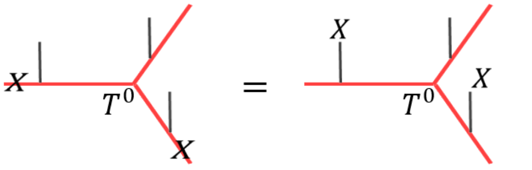
With this lemma, we see why in general variations in the MPO subspace are trivial excitations. They are nothing but a local variation on the physical level, which is a local physical operator and can be removed by another local operator. In fact notice that if A is unitary (within the space $\mathbb{M}$) then so is B and vice-versa. Since all trivial excitations are obtained by local unitaries, or their linear combinations, we conclude that MPO subspace should contain all trivial virtual excitations as well.

This completes the study of first kind of variations (those that are lifted to local physical variations) mentioned above. Now we study the second kind of variations.

2.9 Physical significance of subspace $M_0 - \mathbb{M}$

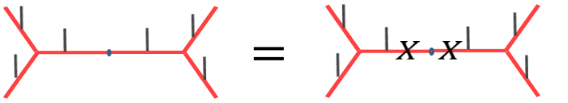
The physical significance of subspace $M_0 - \mathbb{M}$ is that it contains the second kind of variations: the virtual variations that are lifted to a non-local physical operator. So these variations cannot be removed by a local physical operation on the state, and hence represent a topologically non-trivial excitation. And this excitation has to be a boson, as all excitations in stand-alone space are. So it means that the physical significance of $M_0 - \mathbb{M}$ space is that it contains condensable excitations that are topologically non-trivial bosons, and that's why these variations cause a topological phase transition.

Let us first look at some concrete examples to understand this phenomena. We saw how a virtual X variation on the single-line TNR couldn't be lifted to a local physical variation. But, the question is, can it be lifted to a non-local physical variation? The answer is, yes. To see it, first note that although a single X variation cannot be lifted locally, two X variations can be. That is, the fixed point single-line tensor satisfies



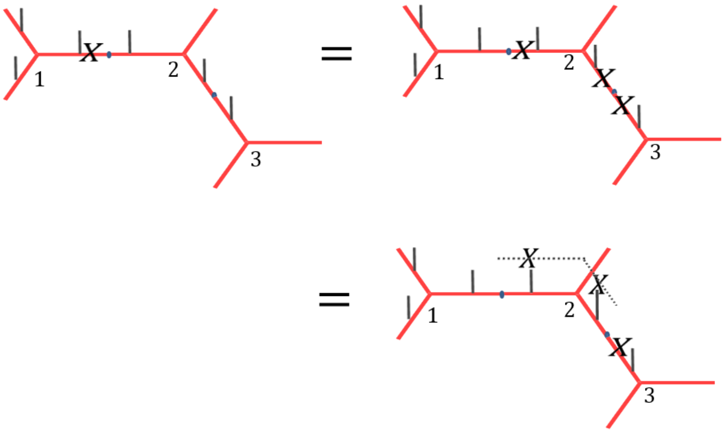
. (2.56)

And also, in any tensor network, we can always multiply two contracting virtual legs with A and A^{-1} for any invertible operator A, since they cancel each other upon contraction. So in particular the single-line TNR satisfies



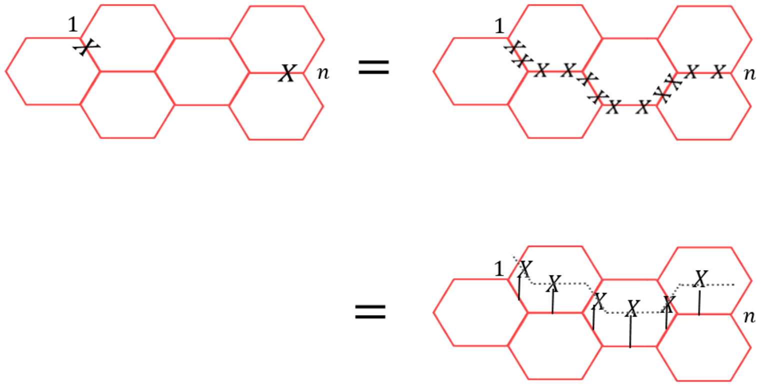
. (2.57)

Using these two relations, we see that a single X virtual variation on can be moved to another tensor on the same sublattice, and this transfer produces an $X \otimes X$ operation on the physical level,



. (2.58)

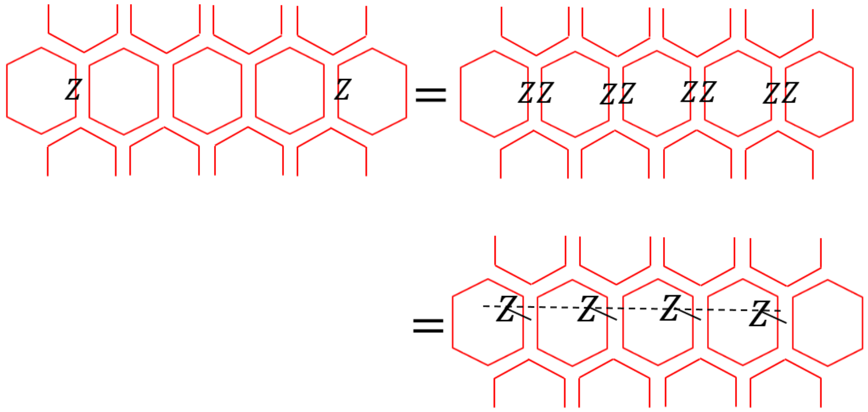
In the first equality, Eq. (2.57) is used while in the second equality Eq. 2.56 is used. We see that the X variation moved from site 1 to site 3 while leaving operator $X \otimes X$ along the path (on site 2). We can repeat this process and move X to the next tensor and so on. After X is moved from site 1 to n there will be an X-string operator applied on the physical level along the path. Finally, if there is already an X variation present at site n, the two will cancel and we will be left with an X-string operator only,



(2.59)

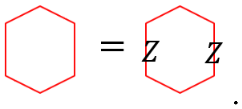
Of course the particular path between site 1 and n chosen is completely arbitrary. We can choose any path between them as we like. So we have successfully shown that though a single X variation cannot be completely lifted to the physical level, two such variations sitting far apart can be, and they are lifted to a non-local physical operator between them. It implies that a X variation on the single-line TNR cannot be removed locally on a physical level. Only two of them can be removed by applying a non-local operator between them. In fact, it is easy to recognize what this excitation is. Since X-string operators correspond to creation or annihilation of e-particles in the toric code, it is clear that the X virtual variation actually is an e particle excitation. It is topologically non-trivial, which is in line with our claim. Condensation of X variations is actually the condensation of e particles, and that is why it leads to topologically phase transition.

A similar analysis can be carried out for the Z variation in the double-line tensor. We noted that it cannot be lifted to a local physical operator. But two Z operators can be lifted to a non-local physical operator,



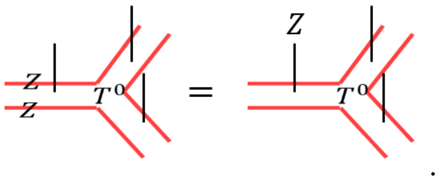
(2.60)

where in the first equality, the following relation (similar to Eq. (2.57)) has been used,



(2.61)

And in the second equality, the following property of the fixed point double-line tensor is used.



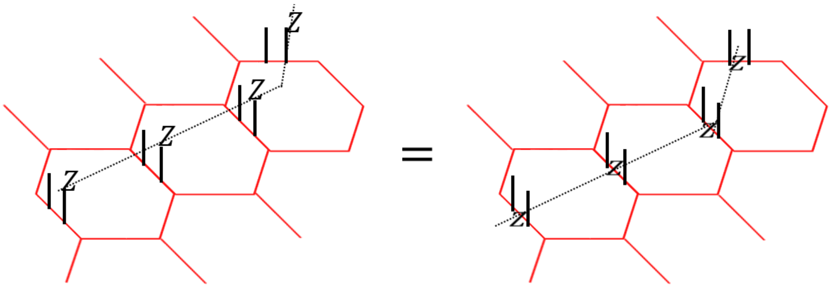
(2.62)

So we see that two Z variations on the virtual level, sitting far apart cannot be removed by local operations on the physical level. They can only be removed by a non-local operator, the Z-string operator. This suggests that the Z variation is a topologically non-trivial excitation. Indeed, it is easy to see that it is nothing but the m particle excitation, since Z-string operator creates and annihilates m-particles.

This is in line with all our claims: Z variation is in the $M_0 - \mathbb{M}$ space; it cannot be removed locally, and that it is a topological boson.

Non-trivial gauge string-operators: Zero-string operators

It may look a little puzzling that a local virtual operator in $M_0 - \mathbb{M}$ can create a non-trivial physical excitation. We analyzed what these variations correspond to by lifting them up to the physical level. To understand the phenomena better we can ask the opposite question: what happens when we 'bring down' a non-trivial quasi-particle excitation on the physical level to the virtual level? Since such an excitation is created by a physical string-operator, the equivalent question is, what happens to string-operators of the model when we bring them down to the virtual level? We can look at the specific examples considered above. For example, if we look at Eq. (2.59) in the opposite way, we see that the physical X-string operator, which creates e-particles, becomes a gauge-string operator on the virtual level, which subsequently creates a variation in the $M_0 - \mathbb{M}$ space. Contrast this with Z-string operator, that creates m-particles. This operator does not map to gauge-string operator,



. (2.63)

Similarly, Eq. (2.60) shows that that the physical Z-string operator, which creates m-particles, becomes a gauge-string operator on the virtual level. So we see that if a physical anyonic string operator maps to a gauge-string operator on the virtual level, it creates an excitation in the $M_0 - \mathbb{M}$ space. This property of the tensor network state in general is the reason why a local virtual variation can actually correspond to non-local variation on the physical level. In other words, certain gauge-string operators are non-trivial because they come from a non-trivial string operator on the physical level. We would call such physical string operators that map to gauge-string operator on the virtual level a *zero-string operator*. The reason behind this terminology will become clearer in the next chapter. So we conclude that the m-particle operator is the zero-string operator of the double-line TNR, while the e-particle string operator is the zero-string operator of the single-line TNR.

Since variations in the unstable space $M_0 - \mathbb{M}$ are created by zero-string operators, it implies that, if, in a given TNR, none of the physical string operators map to a gauge-string operator then it will have no variation in $M_0 - \mathbb{M}$. In that case, we would simply have $M_0 = \mathbb{M}$. Such a TNR will have no instabilities.

Before going to the physical explanation of instabilities, we would look at one more way of decomposing the stand-alone space in trivial and non-trivial excitations: using Wilson-loops.

2.10 Decomposing stand-alone space using Wilson-loops: MPO symmetries

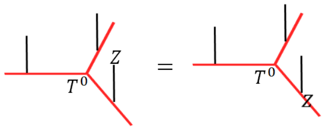
In the last two sections we argued how the stand alone space M_0 decomposes further into two subspaces, $\mathbb{M}$ and $\mathbb{M}_{\not{t}} - \mathbb{M}$ on the basis whether a stand-alone variation can be lifted to the physical level locally or non-locally. In doing so we used the fact about topological models: anyonic excitation cannot be removed by a local operation but an elementary excitation can be.

There is another way to distinguish between trivial and non-trivial excitations. Consider the tensor network state made out of T^0, except at site s_0, T^0 has been replaces by some stand-alone tensor T. Now we want to find out whether this variation/excitation is a topologically non-trivial excitation. In topological models the way to detect the presence of anyon is by measuring Wilson-loop operators around it. We will do the same here, but on the virtual level. Doing so will reveal another interpretation of the MPO subspace/symmetries: these symmetries come from Wilson-loops of anyons of the model.

Consider the following physical process. We generate an anyon a, anti-anyon $\bar{a}$ pair, move a around the site s_0 where T is sitting and finally fuse it with $\bar{a}$. Mathematically, this is equivalent to applying a Wilson loop operator $W_a(C)$ corresponding to particle a. C represents the closed curve/loop around the site. If there was another anyonic excitation b present at s_0 and if a and b have a non-trivial braiding statistics with each other, then this process produces a phase factor. Hence application of $W_a(C)$, where C is a loop around a site can be used to detect if there is a topologically non-trivial excitation present at the site. Of course $W_a(C)$ are symmetries of the ground state for all anyons a. But more than that, it would be a symmetry of any state with a trivial local excitation sitting at s_0.

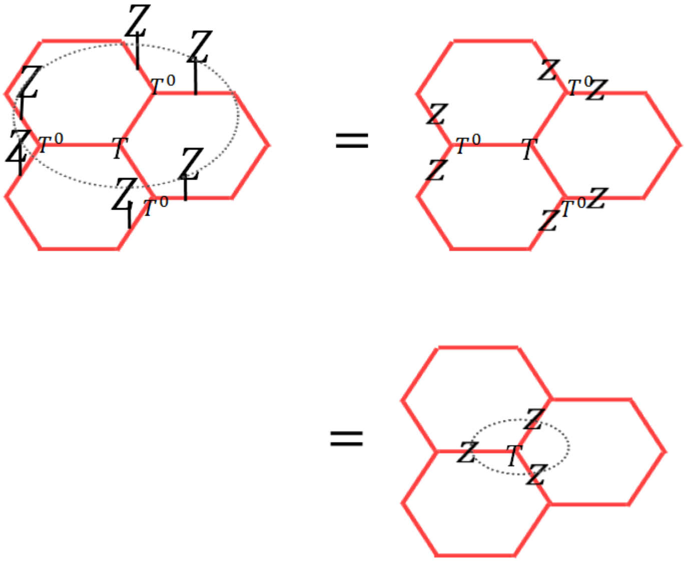
$W_a(C)$ is an operator on the physical degrees of freedom, which induces an operator, $M_a(C)$, on the virtual degrees of freedom. $W_a(C)$ is guaranteed to have a representation $M_a(C)$ on the virtual level because $W_a(C)$ is an operator supported on the ground-state physical space of local tensors, and as we noted in Lemma 1, such an operator can be mapped to an operator on the virtual level. Hence, just as $W_a(C)$ is a symmetry on the physical level, $M_a(C)$ should be a symmetry of the ground state tensor T^0 on the virtual level. But, in fact, any stand-alone variation T that is topologically trivial excitation would be symmetric under $M_a(C)$ for all a. A tensor variation that breaks this symmetry for some a would imply the presence of a non-trivial excitation. So the space of stand-alone tensors T that satisfy $M_a(C)$ symmetries for all a has to the space of topologically trivial excitations. This precisely is the source of MPO symmetries, and $\mathbb{M}$ is nothing but the projector onto the $M_a(C)$ symmetric subspace for all a. In fact this is why the MPO projector for both double-line (Eq. 2.82 and single-line (Eq. 2.33) could be written in terms of loop operators on the virtual level. These loop operators are nothing but the Wilson loop operators on the virtual level.

Let's illustrate the above discussion with the single-line TNR of toric code state. Let's say the stand-alone tensor T is surrounded by T^0. We apply an m-particle Wilson-loop around this stand-alone tensor. This Wilson-loop applies Z operators on the physical legs of the surrounding T^0 tensors. We have already seen that this operation can be brought down to the virtual level (Eq. (2.46) in the opposite direction),



(2.64)

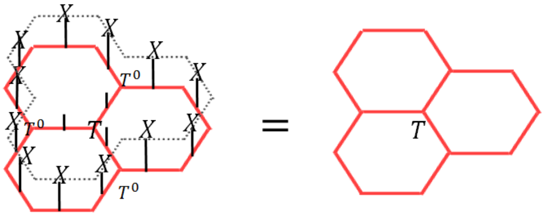
Keeping in mind that T^0 also satisfies the $Z^{\otimes 3}$ symmetry of Eq. (2.13), we see that the m-particle Wilson-loop finally reduces to a $Z^{\otimes 3}$ operator on the stand-alone tensor T. That is,



(2.65)

In first equality, we have used relation (2.64) and in second equality we have used the Z^2 symmetry of the single-line Tensor. So we find that the representation of m-particle Wilson loop, $W_m(C)$ on the stand-alone space is, $M_m(C) = Z^{\otimes 3}$. So we have shown that the presence of $Z^{\otimes 3}$ symmetry constraint inside the stand-alone space actually comes from the m-particle Wilson loop.

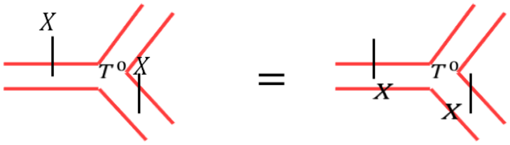
Now a natural question arises: why isn't there an analogous symmetry constraint on the tensor corresponding to an e-string operator Wilson loop? Let's apply the e-particle Wilson-loop, which is a loop of X operators on the single-line TNR, and then bring it down to the virual level. We find,



(2.66)

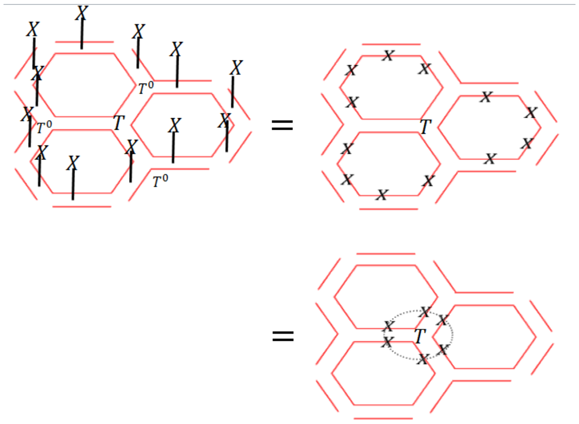
where we have used the fact that X operators on the nearby virtual legs simply cancel each other. This was already noted in the discussion of zero-string operators and in Eq. (2.59). So we see that the e-particle Wilson-loop poses no extra symmetry constraint on the stand-alone tensors.

Now let's see if the MPO symmetry of double-line TNR also comes from a Wilson-loop. Double-line case is more interesting than the single-line case because, as we have already discussed, the double-line has a stand-alone space smaller than the full virtual space. We first look at the e-particle Wilson-loop, which is a loop of X operators on the physical level. We have already seen that this operation can be brought down to the virtual level (Eq. (2.48) in the opposite direction)



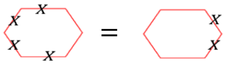
. (2.67)

So we find



, (2.68)

where in first equality we used Eq. 2.67 and in the second equality we simply used the relation

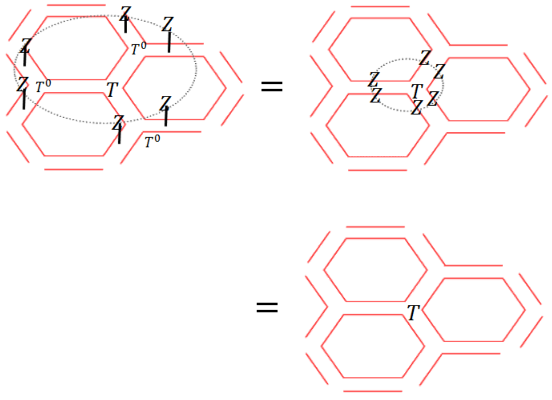


. (2.69)

So we have shown that the MPO symmetry, $X^{\otimes 6}$, of double-line TNR is actually a representation of the e-particle Wilson-loop on the stand-alone space. At this point, it is important to note that relation (2.69) holds only when T is in the stand-alone space, so it has the $Z^{\otimes 2}$ symmetry. If T was outside the stand-alone space, this would not be true. *This is why we say that MPO symmetries, $X^{\otimes 6}$ in this particular case, are representations of the Wilson-loops on the stand-alone space, not on the full virtual space.*

Now we analyze the m-particle Wilson-loop, which is a loop of Z operators. Eq. 2.62 tells us how to bring down the Z operators on double-line fixed point tensor,

T^0. Using this and other obvious properties of T^0 and T, we find



. (2.70)

The first equality follows from Eq. 2.62 and the fact that Z operators can be slid along contracted virtual legs. The last equality follows from the fact that T is a stand-alone tensor, so it satisfies the $Z^{\otimes 2}$ symmetries by definition. Or, in other words, the representation of the Wilson-loop operator on the stand-alone space is $M_m(C) = I^{\otimes 3}$. That is, it is represented trivially. So we see that all stand-alone tensors satisfy the m-particle Wilson-loop symmetry. Hence this symmetry poses no extra constraint within stand-alone space, and that is why the MPO subspace had only one Z_2 symmetry. In fact, this analysis has shown what we already knew from Eq. 2.60: m-string operator is a zero-string operator of the double-line TNR.

At this point, we can notice the similarity between double-line m-particle relation and single-line e-particle relation. But there is a crucial difference. $W_e(C)$ has trivial representation on *all* of the virtual space of single-line TNR, but $W_m(C)$ has trivial representation *only* in a subspace of the virtual space of double-line TNR.

This analysis points toward a representation theoretic way of understanding tensor instabilities. $T^0 : V \to P$ is a linear map from virtual vector space to the physical vector space. This map induces a representation of operators on the physical space in the virtual space. In particular, it induces the representation of Wilson-loop operators, $W_a(C) \to M_a(C)$. Such a representation is always possibly as is guaranteed by the MPO-injectivity (lemma 1). In fact, this representation would be faithful on individual tensors. But there is no guarantee that it would be *faithful* on the whole tensor network, because $M_a(C)$ can be a gauge-string operator, as we have already discussed for $W_e(C)$ in single-line and $W_m(C)$ in double-line. So the string-operator algebra is not faithfully represented on the virtual level. It is this unfaithful representation of anyonic algebra that causes tensor instability.

Now that we have explained the physical meaning of stand-alone and MPO subspaces, we are ready to explain the phenomena of TNR instability.

2.11 Physical reason of instability: topological boson condensation

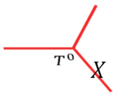
Now we put together the physical understanding of all the relevant subspaces (M_0, $\mathbb{M}$ and $M_0 - \mathbb{M}$) together to make the coherent picture of why variations in the subspace $M_0 - \mathbb{M}$ are unstable, and, in particular, explain the numerical results shown in Fig. 2.5 and Fig. 2.6. The general explanation has already been stated in form of conjecture 2 but we repeat it again informally going through all possible variations one by one.

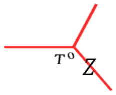
- The variations on the physical indices are of course not stable because they are topologically trivial and can be removed with local operations.
- Variations outside the stand-alone space, $I_V - M_0$ are not unstable because they cannot proliferate.
- Every variation inside M_0 is ‘bosonic’ and does proliferate and the varied wave function is a condensate of that ‘boson’. But when the variations is inside $\mathbb{M}$, it was a topologically trivial boson and hence does not cause a topological phase transition. Or, equivalently, every variation inside MPO subspace was nothing but a variation on the physical level hence stable.
- Finally, when the variation was inside stand-alone, but outside MPO subspace then it can condense and is a topologically non-trivial boson. Hence it causes a topological phase transition, resulting in a TNR instability.

Now we explain this boson condensation specifically for the single-line and double-line TNR considering specific variation.

e-particle condensation in single-line TNR

To guide the discussion, consider two illustrative variations to single-line TNR as before

$T^{(X)} =$ 

, $T^{(Z)} =$ 

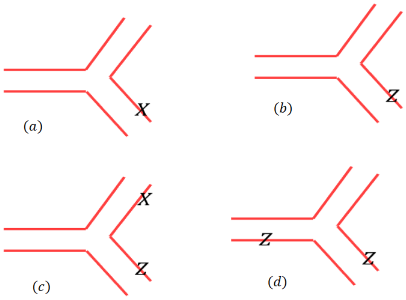
(2.71)

- The T^Z variation exemplifies variations that can condense but correspond to a local physical variation, hence are trivial/elementary excitations. Such variations result in a proliferation of elementary excitation which does not cause a topological phase transition.

- The T^X variation exemplifies variations that can condense but do not correspond to local physical variations. In fact, such variations correspond to e-particle excitation. Hence such variation results in an e-particle condensation and destroys the topological order of the tensor network state.

m-particle condensation in double-line TNR

Let's consider the different variations in double-line TNR as before



(2.72)

1. Variations in (a) and (c) exemplify variations that break the stand-alone symmetry $Z^{\otimes 2}$, hence they cannot proliferate and, therefore, are stable.
2. Variation in (b) exemplifies variations that can stand alone, so can proliferate. But they break the MPO symmetry, so correspond to a non-trivial boson. In fact it corresponds to an m-particle excitation. So this variations causes m particle condensation and results in the loss of topological order.
3. Finally, variation in (d) exemplifies variations that can proliferate. But they also are inside the MPO subspace, so correspond to trivial/elementary excitations. So their proliferation does not cause a topological phase transition.

2.12 double-semion

Double-semion model can be understood as a 'twisted' Z_2 quantum double model[51, 59]. Its Hamiltonian is almost the same as that of toric code, except for the phase factor associated to the plaquette term

$$H_0 = -\sum_v \prod_{l \in v} Z_l - \sum_p \prod_{l \in p} X_l \prod_{r \in \text{legs of } p} i^{\frac{1-Z_r}{2}}, \tag{2.73}$$

where 'legs of p' refers to the six legs attached to a plaquette. Its ground state is

$$|\psi\rangle = \sum_{X \in \text{closed}} (-1)^{n(X)} |X\rangle, \tag{2.74}$$

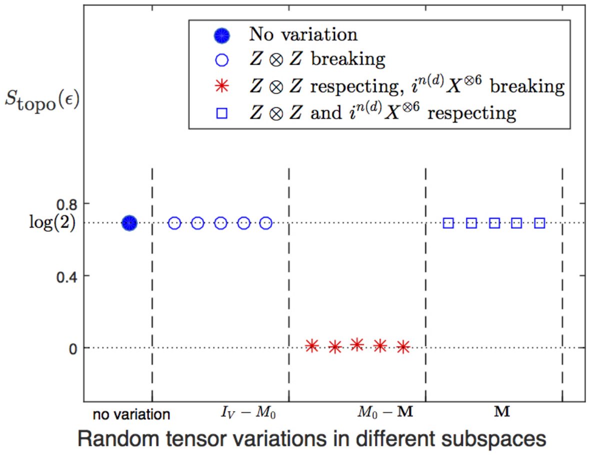


Figure 2.12: Numerical calculation of the topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of the states represented by double-semion fixed point double-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Variations in $I_V - M_0$ violate $Z \otimes Z$ symmetry. Variations in $M_0 - \mathbb{M}$ violate $i^{n(d)} X^{\otimes 6}$ but not $Z \otimes Z$. Variations in $\mathbb{M}$ violate no virtual symmetry.. The details of this numerical calculations are given in appendix A.2.

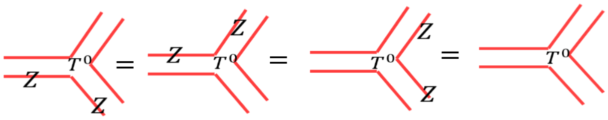
where X again refers to string configurations on the hexagonal lattice. $n(X)$ denotes the number of loops in a given string configuration. The ground state, like that of toric code, is again a superposition of all closed string configurations. But it has a phase factor of $(-1)^{n(X)}$ which is 1 for even number of loops and -1 for odd number of loops. It has 3 quasi-particle excitations: a semion, an anti-semion, and a self-boson. So, unlike the toric code, it has only one boson. There is a known double-line TNR of this state [42, 43], $(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'}$, with the same structure as that of toric code. So,

$$(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'} = S_{\alpha\beta\gamma}\delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{\gamma\gamma'}\delta_{i,\beta+\gamma}\delta_{j,\alpha+\gamma}\delta_{k,\alpha+\beta}. \tag{2.75}$$

But now the values are

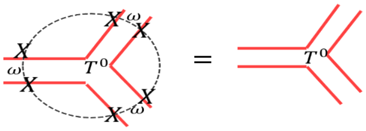
$$S_{\alpha\beta\gamma} = \begin{cases} 1 & \text{if } \alpha+\beta+\gamma = 0, 3 \\ i & \text{if } \alpha+\beta+\gamma = 1 \\ -i & \text{if } \alpha+\beta+\gamma = 2. \end{cases} \tag{2.76}$$

Clearly, it has the same $Z^{\otimes 2}$ symmetry, as the toric code double-line TNR. That is, T^0 satisfies



(2.77)

But it does not have the exact $X^{\otimes 6}$ symmetry as that of toric code double-line TNR. By looking at the tensor values, it can be seen that it has the $X^{\otimes 6}$ with an additional phase factor ω between virtual legs,



(2.78)

where $\omega = i$ if the virtual legs on the two sides of it take different values (that is, there is a domain wall) and $\omega = 1$ otherwise. So T^0 has Z_2 symmetry of the form $(i)^{n(d)}X^{\otimes 6}$ where $n(d)$ is the number of domain walls between α, β and γ. That is,

$$n(d) = \begin{cases} 0 & \text{if } \alpha + \beta + \gamma = 0, 3 \\ 2 & \text{if } \alpha + \beta + \gamma = 1, 2. \end{cases} \tag{2.79}$$

To apply our conjecture we first need to calculate the stand-alone and MPO subspaces of T^0. The double tensor is,

$$\mathbb{T} = \; = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + i^{n(d)}X^{\otimes 6}). \tag{2.80}$$

Comparing it with the toric code double tensor in Eq. (2.20), we can immediately guess that the stand-alone projector is given by

$$M_0 = \frac{1}{2}B_0 = \frac{1}{8}\left(I^{\otimes 2} + Z^{\otimes 2}\right)^{\otimes 3}, \tag{2.81}$$

which is the same as that of toric code. And the MPO projector is,

$$\begin{aligned} \mathbb{M} = \frac{1}{8}\mathbb{T}^0 &= \frac{1}{16}\sum_I (T^0)^I_\alpha (T^{0;*})^I_{\alpha'} \\ &= \frac{1}{16}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + i^n(d)X^{\otimes 6}). \end{aligned} \tag{2.82}$$

So we see that the symmetries identified in Eq. (2.77) are actually the stand-alone symmetries, and the symmetry identified in Eq. (2.78) is actually the MPO symmetry. With this information, our mathematical conjecture predicts:

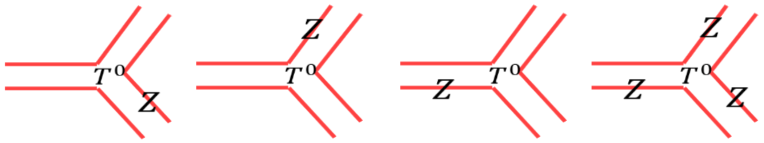
1. Variations that violate $Z \otimes Z$ symmetries are stable.
2. Variations that respect $Z \otimes Z$ but break $(i)^{n(d)} X^{\otimes 6}$ are unstable.
3. Variations that respect both $Z \otimes Z$ and $(i)^{n(d)} X^{\otimes 6}$ are stable.

We test these predictions numerically. The results are shown in Fig. 2.12. We conclude that the conjecture predicts the numerical observation correctly.

What about the physical conjecture? Is it compatible with the numerical observation? The answer is, yes. the double-semion model has one boson whose string operator is the Z-string operator, the same as that of the m-particle in toric code. Since both also have the same stand-alone space, it means bringing down this string operator to the virtual level would again give us a gauge-string operator. Hence the string-operator corresponding to the boson in the double-semion model is a zero-strong operator, which implies that the variations in the stand-alone space corresponds to this boson. So the instability we see is due to the condensation of this topological boson. Another way to see it is to notice that the MPO symmetry in Eq. 2.78 actually comes from the Wilson loop operator corresponding to semion (or anti-semion). So variations that break it actually signify the presence of the boson.

Comparing double-line TNR of toric code and double-semion

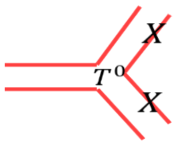
So we see that the space $(M_0 - \mathbb{M})$ for double-semion is 4-dimensional space spanned by basis



(2.83)

This looks exactly similar to the $M_0 - \mathbb{M}$ basis in double-line toric code in Eq. 2.41, which might lead one to believe that they both are unstable for similar variations. But one has to carefully note that the tensor T^0 for both models are different, so the basis shown in Eq. 2.41 and in Eq. 2.41 are also different. To illustrate this consider the following variation:

$$T^{XX} =$$



(2.84)

This variation is in stand-alone space of both toric code and double-semion, but it respects MPO symmetry of the toric code but violates the MPO symmetry of the double-semion. Indeed this variation causes phase transition in double-semion but not toric code. This variation cannot be lifted to the physical level on the double-semion tensor like it did for toric code (Eq. 2.48). But one can readily see that this variation is not spanned by the basis in Eq. 2.84. And the reason for this is that this variation has components both in the MPO subspace *and* the unstable subspace, which can be seen by applying the projector of the two spaces. We find that $(M_0 - \mathbb{M})T^{XX} \neq 0$ and $\mathbb{M}T^{XX} \neq 0$. This examples reminds us that for a variation to be unstable, all it needs is to have a non-zero component in the unstable space. So it should not be thought that only variations spanned by the unstable basis are unstable.

2.13 Implications for the simulation of phase transitions

Projected Entangled Pair States (PEPS), one type of Tensor Network States (TNS), are often used as *ansatz* for different numerical simulations of gapped lattice topological models. In particular, TNS can be used to simulate phase transitions between different topological phases[52]. The fixed point Hamiltonian is perturbed with a local Hamiltonian $H_0 \to H_0 + \eta H_{\text{local}}$ and the perturbation strength, η is increased slowly. At some finite value of η the gap closes and the system goes through a phase transition. For many perturbations, this phase transition consists of boson condensation. For example, for the toric code Hamiltonian Eq. (2.9), two kinds of perturbations can be added:

$$H_1 = -U \sum_v \prod_{l \in v} Z_l - g \sum_p \prod_{l \in p} X_l - \eta \sum_l Z_l, \tag{2.85}$$

$$H_2 = -U \sum_v \prod_{l \in v} Z_l - g \sum_p \prod_{l \in p} X_l - \eta \sum_l X_l. \tag{2.86}$$

Let's first discuss the first kind of perturbation. In the first Hamiltonian, we keep $U = \infty$ and study the ground state as the relative values of η and g change. At $\eta = 0$ the ground state is simply the fixed point toric code state given in Eq. (2.10). That is, it is an equal weight superposition of all closed string configuration. At $g = 0$, the state is the vacuum state, that is, all spins are 0. These two states are topologically different, and hence there must be a phase transition as we change η/g from 0 to ∞. This phase transition can be understood as a condensation of m particles. Recall that $\langle \Psi | B_p | \Psi \rangle = 1$ corresponds to no m particle and $\langle \Psi | B_p | \Psi \rangle = -1$ corresponds to an m particle excitation at a plaquette p, where $B_p = \prod_{l \in p} X_l$ is the plaquette term of the toric code Hamiltonian. For $\eta = 0$ ground state we have $\left\langle \Psi \middle| B_p \middle| \Psi \right\rangle = 1, \forall p$, while for $g = 0$ ground state we have $\left\langle \Psi \middle| B_p \middle| \Psi \right\rangle = 0, \forall p$. It indicates that as η/g is increased, m particles proliferate and at phase transition point, the system goes through a boson (m particle) condensation and the ground state becomes a trivial state. Boson condensation phase transitions are known to be second order phase transitions. That is, ground state energy and its first order derivative as a function of η/g are smooth functions, but its second order derivative is discontinuous at the phase transition point.

It was shown by Gu *et al.* [52] that an attempt to numerically simulate this phase transition point with single-line tensor network state ansatz gives a transition that is wrong both quantitatively and qualitatively. It gives a wrong critical point value of η/g, and it gives a first order phase transition, not a second order one. But with double-line tensor network state ansatz, it gives the correct second order phase transition with correct critical point.

This difference can be easily understood in light of our discussion on single-line and double-line TNR of toric code state. As we showed, double-line TNR is capable of condensing m particles while single-line TNR is not. That is why double-line TNR is suitable for simulating a phase transition that involves m particle condensation.

A similar analysis can be done for the second type of perturbation. We set $g = \infty$ and change relative value of U and η. For $\eta = 0$ the ground state is the toric code ground state in Eq. (2.10), and for $U = 0$ the state is trivial state with all qubits aligned in +x direction. Here the phase transition involves e particle condensation which is again a second order phase transition. Hence, to simulate this phase transition, one should use single-line TNS ansatz and not the double-line TNS ansatz.

This is one of the important points of understanding the unstable direction of variations that a particular TNR possesses. To simulate a boson condensation phase transition, one should choose the TNR that is capable of condensing that particular boson of the model.

Of course, there is also a flip side to this. If one is interested in determining the topological order of a particular TNR by calculating the topological entanglement entropy, one should make sure to keep out of the unstable space, $M_0 - \mathbb{M}$, for numerical stability. A small numerical variation in this space will change the state globally and result in wrong results. For example, in calculations involving Tensor Entanglement Renormalization Group (TERG) [52] and Tensor Network

Renormalization (TNR [1]) [60] steps, we should project the resulting tensor after every RG step back to the stable space, $(I_V - (M_0 - \mathbb{M}))$, or naturally occurring numerical errors might gain a component in $M_0 - \mathbb{M}$ space and change the topological order of the state radically.

Now we will apply what we learned from the toric code example to analyze the TNR of another closely related model, the double-semion model.

2.14 Summary and outlook

In this chapter, we tried to answer the following question: which variations cause instabilities in a TNR of a 2D topological state and why? That is, if we start from the tensor network representation of a topological state and add arbitrarily small variations to the local tensor, does the topological order of the represented state always remain the same? We answered this question by showing that instabilities are caused by a variation if it has a component in a certain subspace of the virtual space because they cause boson condensation. We also gave a precise mathematical procedure to form a projector onto the stable subspace which can be used in numerical works to always protect against instabilities. This is an important result because if the tensor is not actively protected against these variations then the task of determining topological order of a tensor network state is numerically 'ill-posed'. That is, arbitrarily small numerical error in the process may change our conclusion in a qualitative way. Previous work[47] has shown that this is indeed the case for the single line representation of the toric code state.

We demonstrated the case explicitly for the tensor network representation of the toric code (single-line, double-line), and the double-semion by calculating the topological entanglement entropy S_{topo} of tensors with random variations. We observed that in all cases stand-alone variations that break MPO symmetry lower S_{topo} (to zero) and in all cases instability was associated to condensation of a boson. Moreover, we point out that to correctly simulate the local properties of a phase transition induced by such boson condensation, these MPO breaking variations must be *allowed* in the variational calculation; otherwise, one may reach the wrong conclusion about the phase transition (e.g. regarding the order of the transition). This has been observed in the case of toric code in Ref.[52].

We found that the mathematical reason for this instability is that local variations on the virtual space may correspond to non-local variations on the physical level. Or,

[1] not to be confused with TNR that we use for referring to tensor network representation.

stated differently, non-local operation on the physical level may correspond to local operations on the virtual level. This results in an unfaithful representation of the anyonic algerba on the virtual level which in turn can make the TNR an ‘ill-defined’ representation of the topological state.

Given this result, we can ask, how to properly design the tensor network algorithm so that it can correctly simulate topological phases and phase transitions? In particular, if we want to determine whether the ground state of some Hamiltonian has topological order by calculating topological entanglement entropy in the thermodynamic limit, we need to use a variational ansatz with the proper MPO symmetry. How to do that in an efficient and unbiased way is an interesting open question.

On the other hand, if we want to properly simulate a topological phase transition induced by boson condensation, we need to put in the proper variational parameter. However, as we have seen in the case of the toric code, different representations (single line, double line, triple line) contain parameters corresponding to the condensation of different bosons (e or m). In fact, none of the representations contain parameters which correspond to the condensation of both bosons. Therefore, it is not possible to use any of them to correctly obtain the full phase diagram. It implies that, if we want to study a topological phase transition whose nature is unknown, we need to try different ansätze. How to do that in an efficient and unbiased way is again an interesting open problem. We leave these problems to future study.

Chapter 3

TN REPRESENTATION OF GENERAL STRING NET MODELS AND ITS INSTABILITIES

3.1 Background and motivation

In the last chapter we discussed the instabilities in Tensor Network Representations (TNR) of toric code and double-semion models. We made a conjecture about which tensor variations cause instabilities and what is the physical reason behind it. We defined two important subspaces of the virtual space of the TNR: stand-alone and MPO subspaces, and showed how variations that are in the stand-alone subspace but are outside the MPO subspace are responsible for loss of topological phase since they represent the condensation of a topologically non-trivial boson. We also discussed how this instability is related to the fact that certain non-trivial string operators have a trivial representation on the virtual space.

But do these results hold for other topological models as well? In particular, since both toric code and double-semion were abelian models, we can ask whether these results would hold for non-abelian models. Non-abelian models often have more complicated string-operators and it might be the case that none of the anyons have a corresponding string-operator that disappears on the TNR. In fact, as we will show below, for the double-Fibonacci model, there indeed are no irreudicble string-operators that disappear on the TNR. Would the TNR still be unstable?

To answer these questions, we have to investigate a general class of topological models. Toric code is a Quantum Double model [12] while double-semion is a Twisted Quantum Double model [61]. Both, quantum double and twisted quantum double, are subclasses string-net models defined by Levin and Wen [13]. In fact string-nets are the the most general set of lattice topological models that possibly includes all non-chiral topological states in 2D[62]. Also, a general TNR for all string-net ground-states is known[15, 42]. For these reasons, we choose to work with general string-net models to investigate the question of TNR instabilities.

The rest of the chapter is organized as follows. In section 3.2 we give a brief review of the string-net models and required algebraic relations that would be used in the rest of the chapter. In section 3.3 we will give a brief derivation of the TNR of the string-net states. We will then calculate the stand-alone space of this TNR in section

3.4, followed by the calculation of their MPO subspace in section 3.5 both by directly calculating it from the given TNR and by considering the Wilson-loop operators. We would discuss the zero-string operator of string-net TNR in section 3.6 and then discuss the two classes of unstable variations in section 3.7. Before giving the final proof that this TNR has instabilities for all string-net modles in section 3.10, we will first re-examin instabilities in TNR of toric code and double-semion in the string-net paradigm in section 3.8 and discuss the non-abelian example of double-Fibonacci model in detail in section 3.9 and show that it too has instabilities. We conclude the chapter with a summary and outlook.

3.2 A brief review of string-net models

String-net models, which are Hamiltonian realizations of Turaev-Viro TQFTs, are introduced by Levin and Wen [13] as RG fixed point models that describe topological order in $2+1$ spacetime dimensions. Following are the defining data of the string-net states:

1- **Local Hilbert space:** String-nets are lattice spin models. Spins sit on the links of hexagonal lattice. Each spin s can be in $N+1$ state, $s = 0, 1, 2, \ldots, N$. $s = j$ at a link can be understood as a string of 'type j' present on the link. Strings are oriented and $i^\star$ denotes string type i with the opposite orientation. If $i = i^\star$ the strings are called 'unoriented'. We have assumed the strings to be unoriented in the present paper for the sake simplicity, though our results can easily be generalized to the oriented case.

2- **Branching rules:** There are branching rules denoted by δ_{ijk}. $\delta_{ijk} = 1$ if string type i, j, k are allowed to meet at a point, and $\delta_{ijk} = 0$ otherwise.

3- **Quantum dimensions:** For every string type s, there is a value d_s associated to it, called its quantum dimensions. $D = \sum_s d_s^2$ is called the 'total quantum dimension'.

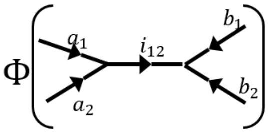
4- **String-net condensed state:** If we assign a particular string to each link, it forms a string-net configuration on the lattice. A string-net condensed quantum state is a superposition of these different string-net configurations on the lattice. Let's denote the string-net configurations with X. So a string-net condensed state is,

$$|\Psi\rangle = \sum_X \Phi_X |X\rangle, \tag{3.1}$$

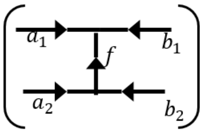
where Φ_X is the amplitude with which a configuration X appears in the description of the state. In general, Φ_X can be complicated and states belonging to the same topological phase might have different wave functions. However, if we perform an RG process, then all states in the same phase would end at the same fixed point state, which is to say that they should look the same at large distances. Φ_X can be described for this fixed point state. Though their absolute values are again complicated, we can give their relative values by describing local constraints on how amplitude Φ_X changes as we deform a configuration X locally. These deformations involve rebranching, removing bubbles, fusing two strings together, etc. These constraint equations are given in equation (4)-(7) of Levin and Wen [13]. The most significant of these local constraint is the so called 'F'-move.

5- **F-symbols:** A local constraint involving rebranching of 5 strings is the following:

$$\Phi$$



$$= \sum_f F^{a_1 a_2 i_{12}}_{b_2 b_1 f}\,\Phi$$



(3.2)

F-symbol is a six indexed object and it satisfies the following properties:

$$F^{ijk}_{j^\star i^\star 0} = \frac{\sqrt{d_k}}{\sqrt{d_i}\sqrt{d_j}}\delta_{ijk}, \tag{3.3}$$

$$F^{ijm}_{kln} = F^{lkm^\star}_{jin} = F^{jim}_{lkn^\star} = F^{imj}_{k^\star nl}\frac{\sqrt{d_m d_n}}{\sqrt{d_j d_l}}. \tag{3.4}$$

Properties of the F-symbol under index permutations can be best captured by defining a new object called G-symbol by $G^{ijk}_{klm} = \frac{F^{ijk}_{klm}}{\sqrt{d_k d_m}}$. G-symbol can be considered as a value associated to a tetrahedron and the six indices sit on the six edges of tetrahedron. Then it is invariant under all tetrahedron symmetries. It satisfies an important equation, the so-called 'Pentagon Identity':

$$\sum_f d_f G^{b_1 b_2 i_{12}}_{a_2 a_1 f} G^{b_2 b_3 i_{23}}_{a_3 a_2 f} G^{b_3 b_1 i_{31}}_{a_1 a_3 f} = G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{3.5}$$

Finally we describe the exactly solvable Hamiltonian such that the RG fixed point state defined as above is one of the ground states,

$$H = -\sum_v A_v - \sum_p B_p, \tag{3.6}$$

where v and p denote the vertices and plaquette of the lattice. The vertex term is

$$A_v = \sum_{i,j,k} \delta_{ijk} \left|ijk\right\rangle \left\langle ijk\right|. \tag{3.7}$$

So, the vertex term simply projects configurations to only the ones that contain the allowed branchings. The plaquette term is more involved:

$$B_p = \sum_s \frac{d_s}{D} B_p^s, \tag{3.8}$$

where B_p^s is an operator that creates an s-type string that fuses with the strings on the plaquette. Two strings can be fused together by assuming a 0-string between them and then using F-moves.

Finally putting all of it together, we see that the data $(N, d_i, \delta_{ijk}, F^{ijk}_{klm})$ describes a string-net model.

Algebraic Identities

Here we enlist multiple algebraic relations regarding string-net data that are used throughout the paper. For rotational convenience, cyclic products will be simply denoted by $\prod_{j=1}^n$ with a cyclic $j = n + 1 = 1$. One of the most important identities is the 'Pentagon Identity',

$$\sum_f d_f \prod_{j=1}^{3} (G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}) = G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{I.1}$$

G symbols also satisfy an 'orthogonality identity',

$$\sum_{i_{12}} G^{b_1 b_2 i_{12}}_{a_2 a_1 f} G^{b_1 b_2 i_{12}}_{a_2 a_1 f'} d_{i_{12}} = \frac{1}{d_f} \delta_{f,f'} \delta_{a_1 a_2 f} \delta_{b_1 b_2 f}. \tag{3.9}$$

G-symbols are normalized as

$$G^{b_1 b_2 i_{12}}_{a_2 a_1 0} = \delta_{a_1, b_1} \delta_{a_2, b_2} \delta_{a_1 b_1 i_{12}} (d_{a_1} d_{a_2})^{-\frac{1}{2}}. \tag{3.10}$$

Cyclic products of G symbols satisfy the following equation:

$$\sum_{\{b_j\}} \prod_{j=1}^{n} (G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} G^{c_j c_{j+1} i_{j,j+1}}_{b_{j+1} b_j f} d_{b_j}) = \sum_s \delta_{f f' s} \prod_{j=1}^{n} (G^{c_j c_{j+1} i_{j,j+1}}_{a_{j+1} a_j s}). \tag{3.11}$$

Plaquette operators B_p^f correspondingly satisfy

$$B_p^f B_p^{f'} = \sum_s \delta_{f f' s} B_p^s. \tag{3.12}$$

We know that if we contract an f-type loop we get a factor of d_f. Combining this with the last two equation, we find that quantum dimensions satisfy the same identity:

$$d_f d_{f'} = \sum_s \delta_{f f' s} d_s, \tag{3.13}$$

where d_f are nothing but the eigenvalues of the plaquette operators B^f operators where the eigenstate is the string-net ground state.

Define matrix N^k as $N^k_{a,b} = \delta_{k,a,b}$. Since N^k matrices are real symmetric matrices, and commute with each other for different values of k, they share a complete set of orthogonal eigenvectors.We write the qth such simultaneous eigenvector of N^k, $\forall k$ as

$$\left|s_q\right\rangle = \sum_a s_{q;a} \left|a\right\rangle . \tag{3.14}$$

Since quantum dimensions form one such eigenvector, we fix $s_{0;a} = d_a$. The following equations follow

$$\begin{aligned} \left\langle s_q \middle| s_{q'} \right\rangle &\propto \delta_{q,q'}, && (3.15) \\ \left\langle s_q \middle| N^k \middle| s_{q'} \right\rangle &= \sum_{a,b} s_{q;a} \delta_{k,a,b} s_{q';b} \propto \delta_{q,q'}. && (3.16) \end{aligned}$$

The branching tensor δ_{ijk} is part of a fusion category data. Under the additional assumptions of braiding defined on the fusion category and braiding being sufficiently non-trivial (modularity), the s above are just the columns of S matrix. But we don't really need this for our results.

3.3 Triple-line TNR of string-net states

We now briefly describe the derivation of triple-line TNR along the lines described in the original paper by Gu *et al.* [42]. It is important to understand this derivation as it gives us a way to apply string-operators on triple-line TNR.

String net RG fixed point ground state can be constructed by applying plaquette operator $B_p = \sum_s d_s B_p^s$ to the vacuum state $|0\rangle$. B_p^s creates an s-type string loop on the plaquette p.

$$\begin{aligned} \left|\Psi_{\text{gs}}\right\rangle &= \prod_p B_p \left|0\right\rangle = \prod_p \sum_s d_s B_p^s \left|0\right\rangle \\ &= \sum_{s_1,s_2,..} d_{s_1} d_{s_2} .. \left|s_1, s_2, ...\right\rangle , \end{aligned} \tag{3.17}$$

where

$$\left|s_1, s_2, ...\right\rangle = B_{p_1}^{s_1} B_{p_2}^{s_2} \ldots \left|0\right\rangle . \tag{3.18}$$

$|s_1, s_2, ...\rangle$ is a string configuration on the 'fattened lattice'. We will refer to $d_{s_1} d_{s_2} .. |s_1, s_2, ...\rangle$ as the 'loop state'. We need to fuse these loops together to get the physical state. We then fuse these strings together to get the final physical state,

$$|s_1, s_2, ...\rangle = \sum_{i_{12}, i_{23}...} \Phi^{i_{12}i_{23}..}_{s_1 s_2 s_3..} \, |i_{12}, i_{23}, ..\rangle \, . \tag{3.19}$$

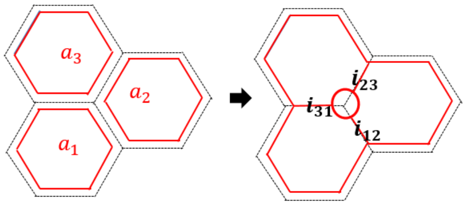
There are essentially 3 steps leading up to the expression of the triple-line TNR. We mention them here explicitly as we will need to refer back to them for other calculations.

Step 1: We start with the ‘loop state’ on the fattened lattice. jth plaquette has a loop in state s_j. The ground state is

$$|\Psi_0\rangle = \sum_{s_1, s_2, ..} d_{s_1} d_{s_2} .. \, |s_1, s_2, ..\rangle \, . \tag{3.20}$$

So every plaquette contributes a factor of d_{s_j}. We distribute it uniformly among the 6 vertices, so each vertex gets a factor of $d_{s_j}^{1/6}$ from each vertex.

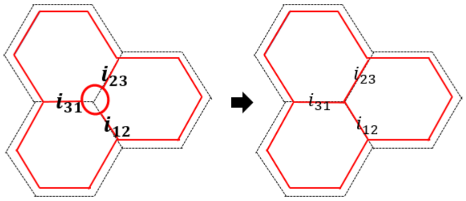
Step 2: We fuse all loops with nearby loops producing a string on the links:



(3.21)

We assume a 0-string between them and perform an F-move. It produces a factor of $\sum_{i_{j,k}} \sqrt{\frac{d_{i_{j,k}}}{d_{s_j} d_{s_k}}}$ on each link between plaquette j and k. A link is shared between two vertices, so each vertex gets a factor of $\sqrt[4]{\frac{d_{i_{j,k}}}{d_{s_j} d_{s_k}}}$.

Step 3: After the previous step, we are left with a ‘bubble’ on the vertex. Now we remove it,



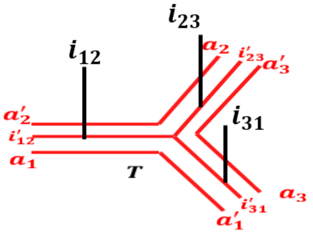
. (3.22)

Removing it produces a factor of$\sqrt{d_{s_j} d_{s_k} d_{s_l}} G^{i_{kl} i_{lj} i_{jk}}_{s_j s_k s_l}$.

Putting the 3 steps together, we get

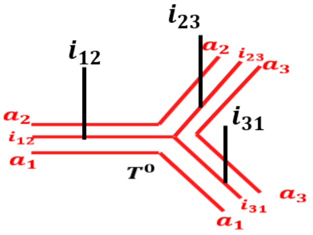
$$(T^0)^{i_{jk}i_{kl}i_{lj}}_{s_l s_j s_k} = \sqrt[4]{d_{i_{jk}} d_{i_{kl}} d_{i_{lj}}}\, G^{i_{jk}i_{kl}i_{lj}}_{s_j s_k s_l} \sqrt[6]{d_{s_j} d_{s_k} d_{s_l}}. \tag{3.23}$$

A general triple-line Tensor is represented diagrammatically as:



$$. \tag{3.24}$$

For the specific RG fixed point tensor in (3.23) we have $a'_j = a_j, i'_{j,j+1} = i_{j,j+1}, \forall j$. So it would be represented diagrammatically as:



$$. \tag{3.25}$$

Before we discuss the properties of the triple-line TNR of the general string-net models, we would like to mention that double-line TNR and single-line TNR are actually reduced versions of the triple-line TNR, and as such, many results about the triple-line TNR apply to double-line and single-line as well. We can discard some of the legs of the triple-line tensor if fewer legs are required to encode the necessary information. For example, for abelian models, the middle leg of the triple-line tensor is redundant; it always assumes value which is a product (fusion) of the two legs on either side of it. That's why for abelian models, double-line tensors suffice and the middle-leg can be discarded. Non-abelian models, such as the double-Fibonacci model we will study in section 3.9, the middle-leg does carry essential information and cannot discarded. So one cannot have a double-line TNR of non-abelian models. Furthermore, if the ground state of a model can be written as an *equal* superposition of states allowed by branching rules then the ground state admits a single-line TNR. For example, toric code ground state is an equal superposition of all closes string configurations, and hence admits a single-line TNR. In fact, *any* quantum double model with an abelian gauge group can have a single-line TNR. The double-semion model, on the other hand, is not an equal superposition of states allowed by the branching rules (it has a phase factor $i^{n(d)}$), and hence it cannot admit a single-line TNR.

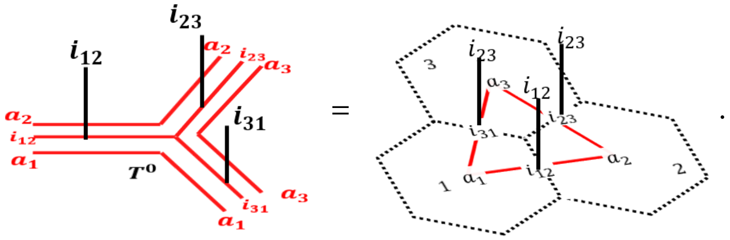
To apply the conjecture to the triple-line TNR of general string-net model, we now first calculate its MPO and stand-alone subspaces. We will do so in the next two sections.

3.4 Stand-alone subspace of string-net triple-line TNR

To calculate the stand-alone space, we need to know how to contract double-tensors on a large region. First we need to define the concept of *boundary operators* that show up in double-tensor contraction.

Boundary operators

It is more convenient to work with the dual lattice of honeycomb lattice. The dual lattice of honeycomb lattice is the triangular lattice. We label the vertices with an integer $j = 1, 2, \ldots$ The edges are labeled by the two vertices on its ends, (j_1, j_2). The triple line tensor is represented as a triangle,



(3.26)

The inner indices $a_1, a_2, ..$ sit on the vertices of the triangles, and the physical legs and the middle legs on the edges. We denote the inner index sitting on vertex j as a_j, and the physical and middle legs sitting on the edge are denoted as i_{j_1,j_2}. With this construction, the tensor component can be written as

$$(T^0)^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} = \prod_{j=1}^{3}\left(d^{\frac{1}{4}}_{i_{j,j+1}} d^{\frac{1}{6}}_{a_j}\right) G^{i_{23}i_{31}i_{12}}_{a_1a_2a_3}. \tag{3.27}$$

A double tensor of a tensor T is defined as $\mathbb{T} = \sum_I T^I (T^*)^I$ and is denoted by $\mathbb{T}$. I denotes the set of physical indices. So we get the double tensor of a tensor by contracting the physical indices between T and its complex conjugate, $T^\dagger$. Since the tensor T is represented by a triangle, the double tensor $\mathbb{T}$ can be represented by a double layer triangle.

The edge labels are the same bottom to top, only the labels of the vertices change. We label the upper vertices as $b_1, b_2, \ldots$ With this a double tensor can be written as

$$\mathbb{T}^0 \;=\; \prod_{j=1}^{n} \left(d_{i_{j,j+1}}^{\frac{1}{2}} (d_{a_j} d_{b_j})^{\frac{1}{6}} \right) G^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} G^{i_{23}i_{31}i_{12}}_{b_1b_2b_3}. \tag{3.28}$$

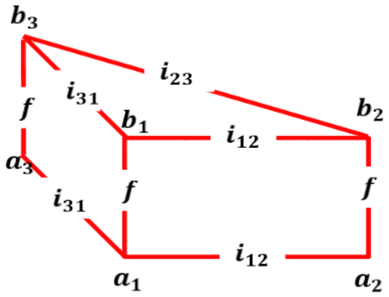
Using the pentagon equation $G^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} G^{i_{23}i_{31}i_{12}}_{b_1b_2b_3} = \sum_f d_f \prod_{j=1}^{3} (G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f})$ we get

$$\mathbb{T}^0 \;=\; \sum_f d_f B_f \tag{3.29}$$

$$B_f \;=\; \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{2}} (d_{a_j} d_{b_j})^{\frac{1}{6}}, G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \right). \tag{3.30}$$

B_f can be represented as the boundary of double-layer triangle,

$$\mathbb{T}^0 \;=\; \sum_f d_f \tag{3.31}$$



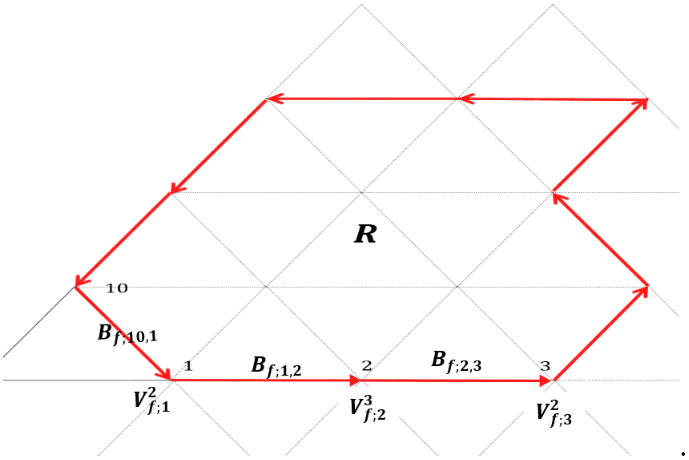
It is useful to decompose B_f into terms that sit on the edge of the triangle and terms that sit on the vertices,

$$B_f \;=\; \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \right) \prod_{j=1}^{3} \left((d_{a_j} d_{b_j})^{\frac{1}{6}} \right). \tag{3.32}$$

The first cyclic product on the RHS sits on the edges while the second term sits on the vertices. So we see that the double tensor on a triangle is (we will denote triangle as Δ)

$$\mathbb{T}^0(\Delta) \;=\; \sum_f d_f B_f(\partial\Delta). \tag{3.33}$$

The tensor resulting from contracting tensors $\mathbb{T}$ on a region R will be denoted as $\mathbb{T}(R)$. We call B_f, *the f-type boundary operator*. It lives on the boundary ∂R of a region R,



$$. \tag{3.34}$$

See Fig. 3.34. Let's say the vertices on the boundary of a region R on the triangluar lattice are labeled as $j = 1, 2, .., n..$ We associate with each vertex a factor of $(a_j b_j)^{\frac{m_j}{6}}$. m_j denotes the number of the triangles inside R meeting at vertex j. It can simply be written as $m_j = \theta_j/(2\pi/6)$, where θ_j is the angle the loop makes on vertex j. Finally, on every edge $(j, j+1)$ we associate an operator $d^{\frac{1}{2}}_{i_{j,j+1}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}$. With this construction, $B_f(\partial R)$ can be written as,

$$B_f(\partial R) = \prod_{j=1}^{n} \left(d^{\frac{1}{2}}_{i_{j,j+1}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \right) \prod_{j=1}^{n} \left((d_{a_j} d_{b_j})^{\frac{m_j}{6}} \right). \tag{3.35}$$

Now we are ready to contract tensors on individual triangles with each other in order to find the double tensor on a region R.

Double-tensor/Virtual density matrix on a general region R

We present the result in a lemma.

Lemma 2. *We find that the double tensor $\mathbb{T}^0(R)$ satisfies the general version of Eq.* (3.33)*:*

$$\mathbb{T}^0(R) = D^V \sum_f d_f^{\chi_R} B_f(\partial R), \tag{3.36}$$

where $\chi_R = V - E + F$ is the Euler characteristic of region R. V,E, and F are the number of vertices, edges, and faces that are completely inside the region R (that is, they are inside the region where tensors have been contracted).

Proof. There is a simple proof of this result. We have to contract $\mathbb{T}^0$ on each triangle with each other on the common edges and vertices to get $\mathbb{T}^0(R)$,

$$\begin{aligned}\mathbb{T}^0(R) &= \mathrm{tTr}\left(\mathbb{T}^0(\Delta_1)\mathbb{T}^0(\Delta_2)\ldots\right)\\ &= \sum_{f_1,f_2,\ldots}(d_{f_1}d_{f_2}\ldots)B_{f_1}(\partial\Delta_1)B_{f_2}(\partial\Delta_2)\ldots. \end{aligned}\tag{3.37}$$

where, as defined before, tTr denotes the operation of contracting a set of tensors along shared indices. So we basically have to see how B_{f_1} contracts with B_{f_2}. They can be contracted in two steps. First we contract all the edges, and then we contract all the vertices, and we will be left with terms sitting only on the boundary of the region. Using the orthogonality identity, Eq. (3.9), edge contraction on the edge $(j, j+1)$ between B_f and $B_{f'}$ gives

$$\begin{aligned}Ev(B_fB_{f'}) &\propto \sum_{i_{j,j+1}} d_{i_{j,j+1}}^{\frac{1}{2}} G^{b_jb_{j+1}i_{j,j+1}}_{a_{j+1}a_jf} d_{i_{j,j+1}}^{\frac{1}{2}} G^{b_jb_{j+1}i_{j,j+1}}_{a_{j+1}a_jf'}\\ &= \frac{1}{d_f}\delta_{f,f'}\delta_{a_ja_{j+1}f}\delta_{b_jb_{j+1}f}.\end{aligned}\tag{3.38}$$

The factor $\delta_{f,f'}$ implies that B_f only contracts with B_f. So the expression in Eq. (3.37) is only non-zero for $f_1 = f_2 = ...f$. So we have

$$\mathbb{T}^0(R) = \sum_f d_f^F B_f(\partial\Delta_1)B_f(\partial\Delta_2)\ldots,\tag{3.39}$$

where F is the number of faces in region R. Then there are factors of $\delta_{a_ja_{j+1}f}\delta_{b_jb_{j+1}f}$ in Eq. (3.38) that will be used in the second step of vertex contraction. Finally note a factor of d_f^{-1} that comes out of every edge contraction. So when we are done with all the edges, we will have an overall factor of d_f^{-E}, where E is the number of edges.

Now we do tensor contraction on each vertex. Note that each of the six triangles around a vertex j contribute a factor of $(d_{a_j}d_{b_j})^{\frac{1}{6}}$, so we have a total factor $d_{a_j}d_{b_j}$ on each vertex. We multiply this with the factor $\delta_{a_jb_jf}$ that came out of edge contraction. So, finally we have the vertex contraction using identity (3.13),

$$\sum_{a_j,b_j} d_{a_j}d_{b_j}\delta_{a_jb_jf} = \sum_{a_j} d_fd_{a_j}d_{a_j} = Dd_f.\tag{3.40}$$

So we see that contraction of 6 tensors on each vertex simply produces a factor of Dd_f for every f-type boundary operator. When we are done with all the vertex contractions, we will have an overall $(Dd_f)^V = D^Vd_f^V$ factor. Putting all the factors together, we get

$$\begin{aligned}\mathbb{T}^0(R) &= \sum_f d_f^F d_f^{-E} (Dd_f)^V B_f(\partial R) \\ &= D^V \sum_f d_f^{\chi_R} B_f. \end{aligned} \tag{3.41}$$

This completes the proof.

To calculate the stand-alone space, we need to know how these boundary operators behave on a large region. We present the result of this calculation in the following important lemma: □

Lemma 3.

$$\lim_{|\partial R|\to\infty} \frac{Tr(B_{f\neq 0}(\partial R))}{Tr(B_0(\partial R))} = 0 \tag{3.42}$$

Proof. To prove this, we would calculate S_{topo} on a sphere using the virtual-density method laid out in 2.2 in the previous chapter, and compare it to the known result, $S_{\text{topo}} = \log D$. We divide the sphere in symmetric two halves, let's say R and L, and calculate $\mathbb{T}(R)$. We assume the state has the appropriate symmetry such that $\sigma_L^T = \sigma_R = \sigma_b = \mathbb{T}(R)$. Using the result by [53], we know that the physical density matrix ρ_R has the same spectrum as σ_b^2, that is, $\rho_R \propto \sigma_b^2$. Let's say $\rho_R = N\sigma_b^2$, where N is the normalization factor. We first calculate N. To do that, we first need to calculate the algebra and the trace of B_f.

Let's put the string-net tensor network state on a sphere. Consider the left hemisphere, denoted as L, and right hemisphere, denoted as R. Let's denote the indices of the vertices on the boundary ∂R as $j = 1, 2, \ldots, n$. Then B_f on this boundary is given by,

$$B_f(\partial R) = \prod_{j=1}^{n} \left(d_{i_{j,j+1}}^{\frac{1}{2}} (d_{a_j} d_{b_j})^{\frac{m_j}{6}} G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}} \right). \tag{3.43}$$

Since R divides the region in to exact two halves, we assume that the boundary ∂R divides the boundary plaquette in to exact two halves, setting $m_j = 3$, $\forall j$. So we get,

$$B_f(\partial R) = \prod_{j=1}^{n} \left(d_{i_{j,j+1}}^{\frac{1}{2}} (d_{a_j} d_{b_j})^{\frac{1}{2}} G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}} \right). \tag{3.44}$$

Note that, using relation (3.10)

$$\begin{aligned} B_0(\partial R) &= \prod_{j=1}^{n}\left(d_{i_{j,j+1}}^{\frac{1}{2}}(d_{a_j}d_{b_j})^{\frac{1}{2}}G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j 0}\right) && (3.45)\\ &= \prod_{j=1}^{n}(d_{i_{j,j+1}}^{\frac{1}{2}})\delta_{a_j b_{j+1} i_{j,j+1}}. && (3.46)\end{aligned}$$

Now, using identity (3.11) the algebra of B_f operators is,

$$\begin{aligned} B_f B_{f'} &= \sum_s \delta_{ff's} B_s \times \prod_{j=1}^{n}(d_{i_{j,j+1}}^{\frac{1}{2}}) && (3.47)\\ &= \sum_s \delta_{ff's} B_s B_0. && (3.48)\end{aligned}$$

We also know how to contract B_f with each other through the calculations done previously in the privious subsection. We learned that B_f only contracts with itself, and it gives a factor of d_f^{-1} for every edge and a factor of Dd_f for every vertex. On a loop the number of vertices is equal to number of edges. So we get,

$$\mathrm{Tr}(B_f B_{f'}) = \delta_{f,f'} D^n. \qquad (3.49)$$

If calculate $\mathrm{Tr}(B_f)$, we find

$$\begin{aligned} \mathrm{Tr}(B_f) &= \sum_{\{a_j i_{j,j+1}\}} \prod_{j=1}^{n}\left(d_{i_{j,j+1}}^{\frac{1}{2}}(d_{a_j}d_{a_j})^{\frac{1}{2}}G^{a_j a_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}\right) \\ &= \mathrm{Tr}(A_f^n), && (3.50)\end{aligned}$$

where A_f is a matrix whose components $A_f(a,b)$ are $A_f(a,b) = \sum_i G^{abi}_{baf}(d_a d_b)^{\frac{1}{2}} d_i^{\frac{1}{2}}$. If A_f^n has a non-degenerate highest eigen-value λ_f, for large n, $\mathrm{Tr}(A_f^n) \approx \lambda_f^n$. Note that Perron-Frobenius theorem makes sure that λ_0, highest eigen-value of A_0, will be non-degenerate. So we have

$$\lim_{n\to\infty} \mathrm{Tr}(B_0) = \lambda_0^n. \qquad (3.51)$$

For abelian models, $\mathrm{Tr}(B_{f\neq 0} = 0$ since $G^{abi}_{baf} = 0, f \neq 0$. For the double-Fibonacci model to be discussed in 3.9, a simple calculation shows $\lambda_0 = 1 + \gamma^{3/2}$ and $\lambda_1 = 1 - \gamma^{-\frac{1}{2}}$, where $\gamma = d_1 = \frac{1+\sqrt{5}}{2}$ is the quantum dimension of the string. Because $\lambda_1 < 1$, for large n $\mathrm{Tr}(B_1) \approx \mathrm{Tr}(A_1^n) \approx 0$.

On a hemisphere, $\chi_R = 1$, so from lemma 2 we have $\sigma_b = \sum_f d_f B_f$ and $\rho_R = N\sigma_b^2$ where N is a normalization factor. First we calculate the normalization factor N,

$$\begin{aligned} \mathrm{Tr}(\sigma_b^2) &= \mathrm{Tr}(\sum_f d_f B_f)^2 \\ &= \sum_{f,f'} d_f d_{f'} \mathrm{Tr}(B_f B_{f'}) \\ &= \sum_{f,f'} d_f d_{f'} \delta_{f,f'} D^n \\ &= D^n(\sum_f d_f^2). \end{aligned} \tag{3.52}$$

Now, calculating Renyi entropy with renyi index $\alpha = 1/2$, we get

$$\begin{aligned} S_{1/2}(\rho_R) &= \frac{1}{1-1/2} \log \mathrm{Tr}(\rho_R^{\frac{1}{2}}) \\ &= 2\log \frac{\mathrm{Tr}(\sum_f d_f B_f)}{\sqrt{D^n \sum_f d_f^2}} \\ &= -n\log D - 2\log \sum_f (d_f \mathrm{Tr} B_f) - \log \sum_f d_f^2 \\ &= -n\log D - 2n\log\lambda_0 - 2\log\left(1 + \sum_{f>0} \frac{\mathrm{Tr}B_f}{\lambda_0^n}\right) - \log\sum_f d_f^2. \end{aligned} \tag{3.53}$$

We know that for a string-net model topological entanglement entropy is $\log \sum_f d_f^2$, which implies $\lim_{n\to\infty} \frac{\mathrm{Tr}B_f}{\lambda_0^n} = 0$, $\forall f > 0$. This completes the proof. □

String-net stand-alone subspace

Now we combine lemma 2 and lemma 3.42 to calculate the stand-alone space.

Theorem 1. *The stand alone space of the triple-line string net TNR is given by*

$$M_0 = \delta_{a_1,a_2,i_{12}} \delta_{a_2,a_3,i_{23}} \delta_{a_3,a_1,i_{31}}. \tag{3.54}$$

Proof. Now we are ready to calculate the stand-alone space. Consider the same tensor network but on a very large disc with one triangle removed from the origin. We will denote this space as $D - \Delta$. This has two disconnected boundaries, one on the outer edge, one on the inner one. $\chi_R = 0$ for this region, so using lemma 2

$$\mathbb{T}(D-\Delta) = \sum_f B_f(\partial(D-\Delta)) \tag{3.55}$$

$$= \sum_f B_f(\partial\Delta) \otimes B_f(\partial D) \tag{3.56}$$

To get the stand-alone space, we simply trace out the inner indices on the outer edge. But according to lemma 3.42, only $Tr(B_0$ contribute in the large disc limit. So we simply get (up to an overall normalization factor which we ignore) B_0 on the triangle,

$$\begin{aligned}\lim_{|D|\to\infty}\mathbb{T}_D(D-\Delta) &= \lim_{|D|\to\infty}\sum_f B_f(\partial\Delta)\otimes Tr(B_f(\partial D)) \qquad (3.57)\\ &= B_0(\partial\Delta)\lambda_0^n. \qquad (3.58)\end{aligned}$$

But using (3.45) we get

$$B_0(\partial\Delta) = (d_{i_{12}}d_{i_{23}}d_{i_{31}})^{\frac{1}{2}}\delta_{a_1,a_2,i_{12}}\delta_{a_2,a_3,i_{23}}\delta_{a_3,a_1,i_{31}}. \tag{3.59}$$

Stand-alone projector, M_0, is simply the projector onto the support space of B_0, which is clearly $delta_{a_1,a_2,i_{12}}\delta_{a_2,a_3,i_{23}}\delta_{a_3,a_1,i_{31}}$. So we have proved that M_0 for triple-line TNR of general string-net is,

$$M_0 = \delta_{a_1,a_2,i_{12}}\delta_{a_2,a_3,i_{23}}\delta_{a_3,a_1,i_{31}}. \tag{3.60}$$

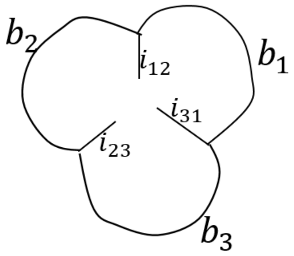
This completes the proof. □

This is the projector on to the stand-alone space of triple-line TNR of general string-net models. For notational convenience we will denote these basis vectors as $\left|\{\prod_{k=1}^{3}\delta_{b_k,b_{k+1},i_{k,k+1}}\}\right\rangle$, that is,

$$\left|\{\prod_{k=1}^{3}\delta_{b_k,b_{k+1},i_{k,k+1}}\}\right\rangle = \delta_{b_1,b_2,i_{12}}\delta_{b_2,b_3,i_{23}}\delta_{b_3,b_1,i_{31}}\left|b_1,b_2,b_3;i_{12},i_{23},i_{31}\right\rangle \tag{3.61}$$

These basis vectors can be represented as string-configurations,

$$\left|\{\prod_{k=1}^{3}\delta_{b_k,b_{k+1},i_{k,k+1}}\}\right\rangle =$$



$$. \tag{3.62}$$

So we get,

$$dim(M_0) = \sum_{b_1,b_2,b_3;i_{12},i_{23},i_{31}}\delta_{b_1,b_2,i_{12}}\delta_{b_2,b_3,i_{23}}\delta_{b_3,b_1,i_{31}}. \qquad . \tag{3.63}$$

3.5 MPO subspace of String-net triple-line TNR

we will use definition 2 to find the MPO subspace of triple-line TNR. Using the triple-line TNR T^0 of string-net states given in Eq. (3.23), the virtual density matrix is found to be

$$\begin{aligned}\sigma &= \sum_I (T^0)^I_\alpha (T^{0;*})^I_\alpha \\ &= \sum_{\{a_k,b_k;i_{k,k+1}\}} G^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} G^{i_{23}i_{31}i_{12}}_{b_1b_2b_3} \prod_j (d^{\frac{1}{6}}_{a_j} d^{\frac{1}{6}}_{b_j} d^{\frac{1}{2}}_{i_{j,j+1}}) \\ &\quad \left|\{a_k; i_{k,k+1}\}\right\rangle \left\langle \{b_k; i_{k,k+1}\}\right| .\end{aligned} \tag{3.64}$$

Clearly, this density matrix can simply be written as

$$\sigma = \sum_{i,j,k} \left|v_{i,j,k}\right\rangle \left\langle v_{i,j,k}\right| \tag{3.65}$$

$$\text{where} \left|v_{i,j,k}\right\rangle = (d_i d_j d_k)^{\frac{1}{4}} \sum_{a_1,a_2,a_3} G^{i,j,k}_{a_1,a_2,a_3} (d_{a_1} d_{a_2} d_{a_3})^{\frac{1}{6}} \left|a_1, a_2, a_3; i, j, k\right\rangle \tag{3.66}$$

So σ has a diagonal form in terms of vectors $v_{i,j,k}$. To get the projector on to its support space,we simply need to use the unit vectos $\frac{1}{N_{i,j,k}} \left|v_{i,j,k}\right\rangle$, where $N_{i,j,k} = \sqrt{\left\langle v_{i,j,k} \middle| v_{i,j,k}\right\rangle}$ is the norm of vector $v_{i,j,k}$. So the string-net MPO projector is

$$\mathbb{M} = \sum_{i,j,k} \frac{1}{N^2_{i,j,k}} \left|v_{i,j,k}\right\rangle \left\langle v_{i,j,k}\right| . \tag{3.67}$$

Note that $\left|v_{i,j,k}\right\rangle = 0$ if $\delta_{i,j,k} = 0$. It means that $\mathbb{M}$ projects on to the physical states allowed by the branching rules, and

$$dim(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} \tag{3.68}$$

Comparing Eq. 3.63 with Eq. 3.68, we can see that $dim(M_0) > dim(\mathbb{M})$. So according to our conjecture, *there must always be unstable directions of variations in the triple-line TNR of any string-net model!* Indeed, we give examples of such unstable directions and will prove in section 3.10 that triple-line TNR of string-net always have instabilities.

String-net MPO from Wilson loop operators

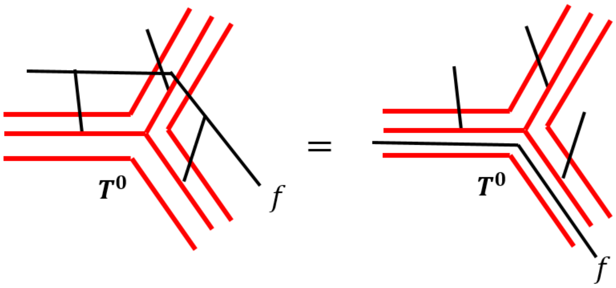
In last chapter, we showed how MPO symmetries or the MPO subspace come from representation of anyonic Wilson-loops of the model on the stand-alone space. It is instructive to do the same with general string-net models.

When an f-type simple string operator passes through the tensor T^0 on the physical level, it induces an operation on the virtual level in the way shown in Fig. 3.69.

That is, it simply becomes a f-type string which is then fused with the plaquette legs. We consider the Wilson loop that encircles the 3 plaquettes of the tensor. This Wilson loop creates an f-type string that then fuses with the 3 plaquette loops. Remember that we need to calculate the representation of this operator on the stand-alone space. That is, we need to calculate the matrix elements $\left\langle \{\delta_{a_k,a_{k+1},i_{k,k+1}} \,\middle|\, W_f \,\middle|\, \{\delta_{b_k,b_{k+1},i_{k,k+1}}\}\right\rangle$. So we imagine a tensor network in which the tensor in the stand-alone basis is surrounded by T^0. We now apply the Wilson loop encircling 3 plaquettes and calculate induced operator on the stand-alone basis. It can be done in a convenient way using string-net diagrams in Eq. (3.62).

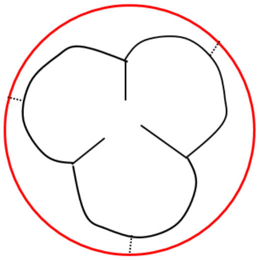
There are essentially 3 steps:

Step 1: Since the surrounding tensors are the fixed point tensor T^0, the Wilson loop on the physical level simply becomes an f-type string that fuses with the plaquette legs,



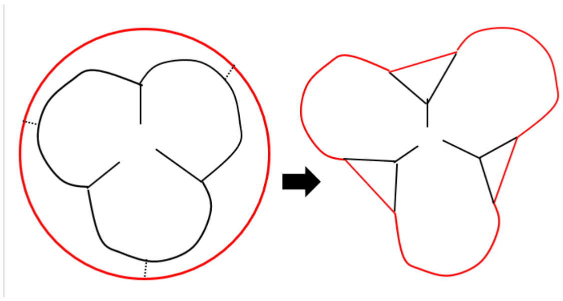
(3.69)

Since these plaquette legs are contracted with the plaquette legs of the stand-alone tensor, it is equivalent to fusing f-string loop with the 3 plaquette legs of the stand-alone tensor,



. (3.70)

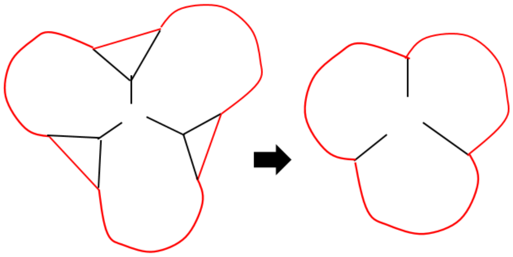
Step 2: We fuse these strings with the three nearby strings a_1, a_2, a_3,



. (3.71)

Let's say they fuse to make strings b_1, b_2, b_3. We gain factors $F^{a_j,a_j,0}_{f,f,b_j} = \frac{\sqrt{d_{b_j}}}{\sqrt{d_f d_{a_j}}}$, $j = 1, 2, 3$ for each fusion.

Step 3: In the last step we remove the three bubbles created in the previous step,



. (3.72)

Each bubble removal produces a factor of $\sqrt{d_f d_{a_j} d_{a_{j+1}}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}$, $j = 1, 2, 3$.

Collecting the factors from step 2 and step 3, we get

$$\left\langle \{\prod_{k=1}^{3} \delta_{b_k,b_{k+1},i_{k,k+1}} \middle| W_f \middle| \{\prod_{k=1}^{3} \delta_{a_k,a_{k+1},i_{k,k+1}}\} \right\rangle = \prod_{j=1}^{3} d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}. \tag{3.73}$$

This is the expression for $M_f = M_0 W_f M_0$. Now considering the projector $\mathbb{M} = \sum_f \frac{d_f}{D} M_f$, we get

$$\begin{aligned} \mathbb{M} &= \sum_f \frac{d_f}{D} \prod_{j=1}^{3} d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \\ &= \frac{1}{D} d_{b_j}^{\frac{1}{6}} d_{a_j}^{\frac{1}{6}} G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \end{aligned} \tag{3.74}$$

It should be understood as an operator written in its components in the basis $\left|\{a_k; i_{k,k+1}\}\right\rangle \left\langle\{b_k; i_{k,k+1}\}\right|$. We used pentagon identity in the second step. We can see that it projects on to the space with $G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} \neq 0$, that is $\delta_{i_{23},i_{31},i_{12}} \neq 0$.

There is a small technical issue though. The factor $d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}}$ does not exactly match the factors in the $TT^\dagger$ support space given in Eq. (3.66)). It is simply because we did not keep track of exactly how to distribute factors that share a plaquette while applying the Wilson loop. In fact, the Wilson loop around a single vertex is somewhat ill-defined. But we are only trying to get a symmetry condition on the individual tensors which makes sure that Wilson loop on a larger region is a symmetry of the state. We can show that this factor has to be exactly $d_{b_j}^{\frac{1}{6}} d_{a_j}^{\frac{1}{6}}$ if the Wilson loop is to be a symmetry of the state. The reason is simply, as concluded in the original string-net paper, a Wilson loop commutes with the plaquette term $B_p = \sum_s a_s B_p^s$ only when $a_s = d_s$. In tensor network language, it translates to the fact that every tensor must contribute a factor of $d_s^{\frac{1}{6}}$ for the Wilson loop to be a symmetry. Also we know that an f-type Wilson loop applied to the ground state produces a factor of d_f. Combining all these we can write the exact Wilson loop operator on a single tensor as:

$$\begin{aligned} M_f &= \prod_{j=1}^{3} d_{a_j} (d_{b_j} d_{a_j}^{-1})^{\frac{1}{6}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} & (3.75) \\ \Rightarrow \mathbb{M} &= \frac{1}{D} d_{a_j} (d_{b_j} d_{a_j}^{-1})^{\frac{1}{6}} G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. & (3.76) \end{aligned}$$

The fixed point triple-line tensor satisfies

$$\begin{aligned} M_f T^0 &= d_f T^0, & (3.77) \\ \mathbb{M} T^0 &= T^0. & (3.78) \end{aligned}$$

One can check that $\mathbb{M} = \sum_f \frac{d_f}{D} M_f$ is indeed a projector and it projects onto the support space of $TT^\dagger$.

Finally, just like boundary operators B_f, f-type MPO can be extended to an arbitrary large region as

$$M_f(\partial R) = \prod_{j=1}^{n} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} d_{a_j} (d_{b_j} d_{a_j}^{-1})^{\frac{\theta_j}{2\pi}}. \tag{3.79}$$

and it represents the operation induced on the virtual level by a Wilson loop applied on the boundary of the region R.

3.6 0-type string operator is a zero-string operator of triple-line TNR

In last chapter we argued how the reason for instability is that some of the non-trivial anyon operators might have a trivial representation on the virtual level. That is, they disappear identically on the ground state tensor network, even in the presence of a topological hole. We saw that, for the single-line and double-line TNR of toric code, X-string and Z-string operators were the zero-string operators respectively. Indeed, the general string-net also has such an operator. These are the operators that only has string-type 0 in it. Remember that string-operators on the string net model act by adding a string-type (possibly more than one) to the string-net and then fusing it with the string-net by some fusion rules. The expression of Wilson-loop operators can be used to see how a string operator with open ends would act on the tensors along the path. It would look the same as in Eq. (3.79) along the path with some changes at the end. But we don't worry too much about the details of how this operator looks at its ends, since those details can always be changed using local unitary operators at its ends. Looking at the Wilson-loop operators in Eq. (3.79), it is immediately clear what the invisible string operators are for the triple-line TNR of general string-nets. For $f = 0$, (using identity (3.10))

$$\begin{aligned} M_0(\partial R) &= \prod_{j=1}^{n} \left(G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j 0} d_{b_j} (d_{a_j} d_{b_j}^{-1})^{\frac{m_j}{6}} \right) \\ &= \prod_{j=1}^{n} \left(\delta_{a_j, b_j} (d_{a_j} d_{b_j})^{-\frac{1}{2}} d_{b_j} (d_{a_j} d_{b_j}^{-1})^{\frac{m_j}{6}} \delta_{a_j, a_{j+1}, i_{j,j+1}} \right) \\ &= \prod_{j=1}^{n} \delta_{a_j, a_{j+1}, i_{j,j+1}} . \end{aligned} \tag{3.80}$$

But the final expression is the very definition of stand-alone space itself. It means this operator will act trivially on the stand-alone space. So, a 0-type simple string operator is a non-trivial invisible string operator, that is, it is a zero-string operator. From this it should be clear why we denoted the stand-alone space M_0 and why we called non-trivial invisible string-operators zero-string operators. These names come from the general string-net formalism.

It is also clear that for $f \neq 0$, M_f acts necessarily non-trivially on the tensors along the path. One should carefully note that, though non-zero-string operators change tensors along the path, it does not mean that this path is a physical observable. These paths can always be deformed as M_f passes through T^0 without any phase accumulation. It is called the 'pulling-through condition' [49]. When there is an MPO violating variation present at a tensor, M_f cannot be pulled through it. Hence our conjecture can be alternatively worded as 'the stand-alone variations which prohibit the pulling-through property of fixed point tensors cause instability.'

3.7 Tensors in the unstable space $M_0 - \mathbb{M}$

We have determined both the stand-alone space, M_0 and the MPO space $\mathbb{M}$. M_0 space is spanned by vectors,

$$\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}\,|a,b,c;i,j,k\rangle\,. \tag{3.81}$$

And the MPO space $\mathbb{M}$ space is spanned by $\left|v_{i,j,k}\right\rangle : \delta_{i,j,k} = 1$, where

$$\left|v_{i,j,k}\right\rangle \;=\; \sum_{a,b,c} G^{i,j,k}_{a,b,c}(d_a d_b d_c)^{\frac{1}{6}}\,|a,b,c;i,j,k\rangle\,. \tag{3.82}$$

The tensors supported on $M_0 - M$ are precisely the tensors that cause instability. To determine the orthogonal basis of this space we simply need to find vectors orthogonal to $v_{i,j,k}$ which are within the stand-alone space. First note that M_0 space decomposes in orthogonal subspaces $M_0 = \bigoplus_{i,j,k}\mathbb{V}_{i,j,k}$ where the subspace $\mathbb{V}_{i,j,k}$ is spanned by $\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}\,|a,b,c;i,j,k\rangle$, that is, a,b,c for which $\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}$ is non-zero. $M_0 - \mathbb{M}$ space can be decomposed into two subspaces,

1- $\delta_{i,j,k} = 0$: This consists of all the string-configurations in Fig. 3.62 for which $\delta_{i,j,k} = 0$. They are obviously orthogonal to all $v_{i,j,k}$ since $v_{i,j,k} = 0$ if $\delta_{i,j,k} = 0$. Since these vectors violate the vertex term of the Hamiltonian we will refer to them as ‘vertex variations’.

2- $\delta_{i,j,k} = 1$: This is the subspace spanned by string configurations for which $\delta_{i,j,k} = 1$. We need to find other vectors in $\mathbb{V}_{i,j,k}$ that are orthogonal to $v_{i,j,k}$. $\dim(\mathbb{V}_{i,j,k}) = \sum_{a,b,c}\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = \sum_{a,b,c}[G^{i,j,k}_{c,a,b}]$ where $[G^{i,j,k}_{c,a,b}] = 1$ if $G^{i,j,k}_{c,a,b} \neq 0$ and 0 otherwise. Note that since $\mathbb{V}_{i,j,k}$ are orthogonal for different values of i,j,k, we just need to find vectors in individual $\mathbb{V}$ subspaces. To find these we will use the orthogonality of G (3.9)

$$\sum_c G^{i,j,k}_{a,b,c}G^{i,j,k}_{a,b,c}d_c = \frac{1}{d_k}\delta_{a,b,k}. \tag{3.83}$$

And the fact that matrices N^k defined by $N^k_{a,b} = \delta_{a,b,k}$ can be simultaneously diagonalized $\forall k$. Let's say $\left|s_q\right\rangle = s_{q;a}\,|a\rangle$ is its qth such simultaneous eigenvector. As discussed in the section 3.2, $s_{0;a} = d_a$, that is, the vector formed by quantum dimensions is an eigenvector to N^k. These vectors are orthogonal, $\left\langle s_q \middle| s_{q'}\right\rangle = \delta_{q,q'}$, which also implies that $\left\langle s_q\middle| N^k \middle| s_{q'}\right\rangle = \sum_{a,b} s_{q;a}\delta_{k,a,b}s_{q';b} \propto \delta_{q,q'}$. Now we are ready to write down the vectors spanning $\mathbb{V}_{i,j,k}$.

Consider vectors

$$\left|v^{q;a}_{i,j,k}\right\rangle = \sum_{a,b,c}\frac{s_{q;a}}{d_a}G^{i,j,k}_{a,b,c}(d_a d_b d_c)^{\frac{5}{6}}\,|a,b,c;i,j,k\rangle\,, \tag{3.84}$$

where superscript $(q;a)$ indicates that the qth eigenvector is used on leg a. Using the orthogonality relation, we get

$$\begin{aligned}
\left\langle v_{i,j,k} \middle| v^{q;a}_{i,j,k} \right\rangle &= \sum_{a,b,c} s_{q;a} d_b d_c G^{i,j,k}_{a,b,c} G^{i,j,k}_{a,b,c} && (3.85)\\
&= \sum_{a,b} s_{q;a} \delta_{a,b,k} d_b && (3.86)\\
&= \sum_{a,b} s_{q;a} \delta_{a,b,k} s_{0;b} && (3.87)\\
&\propto \delta_{q,0}. && (3.88)
\end{aligned}$$

So we see that the vector $v^{q;a}_{i,j,k}$ is orthogonal to $v_{i,j,k}$ if $q \neq 0$. Since q takes $N-1$ non-zero values and it can be put on leg a, b or c we seem to have $3(N-1)$ such vectors. However not all of them will be independent, but they span the full vector space $\mathbb{V}_{i,j,k}$. Since these kinds of variations change the plaquette leg factors, hence violating the plaquette term, we will refer to these variations as ‘plaquette variations’.

With this we have concluded the analysis of general string-net mdodels and their triple-line TNR. Now we turn to some concrete examples to understand how the conjecture in 2.6 explains the instabilities in string-nets.

3.8 Examples: Triple-line TNR of the toric code and double semion states

let’s first examine how models covered in the previous chapter, toric code and double semion, fit into triple-line TNR. One can get the triple-line TNR for them by plugging in the relevant string-net data into Eq. (3.23). We will apply the results about general-string net models developed in previous sections to the two cases.

Toric code string-net data is,

$$\begin{aligned}
N = 1,\ d_0 = 1,\ d_1 &= 1;\\
\delta_{000} = \delta_{110} = \delta_{101} = \delta_{110} &= 1;\\
G^{000}_{000} = G^{000}_{111} &= 1;\\
G^{011}_{011} = G^{011}_{100} = G^{101}_{101} = G^{101}_{010} = G^{110}_{110} = G^{110}_{001} &= 1.
\end{aligned} \tag{3.89}$$

The triple-line TNR of toric code can be built by plugging in this data into the general expression in Eq. (3.23). This tensor has 9 virtual indices, each of which takes 2 values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$ dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 8, \tag{3.90}$$

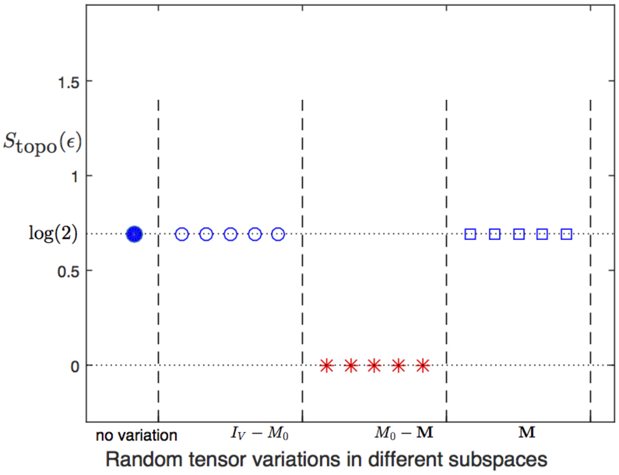


Figure 3.1: Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by toric code fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Details of this numerical calculation are given in appendix A.2.

and the dimension of the MPO subspace is

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 4. \tag{3.91}$$

These imply that $\text{rank}(I_V - M_0) = 512 - 8 = 504$ and $\text{rank}(M_0 - \mathbb{M}) = 8 - 4 = 4$. So we reach the conclusion that out of 512 possible variations, 504 are stable since they are outside the stand-alone space. In the remaining 8 dimensional subspace, perturbations in a 4 dimensional subspace are in stable whereas the ones in the other 4 dimensional subspace are unstable. The numerical calculation supporting this conclusion is shown in Fig. 3.1. Also note that all unstable variations are flux variations, that is, it happens through the condensation of m-particle. It is not possible for the e-particle to condense in this way. The classification of all variations is shown in Fig. 3.2.

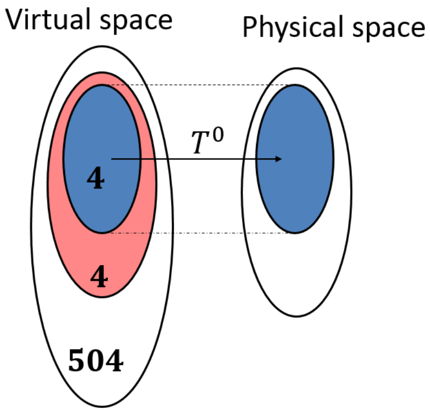


Figure 3.2: Classification of the space of all variations to the toric code triple-line tensor into different subspaces. Subspace in blue is MPO susbpace $\mathbb{M}$ which is 4 dimensional. Region in red is the stand-alone space outside MPO subspace, $M_0 - \mathbb{M}$ and it is 4 dimensional. Finally virtual space outside the stand-alone space is 504 dimensional. Only the region in red is unstable, while the ones in blue and white are stable.

For the double semion model, the string-net data is

$$\begin{aligned} N = 1,\ d_0 = 1,\ d_1 = 1; \\ \delta_{000} = \delta_{110} = \delta_{101} = \delta_{110} = 1; \\ G^{000}_{000} = 1; \\ G^{011}_{011} = G^{101}_{101} = G^{110}_{110} = -1; \\ G^{011}_{100} = G^{101}_{010} = G^{110}_{001} = G^{000}_{111} = -i. \end{aligned} \tag{3.92}$$

The triple-line TNR of the double semion model can be built by plugging in this data into the general expression in Eq. (3.23). This tensor has 9 virtual indices, each of which takes 2 values. So the full virtual space is $\mathrm{rank}(I_V) = 2^9 = 512$ dimensional. Dimension of the stand-alone space is

$$\mathrm{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 8, \tag{3.93}$$

and the dimension of the MPO subspace is

$$\mathrm{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 4. \tag{3.94}$$

These imply that $\mathrm{rank}(I_V - M_0) = 512 - 8 = 504$ and $\mathrm{rank}(M_0 - \mathbb{M}) = 8 - 4 = 4$. So we reach the conclusion that out of 512 possible variations, 504 are stable since

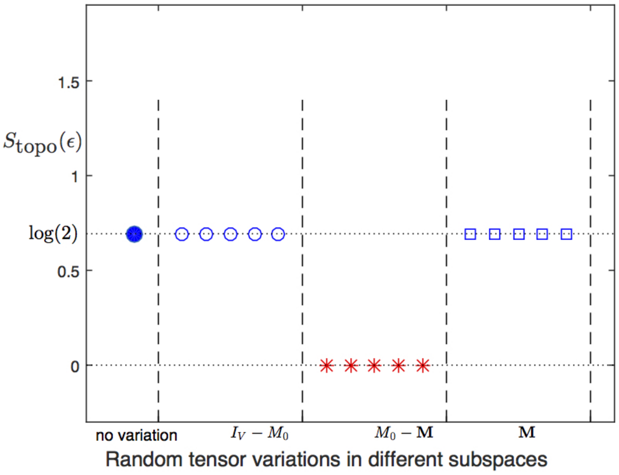


Figure 3.3: Numerical calculation of topologiccal entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by double semion model fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Details of this numerical calculation are given in the appendix A.2.

they are outside the stand-alone space. In the remaining 8, 4 are in stable and 4 are unstable. The numerical calculation supporting this conclusion is shown in Fig. 3.3. Also note that all unstable variations are plaquette variations, that is, it happens through condensation of the boson of the double-semion model. The classification of all variations is the same as that for toric code shown in Fig. 3.2.

Now we are ready to discuss a concrete example of the string-net triple line TNR and its instabilities. We choose double-Fibonacci model for two main reasons: 1- Unlike the toric code and the double-semion model, it is a non-abelian model, so the general triple-line TNR, as far as we know, cannot be reduced to a double-line or single-line TNR. So it serves as a good example to test our conjecture for the general string-net TNR. 2- Unlike toric code and double-semion, its bosonic string operator is not a zero string operator, so it does not disappear along the path.

3.9 A non-abelian example: Double-Fibonacci Model

Toric code and the double-semion models are abelian models. Now we will discuss a non-abelian model: the double-Fibonacci model. The Ground state of non-abelian string net models cannot be described by a single-line or the double-line TNR; it only accepts a triple line TNR (tensor in . (3.25)). Let's first describe the model briefly. The data for this can be found in section IV.B of Levin and Wen [13]. There is one type of string ($N = 1$). Its quantum dimension is, $d_1 = \gamma = \frac{1+\sqrt{5}}{2}$. Its branching rules are,

$$\delta_{ijk} = \begin{cases} 0 & \text{if } i + j + k = 1; \\ 1 & \text{otherwise.} \end{cases}$$

$$d_0 = 1,\ d_1 = \gamma, \quad \text{where } \gamma^2 = \gamma + 1 \tag{3.95}$$

$$G^{111}_{111} = -\frac{1}{\gamma^2};\ G^{110}_{111} = \frac{1}{\gamma};\ G^{110}_{110} = \frac{1}{\gamma};\ G^{000}_{111} = \frac{1}{\sqrt{\gamma}};\ G^{000}_{000} = 1. \tag{3.96}$$

The branching rules tells us that one string is allowed to branch into two, unlike the abelian models we have studied until now. First, let's apply our conjecture to find out how many unstable directions we should expect. The triple-line TNR of the Fibonacci model can be built by plugging in this data into the general expression in Eq. (3.23). This tensor has 9 virtual indices, each of which takes 2 values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$ dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 18, \tag{3.97}$$

which is significantly bigger than that of the toric code and the double-semion models. The dimension of the MPO subspace is

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 5 \tag{3.98}$$

which implies that $\text{rank}(I_V - M_0) = 512 - 18 = 494$ and $\text{rank}(M_0 - \mathbb{M}) = 18 - 5 = 13$. So we reach the conclusion that out of 512 possible (virtual) variations, 494 are stable since they are outside the stand-alone space. In the remaining 18, 5 are in stable as they are in the MPO subspace and remaining 13 are unstable. The numerical calculation supporting this conclusion is shown in Fig. 3.5. The classification of all variations is given in Fig. 3.4.

Comparing it to the toric code and the double-semion models we see that the Fibonacci triple-line TNR is significantly more unstable. Another difference is that the stand-alone space does have vertex unstable variations in addition to plaquette

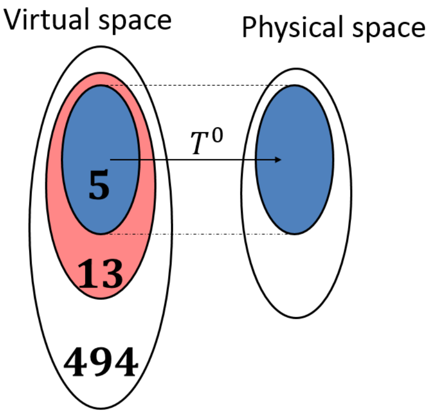
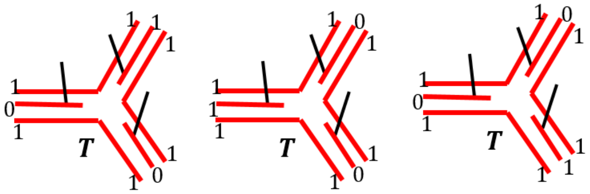


Figure 3.4: Classification of the space of all variations to the toric code triple-line tensor into different subspaces. Subspace in blue is MPO susbpace $\mathbb{M}$ which is 5 dimensional. Region in red is the stand-alone space outside MPO subspace, $M_0 - \mathbb{M}$ and it is 13 dimensional. Finally virtual space outside the stand-alone space is 494 dimensional. Only the region in red is unstable, while the ones in blue and white are stable.

ones. Out of 13 unstable variations in $M_0 - \mathbb{M}$ the following 3 are vertex variations and the rest 10 are plaquette variations:

$$|a, b, c; i, j, k\rangle = |1, 1, 1; 1, 0, 0\rangle, |1, 1, 1; 0, 1, 0\rangle, \\ |1, 1, 1; 0, 0, 1\rangle. \tag{3.99}$$

That is, the following 3 tensor components are allowed in the stand-alone space but not in the physical space:



. (3.100)

Since $\delta_{i,j,k} = \delta_{1,0,0} = \delta_{0,1,0} = \delta_{0,0,1} = 0$ these 3 vectors are not in the MPO subspace $\mathbb{M}$.

To understand the physics behind this, we need to look at the quasi-particles of the Fibonacci model. There are 3 quasi-particles excitations: τ, $\bar{\tau}$, and $\tau\bar{\tau}$. The T and S

matrices of the particles are as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-\frac{4}{5}\pi i} & 0 & 0 \\ 0 & 0 & e^{\frac{4}{5}\pi i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S = \frac{1}{1+\gamma^2} \begin{bmatrix} 1 & \gamma & \gamma & \gamma^2 \\ \gamma & -1 & \gamma^2 & -\gamma \\ \gamma & \gamma^2 & -1 & -\gamma \\ \gamma^2 & -\gamma & -\gamma & 1 \end{bmatrix}. \tag{3.101}$$

It is best seen as two layers of Fibonacci model with opposite chiralities. τ and $\bar{\tau}$ are particles in the two respective layers. They have non-trivial self statistics. But, because they are in different layers, they have a trivial statistics with one another. And the boson, $\tau\bar{\tau}$ is the composition of the Fibonacci particles in the two layers. The string operator for these quasi-particles are given in equation (51) of Levin and Wen [51]. We are most interested in the boson of the model, so let us write its string operator (Ω matrices) explicitly:

$$\begin{gathered} n_{4,0} = 1, n_{4,1} = 1, \Omega^0_{4,000} = 1, \Omega^1_{4,110} = 1, \\ \Omega^1_{4,001} = -\gamma^{-2}, \Omega^0_{4,111} = \gamma^{-1}, \Omega^1_{4,111} = \gamma^{-5/2}, \\ \Omega^1_{4,101} = \Omega^{*1}_{4,011} = \gamma^{-11/4}(2 - e^{3\pi i/5} + \gamma e^{-3\pi i/5}). \end{gathered} \tag{3.102}$$

One can see that it is not a simple-string operator: when applied on the vacuum, it creates both 0-type and 1-type strings. So we see that the double-Fibonacci model is different from the above two examples in one crucial aspect: the boson string operators in the toric code and the double-semion models were zero-string operators for the given TNRs. That is, the string operator 'disappeared' along the path (Fig. 2.592.60), not changing tensors along the path. This is why a single variations standing alone could be thought of as an operator sitting at the ends of an invisible string operator. But the same is not true for the double-Fibonacci model. The string operator corresponding to the boson $\tau\bar{\tau}$ does not disappear in the middle.

Because the bosons don't have a zero string operator, one might conclude that there would be no unstable directions as bosons cannot condense. However, numerical calculations find that there actually are unstable directions. How can we understand that?

We look at how the boson string-operator changes the tensors along the path. In Fig. 3.6, one can see that a wave function corresponding to the boson sitting at two

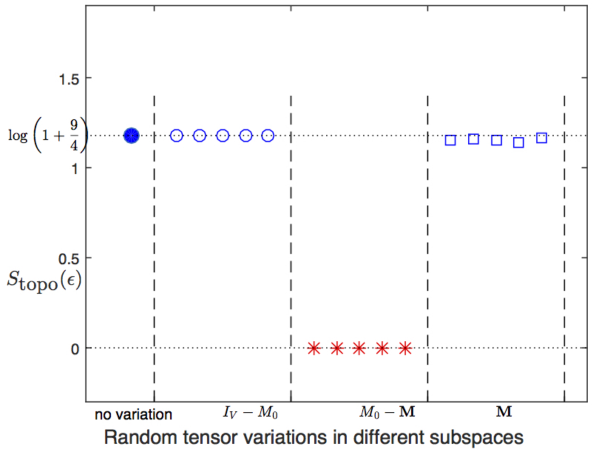


Figure 3.5: Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by Fibonacci model fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. The exact numerical values on this plot can be found in appendix A.2.

places, v_1 and v_2, is actually a superposition of many wave functions:

$$\begin{aligned} |\Psi_{\text{boson}}\rangle &= \sum_{t_1,s,t_2} n_s \Phi_{t_1,s,t_2} \left|\Psi_{\text{gs}}\right\rangle \\ &= \left|\Psi_{0,0,0}\right\rangle + \left|\Psi_{1,0,0}\right\rangle + \left|\Psi_{0,0,1}\right\rangle + \left|\Psi_{1,0,1}\right\rangle \\ &+ \left|\Psi_{0,1,0}\right\rangle + \left|\Psi_{1,1,0}\right\rangle + \left|\Psi_{0,1,1}\right\rangle + \left|\Psi_{1,1,1}\right\rangle, \end{aligned} \tag{3.103}$$

where the operator Φ_{t_1,s,t_2} is explained in Fig. 3.6. Φ_{t_1,s,t_2} is equivalent to applying $\Omega^{s_1'}_{4;t_1,s,s_1}$ and $\bar{\Omega}^{s_n'}_{4;s,t_2,s_n}$ on the loops at the ends of the string operator, and creating a s type string along the path. Fusing the loops with each other and with the s string along path P gives the final state. The important thing to note is that, though a TNR of the full state $|\Psi_{\text{boson}}\rangle$ involves changing tensors along the path, the TNR of $\left|\Psi_{t_1,0,t_2}\right\rangle$, $t_1, t_2 = 0, 1$, have tensors changed only on the ends. Simply putting, the zero-string component of the string operator does not change the tensors T^0 in the middle, as expected. So the boson state has a finite overlap with the state where tensors are changed only at the ends. So when the variations corresponding to the ends of this zero-string component of the boson operator proliferate, it effectively condenses the bosons, as they have finite overlap with the resulting state.

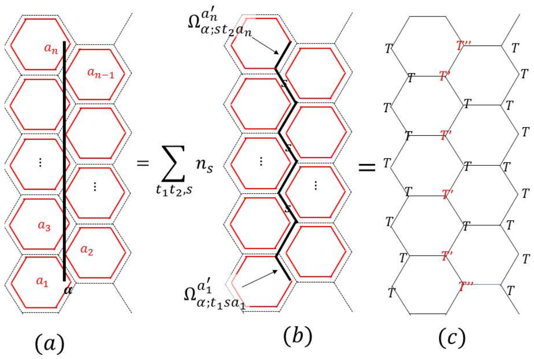

a_n
a_{n-1}
a_3
a_2
a_1
α
$= \sum_{t_1 t_2, s} n_s$
$\Omega^{a'_n}_{\alpha;st_2a_n}$
s
$\Omega^{a'_1}_{\alpha;t_1sa_1}$
$=$
T
T'
T''
(a)
(b)
(c)

Figure 3.6: Action of a generic (simple and non-simple) open-end string operators corresponding to anyon α on tensors can be calculated in a similar fashion as that of simple-string operator Wilson loops. (a) We start with applying the string operator on the 'loop state' on the fattened lattice. (b) The string operator becomes a superposition of operations $\sum_s n_s \Phi_{t_1 s t_2}$. $\Phi_{t_1 s t_2}$ acts as follows: at the ends, the string operator acts as $\Omega_{\alpha;t_1 s}$ and $\omega_{\alpha;st_2}$ matrices on the plaquette-loops, while in the middle, it is simply a s-type string to be fused with the nearby plaquette loops. (c) We fuse all strings in the previous step to get the physical state. The effect of the string operator can be absorbed into redefining the tensors along the path. A generic string operator changes the tensors along its path. The only case where it doesn't change the tensors is for simple-string operators of type 0.

So in conclusion, we see that although the boson string operator is not a zero-string operator, that is, it does not disappear in the middle for the triple-line TNR, its zero-string component still causes an instability because the resulting state has a finite overlap with the boson-condensed state.

Now we have looked through important examples of string-net TNR and their instabilities. Finally, we will give a proof of instability in the generic case.

3.10 Proof of the existence of instability in general string-net triple-line TNR

We will give an analytical proof of why all string-net triple-line TNR have at least one unstable direction which comes from the $M_0 - \mathbb{M}$ subspace. We will do so by directly calculating $S_{\text{topo}}(\epsilon)$.

Topological entanglement entropy on a cylinder with non-RG fixed point tensor

Lemma 4. *let's say we divide the cylinder in two halves (Fig. 2.1(a)). We denote the right half as R. If any given tensor network on this cylinder satisfies,*

$$\lim_{|R|\to\infty} \mathbb{T}(R) = C^{|R|} \sum_f c_f B_f(\partial R), \tag{3.104}$$

where C is some constant, then, S_{topo}, as given in Eq. (2.7)*, is*

$$S_{topo} = \log \sum_f \left(\frac{c_f^2}{c_0^2} \right). \tag{3.105}$$

Proof. The proof is quite simple. We follow the same steps as used in the proof of lemma 3.42, replacing d_f with c_f. We first calculate the normalization of the density matrix.

$$\begin{aligned} \mathrm{Tr}(\sigma_b^2) &= \mathrm{Tr}(\sum_f c_f B_f)^2 \\ &= \sum_{f,f'} c_f c_{f'} \mathrm{Tr}(B_f B_{f'}) \\ &= \sum_{f,f'} c_f c_{f'} \delta_{f,f'} D^n \\ &= D^n(\sum_f c_f^2). \end{aligned} \tag{3.106}$$

By calculating Renyi entropy with renyi index $\alpha = 1/2$, we get

$$\begin{aligned} S_{1/2}(\rho_R) &= \frac{1}{1-1/2} \log \mathrm{Tr}(\rho_R^{\frac{1}{2}}) \\ &= 2 \log \frac{\mathrm{Tr}(\sum_f c_f B_f)}{\sqrt{D^n \sum_f c_f^2}} \\ &= -n \log D - 2 \log \sum_f (d_f \mathrm{Tr} B_f) - \log \sum_f c_f^2 \\ &= -n \log D + 2n \log \lambda_0 - 2 \log \left(1 + \sum_{f>0} c_f \frac{\mathrm{Tr} B_f}{\lambda_0^n} \right) + 2 \log c_0 - \log \sum_f c_f^2. \end{aligned}$$

When we let $n \longrightarrow \infty$ and using Eq. (3.42)

$$S_{1/2}(\rho_R) = n \log \frac{\lambda_0^2}{D} - \log \sum_f \left(\frac{c_f^2}{c_0^2} \right) \tag{3.107}$$

$$\Rightarrow S_{\mathrm{topo}} = \log \sum_f \left(\frac{c_f^2}{c_0^2} \right). \tag{3.108}$$

This completes the the proof.

□

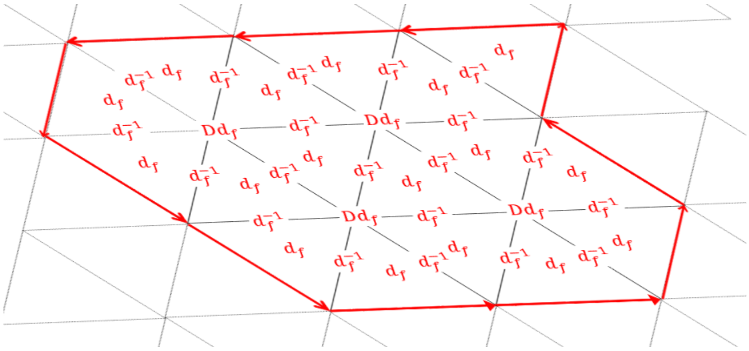
Finally we are ready to show the unstable tensor perturbations in the triple line TNR of the string-net models.

Instability in string-net

Theorem 2. *Let T^0 be the fixed point triple-line TNR of a string net ground state. There exist tensors T^q in the space $M_0 - \mathbb{M}$ (that is,$(M_0 - \mathbb{M})T^q \neq 0$) that for the variation $T^0 \longrightarrow T^0 + \epsilon T^q$, $\lim_{\epsilon \to 0} S_{topo}(\epsilon) \neq S_{topo}(0)$.*

Proof. Combination of the lemma 2, theorem 3.4 and lemma 4 gives a clue to why $T^0 \longrightarrow T^0 + \epsilon T^q$, are unstable variations. We will choose particular variations in $M_0 - \mathbb{M}$ for analytical simplicity, but it should be understood that any arbitrary variation that has a component in those directions will result in instability. We discussed in section 3.7 that there are two kinds of variations in $M_0 - \mathbb{M}$; vertex variations (that violate the vertex term) and plaquette variations (that violate the plaquette term). We will treat them one by one.

Before we do any analytical calculation, let us describe in simple words what the reason for instability is. We saw in the proof of lemma 2 that as fixed point tensors contract, every face, every edge, and every vertex contribute a factor of d_f, d_f^{-1}, and d_f respectively. It can be visualized like this,



(3.109)

It combines to give $c_f = d_f^{F-E+V} = d_f^{\chi_R}$ which is a topological invariant of the lattice. If a tensor variation changes the double tensor in such a way that one of these factors (face, edge or vertices) are changed, even infinitesimally, then the c_f we get is not a topological invariant, and S_{topo} due to lemma 4 changes. We will now show that this is precisely what variations in $M_0 - \mathbb{M}$ do. In particular, the vertex variations change the vertex factors, and the plaquette variations change the face factors.

Let's choose a particular tensor variation

$$T^q = \prod_{j=1}^{3} (d^{\frac{1}{4}}_{i_{j,j+1}}) \delta_{a_j b_{j+1} i_{j,j+1}}, \tag{3.110}$$

such that

$$\mathbb{T}^q = T^q (T^q)^\dagger = B_0. \tag{3.111}$$

This tensor is supported on the full M_0 space and clearly has components outside the MPO subspace because as we showed $M_0 > \mathbb{M}$. So $(M_0 - \mathbb{M})T^q \neq 0$. Now, the double tensor for the varied tensor is

$$\mathbb{T} = (T^0 + \epsilon T^q)(T^0 + \epsilon T^q)^\dagger \approx \mathbb{T}^0 + \epsilon^2 B_0 \tag{3.112}$$

$$= (1 + \epsilon^2) B_0 + \sum_{f>0} d_f B_f. \tag{3.113}$$

We have ignored the linear terms in ϵ as they are contained within the MPO subspace, and we don't need to worry about them. This double tensor will contract with itself in exactly the same way as $\mathbb{T}^0$ did, but the only difference is, now every face will contribute a factor of r_f, where, $r_0 = (1 + \epsilon^2)$, and $r_{f>0} = d_f$. The vertex factors and edge factors will remain to be d_f and d_f^{-1}, respectively. After contracting it on a large region we will get a double tensor $\mathbb{T}(R) = \sum_f c_f B_f(\partial R)$, where $c_f = r_f^F d_f^{V-E}$. So $c_0 = (1 + \epsilon^2)^F$ and $c_{f>0} = d_f^{\chi_R}$. So we see that c_0 is exponentially larger than $c_{f>0}$ even for an infinitesimal ϵ, hence, using Eq. (3.108), $S_{\text{topo}} = 0$.

Now we look an example of plaquette variations. Consider tensors that are exactly the same as the fixed point tensors, except the plaquette factors $d_a^{1/6}$ are replaced by a factor of $(d_a + \epsilon s_{q;a})^{1/6}$, where $s_{q;a}$ is the ath component of the qth eigenvector of δ, as explained in Eq. 3.14.

$$(T^q) = \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{4}} (d_{a_j} + \epsilon s_{q;a_j})^{\frac{1}{6}} \right) G_{a_1 a_2 a_3}^{i_{23} i_{31} i_{12}}. \tag{3.114}$$

This tensor is clearly supported on the stand-alone space, and is outside the MPO subspace as to be inside the MPO subspace it has to have $d_a^{1/6}$ factors. The double tensor will again produce a factor of d_f on the faces, and d_f^{-1} on the edges upon contraction. But now the factors on the vertices would be

$$\sum_{a,b} \delta_{a,b,f} (d_a + \epsilon s_{q;a})(d_b + \epsilon s_{q;b}) = D(d_f + e_{q;f}\epsilon^2), \tag{3.115}$$

where s_q is normalized to give $\langle s_q \,|\, s_q \rangle = D$ and $e_{q;f}$ is the qth eigenvalue of the matrix $N_{a,b}^f = \delta_{a,b,f}$. A conclusion similar to that for vertex variation case follows. $c_f = d_f^{F-E}(d_f + \epsilon^2 e_{q;f})^V = d_f^{\chi_R}(1 + \epsilon^2 \frac{e_{q;f}}{d_f})^V$ is not a topological invariant, as it extensively depends on the number of vertices V. As a result, the weight of one of the boundary operator in $\mathbb{T} = \sum_f c_f B_f$ becomes exponentially larger than the others even for an infinitesimal variation ϵ, and hence the topological order is lost.

Result I-IV together complete the proof that triple-line TNR of general string-net states have at least one unstable direction.

□

3.11 Summary and outlook

In this chapter we analyzed the triple-line TNR of general string nets by first finding their stand-alone and MPO subspaces. We found that the stand-alone space is always bigger than the MPO subspace which suggests, according to our instability conjecture, that TNR of all string-nets have instabilities in them. We showed that triple-line TNR of all string-nets have possible zero-string operators in them. These are the 0-type simple string operators. We saw that unstable variations can be broadly categorized in two categories, the vertex variations, and the plaquette variations. We re-analyzed models from the last chapter, the toric code and double-semion models, under the string-net formalism. We found that triple-line TNR of both models are unstable due to plaquette variations. We then analyzed a non-abelian model, the double-Fibonacci model. This model was interesting because none of the irreducible string-operators disappear on the TNR. So apriori one might expect that the bosons cannot condense in this model. We find instabilities nevertheless. This instability is explained by the fact that state resulting from unstable variation has a finite overlap with the boson condensed state. This also showed that the zero-string operator of a TNR doesn't need to be an irreducible string-operator.

Throughout the last two chapters, we called the tensor instability result a 'conjecture' because, though we proved it for all string-net states, we did not prove it for all TNRs of string-net states. This is an open problem to prove this conjecture for all TNRs. To prove that one would need to show that the stand-alone space is always larger than the MPO space for any TNR, and the stand-alone variations that do not respect the MPO symmetries cause instabilities. This is difficult to prove without a general formalism for all TNRs. But we can set certain guidelines as to how to go about finding if there are instabilities in a given TNR of a topological state. First, if the dimension of the virtual space is the same as that of the local ground-state physical space dimension, then such a TNR definitely would not have any instabilities. The reason is that in such a case all of the virtual space is the MPO space, so there cannot possibly be stand-alone space larger than the MPO subspace. If the virtual space is larger than the local ground-state physical space, then there is a possibility of a stand-alone space. Such a TNR will have virtual variations that can be lifted to the physical level. However then one has to see whether these variations can be applied to a single tensor without collapsing the original tensor network state. If it can be, then that is a definite sign that there is a non-trivial stand-alone space, and so the TNR has instabilities.

There is a general lesson to be learned about interaction of physical symmetries and tensor network states (even beyond topological states) from the results of last two chapters. Symmetries of the wave-function play a crucial role in many areas of physics. A tensor network state however always has a gauge symmetry on the virtual level: one can always apply X and X^{-1} to two contracting virtual legs. This is not a real symmetry of the system, it is just coming from the redundancy of the mathematical description. So in general one has to be aware of how the symmetries on the physical level related with the virtual gauge symmetries on the virtual level. If a physical symmetry becomes a virtual gauge symmetry when mapped to the virtual level, then, in a sense, this symmetry constraint is lost because then this constraint is satisfied identically by the tensor network state.

Chapter 4

TN REPRESENTATION OF LOCALITY-PRESERVING 1D UNITARY OPERATORS

4.1 Introduction

The matrix product formalism [26, 27] has played a significant role in the study of one dimensional systems. In particular, the matrix product representation of 1D quantum states underlies successful numerical algorithms like the Density Matrix Renormalization Group algorithm [28] and the Time-Evolving Block Decimation algorithm [29]. Moreover, the matrix product representation provides a deep insight into the structure of the ground states in 1D [27] which enables rigorous proofs of the efficiency of 1D variational algorithms in search for the ground states [30, 31] and also a complete classification of 1D gapped phases [25, 32–34].

Operators can also be represented in a matrix product form [35–37], which provides a useful tool in the simulation of one dimensional mixed states and real / imaginary time evolutions (see for example Ref. [63, 64]). In particular, matrix product operators which are unitary play an important role in not only the simulation of dynamical processes in 1D , but also the understanding and classification of (symmetry protected) topological phases in 2D [14–18].

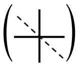
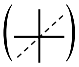
How well does the matrix product formalism represent unitary operators in 1D? Of particular interest are unitaries that preserve the locality structure of the system, that is, unitaries that map local operators to local operators. We want to understand the following: can all locality preserving 1D unitaries be represented using the matrix product form? On the other hand, of course not all matrix product opereators are unitary. But among those that are, what conditions do they have to satisfy to preserve locality? Moreover, it has been shown [1] that locality preserving 1D unitaries can be classified according to how much information they are transmitting across any cut in the 1D chain and each class can be uniquely characterized by a positive rational index, which we refer to below as the GNVW index. We want to know if there is a simple way to extract this GNVW index from the matrix product representation if such a representation exists.

In this chapter, we address the above questions and show that:

- Unitary matrix product operators provide a necessary and sufficient representation of locality preserving unitaries in 1D.

That is, matrix product operators that are unitary are guaranteed to preserve locality by mapping local operators to local operators while at the same time all locality preserving unitaries can be represented in a matrix product way. Moreover, we find that:

- The GNVW index can be extracted in a simple way as the square root of I_{RR}, the 'Rank-Ratio index', which is the ratio between the rank of the left and right singular value decompositions of the tensor representing the operator

$I_{\mathrm{RR}} = \mathrm{rank}\Big($  $\Big) \Big/ \mathrm{rank}\Big($  $\Big)$ (4.1)

$I_{\mathrm{GNVW}} = \sqrt{I_{\mathrm{RR}}}$

The exact meaning and a more rigorous version of this formula is given in section 4.4.

To show this result, we start from the basic requirements for a matrix product operator to be unitary in section 4.2. Based on these basic requirements, we prove in the section 4.3 that after sufficient blocking, the 'fixed point' matrix product operator satisfies a set of nice fixed point properties. Using this set of fixed point conditions, we can show the correspondence between matrix product unitary operators and locality preserving 1D unitaries. Moreover, these conditions enable us to prove in section 4.4 that Eq. 4.1 provides a well-defined index for each equivalence class of 1D locality preserving unitaries and it exactly matches (the square of) the GNVW index. In section 4.5, we compute the index according to Eq. 4.1 numerically for some random locality preserving unitaries and demonstrate how it approaches the expected value as we take larger and larger blocks of the tensor. In section 5.2, we show that the matrix product formalism also provides interesting ways to go beyond the GNVW framework. In particular, we give an example of a simple matrix product operator with 'fractional' index as compared to the locality preserving ones. This example does not contradict with our discussions in the previous sections because it is unitary only in systems of special sizes and does not preserve locality.

The structure of this chapter is illustrated in Fig. 4.1.

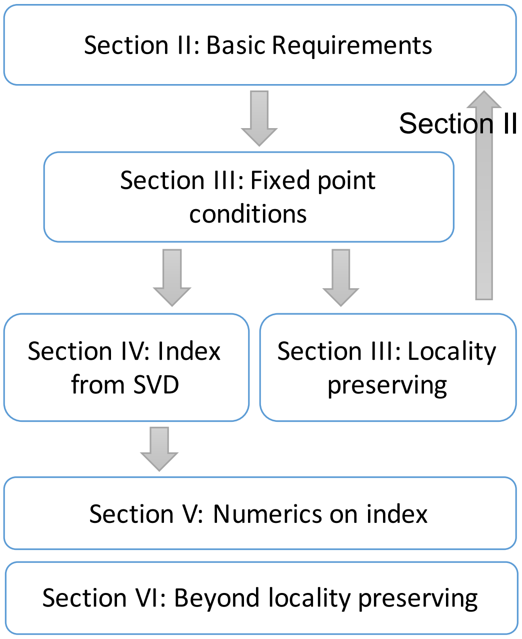

Section II: Basic Requirements
Section II
Section III: Fixed point conditions
Section IV: Index from SVD
Section III: Locality preserving
Section V: Numerics on index
Section VI: Beyond locality preserving

Figure 4.1: Structure and logic of this chapter.

4.2 Matrix Product Unitary: basic requirements

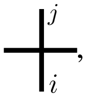
Implication of the unitary condition

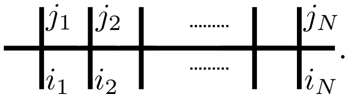
Let's first set up the stage and discuss the basic requirement a matrix product operator (MPO) has to satisfy to represent a unitary operator. Consider an MPO O acting on N sites where each site has a d-dimensional degree of freedom, i.e., O acts on $(\mathbb{C}^d)^{\otimes N}$. In principle, N is very large, ideally goes to infinity. In this chapter we focus on translation invariant MPO with periodic boundary condition. The matrix product form of O is given by

$$O^{j_1 j_2 \ldots j_N}_{i_1 i_2 \ldots i_N} = \mathrm{Tr}\left(M^{j_1 i_1} M^{j_2 i_2} \ldots M^{j_N i_N}\right) \tag{4.2}$$

where each $M^{j_k i_k}$, with fixed i_k and j_k, is a $D \times D$ matrix. $i_1 i_2 \ldots i_N$ label the input physical legs and $j_1 j_2 \ldots j_N$ label the output physical legs. We are going to call the left and right legs of the $M^{j_k i_k}$ matrices the virtual legs and think of M as a four leg tensor.

Pictorially, the local tensor M in the MPO is given by

$M =$  , (4.3) while the total MPO is given by

$O =$ 

. (4.4)

In order for O to be unitary, it has to satisfy the condition that $O^\dagger O = I$. We consider the case where this is true for any finite system size, not just in the thermal dynamic limit. We call such operators Matrix Product Unitary Operators (MPUO).

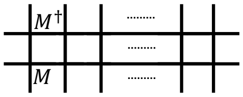
Definition 3 (Matrix Product Unitary Operator)**.** *Consider a matrix product operator O represented with tensor M of finite bond-dimension. O is called a matrix product unitary operator if it is a unitary for all system sizes.*

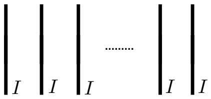
Note that we emphasize 'for all system sizes' for a good reason. In section 5.2 we are going to see that there are matrix product operators which are unitary only for certain system sizes, and hence do not fit into this definition.

If we define a new tensor $M^\dagger$ as

$$M^{\dagger ji} = \left(M^{ij}\right)^* \tag{4.5}$$

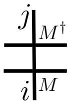
then the MPUO condition is given graphically as

$O^\dagger O =$ 

$=$ 

(4.6)

where we use a straight line to represent the identity matrix. This condition imposes very strong constraints on M. The constraint can be most easily identified on the composite of M and $M^\dagger$, which we define as

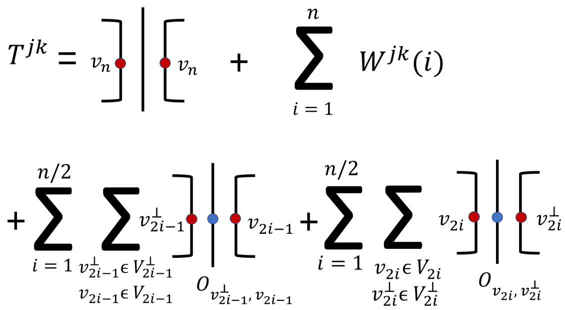
$T^{ij} = \sum_k M^{\dagger ik} \otimes M^{kj} =$ 

(4.7)

The unitarity condition Eq. 4.6 is saying that the matrix product operator with tensor T^{ij} is equivalent to a tensor product of identity operators I on each degree of freedom. If we combine the input and output physical legs of T^{ij}, we can think of it as representing a matrix product state, which would be a tensor product of maximally entangled pairs $|11\rangle + |22\rangle + ...|dd\rangle$.

Based on this observation, we can derive a general form for the T^{ij} tensors. Let's give this as a lemma:

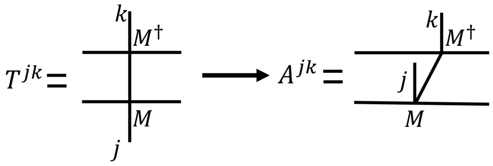
Lemma 5. *Let O be a Matrix Product Unitary Operator (MPUO) described by local tensor M, then the tensor T^{ij}, which is composed of M and $M^\dagger$ as in Eq. 4.7, has to take the following form:*


T^{jk} =
v_n
v_n
+
$\sum_{i=1}^{n}$
$W^{jk}(i)$
+
$\sum_{i=1}^{n/2}$
$\sum_{\substack{v_{2i-1}^\perp \in V_{2i-1}^\perp \\ v_{2i-1} \in V_{2i-1}}}$
$v_{2i-1}^\perp$
v_{2i-1}
$O_{v_{2i-1}^\perp, v_{2i-1}}$
+
$\sum_{i=1}^{n/2}$
$\sum_{\substack{v_{2i} \in V_{2i} \\ v_{2i}^\perp \in V_{2i}^\perp}}$
v_{2i}
$v_{2i}^\perp$
$O_{v_{2i}, v_{2i}^\perp}$

(4.8)

where n is a constant, which denotes the number of steps in the process of finding the canonical form of the associated MPS. $v_1, \ldots, v_n$ denotes vectors in the double virtual Hilbert space $V = \mathbb{C}^D \otimes \mathbb{C}^D$. Namely, each $v_i \in V_i$, $v_i^\perp \in V_i^\perp$ is an orthonormal basis vector in $V = V_n \oplus V_n^\perp \oplus V_{n-1}^\perp \oplus V_{n-2}^\perp \oplus \ldots \oplus V_1^\perp$ and $V_i = V_{i+1} \oplus V_{i+1}^\perp$ for all $0 \leq i \leq n-1$. Each $W^{jk}(i)$ denotes a block on $V_i^\perp$ which of similar form of T^{jk} except $\langle v_i^\perp | W_i^{jk} | v_i^\perp \rangle = 0$ for all j, k and all $v_i^\perp \in V_i^\perp$.

Proof. This form of the tensor T follows directly from the definition of the canonical form given in Ref.[27] and the requirement that O is an MPO which is a unitary for all system sizes. We define an MPS form for the operator $O^\dagger O$ which is described by local tensor A^{jk} obtained by combining the input and output legs, j and k, of T^{jk} as the physical legs, i.e.,

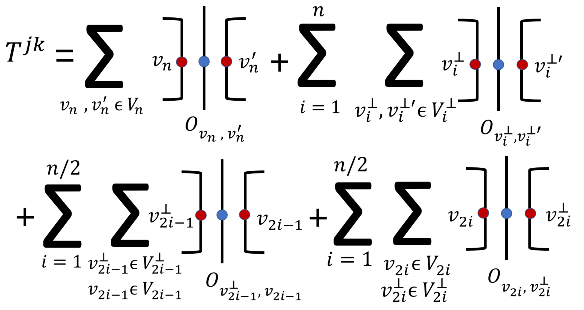


(4.9)

Following the procedure of finding the canonical form given in Ref. [27], we step by step decompose the left and the right virtual vector space of the tensors A^{jk} into orthogonal subspaces. The procedure does this alternatively: first A^{ij} gets updated to $(P_{V_1} + P_{V_1^\perp})A^{jk}$, where P_{V_1} and $P_{V_1^\perp}$ are projectors onto V_1 and $V_1^\perp$ respectively, and $P_{V_1} + P_{V_1^\perp} = P_{V_0}$ is the projector on the whole virtual space. As proved in

Ref. [27] the terms $P_{V_1} A^{jk} P_{V_1^\perp}$ vanishes. Now we update the MPS tensor A^{jk} to $P_{V_1} A^{jk} P_{V_1} + P_{V_1^\perp} A^{jk} P_{V_1} + P_{V_1^\perp} A^{jk} P_{V_1^\perp}$.

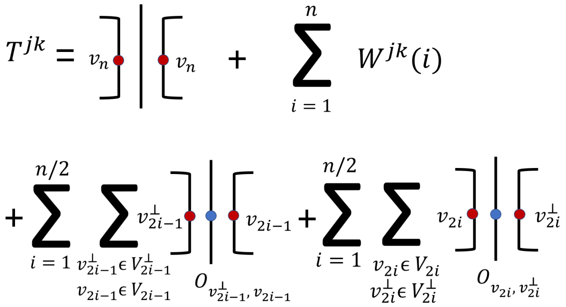
Repeating this procedure alternatively for decomposing left and right virtual vector spaces, we obtain the following general form of the tensor T^{jk}:


$T^{jk} = \sum_{v_n, v_n' \in V_n}$
v_n
v_n'
$O_{v_n, v_n'}$
$+ \sum_{i=1}^{n} \sum_{v_i^\perp, v_i^{\perp\prime} \in V_i^\perp}$
$v_i^\perp$
$v_i^{\perp\prime}$
$O_{v_i^\perp, v_i^{\perp\prime}}$
$+ \sum_{i=1}^{n/2} \sum_{v_{2i-1}^\perp \in V_{2i-1}^\perp, v_{2i-1} \in V_{2i-1}}$
$v_{2i-1}^\perp$
v_{2i-1}
$O_{v_{2i-1}^\perp, v_{2i-1}}$
$+ \sum_{i=1}^{n/2} \sum_{v_{2i} \in V_{2i}, v_{2i}^\perp \in V_{2i}^\perp}$
v_{2i}
$v_{2i}^\perp$
$O_{v_{2i}, v_{2i}^\perp}$

(4.10)

where the subspaces are split as $V = V_n \oplus V_n^\perp \oplus V_{n-1}^\perp \oplus \ldots V_1^\perp$, and $V_{i-1} = V_i \oplus V_i^\perp$ for all $1 \leq i \leq n$ and $V_0 = V = \mathbb{C}^D \otimes \mathbb{C}^D$.

Now we impose the requirement that the MPO O represented by the local tensor M is unitary for all system sizes. Since the operator O is obtained with periodic boundary conditions as seen in Eq. (4.2), we must investigate the associated MPS represented by local tensors A^{ij} with periodic boundary conditions. This means that only the operators of the form $O_{w,w'}$ with $w, w' \in V_i^\perp$ or $w, w' \in V_n$ appear in the expression of $O^\dagger O$. Since we know that $O^\dagger O = I^{\otimes N}$ for all system sizes, each of the operators $O_{ww'}$ must be individually equal to I. We can immediately see that only one block of these operators can have diagonal terms, since otherwise it would imply that $O^\dagger O$ is only proportional to $I^{\otimes N}$ and there is no way to make it exactly equal to $I^{\otimes N}$ by normalization. Let this block be the nth block that maps V_n to V_n from right to left virtual legs. This implies that in the general form of the MPS the blocks that map $V_i^\perp$ to $V_i^\perp$ can be decomposed further with the same procedure but without any diagonal term. Say all $\dim V_i^\perp = 1$ for all $i \leq n$, then we only have diagonal term in the block that maps V_n to V_n from right to left virtual legs. When one of the blocks $V_i^\perp$ is such that $\dim V_i^\perp > 1$, we have additional terms in the expression of A^{jk} that has only non diagonal terms in the the block $V_i^\perp$, which further decomposes as described above , but only within the vector space of $V_i^\perp$. We denote these terms in the sum as $W^{jk}(i)$ for each block of $V_i^\perp$. Furthermore, the fact that MPS is a product state means that $\dim V_n = 1$. Hence, Eq. (4.10) and the fact that O is an MPUO as defined in Def. 3 imply that T^{jk} is of the following form:


$T^{jk} =$
v_n
v_n
$+ \sum_{i=1}^{n} W^{jk}(i)$
$+ \sum_{i=1}^{n/2} \sum_{\substack{v_{2i-1}^{\perp} \in V_{2i-1}^{\perp} \\ v_{2i-1} \in V_{2i-1}}}$
$v_{2i-1}^{\perp}$
$O_{v_{2i-1}^{\perp}, v_{2i-1}}$
v_{2i-1}
$+ \sum_{i=1}^{n/2} \sum_{\substack{v_{2i} \in V_{2i} \\ v_{2i}^{\perp} \in V_{2i}^{\perp}}}$
v_{2i}
$O_{v_{2i}, v_{2i}^{\perp}}$
$v_{2i}^{\perp}$

(4.11)

□

Note that $n \leq D^2 - 1$ which simply follows from dimension considerations.

1D locality preserving unitaries as MPUO

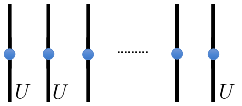
In this section, we are going to show that all locality preserving 1D unitaries can be represented as MPUO.

Let's look at a few examples first and see how their representation fits the form in Lemma 5.

- Example 1: Tensor product of unitary operators

  This is a trivial case where the dimension of the virtual legs is 1. Graphically, we denote it as

$$M_{\text{product}} =$$



(4.12)

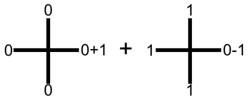
  where a line with a dot in the middle represents a nontrivial matrix, a unitary U in this case. The T tensor as defined in Eq. 4.7 is automatically identity.

- Example 2: Controlled-phase between nearest neighbor spin 1/2s.

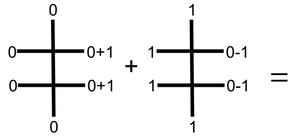
  Let's consider a simple entangled unitary in 1D $\prod_{k=1}^{N} CP_{k,k+1}$, where each $CP_{k,k+1}$ is a two body unitary of the form

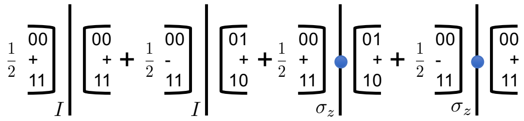
$$CP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{4.13}$$

This unitary can be represented with

$$M_{CP} =$$  (4.14)

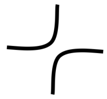
We can check that M_{CP} satisfies the condition in Lemma 5. We can calculate T_{CP} to be

$$T_{CP} =$$  (4.15)

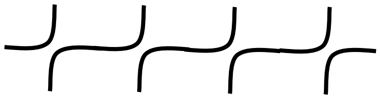


- Example 3: Translation

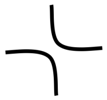
  Translation, which is a locality preserving unitary that cannot be written as a finite depth circuit, can also be represented as a MPUO in a simple way. Consider the translation to the right by one step in a spin $1/2$ chain. The operator can be represented with

$$M_r =$$  (4.16)

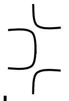
  where the curved lines again represent the identity matrix between the left and up legs, and the right and down legs. When connected into a chain, it is straight forward to see that it represents translation.

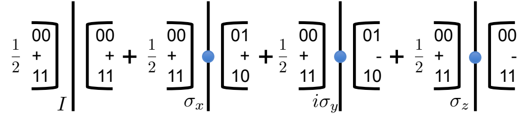
 (4.17)

  Similarly, translation to the left by one step can be represented with

$$M_l =$$  (4.18)

  M_r and M_l also satisfy the condition in Lemma 5. In particular,

$$T_r =$$  $=$ (4.19)



  And a similar expansion holds for T_l.

In fact, all locality preserving unitaries in 1D can be represented as MPUO satisfying Lemma 5.

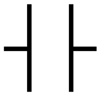
Theorem 3 (Locality preserving 1D unitaries as MPUO)**.** *Let O be a locality preserving 1D unitary. It is possible to represent it as a Matrix Product Unitary Operator, as defined in Definition 3.*

Proof. We prove this statement in the following steps:

1. Translation operator by one step can be represented with an MPO as shown with Example 3 above, such that the MPO is unitary for any system size.

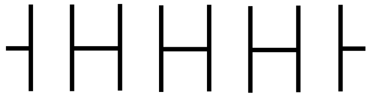
2. One layer of non-overlapping unitaries can be represented with an MPO. WLOG, consider a layer of non-overlapping two-body unitaries, which can be represented with a tensor

$$M_{tb} =$$



(4.20)

when connected together into a chain, this tensor gives the two-body unitaries.



(4.21)

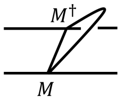
Such an MPO is unitary for all system sizes.

3. According to Ref.[1], all 1D locality preserving unitaries can be decomposed into a finite number of layers of translation and finite depth local unitary circuits which can be further decomposed into a finite number of layers of non-overlapping few-body unitaries. The MPO representation of such a composite can be obtained by stacking the MPO representation for each component. As each component satisfies the MPUO condition that the MPO is unitary for all system sizes, the same is true for the composite MPO. Therefore, all 1D locality preserving unitaries can be represented as a MPUO, with tensors satisfying Lemma 5. □

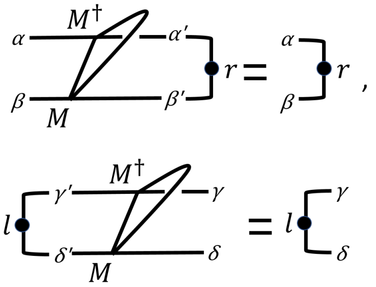
4.3 Characterization of Matrix Product Unitary Operators

In this section we prove fixed-point properties of MPUOs. Suppose that O is an MPUO described by tensor M. We show that when the individual tensors are blocked, they satisfy equations that we call *fixed-point equations*. These equations give a characterization of finite-bond dimension MPUOs. More importantly they imply that MPUOs are *locality-preserving*.

In order to obtain these results, we use basic facts about MPS[27]. So, let us first review these starting from the transfer matrix. Define the transfer matrix $\mathbb{E}_M$ of M as

$$\mathbb{E}_M = \sum_{ij} M^{ij} \otimes M^{ij*} =$$  $$= \sum_i T^{ii}, \tag{4.22}$$

and denote the right eigen-vector of $\mathbb{E}_M$ with largest eigenvalue as r and the left eigen-vector with largest eigenvalue as l, such that $\langle l|r\rangle = 1$. Assuming the spectral radius of $\mathbb{E}$ is 1, we have



(4.23)

Based on Lemma 5, we can see that if M describes an MPUO, the transfer matrix $\mathbb{E}_M$ is of the following form:

$$\begin{aligned}\mathbb{E}_M = \; & |v_n\rangle\langle v_n| + \sum_{i=1}^{n/2}\Bigg(\sum_{v_{2i},v_{2i}^\perp} tr(O_{v_{2i},v_{2i}^\perp})|v_{2i}\rangle\langle v_{2i}^\perp| \\ & + \sum_{v_{2i-1},v_{2i-1}^\perp} tr(O_{v_{2i-1}^\perp,v_{2i-1}})|v_{2i-1}^\perp\rangle\langle v_{2i-1}|\Bigg) \\ & + \sum_{i=1}^{n}\sum_{j=1}^{d} W^{jj}(i)\end{aligned} \tag{4.24}$$

Above we do not know the values of the trace of the operators, but we do know that the left and right eigen-vectors of $\mathbb{E}_M$ have to take the following form:

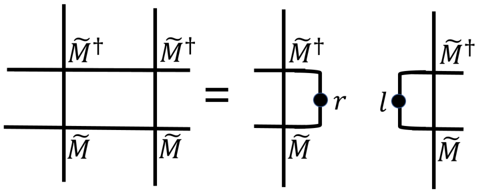
$$\begin{aligned}\langle l| \; &= \langle v_n| + \sum_{i=1}^{n/2} c_{2i-1}\langle v_{2i-1}^\perp|, \\ |r\rangle \; &= |v_n\rangle + \sum_{i=1}^{n/2} c_{2i}|v_{2i}^\perp\rangle\end{aligned} \tag{4.25}$$

where c_is are complex coefficients.

The left and right eigenvectors, when seen as matrices $r = \sum_{\alpha\beta} r_{\alpha\beta}|\alpha\rangle\langle\beta|$ with elements $r_{\alpha\beta}$ and $l = \sum_{\gamma\delta} l_{\gamma\delta}|\gamma\rangle\langle\delta|$ with elements $l_{\gamma\delta}$, are positive matrices. $\langle l|r\rangle = 1$ since $\langle v_n|v_n\rangle = 1$ and $\langle v_{2i}^{\perp}|v_{2j-1}^{\perp}\rangle = 0$ for all i, j.

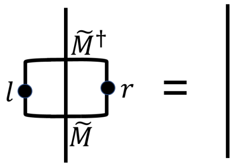
Now we are ready to state the results. We define $\tilde{M}^{JI} = M^{j_1 i_1} M^{j_2 i_2} \ldots M^{j_n i_n}$, where $I = i_1 i_2 ... i_n$, $J = j_1 j_2 ... j_n$, as the tensor obtained by blocking the individual tensor M. The blocked tensor $\tilde{M}$ satisfies the following fixed-point equations:

1. Fixed-point equation 1 - Separation:



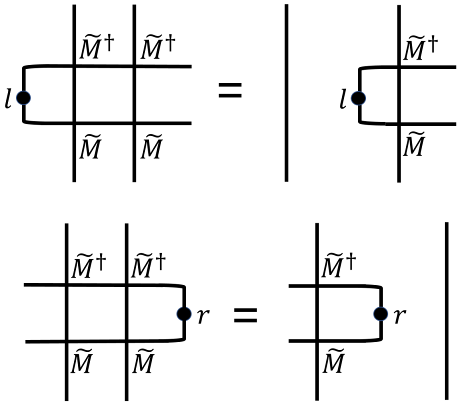
(4.26)

2. Fixed-point equation 2 - Isometry:



(4.27)

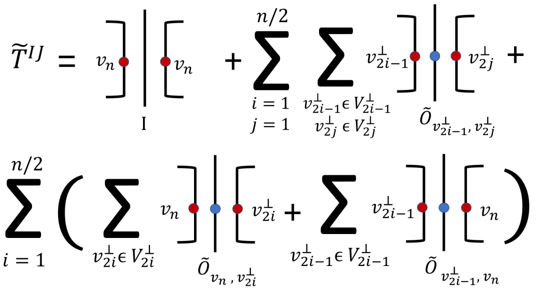
where l and r denote the left and right eigenvectors of the transfer matrix $\mathbb{E}_M$ as given in Eq. (4.25). Eq. 4.26 (separation) and Eq. (4.27) (isometry) imply the following equations called *pulling through* conditions, which we frequently make use of in the chapter.



(4.28)

Before proving the above claims, we first give a lemma that explicitly shows the form of the tensor $\tilde{T}^{IJ}$ which is obtained by blocking the tensor T^{ij} D^2-times, i.e., $\tilde{T}^{IJ} = T^{i_1 j_1} T^{i_2 j_2} \dots T^{i_{D^2} j_{D^2}}$.

Lemma 6. *Let the general form of the tensor T be as in Eq.* (4.8) *in Lemma 5. Then, the blocked tensor $\tilde{T}^{IJ} = T^{i_1 j_1} T^{i_2 j_2} \dots T^{i_{D^2} j_{D^2}}$, where D^2 is the bond dimension of the tensor T, is of the following form*


$\tilde{T}^{IJ}$ =
v_n
I
v_n
+
$\sum_{i=1, j=1}^{n/2}$
$\sum_{v_{2i-1}^\perp \in V_{2i-1}^\perp, v_{2j}^\perp \in V_{2j}^\perp}$
$v_{2i-1}^\perp$
$\tilde{O}_{v_{2i-1}^\perp, v_{2j}^\perp}$
$v_{2j}^\perp$
+
$\sum_{i=1}^{n/2}$
$\sum_{v_{2i}^\perp \in V_{2i}^\perp}$
v_n
$\tilde{O}_{v_n, v_{2i}^\perp}$
$v_{2i}^\perp$
+
$\sum_{v_{2i-1}^\perp \in V_{2i-1}^\perp}$
$v_{2i-1}^\perp$
$\tilde{O}_{v_{2i-1}^\perp, v_n}$
v_n

(4.29)

Proof. By Lemma 5, the general form of the tensor T can be taken as

$$T = |v_n\rangle I \langle v_n| + \sum_{i=1}^{n/2} \left(\sum_{v_{2i}, v_{2i}^\perp} |v_{2i}\rangle O_{v_{2i}, v_{2i}^\perp} \langle v_{2i}^\perp| \right.$$
$$\left. + \sum_{v_{2i-1}, v_{2i-1}^\perp} |v_{2i-1}^\perp\rangle O_{v_{2i-1}^\perp, v_{2i-1}} \langle v_{2i-1}| \right) + \sum_i W(i) \tag{4.30}$$

where $|v_i\rangle \in V_i$, $|v_i^\perp\rangle \in V_i^\perp$, and $V = V_n \oplus V_n^\perp \oplus V_{n-1}^\perp \oplus \dots \oplus V_1^\perp$. Now, imagine that we block D^2 of these tensors and obtain the tensor $\tilde{T}^{IJ} = T^{i_1 j_1} T^{i_2 j_2} \dots T^{i_{D^2} j_{D^2}}$. Using the facts that $\langle v_i^\perp | v_j^\perp \rangle = 0$ for all i and j, and $\langle v_i | v_j^\perp \rangle = 0$ for all $j \leq i$, it's only a matter of careful book-keeping to show that only the terms with $|v_n\rangle\langle v_n|$, $|v_{2i-1}^\perp\rangle\langle v_{2j}^\perp|$, $|v_n\rangle\langle v_{2i}^\perp|$ and $|v_{2i-1}^\perp\rangle\langle v_n|$ appear in the expression of the tensor $\tilde{T}^{IJ}$. Notice that the $W(i)$ denotes the operator components within the block $V_i^\perp$ which only has nondiagonal elements and except for the diagonal element is in the same form as T. After blocking D^2 times, the terms that come from $W(i)$ do not contract anymore either from left or right, and hence they don't appear in the tensor $\tilde{T}$. Note that the operator component with left and right indices $|v_n\rangle\langle v_n|$ acts as $I^{\otimes (D^2)}$. □

Now, we prove that the blocked tensor $\tilde{M}$ that describes the MPUO satisfies the separation and isometry fixed-point equations given above in Eq. (4.26) and Eq. (4.27).

Theorem 4 (MPUO implies fixed-point equations)**.** *Let O be an MPUO described by the tensor M. Then there exists a finite number n such that the blocked tensor $\tilde{M}$,* *which is obtained by blocking D^2 of the tensor M, satisfies the fixed point equations, i.e., Eq.*(4.26) *and Eq.*(4.27).

Proof. By Lemma 5 and Lemma 6, we know that an MPUO implies the general form for $\tilde{T}$ as in Eq. (4.29). By direct calculation the LHS of Eq. (4.26) is given as

$$\begin{aligned} |v_n\rangle I \otimes I\langle v_n| \quad & + \sum_{i,j}^{n/2} \sum_{v^{\perp}_{2i-1}, v^{\perp}_{2j}} |v^{\perp}_{2i-1}\rangle \tilde{O}_{v^{\perp}_{2i-1}, v_n} \otimes \tilde{O}_{v_n, v^{\perp}_{2j}} \langle v^{\perp}_{2j}| \\ & + \sum_{i}^{n/2} \left(\sum_{v^{\perp}_{2i}} |v_n\rangle I \otimes \tilde{O}_{v_n, v^{\perp}_{2i}} \langle v^{\perp}_{2i}| \right. \\ & \left. + \sum_{v^{\perp}_{2i-1}} |v^{\perp}_{2i-1}\rangle \tilde{O}_{v^{\perp}_{2i-1}, v_n} \otimes I \langle v_n| \right) \end{aligned} \tag{4.31}$$

which is also equal to the RHS of the same equation, considering the fact that $\langle v_n|r\rangle = \langle l|v_n\rangle = 1$ and $\langle v^{\perp}_{2i}|r\rangle = \langle l|v^{\perp}_{2i-1}\rangle = 0$ for all i, which are easily seen from the form of the left and right eigen-vectors derived in Eq. (4.25). This concludes the proof of the separation equation. Using the same facts, it is straightforward to prove the isometry condition given in Eq. (4.27). It is the following equation that follows immediately from the above facts:

$$\langle l|\tilde{T}|r\rangle = I. \tag{4.32}$$

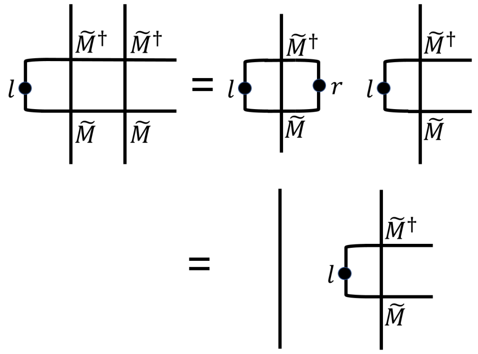
This completes the proof. As a side remark it's also straightforward to see that the isometry equation (4.27) is true even before blocking, i.e., $\langle l|T|r\rangle = I$. □

Theorem 4 gives a characterization of MPUOs O by tensors $\tilde{M}$ that satisfies the fixed-point equations, i.e., Eqs.(4.26) and (4.27).

Another consequence of the fixed-point equations is what we call the pulling through equations, which is given as a corollary as follows.

Corollary 3. *The fixed point equations, i.e., Eq.* (4.26)*, and Eq.* (4.27)*, imply the pulling through equations, i.e., Eq.* (4.28).

Proof. We start with the LHS of the pulling through equation, i.e., Eq. (4.28). We apply the fixed-point equations, namely separation, i.e., Eq. (4.26) and then apply the isometry, i.e., Eq. (4.27), respectively. Pictorially, it follows as below.



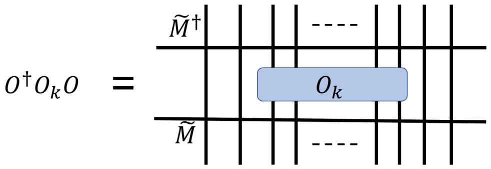
(4.33)

The other pulling through equation from right to left follows from separation and isometry fixed-point equations in the same way. □

Finally we close this section by showing that all finite-bond dimension MPUOs are locality-preserving. It means that, it maps any geometrically k-local operator to a geometrically $(k + c)$-local operator, where c is a constant independent of the system size. This is proven in the following corollary.

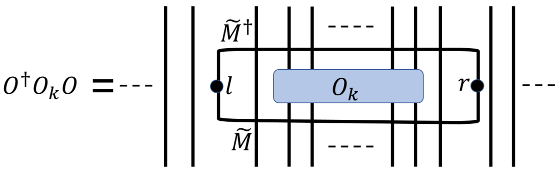
Corollary 4 (MPUOs are locality-preserving)**.** *Every MPUO is locality preserving, namely they map geometrically k-local operators to geometrically at most $(k + c)$-local operators where c is a constant independent of the system size.*

Proof. An MPUO O acts on an operator O_k as $O : O_k \rightarrow O^\dagger O_k O$.Pictorially it is shown by



(4.34)

Using fixed-point equations, it is straightforward to see that



(4.35)

is a $(k+2)$-local operator. Hence after blocking sites, MPUOs map k-local operators to at most $(k+2)$-local operators. This means that before blocking, a k-local operator is mapped to at most a $(k + 2D^2)$-local operator, since we are guaranteed to reach the fixed point after blocking D^2 sites. □

4.4 Extracting GNVW index from MPUO representation

Review of GNVW index

In Ref.[1], Gross, Nesme, Vogts and Werner proved that 1D locality preserving unitaries (called cellular automata in that paper) can be classified according to how much information is flowing across a cut in the chain. For example, finite depth local unitary circuits – a finite number of layers of local unitaries where unitaries within each layer do not overlap with each other – all belong to one class and there is zero information flow. On the other hand, translation by one step in a spin $1/2$ chain belongs to another class and there is a flow of a single spin $1/2$ across any cut.

More specifically, Ref.[1] defined two 1D locality preserving untiaries to be equivalent to each other if and only if they differ from each other by a finite depth local unitary circuit and showed that every 1D locality preserving unitary is then equivalent to some translation operation. Each equivalence class is characterized by an index (the GNVW index) which measures how much translation is taking place: if there is a translation of p dimensional Hilbert space by m steps to the right, the index is p^m; if there is a translation of q dimensional Hilbert space by n steps to the left, the index is $1/q^n$; if there is translation in both directions, the index is p^m/q^n. Such an index is consistent with the equivalence class structure of locality preserving unitaries because it was shown that when two locality preserving operators multiply, their GNVW index also multiply

$$I_{\text{GNVW}}(O_1O_2) = I_{\text{GNVW}}(O_1)I_{\text{GNVW}}(O_2) \tag{4.36}$$

For 1D locality preserving unitaries, the index is always a positive rational number and can be calculated as

$$I_{\text{GNVW}}(O) := \frac{\eta(O\mathcal{A}_L O^\dagger, \mathcal{A}_R)}{\eta(\mathcal{A}_L, O\mathcal{A}_R O^\dagger)} \tag{4.37}$$

where $\mathcal{A}_L$ is the set of operators within distance l_0 on the left hand side of a cut and $\mathcal{A}_R$ is the set of operators within distance l_0 on the right hand side of the cut. $\eta(\mathcal{A}, \mathcal{B})$ measures the overlap between the two sets of operators and is defined as

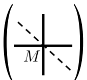
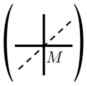
$$\eta(\mathcal{A}, \mathcal{B}) := \frac{\sqrt{p_a p_b}}{p_\Lambda} \sqrt{\sum_{i,j=1}^{p_a} \sum_{l,m=1}^{p_b} \left| Tr_\Lambda \left(\hat{e}_{ij}^{a\dagger} \hat{e}_{lm}^b \right) \right|^2} \tag{4.38}$$

where $\hat{e}_{ij}^a$ is the set of basis operators in $\mathcal{A}$ and there are p_a of them; $\hat{e}_{lm}^b$ is the set of basis operators in $\mathcal{B}$ and there are p_b of them; Λ is a segment in the chain containing both a and b. The GNVW index defined in this way converges to the positive rational number characterizing information flow when l_0 becomes large.

Rank-ratio index = (**GNVW index**)^2

How does one extract the GNVW index from the matrix product representation of the locality preserving unitary operators? In this section, we show that it can be extracted as the square root of the Rank-Ratio index, which is defined as the ratio between the rank of the left and right SVD decompositions of the tensor M in the representation.

Definition 4 (Rank-Ratio Index)**.** *Let M be the tensor in the matrix product representation of a unitary operator with physical legs in the up and down directions and virtual legs in the left and right directions. The Rank-Ratio Index is defined as the ratio between the rank of the SVD decomposition between left,down–right,up legs and the rank of the SVD decomposition between left,up–right,down legs. Graphically, the Rank-Ratio Index is given by*

$I_{RR}(M) = rank\Bigg($  $\Bigg) \Bigg/ rank\Bigg($  $\Bigg)$ (4.39)

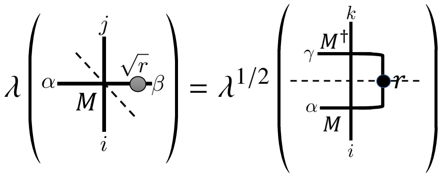
To demonstrate the connection between the Rank-Ratio index defined above and the GNVW index in Ref. [1], first we need to define the injectivity condition for matrix product operators. This definition is the same as the definition of injectivity as given in Ref. [27] if we combine the input and output physical legs of the MPO tensor and treat it as a matrix product state. We state this condition in detail below for subsequent discussions.

Definition 5 (Injective matrix product operator)**.** *Consider a matrix product operator given by a set of matrices $\{M^{ij}\}$, where $i, j = 1, .., d$ label the input and output physical legs. The MPO is called injective if $r_{\alpha\beta}$ and $l_{\gamma\delta}$ defined in Eq.* (4.23) *are full rank matrices with row and column indices α, β and γ, δ respectively.*

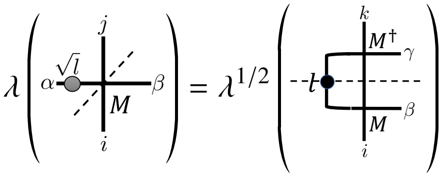
The notion of injectivity is relevant to our discussion of MPUO because if M represents an MPUO, then it can always be put into an injective form by removing redundant virtual leg dimensions. This can be shown by noticing that, if M cannot be put into an injective form by removing redundant virtual dimensions, then each M^{ij} contains at least two blocks in their canonical form. Then correspondingly T^{ij} contains at least two blocks in its canonical form, which is not possible if it represents identity for all system sizes, as we argued below Eq. (4.10).

Moreover, we need the following lemma.

Lemma 7. *Consider an MPO represented by an injective tensor M. Then*



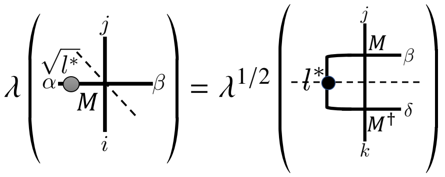
(4.40)



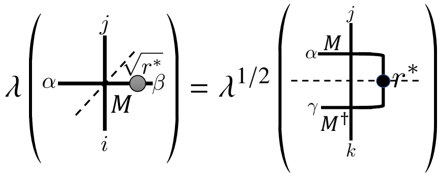
(4.41)

where the dashed lines denote SVD decompositions across the cut, λ denotes the set of singular values of the decomposition, and the square root on λ is taken element-wise. l and r are the left and right eigenvectors of the transfer matrix $\mathbb{E}_M$ as defined in Eq. (4.23)*. l and r are denoted with black dots and their square roots are denoted with grey dots.*

Similarly, we have



(4.42)



(4.43)

where l^* is the complex conjugation of l and r^* is the complex conjugation of r.

Note that as singular values are non-negative, there is no ambiguity in taking the square root. Moreover, as M is injective, l and r have full rank and have a well defined square root.

Proof. We are going to prove Eq. (4.40) and then the proof of Eq. (4.41), (4.42), and (4.43) will follow in a similar way.

Consider the SVD decomposition on the left hand side of Eq. 4.40 and suppose it takes the form

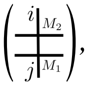
$$\sum_{\beta'} M_{i\alpha,j\beta'} \sqrt{r}_{\beta',\beta} = \sum_{s} U_{i\alpha,s} \lambda_s V_{s,j\beta} \tag{4.44}$$

Then the tensor on the right hand side of Eq. 4.40 becomes

$$\begin{aligned} & \textstyle\sum_{\beta',\delta',j} M_{i\alpha,j\beta'} r_{\beta',\delta'} M^{\dagger}_{j\delta',k\gamma} \\ = & \textstyle\sum_{j,s,s'} U_{i\alpha,s}\lambda_s V_{s,j\beta} V^{\dagger}_{s',j\beta}\lambda_{s'} U^{\dagger}_{k\gamma,s'} \\ = & \textstyle\sum_{s} U_{i\alpha,s}\lambda_s^2 U^{\dagger}_{k\gamma,s} \end{aligned} \tag{4.45}$$

Therefore, the singular value for the tensor on the right hand side is the square of the singular value on the left hand side. Hence we get Eq. 4.40. □

The Rank-Ratio Index defined above can be directly related to the GNVW index if the MPUO is either injective or a stack of injective MPUOs.

Theorem 5 (Rank-Ratio index = (GNVW index)^2 for injective or stack of injective MPUO)**.** *Consider an MPUO O represented with tensor M. Take a sufficiently long but finite block so that the blocked tensor $\tilde{M}$ satisfies the Separation, Isometry and Pulling Through conditions in Eq.* (4.26)*,* (4.27) *and* (4.28)*. If M is injective, or a stack of several injective tensors as*  *, then*

$$I_{RR}(\tilde{M}) = (I_{GNVW}(O))^2 \tag{4.46}$$

We are going to proceed to prove theorem 5 in the following steps:

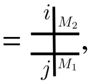
1. For an injective MPUO representation of non-overlapping two-body unitaries,

$$I_{\mathrm{RR}}(\tilde{M}) = 1 = I^2_{\mathrm{GNVW}}(O). \tag{4.47}$$

2. For an injective MPUO representation of translation (to the right) by one step,

$$I_{\mathrm{RR}}(\tilde{M}) = d^2 = I^2_{\mathrm{GNVW}}(O). \tag{4.48}$$

   where d is the dimension of the local physical Hilbert space.

3. If we stack two injective MPUOs as $M_{12} =$ , then

$$I_{\mathrm{RR}}(\tilde{M}_{12}) = I_{\mathrm{RR}}(\tilde{M}_1) I_{\mathrm{RR}}(\tilde{M}_2). \tag{4.49}$$

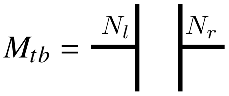
   According to Ref.[1], any locality preserving unitary can be obtained by stacking translation and layers of non-overlapping few body unitaries and their GNVW index multiply when stacked. Therefore, using the above equations we can show that the Rank-Ratio index of the stacked tensor is the square of the GNVW index.

4. On the other hand, the stacked M_{12} may not be injective itself but can be made injective. We will show that its Rank-Ratio index does not change even if we reduce it to the injective form.

5. Finally, we show that the Rank Ratio index is stable in that if $\tilde{M}$ is the fixed point form (which satisfies Eq. (4.26), (4.27), and (4.28)) of an injective tensor or a stack of injective tensors, then the Rank-Ratio index does not change if we keep blocking $\tilde{M}$.

Proof. Let's follow the procedure listed above.

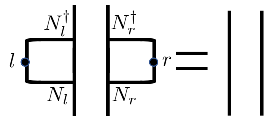
1. Consider the tensor given in Eq. 4.20 to represent non-overlapping two-body unitaries.



(4.50)

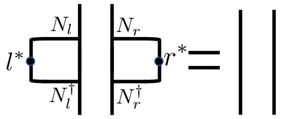
where we have labeled the left and right part of the tensor N_l and N_r respectively. As this representation can be obtained by decomposing each two-body unitary into a matrix product form, we can always choose M_{tb} to be injective.

According to the isometry condition in Eq. 4.27, which is true even before blocking, we have



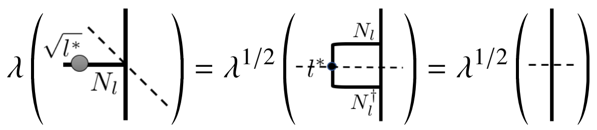
(4.51)

and similarly



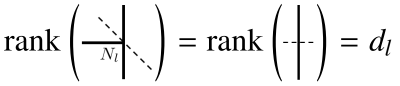
(4.52)

Each of these two equations actually contains two parts: the left halves on the two sides are equal to each other and right halves on the two sides are equal to each other. Both halves have to be satisfied simultaneously. Then using Eq. 4.40, we have



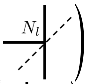
(4.53)

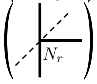
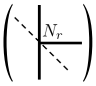
As $\sqrt{l^*}$ is a positive matrix, applying it does not change the rank of the SVD decomposition, so we have



(4.54)

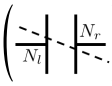
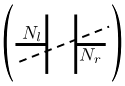
where d_l is the dimension of the physical index in N_l. Similarly, we have

rank $\Bigg($  $\Bigg) = d_l,$

rank $\Bigg($  $\Bigg) =$ rank $\Bigg($  $\Bigg) = d_r.$ (4.55)

where d_r is the dimension of the physical index in N_r. Now if we calculate the Rank-Ratio index for M_{tb}, we find that

$I_{RR}(M_{tb})$

$=$ rank $\Bigg($  $\Bigg) \Big/$ rank $\Bigg($  $\Bigg)$ (4.56)

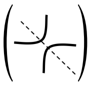
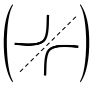
$= (d_l d_r)/(d_l d_r) = 1 = I^2_{\text{GNVW}}(O_{tb})$

Moreover, since the tensor M_{tb} already satisfies the separation, isometry, pulling-through conditions in Eq. (4.26), (4.27), and (4.28), we have

$$I_{\text{RR}}(\tilde{M}_{tb}) = 1 = I^2_{\text{GNVW}}(O_{tb}). \tag{4.57}$$

2. For translation operator, the relation between the Rank-Ratio index and the GNVW index can be found through direct calculation. Consider translation by one step to the right represented by M_r in Eq. 4.16.

$I_{RR}(M_r)$

$=$ rank $\Bigg($  $\Bigg) \Big/$ rank $\Bigg($  $\Bigg)$ (4.58)

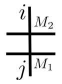
$= d^2/1 = d^2 = I^2_{\text{GNVW}}(O_r)$

Since the tensor M_r already satisfies the fixed point conditions, we have

$$I_{\text{RR}}(\tilde{M}_r) = d^2 = I^2_{\text{GNVW}}(O_r). \tag{4.59}$$

Moreover, even though we have only checked this relation for one possible representation of the translation operator, it holds for all possible injective representations as they differ from each other at most by a basis transformation on the virtual legs[27].

3. Now let us stack two layers of MPUOs which are injective individually. The composite tensor

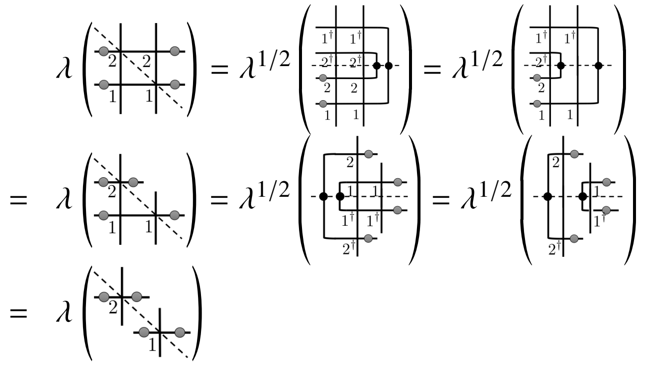
$M_{12} =$  (4.60)

is in general not injective. But we will show that its Rank-Ratio index is still the square of the GNVW index of the corresponding unitary operator.

Let's assume that M_1 and M_2 are already at fixed point form satisfying the separation, isometry, pulling through conditions Eq. 4.26, 4.27, 4.28. M_{12} is in general not in a fixed point form, but by blocking sites we can take it to a fixed point form. Suppose that the fixed point for M_{12} can be achieved by blocking two sites. (Our proof below also works if we take larger blocks.) Now we are going to use Eq. 4.40 through 4.43 in Lemma 7 to prove that

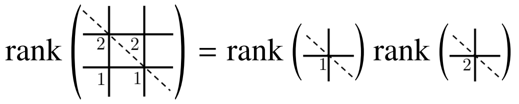
$$I_{\mathrm{RR}}(\tilde{M}_{12}) = I_{\mathrm{RR}}(\tilde{M}_1) I_{\mathrm{RR}}(\tilde{M}_2). \tag{4.61}$$

To see this, we find that



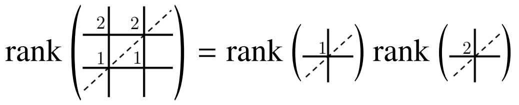
(4.62)

where we have used simplified notation 1, 2, $1^\dagger$, $2^\dagger$ to refer to M_1, M_2, $M_1^\dagger$, and $M_2^\dagger$. The black dots represent the left and right eigenvectors of the transfer matrices of M_1, M_2, $M_1^\dagger$, and $M_2^\dagger$ while the grey dots are the square root of the black dots. As long as M_1, M_2 are injective (so are $M_1^\dagger$ and $M_2^\dagger$), the grey dots do not change the rank of the SVD decomposition. Therefore we have



(4.63)

Similarly, we have



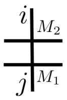
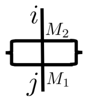
(4.64)

Dividing these two equations, we get as promised

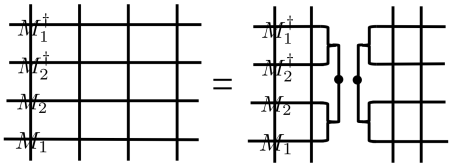
$$I_{\mathrm{RR}}(\tilde{M}_{12}) = I_{\mathrm{RR}}(\tilde{M}_1) I_{\mathrm{RR}}(\tilde{M}_2). \tag{4.65}$$

4. M_{12} as a stack of M_1 and M_2 may not be injective itself. But as we show below, the Rank-Ratio index does not change if we reduce it to the injective form. Suppose

that to reduce M_{12} to the injective form and remove redundant virtual dimensions, we need to do a projection P to the pair of virtual legs in each direction, as denoted by the { } in the following equation:

$M_{12} =$  $\rightarrow \mathcal{M}_{12} =$  (4.66)

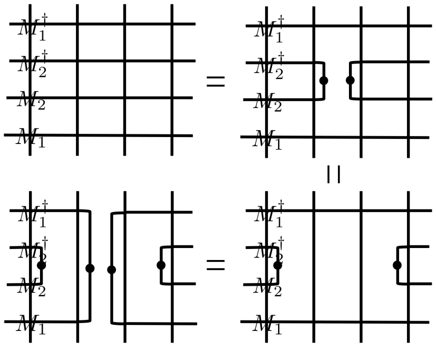
The separation condition on M_{12} reads



(4.67)

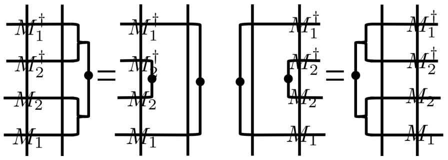
The left and right eigen-vectors (the black dots) that we insert in the middle are supported on P, so we are free to add those projections. (Note that the separation condition holds even if the MPO is not injective.) The tensors in each row are the same and we have labeled only one of them.

On the other hand, we have



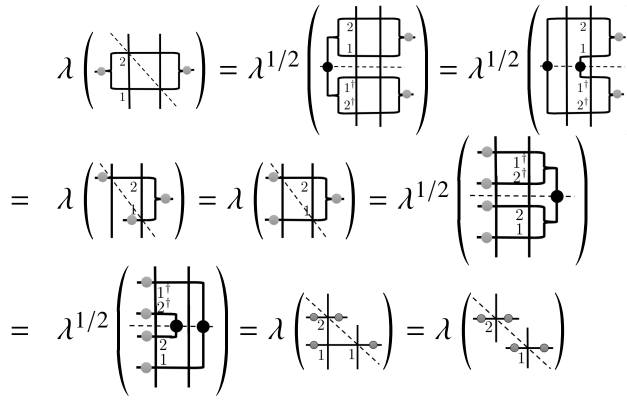
(4.68)

where the first step uses the separation condition for M_2, the second step uses the pulling through condition for M_2, and the third step uses the separation condition for M_1. Comparing Eq. 4.67 and 4.68, we find that



(4.69)

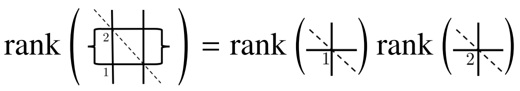
And a similar relation holds if we switch the place of M_1M_2 and $M_2^\dagger M_1^\dagger$. From these relations we find that



(4.70)

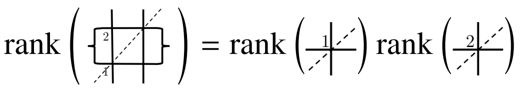
In this equation, step 1, 3, 5, 7 uses Lemma 7, step 2, 6 uses Eq. 4.69 (or similar), step 4, 8 uses derivations similar to that in Eq. 4.62. In particular, in step 4 adding the projection P does not affect the relation as the two tensors before adding the projection are related by a unitary on the left and down legs.

Therefore, we get



(4.71)

Similarly, we have



(4.72)

Dividing these two equations we get

$$I_{\mathrm{RR}}(\tilde{\mathcal{M}}_{12}) = I_{\mathrm{RR}}(\tilde{M}_1)I_{\mathrm{RR}}(\tilde{M}_2) = I_{\mathrm{RR}}(\tilde{M}_{12}). \tag{4.73}$$

Therefore, the Rank-Ratio index of the stack MPUO M_{12} remains the same whether we reduce it to the injective form or not and we always have

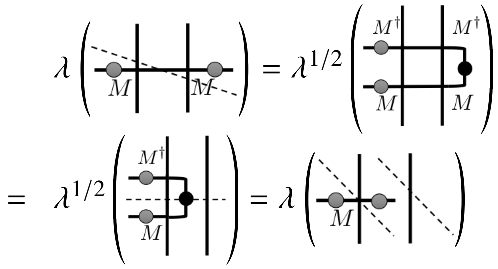
$$I_{\mathrm{RR}}(\tilde{M}_{12}) = I_{\mathrm{RR}}(\tilde{M}_1)I_{\mathrm{RR}}(\tilde{M}_2). \tag{4.74}$$

This property of the Rank-Ratio index is the same as that of the GNVW index which multiply when we combine two locality preserving unitaries (Eq. 4.36). As in each of the injective layers (either representing local unitary or translation) the Rank-Ratio index is equal to the square of the GNVW index, when we stack the layers, the Rank-Ratio index is still equal to the square of the GNVW index. Therefore, for injective or stack of injective MPUO representations, we always have

$$I_{\mathrm{RR}}(\tilde{M}) = (I_{\mathrm{GNVW}}(O))^2 \tag{4.75}$$

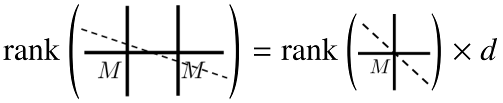
5. Finally, we need to show that our definition of Rank-Ratio index is stable. That is, it does not change if we keep blocking the tensor M once it has reached the fixed point form. This is true for both injective and stack of injective tensors.

Suppose that M is at the fixed point form satisfying the separation, isometry, pulling through conditions in Eq. 4.26, 4.27, 4.28. Then we have



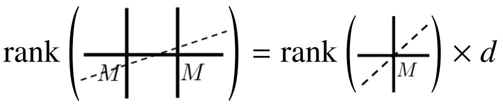
(4.76)

Therefore, we have



(4.77)

Similarly we have



(4.78)

Dividing these two equations we find that the Rank-Ratio index does not change if we block tensors at the fixed point. Note that when M is a stack of injective tensors, the grey dots in the previous equations actually correspond to several grey dots, one on each injective virtual leg.

With these steps, we complete the proof of Theorem 5. Note that, as our proof relies on Lemma 7 which is about the spectrum of the SVD decomposition, so in principle we can define our index as the ratio of the exponential of the entropy of the left and right SVD decompositions. The only tricky part is that we need to add the grey dots, the square root of the left and right eigenvectors of the transfer matrices, to the virtual legs for the index to work. This is doable but procedural-wise complicated. Therefore, we choose to define the index using the rank, instead of the entropy, of the SVD decomposition. □

4.5 Numerical calculation of index for random MPUO

In this section we are going to calculate the rank-ratio index of some examples of random MPUO. The examples of random MPUO considered are drawn in Fig. 4.2, and the corresponding numerical results are given in Tables 4.1, 4.2, and 4.3 respectively.

To generate random k-body unitaries we use the QR-decomposition of random matrices. The algorithm is as follows:

1. Generate d^k dimensional random matrix $M_{d^k \times d^k}$. d is the dimension of the physical Hilbert space at each site.
2. Perform a QR-decomposition: $M = QR$. Q is a d^k dimensional unitary while R is an upper triangular matrix.
3. The Q and R are not unique since for any d^k dimensional unitary diagonal matrix Λ, $QR = (Q\Lambda)(\Lambda^{-1}R)$. To fix this, we demand that R has positive diagonal entries. This fixes Λ to be identity. If $R = \sum_{ij} r_{ij} |i\rangle\langle j|$, create a diagonal matrix $\Lambda' = \sum_i \frac{r_{ii}}{|r_{ii}|} |i\rangle\langle i|$, and $Q' = Q\Lambda'$. Now for every random matrix M, Q' is a unique d^k dimensional unitary.

From these examples, we can see that

- The Rank-Ratio index fluctuates for small block sizes but saturates to a fixed value for large enough block sizes;
- The saturated value is equal to the square of the GNVW index and only depends on the equivalence class of the MPUO, which is invariant under stacking with any finite depth local unitary operation.

4.6 A non-locality-preserving MPO: fractional GNVW index

In the previous section, we have discussed how matrix product operators satisfying a simple unitary condition (Definition 3 and Eq. 4.6) provides a necessary and sufficient representation of locality preserving unitaries classified by the GNVW index. On the other hand, if we relax the condition in Eq. 4.6, we can obtain matrix product operators, which are unitary in a more general sense, with index beyond the GNVW framework. In this section, we are going to give one example of such matrix product operators. We are going to show that this operator is unitary in systems of odd size and non-unitary in systems of even size. It does not preserve locality and can have a ‘fractional’ index!

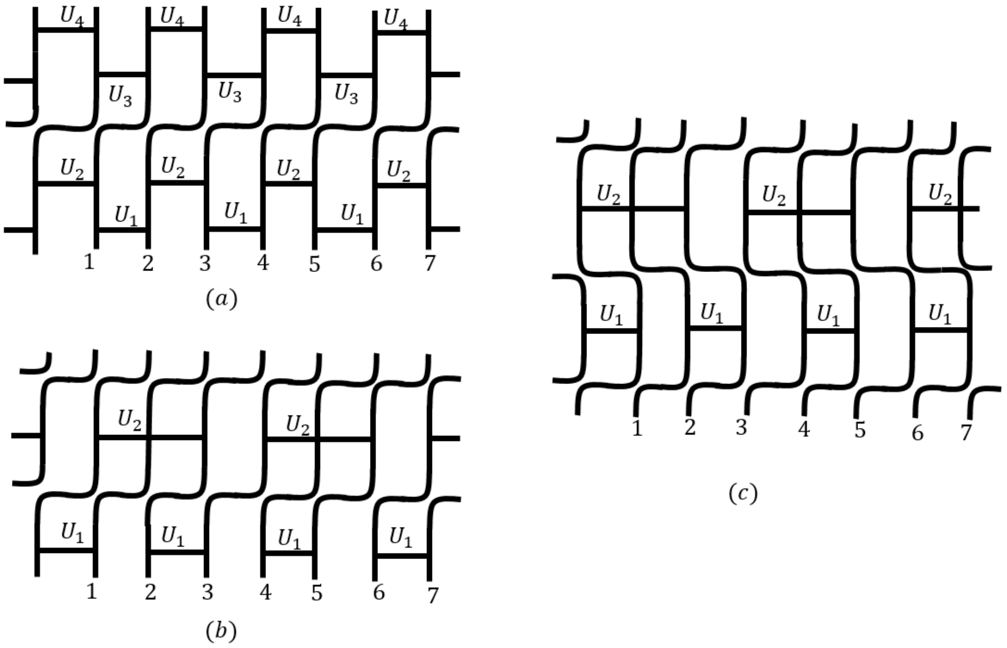


Figure 4.2: Some examples of random MPUOs. Local physical Hilbert space has dimension $d = 2$ in all cases. (a) We combine a single right-translation operator with random finite depth local unitary operators. U_1, U_2, U_3, and U_4 are all random 2-local unitaries, (b) we combine layers of random local unitaries with layers of right-translation. First layer is made of 2-local random unitary U_1, second layer is right-translation, third layer is 3-local random unitary and fourth layer is again a right translation operator. (c) Finally as an example of the most general case we combine random local unitary operators with left and right translational operators. The first layer is right-translation, the second layer is random 2-local unitaries, the third layer is left-translation, the fourth layer is random 3-local untaries and final layer is right-translation again. Numerical calculation of RR indices of MPUOs in (a),(b), and (c) are given in tables 4.1, 4.2, and 4.3 respectively.

Consider the MPO O_f represented with local tensor

$$M_f = \sum_{\substack{a,b,c=1,2,3\\ a\neq b, b\neq c, c\neq a}} a\overset{a}{\underset{a}{\Large +}}a + b\overset{c}{\underset{a}{\Large +}}a \tag{4.79}$$

This is a special MPO in that it represents a unitary operator when system size is odd and a non-unitary operator when system size is even. For example, when the system size is two, the operator maps both input states $|12\rangle$ and $|21\rangle$ to $|33\rangle$. Similar non-unitary mappings exist whenever the system size is even. This is different from all the other examples we discussed in this chapter, which are unitary and satisfy Eq. 4.6 for all system sizes. (And this operator does not satisfy Eq. 4.6 even

Length of blocked MPO	rank of left SVD	rank of right SVD	RR index
1	64	16	4
2	8	8	1
3	16	4	4
4	32	8	4
5	64	16	4
6	128	32	4
7	256	64	4

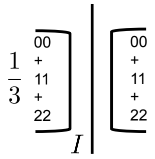
Table 4.1: Numerical calculation of RR index of MPUO shown in Fig. 4.2(a). We start with site labeled 1 and block sites one by one to the right. We see that after blocking 3 sites the index stabilizes to value 4, which is expected since this MPUO is, by construction, equivalent (up to finite depth local untiaries) to a pure right-translation and hence has index $I_{\text{RR}}(M_r) = 2^2 = 4$.

Length of blocked MPO	rank of left SVD	rank of right SVD	RR index
1	8	8	1
2	16	4	4
3	32	2	16
4	64	4	16
5	128	8	16
6	256	16	16
7	512	32	16

Table 4.2: Numerical calculation of RR index of MPUO shown in Fig. 4.2(b). We start with site labeled 1 and block sites one by one to the right. We see that after blocking 3 sites index stabilizes to value 16, which is expected since this MPUO is, by construction, equivalent (up to finite depth local untiaries) to the combination of two pure right-translation and hence has total index $I_{\text{RR}}(M_r)^2 = 4^2 = 16$.

after blocking.) Therefore, it does not belong to the set of MPUO as defined in Definition 3.

To understand the property of this MPO, we can construct T_f according to Eq. 4.7 and, from its general form, identify the operator $O_f^\dagger O_f$. The general form of T_f, which we calculate using the procedure in Ref.[27], contains two blocks. The first block is what we would expect if O is a unitary for all system sizes



(4.80)

Different from a usual unitary MPO, there is a second block, which represents the

Length of blocked MPO	rank of left SVD	rank of right SVD	RR index
1	16	4	4
2	32	8	4
3	64	4	16
4	32	8	4
5	64	16	4
6	128	32	4
7	512	128	4

Table 4.3: Numerical calculation of RR index of MPUO shown in Fig. 4.2(c). We start with site labeled 1 and block sites one by one to the right. We see that after blocking 3 sites index stabilizes to value 4, which is expected since this MPUO is, by construction, equivalent (up to finite depth local untiaries) to the combination of two pure right-translation and one left-translation, and hence has total index $I_{\mathrm{RR}}(M_r)I_{\mathrm{RR}}(M_l)I_{\mathrm{RR}}(M_r) = 4.\frac{1}{4}.4 = 4$.

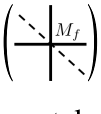
superposition of two translation symmetry breaking operators. The two operators each have period 2 and they map into each other under a single step of translation. Therefore, this part of the MPO is zero when the system size is odd, leaving the MPO O_f to be unitary. When the system size is even, the second block gives rise to a nontrivial operator, which breaks the unitarity of O_f.

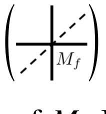
When the system size is odd $(2n+1)$, O_f is a unitary operator, but it is a highly non-locality preserving. To see this, consider the operator $P_n = |1\rangle\langle 2| + |2\rangle\langle 3| + |3\rangle\langle 1|$ on the nth qutrit and the conjugation of P_n by O_f. Applying $O_f^\dagger P_n O_f$ on an initial state $|11...1...11\rangle$, we find that the state is mapped to

$$\begin{aligned} & |11...1...11\rangle \xrightarrow{O_f} |11...1...11\rangle \\ \xrightarrow{P_n} \; & |11...3...11\rangle \xrightarrow{O_f^\dagger} |32...a...32\rangle \end{aligned} \qquad (4.81)$$

where $a = 3$ if n is odd and $a = 1$ if n is even. As the final state $|32...a...32\rangle$ is globally different from the initial state $|11...1...11\rangle$, $O_f^\dagger P_n O_f$ has to be a nonlocal operator even tough P_n is local. Therefore, O_f is a non-locality-preserving unitary when system size is odd.

Interestingly, if we calculate the index of M_f according to Eq. 4.39, we find that

$I_{\mathrm{RR}}(O_f) = \mathrm{rank}$ 

$\Big/ \mathrm{rank}$ 

$= 3$ (4.82)

and this number stays invariant if we take blocks of M_f.If we were to convert it to the GNVW index, we would find it to be $\sqrt{3}$ which is not a rational number and hence not allowed as a GNVW index. This is of course expected because O_f is not a locality preserving unitary and this example illustrates that it is possible to represent some non-locality-preserving unitaries with drastically different properties from the locality-preserving ones using the matrix product operator formalism. We would expand more on these kinds of MPOs in the next chapter.

4.7 Summary and outlook

In this chapter we studied the representation of one dimensional locality preserving unitaries using the matrix product operator (MPO) formalism. We show that matrix product operators, which are unitary (for all system sizes), are guaranteed to preserve locality and all locality preserving unitaries can be represented in a matrix product way. Moreover, we show that the GNVW index[1] classifying locality preserving unitaries in 1D can be extracted in a simple way as in Eq. 4.39 for injective or a stack of injective tensors. On the other hand, matrix product operators satisfying a more general unitarity condition – unitary only for systems of certain sizes – can have very different properties. In particular, we present one example of MPO which is unitary for odd size systems but not for even size systems and find that it is non-locality preserving and has a fractional index as compared to the locality preserving ones.

Many interesting questions remain open regarding the matrix product representation of unitaries. First of all, Lemma 5 provides a complete characterization of MPOs which are unitary for any system size. However, this characterization is in terms of T rather than M. In particular, if one wants to simulate a unitary evolution process using finite bond dimension MPO, it is not clear which parameter space one should choose from such that the MPO is guaranteed to be unitary. If such a parameter space can be identified, we can generate 1D unitaries without having to check the condition on the T tensor. With the matrix product representation of states, we do not need to worry about this problem because any tensor generates a legitimate quantum state. This is essential for variational algorithms based on matrix product states. If we want to have similar simulation algorithms for unitary dynamics with matrix product operator, this problem needs to be addressed.

Secondly, adding symmetry requirement to the 1D unitary operators can result in more detailed classifications. This has been discussed in terms of (dynamical) interacting Floquet phases with symmetry where a classification in 1D has been proposed in Ref.[65–69]. Similar to the case of 1D gapped (nondynamical) phases, adding symmetry can result in symmetry-protected Floquet phases. It would be interesting to see how to distinguish different symmetry protected Floquet phases based on the MPO representation of their Floquet operator.

Finally, the example we discussed in section 5.2 shows that if we relax the definition of unitarity, MPO can represent non-locality-preserving unitaries with fractional index. What is the full power of MPO in representing 1D unitaries in this more general sense? For matrix product state, we know that with a translation invariant finite bond dimension representation, the state represented is either gapped or a superposition of several gapped states. Can we obtain a similar understanding of the MPO representation of 1D unitaries? This is a question we plan to study in the future.

Chapter 5

TN REPRESENTATION BEYOND LOCALITY-PRESERVING 1D UNITARY OPERATORS

5.1 Background and Motivation

In the last chapter we developed the tensor representation of locality-preserving 1D unitary operators. We observed that such MPUs are generated by a local-tensor with certain general form. Namely, its $T = \sum_j M^{ij}(M^{j,k})*$ has to have only one block of rank one in its canonical form (Eq. 5). To show that such MPUs can represent all locality preserving 1D unitaries, we proved that they can represent the basic building blocks of such unitaries, and hence can be stacked to construct any 1D locality-preserving unitary. It was shown that if we block such local-tensors enough times, they satisfy certain fixed point equations, namely, separation (Eq. (4.26)) and isometry (Eq. 4.27). We proved that it follows from the fixed point equations that these MPUs are bound to be locality-preserving. That is, they map local operators to local operators. To characterize such MPUs, we defined an index called Rank-Ratio(RR) index as the ratio of ranks between left SVD and right SVD decomposition of the local-tensor (definition 4) and showed (theorem 5), using fixed point equations, that this index satisfies all the required properties for it to be an equivalent of GNVW index defined in Ref. [1]. Finally, in section 5.2 we noted an example where the local-tensor generated unitaries but not for all system sizes. It had a periodicity of 2, that is, it was only unitary for odd system sizes. The present chapter is inspired by and is an expansion by this example.

We start with noting that everything we proved about the locality-preserving MPUs followed from the following implicite assumption (or requirement): *there is a 4-leg local tensor that generates MPOs on periodic N-site 1D chains and we rquire this MPO to be unitary for all system sizes, N.* This was the justification in assuming that T has only one block of rank in its canonical form. But the reason for this assumption is not clear apriori. We often work with certain range of system sizes. Why should we care whether the MPU at hand is generated by a local-tensor that generates unitaries for other system sizes as well? Though this assumption gave us locality-preserving MPUs, do we know that *all* locality-preserving MPUs on a given system size would be of this category? To understand MPUs better we would relax this requirement in this chapter and would ask a more general question: given a system sizes N, what is the condition on local-tensor M such that the generate MPO $O^{(N)}$ is unitary?

Another motivation of this investigation is to find an easier test that can be performed on a local tensor to predict whether or not it generates unitary on a given 1D chain of fixed size, or a set of sizes. Currently the only test to be performed on local tensor is to calculate the canonical form of its T tensor and see whether it satisfies the general form given in Eq. 5. But this can be a tedious calculation for a given arbitrary tensor, and moreover it can only tell us whether the local-tensor generates unitary on all system sizes.

Comparisons can be drawn with the theory of MPS, where just calculating the transfer matrix of the local tensor gives us a lot of information about the generic behavior of MPS on a finite periodic chain. In particular, it is well known that all transnational-invariant MPS with injective local-tensor (which means the transfer matrix has a non-degenerate highest eigenvalue) share a generic physical property: they all have short ranged correlations. This is very useful because calculation of transfer matrix and its eigenvalues is very convenient. Can we make similar claims about an MPU? The equivalent of short ranged correlation for MPU is locality-preservation. So we can ask, does an injective local tensor (which means that the transfer matrix of local-tensor, $\mathbb{E}_M$, has a non-degenrate highest eigenvalue) always creates a locality-preserving MPU? We will be investigating such questions by general approach and specific examples in this chapter. In particular we ask the following questions:

1. Are MPUs generated by an injective local-tensor always locality-preserving? If not, what is the sufficient condition for locality-preservation?
2. Given a local-tensor, how can we predict whether it generates unitary on a given 1D chain, or a set of such chains (where the set can potentially include all system sizes)?
3. Are there finite bond-dimensional non locality-preserving MPUs generated by a single local-tensor? If so, how can a locality-preserving MPU be differentiated by a non locality-preserving MPU? Is there an index that measures the amount of non locality-preservation of an MPU?

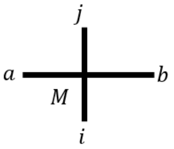
In this chapter we answer the above questions as follows ($\mathbb{E}_M, \mathbb{E}_T$ denote transfer matrix of M and T respectively):

1. Locality-preservation does not depend on the injectivity of the local-tensor M. It depends on the injectivity of T tensor. More specifically, the sufficient condition for locality-preservation is that the transfer matrix of T, $\mathbb{E}_T$ has only one non-zero eigenvalue which is equal to 1.
2. A local tensor generates unitary on 1D chain of length N if and only if $Tr(\mathbb{E}_T^N) = Tr(\mathbb{E}_M^N) = 1$. In particular, it generates unitary for all system sizes if and only if both $\mathbb{E}_M$ and $\mathbb{E}_T$ have only one non-zero eigenvalue which is equal to 1.
3. Yes, there are infinitely many finite-bond dimensional non locality-preserving MPUs. They have non-locality index which measures how much information they transfer to long distances. They are characterized by a presence of 'MPO type' symmetry similar to those in the 2D tensors that represent topological phases.

The rest of the chapter is organized as follows. We first introduce a notion of N-unitarity of a local-tensor in section 5.2 as a point of departure from the last chapter. Then in section 5.3 we would prove the central theorem of this chapter which gives a necessary and sufficient condition for a tensor to be N-unitary. A key characteristic of an MPU is how it acts on local algebras. So we would revise the formalism of support-algebras and how they relate to tensor networks in section 5.4. After this we would revisit the locality-preserving MPUs in this new formalism in section 5.5 and re-derive and refine their properties noted in the last chapter under the general notion of unitarity and support-algebras. In particular we would show that GNVW index definition directly translates to RR index in tensor networks. Finally, in section 5.6, we will discuss examples of non locality-preserving MPU and show how they have a 'long-range order' in a certain sense. We will conclude the chapter with summary and outlook.

5.2 N-unitarity

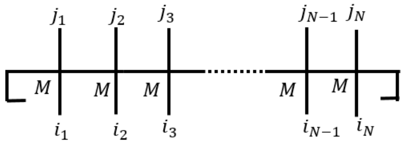
First we set the terminology for the rest of this chapter. A *local-tensor* M is a 4-index tensor with two virtual indices and two physical indices (an input and an output). Pictorially,



(5.1)

where i and j are the the input and output physical indices and a and b are the left and right virtual indices. We can put these tensors on a periodic 1D system with N sites and contract all the virtual indices. This will give us a physical operator O^N on the N sites which will refer to as the *MPO generated by the local tensor* M and denote it as $O^N(M)$,

$O^N(M) =$



(5.2)

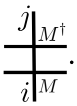
where the twists at both ends denote periodic boundary condition. We will see below how the same local-tensor can generate MPOs O^N which are unitary for some values of N and not unitary for other values. To make this precise we define

Definition 6. *A local-tensor M is N-unitary if the MPO generated by it on a periodic 1 D chain of size N, $O^N(M)$, is unitary. In such a case we denote the MPO as $U^{(N)}$.*

If a local-tensor generates unitary for all sizes in some set, $N \in \mathbb{S}$, then we would refer to them as $\mathbb{S}$-unitary. We will see that MPUs discussed in the last chapter are generated by special local-tensors which generate unitary for all system sizes. So they can be called $\mathbb{N}$-uniary local tensors. Similarly, the local-tensor in Eq. 4.79 discussed in the section is an example which generates unitary MPO only for odd system sizes. So it can denoted as $(2\mathbb{N} - 1)$ - unitary local-tensor. One of the main objectives of this chapter is to understand N-unitary local-tensors which are not $\mathbb{N}$-unitary, that is, local-tensors that generate unitaries for some but not for all system sizes.

Recall from last chapter that for any local-tensor M we can define another local-tensor $T(M)$ as

$$T^{ij}(M) = \sum_k M^{\dagger ik} \otimes M^{kj} =$$

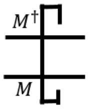


. (5.3)

When the underlying M is clear from the context, we would simply refer to it as the T-tensor. As we have already seen, tensor T plays a central role in understanding the properties of M. Sometimes it is more useful to consider the T-tensor of $M^\dagger$ instead that of M. We would simply denote them as $T(M^\dagger)$.

For any local-tensor we can define a transfer matrix as in Eq. (4.22) which we repeat here with slight modification for convenience,

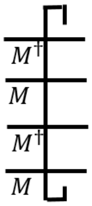
$$\mathbb{E}_M = \frac{1}{d}\sum_{ij} M^{ij} M^{ij*} = \frac{1}{d}$$



. (5.4)

Note the factor of $\frac{1}{d}$, where d is the dimension of the local physical Hilbert space. Similarly, transfer matrix of T turns out to be,

$$\mathbb{E}_T = \frac{1}{d} \sum_{ij} T^{ij} T^{ij*} = \frac{1}{d}$$



. (5.5)

Notice that T satisfies $T^\dagger = T$, but $T(M^\dagger) \neq T^\dagger(M)$.Also one can see $\mathbb{E}_M \equiv \mathbb{E}_{M^\dagger}$ and $\mathbb{E}_{T(M)} = \mathbb{E}_{T(M^\dagger)}$ in the sense that they are the same matrices with just indices shifted.

We will now see that $\mathbb{E}_M$ and $\mathbb{E}_T$ play a key role in determining N-unitarity.

5.3 Necessary and sufficient condition for N-unitarity

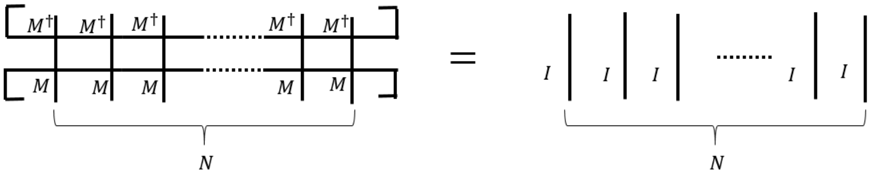
Now we present the central theorem on N-unitarity.

Theorem 6 (N-unitarity theorem)**.** *A local-tensor M is N-unitary if and only if*

$$Tr(\mathbb{E}_M^N) = Tr(\mathbb{E}_T^N) = 1. \tag{5.6}$$

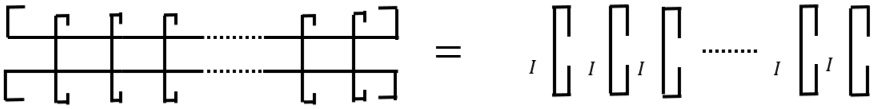
Proof. (i) First we prove: N-unitarity $\Rightarrow Tr(\mathbb{E}_M^N) = Tr(\mathbb{E}_T^N) = 1$.

This is easy to prove. Since M generates unitary on N-sites, we have



. (5.7)

Now take trace on physical indices on both sides,



$$\Rightarrow Tr((d\mathbb{E}_M)^N) = d^N$$
$$\Rightarrow Tr(\mathbb{E}_M) = 1 \tag{5.8}$$

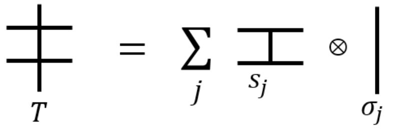
Note that since M generates unitary on N sites, $T(M)$ also generates unitary on N sites. In fact it generates identity. Since above equaltion is true for any N-unitary local tensor, it must be true for $T(M)$ also. So we have proved $Tr(\mathbb{E}_M^N) = Tr(\mathbb{E}_T^N) = 1$.

(ii) Now we prove: $Tr(\mathbb{E}_M^N) = Tr(\mathbb{E}_T^N) = 1 \Rightarrow O^{(N)\dagger}O^{(N)} = O^{(N)}O^{(N)\dagger} = I^{\otimes N}$, that is, M is N-unitary.

let's choose an orthonormal basis for the space of operators on a d-dimensional Hilbert space as $\{\sigma_j, j = 0, 1, \ldots d^2 - 1\}$, where $\sigma_0 = I$ and $\sigma_{j\neq 0}$ are traceless operators and with the property,

$$Tr(\sigma_j \sigma_k^\dagger) = d\delta_{j,k}, \quad \forall j, k. \tag{5.9}$$

Now we decompose $T(M)$ in terms of these operators on the physical space and operators on the vitrual space,

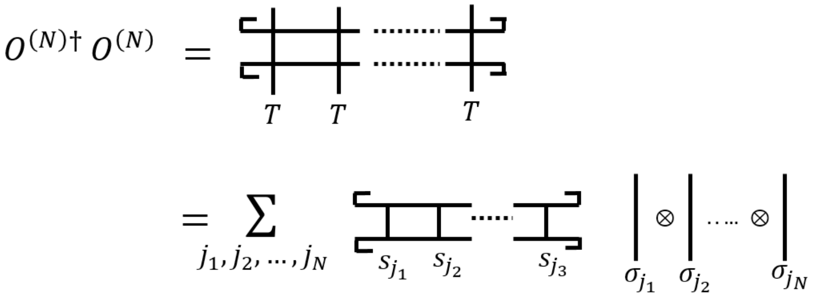


(5.10)

This is not an SVD decomposition. We have simply written T choosing some basis in the physical space. At this point we do not know anything about s_j operators. Now taking the trace on the physical legs above gives us $d\mathbb{E}_M$ on the LHS and $d \otimes s_0$ on the RHS as all other terms vanish since $Tr(\sigma_0) = d$ and $Tr(\sigma_{j\neq 0}) = 0$. So we simply get,

$$s_0 = \mathbb{E}_M. \tag{5.11}$$

Since we want to prove $O^{(N)\dagger}O^{(N)} = I^{\otimes N}$, we note that $O^{(N)\dagger}O^{(N)}$ can be expressed in terms of σ_j and s_j as


$O^{(N)\dagger}\,O^{(N)}$ =
T T T
= $\sum_{j_1, j_2, \dots, j_N}$
s_{j_1} s_{j_2} s_{j_3}
$\otimes$ ..... $\otimes$
σ_{j_1} σ_{j_2} σ_{j_N}

, (5.12)

or, algebraically,

$$O^{(N)\dagger}O^{(N)} = \sum_{j_1,j_2,\dots,j_N} Tr(s_{j_1}s_{j_2}\dots s_{j_N})\sigma_{j_1}\otimes\sigma_{j_2}\otimes\dots\otimes\sigma_{j_N}. \tag{5.13}$$

For convenience of representation, we combine indices $j_1, j_2, \dots, j_N$ into a vector index $J = |j_1, j_2, \dots, j_N\rangle$ and write

$$O^{(N)\dagger}O^{(N)} = \sum_J Tr(s_J^{(N)})\sigma_J^{(N)}, \tag{5.14}$$

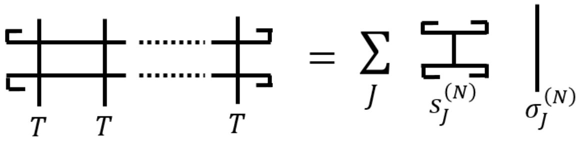
where $s_J^{(N)} = s_{j_1}s_{j_2}\dots s_{j_N}$ and $\sigma_J^{(N)} = \sigma_{j_1}\otimes\sigma_{j_2}\otimes\dots\otimes\sigma_{j_N}$ Note that $\sigma_J^{(N)}$ form an orthonormal basis for operators on $d^{\otimes N}$ space and satisfy

$$Tr(\sigma_J^{(N)}\sigma_K^{(N)\dagger}) = d^N\delta_{J,K}. \tag{5.15}$$

And $\sigma_{J=00\dots0}^{N)} = I^{\otimes N}$. So to prove $O^{(N)}$ is unitary, we need to prove

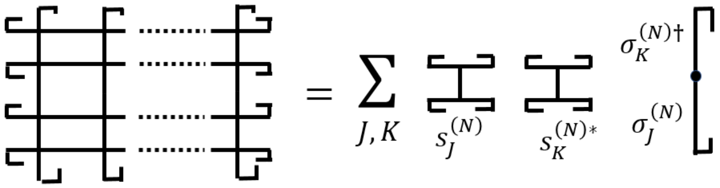
$$Tr(s_J^{(N)}) = 0, \quad \forall J \neq |00\dots0\rangle. \tag{5.16}$$

To do that, let's write equation (5.14) pictorially,



, (5.17)

and now multiply both side with their respective complex conjugates, amd take the trace on physical indices,



$$\Rightarrow\ Tr(d^N\mathbb{E}_T^N) = d^N\sum_J \left|Tr(s_J^{(N)})\right|^2$$

$$\Rightarrow\ Tr(\mathbb{E}_T^N) = \sum_J \left|Tr(s_J^{(N)})\right|^2 \qquad , \qquad (5.18)$$

where in the second step we have used the definition of $\mathbb{E}_T$ (Eq. 5.5) on the LHS and orthonormality of $\sigma_J^{(N)}$ (Eq. (5.15)) on the RHS. So now we use the assumptions $Tr(\mathbb{E}_T^N) = Tr(\mathbb{E}_M^N) = 1$ in above and get,

$$\begin{aligned}
Tr(\mathbb{E}_T^N) &= \sum_J |Tr(s_J^{(N)}|^2 \\
1 &= |Tr(s_{J=|00...\rangle}^{(N)}|^2 + \sum_{J\neq|00...\rangle} |Tr(s_J^{(N)}|^2 \\
1 &= |Tr(s_0^N)|^2 + \sum_{J\neq|00...\rangle} |Tr(s_J^{(N)}|^2 \\
1 &= 1 + \sum_{J\neq|00...\rangle} |Tr(s_J^{(N)}|^2 \\
\Rightarrow Tr(s_{J\neq|00...\rangle}^{(N)} &= 0 \qquad (5.19)
\end{aligned}$$

where $Tr(s_{J=|00...\rangle}^{(N)}) = Tr(s_0^N) = Tr(\mathbb{E}_M^N) = 1$ has been used in the 3rd step. We have proved the key step required (Eq. 5.16). So simply putting Eq. (5.19) in Eq. (5.14) we get

$$\begin{aligned}
O^{(N)\dagger}O^{(N)} &= Tr(s_{00...0}^{(N)})\sigma_J^{(N)} \qquad (5.20) \\
&= I\otimes I\otimes\ldots\otimes I. \qquad (5.21)
\end{aligned}$$

Since $\mathbb{E}_{M^\dagger}$ and $\mathbb{E}_{T(M^\dagger)}$ have the same spectrum as that of $\mathbb{E}_M$ and $\mathbb{E}_{T(M)}$ respectively (they are the same matrices with indices permuted), we also have $Tr(\mathbb{E}_{M^\dagger}^N) = Tr(\mathbb{E}_{M^\dagger} = 1$. So following the same steps as above but for $M^\dagger$ we can prove $O^{(N)}O^{(N)\dagger} = I^{\otimes N}$. So we have proved $O^{(N)\dagger}O^{(N)} = O^{(N)}O^{(N)\dagger} = I^{\otimes N}$ as desired. This completes the proof. □

The N-unitarity theorem is very useful in general because it has the following implication for any translation-invariant Wilson-loop operator: *If a translation-invariant Wilson-loop operator (of any length) can be represented by a finite bond-dimensional MPO, then its unitarity can be checked locally and efficiently.*

The N-unitarity theorem completely characterizes all possible translation-invariant unitary MPOs. It is very useful because it gives an efficient algorithm to find out whether a translation-invariant MPO is unitary. Conversely, all translational-invariant unitary MPOs can be thought of as particular solutions to N-unitarity equation (5.6).

Now we turn to other aspect of an MPU, its action on local algebras.

5.4 Action of MPU on local operators

An MPU acts on local operators A_x by conjugation $A_x \rightarrow U^{(N)} A_x U^{(N)\dagger}$. Under this operation, local algebra gets mapped to an algebra supported on a larger space. We want a good measure of this "spreading" done by MPU. In the last chapter we talked simply in terms of how much of the local algebra gets mapped to the right and how much gets mapped to the left. However, as we will see, for MPUs that map local algebra to a non-local (global) algebra this measure does not work. So we need a slightly more general quantity to work with. For this, we turn to the notion of support algebras.

First we need to define the notion of support algebras (See Ref. [1]).

Definition 7. *let's say there is a subalgebra $\mathcal{A}$ of a tensor product of algebras, $\mathcal{A} \subset \mathcal{B}_y \otimes \mathcal{B}_z$. If there is a smallest subsalgebra $C_y \subset \mathcal{B}_y$ such that $\mathcal{A} \subset C_y \otimes \mathcal{B}_z$ then C_y is called the support algebra of $\mathcal{A}$ on $\mathcal{B}_y$ and is denoted as $S(\mathcal{A}, \mathcal{B}_y)$*

In the present context we can think of $\mathcal{B}_y$ as the algebra of operators on yth site on the 1D chain, and set $z = \bar{y}$ to denote algebra of operators on all the sites excluding y. It is known that $S(\mathcal{A}, \mathcal{B}_y)$ can be calculated using SVD decomposition of $\mathcal{A}$. If we SVD decompose operators in $A \in \mathcal{A}$ between operators in $\mathcal{B}_y$ and operators in $\mathcal{B}_{\bar{y}}$,

$$A = \sum_{\mu} \lambda_\mu B_y^\mu \otimes C_{\bar{y}}^\mu, \tag{5.22}$$

where B_y^μ represent an operator on the yth site and $C_{\bar{y}}^\mu$ represents operator supported on the rest of the Hilbert space. Then support algebra $S(\mathcal{A}, \mathcal{B}_y)$ is generated by all

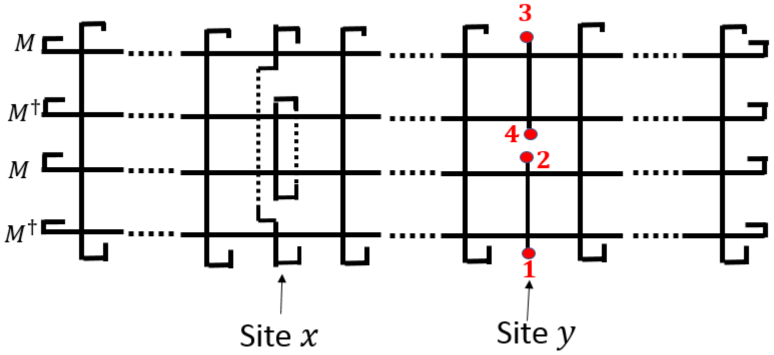
B_y^μ arising in this way. Mathematically, this is equivalent to finding the reduced density matrix of a many-body state on a particular site. Except here, we do this with operators, not vectors.

So we see that the quantity of interest for MPU is the dimension of the support algebra, $dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_y))$. Roughly speaking, it measures how much of the algebra at site y can be reached by applying MPU on algebra at site x.

Note that the GNVW index of an MPU is defined (after possible blocking) as the ratio of $dim(S(U^{(N)}(\mathcal{A}_x \otimes \mathcal{A}_{x+1})U^{(N)\dagger}, \mathcal{A}_{x-1} \otimes \mathcal{A}_x)$ and d^2, where d is the dimension of Hilbert space at each site.

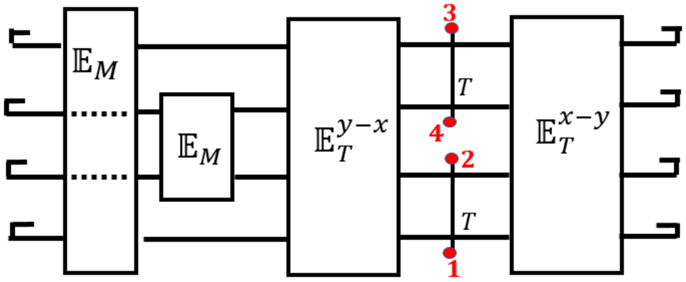
With the following proposition, we show what the notion of support algebra corresponds to in terms of tensor networks.

Proposition 1 (Action on local algebra and $\mathbb{E}_T$)**.** *$dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_y))$ is the rank of the following tensor network as a map from legs* 1, 2 *to* 3, 4*,*



(5.23)

or written in terms of T, $\mathbb{E}_T$ and $\mathbb{E}_M$,

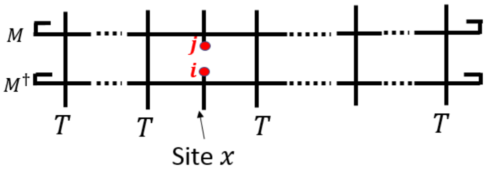


(5.24)

Proof. let's say $U^{(N)}(M)$ maps $|i\rangle\langle j|$ to $A^{(ij)}$. That is, $A^{(ij)} = U^{(N)}|i\rangle\langle j|U^{(N)\dagger}$. $A^{(ij)}$

can be written as a tensor,

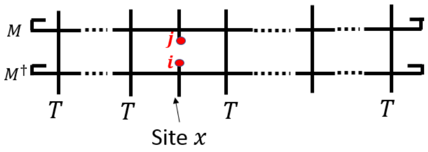
$$A^{(ij)} =$$



(5.25)

The T-tensor in the above is actually $T(M^\dagger)$ but we have denoted it simly as T for convenience. Now, Eq. (5.25) implies $A^{(ij)\dagger} = U^{(N)}|j\rangle\langle i|U^{(N)\dagger}$. In a the tensor network language

$$A^{(ij)\dagger} =$$



(5.26)

Now let's say the SVD decomposition of $A^{(ij)}$ between operators on site y and the rest is,

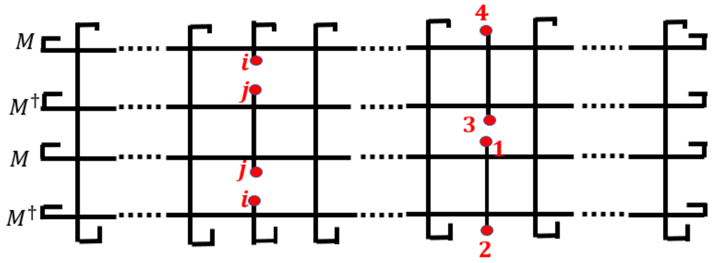
$$A^{(ij)} = \sum_\mu \lambda_\mu^{(ij)} B_{y,\mu}^{(ij)} \otimes C_{\bar{y},\mu}^{(ij)}, \tag{5.27}$$

where $\bar{y}$ denotes all sites except y. Note that operators $B_{y,\mu}^{(ij)}$ span the support algebra of $A^{(ij)}$ on site y. To get this space, we need to take a 'trace out' the C operators,

$$\begin{aligned} Tr_{\bar{y}}(A^{(ij)\dagger}A^{(ij)}) &= \sum_{\mu,\nu} \lambda_\mu^{(ij)}\lambda_\nu^{(ij)} B_{y,\mu}^{(ij)} \otimes B_{y,\nu}^{(ij)\dagger} Tr_{\bar{y}}(C_{\bar{y},\mu}^{(ij)} C_{\bar{y},\nu}^{(ij)\dagger}) \\ &= \sum_\mu (\lambda_\mu^{(ij)})^2 B_{y,\mu}^{(ij)} \otimes B_{y,\mu}^{(ij)\dagger}, \end{aligned} \tag{5.28}$$

where we have used the mutual orthogonality of C_μ as they are basis of SVD decomposition. To do the same 'tracing out' of space $\bar{y}$ in tensor network language we multiply Eq. (5.25) and Eq. (5.26) and take a trace on all sites except y. We get

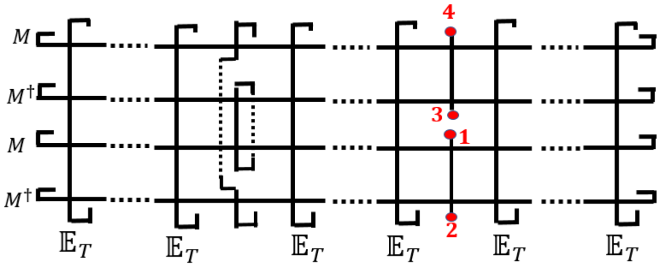
$$Tr_{\bar{y}}(A^{(ij)\dagger}A^{(ij)}) =$$



$$= \sum_\mu (\lambda_\mu^{(ij)})^2 B_{y,\mu}^{(ij)} \otimes B_{y,\mu}^{(ij)\dagger}. \tag{5.29}$$

Hence this tensor network is nothing but $\sum_\mu(\lambda_\mu^{(ij)})^2 B^{(ij)}_{y,\mu}\otimes B^{(ij)\dagger}_{y,\mu}$ Which means that operator space on indices k, l is simply spanned by operators $B^{(ij)\dagger}_{y,\mu}$. As we change i, j we get different operators $B^{(ij)\dagger}_{y,\mu}$ on site y. So to get all such Bs we simply sum over all i, j which corresponds to contracting the bottom j, i with top legs j, i on site x. So we get

$$\sum_{\mu,i,j}(\lambda_\mu^{(ij)})^2 B^{(ij)}_{y,\mu}\otimes B^{(ij)\dagger}_{y,\mu} \quad =$$



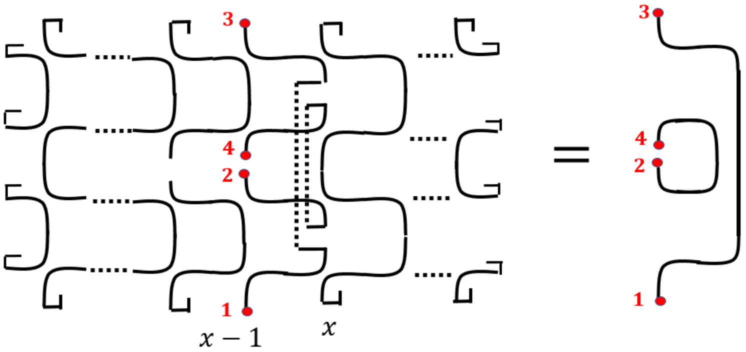
.

So we see that the tensor network on the right is nothing but the matrix on the left. Since it is a positive matrix, its rank is equal to dimension of space spanned by different $B^{(ij)}_{y,\mu}$, which is nothing but the dimension of desired subalgebra. This completes the proof. □

let's give some examples of this proposition for illustrate. let's say we wanted to calculate the support algebras of the left-shift operators. In particular we want to calculate $dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_{x-1}))$. Before we do the computation using the tensor network (5.23), we first guess what the answer should be intuitively. We know that the left-shift operators transfer the "information " to the immediate left site. So all of the algebra $\mathcal{A}_x$ should get mapped to the algebra $\mathcal{A}_{x-1}$, and nowhere else. So we expect

$$dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_y)) = \begin{cases} d^2 & \text{if } y = x-1 \\ 1 & \text{otherwise.} \end{cases} \tag{5.30}$$

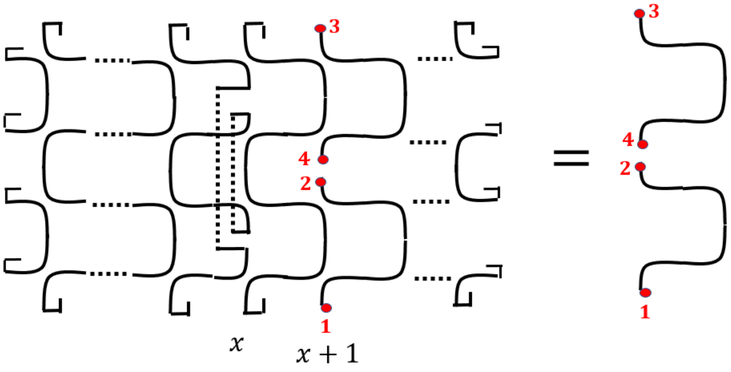
Let's see if the proposition 1 matches this intution. Drawing the required tensor network for $dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_{x-1}))$ we get,



(5.31)

Other terms simply produce constant factors as they they are traces of identity, and it does not affect the rank we are interested in. Looking at the RHS it is immediately clear that indeed the rank between 1, 2 and 3, 4 is d^2 as they are simply connected by a rank d leg. It matches what we expect from (5.30).

Now we compute $dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_{x+1}))$,



. (5.32)

One can clearly see that 1, 2 are completely disconnected from 3, 4 in the tensor network, and hence the rank of the operator from 1, 2 to 3, 4 is simply 1, which matches our expectation in (5.30).

Though it might look like a tedious calculation by hand, but the importance of this proposition lies in the description of this tensor network written in (5.24). Since it only uses matrices, this calculation can be done efficiently for any x and y. Also notice that x and y only appear with $\mathbb{E}_T$ which suggests that it is mainly $\mathbb{E}_T$ that determines how local algebra is spreaded out by the action of an MPU. In the theory of MPUs $\mathbb{E}_T$ plays a more central role than $\mathbb{E}_M$. Indeed we will see that MPU's whose $\mathbb{E}_M$ are same but $\mathbb{E}_T$ are different can differ radically from each other. In particular, at long distances, its only the degeneracy of the highest eigenvalue of $\mathbb{E}_T$ that determines the correlation length.

With the theorem 6 on unitarity, and proposition 1 at hand, now we revisit the question of locality-preserving MPUs and refine and expand our results and understanding from the last chapter.

5.5 Revisiting locality-preserving MPUs

Theorem 6 immediately gives a necessary and sufficient condition on the local tensors that generate locality-preserving MPUs of last chapter. But before we present that, we need to note a small lemma whose proof can be found in whose proof can be found in Ref. [70]. We just repeat the result here for convenience.

Lemma 8. *Consider two sets of complex numbers, $\lambda_{\alpha,k_\alpha} = |\lambda_{\alpha,k_\alpha}|e^{i\phi_{\alpha,k_\alpha}}, \alpha = a, b; k_\alpha = 1, 2, \dots x_\alpha$. If $\forall N \leq max\{x_a, x_b\}$ the following is true,*

$$\sum_{k=1}^{x_a} \lambda_{a,k}^N = \sum_{k=1}^{x_b} \lambda_{b,k}^N \tag{5.33}$$

then $x_a = x_b$ and $\lambda_{a,k} = \lambda_{b,k}$ (up to permutation).

Now we present the result for locality-preserving MPUs.

Corollary 5. *A local-tensor generates MPUs for all system sizes (that is, it is N-unitary for all $N \geq 1$) if and only if both $\mathbb{E}_M$ and $\mathbb{E}_T$ have only one non-zero eigenvalue which is equal to 1.*

Proof. let's say $\mathbb{E}_M$ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \lambda_m$. Then, since M is unitary for all system sizes, we have

$$\sum_j \lambda_j^N = 1, \quad \forall N \geq 1. \tag{5.34}$$

Applying Lemma 8 on this equation we get $\lambda_1 = 1, \lambda_2 = \lambda_3 = \dots = 0$. So $\mathbb{E}_M$ has only one non-zero eigenvalue which is equal to 1. The same can be applied to $\mathbb{E}_T$ and we get that $\mathbb{E}_M$ has only one non-zero eigenvalue which is equal to 1. This completes the proof. □

It is also interesting to note another immediate implication of Lemma 8

Corollary 6. *If a local-tensor with virtual bond dimension D generates MPU for all system sizes $N \leq D^4$, then it generates unitary for all system sizes, and hence is locality-preserving.*

Proof. Since $\mathbb{E}_T$ can have maximum D^4 eigenvalues, if $\sum_j \lambda_j^N = 1, \quad \forall N \leq D^4$, then using Lemma 8 we have $\lambda_1 = 1, \lambda_2 = \lambda_3 = \dots = \lambda_{D^2} = 0$. The same goes for eigenvalues of $\mathbb{E}_M$ □

Now we prove that just as injectivity of local-tensor in MPS theory implies short ranged correlation, injectivity of T-tensor in the theory of MPUs implies short-ranged mapping.

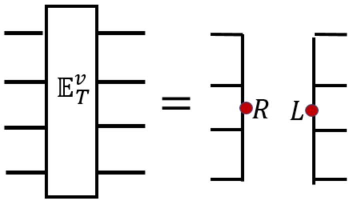
Theorem 7 (T-injectivity implies locality-preservation)**.** *If the T-tensor of a MPU generating local-tensor M is injective (has only one non-zero eigenvalue which is equal to 1), then the MPU, $U^{(N)}$, is locality-preserving. In particular, $\mathbb{N}$-unitary local-tensors generate locality-preserving MPUs.*

Proof. let's say $\mathbb{E}_T$ has only one non-zero eigenvalue which is equal to one. The eigenvalues 0 may have a jordan block associated to them. So there must exist a number $v \leq D^4$, where D is the dimension of the virtual legs, such that $\mathbb{E}_T^v$ has no jordan blocks, and hence is a rank 1 matrix. Any rank 1 matrix can be written as an outer product of two vectors, so

$$\mathbb{E}_T^v = |R\rangle\langle L|, \tag{5.35}$$

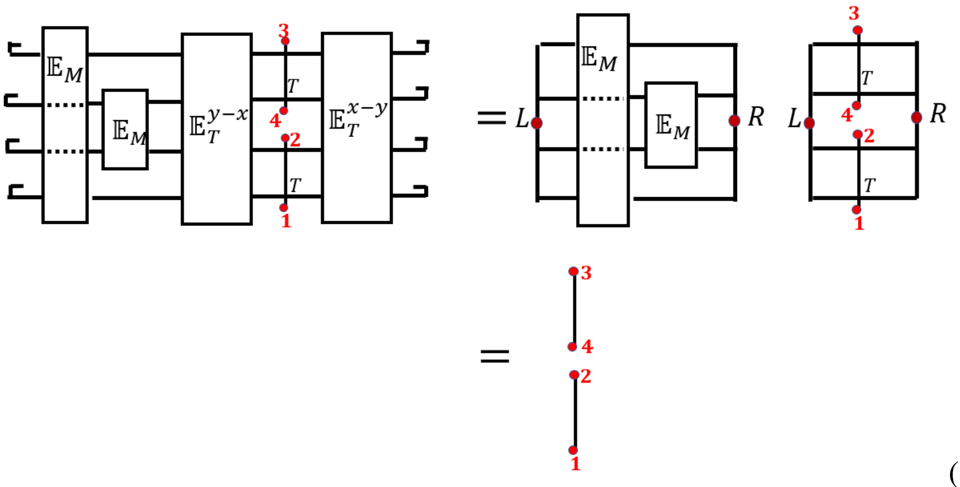
$$\text{with } \langle L|R\rangle = 1 \tag{5.36}$$

or pictorially,



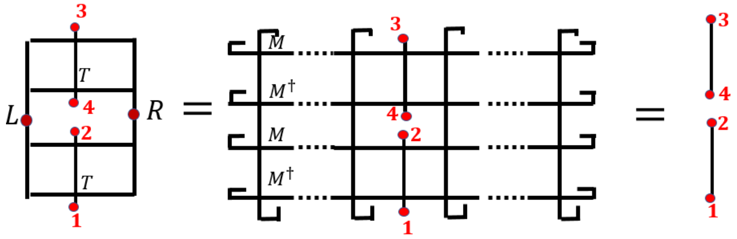
(5.37)

where R_T and L_T are the right and left eigenvectors of $\mathbb{E}_T$ with eigenvalue 1. This implies that the diagram in Eq. 5.24 decomposes if $|y - x| > v$, that is,



(5.38)

The first term on the RHS is just a number so we ignore it. To see why the second term is just identity on $1, 2$ and $3, 4$, we can write it as



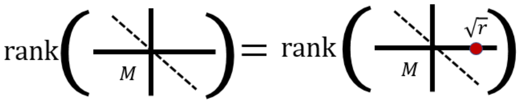
(5.39)

since M generates unitary, it gives identity on each site, and that's why we end up with identity on $1, 2$ and $3, 4$. In fact without having done the detailed calculation we can see that if $\mathbb{E}_T^v$ decomposes for some v, then looking at Eq. (5.29) we can see that no operator on site x can be mapped beyond $x+v$. Hence $U^{(N)}\mathcal{A}_x U^{(N)\dagger}$ cannot have a support on sites more than v distance away from x. So $dim(S(U^{(N)}\mathcal{A}_x U^{(N)\dagger}, \mathcal{A}_y)) = 1, \forall |y - x| > v$. This completes the proof. □

This theorem confirms the general implication of proposition 1, that the long distance behavior of an MPU is dictated by the highest eigenvalue degeneracy of the $\mathbb{E}_T$. So in particular if this degeneracy is 1, we get short ranged mapping by the MPU.

Notice that we did not need the injectivity assumption like we did in theorem 5 of last chapter. It means theorem 5 should be true irrespective of whether or not M is injective, or a stack of injective MPUs. We will now show that indeed that is the case. We can remove the injectivity assumption if we block sites once more after we reach the fixed point equations of separation and isometry.

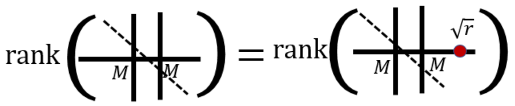
To be more precise, we needed injective assumption (Chapter 4, definition 5) because if the left eigenvector, l, and right eigenvector, r, of the transfer matrix $\mathbb{E}_M$ are not supported on the full virtual space, we cannot in general claim that multiplying M with $\sqrt{r}$ from the right or multiplying l from the left does not change its SVD ranks needed for RR-index definition. That is, we cannot claim, for example, that the left SVD rank remains the same under following modification.



(5.40)

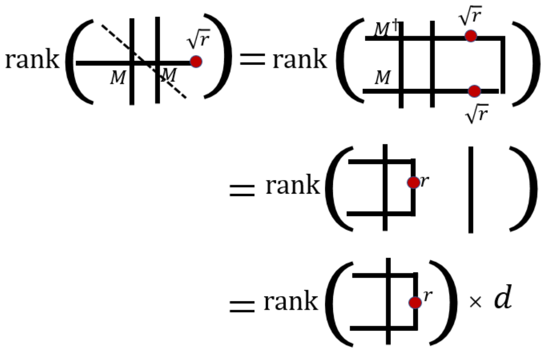
But since we proved index multiplicability under stacking using such modifications, we were forced to use the injectivity assumption. We now show that the above equation is always true, if we block enough sites.

Lemma 9. *For a local tensor M that satisfies fixed point equations (separation as in Eq. 4.26 and isomerty as in Eq. 4.27) the following holds true:*



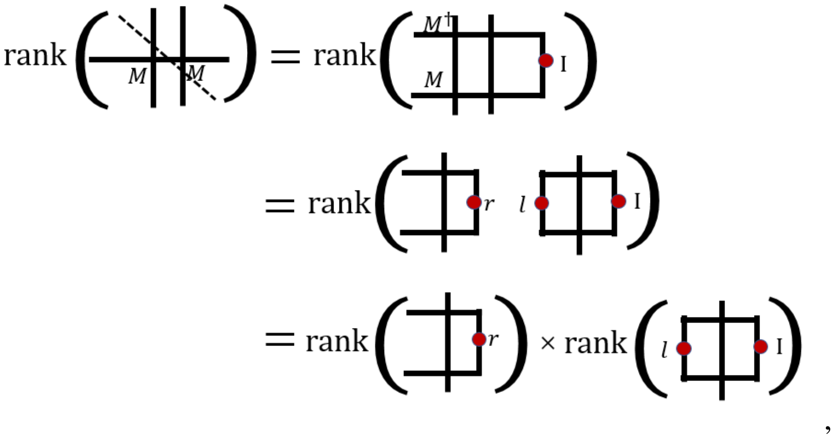
(5.41)

Proof. First we calculate the rank of the RHS:



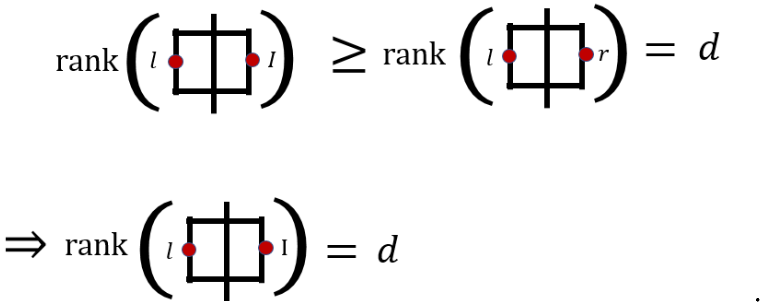
(5.42)

We do not need to know the rank of the first term on the RHS of this equation to prove the desired result. Now we calculate the rank of the LHS of Eq. 5.41,



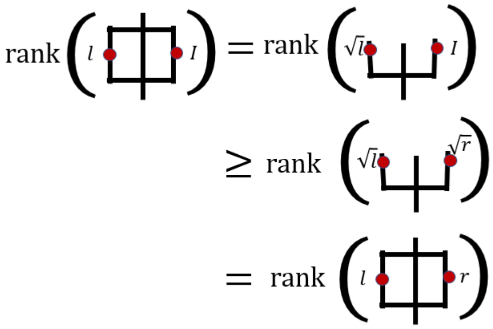
(5.43)

where where l is the left eigen-vector of $\mathbb{E}_M$ and I is just identity operator on virtual index. The second equality comes from separation. But now notice:



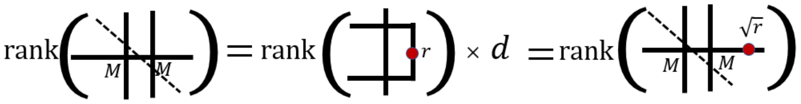
(5.44)

The final equality comes from the fact that the rank cannot possibly be greater than d. To see why the first inequality is true, notice, more explicitly,



. (5.45)

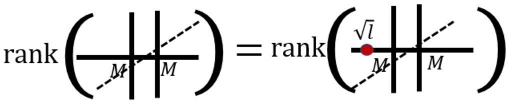
The inequality in second line comes from the fact that $\sqrt{r}$ is a non-negative matrix and for any matrix A and a non-negative matrix B, $rank(A) \geq rank(AB)$. (To see why, diagonlize B, (which can always be done since its Hermitian), $B = U\Lambda U^{\dagger}$, where Λ is diagonal matrix. Now keeping in mind that multiplication with a diagonal matrix can only possily reduce rank, we have $rank(A) = rank(AU) \geq rank(AU\Lambda) = rank(AU\Lambda U^{\dagger}) = rank(AB)$.) The first and final equality are simply the expression $rank(A) = rank(AA^{\dagger})$ for any matrix A. So we see that the Eq. (5.44) is true. Substituting relation (5.44) back in Eq. (5.43) we get the desired result



. (5.46)

This completes the proof. □

It is an obvious to see that in a similar fashion we can prove,



(5.47)

So, now on, we assume that we have blocked enough sites such that injectivity condition is automatically satisfied and hence results of the last chapter are valid for all local-tensors that generate unitary for all system sizes.

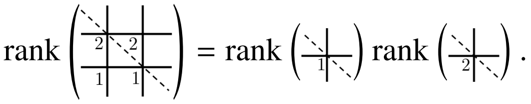
In last chapter, in theorem 5 we showed that RR-index and GNVW index are related by first showing it to be the case for all the building blocks of locality preserving MPUs, and then showing it to remain true under stacking of MPUs. Now we take a different approach, and show that RR-index definition is actually directly implied by the GNVW index definition when we apply it to tensor networks.

Deriving RR-index from GNVW index

Now we will show that RR-index actually directly comes from GNVW index. To prove that we first need a small result

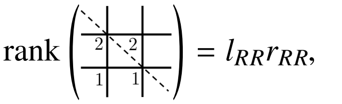
Lemma 10. *If a locality-preserving MPU satisfies the fixed point equations, and l_{RR} and r_{RR} are its left and right SVD ranks respectively, then $l_{RR}r_{RR} = d^2$.*

Proof. It is easy to prove this. In the last chapter we proved that if $M_{12} = M_1 M_2$ then we have,



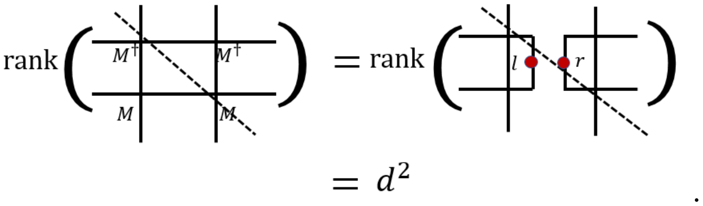
(5.48)

let's put $M_1 = M$ and $M_2 = M_1^\dagger$. So the above equation gives us



(5.49)

where we have used $l_{RR}(M^\dagger) = r_{RR}(M)$, which follows from the definition of $M^\dagger$. But we also have the following due to separation:



(5.50)

The reason for the equality is similar to that argued in the proof of lemma 9. (we can multiply the two sides with r and l and get identity on both sites.) So we get $l_{RR}r_{RR} = d^2$. This completes the proof. □

We proved in theorem 5 in last chapter that $I_{GNVW} = \sqrt{I_{RR}}$ by proving it first for the building blocks of locality-preserving MPUs and then for stacks of them. Now we will reprove the same relation but in a different way. In this way of proof we will see that RR-index actually directly comes from the GNVW index definition.

Theorem 8. *If a local-tensor M that generates MPU for all system sizes is block enough times so that it satisfies the fixed-point equations (separation(Eq. 4.26), isometry (Eq. 4.27) and lemma 9) and has l_{RR} and r_{RR} as its left and right SVD ranks, then its GNVW index is*

$$I_{GNVW}(U^{(N)}) = \frac{l_{RR}^2}{d^2} = \frac{d^2}{r_{RR}^2} = \frac{l_{RR}}{r_{RR}} = \sqrt{I_{RR}(M)}, \tag{5.51}$$

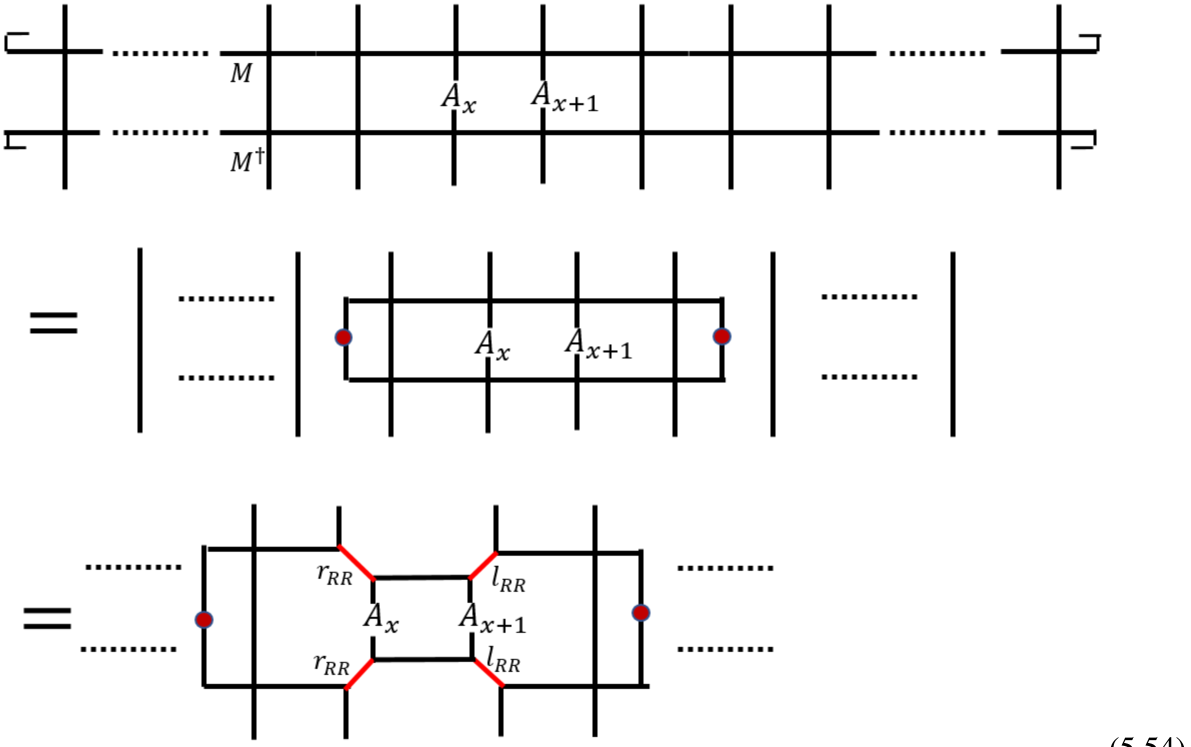
where d is the dimension of the Hilbert space on each site.

Proof. First we recall that the GNVW index of a 1D unitary operator is defined in terms of support algebras. That is, let's say,

$$l_{gnvw} = dim(S(U^{(N)}(\mathcal{A}_x \otimes \mathcal{A}_{x+1})U^{(N)\dagger}, \mathcal{A}_{x-1} \otimes \mathcal{A}_x)) \tag{5.52}$$

$$r_{gnvw} = dim(S(U^{(N)}(\mathcal{A}_x \otimes \mathcal{A}_{x+1})U^{(N)\dagger}, \mathcal{A}_{x+1} \otimes \mathcal{A}_{x+2})). \tag{5.53}$$

Then $I_{GNVW} = l_{gnvw}/d^2 = d^2/r_{gnvw}$. It was shown $l_{gnvw}r_{gnvw} = d^4$. So to prove the theorem we just need to prove $l_{gnvw} = l_{RR}^2$ and $r_{gnvw} = r_{RR}$. To do that, let's see how an MPU acts on operators $A_x \otimes A_{x+1} \in \mathcal{A}_x \otimes \mathcal{A}_{x+1}$,



(5.54)

where in the first step we have used separation and isometry, and in the second step we have performed left and right SVD on M and $M^\dagger$. We have used $l_{RR}(M^\dagger) = r_{RR}(M)$ and vice-versa. Recall that the definition of $S(U^{(N)}(\mathcal{A}_x \otimes \mathcal{A}_{x+1})U^{(N)\dagger}, \mathcal{A}_{x-1} \otimes \mathcal{A}_x)$ is that it is the space of operators on $\mathcal{A}_{x-1} \otimes \mathcal{A}_x$ that appear in the SVD decomposition of operators in $U^{(N)}(\mathcal{A}_x \otimes \mathcal{A}_{x+1})U^{(N)\dagger}$. But we do not need to perform this SVD decomposition. We can directly see from the tensor diagram that the number of non-zero singular values of this SVD cannot exceed l_{RR}^2. (We can use proposition 1 for support algebras to see it more explicitly.) So we get

$$l_{gnvw} \leq l_{RR}^2 \tag{5.55}$$

By similar argument with the space $S(U^{(N)}(\mathcal{A}_x \otimes \mathcal{A}_{x+1})U^{(N)\dagger}, \mathcal{A}_{x+1} \otimes \mathcal{A}_{x+2})$ we get

$$r_{gnvw} \leq r_{RR}^2. \tag{5.56}$$

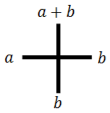
These two inequalities together with equalities $l_{gnvw} r_{gnvw} = d^4$ and $l_{RR} r_{RR} = d^2$ (lemma 9) give us $l_{gnvw} = l_{RR}^2$ and $r_{gnvw} = r_{RR}^2$. Plugging it in the definitions of the GNVW index and RR-index with get the desired result. This completes the proof. □

With this theorem we conclude our discussion of locality-preserving MPUs. Now we will turn to solutions of the N-unitarity equation (Eq. (5.6)) which depend on the values of N. That is, they satisfy the equation only for certain set of N. We will see that such MPUs are often Non locality-preserving. That is, they map local operators to non-local operators.

5.6 Non locality-preserving MPUs: 'long-range ordered' MPUs

The MPUs of last chapter and last section can be seen as solutions to the equations $\mathbb{E}_M^N = \mathbb{E}_T^N = 1$ for all $N \in \mathbb{N}$. Are there solutions that only work for a subset of natural numbers? As we will show now there are infinitely many such solutions. The local tensor mentioned in Sect. 5.2 in last chapter is one of them. let's consider a similar example here. Consider a local-tensor

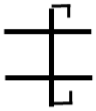
$$M = \sum_{a,b=0}^{2}$$



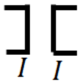
(5.57)

(This local-tensor can be obtained from the one in last chapter by applying a unitary operator on each site, so it has essentially the same properties.) let's first study how this examples fits with the theorem 6. Its transfer matrix can be calculated simply to be

$$\mathbb{E}_M = \frac{1}{3}$$



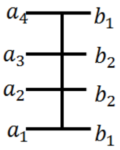
$$= \frac{1}{3}$$



, (5.58)

which has only one non-zero eigenvalue which is equal to 1. So we see that transfer matrix for this M is exactly similar to those of locality-preserving MPUs. But what makes it different is the matrix $\mathbb{E}_T$

$$\mathbb{E}_T = \frac{1}{3}\sum_{a_j,\,b_j=0}^{2} \delta_{a_1+b_1,a_2+b_2}\ \delta_{a_3+b_2,a_4+b_1}$$

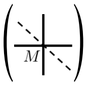


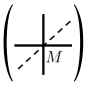
. (5.59)

By calculating its eigenvalues, we find that the only non-zero eigen values are $\{1, 1, -1\}$. So if it is to satisfy the condition of theorem 6, we should have

$$Tr(\mathbb{E}_T^N) = 1^N + 1^N + (-1)^N = 1, \tag{5.60}$$

which is true if and only if N is odd. This explains why in the example in section 5.2 of the last chapter we found that the MPU was unitary for all odd system sizes. Also, if we calculate the left and right SVD ranks for this local-tensor we find,

$l_{RR} = \text{rank}\Big($  $\Big) = 9$ (5.61)

$r_{RR} = \text{rank}\Big($  $\Big) = 3.$ (5.62)

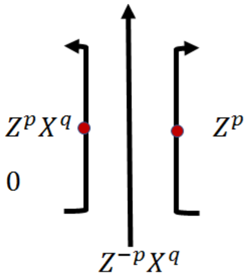
So

$$l_{RR} r_{RR} = 9 \times 3 = 27 > 3^2 = d^2. \tag{5.63}$$

So this local-tensor violates the relation required for locality-preserving MPUs as proved in Lemma 10. This is another indication that it is not a locality-preserving MPU.

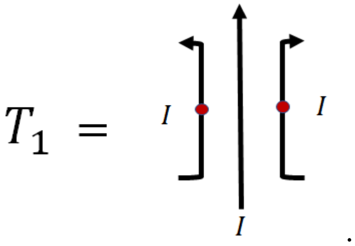
Recall that we proved in the last chapter that the tensor T has to have only one block if M is to produce unitary for all system sizes. So let's analyze the block structure of T. But we will use $T(M^\dagger)$ instead of $T(M)$ for reasons to be explained later. We find

$$T(M^\dagger) = \frac{1}{3}\sum_{p,q=0}^{2}$$



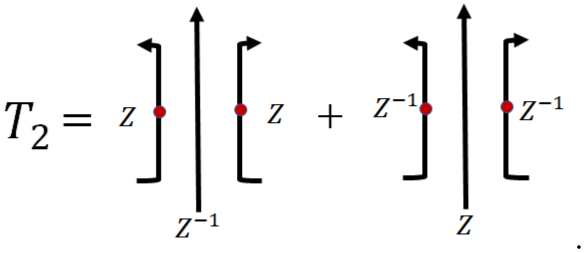
. (5.64)

One can see that this T actually has 2 blocks. This first block corresponds to $p = q = 0$,



(5.65)

This block clearly has rank one and has identity as the physical operator. So this is exactly the same as that in the general form in (5). If there was only this block, this local-tensor would behave like locality preserving tensors and generate unitary for all system sizes. But it doesn't because there is a second block as well. This block corresponds to $p = 0, q = 1, q = -1$,



(5.66)

Clearly, it has rank 2, and its a periodic block. It is actually an MPO version of transnational symmetry breaking MPS first noted in [27]. As T contract with each other, the first term in this block can only contract with the second term and vice-versa. So these terms alternate throughout the chain. But notice that when the number of sites is odd, it cannot connect back to itself and it disappears, in which case we are only left with the rank 1 block, and hence the overall MPU is unitary. But when number sites are even, this block doesn't vanish, and we get a non-unitary operator. In fact, one can see that on even sites we get an operator

$$O^{(N)} = I^{\otimes N} + \bigotimes_{x=1}^{N/2} Z_{2x-1} \otimes Z_{2x}^{-1} + \bigotimes_{x=1}^{N/2} Z_{2x-1}^{-1} \otimes Z_{2x}. \tag{5.67}$$

So we see that, other can calculating eigenvalues of $\mathbb{E}_M$ and $\mathbb{E}_T$, calculating the block structure of T can also tell us about the periodic nature of N-unitarity of the given local-tensor. We will do the same with similar examples below.

How does this MPU act on local operators? Remember that in proving locality-preservation in theorem 7 we used the fact that $\mathbb{E}_T$ was injective (had only one non-zero eigenvalue). Does it imply that if $\mathbb{E}_T$ has more than one eigenvalues, then the MPU would be non locality preserving? The answer to this question is, generically, yes, but not always. let's choose convenient basis to see how the MPU generated by local-tensor in (5.57) acts on local algebra. We choose the basis $Z^a X^b,\ a, b = 0, 1, -1$ where ($\omega = exp\left(\frac{2\pi i}{3}\right)$)

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Z and X are generalized pauli operators. With this the basis $Z_x^a X_x^b$ spans the whole algebra of operators on site x. In fact, Z_x and X_x generates the whole algebra on site x, so we just need to calculate $U^{(N)} Z U^{(N)\dagger}$ and $U^{(N)} X U^{(N)\dagger}$. We will give the detaoil calculation below, but here we just note the results

$$\begin{aligned} UZ_xU^\dagger &= \dots (Z_x^{-1} Z_{x+1})(Z_{x+2}^{-1} Z_{x+3}) \dots, \qquad (5.68) \\ UX_xU^\dagger &= X_x X_{x+1}, \qquad (5.69) \end{aligned}$$

where "$\dots''$ denotes that the alternating powers continue through the whole periodic 1D chain. So we see that though it acts locally on X operators, it maps Z operators to global operators: operators that are supported on all sites. X alone generates a 3-dimensional subalgebra of 1-site algebra. It means 3-dimensional subalgebra of the local algebra is mapped locally, while the rest is mapped nonlocally. Note that locally mapped operators form a subalgebra: if A and B are mapped locally then so are $A+B$ and AB since $U(A + B)U^\dagger = U(A)U^\dagger + U(B)U^\dagger$ and $U(AB)U^\dagger = U(A)U^\dagger U(B)U^\dagger$ and sum/prdouct of local operators is local. We would call this subalgebra simply 'locally-mapped subalgera'. The importance of 'locally-mapped subalgera' is that a GNVW index can be defined on this subalgebra, and it interacts with other locality-preserving MPUs in the usual way. We will see later that some MPUs do not even have any non-trivial locally-mapped subalgebra. That is, they map all local operatos to non-local operators.

It is simple to see that in the language of support algebras, we have

$$dim(S(U^{(N)} \mathcal{A}_x, U^{(N)\dagger}, \mathcal{A}_y)) = \begin{cases} 9 & \text{if } y = x, x + 1. \\ 3 & \text{otherwise} \end{cases} \qquad (5.70)$$

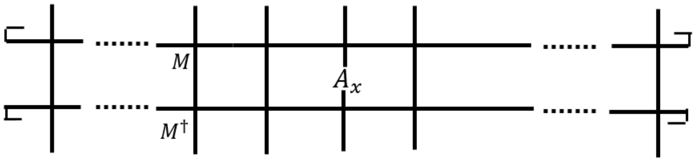
So as $N \longrightarrow \infty$, $U^{(N)}$ still maps local operators to operators supported on 3 dimensional subalgebra of algebras far away on the chain. In other words, we can say that it has a long range correlation where a 3-dimensional subalgerba is correlated with the local algebra.

Non-locality index

It is obvious that GNVW index is not well defined for non locality-preserving MPUs. But notice that it is easy to see that the number $dim(S(U^{(N)}\mathcal{A}_x, U^{(N)\dagger}, \mathcal{A}_y)),\ |x-y| \gg 1$ does not change if we combine the above MPU with locality preserving MPUs of last section. Locality preserving map local operators to local operators, so they have no effect on the long range behavior of this MPU. Hence we can say that the quantity $dim(S(U^{(N)}\mathcal{A}_x, U^{(N)\dagger}, \mathcal{A}_\dagger)),\ |x-y| \gg 1$ is protected against locality-preserving MPUs. We would refer to this quantity as the *non-locality index* of non locality-preserving MPUs.

Symmetries of local tensor and non-local mapping

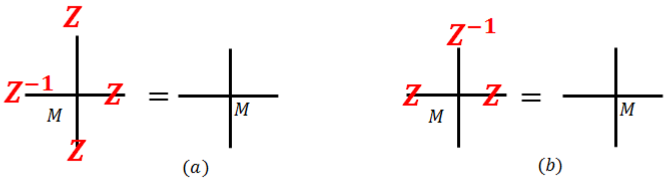
Recall that in Chapter 2, Sect. 2.7 we observed the reason for instability was that, due to certain symmetries of the tensors, some local operators on the virtual level get mapped to non-local operator on the physical level. Interestingly, the non local-mapping of the MPU can be understood in a similar way. First notice that when we calculate the operator, $U^{(N)}A_xU^{(N)\dagger}$, we are calculating the tensor network diagram,



. (5.71)

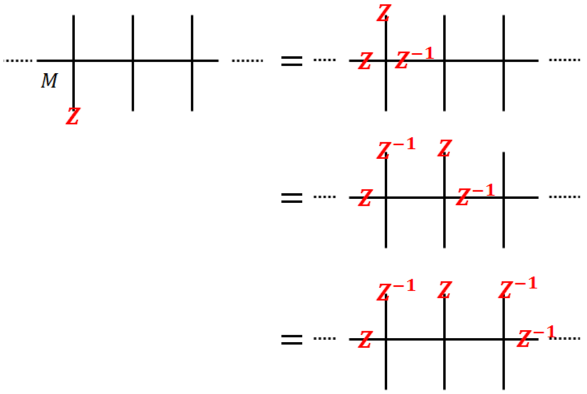
So it can be understood as a mapping from 'virtual leg' to physical leg because A_x is applied on the legs being contracted in the tensor network. So, mathematically, the source of non-local mapping can be the same as it did in the 2D case.

So how can we see non-local mapping of the local-tensor in (5.57) by its symmetries? let's try to understand the Eq. (5.68). Notice that the local-tensor (5.57) satisfies the following two important symmetries,



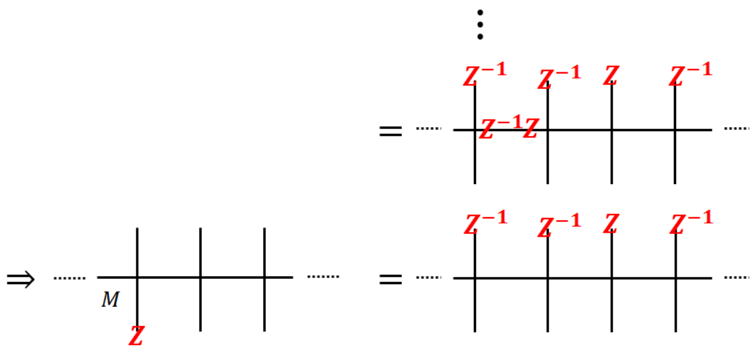
. (5.72)

With these symmetries we can see how $U^{(N)}$ acts on Z,



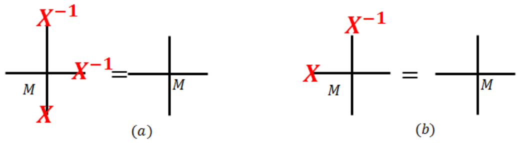
, (5.73)

where in the first step we have used symmetry in (5.72)(a), and then we have recursively used symmetry (5.72)(b). This process continues until it reaches back to the starting place due to periodic boundary condition, and we get



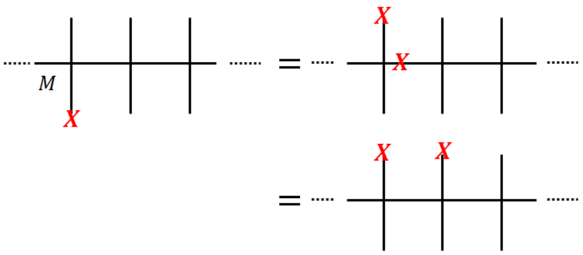
. (5.74)

This explains the result in Eq. (5.68). Similarly, to under its action on X operator, we note the relevant symmetries,



. (5.75)

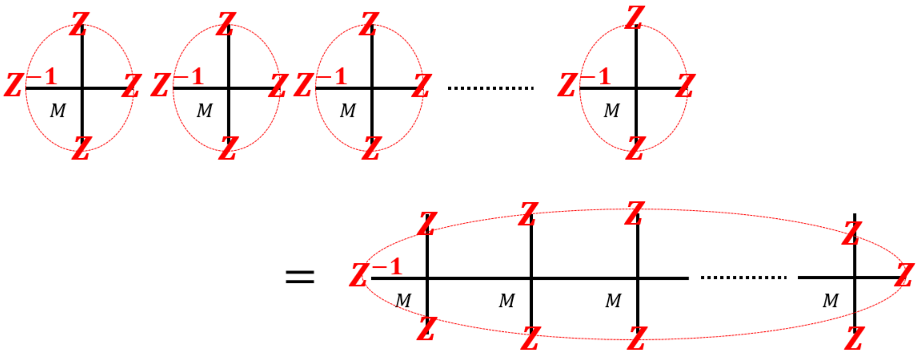
Using this we obtain



, (5.76)

where in the first step symmetry (5.75)(a) is used and then symmetry (5.75)(a) in the second step. Notice that X, unlike Z, cannot travel beyond the 2nd site and stops there. Thus we recover the result (5.69).

So what was the key property of this local-tensor that made it a non locality-preserving MPU? If we look carefully, we find that the symmetry noted (5.72)(a) plays a central role in its non-local behavior. In fact, it is the kind of 'MPO symmetry' which plays a key role in the topological/long-range entangled behavior of 2D states. Recall that in Chapter 2 we noted how the Z_2 symmetry $Z^{\otimes 3}$ symmetry of the local-tensor was essential for maintaining topological order of the toric code state. The essential feature of such a symmetry, that differentiates it from other local symmetries, is that it is a long range symmetry constraint on the tensor network. The toric code single-line TNR satisfies the Z_2 symmetry generated by $Z^{\otimes n}$ on any arbitrary region. Something similar is true with the symmetry (5.72)(a). As we contract local-tensors together, it remains to be a non-local Z_3 symmetry,

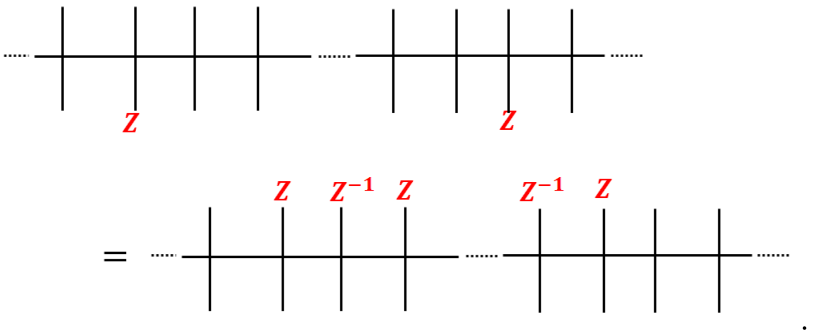


(5.77)

Finally when we close the periodic boundaries, we get

$$Z^{\otimes N} U^{(N)} Z^{\otimes N} = U^{(N)}. \tag{5.78}$$

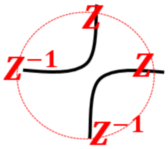
To see the resemblance with the phenomena of local virtual operators being mapped to non-local physical operators more clearly, note that this MPU satisfies



(5.79)

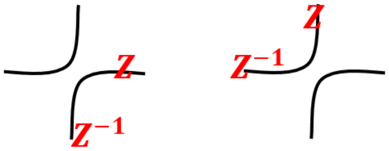
That is, two Z sitting far away get mapped to a string of operators stretching between them. This is exactly like Eq. 2.60 in Chapter 2.

Since $Z^3 = I$, we can say that this MPU has a long-ranged Z_3 order. Note that these MPO symmetries of the local tensor that generate non-local symmetries of the tensor network should not be confused with other local symmetries that might look the same. For example, even for a shift MPU one can claim that it has the symmetry that looks like an MPO symmetry,



. (5.80)

But of course it is not an MPO symmetry since it can be broken into smaller symmetries,



. (5.81)

In fact, it is easy to see that locality-preserving MPUs do not have such long-ranged MPO symmetries. We conjecture that the presence of such MPO symmetry is precisely what distinguishes locality-preserving from non locality-preserving MPUs.

Now that we have analyzed this particular non locality-preserving MPU in detail, we would now show that there are infinitely many such MPUs with similar properties.

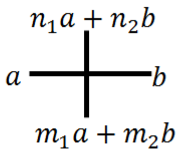
Generic examples of non locality-preserving MPU

let's say the physical space dimension, d, is a prime number. Consider a general integer matrix S with entries in $0, 1, \ldots, d-1$,

$$S = \begin{bmatrix} m_1 & m_2 \\ n_1 & n_2, \end{bmatrix} \tag{5.82}$$

where $m_1, m_2, n_1, n_2 \in [0, 1, \ldots, d]$. Now consider a local tensor as a function of this matrix S,

$$M(S) = \sum_{a,b=0}^{d-1}$$

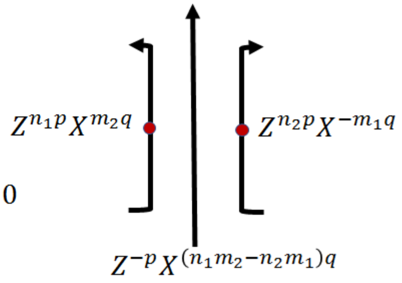


, (5.83)

where the sum and multiplications should be understood as modulo d. The non locality-preserving MPU (Eq. 5.57) discussed so far is a specific instant of the above with $d = 3$ and $S = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. We can also easily see that $M\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$ is nothing but the right-shift MPU and $M\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right)$ is nothing but the left-shift MPU. And $M\left(\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}\right)$ or $M\left(\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}\right)$ gives an identity operator.

One can calculate the transfer matrices and apply theorem 6 to find a constraint on S for M to generate unitaries for certain, or all system sizes. But instead of doing that, we choose to work with specific examples and will see how this simple expression can produce very interesting MPUs. But first we give the general block structure of this general local-tensor which can be used in specific case to predict the periodicity of the its N-unitarity. One can calculate that the T matrix of above local-tensor is,

$$T(M^\dagger) = \frac{1}{d}\sum_{p,q=0}^{d-1}$$



. (5.84)

For a given matrix S, we would be able to immediately see by inspection the block structure of T, and hence the periodicity of N-uniartity of the local-tensor M. Now we look at an interesting example and will calculate the relevant quantities discusses so far.

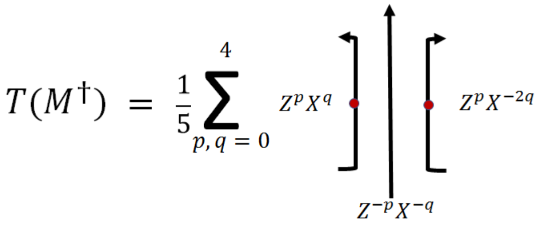
A completely non locality-preserving MPU

We saw that local-tensor in (5.57) has a 3 dimensional locally mapped subalgebra. It means it maps operator in that 3 dimensional subsalgebra to local operators and rest of the algebra to non-local operators. Now we will see an example of MPU that has no non-trivial locally mapped subalgebra. That is, it will map *all* of local (1-site) operators to non-local operators.

If we use $d = 5$ and $S = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ in (5.83) we get the local tensor

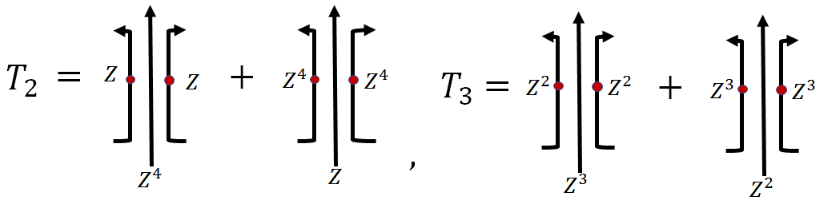
. (5.85)

Let’s first calculate the T tensor putting $m_1 = 2, m_2 = 1, n_1 = 1, n_2 = 1$ in expression (5.84). We get



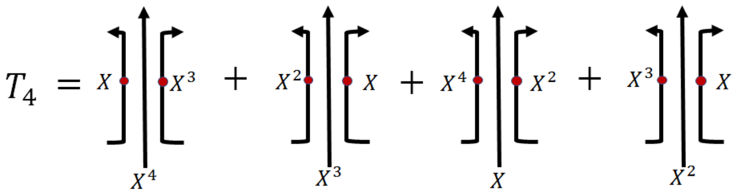
. (5.86)

We see that it has one rank 1 block corresponding to $p = q = 0$ which looks exactly similar to that in Eq. (5.65). Then there are 2, rank-2 periodic blocks corresponding to $p = 0, q = 1, 4$ and $p = 0, q = 2, 3$,



. (5.87)

Finally there is one rank-4 periodic block corresponding to $q = 0, p = 1, 2, 3, 4$,



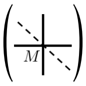
. (5.88)

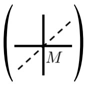
Note that the terms in the block are written in the order they connect to each other as we blocks sites to the right. That is, the first term only connects to second one, second one only connects to third on and so on. Based on this we expect the eigenvalues of $\mathbb{E}_T$ to be $\{1\}$, $\{1, -1\}$, $\{1, -1\}$ and finally $\{1, \omega_4, \omega_4^2, \omega_4^3\}$, where $_n$ stands for nth root of unity. Indeed we find this to be the case numerically. According to theorem 6 for N-unitarity we need to have

$$1^N + (1^N + (-1)^N) + (1^N + (-1)^N) + (1^N + (\omega_4)^N + (\omega_4^2)^N + (\omega_4^3)^N) = 1, \quad (5.89)$$

which holds if and only if N is odd. Indeed we find that this local tensor generates MPU only on odd system sizes.

Calculating the left SVD and right SVD rank, we find

$l_{RR} = \text{rank}\Big($  $\Big) = 25, \qquad (5.90)$

$r_{RR} = \text{rank}\Big($  $\Big) = 25. \qquad (5.91)$

So we again see $l_{RR} r_{RR} > d^2$, which is another indication that it is not a locality-preserving MPU. In fact, this is a maximal violation of this constraint as l_{RR} and r_{RR} attain their maximum values, d^2. Based on this we expect this local-tensor to be even more non locality-preserving than the one we analyzed before in 5.57. We indeed find it to be case as we will show now. To see how it acts on the local algebra, we calculate $U^{(N)} Z U^{(N)\dagger}$ and $U^{(N)} X U^{(N)\dagger}$, where X and Z are genealized pauli operators for dimension d,

$$\begin{aligned} Z &= \sum_{j=0}^{d-1} \omega_d^j |j\rangle\langle j|, \text{ where } \omega = e^{\frac{2\pi i}{d}} && (5.92) \\ X &= \sum_{j=0}^{d-1} |j\rangle\langle j+1|. && (5.93) \end{aligned}$$

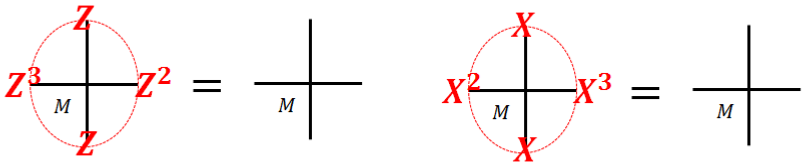
We use Z, X with $d = 5$ here. We observe,

$$\begin{aligned} U^{(N)} Z_x U^{(N)\dagger} &= Z_x^4 (Z_{x+1}^2 Z_{x+2}^3)(Z_{x+3}^2 Z_{x+4}^3)\ldots && (5.94) \\ U^{(N)} X_x U^{(N)\dagger} &= \begin{cases} I_x (X_{x+1} X_{x+2}^3 X_{x+3}^4 X_{x+4}^2)(\ldots)\ldots & \text{if } N = 3 + 4 \times k \\ X_x^4 (X_{x+1}^3 X_{x+2}^4 X_{x+3}^2 X_{x+4})(\ldots)\ldots & \text{if } N = 1 + 4 \times k, \end{cases} && (5.95) \end{aligned}$$

where "…" means that the power pattern in the parenthesis continues through the chain. So unlike the previous example we see that this MPU, 1- maps both Z and X to non-local operators, which means, it has no locally-mapped subalgebra, and 2- maps X differently depending on the length of the chain. This complexity arises because there are two inheren periodicity, one of period 4 and another of period 2.

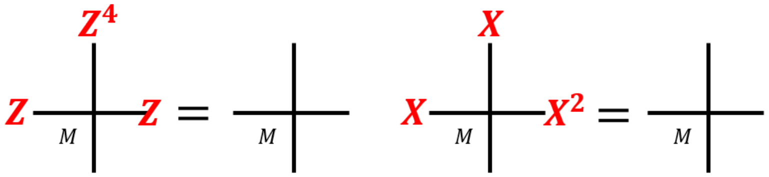
It can be seen clearly that its non-locality index is $d^2 = 25$, which means the support subalgebra of $U^{(N)} \mathcal{A}_x U^{(N)\dagger}$ is in fact the whole algebra on a site far away from x.

Does it have accompanying non-local MPO symmetries as we conjectured? Indeed it does. It has a $Z_5 \times Z_5$ MPO symmetry group generated by the following two symmetries:



. (5.96)

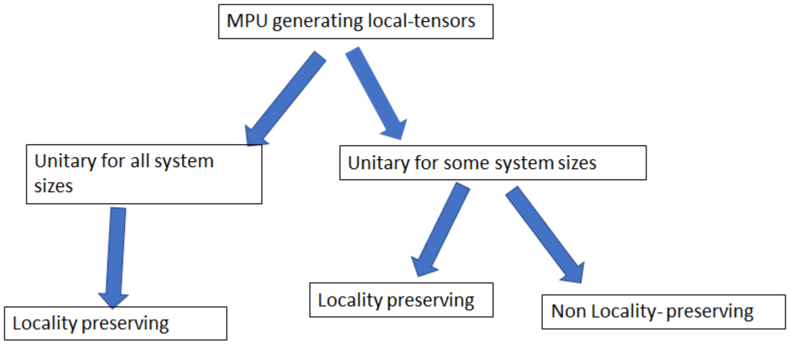
Using this, and two other local symmetries, namely,



. (5.97)

Is locality-preservation a sufficient condition for $\mathbb{N}$-unitarity?

We proved (theorem 7) that $\mathbb{N}$-unitarity is a sufficient condition for locality-preservation. That is, if a local tensor generates unitaries for all system sizes, then the MPUs so generated are bound to be locality preserving. But is the reverse true as well? If a local-tensor generates locality-preserving MPUs for infinite system sizes, does it have to generate MPU for *all* system sizes? It can be easily seen that the answer is, no. The simplest counter example is the T tensor of any of these non-locality preserving local-tensors. This T tensor is unitary (in fact, identity) only for certain system sizes, but is not unitary for other. But of course it is locality-preserving since it is just identity. Hence the general form 5 does not exhuast all locality-preserving MPUs, if we are working with a fixed system size. With this understanding we see that the MPUs generated by finite bond dimensional local-tensors can be categorized as shown in the following diagram:


MPU generating local-tensors
Unitary for all system sizes
Unitary for some system sizes
Locality preserving
Locality preserving
Non Locality- preserving

(5.98)

5.7 Summary and Outlook

In this chapter we investigated the general notion of unitary matrix product operators. We found that there is a necessary and sufficient condition (the N-unitarity condition) for a local-tensor to generate a unitary operator of a given length. This condition simply depends on the spectrum of transfer matrices of the local-tensor M and that of its T-tensor. This allowed us to categorize the locality-preserving MPUs as the particular solutions to these conditions that satisfy it for all lengths. Which in turn implied that such MPUs have to be locality-preserving.

But the N-unitarity condition has infinitely many other solutions as well that do not fall into the locality-preserving category. We found that these MPUs map local algebras to global algebra. We identified certain MPO-type symmetries in such MPUs which was responsible for this non-local mapping. It suggests that 1D unitary operators can have a ‘long-range order’ somewhat similar to those found in 2D topological states.

We draw parallels with the MPS theory as to what characterizes the general behavior of an MPU. In MPS theory if the local-tensor is injective, the MPS is known to have short ranged correlations. Similarly, we found that if the local T-tensor is injective, the MPU is a short-ranged map on local algebras.

There are still some open questions and future directions. The main remaining question is what are other solutions to the N-unitarity conditions? ALl solutions we found were either locality-preserving or periodic. It would be interesting to find an example where the transfer matrix of the T-tensor has several non-zero eigenvalues with a non-degenerate highest eigenvalue. Such an MPU would imply a mapping of local operators with exponentially decaying terms with the distance. Such and MPU would be able to represent time evolution under a generic 1D Hamiltonian.

Are there solutions to N-unitarity conditions that work for finite number of system sizes? We have not found such a solution but they might be possible. Another interesting observation is that somehow all MPUs that were a long-ranged maps were on Hilbert space of dimension more than 2. We do not know if there are such MPUs for qubits a chain of qubits as well.

One obvious remaining question is to understand the physics behind the non-locally mapping MPUs. In particular, what are the consequence of their long-ranged behavior and if they form the symmetry groups of certain 1D phases. We suspect they might be related to the system of 1D Hamiltonians whose ground states are known to break area law[48, 71].

APPENDIX

A.1 Dependence of S_{topo} on boundary conditions in cylindrical geometry

Topological entanglement entropy calculation is done by calculating the entanglement entropy of a subsystem A. When the boundary of A consists of topologically trivial loops, for example when A has a disc geometry, S_{topo} is known to depend only on the total quantum dimension D, $S_{\text{topo}} = \log D$. However when the boundary of A consists of non-contractible topologically non-trivial loops, for example when a torus or cylinder is divided into two cylinders, it has been shown by Zhang *et al.* [56] that S_{topo} also depends on the linear combination of ground states. For a ground state wave function on a torus

$$|\Psi\rangle = \sum_a c_a |\Xi_a\rangle \tag{5.99}$$

where the sum is over the degenerate ground states labeled by quasi-particles of the model, the nth Rényi entropy is given by

$$\begin{aligned} S_n &= \alpha_n L - S_{\text{topo}}, & (5.100) \\ S_{\text{topo}} &= 2\log D - \frac{1}{1-n}\log\left(\sum_a p_a^n d_a^{2(1-n)}\right) & \\ && (5.101) \end{aligned}$$

where d_a is the quantum dimension of ath quasi-particle and $p_a = |c_a|^2$. $|\Xi_a\rangle$ are special basis for which S_{topo} is maximal, or entanglement entropy is minimal. These states are called the *Minimum Entropic States (MES)* . It was shown that MES correspond to eigenstates of Wilson-loop operators along the entanglement cut.

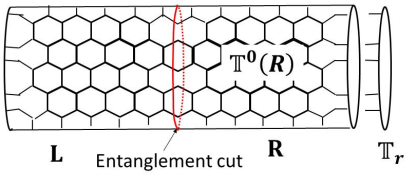


Figure 5.1: We calculate entanglement entropy of the right-half of the cylinder with a certain boundary condition $\mathbb{T}_r$. The entanglement cut is in the middle of the cylinder.

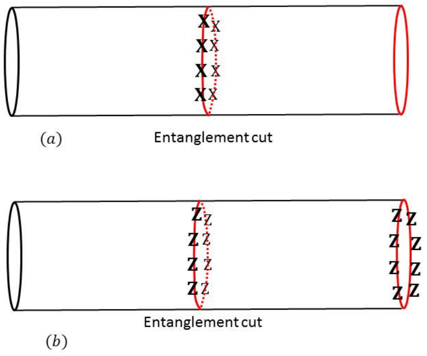


Figure 5.2: MESs are eigenstates of different Wilson loop operators at the entanglement cut. (a)For fixed point single-line TNR, the state on the cylinder is always in +1 eignestate of X-loop, as it identically disappears. (b)The state is also in +1 eigenstate of simultaneous operation of two Z-loops, one at the entanglement cut, other at the right-most boundary. It implies, we can be in two MESs depending on the the boundary tensor choice. If the boundary tensor is in +1 eigenstate of the boundary Z loop, then the state is in +1 eigenstate of the entanglement-cut Z loop. Similarly, if the boundary tensor is in -1 eigenstate of the boundary Z loop, then the state is in -1 eigenstate of the entanglement-cut Z-loop.

This dependence of S_{topo} on the ground state is of crucial importance to us since we have used cylinder with a boundary for S_{topo} calculations. So, numerically obtained S_{topo} contain information about the boundary as well. For example, consider the toric code.

$$S_{\text{topo}} = 2\log 2 - \frac{1}{1-n}\log(p_1^n + p_2^n + p_3^n + p_4^n)$$

When $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$ we get $S_{\text{topo}} = 0$ although the the topological order is not lost. So one has to be careful using S_{topo} as an indicator of topological order.

Let's first take the example of the single-line TNR of the toric code. See Fig. 5.2. We put our system on a cylinder with some boundary conditions to be determined later. The entanglement cut is in the middle of the cylinder, and the right half cylinder, denoted as R, is the subsystem whose entanglement entropy we are calculating (see Fig. 5.1). The four MES correspond to four eigen states of e and m Wilson-loops on the entanglement cut. But, since e-Wilson loop is a zero-string operator, the state is always in its +1 eigenstate (Fig. 5.2(a)). So we have access to only two MES corresponding to ±1 eigenstates of m-Wilson loop. We also know that the state is in +1 eigenstate of the $Z^{\otimes}_{\partial R} = Z^{\otimes L}_{ec} \otimes Z^{\otimes L}_{r}$, where subscript ec stands for the loop at entanglement cut, and r stands for the loop at the right boundary of R.

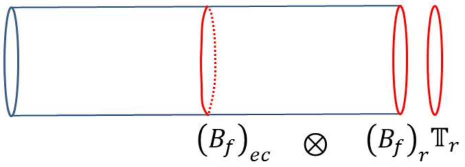


Figure 5.3: Bulk double tensor is a sum of tensor product between $(B_f)_{ec}$ (B_f on the entanglement cut) and $(B_f)_r$ (B_f on the right boundary). So, when we contract a boundary tensor $\mathbb{T}_r$ with the bulk tensor, it contract with $(B_f)_r$ giving a scalar c_f. So resulting tensor is $\mathrm{Ev}(\mathbb{T}(R)\mathbb{T}_r) = \sum_f c_f(B_f)_{ec}$. Consequently, S_{topo} using Eq. (3.108) is simply $\log\left(\sum_f \frac{c_f^2}{c_0^2}\right)$.

Since the state is in +1 eigenstate of $Z_{ec}^{\otimes L} \otimes Z_r^{\otimes L}$ (see Fig. 5.2)(b), the state can be either in +1 eigen-state of both $Z_r^{\otimes L}$ and $Z_{ec}^{\otimes L}$ or in -1 eigenstate of the both. The boundary tensor determines which eigenstate of $Z_r^{\otimes L}$ the wave function is in, and consequently also which eigenstate of $Z_{ec}^{\otimes L}$. This is how the boundary tensors and MES are connected. Since we have access to only two MES

$$S_{\text{topo}} = \log 2 - \frac{1}{1-n}\log(p_1^n + p_2^n). \tag{5.102}$$

A similar analysis follows in the double-line TNR, with the role of e and m Wilson loop operators reversed: now the state is always in the +1 eigen state of m-Wilson loop and the two MES correspond to the two eigenstates of e Wilson loop at the entanglement cut, which in turn depends on the boundary tensors.

We saw in the section 2.2, $\rho_R = N\sigma_b^2$, where

$$\sigma_b = \mathbb{T}^0(R)\mathbb{T}_r, \tag{5.103}$$

where $\mathbb{T}_r$ denotes the double tensor on the boundary. We know that, up to an irrelevant normalization constant,

$$\begin{aligned} \mathbb{T}^0(R) &= \sum_f d_f^{\chi_R} B_f(\partial R) \\ &= (B_0)_{ec} \otimes (B_0)_r + (B_1)_{ec} \otimes (B_1)_r, \end{aligned} \tag{5.104}$$

where $B_0 = I^{\otimes L}$ and $B_1 = Z^{\otimes L}$ for the single-line TNR and $B_1 = X^{\otimes L}$ for the double-line TNR. Let's say the boundary double tensor $\mathbb{T}_r$ contracts with $(B_f)_r$ to produce the constants c_f (see Fig. 5.3)

$$\begin{aligned}\sigma_b &= \left((B_0)_{ec} \otimes (B_0)_r + (B_1)_{ec} \otimes (B_1)_r\right) \mathbb{T}_r \\ &= c_0 (B_0)_{ec} + c_1 (B_1)_{ec} \\ &= c_- B_- + c_+ B_+. \end{aligned} \tag{5.105}$$

where $c_0 = (B_0)_r \mathbb{T}_r$, $c_1 = (B_1)_r \mathbb{T}_r$ and $B_\pm = \frac{1}{2}(B_0 \pm B_1)$ and $c_\pm = (c_0 \pm c_1)$. Note that $B_\pm$ satisfy the following:

$$B_\pm^2 = B_\pm, \quad \mathrm{Tr}(B_\pm) = 2^{L-1}. \tag{5.106}$$

With this, we get the normalized density matrix as,

$$\rho_R = \frac{1}{2^L}\left(\frac{c_-^2}{c_-^2 + c_+^2} B_- + \frac{c_+^2}{c_-^2 + c_+^2} B_+\right) \tag{5.107}$$

$$= \frac{1}{2^L}\left(p_- B_- + p_+ B_+\right). \tag{5.108}$$

The nth Renyi entropy is,

$$\begin{aligned} S_n(\rho_R) &= \frac{1}{1-n} \log \mathrm{Tr}(\rho_R^n) \\ &= \frac{1}{1-n} \log \mathrm{Tr}\left(\frac{1}{2^{nL}}(p_-^n B_- + p_+^n B_+)\right) \\ &= \frac{1}{1-n} \log \left(\frac{1}{2^{nL}}(p_-^n 2^{L-1} + p_+^n 2^{L-1})\right) \\ &= L \log 2 - \left(\log 2 - \frac{1}{1-n} \log(p_-^n + p_+^n)\right). \end{aligned} \tag{5.109}$$

Comparing it with the MES formula in Eq. (5.102), we see that $p_1 = p_- = c_0 - c_1$ and $p_2 = p_+ = c_- + c_+$. So the state is an MES if $p_\pm = 0 \Rightarrow c_0 = \pm c_1$ for which we get maximal topological entanglement entropy, $S_{\text{topo}} = \log 2$. This illustrates the direct dependence of S_{topo} on $\mathbb{T}_r$.

Of course the above analysis is done for the RG fixed point tensors only. We have to choose a boundary double tensor $\mathbb{T}_r$ such that S_{topo} is truly indicative for topological order, or lack of it, for both RG fixed point and varied tensors. We choose the following boundary tensor for our numerical calculations: for any tensor network, fixed point or varied, We use a ‘smooth boundary condition’. It is explained in the Fig. 5.4. First we will explain it for the triple-line tensors. For double-line and single-line an appropriately reduced version of T_b will be used. Note that we haven’t drawn the physical index explicitly and it should be understood the same as the

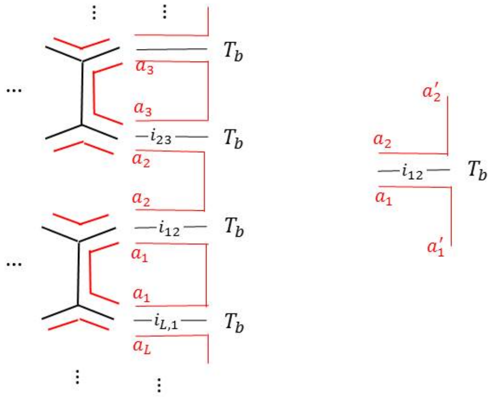


Figure 5.4: Smooth boundary condition for triple-line tensor network. Tensors T_b are used on the boundary. T_b has 5 virtual legs, $a_1, a_1', a_2, a_2', i_{12}$ and 1 physical leg, i_{12}. Physical leg and the middle leg take the same values. We assign a particular value to the components of this tensor, $(T_b)^{i_{12}}_{i_{12}a_1a_1';a_2a_2'} = \delta_{i_{12},0}\delta_{a_1,a_1'}\delta_{a_2,a_2'}\delta_{a_1a_2i_{12}}$.

middle index (the index in black color). So the boundary tensor T_b has four virtual indices, and we fix its components to be,

$$(T_b)^{i_{12}}_{i_{12}a_1a_1';a_2a_2'} = \delta_{i_{12},0}\delta_{a_1,a_1'}\delta_{a_2,a_2'}\delta_{a_1a_2i_{12}} \tag{5.110}$$

that is, we put the physical/middle index to zero (vacuum) and allow the plaquette legs to vary with this restriction. For double-line we don't have a middle leg, but we can simply put the physical leg to 0. For single-line we only have the middle legs and we put them to zero.

Before we discuss why we choose this particular boundary, let us calculate what S_{topo} we are supposed to get with this particular choice of boundary tensor. For that, we need to calculate $c_f = B_f \mathbb{T}_r$. Note that $\delta_{a_j,a_{j+1},0}$ implies $a_j = a_{j+1}$. So the double tensor $\mathbb{T}_r$ is

$$\mathbb{T}_r = \sum_{a,b} |a, a, a..; 000..\rangle \langle b, b, b, ...; 000...| \, . \tag{5.111}$$

So

$$\begin{aligned} c_f &= \mathrm{Ev}(B_f \mathbb{T}_r) \\ &= \sum_{a,b} \prod_{j=1}^{m} G^{b,b,0}_{a,a,f} (d_a d_b)^{\frac{1}{2}} \\ &= \sum_{a,b} \prod_{j=1}^{m} \delta_{a,b,f} \\ &= \sum_{a,b} \delta_{a,b,f}. \end{aligned} \tag{5.112}$$

Then using Eq. (3.108), S_{topo} is simply $\log(\sum_f \frac{c_f^2}{c_0^2})$. For the toric code, and double semion models $c_0 = c_1 = 2$, so we get $S_{\text{topo}} = \log 2$. For the double Fibonacci model, however, we get

$$c_0 = \sum_{a,b} \delta_{a,b,0} = \delta_{0,0,0} + \delta_{1,1,0} = 2 \tag{5.113}$$

$$c_1 = \sum_{a,b} \delta_{a,b,1} = \delta_{1,0,1} + \delta_{0,1,1} + \delta_{1,1,1} = 3. \tag{5.114}$$

$$\tag{5.115}$$

So we get $S_{\text{topo}} = \log(1 + \frac{3^2}{2^2}) = \log(1 + \frac{9}{4})$, which is consistent with our numerical result.

There are mainly two reasons why we choose this particular boundary condition
1- This is a very simple boundary condition which gives us a precise analytical value of the topological entanglement entropy (namely, $\log(\sum_f \frac{c_f^2}{c_0^2})$, with c_f given in Eq. (5.112)) against which numerical calculations can be checked.
2- Though situation for non-abelian cases is more complicated, this boundary is definitely MPO symmetric for abelian models. That is, we expect the tensor network state to be an MES with maximal S_{topo} (=log D).

Numerical calculations of S_{topo} will be checked against the analytical result in Eq. 5.112. Now the remaining question is about the trustworthiness of the same calculation for varied tensor. That is, how can we deduce the conclusion about the topological order of the varied tensor by $S_{\text{topo}}(\epsilon)$? The first point is, if $S_{\text{topo}}(\epsilon) = S_{\text{topo}}(0)$, then we can definitely say that the state is in the same topological phase. But $S_{\text{topo}}(\epsilon) = 0$ needs to be further verified as it might be because of the particular boundary conditions imposed. To verify, we will test for S_{topo} dependence on infinitesimal variation on the boundary tensors. The reason for this is made clear

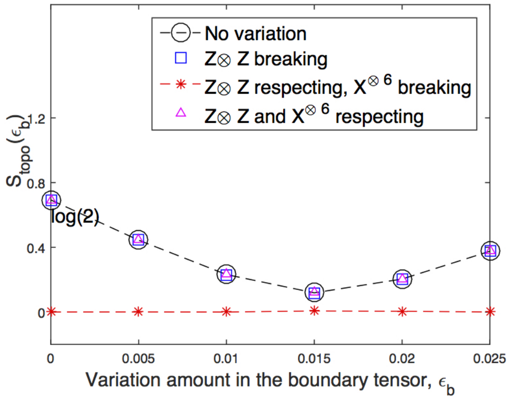


Figure 5.5: Dependence of S_{topo} on boundary condition for toric-code double line TNR. We start with the boundary tensor, T_b, shown in Fig. 5.4. We add a random variation $\epsilon_b T_b^r$ to T_b and calculate $S_{\text{topo}}(\epsilon_b)$ for random bulk variations in different subspaces. We keep T_b^r fixed and increase the variation strength ϵ_b. We see all classes of stable bulk variations have the same S_{topo} for each ϵ_b as the fixed point (no-variation) tensor. And the unstable class of bulk variation shows no dependence on ϵ_b. It shows that stable variations indeed are in the same topological phase as the RG fixed point state, and unstable variation is a trivial phase.

by looking at the dependence of S_{topo} on p_1, p_2 etc. So, if the state indeed has a topological order, S_{topo} should sensitively depend on the $c_0 = (B_0)_r \mathbb{T}_r, c_1 = (B_1)_r \mathbb{T}_r$. If the state has lost its topological order, S_{topo} will remain zero under any changes of the boundary tensor. This way, we can avoid getting any 'accidental $S_{\text{topo}} = 0$' cases, for example when $p_1 = p_2 = \frac{1}{2}$.

One such verification is shown in Fig. 5.5. We first fix the boundary tensor to be T_b given in Eq. 5.4 and calculate the S_{topo} for variations in $I_V - M_0$, $M_0 - \mathbb{M}$ and $\mathbb{M}$ subspaces added to the fixed point bulk tensor. Now we add an infinitesimal random variation to the boundary tensor, $T_b \longrightarrow T_b + \epsilon_b T_b^r$. ϵ_b (different from ϵ, which the bulk variation strength) is the strength of the boundary variation. We increase ϵ_b slowly and for each value of the ϵ_b we calculate $S_{\text{topo}}(\epsilon)$ for random bulk variations in different subspaces. Fig. 5.5 shows S_{topo} as a function of ϵ_b for bulk variations in different subspaces. (The bulk variation strength ϵ is kept fixed throughout). We observe that

1- The variations which are unstable (i.e. $S_{\text{topo}} = 0$) for T_b, continue to be unstable for $T_b + \epsilon_b T_b^r$ for all values of ϵ_b. It implies that we get $S_{\text{topo}} = 0$ for these variation because the bulk topological order is indeed destroyed and not because of a specific boundary tensor chosen which gave an accidental zero.

2- The variations which are stable (i.e. $S_{\text{topo}} = \log 2$) for T_b have the same value

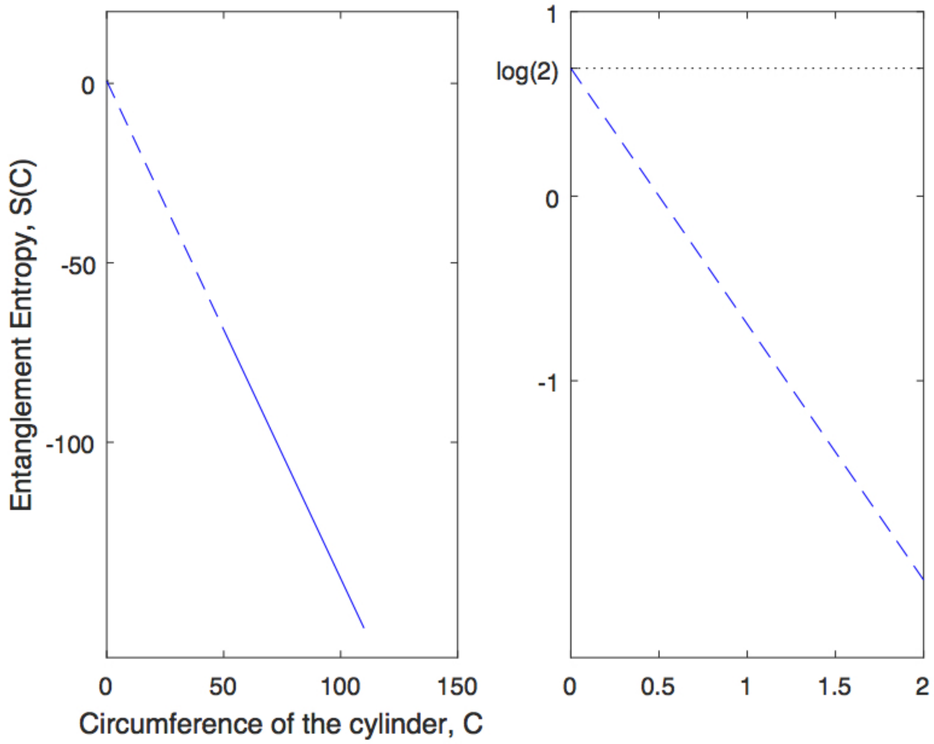


Figure 5.6: Calculation of S_{topo} for single-line toric code fixed point tensor network state. We fix half cylinder length as $L = 500$. Circumference is varied from 50 to 110. S varies linearly with C. This line is extrapolated back to $C = 0$. Its intersection with the y-axis gives S_{topo}. Right figure is a zoomed in version of the left figure to show the intersection point clearly. We find $S_{\text{topo}} \approx \log(2)$.

of S_{topo} as the fixed point tensor for all boundary tensors. It implies that tensor network state with these variations indeed have the same topological order as the fixed point tensor network state. Though this verification is shown for double-line toric code only, we find the same behavior for all numerical calculations presented in this paper.

It should be noted that any strictly positive value of S_{topo} (assuming sufficiently large cylinder was considered) is a sufficient condition for topological order but it is not a necessary condition. So all we need to do is to avoid getting accidental zeros.

A.2 Details of numerical calculations

Here we will provide the various numerical details and data regarding the numerical calculations whose results were presented in the main text.

First, we will show convergence of numerical calculation of S_{topo}. We choose the

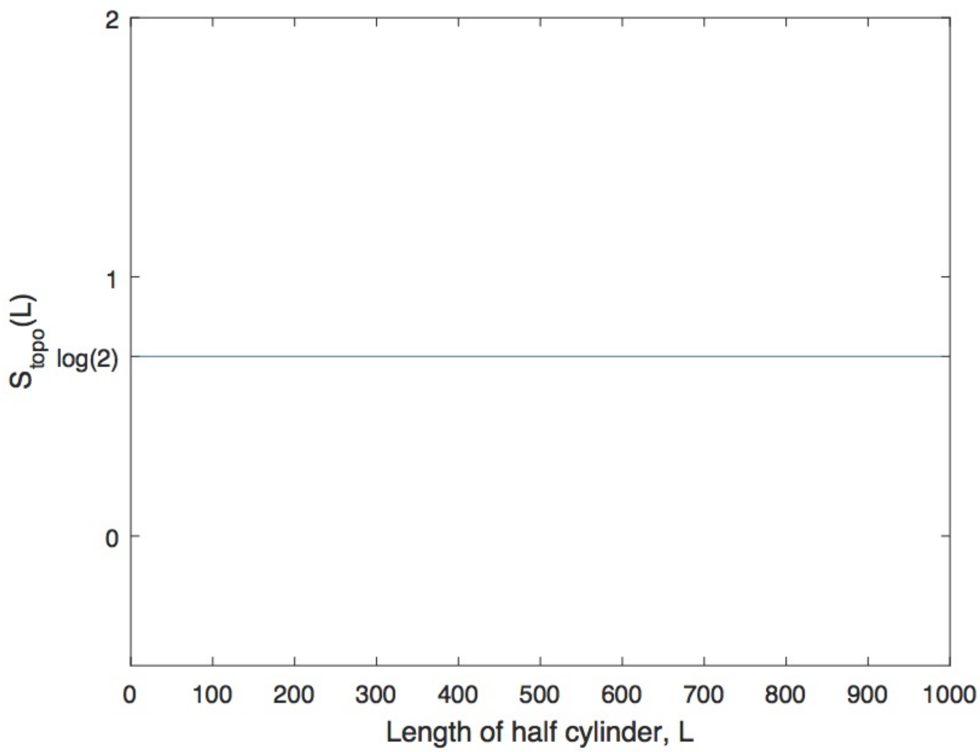


Figure 5.7: S_{topo} was calculated for a fixed half cylinder length, $L = 500$, in Fig. 5.6. We now vary L from 10 to 1000. We see that S_{topo} is converged even for small values of L. So one does not need to large cylinder length to get the right S_{topo} value. It is expected as it is an RG fixed point tensor network state.

simplest case, the single-line TNR of toric code. We first repeat the algorithm described in section 2.2 in simple words here for convenience. In the first step, the transfer matrix is calculated using the tensor given (fixed point or varied). Then we choose a specific boundary double tensor as explained in the appendix 5.7. We apply the transfer matrix on this boundary double tensor and approximate the resulting tensor as an MPS of bond dimensions $D_{\text{cut}} = 8$. We apply transfer matrix again and approximate the resulting tensor as an MPS of bond dimension 8. We repeat this process and each repetition physically corresponds to increasing the longitudinal length of the our cylindrical subsystem by one unit. Let's say we repeat this process until the length of the half cylinder subsystem is equal to L. This process gives us the virtual density matrix σ, and assuming the mirror symmetry of transfer matrix, the physical reduced density matrix of the half cylinder is $\rho_L \propto \sigma^2$. With this reduced density matrix we calculate the entanglement entropy S of the half cylinder subsystem for different circumferences C. We plot $-S$ vs C and extrapolate it to

$C = 0$ which gives us the topological entanglement entropy $S_{\text{topo}} = S(C = 0)$. In principle, one needs to take infinitely large cylinder to achieve the precise value of S_{topo}. Practically, we need to keep increasing L until we get a fixed point MPS and keep increasing C until the S_{topo} value converges to a fixed point.

Let's first look at the calculation for the single-line toric code fixed point tensor in Eq. 2.11. Half cylinder length is fixed at $L = 500$. C is varied from 50 to 110. Fig. 5.6 shows the entanglement entropy S vs the circumference C. We get a straight line which is extrapolated to $C = 0$. The right figure is a zoomed in version of the left figure to see clearly where the extrapolated line crosses the y-axis. We get $S_{\text{topo}} = S(C = 0) \approx \log(2)$ as expected. Fig. 5.7 shows the dependence of S_{topo} on the half cylinder length L. We see that there is no dependence, that is, the fixed point MPS is achieved immediately. It is expected as it is an RG fixed point tensor network state.

Now we look at the calculation for single-line toric code fixed point tensor *varied with an MPO symmetry breaking tensor*. Remember that it is claimed in the main text that this is a trivial state. The variation strength is fixed at $\epsilon = 0.01$. Half cylinder length is fixed at $L = 500$. C is varied from 50 to 110. Fig. 5.8 shows entanglement entropy S vs the circumference C. We get a straight line which is extrapolated to $C = 0$. The right figure is a zoomed in version of the left figure to show clearly where the extrapolated line crosses the y-axis. We see $S_{\text{topo}} \approx 0$. To see the effect of cylinder length we calculate S_{topo} again but with different cylinder lengths. The results are shown in Fig. 5.9. We see that S_{topo} is $\log(2)$ for small cylinders but converges to zero as the length is increased. Comparing it to Fig. 5.7 we see that, unlike the fixed point case, we need to consider a large enough cylinder ($L > 600$ in this case) to calculate the correct S_{topo} value for the non-fixed point tensor network state.

Finally we show the effect of variation strength, ϵ, on the convergence. In the above calculation we fixed $\epsilon = 0.01$. Now we vary ϵ from 0.01 to 0.02 (making sure it is well below any critical points) and calculate corresponding convergence plots similar to Fig. 5.9. The results are shown in Fig. 5.10. We see that the strength of the variation has a huge effect on convergence. Bigger variations lead to faster convergence.

Though we have presented details of calculation only for one case (single-line toric code TNR), it should be understood that similar patterns are followed in all other cases. For completeness, we present the numerical data plotted in the main text and

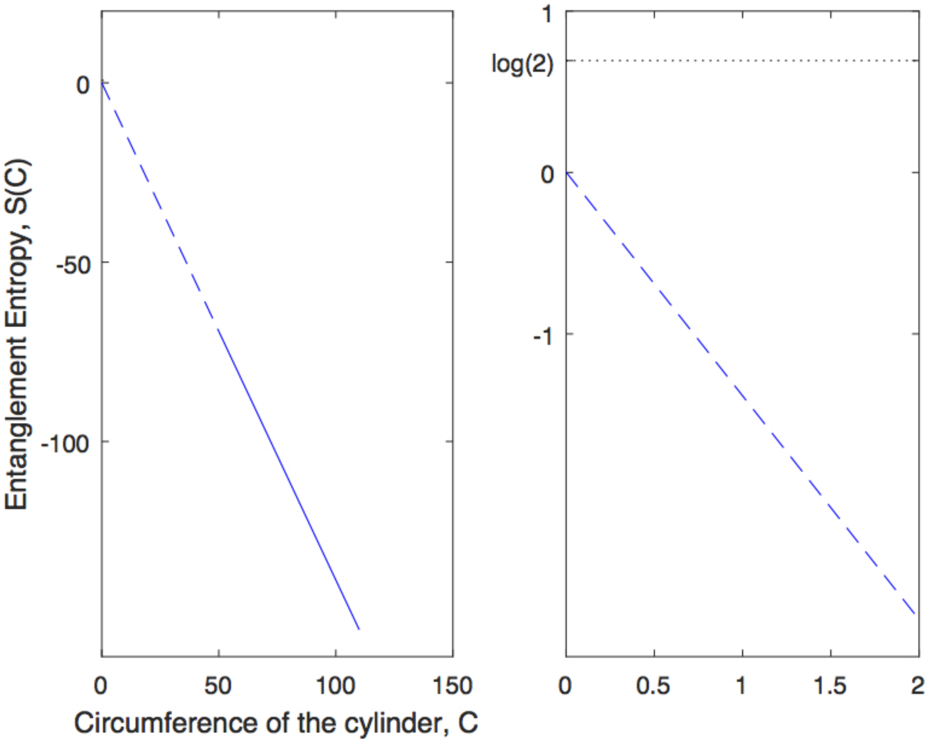


Figure 5.8: Calculation of S_{topo} for a state represented by single-line toric code fixed point tensor varied with an MPO violating tensor. The strength of the variation is fixed at $\epsilon = 0.01$. We fix half cylinder length as $L = 500$. Circumference is varied from 50 to 110. S varies linearly with C. This line is extrapolated back to $C = 0$. Its intersection with the y-axis gives S_{topo}. Right figure is a zoomed in version of the left figure to show the intersection point clearly. We find $S_{\text{topo}} \approx 0$, that is, it is a trivial state.

the relevant parameters used in each case.

Single-line TNR toric code

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 8$ at each step of the iteration. The starting MPS is as explained in the appendix 5.7. The strength of the variations is fixed at $\epsilon = 0.01$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110.

The following table contains the exact values of the S_{topo} plotted in Fig. 2.5.

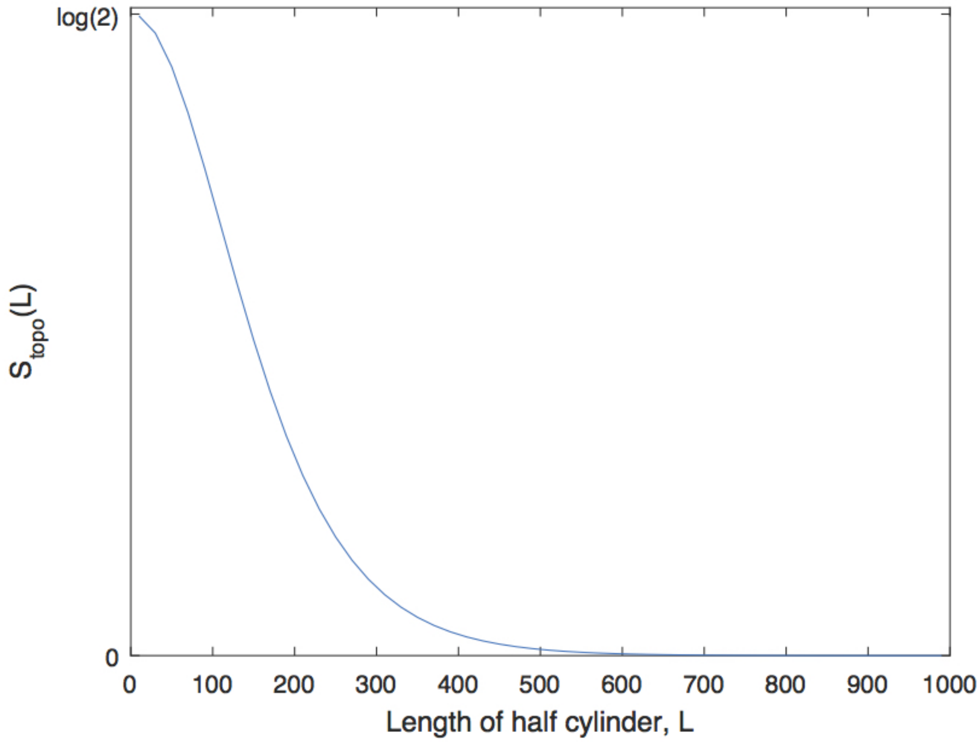


Figure 5.9: S_{topo} was calculated for a fixed half cylinder length, $L = 500$, in Fig. 5.8. We now vary L from 10 to 1000. We see that S_{topo} is close to log(2) for small cylinders but converges to zero cylinder length L is increased from 1 to 1000. So it is indeed a topologically trivial state.

No Variation	0.6931
$Z^{\otimes 3}$ respecting variations	0.6931 0.6931 0.6931 0.6931 0.6931 0.6931 0.6931 0.6931 0.6931
$Z^{\otimes 3}$ violating variations	$10^{-12}\times$ 0.9095 0 -0.4547 -0.4547 0 0 0.9095 0.4547 -0.4547

Double-line TNR toric code

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 16$ at each step of the iteration. The starting MPS is as explained in the appendix 5.7. The strength of the variations is fixed at $\epsilon = 0.01$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The

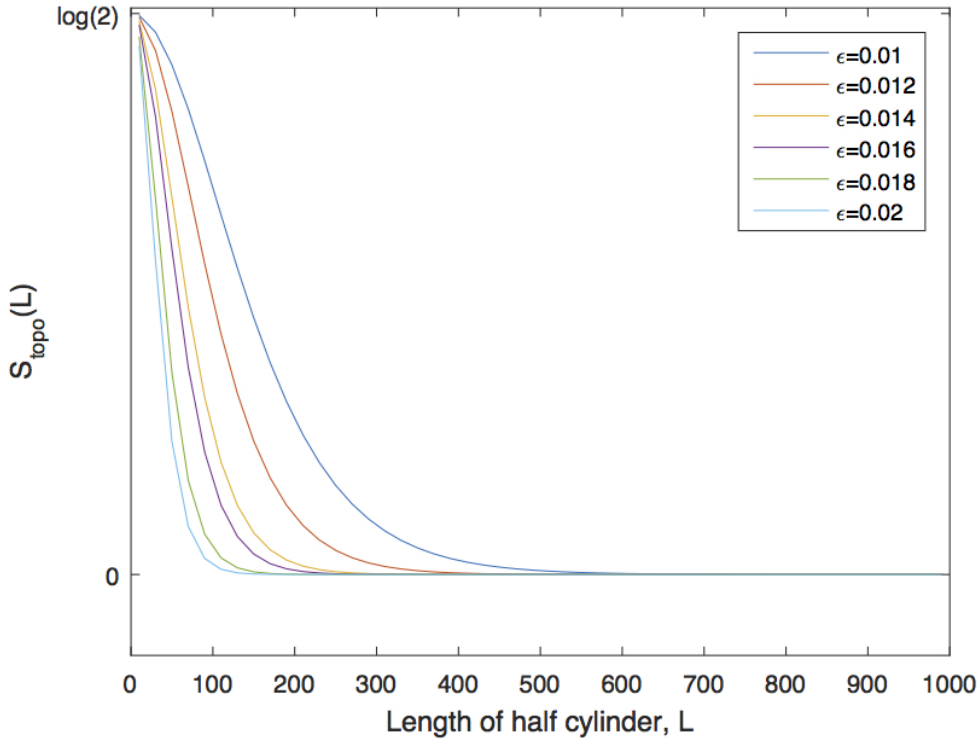


Figure 5.10: The variation strength ϵ affects convergence. The higher the variation strength (as long as it is below any critical points) the faster the convergence with the length of the size of the system.

circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 2.6.

No Variation	0.6931
$Z \otimes Z$ breaking variations	0.6931 0.6931 0.6931 0.6931 0.6931
$Z \otimes Z$ respecting, $X^{\otimes 6}$ breaking variations	0.0000 0.0015 0.0000 0.0000 0.0002
$Z \otimes Z$ and $X^{\otimes 6}$ respecting variations	0.6931 0.6931 0.6931 0.6931 0.6931

Double-line TNR double semion code

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 16$ at each step of the iteration. The starting MPS is as explained in the appendix 5.7. The strength of the variations is fixed at $\epsilon = 0.01$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. The following table contains the exact values of the S_{topo} plotted in Fig. 2.12.

No Variation	0.6931
$Z \otimes Z$ breaking variations	0.6931 0.6931 0.6931 0.6931 0.6931
$Z \otimes Z$ respecting, $X^{\otimes 6}$ breaking variations	0.0133 0.0047 0.0191 0.0086 0.0063
$Z \otimes Z$ and $X^{\otimes 6}$ respecting variations	0.6931 0.6931 0.6931 0.6931 0.6931

Triple-line toric code

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 16$ at each step of the iteration. The starting MPS is as explained in the appendix 5.7. The strength of the variations is fixed at $\epsilon = 0.2$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 3.1.

No Variation	0.6931
Variations in $I_V - M_0$	0.6931 0.6931 0.6931 0.6931 0.6931
Variations in $M_0 - \mathbb{M}$	$10^{-3}\times$ 0.2467 0.0986 0.2658 0.0257 0.0005
Variations in $\mathbb{M}$	0.6931 0.6931 0.6931 0.6931 0.6931

Triple-line double-semion

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 16$ at each step of the iteration. The starting MPS is as explained in the appendix 5.7. The strength of the variations is fixed at $\epsilon = 0.2$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 3.3.

No Variation	0.6931
Variations in $I_V - M_0$	0.6932 0.6931 0.6932 0.6931 0.6932
Variations in $M_0 - \mathbb{M}$	$10^{-7}\times$ 0.7877 0.0849 0.0003 0.0006 0.0000
Variations in $\mathbb{M}$	0.6931 0.6931 0.6931 0.6931 0.6931

Triple-line Fibonacci model

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 16$ at each step of the iteration. The starting MPS is as explained in the appendix 5.7. The strength of the variations is fixed at $\epsilon = 0.1$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 2000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 3.5.

No Variation	1.1787
Variations in $I_V - M_0$	1.1779 1.1776 1.1774 1.1778 1.1779
Variations in $M_0 - \mathbb{M}$	$10^{-7}\times$ -0.2330 0.2841 0.0517 0.0335 0.0299
Variations in $\mathbb{M}$	1.1535 1.1623 1.1556 1.1386 1.1667

BIBLIOGRAPHY

[1] D. Gross, V. Nesme, H. Vogts, and R. F. Werner, Communications in Mathematical Physics **310**, 419 (2012).

[2] J. Nagamatsu, N. Nakagawa, T. Muranaka, Y. Zenitani, and J. Akimitsu, **410**, 63 (2001).

[3] V. Kalmeyer and R. B. Laughlin, Phys. Rev. Lett. **59**, 2095 (1987).

[4] X. G. Wen, F. Wilczek, and A. Zee, Phys. Rev. B **39**, 11413 (1989).

[5] X. G. Wen, Phys. Rev. B **40**, 7387 (1989).

[6] D. C. Tsui, H. L. Stormer, and A. C. Gossard, Phys. Rev. Lett. **48**, 1559 (1982).

[7] R. B. Laughlin, Phys. Rev. Lett. **50**, 1395 (1983).

[8] X.-G. Wen, Advances in Physics **44**, 405 (1995), http://dx.doi.org/10.1080/00018739500101566 .

[9] N. Read and S. Sachdev, Phys. Rev. Lett. **66**, 1773 (1991).

[10] B. I. Halperin, Phys. Rev. Lett. **52**, 1583 (1984).

[11] G. Moore and N. Read, Nuclear Physics B **360**, 362 (1991).

[12] A. Kitaev, Annals of Physics **303**, 2 (2003).

[13] M. A. Levin and X.-G. Wen, **71**, 045110 (2005), cond-mat/0404617 .

[14] X. Chen, Z.-X. Liu, and X.-G. Wen, Phys. Rev. B **84**, 235141 (2011).

[15] O. Buerschaper, Annals of Physics **351**, 447 (2014).

[16] M. B. Şahinoğlu, D. Williamson, N. Bultinck, M. Mariën, J. Haegeman, N. Schuch, and F. Verstraete, ArXiv e-prints (2014), arXiv:1409.2150 [quant-ph] .

[17] D. J. Williamson, N. Bultinck, M. Mariën, M. B. Şahinoğlu, J. Haegeman, and F. Verstraete, Phys. Rev. B **94**, 205150 (2016).

[18] N. Bultinck, M. Mariën, D. Williamson, M. Şahinoğlu, J. Haegeman, and F. Verstraete, Annals of Physics **378**, 183 (2017).

[19] C. L. Kane and E. J. Mele, Phys. Rev. Lett. **95**, 226801 (2005).

[20] B. A. Bernevig, T. L. Hughes, and S.-C. Zhang, Science **314**, 1757 (2006), cond-mat/0611399 .

[21] L. Fu, C. L. Kane, and E. J. Mele, Phys. Rev. Lett. **98**, 106803 (2007).

[22] A. Mesaros and Y. Ran, **87**, 155115 (2013), arXiv:1212.0835 [cond-mat.str-el] .

[23] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, Phys. Rev. Lett. **59**, 799 (1987).

[24] R. Raussendorf, D. E. Browne, and H. J. Briegel, Journal of Modern Optics **49**, 1299 (2002), quant-ph/0108118 .

[25] N. Schuch, D. Perez-Garcia, and I. Cirac, Phys. Rev. B **84**, 165139 (2011).

[26] M. Fannes, B. Nachtergaele, and R. Werner, Communications in Mathematical Physics **144**, 443 (1992), 10.1007/BF02099178.

[27] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, Quantum Inf. Comput. **7**, 401 (2007), arXiv:quant-ph/0608197 .

[28] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[29] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[30] N. Schuch and J. I. Cirac, Phys. Rev. A **82**, 012314 (2010).

[31] Z. Landau, U. Vazirani, and T. Vidick, ArXiv e-prints (2013), arXiv:1307.5143 [quant-ph] .

[32] F. Pollmann, A. M. Turner, E. Berg, and M. Oshikawa, Phys. Rev. B **81**, 064439 (2010).

[33] F. Pollmann, E. Berg, A. M. Turner, and M. Oshikawa, Phys. Rev. B **85**, 075125 (2012).

[34] X. Chen, Z.-C. Gu, and X.-G. Wen, Phys. Rev. B **83**, 035107 (2011).

[35] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, Phys. Rev. Lett. **93**, 207204 (2004).

[36] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, New Journal of Physics **12**, 025012 (2010).

[37] J. Haegeman and F. Verstraete, ArXiv e-prints (2016), arXiv:1611.08519 [cond-mat.str-el] .

[38] M. Fannes, B. Nachtergaele, and R. Werner, Communications in Mathematical Physics **144**, 443 (1992), 10.1007/BF02099178.

[39] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[40] F. Verstraete, V. Murg, and J. Cirac, Advances in Physics **57**, 143 (2008).

[41] G. Vidal, ArXiv e-prints 0912.1651 (2009), arXiv:0912.1651 [cond-mat.str-el] .

[42] Z.-C. Gu, M. Levin, B. Swingle, and X.-G. Wen, **79**, 085118 (2009), arXiv:0809.2821 [cond-mat.str-el] .

[43] O. Buerschaper, M. Aguado, and G. Vidal, Phys. Rev. B **79**, 085119 (2009).

[44] S. Yan, D. A. Huse, and S. R. White, Science **332**, 1173 (2011).

[45] H.-C. Jiang, Z. Wang, and L. Balents, Nat Phys **8**, 902 (2012).

[46] S. Depenbrock, I. P. McCulloch, and U. Schollwöck, Phys. Rev. Lett. **109**, 067201 (2012).

[47] X. Chen, B. Zeng, Z.-C. Gu, I. L. Chuang, and X.-G. Wen, Phys. Rev. B **82**, 165119 (2010).

[48] S. Bravyi, M. B. Hastings, and S. Michalakis, Journal of Mathematical Physics **51**, 093512 (2010).

[49] M. B. Şahinoğlu, D. Williamson, N. Bultinck, M. Mariën, J. Haegeman, N. Schuch, and F. Verstraete, ArXiv e-prints (2014), arXiv:1409.2150 [quant-ph] .

[50] A. Kitaev and J. Preskill, Phys. Rev. Lett. **96**, 110404 (2006).

[51] M. Levin and X.-G. Wen, Phys. Rev. Lett. **96**, 110405 (2006).

[52] Z.-C. Gu, M. Levin, and X.-G. Wen, **78**, 205116 (2008), arXiv:0806.3509 [cond-mat.str-el] .

[53] J. I. Cirac, D. Poilblanc, N. Schuch, and F. Verstraete, Phys. Rev. B **83**, 245134 (2011).

[54] S. T. Flammia, A. Hamma, T. L. Hughes, and X.-G. Wen, Phys. Rev. Lett. **103**, 261601 (2009).

[55] S. Dong, E. Fradkin, R. G. Leigh, and S. Nowling, Journal of High Energy Physics **5**, 016 (2008), arXiv:0802.3231 [hep-th] .

[56] Y. Zhang, T. Grover, A. Turner, M. Oshikawa, and A. Vishwanath, Phys. Rev. **B85**, 235151 (2012), arXiv:1111.2342 [cond-mat.str-el] .

[57] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[58] N. Schuch, I. Cirac, and D. Pérez-García, Annals of Physics **325**, 2153 (2010).

[59] M. Freedman, C. Nayak, K. Shtengel, K. Walker, and Z. Wang, Annals of Physics **310**, 428 (2004).

[60] G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015).

[61] Y. Hu, Y. Wan, and Y.-S. Wu, Phys. Rev. B **87**, 125114 (2013).

[62] C.-H. Lin and M. Levin, Phys. Rev. B **89**, 195130 (2014).

[63] E. Mascarenhas, H. Flayac, and V. Savona, Phys. Rev. A **92**, 022116 (2015).

[64] M. L. Wall and L. D. Carr, New Journal of Physics **14**, 125015 (2012).

[65] D. V. Else and C. Nayak, Phys. Rev. B **93**, 201103 (2016).

[66] C. W. von Keyserlingk and S. L. Sondhi, Phys. Rev. B **93**, 245145 (2016).

[67] C. W. von Keyserlingk and S. L. Sondhi, Phys. Rev. B **93**, 245146 (2016).

[68] A. C. Potter, T. Morimoto, and A. Vishwanath, Phys. Rev. X **6**, 041001 (2016).

[69] R. Roy and F. Harper, Phys. Rev. B **94**, 125105 (2016).

[70] G. De las Cuevas, T. S. Cubitt, J. I. Cirac, M. M. Wolf, and D. Pérez-García, Journal of Mathematical Physics **57**, 071902 (2016), arXiv:1512.05709 [quant-ph] .

[71] R. Movassagh and P. W. Shor, ArXiv e-prints (2014), arXiv:1408.1657 [quant-ph] .