Boson condensation and instability in the tensor network representation of string-net states

Sujeet K. Shukla
Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, California 91125, USA

M. Burak Şahinoğlu
Vienna Center for Quantum Technology, University of Vienna, Boltzmanngasse 5, 1090 Vienna, Austria

Frank Pollmann
Max-Planck-Institut für Physik komplexer Systeme, D-01187 Dresden, Germany

Xie Chen
Department of Physics and Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, California 91125, USA

(Received 15 March 2018; revised manuscript received 20 July 2018; published 7 September 2018)

The tensor network representation of many-body quantum states, given by local tensors, provides a promising numerical tool for the study of strongly correlated topological phases in two dimensions. However, the representation may be vulnerable to instabilities caused by small variations in the local tensors. For example, the topological order in the tensor network representations of the toric code ground state has been shown in Chen, Zeng, Gu, Chuang, and Wen, Phys. Rev. B **82**, 165119 (2010) to be unstable if the variations break certain Z_2 symmetry of the tensor. In this work, we ask whether other types of topological orders suffer from similar kinds of instability and if so, what is the underlying physical mechanism and whether we can protect the order by enforcing certain symmetries on the tensor. We answer these questions by showing that the tensor network representations of all string-net models are indeed unstable, but the matrix product operator (MPO) symmetries of the tensors identified in Şahinoğlu, Williamson, Bultinck, Mariën, Haegeman, Schuch, and Verstraete, arXiv:1409.2150 can help to protect the order. In particular, we show that a subset of variations that break the MPO symmetries lead to instability by inducing the condensation of bosonic quasiparticles, which destroys the topological order in the wave function. Therefore such variations must be forbidden for the encoded topological order to be reliably extracted from the local tensors. On the other hand, if a tensor network based variational algorithm is used to simulate the phase transition due to boson condensation, such variation directions may prove important to access the continuous transition correctly.

DOI: 10.1103/PhysRevB.98.125112

I. INTRODUCTION

The tensor network representation of quantum states (including the matrix product states in 1D) [1–4] 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 [5], can be represented exactly with simple tensors [6,7]. Moreover, numerical studies applied to realistic models have identified nontrivial topological features in the ground-state wave function (see, e.g., Ref. [8–10]).

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. [11] 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. [11] 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 [13]. 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 similar problem occur for general string-net states as well? This is the question we address in this paper. In particular, we ask the following. (1) Does the tensor network representation of other string-net states also have such unstable directions of variation? (2) If so, can they be avoided by preserving certain symmetries in the local tensor? (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 requirements are necessary and if so what they are.

In this paper, we answer the above questions as follows. (1) All string-net tensors have unstable directions of variation. (2) To avoid such instabilities, we need to avoid "stand-alone" variations that break the matrix-product-operator(MPO) symmetry introduced in Refs. [12,14]. (We are going to explain in detail what "stand-alone" and MPO symmetry means in the following sections.) (3) The physical reason for the instability is that stand-alone variations, which violate these symmetries, induce condensation of bosonic quasiparticles and hence destroy (totally or partially) the topological order. To support the claims enlisted above, we calculate the topological entanglement entropy S_{topo} [15,16] 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, $T \to T + \epsilon T'$, 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 the 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

$$\lim_{\epsilon \to 0} S_{\text{topo}}(\epsilon) \neq S_{\text{topo}}(0). \tag{1}$$

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. Accuracy of a phase transition simulation using tensor network ansatz depends on the variational parameters in a complex way [17–21]. In particular, if one is to use the tensor network approach to study phase transitions due to boson condensation, then it is conjectured that the corresponding variation direction must be *allowed* in order for the simulation to give correct results. For example, in Ref. [22], 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 in the paper.

A word of caution about the terminology used in this paper. Since in this paper we are mostly concerned with the (tensor network) wave functions and *not* the underlying Hamiltonian, certain terms are somewhat re-contextualized from their common meaning in the literature. For example, throughout this paper, we will use the term "variation" to refer to mathematical variations in the local tensor (which may or may not be physical), or the wave function, while the term "perturbation" will be reserved for perturbations to the underlying parent Hamiltonian. A *local* variation of the wave function will also be referred to as an "excitation" when we want to stress the physical interpretation of the local variation at hand. In fact, these two terms will be used somewhat interchangeably (which is justified since we do not have a Hamiltonian to keep that distinction). Similarly, the term "proliferation" of a variation will denote the mathematical fact that the wave function is a superposition of a variation appearing everywhere on the original ground state, while the term "condensation" of an excitation/variation will be used when we want to stress on the physical interpretation of the proliferated variation. The term "topological phase transition," or just the "phase transition," will always refer to changes in the topological order of the tensor network state upon changes in the local tensor, regardless of whether or not a Hamiltonian is explicitly referred to. So, for example, with this terminology, a "condensation of an excitation" may or may not lead to a phase transition (depending on whether or not it leads to a change in the topological order). Lastly, we use the abbreviation "TNR" for the "tensor network representation" throughout this paper and it should not be confused for the "tensor network renormalization schemes," as used by some authors.

The paper is organized as follows. In Sec. II, we start from the simplest string-net model—the toric code model [23], and study two types of tensor network representation (TNR) of its ground state. The single line representation was studied in Ref. [11] and here we recover the result on the instability of the tensor with respect to certain Z_2 symmetry breaking variations. The second representation we study for the toric code is the double line representation, as discussed in Ref. [22]. While the single line representation has only one virtual Z_2 symmetry, the double line representation has multiple of them. Do they all protect the encoded topological order in the same way? To find out, we calculate, with the new algorithm, the topological entanglement entropy for the double line representation with different variations. It reveals that there are actually two kinds of symmetries, and their relation to the topological order is actually *opposite* to one another. That is, the only variations that change the topological order are the ones that *respect* the first kind of symmetry (which we call "stand-alone") but actually *break* the second kind (MPO symmetry [12]). One of the key results of this paper is contained in the next section, Sec. III, where we first identify the source of these two symmetries and define them precisely, and make the general conjecture of TNR instability. This conjecture says that all TNR have these two kinds of symmetries it is always the variations that respect the stand-alone but break the MPO symmetry that are unstable.

Then in Sec. IV, we put forward the physical understanding of TNR instability by systematically understanding the physical significance of the two symmetries and their interplay. This understanding concludes that the instability occurs due to condensation of topological bosons of the string-net model under consideration. In the next section, we note that tensor instability actually has implications for phase transition simulations using the tensor network ansatz. Hence our result should guide the choice of tensor network anstaz in phase transition simulations. To generalize our study to generic string-net models, we study next the double semion model in Sec. VI. We first predict instabilities using our conjecture, and then find them to be true in our numerical calculation.

We then directly apply our study to the general string-net model and its triple-line TNR in Sec. VII. We calculate the required symmetries and conclude that our conjecture predicts that the triple-line TNRs of *all* string-nets have unstable directions of variations. An analytical proof of this prediction can be found in Appendix G. Finally, in Sec. VIII, we test our understanding of the general string model on the double-Fibonacci model, which has a non-Abelian topological order as opposed to toric code and double semion. We again find our conjectures to be precisely accurate. Our results also reveal the physical meaning of the virtual tensor network symmetries for topologically ordered ground states that have been found for Kitaev quantum double models (G injectivity [24]) and later generalized to twisted quantum doubles (twisted G injectivity [14] and MPO injectivity [12]) and general string-net models (MPO injectivity [12]).

Finally, a summary of the results is given in Sec. IX and open questions are discussed. Some details of our analysis are explained in the appendices, including the relations of MPO symmetries to the Wilson-loop operators, a brief review of string-net models, their tensor network representation and their transformation under the application of string-operators, a proof of the existence of unstable directions in triple-line representations of general string-net ground states, and finally the dependence of topological entanglement entropy on the choice of boundary conditions in our calculations.

II. INSTABILITIES IN TNRS OF THE TORIC CODE

We start from the simplest illustrative example of nonchiral intrinsic topological order: the toric code [23]. We work on a hexagonal lattice and assign local degrees of freedom, i.e., 0-spin up or 1-spin down, on the edges of the lattice. It is convenient to consider $|1\rangle$ as a presence of a string and $|0\rangle$ as the absence of the string. So the total Hilbert space can be thought of as the space of all string configurations on a hexagonal lattice. The toric code Hamiltonian is a sum of local commuting projectors, given as

$$H = -\sum_v A_v - \sum_p B_p = -\sum_v \prod_{l\in v} Z_l - \sum_p \prod_{l\in p} X_l, \tag{2}$$

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 (see Fig. 1). Vertex terms restrict the ground states to closed strings of 1s and plaquette terms make all possible loop configurations of equal weight. Hence the toric code ground state (up to normalization) can be written as

$$|\Psi_{\rm gs}\rangle = \sum_{X\in \text{closed}} |X\rangle, \tag{3}$$

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 $S_{\rm topo} = \ln 2$. The toric code model has four anyons (superselection sectors), $\{\mathbf{1}, e, m, em\}$. $\mathbf{1}$ is the vacuum, e particle is the Z_2-gauge charge (violates the vertex term), and m particle is the Z_2-gauge flux (violates the plaquette term). Both e and m have a trivial topological spin, so we call them *topological bosons*, or simply bosons. Braiding e with m produces a phase factor of -1.

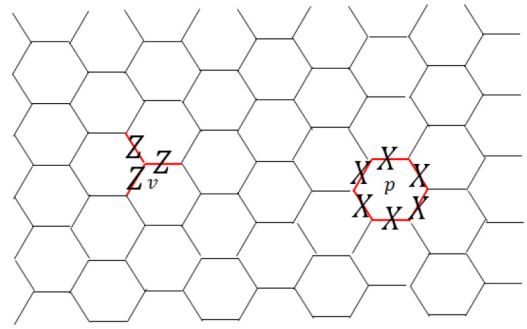


FIG. 1. Vertex and plaquette terms of the toric code Hamiltonian.

Now, we look at the tensor network representations (TNR) of the toric code ground state. Specifically, we will first explain the *single-line* tensor representation, and then the *double-line* tensor representation. We will see that different TNRs of the same state can have different kinds of instabilities.

A. Single-line TNR of the toric code and its instability

This is the simplest TNR of the toric code state. We first copy each computational basis into two, as shown in Fig. 2. That is, the labels 0 and 1 on every edge become 00 and 11 on the same edge. Now the local Hilbert space neighboring 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, and γ are the three virtual indices, as shown in Fig. 2. The components of the tensor are

$$(T^0)^{ijk}_{\alpha\beta\gamma} = \begin{cases} \delta_{i\alpha}\delta_{j\beta}\delta_{k\gamma} & \text{if } \alpha+\beta+\gamma = \text{even}, \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

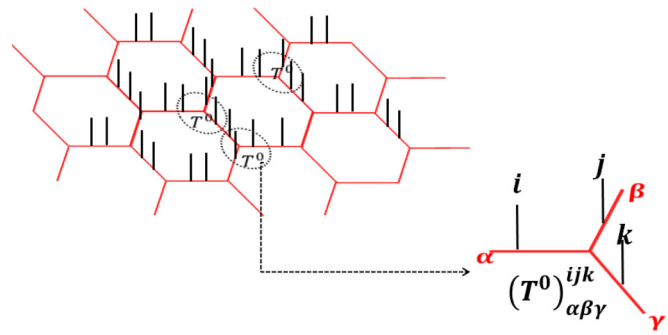
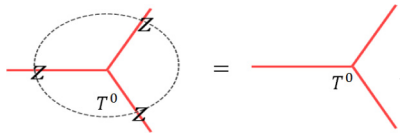


FIG. 2. Single-line TNR of the toric code state: we double the local Hilbert space on each edge and take $|i\rangle \to |ii\rangle, i = 0, 1$. So the state has a Z_2 topological order but on a bigger Hilbert space. We associate to each vertex the tensor product of three Hilbert spaces closest to it. We can now place a tensor, T^0 on each vertex with three out of plane physical legs, i, j, and k, and three in-plane virtual legs, α, β, and γ with values as given in Eq. (4).

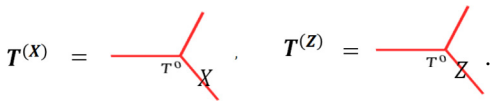
where δ is the Kronecker delta function. So, physical and virtual legs are identified and an even number of indices carry label 1 out of every three edges neighboring a vertex, i.e., we satisfy the vertex condition. The plaquette condition is also satisfied since every configuration is of equal weight. Therefore the tensor network state constructed using the above local tensor leads to the toric code ground state given in Eq. (3).

It was shown by [11] that the 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 nonzero only when virtual legs have even number of 1s, it has a natural $Z \otimes Z \otimes Z$ virtual symmetry (see [24] for TNR virtual symmetries of the quantum double models). That is, the tensor in Eq. (4) satisfies the relation



(5)

where we have omitted the physical legs for visual clarity. (We would often omit physical legs from tensor diagrams throughout the paper when we are mostly concerned with the virtual space/indices.) It is a Z_2 symmetry with group elements $I \otimes I \otimes I$ and $Z \otimes Z \otimes Z$ acting on the virtual legs of the local tensor. Reference [11] 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 a single-line TNR. We can add an X or Z variation on one of the virtual indices of the tensor,



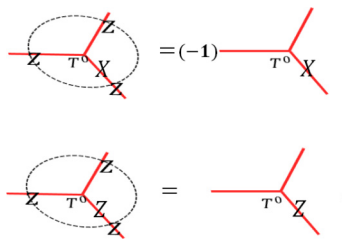
(6)

More explicitly, these tensor components are given by

$$T^{(X)}_{\alpha\beta\gamma} = \sum_{\gamma'} X_{\gamma,\gamma'} T^0_{\alpha\beta\gamma}, \tag{7}$$

$$T^{(Z)}_{\alpha\beta\gamma} = \sum_{\gamma'} Z_{\gamma,\gamma'} T^0_{\alpha\beta\gamma}. \tag{8}$$

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



(9)

and it was shown that $T^{(X)}$-type variations cause an instability, 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 *always* be stable.

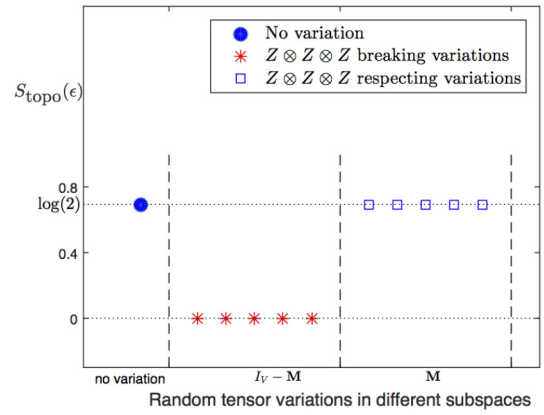


FIG. 3. Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by toric code fixed point single-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. $I_V = I^{\otimes 3}$ is the projector on to the full virtual space. $\mathbb{M} = \frac{1}{2}(I^{\otimes 3} + Z^{\otimes 3})$ is the projector onto the space of variations that respect the $Z^{\otimes 3}$ symmetries. So, $I_V - \mathbb{M}$ is a projector onto 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 Appendix I 1.

We reproduce this known result with a new algorithm for calculating S_{topo}. This algorithm allows us to calculate S_{topo} in more complicated examples to be dealt with later. This algorithm has been explained in detail in Appendix A. Interested readers are encouraged to study the details of the algorithm, however, it is not necessary to follow the key results in the main body of the paper.

B. Numerical result for single-line TNR with random variations

We use the algorithm described in Appendix A 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. (4). $I_V = I^{\otimes 3}$ is a projector onto the full virtual space. $\mathbb{M} = \frac{1}{2}(I^{\otimes 3} + Z^{\otimes 3})$ is a projector onto the space of variations that respect the $Z^{\otimes 3}$ symmetries. So, $I_V - \mathbb{M}$ is a projector onto 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 onto 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 onto $Z^{\otimes 3}$ respecting subspace $\mathbb{M}$, add it to the fixed point value, $T^0 \to T^0 + \epsilon \mathbb{M} T^r$, and 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. 3.

We see that $Z^{\otimes 3}$ respecting variations lead to the same topological entanglement entropy as the fixed point state, while

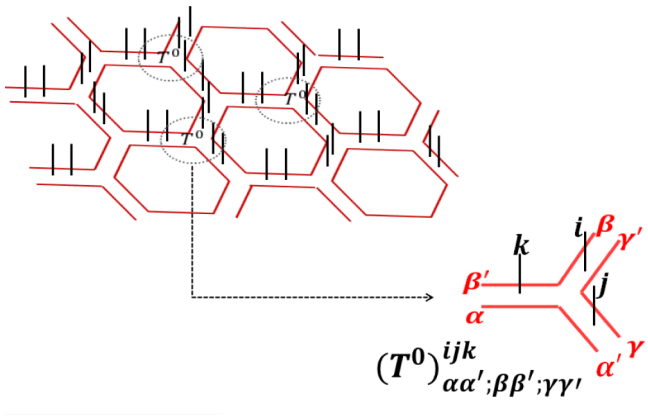


FIG. 4. Double-line TNR of the toric code state. We double the local Hilbert spaces in the same way as we do for single-line TNRs (Fig. 2). We associate to each vertex a tensor $T^{i,j,k}_{\alpha,\alpha';\beta,\beta';\gamma,\gamma'}$, where the out of plane legs, i, j, and k, correspond to the three physical indices, and the in-plane legs α, α', β, β', γ, and γ' are the virtual indices. The virtual indices of the tensors contract along the shared edges to produce the toric code state on the physical indices.

$Z^{\otimes 3}$ violating variations lead to zero topological entanglement entropy. This reproduces the result by Ref. [11].

C. Double-line TNR of the toric code state and its instablities

Now we discuss a TNR different from the single-line TNR above. It describes the same Z_2 topological order as the single-line TNR, but has different kinds of instabilities. So this helps us see the TNR dependency of instabilities. Moreover, as we will explain in Sec. III A, a single-line TNR is not the best example to illustrate general TNR instabilities because it has trivial "stand-alone" symmetries (to be defined below). Double-line TNRs on the other hand illustrate all the symmetries that we need to understand the general case.

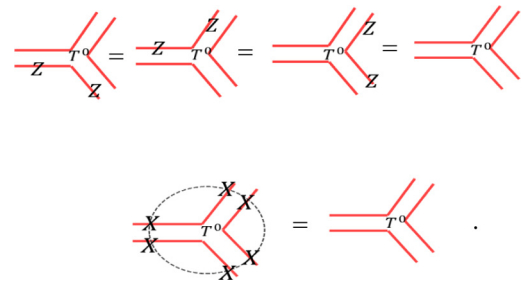
In the double-line TNR of the toric code state, we associate with each vertex a tensor with three physical legs and six virtual legs, $T^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'}$ (see Fig. 4). We will refer to these virtual indices as "plaquette indices" or "plaquette legs" sometimes, because 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.) The 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 nonzero components explicitly,

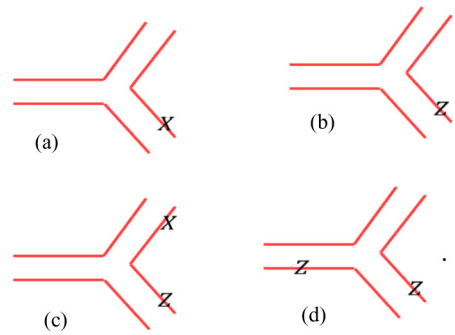
$$T^{000}_{00;00;00} = T^{000}_{11;11;11} = 1, \quad T^{011}_{00;11;11} = T^{011}_{11;00;00} = 1,$$
$$T^{101}_{11;00;11} = T^{101}_{00;11;00} = 1, \quad T^{110}_{11;11;00} = T^{110}_{00;00;11} = 1. \tag{10}$$

Is the double-line TNR unstable too? We find that it is unstable as well 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 us 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 three $Z \otimes Z$ symmetries, where $Z \otimes Z$ are applied to the two virtual indices on the same plaquette. So the double-line tensor in Eq. (10) satisfies these symmetry equations:



(11)

The single-line TNR has only one such Z_2 symmetry and it turns 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:



(12)

A variation can violate $Z \otimes Z$ but not $X^{\otimes 6}$ [for example, 12(a)], or it can violate $X^{\otimes 6}$ but not $Z \otimes Z$ [for example, 12(b)], or it can violate both [for example, 12(c)], or it can violate neither [for example, 12(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. 5) (1) if a variation violates any of the $Z \otimes Z$ symmetries then it is stable. (2) If a variation respects all $Z \otimes Z$ symmetres then there are two subcases: (i) if it respects $X^{\otimes 6}$

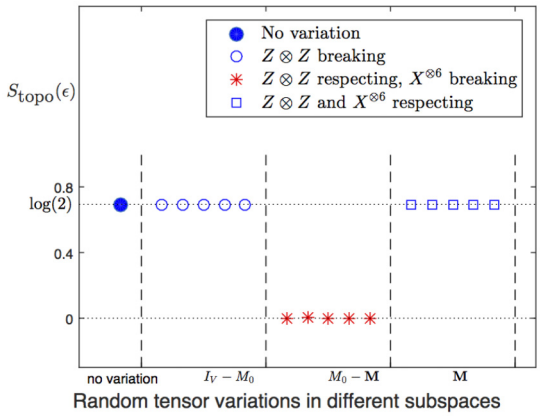


FIG. 5. Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of the states represented by toric code fixed point double-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. I_V is the projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO-injective 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.

symmetry then it is stable and (ii) if it breaks $X^{\otimes 6}$ symmetries then it is unstable.

So we see that the relation between unstable variations and virtual symmetries of the double-line TNR is more complicated than that for the 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 loop operators, and unstable variations in both TNR violate these loop symmetries. However, the crucial difference in the double-line is that then there are extra symmetries (the three $Z^{\otimes 2}$ 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.

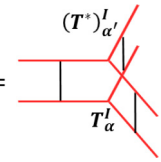
How can we understand this phenomena? We will now show that the sources of these symmetries are indeed different, and it is the interplay between these two symmetries that determines the tensor instability phenomena. The symmetries whose violation causes instability comes from the so-called *MPO-injective subspace* [12] of the virtual space, while the symmetries whose violation causes stability comes from what we define to be *stand-alone subspace.* We now define these subspaces precisely and put forward the conjecture regarding their relationship to the TNR instability.

III. VIRTUAL SYMMETRIES/SUBSPACES OF A TNR AND TENSOR-INSTABILITY CONJECTURE

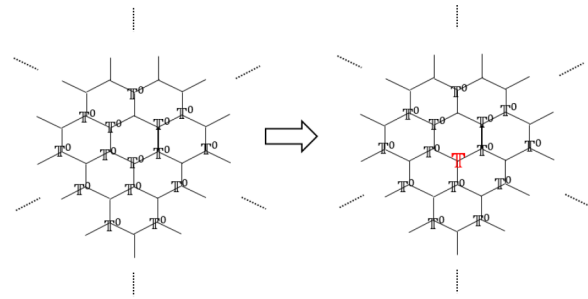
We first give a constructive definition of stand-alone subspace of a TNR.

A. Stand-alone subspace

A generic tensor can be thought of as a linear map from a virtual vector space to the physical vector space. Consider a generic tensor T^I_α, where I is the collection of all physical indices and α is the collection of all virtual indices. The *double tensor* $\mathbb{T}$ of a tensor T is defined as $\mathbb{T} = \sum_I T^I_\alpha (T^*)^I_{\alpha'}$. For example, for a single-line TNR, the double tensor can be represented diagrammatically as follows:

$\mathbb{T}_{\alpha,\alpha'} =$  (13)

In general, we would think of double tensor as having two layers of virtual indices, lower and upper. A double tensor can be interpreted as a density matrix on the virtual space. Now consider the double tensor of an RG fixed point TNR, $\mathbb{T}^0$, contracted over some large region R. Let us say we remove $\mathbb{T}^0$ from one site and replace it with some other double tensor, $\mathbb{T}$ as follows:



(14)

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 for tensors that represent topological order. In fact, as we will 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 (see Fig. 6). Consider contracting the fixed-point double tensors $\mathbb{T}^0$ on a large disk with an open boundary. Now we remove the tensor at the origin. We get a tensor network with dangling virtual indices at the origin and at the boundary of the disk. We want to find 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 upper and lower virtual indices 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.

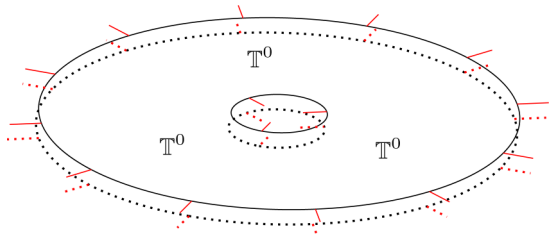


FIG. 6. Calculation of stand-alone space. We put the fixed point double tensor network on a large disk with a hole at the origin (one double tensor removed). This tensor network has dangling virtual indices (red legs) at the outer and inner boundaries. The upper/lower virtual legs are shown as dotted/solid red lines. 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.

Note that the definition of stand-alone space implies that it is a vector space. If T_1 and T_2 do not collapse then $aT_1 + bT_2$ will also not collapse the network.

Stand-alone subspace of a double-line TNR. Now let us first calculate the stand-alone subspace of a double-line TNR of toric code as it is more interesting than that of a single-line TNR. The double tensor of T^0 in Eq. (10) can be written as (ignoring an overall normalization factor)

$$\mathbb{T}^0 = \sum_I (T^0)^I_\alpha (T^{0;*})^I_{\alpha'} = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + X^{\otimes 6}), \tag{15}$$

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. 7. We need to contract this tensor on a disk 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 straightforward calculation shows that double tensors contracted on a region

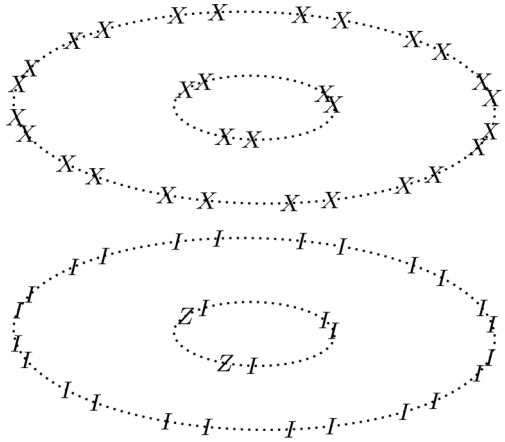


FIG. 7. X and Z operators appear differently in the toric code double-line double tensor contracted on a disk with a hole. X operators on the inner boundary only appear with X operators on the outer boundary, which vanish upon taking the trace. However, Z operators on the inner boundary appear with identity on the outer boundary. So these terms survive the trace. This is why $Z^{\otimes 2}$ symmetry is imposed on the stand-alone space but not the $X^{\otimes 6}$ symmetry.

R give (ignoring an overall normalization factor)

$$\mathbb{T}^0(R) = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes m}(\partial R)(I^{\otimes 2m}(\partial R) + X^{\otimes 2m}(\partial R)), \tag{16}$$

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 disk with a vertex removed, $R = D_{2m} - D_6$. D_n denotes the disk with there are n virtual legs at the boundary. It has two disconnected boundaries, one is the boundary of D_{2m} and the other is the boundary of D_6 (with opposite orientation):

$$\mathbb{T}(D_{2m} - D_6) = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes m} \otimes (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} \times (I^{\otimes 2m} \otimes I^{\otimes 6} + X^{\otimes 2m} \otimes X^{\otimes 6}). \tag{17}$$

As explained in Fig. 7, 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 a 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 six virtual indices incident on a single vertex:

$$B_0 = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}. \tag{18}$$

$B_0^2 = 8B_0$, so

$$M_0 = \tfrac{1}{8}B_0 = \tfrac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} \tag{19}$$

is a projector onto 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 onto a 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.

Stand-alone subspace of a single-line TNR. We can also calculate the stand-alone subspace of a single-line TNR of toric code. One can calculate the double tensor of single-line TNR in Fig. 4 to be

$$\mathbb{T}^0 = \tfrac{1}{2}(I^{\otimes 3} + Z^{\otimes 3}). \tag{20}$$

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

$$\mathbb{T}^0(D_m - D_3) = I^{\otimes m} \otimes I^{\otimes 3} + Z^{\otimes m} \otimes Z^{\otimes 3}. \tag{21}$$

Now contracting the outer circle gives us

$$B_0 = M_0 = I^{\otimes 3}. \tag{22}$$

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

B. MPO-injective subspace

Here, we repeat the definition of MPO-injective subspace given in Ref. [12] for convenience. As explained above, a

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 a subspace of the stand-alone space, is what we would call the *MPO-injective subspace*. The reason for calling it an MPO-injective space is that, as it turns out, this space is protected by symmetry operators which are matrix product operators (MPO). Let us make the notion of virtual space of $(T^0)^I_\alpha$ precise. We can think of it as a matrix with indices α and I and perform an SVD decomposition,

$$(T^0)^I_\alpha = \sum_{\alpha',I'} V_{\alpha,\alpha'} \Lambda_{\alpha',I'} P_{I',I}, \tag{23}$$

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-injective space, defined as the virtual support space of T^0, is the virtual subspace spanned by columns of V for which the corresponding singular value is nonzero.

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

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

Using Eq. (23), we can write

$$T^0 = \sum_{j;\lambda_j \neq 0} \lambda_j |v_j\rangle\langle p_j| \tag{24}$$

$$\Rightarrow \mathbb{T}^0 = \sum_{j;\lambda_j \neq 0} \lambda_j^2 |v_j\rangle\langle v_j|, \tag{25}$$

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

$$\mathbb{M} = \sum_{j;\lambda_j \neq 0} |v_j\rangle\langle v_j|. \tag{26}$$

The mathematical understanding of the MPO-injective space is that it is the virtual subspace which is isomorphic to the ground-state physical subspace and T^0 is the isomorphism. The MPO-injective subspace is 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. 8 for clarity.

MPO-injective subspace of a single-line TNR. We can write the single-line tensor in Eq. (4) in the Eq. (23) form:

$$T^0 = |000\rangle\langle 000| + |011\rangle\langle 011| + |101\rangle\langle 101| + |110\rangle\langle 110|. \tag{27}$$

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

$$\mathbb{M} = |000\rangle\langle 000| + |011\rangle\langle 011| + |101\rangle\langle 101| + |110\rangle\langle 110|$$

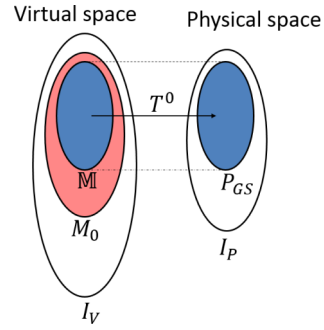


FIG. 8. A pictorial representation of relevant vector spaces. A tensor T^0 is a linear map from the virtual space to the physical space. We denote the full virtual space as I_V and the full physical space as I_P. M_0 (region in red and blue) is the stand-alone subspace of the virtual space. $\mathbb{M}$ (region in blue) is the MPO-injective subspace of the stand-alone space. MPO-injective 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.

$$= \tfrac{1}{2}(I^{\otimes 3} + Z^{\otimes 3}). \tag{28}$$

We see that this projector can be written as a translation invariant superposition of a tensor product of matrices. This is why we call it the MPO-injective subspace. Remember that the 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-injective subspace of a double-line TNR. For calculation of the MPO-injective subspace of double-line tensor in Eq. (10), we use the second definition in definition 2 above to avoid a cumbersome but straightforward calculation. We already calculated the double tensor of a double-line TNR in Eq. (15). We ignored the normalization factor there. If we use a normalization factor of $1/16$ and write

$$\mathbb{M} = \frac{1}{8}\mathbb{T}^0 = \frac{1}{16}\sum_I (T^0)^I_\alpha (T^{0;*})^I_{\alpha'}$$

$$= \frac{1}{16}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + X^{\otimes 6}). \tag{29}$$

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 the stand-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.

C. TNR instability conjecture

Now that we have defined the stand-alone and MPO-injective 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-injective 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 $P_S T = T$.

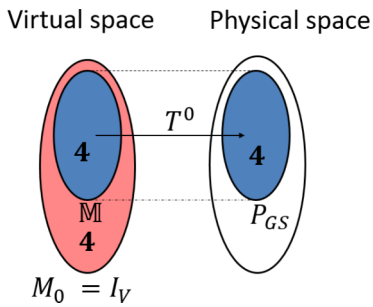
Corollary 2. For any tensors T, the variation $T^0 \to T^0 + \epsilon P_S T$, with $\epsilon \ll 1$ does not change the topological phase.

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-injective subspace.* A pictorial representation of the decomposition of the virtual space through these projectors is shown in Fig. 8. We will denote $(M_0 - \mathbb{M})$ as P_U for convenience. Note that P_U should not 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 us first see how this conjecture is true for the single-line and double-line TNR of the toric code. For a single-line one, we have already calculated the stand-alone and MPO-injective subspaces and found $M_0 = I_V$ and $\mathbb{M} = \frac{1}{2}(I^{\otimes 3} + Z^{\otimes 3})$. So,

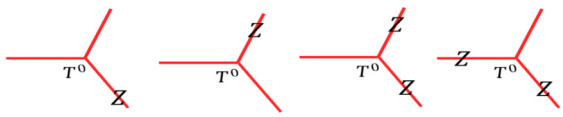
$$P_U = M_0 - \mathbb{M} = \tfrac{1}{2}(I^{\otimes 3} - Z^{\otimes 3}). \tag{30}$$

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. All variations can be summarized visually using Fig. 8 as follows:



(31)

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



(32)

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

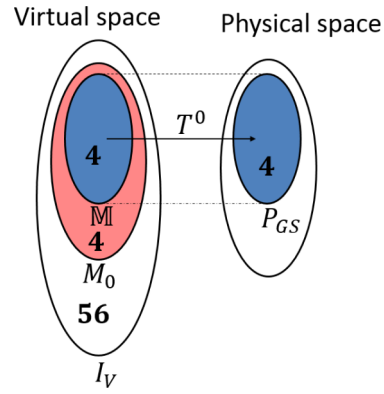
For a double-line TNR, we found

$$\begin{aligned} M_0 &= \tfrac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}, \\ \mathbb{M} &= M_0 \tfrac{1}{2}(I^{\otimes} + X^{\otimes 6}). \end{aligned} \tag{33}$$

So,

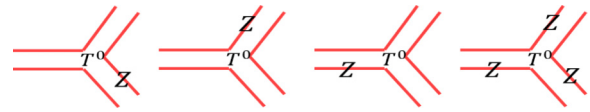
$$\begin{aligned} P_U &= M_0 - \mathbb{M} \\ &= \tfrac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} \tfrac{1}{2}(I^{\otimes} - X^{\otimes 6}). \end{aligned}$$

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. 4. All variations can be summarized visually as follows:



(34)

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 a four-dimensional space spanned by basis



(35)

IV. PHYSICAL UNDERSTANDING OF TNR INSTABILITY

TNR-instablity conjecture would predict mathematically exactly which variations would be unstable. However, what is the physical reason behind such instabilities? To answer, we put forward the following physical conjecture, which we would justify and explain in detail in the rest of this section.

Conjecture 2. Variations in stand-alone subspace M_0 correspond to "bosonic excitations" that can proliferate/condense in the given TNR. Variations in the MPO-injective 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 nontrivial condensable bosons. So, such a variation results in topological boson condensation and causes a topological phase transition of the state.

By "excitation" we mean any pointlike variation to the ground state, or its TNR. It should be carefully noted that the word "boson" here refers to any pointlike excitation (not necessary an irreducible excitation) that has trivial topological spin. For example, if a is an anyon of the given model, then a 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 say that the resulting state contains a boson. Though, of course, it is again a topologically trivial boson. Now, we turn to the first part of the claim, which is basically the physical significance of stand-alone space.

A. Physical understanding of stand-alone space M_0

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

1. 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 quasiparticle 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 us say T^0 is the RG fixed point tensor of some topological ground-state wave function $|\Psi_0\rangle$. Let us say we add a variation, $T^0 \to T^0 + \epsilon T$ and the resulting wave function is $|\Psi\rangle$:

$$\begin{aligned} |\Psi_0\rangle &= \sum_{\{i_j\}} (T^0)^{i_1} (T^0)^{i_2} \dots (T^0)^{i_n} |i_1 i_2 \dots i_n\rangle, \\ |\Psi\rangle &= \sum_{\{i_j\}} (T^0 + \epsilon T)^{i_1} (T^0 + \epsilon T)^{i_2} \dots \\ &\quad (T^0 + \epsilon T)^{i_n} |i_1 i_2 \dots i_n\rangle \\ &= |\Psi_0\rangle + \epsilon \sum_{s_1} |\Psi_{s_1}\rangle + \epsilon^2 \sum_{s_1,s_2} |\Psi_{s_1,s_2}\rangle + \cdots, \end{aligned} \tag{36}$$

where $|\psi_{s_1}\rangle$ denotes a tensor network state similar to $|\Psi_0\rangle$ except T^0 has been replaced with T at site s_1. Similarly, $|\Psi_{s_1,s_2}\rangle$ denotes a 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, $|\Psi_{s_1,s_2}\rangle$ can be interpreted as excitation T sitting at sites s_1 and s_2 with probability $\propto \epsilon^4$. 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, 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 the RG process. Tensors within the stand-alone space, on the other hand, can appear at any scale and would not vanish under the RG process. So we can call the new wave functions 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. However, 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" does not necessarily mean a phase transition. It just means a particular mathematical variation, 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 s_1 and s_2 can be at arbitrary distance from each other but the contribution of this term in the superposition remains ϵ^2. Let us compare this with how the ground state changes with respect to a perturbation on the Hamiltonian level. Let us perturb the toric code Hamiltonian in Eq. (2) with X perturbations on every link,

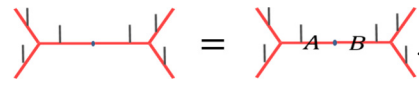
$$H = H_0 + \epsilon \sum_l X_l. \tag{37}$$

The ground state of this perturbed Hamiltonian is also a superposition of $|\Psi_0\rangle$ and terms like $|\Psi_{s_1,s_2}\rangle$. However, the weight that appears with $|\Psi_{s_1,s_2}\rangle$ is of the order of $\epsilon^{\text{distance}(s_1,s_2)}$, that is, the separation between two e particles is exponentially suppressed. So, in the thermodynamic limit, these excitations disappear. But this is not the case with the state in Eq. (36). That is why the state in Eq. (36) 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.

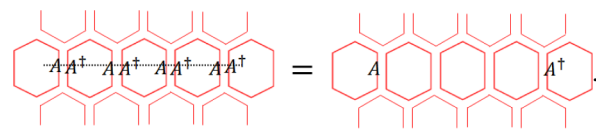
2. Condensable excitations are "bosons"

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 quasiparticles in topological models are calculated using the string operators that create them. So we need to first define a string-operator that creates these variations. The anyonic string operators in topological models have the property that they commute with the Hamiltonian everywhere except possibly at its ends. However, 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 does not change (though the individual tensors may change). That is,



(38)

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:



(39)

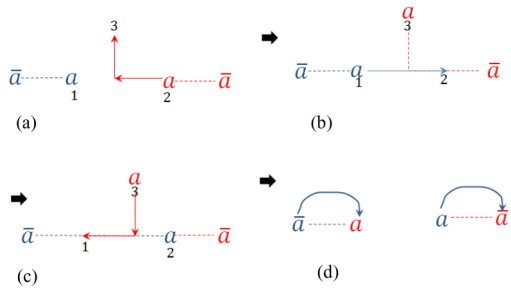


FIG. 9. Calculation of topological spin. We create two pairs (shown as red and blue) of particle, antiparticle pairs $a-\bar{a}$, with a situated at site 1 and 2. We apply the following procedure in this order: (a) move first a (red) from 2 to 3, (b) move second a (blue) from 1 to 2, (c) move first a (red) from 3 to 1. Finally, (d) we annihilate each a with the antiparticles 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.

The A and $A^\dagger$ cancel each other on each plaquette as all the six virtual legs are contracted. We chose a double-line tensor network for illustration but, of course, it can be done for any tensor network. So wave functions like $|\Psi_{s_1,s_2}\rangle$ can be created by such string operators. Note that since the tensor network did not change along the path, $|\Psi_{s_1,s_2}\rangle$ 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 deformable 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 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 which 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 statistics and self-statistics. However, to prove they are bosons, we need to do the topological spin calculation. Though again, using the same reasoning, it can readily be seen that the topological spin of the stand-alone excitation is 1, as explained in Fig. 9.

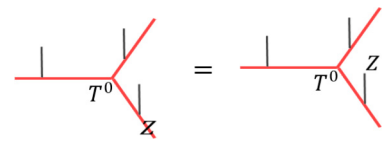
We have determined that the variations in the stand-alone space are condensable bosons. So any such variation results in a wave function, which is a condensate of the boson the variation corresponds to. However, this alone does not necessarily cause a phase transition, because if the boson was topologically trivial, there should be no topological phase transition. Or, mathematically speaking, the stand-alone projector projects out variations that cannot proliferate, but it does not 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-injective 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 *nonlocal* 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-injective 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 us first focus on the first kind of variations.

B. Physical understanding of MPO-injective subspace $\mathbb{M}$

As claimed above, the physical significance of MPO-injective 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-injective subspace is that it contains all the topologically trivial excitations.

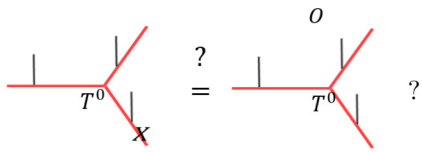
To understand it better, let us look at concrete examples of variations that are lifted to local physical variation. Consider a Z variation on the virtual leg of the fixed point single-line TNR. If we lift it to the physical level, what do we get? Since the virtual legs are just copies of the physical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



(40)

So Z virtual variation is lifted to a Z physical variation, which is local. According to our claim, it should be in the MPO-injective 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

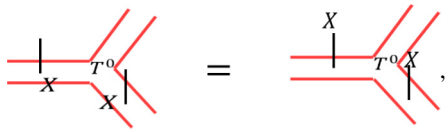
such that



(41)

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 nonlocal operator.)

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



(42)

but a Z variation cannot be. (We will later show that Z can be lifted to the physical level, but it results in a nonlocal 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-injective subspace is a local physical variation.

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

$$T^0 = \sum_j \lambda_j |v_j\rangle\langle p_j|, \tag{43}$$

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-injective subspace is the virtual subspace spanned by vectors v_j such that the corresponding singular values $\lambda \neq 0$. So the MPO projector is

$$\mathbb{M} = \sum_{j;\lambda_j \neq 0} |v_j\rangle\langle v_j|. \tag{44}$$

A mathematically inclined reader would note that MPO-injective 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-injective subspace to ground-state physical subspace. *Since these spaces are isomorphic, any operator in MPO-injective 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. Suppose a virtual operator A is given which is completely supported on subspace $\mathbb{M}$. Define pseudoinverse of T^0 as

$$T^{0;+} = \sum_{j;\lambda_j \neq 0} \frac{1}{\lambda_j} |p_j\rangle\langle v_j|. \tag{45}$$

It is a pseudoinverse since

$$T^0 T^{0;+} = \mathbb{M}, \tag{46}$$

$$T^{0;+} T^0 = P_{\mathrm{GS}}, \tag{47}$$

where $P_{\mathrm{GS}} \subset I_P$ denotes a 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

$$B = T^{0;+} A T^0, \tag{48}$$

then

$$T^0 B = T^0 T^{0;+} A T^0 = \mathbb{M} A T^0 = A T^0. \tag{49}$$

The last equality follows from the assumption that A is completely supported on MPO-injective subspace. Similarly, given a 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-injective 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-injective subspace should contain all trivial virtual excitations as well.

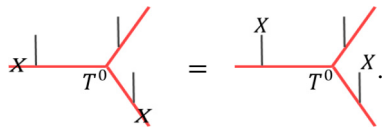
This completes the study of first kind of variations (those that are lifted to local physical variations) mentioned above. Now we study the second kind of variations.

C. Physical understanding 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 nonlocal physical operator. So these variations cannot be removed by a local physical operation on the state, and hence represent a topologically nontrivial 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 nontrivial bosons, and this is why these variations cause a topological phase transition.

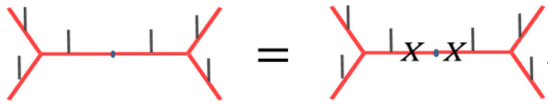
Let us first look at some concrete examples to understand this phenomena. We saw how a virtual X variation on the single-line TNR could not be lifted to a local physical variation. However, the question is, can it be lifted to a nonlocal physical variation? The answer is, yes. To see it, first note that although a single X variation cannot be lifted locally, two X variations

can be. That is, the fixed point single-line tensor satisfies



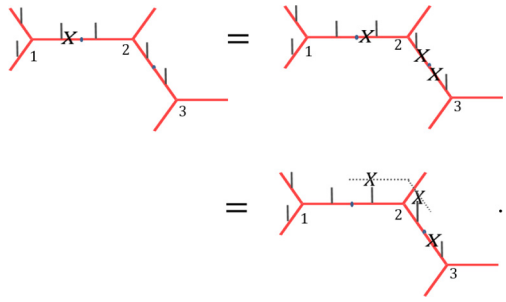
(50)

And also, we have the usual gauge symmetry



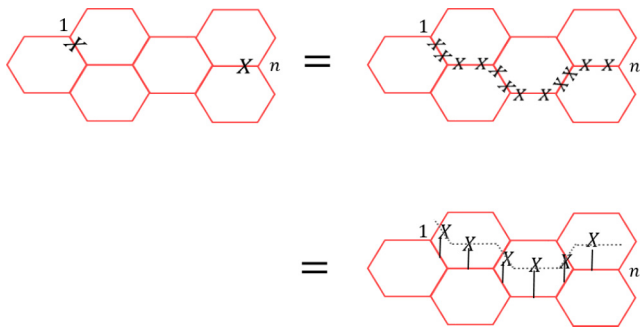
(51)

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,



(52)

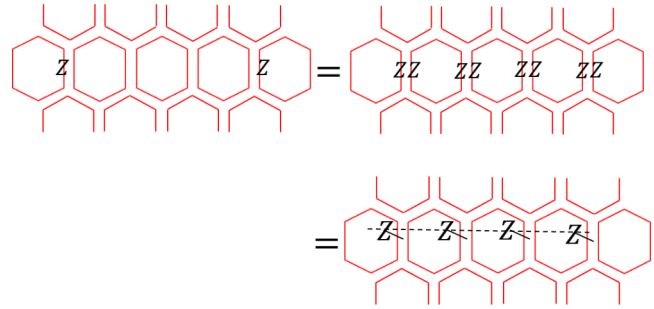
In the first equality, Eq. (51) is used while in the second equality, Eq. (50) is used. We see that the X variation moved from site 1 to site 3 while leaving operator $X \otimes X$ along the path (on site 2). We can repeat this process and move X to the next tensor and so on. After X is moved from site 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,



(53)

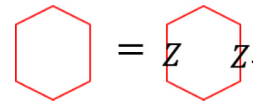
Of course, the particular path between site 1 and n chosen is completely arbitrary. We can choose any path between them as we like. So we have successfully shown that though a single X variation cannot be completely lifted to the physical level, two such variations sitting far apart can be, and they are lifted to a nonlocal physical operator between them. It implies that an 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 nonlocal 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 nontrivial, 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 a topological 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. However, two Z operators can be lifted to a nonlocal physical operator,



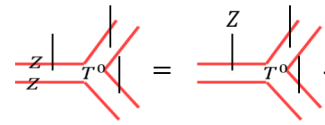
(54)

where in the first equality, the following relation [similar to Eq. (51)] has been used:



(55)

In the second equality, the following property of the fixed point double-line tensor is used:



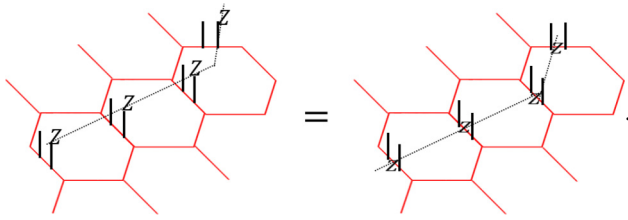
(56)

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 nonlocal operator, the Z-string operator. This suggests that the Z variation is a topologically nontrivial excitation. Indeed, it is easy to see that it is nothing but the m particle excitation, since a 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.

D. Nontrivial 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 nontrivial 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 nontrivial quasiparticle 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 the 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. (53) 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 in the single-line TNR, which subsequently creates a variation in the $M_0 - \mathbb{M}$ space. Contrast this with a physical Z-string operator that creates m particles. This operator does not map to a gauge-string operator in the single-line TNR,



(57)

Similarly, Eq. (54) shows that that the physical Z-string operator, which creates m particles, becomes a gauge-string operator on the virtual level in the double-line TNR. 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 nonlocal variation on the physical level. In other words, certain gauge-string operators are nontrivial because they come from a nontrivial string operator on the physical level. We would call such a physical string operator that maps to a gauge-string operator on the virtual level a *zero-string operator*. The reason behind this terminology will become clearer in the next section. 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.

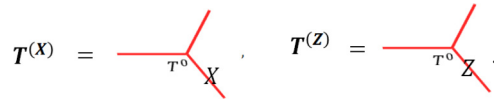
Before going to the physical explanation of instabilities, we would like to mention that there is one more way of decomposing the stand-alone space in trivial and nontrivial excitations: using Wilson-loops of anyonic string operators. We can use these operators to detect whether a nontrivial excitation is sitting at a site, hence can potentially differentiate between $M_0 - \mathbf{M}$ and $\mathbf{M}$. This has been explored in Appendix B for readers who are interested in this perspective.

E. Physical reason of instability: topological boson condensation

Now we put together the physical understanding of all the relevant subspaces (M_0, $\mathbb{M}$, and $M_0 - \mathbb{M}$) 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 Figs. 3 and 5. The general explanation has already been stated in the form of conjecture 2 but we repeat it again informally going through all possible variations one by one. (1) The variations on the physical indices are of course not unstable because they are topologically trivial and can be removed with local operations. (2) Variations outside the stand-alone space, $I_V - M_0$ are not unstable because they cannot proliferate. (3) Every variation inside M_0 is "bosonic" and does proliferate and the varied wave function is a condensate of that "boson." But when the variation is inside $\mathbb{M}$, it is a topologically trivial boson and hence does not cause a topological phase transition. Or, equivalently, every variation inside the MPO-injective subspace is nothing but a variation on the physical level hence stable. (4) Finally, when the variation is inside stand-alone, but outside the MPO-injective subspace then it can condense and is a topologically nontrivial 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 TNRs considering a specific variation.

1. *e-particle condensation in single-line TNR*

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



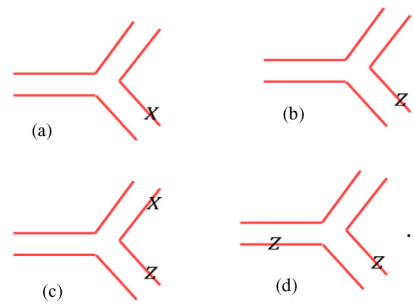
(58)

(1) 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.

(2) 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 excitations. Hence such a variation results in an e-particle condensation and destroys the topological order of the tensor network state.

2. *m-particle condensation in double-line TNR*

Let us consider the different variations in a double-line TNR as before:



(59)

(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. However, they break the MPO symmetry, so correspond to a nontrivial boson. In fact, it corresponds to an m-particle excitation. So this variation causes m-particle condensation and results in the loss of topological order. (3) Finally, variation in (d) exemplifies variations that can proliferate. However, they also are inside the MPO-injective subspace, so correspond to trivial/elementary excitations. So their proliferation does not cause a topological phase transition.

V. IMPLICATIONS FOR THE SIMULATION OF PHASE TRANSITIONS

Projected entangled pair states (PEPS), one type of tensor network states (TNS), are often used as an ansatz for different numerical simulations of gapped lattice topological models. In particular, TNS can be used to simulate phase transitions between different topological phases [17–22]. 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 (2), two kinds of perturbations can be added:

$$H_1 = -U \sum_v \prod_{l\in v} Z_l - g \sum_p \prod_{l\in p} X_l - \eta \sum_l Z_l, \quad (60)$$

$$H_2 = -U \sum_v \prod_{l\in v} Z_l - g \sum_p \prod_{l\in p} X_l - \eta \sum_l X_l. \quad (61)$$

Let us 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. (3). That is, it is an equal weight superposition of all closed string configurations. At $g = 0$, the state is the vacuum state, that is, all spins are 0. These two states are topologically different, 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 $\langle\Psi|B_p|\Psi\rangle = 1, \forall p$, while for $g = 0$ ground state, we have $\langle\Psi|B_p|\Psi\rangle = 0, \forall p$. It indicates that as η/g is increased, m particles proliferate and at the 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, the ground-state energy and its first-order derivative as a function of η/g are smooth functions, but the ground-state energy second-order derivative is discontinuous at the phase transition point.

It was shown by Ref. [22] that an attempt to numerically simulate this phase transition point with a 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. However, with a double-line tensor network state ansatz, it gives the correct second-order phase transition with a correct critical point.

This difference can be easily understood in light of our discussion of the single-line and double-line TNRs of a toric code state. As we showed, a double-line TNR is capable of condensing m particles while a single-line TNR is not. This is why a 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 the relative value of U and η. For $\eta = 0$, the ground state is the toric code ground state in Eq. (3), 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 a single-line TNS ansatz and not the a double-line TNS one.

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) [22] and tensor network renormalization (TNR)[1] [25] 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.

VI. DOUBLE SEMION

The double-semion model can be understood as a "twisted" Z_2 quantum double model [16,26]. Its Hamiltonian is almost the same as that of toric code, except for the phase factor associated to the plaquette term,

$$H_0 = -\sum_v \prod_{l\in v} Z_l - \sum_p \prod_{l\in p} X_l \prod_{r\in \text{legs of } p} i^{\frac{1-Z_r}{2}}, \quad (62)$$

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

$$|\psi\rangle = \sum_{X\in\text{closed}} (-1)^{n(X)} |X\rangle, \quad (63)$$

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. However, 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 three quasiparticle excitations: a semion, an antisemion, and a self-boson. So, unlike the toric code, it has only one boson. There is a known double-line TNR of this state [6,7], $(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'}$, with the same structure as that of toric code. So,

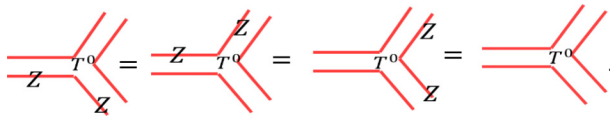
$$(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'} = S_{\alpha\beta\gamma}\delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{\gamma\gamma'}\delta_{i,\beta+\gamma}\delta_{j,\alpha+\gamma}\delta_{k,\alpha+\beta}. \quad (64)$$

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

However, now the values are

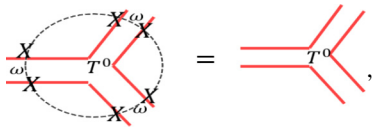
$$S_{\alpha\beta\gamma} = \begin{cases} 1 & \text{if } \alpha+\beta+\gamma = 0,\ 3; \\ i & \text{if } \alpha+\beta+\gamma = 1; \\ -i & \text{if } \alpha+\beta+\gamma = 2. \end{cases} \tag{65}$$

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



(66)

However, it does not have the exact $X^{\otimes 6}$ symmetry as that of a 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,



(67)

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,

$$n(d) = \begin{cases} 0 & \text{if } \alpha+\beta+\gamma = 0, 3 \\ 2 & \text{if } \alpha+\beta+\gamma = 1, 2. \end{cases} \tag{68}$$

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

$$\mathbb{T}^0 = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + i^{n(d)}X^{\otimes 6}). \tag{69}$$

Comparing it with the toric code double tensor in Eq. (15), we can immediately guess that the stand-alone projector is given by

$$M_0 = \tfrac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}, \tag{70}$$

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

$$\mathbb{M} = \tfrac{1}{16}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes} + i^{n(d)}X^{\otimes 6}). \tag{71}$$

So we see that the symmetries identified in Eq. (66) are actually the stand-alone symmetries, and the symmetry identified in Eq. (67) is actually the MPO symmetry. With this information, our mathematical conjecture predicts (1) if the variations breaks the $Z \otimes Z$ symmetries, then it is stable; (2) if the variations respects all $Z \otimes Z$ symmetries then there are two subscases: (i) if it also respects the $(i)^{n(d)}X^{\otimes 6}$ symmetry then it is stable and (ii) if it breaks the $(i)^{n(d)}X^{\otimes 6}$ symmetry then it is stable. We test these predictions numerically. The results are shown in Fig. 10. 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 correspond 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. (67) actually comes from the Wilson loop operator corresponding to semion (or antisemion). So variations that break it actually signify the presence of the boson.

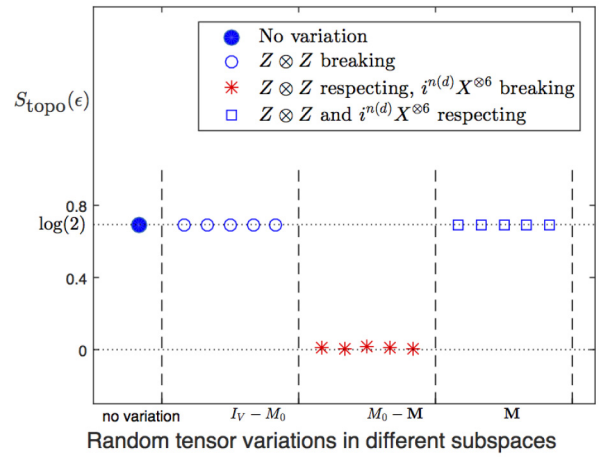
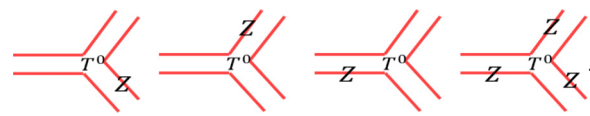


FIG. 10. Numerical calculation of the topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of the states represented by double-semion fixed point double-line tensors T^0 varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. I_V is the projector onto the full virtual space. M_0 is the projector onto the stand-alone subspace. $\mathbb{M}$ is the MPO-injective 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.

Comparing double-line TNR of toric code and double semion

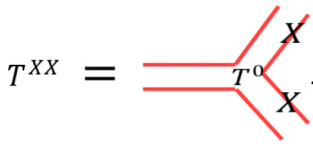
So we see that the space $(M_0 - \mathbb{M})$ for double semion is a four-dimensional space spanned by basis



(72)

This looks exactly similar to the $M_0 - \mathbb{M}$ basis in double-line toric code in Eq. (35), which might lead one to believe that they are both unstable for similar variations. However, one has to carefully note that the tensors T^0 for both models are different, so the bases shown in Eq. (35) and in Eq. (72) are actually

different. To illustrate this, consider the following variation:



(73)

This variation is in stand-alone space of both toric code and double semion, but it respects the MPO symmetry of the toric code but violates the MPO symmetry of the double semion. Indeed, this variation causes a 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. (42)]. However, one can readily see that this variation is not spanned by the basis in Eq. (72). The reason for this is that this variation has components both in the MPO-injective 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 example reminds us that for a variation to be unstable, all it needs is to have a nonzero component in the unstable space. So it should not be thought that only variations spanned by the unstable basis are unstable.

VII. GENERAL STRING-NET MODELS

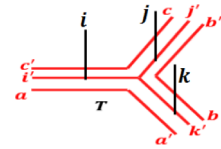
The models discussed so far, the toric code model and the double-semion model, are particular examples of a general class of 2D topological models known as *string-net models* [16]. Also, the TNRs discussed so far (single-line and double-line) are reduced versions of the general *triple-line* TNR of the string-net states [6,7].

A string-net construction defines a topological model on a honeycomb lattice for any arbitrary *unitary tensor fusion category* [16,23]. The local Hilbert space has spins sitting on the edges. These spins can take $i = 0, 1, \ldots, N-1$ values called *string-types*. $i = 0$ corresponds to the vacuum state. In general, strings have an orientation and for each string-type i we have a unique string-type $i*$ with opposite orientation. If $i^* = i$, then the string is called "unoriented." *In this paper, we would assume all strings are unoriented for simplicity but we believe our results are easily generalizable to oriented links.* A branching rule $\delta_{i,j,k}$ defines what string types are allowed to meet at a vertex. An F symbol guides how the strings fuse with each other. The F symbol comes from the unitary tensor category data and satisfies the so-called *pentagon equations*. A local commuting Hamiltonian is defined, $H = -\sum_v A_v - \sum_p B_p$, where v and p denote the vertices and plaquettes of the honeycomb lattice. The vertex term projects onto the space allowed by the branching rule. The plaquette term acts by creating loops of s-type strings which subsequently fuse with the existing string. As for any local commuting Hamiltonian, the ground state can be obtained by applying the projector $P_{gs} = (\sum_p B_p)(\sum_v A_v)$ on the vacuum state. A brief review of the string-net models has been given in Appendix C. Readers can refer to the original papers for more details on the subject [6,7,12,16].

A. Triple-line TNR of RG fixed point string-net state

As shown by [6,7], RG fixed point string-net states described above are known to have a triple-line TNR. We will only briefly discuss the relevant details here. A short derivation of the triple-line TNR is given in Appendix D. An interested reader may refer to the original papers [5–7] for more details.

A general triple-line tensor is represented diagrammatically as

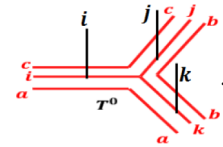


(74)

A string-net fixed point state has a triple line TNR with components given by

$$T^{ijk}_{aa';bb';cc';i'j'k'} = S^{ijk}_{abc}\delta_{a,a'}\delta_{b,b'}\delta_{c,c'}\delta_{i,i'}\delta_{j,j'}\delta_{k,k'},$$
$$\text{where } S^{ijk}_{abc} = \sqrt[4]{d_i d_j d_k}\, G^{ijk}_{abc} \sqrt[6]{d_a d_b d_v}. \quad (75)$$

So it would be represented diagrammatically as



(76)

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 a value which is a product (fusion) of the two legs on either side of it. This is why for Abelian models, double-line tensors suffice and the middle leg can be discarded. In non-Abelian models, such as the double-Fibonacci model we will study in Sec. VIII, 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, a toric code ground state is an equal superposition of all closed 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-injective and stand-alone subspaces. We will do so in the next two sections.

B. Stand-alone space of triple-line TNR string net

We find that the stand-alone space of the triple-line TNR is given by the following theorem:

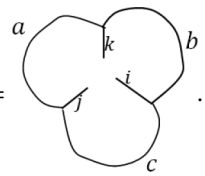
Theorem 1. The stand alone space, M_0, of the triple-line string net TNR is spanned by the orthonormal vectors

$$\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}|a, b, c; i, j, k\rangle, \quad (77)$$

where δ is the branching rule of the string-net model. i, j, and k label the middle legs and a, b, c label the plaquette legs.

The proof of this result is rather involved and is given in Appendix E. These basis vectors can be represented as string configurations,

$$|\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}\rangle = \quad . \tag{78}$$



So we get

$$\dim(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}. \tag{79}$$

C. MPO-injective subspace of a string-net triple-line TNR

We will use definition 2 to find the MPO-injective subspace of a triple-line TNR. Using the triple-line TNR T^0 of string-net states given in Eq. (75), the virtual density matrix is found to be

$$\begin{aligned}\sigma &= \sum_I (T^0)^I_\alpha (T^{0;*})^I_\alpha \\ &= \sum_{\{a_k,b_k;i_{k,k+1}\}} G^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} G^{i_{23}i_{31}i_{12}}_{b_1b_2b_3} \prod_j \left(d_{a_j}^{\frac{1}{6}} d_{b_j}^{\frac{1}{6}} d_{i_{j,j+1}}^{\frac{1}{2}}\right) \\ &\quad \times |\{a_k;i_{k,k+1}\}\rangle\langle\{b_k;i_{k,k+1}\}|. \end{aligned} \tag{80}$$

Clearly, this density matrix can simply be written as

$$\sigma = \sum_{i,j,k} |v_{i,j,k}\rangle\langle v_{i,j,k}|, \tag{81}$$

where

$$\begin{aligned}|v_{i,j,k}\rangle &= (d_i d_j d_k)^{\frac{1}{4}} \sum_{a_1,a_2,a_3} G^{i,j,k}_{a_1,a_2,a_3} \left(d_{a_1} d_{a_2} d_{a_3}\right)^{\frac{1}{6}} \\ &\quad \times |a_1,a_2,a_3;i,j,k\rangle. \end{aligned} \tag{82}$$

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 vectors $\frac{1}{N_{i,j,k}}|v_{i,j,k}\rangle$, where $N_{i,j,k} = \sqrt{\langle v_{i,j,k}|v_{i,j,k}\rangle}$ is the norm of vector $v_{i,j,k}$. So the string-net MPO projector is

$$\mathbb{M} = \sum_{i,j,k} \frac{1}{N^2_{i,j,k}} |v_{i,j,k}\rangle\langle v_{i,j,k}|. \tag{83}$$

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

$$\dim(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k}. \tag{84}$$

Comparing Eq. (79) with Eq. (84), 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 Appendix G that a triple-line TNR of a string net always has instabilities.

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

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

$$\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b}|a,b,c;i,j,k\rangle. \tag{85}$$

The MPO-injective space $\mathbb{M}$ space is spanned by $|v_{i,j,k}\rangle : \delta_{i,j,k} = 1$, where

$$|v_{i,j,k}\rangle = \sum_{a,b,c} G^{i,j,k}_{a,b,c} (d_a d_b d_c)^{\frac{1}{6}} |a,b,c;i,j,k\rangle. \tag{86}$$

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 $v_{i,j,k}$ that 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 nonzero. $M_0 - \mathbb{M}$ space can be decomposed into two subspaces.

(1) $\delta_{i,j,k} = 0$. This consists of all the string-configurations in Eq. (78) 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, and k, we just need to find vectors in individual $\mathbb{V}$ subspaces. To find these, we will use the orthogonality of G (C9):

$$\sum_c G^{i,j,k}_{a,b,c} G^{i,j,k}_{a,b,c} d_c = \frac{1}{d_k} \delta_{a,b,k}. \tag{87}$$

And the fact that matrices N^k defined by $N^k_{a,b} = \delta_{a,b,k}$ can be simultaneously diagonalized $\forall k$. Suppose $|s_q\rangle = s_{q;a}|a\rangle$ is its qth such simultaneous eigenvector. As discussed in Appendix C 1, $s_{0;a} = d_a$, that is, the vector formed by quantum dimensions is an eigenvector to N^k. These vectors are orthogonal, $\langle s_q|s_{q'}\rangle = \delta_{q,q'}$, which also implies that $\langle s_q|N^k|s_{q'}\rangle = \sum_{a,b} s_{q;a}\delta_{k,a,b}s_{q';b} \propto \delta_{q,q'}$. Now we are ready to write down the vectors spanning $\mathbb{V}_{i,j,k}$.

Consider vectors

$$\left|v^{q;a}_{i,j,k}\right\rangle = \sum_{a,b,c} \frac{s_{q;a}}{d_a} G^{i,j,k}_{a,b,c} (d_a d_b d_c)^{\frac{5}{6}} |a,b,c;i,j,k\rangle, \tag{88}$$

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

$$\begin{aligned}\left\langle v_{i,j,k}\middle|v^{q;a}_{i,j,k}\right\rangle &= \sum_{a,b,c} s_{q;a} d_b d_c G^{i,j,k}_{a,b,c} G^{i,j,k}_{a,b,c} && (89)\\ &= \sum_{a,b} s_{q;a}\delta_{a,b,k} d_b && (90)\\ &= \sum_{a,b} s_{q;a}\delta_{a,b,k} s_{0;b} && (91)\\ &\propto \delta_{q,0}. && (92)\end{aligned}$$

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 $N-1$ nonzero 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."

E. Instability of triple-line TNR

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 $T^0 \to T^0 + \epsilon T^q$, $\lim_{\epsilon\to 0} S_{\text{topo}}(\epsilon) \neq S_{\text{topo}}(0)$.

The proof of this theorem is rather involved and has been included in Appendix G.

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 Sec. III C explains the instabilities in string nets.

F. Examples: triple-line TNR of the toric code and double-semion states

Let us first examine how models covered in the previous chapter, toric code, and double semion fit into a triple-line TNR. One can get the triple-line TNR for them by plugging in the relevant string-net data into Eq. (75). We will apply the results about general-string net models developed in previous sections to the two cases.

The toric code string-net data are

$$\begin{aligned} &N = 1, \quad d_0 = 1, \quad d_1 = 1; \\ &\delta_{000} = \delta_{110} = \delta_{101} = \delta_{110} = 1; \\ &G^{000}_{000} = G^{000}_{111} = 1; \\ &G^{011}_{011} = G^{011}_{100} = G^{101}_{101} = G^{101}_{010} = G^{110}_{110} = G^{110}_{001} = 1. \end{aligned} \tag{93}$$

The triple-line TNR of toric code can be built by plugging in these data into the general expression in Eq. (75). This tensor has nine virtual indices, each of which takes two values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$ dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 8, \tag{94}$$

and the dimension of the MPO-injective subspace is

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 4. \tag{95}$$

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 eight-dimensional subspace, perturbations in a four-dimensional subspace are stable whereas the ones in the other four-dimensional subspace are unstable. Using Fig. 8, the classification of all variations can be represented as follows:

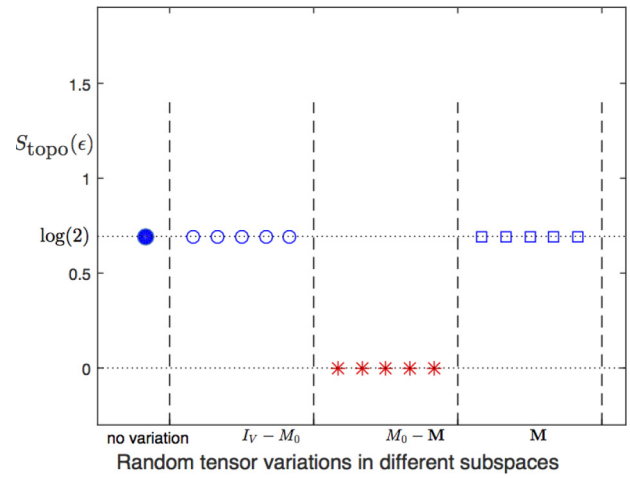
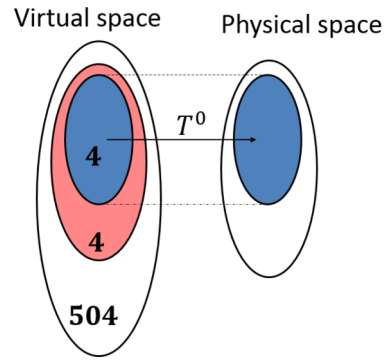


FIG. 11. Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by toric code fixed point triple-line tensors T^0 varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is the projector onto the full virtual space. M_0 is the projector onto the stand-alone subspace. $\mathbb{M}$ is the MPO-injective 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.



(96)

The numerical calculation supporting this conclusion is shown in Fig. 11. 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.

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

$$\begin{aligned} &N = 1, \quad d_0 = 1, \quad d_1 = 1; \\ &\delta_{000} = \delta_{110} = \delta_{101} = \delta_{110} = 1; \\ &G^{000}_{000} = 1; \\ &G^{011}_{011} = G^{101}_{101} = G^{110}_{110} = -1; \\ &G^{011}_{100} = G^{101}_{010} = G^{110}_{001} = G^{000}_{111} = -i. \end{aligned} \tag{97}$$

The triple-line TNR of the double semion model can be built by plugging in these data into the general expression in Eq. (75). This tensor has nine virtual indices, each of which takes two values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$

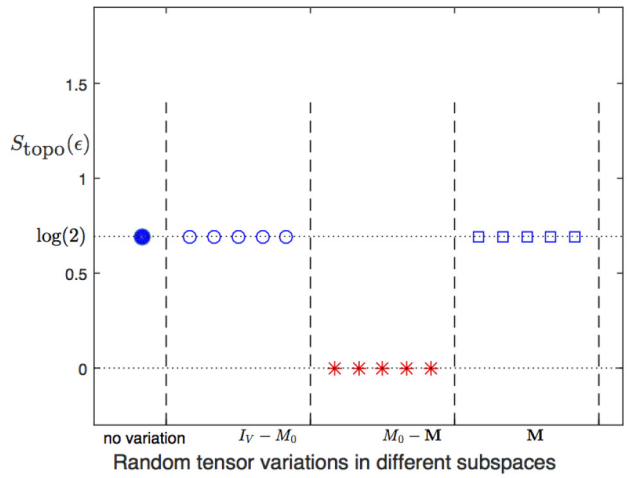


FIG. 12. Numerical calculation of topologiccal entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by double semion model fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is the projector onto the full virtual space. M_0 is the projector onto the stand-alone subspace. $\mathbb{M}$ is the MPO 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.

dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 8, \tag{98}$$

and the dimension of the MPO-injective subspace is

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 4. \tag{99}$$

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 eight: four are in stable and four are unstable. The numerical calculation supporting this conclusion is shown in Fig. 12. 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 Eq. (96) above.

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.

VIII. 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 Eq. (75)]. Let us first describe the model briefly. The data for this can be found in Sec. IV B of Ref. [5]. There is one type of string ($N = 1$). Its quantum dimension is $d_1 = \gamma = \frac{1+\sqrt{5}}{2}$. Its branching rules are

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

$$d_0 = 1,\ d_1 = \gamma, \quad \text{where } \gamma^2 = \gamma + 1; \tag{100}$$

$$G^{111}_{111} = -\frac{1}{\gamma^2}; \quad G^{110}_{111} = \frac{1}{\gamma};$$

$$G^{110}_{110} = \frac{1}{\gamma}; \quad G^{000}_{111} = \frac{1}{\sqrt{\gamma}}; \quad G^{000}_{000} = 1. \tag{101}$$

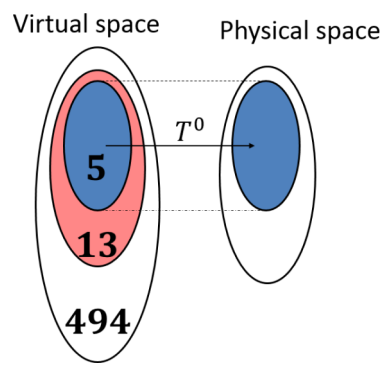
The branching rules tell us that one string is allowed to branch into two, unlike the Abelian models we have studied until now. First, let us apply our conjecture to find out how many unstable directions we should expect. The triple-line TNR of the Fibonacci model can be built by plugging in these data into the general expression in Eq. (75). This tensor has nine virtual indices, each of which takes two values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$ dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 18, \tag{102}$$

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

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 5, \tag{103}$$

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 stable as they are in the MPO-injective subspace and the remaining 13 are unstable. The classification of all variations can be represented pictorially as follows:



(104)

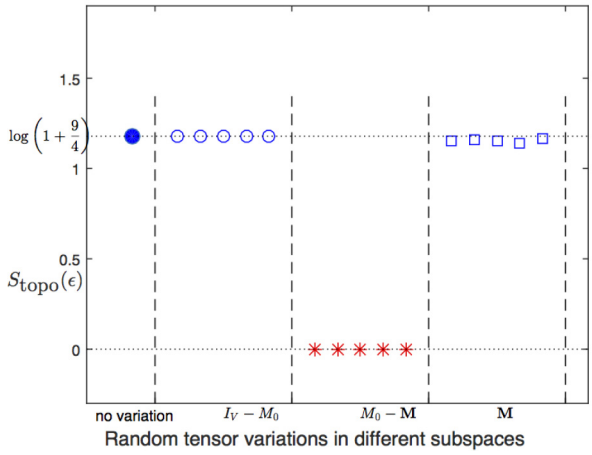


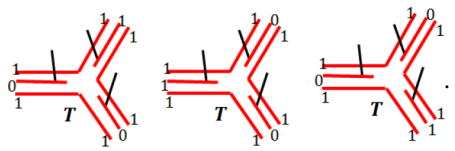
FIG. 13. Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by Fibonacci model fixed point triple-line tensors T^0 varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is the projector onto the full virtual space. M_0 is the projector onto the stand-alone subspace. $\mathbb{M}$ is the MPO-injective 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.

The numerical calculation supporting this conclusion is shown in Fig. 13.

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

$$|a,b,c;i,j,k\rangle = |1,1,1;1,0,0\rangle, |1,1,1;0,1,0\rangle, |1,1,1;0,0,1\rangle. \tag{105}$$

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



(106)

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

To understand the physics behind this, we need to look at the quasiparticles of the Fibonacci model. There are three quasiparticle excitations: τ, $\bar{\tau}$, and $\tau\bar{\tau}$. The T and S matrices of the particles are as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-\frac{4}{5}\pi i} & 0 & 0 \\ 0 & 0 & e^{\frac{4}{5}\pi i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$S = \frac{1}{1+\gamma^2}\begin{bmatrix} 1 & \gamma & \gamma & \gamma^2 \\ \gamma & -1 & \gamma^2 & -\gamma \\ \gamma & \gamma^2 & -1 & -\gamma \\ \gamma^2 & -\gamma & -\gamma & 1 \end{bmatrix}. \tag{107}$$

It is best seen as two layers of the Fibonacci model with opposite chiralities. τ and $\bar{\tau}$ are particles in the two respective layers. They have nontrivial self-statistics. However, because they are in different layers, they have a trivial statistics with one another. The boson $\tau\bar{\tau}$ is the composition of the Fibonacci particles in the two layers. The string operator for these quasiparticles is given in Eq. (51) of [16]. We are most interested in the boson of the model, so let us write its string operator (Ω matrices) explicitly:

$$\begin{aligned} &n_{4,0} = 1, \quad n_{4,1} = 1, \quad \Omega^0_{4,000} = 1, \quad \Omega^1_{4,110} = 1, \\ &\Omega^1_{4,001} = -\gamma^{-2}, \quad \Omega^0_{4,111} = \gamma^{-1}, \quad \Omega^1_{4,111} = \gamma^{-5/2}, \\ &\Omega^1_{4,101} = \Omega^{*1}_{4,011} = \gamma^{-11/4}(2 - e^{3\pi i/5} + \gamma e^{-3\pi i/5}). \end{aligned} \tag{108}$$

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 Eqs. (53) and (54), not changing tensors along the path. This is why a single variation standing alone could be thought of as an operator sitting at the ends of an invisible string operator. However, 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 do not 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. 14, one can see that a wave function corresponding to the boson sitting at two places, s_1 and s_2, is actually a superposition of many wave functions:

$$\begin{aligned} |\Psi_{\text{boson}}\rangle &= \sum_{t_1,s,t_2} n_s \Phi_{t_1,s,t_2} |\Psi_{\text{gs}}\rangle \\ &= |\Psi_{0,0,0}\rangle + |\Psi_{1,0,0}\rangle + |\Psi_{0,0,1}\rangle + |\Psi_{1,0,1}\rangle \\ &\quad + |\Psi_{0,1,0}\rangle + |\Psi_{1,1,0}\rangle + |\Psi_{0,1,1}\rangle + |\Psi_{1,1,1}\rangle, \end{aligned} \tag{109}$$

where the operator Φ_{t_1,s,t_2} is explained in Fig. 14. 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 an s-type string along the path. Fusing the loops with each other and with the

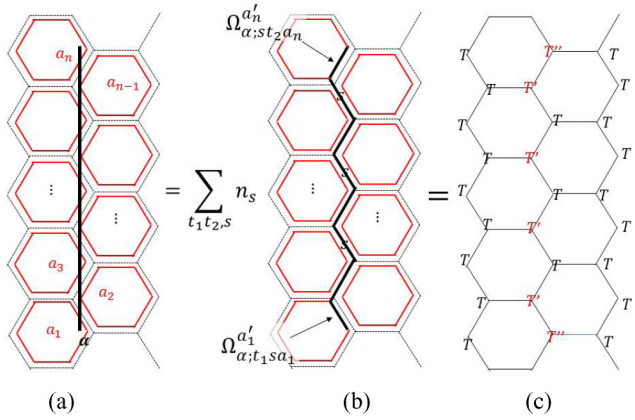


FIG. 14. The action of a generic (simple and nonsimple) open-end string operator corresponding to anyon α on tensors can be calculated in a similar fashion as that of simple-string operator Wilson loops. (a) We start by applying the string operator on the "loop state" on the fattened lattice. (b) The 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 an 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 does not change the tensors is for simple-string operators of type 0.

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 TNRs of $|\Psi_{t_1,0,t_2}\rangle$, $t_1, t_2 = 0, 1$, 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.

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.

IX. CONCLUSIONS AND DISCUSSION

In this paper, we try to answer the following question: are the tensor network representations of string-net states stable? That is, if we start from the tensor network representation of a string-net state and add arbitrarily small variations to the local tensor, does the topological order of the represented state always remain the same? This is an important question because if the answer is no, then the task of determining the topological order of a tensor network state may be numerically "ill-posed." That is, an arbitrarily small numerical error in the process may change our conclusion in a qualitative way. Previous work [11] has shown that this is indeed the case for the single line representation of the toric code state. While this may seem to seriously limit the applicability of tensor network methods to the study of the toric code type topological order, Ref. [11] also identified an inner Z_2 symmetry by preserving which the numerical task becomes "well-posed" again.

We want to know if similar problems happen for general string-net states. In particular, we asked the following. (1) Does the tensor network representation of other string-net states also have unstable directions of variation? (2) If so, can they be avoided by preserving certain symmetries in the tensor? (3) What is the physical reason behind such instabilities and their prevention?

We found that (1) all string-net tensors have unstable directions of variation. (2) To avoid such instabilities, we need to avoid the stand-alone variations that break the matrix-product-operator (MPO) symmetry introduced in Refs. [12,14]. (3) The physical reason for the instability is that the stand-alone variations, which violate these symmetries, induce condensation of bosonic quasiparticles and hence destroy (totally or partially) the topological order.

We demonstrated the case explicitly for the tensor network representation of the toric code (single, double, triple line), the double semion, and the double Fibonacci model, by calculating the topological entanglement entropy S_{topo} of tensors with random variations. We observe that MPO symmetry preserving variations keep S_{topo} invariant and MPO symmetry breaking variations lower S_{topo} (to zero). While for general string-net models we cannot prove the above claim analytically, we are able to show that (1) the fixed point tensor of any string-net has unstable directions (which break the MPO symmetry). (2) MPO breaking variations induce the condensation of bosons in the state, and therefore destroy (at least partially) the topological order. 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. [22].

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 that 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,

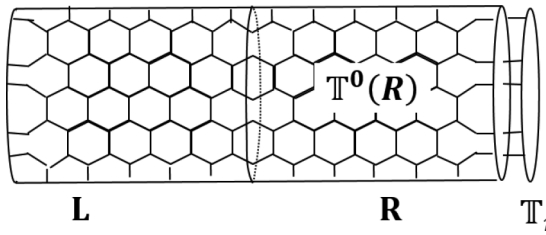


FIG. 15. The honeycomb lattice is put on a cylinder with some boundary tensors T_r. We calculate the topological entanglement entropy by calculating the entanglement entropy of the right half of the cylinder.

we need to try a different ansatz. How to do that in an efficient and unbiased way is again an interesting open problem. We leave these problems to a future study.

ACKNOWLEDGMENTS

S.K.S. would like to thank Pinaky Bhattacharyya for help with the numerical calculations. X.C. is supported by the Caltech Institute for Quantum Information and Matter and the Walter Burke Institute for Theoretical Physics.

APPENDIX A: ALGORITHM FOR CALCULATING TOPOLOGICAL ENTANGLEMENT ENTROPY

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 Ref. [27] to calculate the reduced density matrix on a region and hence its entanglement entropy. We consider the 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, $I = (i_1, i_2, \dots), \alpha = (\alpha_1, \alpha_2, \dots)$. The state represented by these tensors can be written as

$$|\Psi\rangle = \sum_{I_1, I_2, \dots} \mathrm{Tr}(T^{I_1} T^{I_2} \dots)|I_1, I_2, \dots\rangle. \tag{A1}$$

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, $\mathbb{T} = TT^\dagger = \sum_I T^I_\alpha (T^I_{\alpha'})^*$. 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 Fig. 15. 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,

$$\sigma_L = \mathbb{T}_l(\partial L)\mathbb{T}(L), \quad \sigma_R = \mathbb{T}(R)\mathbb{T}_r(\partial R). \tag{A2}$$

Reference [27] showed that the physical reduced density matrix on one of these halves, lets say the left one, is related to the density operator on the virtual indices as

$$\rho_L = U\sqrt{\sigma_L^T}\sigma_R\sqrt{\sigma_L^T}U^\dagger, \tag{A3}$$

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

$$\rho_L = \frac{\sigma_b^2}{\mathrm{Tr}(\sigma_b^2)}. \tag{A4}$$

It is known that the Rényi entropy with any Rényi index gives the same topological entanglement entropy [28]. So we calculate the Rényi entropy with a Rényi index $\frac{1}{2}$,

$$\begin{aligned} S_{1/2}(\rho_L) &= \frac{1}{1-1/2}\ln \mathrm{Tr}\left(\rho_L^{1/2}\right) \\ &= 2\ln \mathrm{Tr}(\sigma_b) - \ln \mathrm{Tr}\left(\sigma_b^2\right). \end{aligned} \tag{A5}$$

In the limit of large cylinder, it should behave like

$$S_{1/2}(\rho_L) = \alpha_0|C| - S_{\mathrm{topo}}, \tag{A6}$$

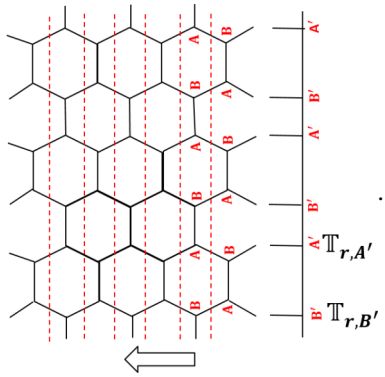
where $|C|$ is the circumference of the cylinder. This is how we calculate S_{topo} starting with a tensor network state.

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 Refs. [29,30], 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 Appendix H. Our findings are consistent with the conclusion in Ref. [30].

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 that is a slight modification of known tensor renormalization algorithms [4,22,31]. 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 states 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 the 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)* 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, the approximation to the eigenvector with the highest eigenvalue improves and we do this recursively until we reach the fixed point giving us the desired eigenvector. Note that we

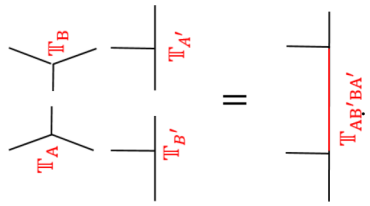
require transfer matrix operators to be reflection symmetric for the condition $\sigma_L^T = \sigma_R = \sigma_b \Rightarrow \rho_L \propto \sigma_b^2$ to hold true.

The recursive algorithm is as follows. (1) Initiate the boundary double tensor $\mathbb{T}_{A'} = \mathbb{T}_{r,A'}$ and $\mathbb{T}_{B'} = \mathbb{T}_{r,B'}$. The tensor network to be contracted looks as



(A7)

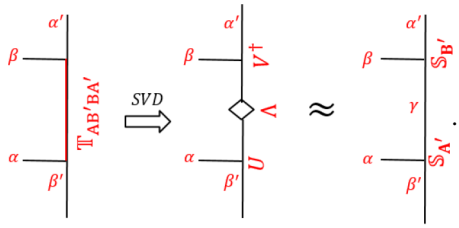
(2) Contract the bulk double tensors, $\mathbb{T}_A$ and $\mathbb{T}_B$ with the boundary tensors $\mathbb{T}_{A'}$ and $\mathbb{T}_{B'}$ in the following way to make the four leg tensor $\mathbb{T}_{AB'BA'}$,



(A8)

(3) Reshape the tensor $\mathbb{T}_{AB'BA'}$ into a matrix M, where $M_{\alpha\beta',\beta\alpha'} = (\mathbb{T}_{AB'BA'})_{\alpha\beta'\beta\alpha'}$ [31]. 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 $\mathbb{T}_{AB'BA'}$,

$$\sum_{\gamma=1}^{D_{\text{cut}}} (S_{A'})_{\alpha\beta'\gamma}(S^\dagger_{B'})_{\gamma\beta\alpha'} \approx (\mathbb{T}_{AB'BA'})_{\alpha\beta'\beta\alpha'} \tag{A9}$$



(A10)

(4) Check convergence of Λ. $\eta \ll 1$ is the precision 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.

APPENDIX B: DECOMPOSING STAND-ALONE SPACE USING WILSON-LOOPS: MPO SYMMETRIES

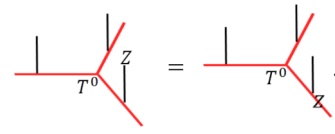
In Secs. IV B and IV C, we argued how the stand alone space M_0 decomposes further into two subspaces, $\mathbb{M}$ and $\mathbb{M}_0 - \mathbb{M}$ on the basis whether a stand-alone variation can be lifted to the physical level locally or nonlocally. In doing so, we used the fact about topological models: an anyonic excitation cannot be removed by a local operation but an elementary excitation can be.

There is another way to distinguish between trivial and nontrivial excitations. Consider the tensor network state made out of T^0, except at site s_0, T^0 has been replaced by some stand-alone tensor T. Now we want to find out whether this variation/excitation is a topologically nontrivial 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 nontrivial 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 nontrivial 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.

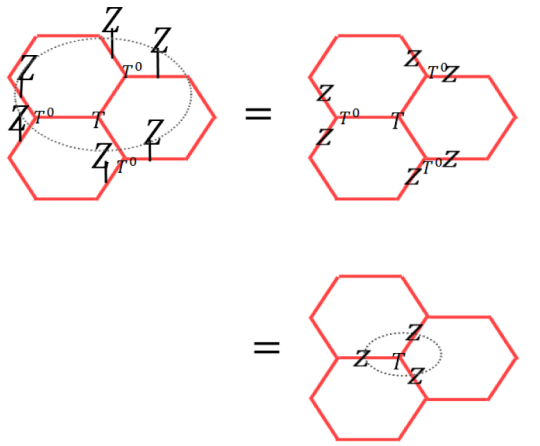
$W_a(C)$ is an operator on the physical degrees of freedom, which induces an operator, $M_a(C)$, on the virtual degrees of freedom. $W_a(C)$ is guaranteed to have a representation $M_a(C)$ on the virtual level because $W_a(C)$ is an operator supported on the ground-state physical space of local tensors, 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. However, in fact, any stand-alone variation T that is a 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 nontrivial 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. (29)] and single-line [Eq. (28)] 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 us illustrate the above discussion with the single-line TNR of toric code state. Let us assume 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. (40) in the opposite direction],



(B1)

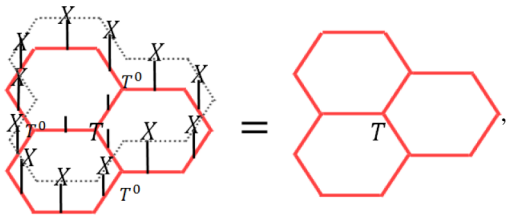
Keeping in mind that T^0 also satisfies the $Z^{\otimes 3}$ symmetry of Eq. (5), we see that the m-particle Wilson-loop finally reduces to a $Z^{\otimes 3}$ operator on the stand-alone tensor T. That is,



(B2)

In the first equality, we have used relation (B1) and in the 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.

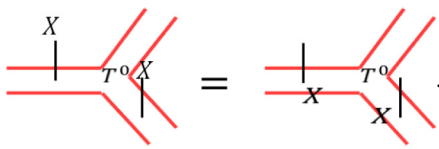
Now a natural question arises: why is not there an analogous symmetry constraint on the tensor corresponding to an e-string operator Wilson loop? Let us apply the e-particle Wilson loop, which is a loop of X operators on the single-line TNR, and then bring it down to the virual level. We find



(B3)

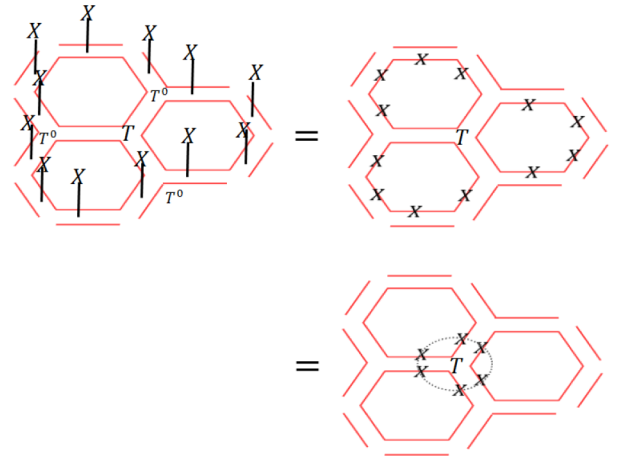
where we have used the fact that X operators on the nearby virtual legs simply cancel each other. This was already noted in the discussion of zero-string operators and in Eq. (53). So we see that the e-particle Wilson-loop poses no extra symmetry constraint on the stand-alone tensors.

Now let us see if the MPO symmetry of double-line TNR also comes from a Wilson-loop. The 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. (42) in the opposite direction]



(B4)

So we find



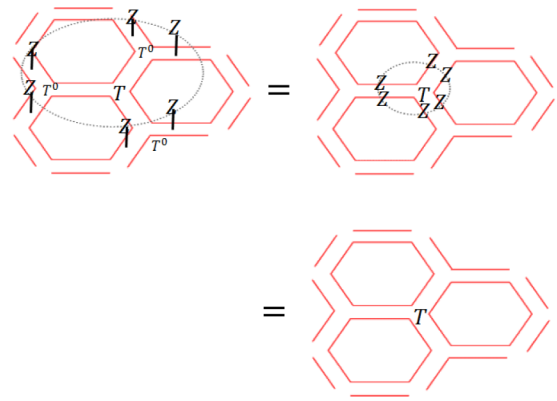
(B5)

where in the first equality, we used Eq. (B4), and in the second equality, we simply used the relation

(B6)

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 (B6) 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. Equation (56) 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



(B7)

The first equality follows from Eq. (56) 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-injective subspace had only one Z_2 symmetry. In fact, this analysis has shown what we already knew from Eq. (54): 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. However, there is a crucial difference. $W_e(C)$ has trivial representation on *all* of the virtual spaces of single-line TNRs, but $W_m(C)$ has a trivial representation *only* in a subspace of the virtual space of the 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 possible as is guaranteed by the MPO-injectivity (lemma 1). In fact, this representation would be faithful on individual tensors. However, 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.

APPENDIX C: A BRIEF REVIEW OF STRING-NET MODELS

String-net models, which are Hamiltonian realizations of Turaev-Viro TQFTs, are introduced by Ref. [5] as RG fixed point models that describe topological order in $2+1$-space-time 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 of 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, and k are allowed to meet at a point, and $\delta_{ijk} = 0$ otherwise.

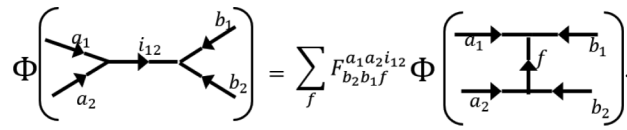
(3) *Quantum dimensions.* For every string type s, there is a value d_s associated to it, called its quantum dimensions. $D = \sum_s d_s^2$ is called the "total quantum dimension."

(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 us denote the string-net configurations with X. So a string-net condensed state is

$$|\Psi\rangle = \sum_X \Phi_X |X\rangle, \tag{C1}$$

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 Eqs. (4)–(7) of [5]. The most significant of these local constraints is the so-called "F" move.

(5) *F symbols.* A local constraint involving rebranching of five strings is the following:



(C2)

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

$$F^{ijk}_{j^\star i^\star 0} = \frac{\sqrt{d_k}}{\sqrt{d_i}\sqrt{d_j}} \delta_{ijk}, \tag{C3}$$

$$F^{ijm}_{kln} = F^{lkm^\star}_{jin} = F^{jim}_{lkn^\star} = F^{imj}_{k^\star nl} \frac{\sqrt{d_m d_n}}{\sqrt{d_j d_l}}. \tag{C4}$$

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 the tetrahedron. Then it is invariant under all tetrahedron symmetries. It satisfies an important equation, the so-called "pentagon identity":

$$\sum_f d_f G^{b_1 b_2 i_{12}}_{a_2 a_1 f} G^{b_2 b_3 i_{23}}_{a_3 a_2 f} G^{b_3 b_1 i_{31}}_{a_1 a_3 f} = G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{C5}$$

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

$$H = -\sum_v A_v - \sum_p B_p, \tag{C6}$$

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

$$A_v = \sum_{i,j,k} \delta_{ijk} |ijk\rangle\langle ijk|. \tag{C7}$$

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

$$B_p = \sum_s \frac{d_s}{D} B^s_p, \tag{C8}$$

where B^s_p 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 $(N, d_i, \delta_{ijk}, F^{ijk}_{klm})$ describe a string-net model.

1. 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,

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

G symbols also satisfy an "orthogonality identity,"

$$\sum_{i_{12}} G^{b_1 b_2 i_{12}}_{a_2 a_1 f} G^{b_1 b_2 i_{12}}_{a_2 a_1 f'} d_{i_{12}} = \frac{1}{d_f} \delta_{f,f'} \delta_{a_1 a_2 f} \delta_{b_1 b_2 f}. \tag{C9}$$

G symbols are normalized as

$$G^{b_1 b_2 i_{12}}_{a_2 a_1 0} = \delta_{a_1,b_1} \delta_{a_2,b_2} \delta_{a_1 b_1 i_{12}} \left(d_{a_1} d_{a_2}\right)^{-\frac{1}{2}}. \tag{C10}$$

Cyclic products of G symbols satisfy the following equation:

$$\sum_{\{b_j\}} \prod_{j=1}^{n} \left(G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} G^{c_j c_{j+1} i_{j,j+1}}_{b_{j+1} b_j f} d_{b_j}\right)$$

$$= \sum_s \delta_{f f' s} \prod_{j=1}^{n} \left(G^{c_j c_{j+1} i_{j,j+1}}_{a_{j+1} a_j s}\right). \tag{C11}$$

Plaquette operators B_p^f correspondingly satisfy

$$B_p^f B_p^{f'} = \sum_s \delta_{f f' s} B_p^s. \tag{C12}$$

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

$$d_f d_{f'} = \sum_s \delta_{f f' s} d_s, \tag{C13}$$

where d_f are nothing but the eigenvalues of the plaquette operators B^f 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

$$|s_q\rangle = \sum_a s_{q;a} |a\rangle. \tag{C14}$$

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

$$\langle s_q | s_{q'} \rangle \propto \delta_{q,q'}, \tag{C15}$$

$$\langle s_q | N^k | s_{q'} \rangle = \sum_{a,b} s_{q;a} \delta_{k,a,b} s_{q';b} \propto \delta_{q,q'}. \tag{C16}$$

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 nontrivial (modularity), the s above are just the columns of S matrix. However, we do not really need this for our results.

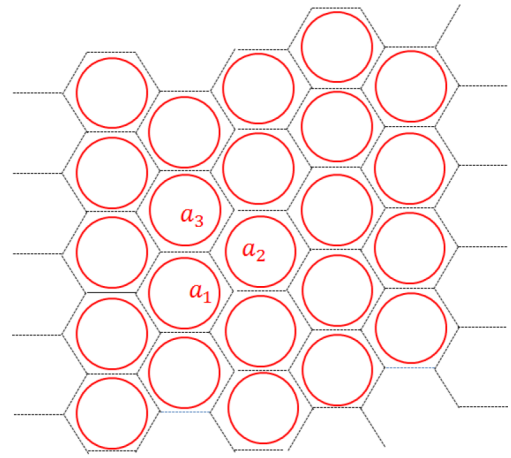


FIG. 16. A loop state on the fat lattice. Fat lattice means strings are allowed to move away from the edges, as long as they do not cross the center of the plaquettes.

APPENDIX D: 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 Ref. [6]. It is important to understand this derivation as it gives us a way to apply string-operators on triple-line TNR.

A string net RG fixed point ground state can be constructed by applying the plaquette operator $B_p = \sum_a d_a B_p^a$ to the vacuum state $|0\rangle$. B_p^a creates an a-type string loop on the plaquette p,

$$|\Psi_{\rm gs}\rangle = \prod_p B_p |0\rangle = \prod_p \sum_a d_a B_p^a |0\rangle$$

$$= \sum_{a_1, a_2, \ldots} d_{a_1} d_{a_2} \ldots |a_1, a_2, \ldots\rangle, \tag{D1}$$

where

$$|a_1, a_2, \ldots\rangle = B_{p_1}^{a_1} B_{p_2}^{a_2} \ldots |0\rangle. \tag{D2}$$

$|a_1, a_2, \ldots\rangle$ is a string configuration on the "fattened lattice." We will refer to $d_{a_1} d_{a_2} \ldots |a_1, a_2, \ldots\rangle$ as the "loop state." See Fig. 16. We need to fuse these loops together to get the physical state. We then fuse these strings together to get the final physical state,

$$|a_1, a_2, \ldots\rangle = \sum_{i_{12}, i_{23} \ldots} \Phi^{i_{12} i_{23} \ldots}_{a_1 a_2 a_3 \ldots} |i_{12}, i_{23}, \ldots\rangle. \tag{D3}$$

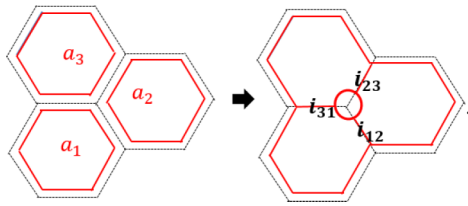
There are essentially three steps leading up to the expression of the triple-line TNR. We mention them here explicitly as we will need to refer back to them for other calculations.

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

$$|\Psi_0\rangle = \sum_{a_1, a_2, \ldots} d_{a_1} d_{a_2} \ldots |a_1, a_2, \ldots\rangle. \tag{D4}$$

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

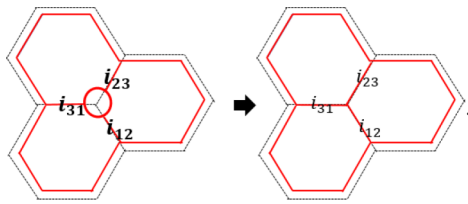
Step 2. We fuse all loops with nearby loops producing a string on the links



(D5)

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

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



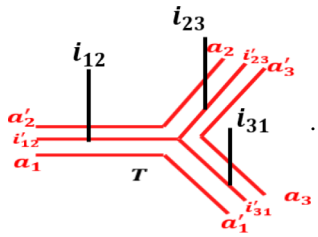
(D6)

Removing it produces a factor of $\sqrt{d_{a_j} d_{a_k} d_{a_l}} G^{i_{kl} i_{lj} i_{jk}}_{a_j a_k a_l}$.

Putting the three steps together, we get

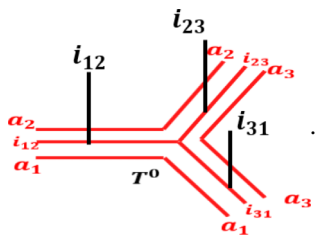
$$(T^0)^{i_{jk} i_{kl} i_{lj}}_{s_l s_j s_k} = \sqrt[4]{d_{i_{jk}} d_{i_{kl}} d_{i_{lj}}} G^{i_{jk} i_{kl} i_{lj}}_{a_j a_k a_l} \sqrt[6]{d_{a_j} d_{a_k} d_{a_l}}. \tag{D7}$$

A general triple-line tensor is represented diagrammatically as



(D8)

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



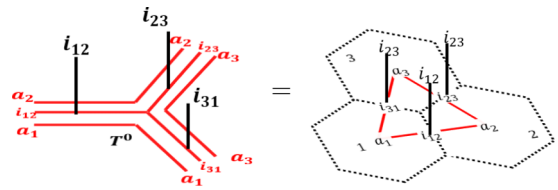
(D9)

APPENDIX E: PROOF OF THEOREM 1

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.

1. 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,



(E1)

The inner indices $a_1, a_2, \ldots$ 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

$$(T^0)^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} = \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{4}} d_{a_j}^{\frac{1}{6}}\right) G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3}. \tag{E2}$$

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

$$\mathbb{T}^0 = \prod_{j=1}^{n} \left(d_{i_{j,j+1}}^{\frac{1}{2}} \left(d_{a_j} d_{b_j}\right)^{\frac{1}{6}}\right) G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{E3}$$

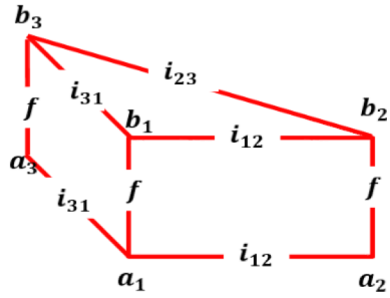
Using the pentagon equation $G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3} = \sum_f d_f \prod_{j=1}^{3} (G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f})$, we get

$$\mathbb{T}^0 = \sum_f d_f B_f, \tag{E4}$$

$$B_f = \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{2}} \left(d_{a_j} d_{b_j}\right)^{\frac{1}{6}}, G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}\right). \tag{E5}$$

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

$$\mathbb{T}^0 = \sum_f d_f$$



(E6)

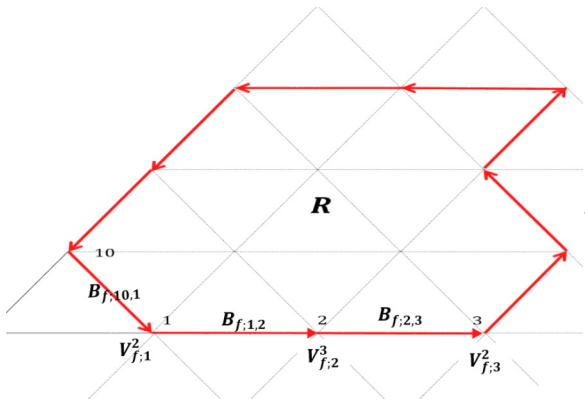
It is useful to decompose B_f into terms that sit on the edge of the triangle and terms that sit on the vertices,

$$B_f = \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}}\right) \prod_{j=1}^{3} \left((d_{a_j} d_{b_j})^{\frac{1}{6}}\right). \quad \text{(E7)}$$

The first cyclic product on the right-hand side (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)

$$\mathbb{T}^0(\Delta) = \sum_f d_f B_f(\partial\Delta). \quad \text{(E8)}$$

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,



(E9)

Suppose the vertices on the boundary of a region R on the triangluar lattice are labeled as $j = 1, 2, \ldots, n$ [see Eq. (E9)]. 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 $d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}}$. With this construction, $B_f(\partial R)$ can be written as

$$B_f(\partial R) = \prod_{j=1}^{n} \left(d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}}\right) \prod_{j=1}^{n} \left((d_{a_j} d_{b_j})^{\frac{m_j}{6}}\right). \quad \text{(E10)}$$

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

2. 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. (E8):

$$\mathbb{T}^0(R) = D^V \sum_f d_f^{\chi_R} B_f(\partial R), \quad \text{(E11)}$$

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 $\mathbb{T}^0(R)$,

$$\begin{aligned}\mathbb{T}^0(R) &= \mathrm{tTr}(\mathbb{T}^0(\Delta_1)\mathbb{T}^0(\Delta_2)\ldots) \\ &= \sum_{f_1, f_2, \ldots} \left(d_{f_1} d_{f_2} \ldots\right) B_{f_1}(\partial\Delta_1) B_{f_2}(\partial\Delta_2)\ldots, \end{aligned} \quad \text{(E12)}$$

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. (C9), edge contraction on the edge $(j, j+1)$ between B_f and $B_{f'}$ gives

$$\begin{aligned} Ev(B_f B_{f'}) &\propto \sum_{i_{j,j+1}} d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}} d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f'}^{b_j b_{j+1} i_{j,j+1}} \\ &= \frac{1}{d_f} \delta_{f,f'} \delta_{a_j a_{j+1} f} \delta_{b_j b_{j+1} f}. \end{aligned} \quad \text{(E13)}$$

The factor $\delta_{f,f'}$ implies that B_f only contracts with B_f. So the expression in Eq. (E12) is only nonzero for $f_1 = f_2 = \ldots f$. So we have

$$\mathbb{T}^0(R) = \sum_f d_f^F B_f(\partial\Delta_1) B_f(\partial\Delta_2)\ldots, \quad \text{(E14)}$$

where F is the number of faces in region R. Then there are factors of $\delta_{a_j a_{j+1} f} \delta_{b_j b_{j+1} f}$ in Eq. (E13) 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 $\delta_{a_j b_j f}$ that came out of edge contraction. So, finally we have the vertex contraction using identity (C13),

$$\sum_{a_j, b_j} d_{a_j} d_{b_j} \delta_{a_j b_j f} = \sum_{a_j} d_f d_{a_j} d_{a_j} = D d_f. \quad \text{(E15)}$$

So we see that contraction of 6 tensors on each vertex simply produces a factor of $D d_f$ for every f-type boundary operator.

When we are done with all the vertex contractions, we will have an overall $(Dd_f)^V = D^V d_f^V$ factor. Putting all the factors together, we get

$$\mathbb{T}^0(R) = \sum_f d_f^F d_f^{-E} (Dd_f)^V B_f(\partial R) = D^V \sum_f d_f^{\chi_R} B_f. \tag{E16}$$

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.

$$\lim_{|\partial R|\to\infty} \frac{\mathrm{Tr}(B_{f\neq 0}(\partial R))}{\mathrm{Tr}(B_0(\partial R))} = 0. \tag{E17}$$

Proof. To prove this, we would calculate S_{topo} on a sphere using the virtual-density method laid out in Appendix A in the previous chapter, and compare it to the known result, $S_{\mathrm{topo}} = \ln D$. We divide the sphere in symmetric two halves, 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 Ref. [27], 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 us assume $\rho_R = N\sigma_b^2$, 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 us put the string-net tensor network state on a sphere. Consider the left hemisphere, denoted as L, and right hemisphere, denoted as R. Let us 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

$$B_f(\partial R) = \prod_{j=1}^n \left(d_{i_{j,j+1}}^{\frac{1}{2}} \left(d_{a_j} d_{b_j}\right)^{\frac{m_j}{6}} G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}}\right). \tag{E18}$$

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

$$B_f(\partial R) = \prod_{j=1}^n \left(d_{i_{j,j+1}}^{\frac{1}{2}} \left(d_{a_j} d_{b_j}\right)^{\frac{1}{2}} G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}}\right). \tag{E19}$$

Note that, using relation (C10),

$$B_0(\partial R) = \prod_{j=1}^n \left(d_{i_{j,j+1}}^{\frac{1}{2}} \left(d_{a_j} d_{b_j}\right)^{\frac{1}{2}} G_{a_{j+1} a_j 0}^{b_j b_{j+1} i_{j,j+1}}\right) \tag{E20}$$

$$= \prod_{j=1}^n \left(d_{i_{j,j+1}}^{\frac{1}{2}}\right) \delta_{a_j b_{j+1} i_{j,j+1}}. \tag{E21}$$

Now, using identity (C11) the algebra of B_f operators is

$$B_f B_{f'} = \sum_s \delta_{ff's} B_s \times \prod_{j=1}^n \left(d_{i_{j,j+1}}^{\frac{1}{2}}\right) \tag{E22}$$

$$= \sum_s \delta_{ff's} B_s B_0. \tag{E23}$$

We also know how to contract B_f with each other through the calculations done previously in the previous section. 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 the number of edges. So we get

$$\mathrm{Tr}(B_f B_{f'}) = \delta_{f,f'} D^n. \tag{E24}$$

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

$$\mathrm{Tr}(B_f) = \sum_{\{a_j i_{j,j+1}\}} \prod_{j=1}^n \left(d_{i_{j,j+1}}^{\frac{1}{2}} \left(d_{a_j} d_{a_j}\right)^{\frac{1}{2}} G_{a_{j+1} a_j f}^{a_j a_{j+1} i_{j,j+1}}\right) = \mathrm{Tr}\left(A_f^n\right), \tag{E25}$$

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

$$\lim_{n\to\infty} \mathrm{Tr}(B_0) = \lambda_0^n. \tag{E26}$$

For Abelian models, $\mathrm{Tr}(B_{f\neq 0}) = 0$ since $G_{baf}^{abi} = 0$, $f \neq 0$. For the double-Fibonacci model, see Sec. VIII, 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,

$$\mathrm{Tr}\left(\sigma_b^2\right) = \mathrm{Tr}\left(\sum_f d_f B_f\right)^2 = \sum_{f,f'} d_f d_{f'} \mathrm{Tr}(B_f B_{f'}) = \sum_{f,f'} d_f d_{f'} \delta_{f,f'} D^n = D^n \left(\sum_f d_f^2\right). \tag{E27}$$

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

$$\begin{aligned} S_{1/2}(\rho_R) &= \frac{1}{1-1/2} \ln \mathrm{Tr}\left(\rho_R^{\frac{1}{2}}\right) \\ &= 2 \ln \frac{\mathrm{Tr}\left(\sum_f d_f B_f\right)}{\sqrt{D^n \sum_f d_f^2}} \\ &= -n \ln D - 2 \ln \sum_f (d_f \mathrm{Tr} B_f) - \ln \sum_f d_f^2 \\ &= -n \ln D - 2n \ln \lambda_0 - 2 \ln \left(1 + \sum_{f>0} \frac{\mathrm{Tr} B_f}{\lambda_0^n}\right) \\ &\quad - \ln \sum_f d_f^2. \end{aligned} \tag{E28}$$

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

3. String-net stand-alone subspace

Now we combine lemmas 2 and 3 to prove theorem 1. That is, we prove now that the stand alone space of the triple-line string net TNR is given by

$$M_0 = \delta_{a_1,a_2,i_{12}}\delta_{a_2,a_3,i_{23}}\delta_{a_3,a_1,i_{31}}. \tag{E29}$$

Proof. Now we are ready to calculate the stand-alone space. Consider the same tensor network but on a very large disk 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,

$$\mathbb{T}(D-\Delta) = \sum_f B_f(\partial(D-\Delta)) \tag{E30}$$

$$= \sum_f B_f(\partial\Delta)\otimes B_f(\partial D). \tag{E31}$$

To get the stand-alone space, we simply trace out the inner indices on the outer edge. But according to lemma 3, only $\mathrm{Tr}(B_0)$ contribute in the large disk limit. So we simply get (up to an overall normalization factor which we ignore) B_0 on the triangle,

$$\lim_{|D|\to\infty} \mathbb{T}_D(D-\Delta) = \lim_{|D|\to\infty} \sum_f B_f(\partial\Delta)\otimes \mathrm{Tr}(B_f(\partial D)) \tag{E32}$$

$$= B_0(\partial\Delta)\lambda_0^n. \tag{E33}$$

However, using (E20), we get

$$B_0(\partial\Delta) = \left(d_{i_{12}}d_{i_{23}}d_{i_{31}}\right)^{\frac{1}{2}}\delta_{a_1,a_2,i_{12}}\delta_{a_2,a_3,i_{23}}\delta_{a_3,a_1,i_{31}}. \tag{E34}$$

A 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 a triple-line TNR of a general string net is

$$M_0 = \delta_{a_1,a_2,i_{12}}\delta_{a_2,a_3,i_{23}}\delta_{a_3,a_1,i_{31}}. \tag{E35}$$

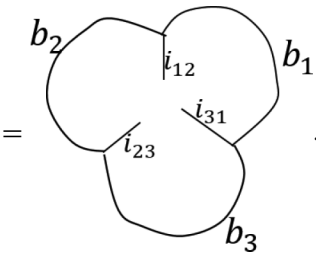
This completes the proof. ■

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

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

$$|b_1,b_2,b_3;i_{12},i_{23},i_{31}\rangle, \tag{E36}$$

$$\left|\left\{\prod_{k=1}^{3}\delta_{b_k,b_{k+1},i_{k,k+1}}\right\}\right\rangle = \tag{E37}$$



So we get

$$\dim(M_0) = \sum_{b_1,b_2,b_3;i_{12},i_{23},i_{31}} \delta_{b_1,b_2,i_{12}}\delta_{b_2,b_3,i_{23}}\delta_{b_3,b_1,i_{31}}. \tag{E38}$$

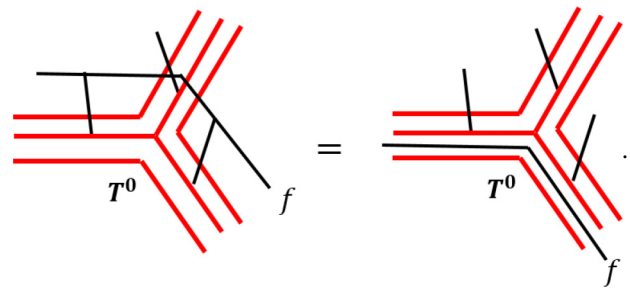
4. String-net MPO symmetries from Wilson-loop operators

In last chapter, we showed how MPO symmetries or the MPO-injective 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 Eq. (E39).

That is, it simply becomes an f-type string which is then fused with the plaquette legs. We consider the Wilson loop that encircles the three plaquettes of the tensor. This Wilson loop creates an f-type string that then fuses with the three 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 $\langle\{\delta_{a_k,a_{k+1},i_{k,k+1}}|W_f|\{\delta_{b_k,b_{k+1},i_{k,k+1}}\}\rangle$. 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 three plaquettes and calculate the induced operator on the stand-alone basis. It can be done in a convenient way using string-net diagrams in Eq. (E37).

There are essentially 3 steps.

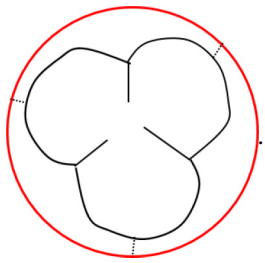
Step 1. Since the surrounding tensors are the fixed point tensor T^0, the Wilson loop on the physical level simply becomes an f-type string that fuses with the plaquette legs,



(E39)

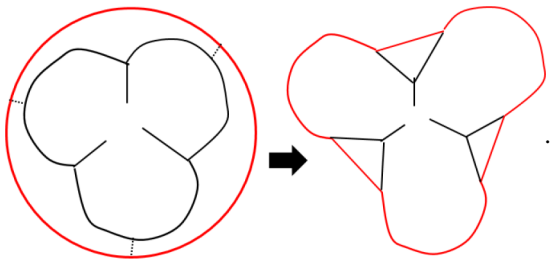
Since these plaquette legs are contracted with the plaquette legs of the stand-alone tensor, it is equivalent to fusing the f-string

loop with the three plaquette legs of the stand-alone tensor,



(E40)

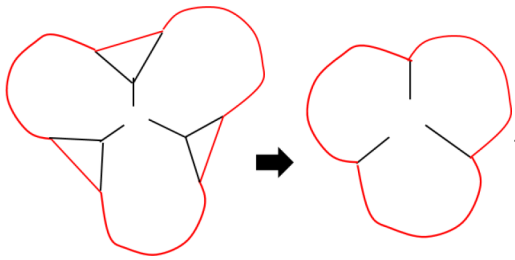
Step 2. We fuse these strings with the three nearby strings a_1, a_2, and a_3,



(E41)

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

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



(E42)

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

Collecting the factors from steps 2 and 3, we get

$$\left\langle \left\{ \prod_{k=1}^{3} \delta_{b_k, b_{k+1}, i_{k,k+1}} \right| W_f \left| \left\{ \prod_{k=1}^{3} \delta_{a_k, a_{k+1}, i_{k,k+1}} \right\} \right\rangle = \prod_{j=1}^{3} d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}. \tag{E43}$$

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

$$\mathbb{M} = \sum_f \frac{d_f}{D} \prod_{j=1}^{3} d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} = \frac{1}{D} d_{b_j}^{\frac{1}{6}} d_{a_j}^{\frac{1}{6}} G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{E44}$$

It should be understood as an operator written in its components in the basis $|\{a_k; i_{k,k+1}\}\rangle\langle\{b_k; i_{k,k+1}\}|$. We used the pentagon identity in the second step. We can see that it projects onto 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. (82). 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. However, 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 on 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

$$M_f = \prod_{j=1}^{3} d_{a_j} \left(d_{b_j} d_{a_j}^{-1}\right)^{\frac{1}{6}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \tag{E45}$$

$$\Rightarrow \mathbb{M} = \frac{1}{D} d_{a_j} \left(d_{b_j} d_{a_j}^{-1}\right)^{\frac{1}{6}} G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{E46}$$

The fixed point triple-line tensor satisfies

$$M_f T^0 = d_f T^0, \tag{E47}$$

$$\mathbb{M} T^0 = T^0. \tag{E48}$$

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

$$M_f(\partial R) = \prod_{j=1}^{n} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} d_{a_j} \left(d_{b_j} d_{a_j}^{-1}\right)^{\frac{\theta_j}{2\pi}}. \tag{E49}$$

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

APPENDIX F: 0-TYPE STRING OPERATOR IS A ZERO-STRING OPERATOR OF TRIPLE-LINE TNR

In the last Appendix, we argued how the reason for instability is that some of the nontrivial 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 have 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. (E49) along the path with some changes at the end. However, we do not 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. (E49), it is immediately clear what the invisible string operators are for the triple-line TNR of general string nets. For $f = 0$ [using identity (C10)],

$$\begin{aligned} M_0(\partial R) &= \prod_{j=1}^{n} \left(G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j 0} d_{b_j} \left(d_{a_j} d_{b_j}^{-1}\right)^{\frac{m_j}{6}}\right) \\ &= \prod_{j=1}^{n} \left(\delta_{a_j,b_j} \left(d_{a_j} d_{b_j}\right)^{-\frac{1}{2}} d_{b_j} \left(d_{a_j} d_{b_j}^{-1}\right)^{\frac{m_j}{6}} \delta_{a_j,a_{j+1},i_{j,j+1}}\right) \\ &= \prod_{j=1}^{n} \delta_{a_j,a_{j+1},i_{j,j+1}}. \end{aligned} \tag{F1}$$

However, 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 nontrivial 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 nontrivial 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 nontrivially 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" [12]. 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."

APPENDIX G: PROOF OF THEOREM 2

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 $S_{\text{topo}}(\epsilon)$.

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

Lemma 4. let us divide the cylinder in two halves [Fig. 15(a)]. We denote the right half as R. If any given tensor network on this cylinder satisfies

$$\lim_{|R|\to\infty} \mathbb{T}(R) = C^{|R|} \sum_f c_f B_f(\partial R), \tag{G1}$$

where C is some constant, then S_{topo}, as given in Eq. (A6), is

$$S_{\text{topo}} = \ln \sum_f \left(\frac{c_f^2}{c_0^2}\right). \tag{G2}$$

Proof. The proof is quite simple. We follow the same steps as used in the proof of lemma 3, replacing d_f with c_f. We first calculate the normalization of the density matrix.

$$\begin{aligned} \mathrm{Tr}(\sigma_b^2) &= \mathrm{Tr}\left(\sum_f c_f B_f\right)^2 \\ &= \sum_{f,f'} c_f c_{f'} \mathrm{Tr}(B_f B_{f'}) \\ &= \sum_{f,f'} c_f c_{f'} \delta_{f,f'} D^n \\ &= D^n \left(\sum_f c_f^2\right). \end{aligned} \tag{G3}$$

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

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

When we let $n \to \infty$ and using Eq. (3),

$$S_{1/2}(\rho_R) = n \ln \frac{\lambda_0^2}{D} - \ln \sum_f \left(\frac{c_f^2}{c_0^2}\right) \tag{G4}$$

$$\Rightarrow S_{\text{topo}} = \ln \sum_f \left(\frac{c_f^2}{c_0^2}\right). \tag{G5}$$

This completes the proof. ■

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

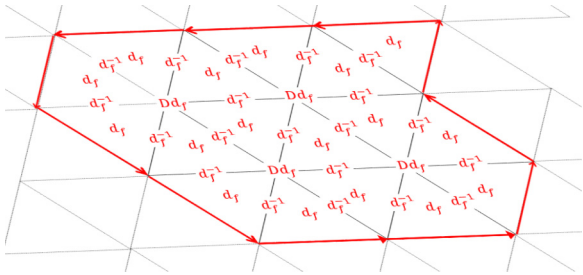
2. Instability in string net

Now we provide a proof of theorem 2.

Proof. Combination of the lemma 2, theorem E 3, and lemma 4 gives a clue to why $T^0 \to T^0 + \epsilon T^q$, 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 Sec. VII D 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 the 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



(G6)

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 us choose a particular tensor variation

$$T^q = \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{4}}\right) \delta_{a_j b_{j+1} i_{j,j+1}}, \tag{G7}$$

such that

$$\mathbb{T}^q = T^q (T^q)^\dagger = B_0. \tag{G8}$$

This tensor is supported on the full M_0 space and clearly has components outside the MPO-injective 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

$$\mathbb{T} = (T^0 + \epsilon T^q)(T^0 + \epsilon T^q)^\dagger \approx \mathbb{T}^0 + \epsilon^2 B_0 \tag{G9}$$

$$= (1+\epsilon^2) B_0 + \sum_{f>0} d_f B_f. \tag{G10}$$

We have ignored the linear terms in ϵ as they are contained within the MPO-injective subspace, and we do not 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. (G5), $S_{\text{topo}} = 0$.

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. (C14):

$$(T^q) = \prod_{j=1}^{3} \left(d_{i_{j,j+1}}^{\frac{1}{4}} \left(d_{a_j} + \epsilon s_{q;a_j}\right)^{\frac{1}{6}}\right) G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3}. \tag{G11}$$

This tensor is clearly supported on the stand-alone space and is outside the MPO-injective subspace as to be inside the MPO-injective 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. However, now the factors on the vertices would be

$$\sum_{a,b} \delta_{a,b,f} (d_a + \epsilon s_{q;a})(d_b + \epsilon s_{q;b}) = D(d_f + e_{q;f}\epsilon^2), \tag{G12}$$

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 $N^f_{a,b} = \delta_{a,b,f}$. A conclusion similar to that for the 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 TNRs of general string-net states have at least one unstable direction. ■

APPENDIX H: 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 disk geometry, S_{topo} is known to depend only on the total quantum dimension D, $S_{\text{topo}} = \ln D$. However, when the boundary of A consists of noncontractible topologically nontrivial loops, for example, when a torus or cylinder is divided into two cylinders, it has been shown by Ref. [30] that S_{topo} also depends on the linear combination of ground states. For a ground-state wave function on a torus,

$$|\Psi\rangle = \sum_a c_a |\Xi_a\rangle, \tag{H1}$$

where the sum is over the degenerate ground states labeled by quasiparticles of the model, the nth Rényi entropy is given by

$$S_n = \alpha_n L - S_{\text{topo}}, \tag{H2}$$

$$S_{\text{topo}} = 2 \ln D - \frac{1}{1-n} \ln \left(\sum_a p_a^n d_a^{2(1-n)} \right), \tag{H3}$$

where d_a is the quantum dimension of ath quasiparticle 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.

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

$$S_{\text{topo}} = 2 \ln 2 - \frac{1}{1-n} \ln \left(p_1^n + p_2^n + p_3^n + p_4^n\right),$$

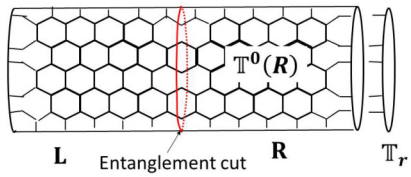


FIG. 17. We calculate entanglement entropy of the right-half of the cylinder with a certain boundary condition $\mathbb{T}_r$. The entanglement cut is in the middle of the cylinder.

when $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$, we get $S_{\text{topo}} = 0$ although the topological order is not lost. So one has to be careful using S_{topo} as an indicator of topological order.

Let us first take the example of the single-line TNR of the toric code (see Fig. 18). 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. 17). The four MES correspond to four eigenstates of e and m Wilson loops on the entanglement cut. However, since e-Wilson loop is a zero-string operator, the state is always in its $+1$ eigenstate [Fig. 18(a)]. So we have access to only two MES corresponding to ± 1 eigenstates of the m-Wilson loop. We also know that the state is in the $+1$ eigenstate of the $Z^{\otimes}_{\partial R} = Z^{\otimes L}_{ec} \otimes Z^{\otimes L}_{r}$, where subscript ec stands for loop at entanglement cut, and r stands for the loop at the right boundary of R. Since the state is in the $+1$ eigenstate of $Z^{\otimes L}_{ec} \otimes Z^{\otimes L}_{r}$ [see Fig. 18(b)], the state can be either in the $+1$ eigenstate of both $Z^{\otimes L}_{r}$ and $Z^{\otimes L}_{ec}$ or in the -1 eigenstate of both. The boundary tensor determines which eigenstate of $Z^{\otimes L}_{r}$ the wave function is in, and consequently also which eigenstate of $Z^{\otimes L}_{ec}$. This is how the boundary tensors and MES are connected. Since we have access to only two MES

$$S_{\text{topo}} = \ln 2 - \frac{1}{1-n} \ln\left(p_1^2 + p_2^2\right). \tag{H4}$$

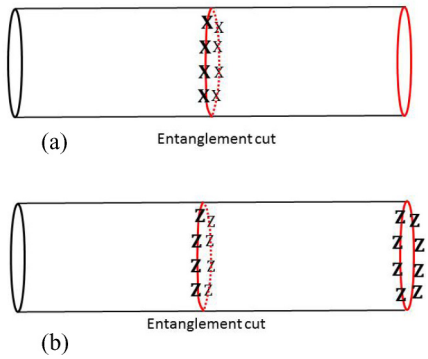


FIG. 18. MESs are eigenstates of different Wilson loop operators at the entanglement cut. (a) For a fixed point single-line TNR, the state on the cylinder is always in $+1$ eignestate of X loop, as it identically disappears. (b) The state is also in $+1$ eigenstate of a simultaneous operation of two Z loops, one at the entanglement cut, other at the right-most boundary. It implies, we can be in two MESs depending on the 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.

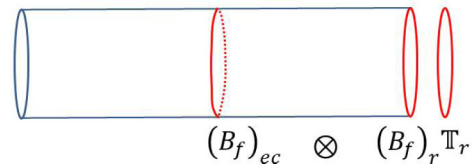


FIG. 19. Bulk double tensor is a sum of tensor products between $(B_f)_{ec}$ (B_f on the entanglement cut) and $(B_f)_r$ (B_f on the right boundary). So, when we contract a boundary tensor $\mathbb{T}_r$ with the bulk tensor, it contracts with $(B_f)_r$ giving a scalar c_f. So the resulting tensor is $\text{Ev}(\mathbb{T}(R)\mathbb{T}_r) = \sum_f c_f (B_f)_{ec}$. Consequently, S_{topo} using Eq. (G5) is simply $\ln\left(\sum_f \frac{c_f^2}{c_0^2}\right)$.

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 $+1$ eigenstate 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 lemma 4, $\rho_R = N\sigma_b^2$, where

$$\sigma_b = \mathbb{T}^0(R)\mathbb{T}_r, \tag{H5}$$

where $\mathbb{T}_r$ denotes the double tensor on the boundary. We know that, up to an irrelevant normalization constant,

$$\begin{aligned}\mathbb{T}^0(R) &= \sum_f d_f^{\chi_R} B_f(\partial R) \\ &= (B_0)_{ec} \otimes (B_0)_r + (B_1)_{ec} \otimes (B_1)_r,\end{aligned} \tag{H6}$$

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 us assume the boundary double tensor $\mathbb{T}_r$ contracts with $(B_f)_r$ to produce the constants c_f (see Fig. 19):

$$\begin{aligned}\sigma_b &= ((B_0)_{ec} \otimes (B_0)_r + (B_1)_{ec} \otimes (B_1)_r)\mathbb{T}_r \\ &= c_0 (B_0)_{ec} + c_1 (B_1)_{ec} \\ &= c_- B_- + c_+ B_+,\end{aligned} \tag{H7}$$

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:

$$B_\pm^2 = B_\pm, \quad \text{Tr}(B_\pm) = 2^{L-1}. \tag{H8}$$

With this, we get the normalized density matrix as

$$\rho_R = \frac{1}{2^L}\left(\frac{c_-^2}{c_-^2 + c_+^2} B_- + \frac{c_+^2}{c_-^2 + c_+^2} B_+\right) \tag{H9}$$

$$= \frac{1}{2^L}(p_- B_- + p_+ B_+). \tag{H10}$$

The nth Renyi entropy is

$$\begin{aligned}S_n(\rho_R) &= \frac{1}{1-n} \ln \text{Tr}\left(\rho_R^n\right) \\ &= \frac{1}{1-n} \ln \text{Tr}\left(\frac{1}{2^{nL}}(p_-^n B_- + p_+^n B_+)\right) \\ &= \frac{1}{1-n} \ln\left(\frac{1}{2^{nL}}(p_-^n 2^{L-1} + p_+^n 2^{L-1})\right) \\ &= L \ln 2 - \left(\ln 2 - \frac{1}{1-n} \ln(p_-^n + p_+^n)\right).\end{aligned} \tag{H11}$$

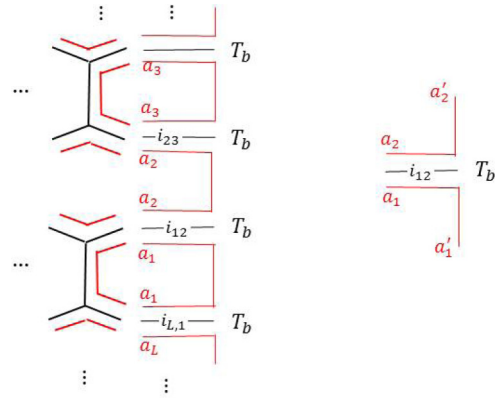


FIG. 20. Smooth boundary condition for triple-line tensor network. Tensors T_b are used on the boundary. T_b has five virtual legs, a_1, a_1', a_2, a_2', i_{12}, and one physical leg, i_{12}. Physical leg and the middle leg take the same values. We assign a particular value to the components of this tensor, $(T_b)^{i_{12}}_{i_{12}a_1a_1';a_2a_2'} = \delta_{i_{12},0}\delta_{a_1,a_1'}\delta_{a_2,a_2'}\delta_{a_1a_2i_{12}}$.

Comparing it with the MES formula in Eq. (H4), 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, $S_{\text{topo}} = \ln 2$. 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 Fig. 20. First, we will explain it for the triple-line tensors. For the double-line and single-line tensors, an appropriately reduced version of T_b will be used. Note that we have not drawn the physical index explicitly and it should be understood the same as the middle index (the index in black color). So the boundary tensor T_b has four virtual indices, and we fix its components to be

$$(T_b)^{i_{12}}_{i_{12}a_1a_1';a_2a_2'} = \delta_{i_{12},0}\delta_{a_1,a_1'}\delta_{a_2,a_2'}\delta_{a_1a_2i_{12}}, \tag{H12}$$

that is, we put the physical/middle index to zero (vacuum) and allow the plaquette legs to vary with this restriction. For the double-line tensors, we do not have a middle leg, but we can simply put the physical leg to 0. For the single-line tensors, 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

$$\mathbb{T}_r = \sum_{a,b} |a,a,a\ldots;000\ldots\rangle\langle b,b,b,\ldots;000\ldots|. \tag{H13}$$

So

$$\begin{aligned} c_f &= \text{Ev}(B_f\mathbb{T}_r) \\ &= \sum_{a,b}\prod_{j=1}^{m} G^{b,b,0}_{a,a,f}(d_a d_b)^{\frac{1}{2}} \\ &= \sum_{a,b}\prod_{j=1}^{m}\delta_{a,b,f} \\ &= \sum_{a,b}\delta_{a,b,f}. \end{aligned} \tag{H14}$$

Then using Eq. (G5), S_{topo} is simply $\ln(\sum_f \frac{c_f^2}{c_0^2})$. For the toric code, and double semion models $c_0 = c_1 = 2$, so we get $S_{\text{topo}} = \ln 2$. For the double Fibonacci model, however, we get

$$c_0 = \sum_{a,b}\delta_{a,b,0} = \delta_{0,0,0} + \delta_{1,1,0} = 2, \tag{H15}$$

$$c_1 = \sum_{a,b}\delta_{a,b,1} = \delta_{1,0,1} + \delta_{0,1,1} + \delta_{1,1,1} = 3. \tag{H16}$$

So we get $S_{\text{topo}} = \ln(1 + \frac{3^2}{2^2}) = \ln(1 + \frac{9}{4})$, 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 that gives us a precise analytical value of the topological entanglement entropy [namely, $\ln(\sum_f \frac{c_f^2}{c_0^2})$, with c_f given in Eq. (H14)] against which numerical calculations can be checked. (2) Though the 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} (=$\ln D$).

Numerical calculations of S_{topo} will be checked against the analytical result in Eq. (H14). 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 $S_{\text{topo}}(\epsilon)$? 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. However, $S_{\text{topo}}(\epsilon) = 0$ 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 clear 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. 21. We first fix the boundary tensor to be T_b given in Eq. (20) 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, $T_b \to T_b + \epsilon_b T_b^r$. 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. Figure 21 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 that 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

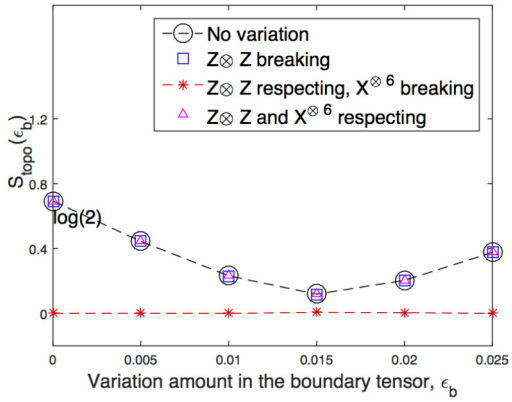


FIG. 21. Dependence of S_{topo} on boundary condition for toric-code double line TNR. We start with the boundary tensor, T_b, shown in Fig. 20. We add a random variation $\epsilon_b T_b^r$ to T_b and calculate $S_{\text{topo}}(\epsilon_b)$ for random bulk variations in different subspaces. We keep T_b^r 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.

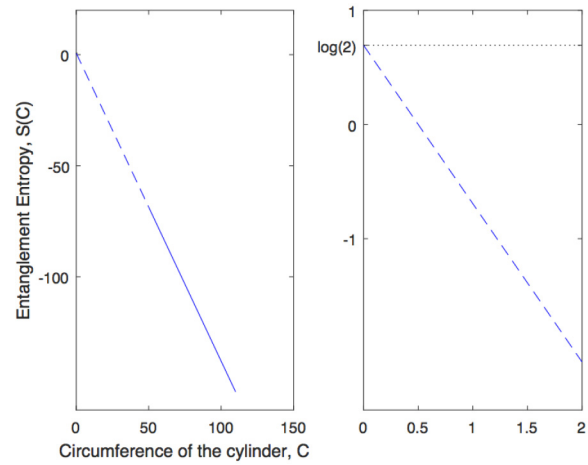


FIG. 22. Calculation of S_{topo} for a single-line toric code fixed point tensor network state. We fix the half-cylinder length as $L = 500$. The circumference is varied from 50 to 110. S varies linearly with C. This line is extrapolated back to $C = 0$. Its intersection with the y axis gives S_{topo}. Right figure is a zoomed in version of the left figure to show the intersection point clearly. We find $S_{\text{topo}} \approx \ln(2)$.

a specific boundary tensor chosen which gave an accidental zero.

(2) The variations that are stable (i.e., $S_{\text{topo}} = \ln 2$) for T_b, have the same value of S_{topo} as the fixed point tensor for all boundary tensors. It implies that a tensor network state with these variations indeed has 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.

APPENDIX I: 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 the numerical calculation of S_{topo}. We choose the simplest case, the single-line TNR of toric code. We first repeat the algorithm described in Appendix A 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 Appendix H. 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 us 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 $-S$ versus 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 an 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 us first look at the calculation for the single-line toric code fixed point tensor in Eq. (4). The half-cylinder length is fixed at $L = 500$. C is varied from 50 to 110. Figure 22 shows the entanglement entropy S versus 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 $S_{\text{topo}} = S(C = 0) \approx \ln(2)$ as expected. Figure 23 shows the dependence of S_{topo} on the half-cylinder length L. We see that there is no dependence, that is, 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 $\epsilon = 0.01$. The half-cylinder length is fixed at $L = 500$. C is varied from 50 to 110. Figure 24 shows entanglement entropy S versus 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. 25. We see that S_{topo} is ln(2) for

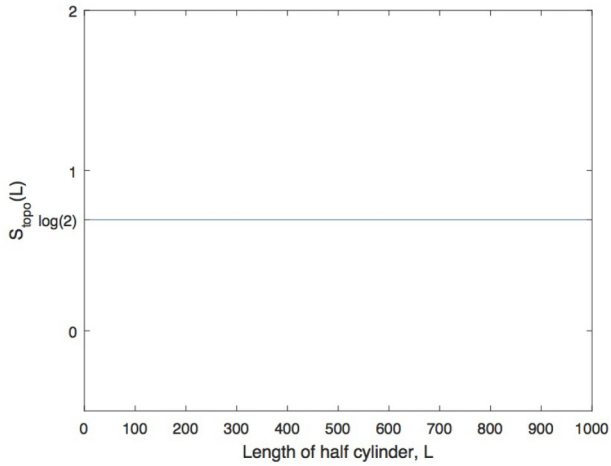


FIG. 23. S_{topo} was calculated for a fixed half-cylinder length, $L = 500$, in Fig. 22. We now vary L from 10 to 1000. We see that S_{topo} is converged even for small values of L. So one does not need a large cylinder length to get the right S_{topo} value. It is expected as it is an RG fixed point tensor network state.

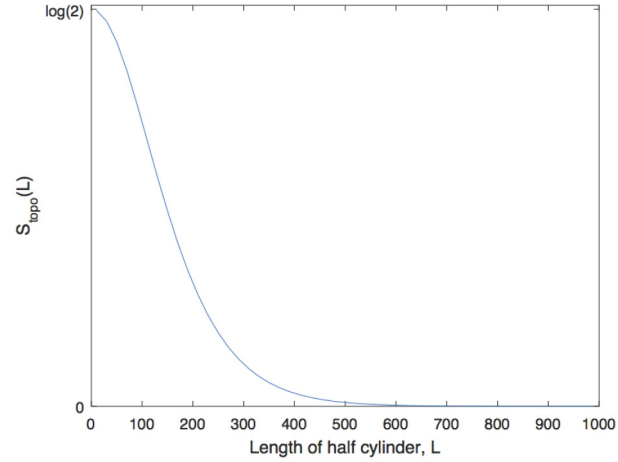


FIG. 25. S_{topo} was calculated for a fixed half-cylinder length, $L = 500$, in Fig. 24. We now vary L from 10 to 1000. We see that S_{topo} is close to ln(2) for small cylinders but converges to zero when the cylinder length L is increased from 1 to 1000. So it is indeed a topologically trivial state.

small cylinders but converges to zero as the length is increased. Comparing it to Fig. 23, 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 nonfixed 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 the corresponding convergence

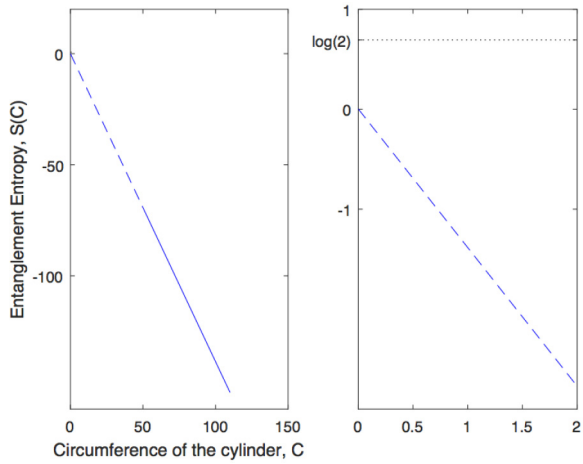


FIG. 24. Calculation of S_{topo} for a state represented by single-line toric code fixed point tensor varied with an MPO violating tensor. The strength of the variation is fixed at $\epsilon = 0.01$. We fix the half-cylinder length at $L = 500$. The circumference is varied from 50 to 110. S varies linearly with C. This line is extrapolated back to $C = 0$. Its intersection with the y axis gives S_{topo}. The right figure is a zoomed in version of the left figure to show the intersection point clearly. We find $S_{topo} \approx 0$, that is, it is a trivial state.

plots similar to Fig. 25. The results are shown in Fig. 26. We see that the strength of the variation has a huge effect on the convergence. Bigger variations lead to faster convergence.

Though we have presented details of the 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 the relevant parameters used in each case.

1. Single-line TNR toric code

The bond dimension of the MPS is kept fixed at $D_{cut} = 8$ at each step of the iteration. The starting MPS is as explained

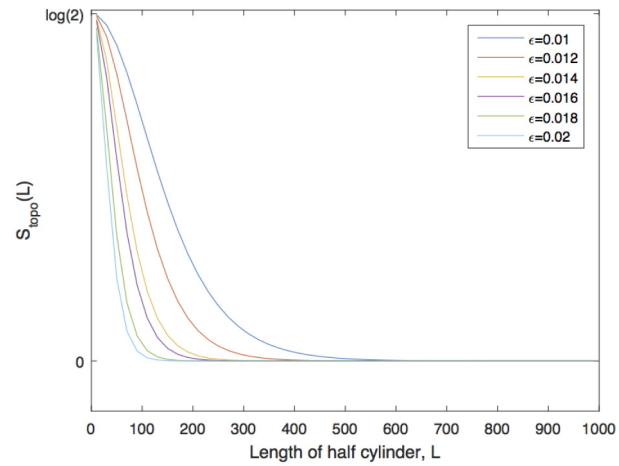


FIG. 26. The variation strength ϵ affects convergence. The higher the variation strength (as long as it is below any critical points) the faster is the convergence with the length of the size of the system.

in Appendix H. The strength of the variations is fixed at $\epsilon = 0.01$. The half-cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when the S_{topo} values 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. 3.

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

2. 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 Appendix H. The strength of the variations is fixed at $\epsilon = 0.01$. The half-cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when the S_{topo} values 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. 5.

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

3. 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 Appendix H. The strength of the variations is fixed at $\epsilon = 0.01$. The half-cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when the S_{topo} values 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. 10.

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

4. 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 Appendix H. The strength of the variations is fixed at $\epsilon = 0.2$. The half-cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when the S_{topo} values 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. 11.

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

5. 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 Appendix H. The strength of the variations is fixed at $\epsilon = 0.2$. The half-cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when the S_{topo} values 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. 12.

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

6. 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 Appendix H. The strength of the variations is fixed at $\epsilon = 0.1$. The half-cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when the S_{topo} values in two successive steps differ by less than 10^{-7}) or $L = 2000$, 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. 13.

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

[1] M. Fannes, B. Nachtergaele, and R. Werner, Commun. Math. Phys. **144**, 443 (1992).
[2] S. R. White, Phys. Rev. B **48**, 10345 (1993).
[3] F. Verstraete, V. Murg, and J. Cirac, Adv. Phys. **57**, 143 (2008).
[4] G. Vidal, arXiv:0912.1651 [cond-mat.str-el].
[5] M. A. Levin and X.-G. Wen, Phys. Rev. B **71**, 045110 (2005).
[6] Z.-C. Gu, M. Levin, B. Swingle, and X.-G. Wen, Phys. Rev. B **79**, 085118 (2009).
[7] O. Buerschaper, M. Aguado, and G. Vidal, Phys. Rev. B **79**, 085119 (2009).
[8] S. Yan, D. A. Huse, and S. R. White, Science **332**, 1173 (2011).
[9] H.-C. Jiang, Z. Wang, and L. Balents, Nat. Phys. **8**, 902 (2012).
[10] S. Depenbrock, I. P. McCulloch, and U. Schollwöck, Phys. Rev. Lett. **109**, 067201 (2012).
[11] X. Chen, B. Zeng, Z.-C. Gu, I. L. Chuang, and X.-G. Wen, Phys. Rev. B **82**, 165119 (2010).
[12] M. B. Şahinoğlu, D. Williamson, N. Bultinck, M. Mariën, J. Haegeman, N. Schuch, and F. Verstraete, arXiv:1409.2150.
[13] S. Bravyi, M. B. Hastings, and S. Michalakis, J. Math. Phys. **51**, 093512 (2010).
[14] O. Buerschaper, Ann. Phys. **351**, 447 (2014).
[15] A. Kitaev and J. Preskill, Phys. Rev. Lett. **96**, 110404 (2006).
[16] M. Levin and X.-G. Wen, Phys. Rev. Lett. **96**, 110405 (2006).
[17] T. Neupert, H. He, C. von Keyserlingk, G. Sierra, and B. A. Bernevig, Phys. Rev. B **93**, 115103 (2016).
[18] J.-W. Mei, J.-Y. Chen, H. He, and X.-G. Wen, Phys. Rev. B **95**, 235107 (2017).
[19] S. Dusuel and J. Vidal, Phys. Rev. B **92**, 125150 (2015).
[20] M. D. Schulz, S. Dusuel, K. P. Schmidt, and J. Vidal, Phys. Rev. Lett. **110**, 147203 (2013).
[21] M. D. Schulz, S. Dusuel, G. Misguich, K. P. Schmidt, and J. Vidal, Phys. Rev. B **89**, 201103 (2014).
[22] Z.-C. Gu, M. Levin, and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008).
[23] A. Kitaev, Ann. Phys. **303**, 2 (2003).
[24] N. Schuch, I. Cirac, and D. Pérez-García, Ann. Phys. **325**, 2153 (2010).
[25] G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015).
[26] M. Freedman, C. Nayak, K. Shtengel, K. Walker, and Z. Wang, Ann. Phys. **310**, 428 (2004).
[27] J. I. Cirac, D. Poilblanc, N. Schuch, and F. Verstraete, Phys. Rev. B **83**, 245134 (2011).
[28] S. T. Flammia, A. Hamma, T. L. Hughes, and X.-G. Wen, Phys. Rev. Lett. **103**, 261601 (2009).
[29] S. Dong, E. Fradkin, R. G. Leigh, and S. Nowling, J. High Energy Phys. 05 (2008) 016.
[30] Y. Zhang, T. Grover, A. Turner, M. Oshikawa, and A. Vishwanath, Phys. Rev. B **85**, 235151 (2012).
[31] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).