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

Sujeet K. Shukla

Institute of Quantum Information and Matter, California Institute of Technology, California, 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, CA 91125, USA
(Dated: October 4, 2016)

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 dimension. However, tensor network representations may be vulnerable to instabilities caused by small perturbations of the local tensor, especially when the local tensor is not injective. For example, the topological order in tensor network representations of the toric code ground state has been shown in Ref.1 to be unstable under certain small variations of the local tensor, if these small variations do not obey a local Z_2 symmetry of the tensor. In this paper, we ask the questions of 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 representation of all string-net models are indeed unstable, but the matrix product operator (MPO) symmetries of the local tensor identified in Ref.2 can help to protect the order. We find that, 'stand-alone' variations that break the MPO symmetries lead to instability because they induce the condensation of bosonic quasi-particles and destroy the topological order in the system. Therefore, such variations must be forbidden for the encoded topological order to be reliably extracted from the local tensor. On the other hand, if a tensor network based variational algorithm is used to simulate the phase transition due to boson condensation, then such variation directions must be allowed in order to access the continuous phase transition process correctly.

CONTENTS

I. INTRODUCTION

The tensor network representation of quantum states (including the matrix product states in 1D)[3–6] 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[7], can be represented exactly with simple tensors[8,9]. Moreover, numerical studies applied to realistic models have identified nontrivial topological features in the ground state wave function (see e.g. Ref.10–12).

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 completely different result as to what topological order it represents. In particular, Ref. 1 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. 1 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:

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 requirement is 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 Ref.2 and 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 quasi-particles and hence destroys (totally or partially) the topological order.

To support the above claims, 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 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 $\lim_{\epsilon \to 0} S_{\text{topo}}(\epsilon) =$
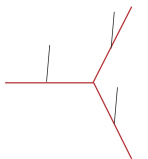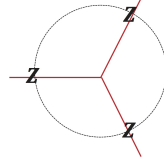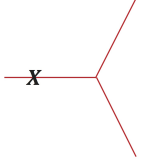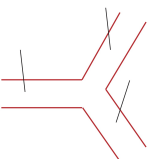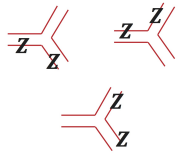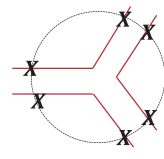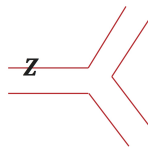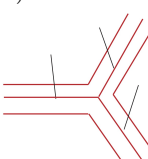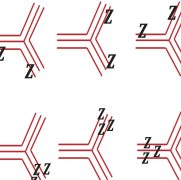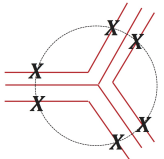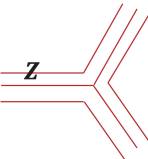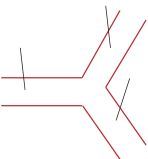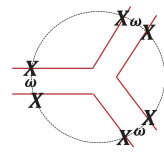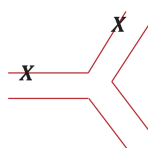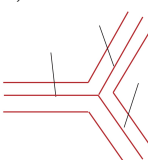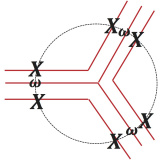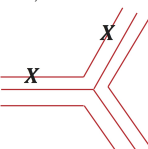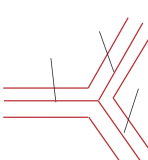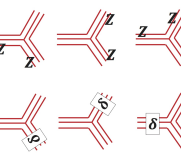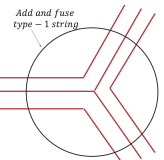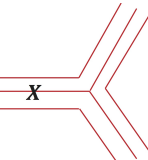
Topological Model	RG Fixed Point TNR	"Stand-alone" Symmetries	MPO Symmetries	Example of unstable variations (i.e., tensors that respect stand-alone symmetries but violate MPO symmetries)	Boson Condensed
Toric code	Single-line (section II C)	none			e-particle
	Double-line (section II E)				m-particle
	Triple-line (section IV H)				m-particle
Double-semion	Double-line (section III)	same as those in toric code above		same as that in toric code above, and also:	Z_2 gauge charge
	Triple-line (section IV H)	same as those in toric code above		same as that in toric code above, and also:	Z_2 gauge charge
Double-fibonacci	Triple-line (section V)				$\tau\bar{\tau}$-particle

TABLE I. A summary of different models, their different TNRs, stand-alone space of the TNR, physical (MPO) symmetries of the TNR, space of unstable variations obtained by numerical calculations and finally the boson of the model we observe to be condensed in the instability process. Z and X are the usual Pauli matrices. $\omega = i^s$ where $s = 0$ if the two legs on either side of it are the same, otherwise $s = 1$. δ denotes the "branching rule" of double-Fibonacci model applied on 3 legs. $\delta = 0$ when exactly 2 of the 3 legs are 0, otherwise its 1. Note that, though we have shown just one example of unstable variation in the fifth column for illustration, *any* variation that respects the stand-alone symmetries but violates the MPO symmetries causes instability. These labels and notations are explained in more details in the corresponding sections mentioned in the second column.

$S_{\text{topo}}(0)$ we call that direction stable meaning that topological order is still present and remains the same. This understanding of tensor instability is important not only for the identification of topological order for a particular model, but also for the numerical study of phase transitions between topological phases. In particular, if one is to use the tensor network approach to study phase transition due to boson condensation, then the corresponding variation direction must be *allowed* in order for the simulation to give correct results. For example, in Ref. 17, 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.

The main results of the paper are, (1) the tensor instability conjecture below, (2) the data from specific models supporting the conjecture as summarized in Table I, and (3) analytic proof that all general string-net TNR have instabilities as outlined in appendix B. The precise meaning of the various terms will be elaborated in subsequent sections.

Tensor Instability Conjecture: Given an RG fixed point Tensor Network Representation of a string-net model, there are two subspaces of tensor variations, (i)-M_0, the space spanned by tensor variations that do not "collapse" the tensor network. This subspace is called "stand-alone" space. (ii)- $\mathbb{M}$, a subspace of the stand-alone space, spanned by tensor variations that can be represented as variations on the physical indices. A tensor in this subspace respects the so called MPO symmetries[2]. A tensor variation is an unstable variation if and only if it has a component in the stand-alone space that does not respect the MPO symmetry, that is, it has a component in the subspace $M_0 - \mathbb{M}$.

We call it a conjecture and not a theorem because, though it is true for all models and all their TNRs that we analyzed, we do not have an analytical proof for all possible cases.

The paper is organized as follows. In section II, we start from the simplest string-net model – the toric code model[18], and study two types of tensor network representation of its ground state. The single line representation was studied in Ref. 1 and here we recover the result on the instability of the tensor with respect to certain Z_2 symmetry breaking variations. While reproducing the result, we introduce a new algorithm which allows us to investigate more complicated string-net models in later parts of this paper. Moreover, we explain in detail the physical mechanism behind such instabilities – the boson condensation induced by Z_2-symmetry breaking variations – which we generalize to all other string-net models in the upcoming sections.

The second representation we study for the toric code is the double line representation, as discussed in Ref.17. While the single line representation has only one virtual Z_2 symmetry, the double line representation has multiple of them. Are they equally important in protecting the encoded topological order? We find that only one of

the Z_2 symmetries is important as breaking it induces the condensation of bosons in the ground state while breaking other symmetries infinitesimally does not affect the topological order. We justify this conclusion by calculating explicitly, with a new algorithm, the topological entanglement entropy for the double line representation with different variations.

To generalize our study to generic string-net models, we study next the double semion model in section III. The double semion topological order is a Z_2 gauge theory, similar to toric code, and we find that a virtual Z_2-symmetry is again responsible for protecting the stability of the encoded topological order although the symmetry operator takes a very different form compared to the one in the toric code model: It is no longer a tensor product of operators on different virtual indices; instead it can only be expressed in terms of an MPO[2].

The form of the symmetry operator motivates the general analysis of string-net models in terms of the MPO symmetry in section IV where we show that the tensor network representation of all string-net models have unstable directions of variation (proven in Appendix B) and we conjecture that a variation direction is unstable if and only if it can stand alone *and* violates the MPO symmetry of the tensor. We support this claim by studying a third string-net model – the doubled Fibonacci model – explicitly in section V. Through explicit numerical calculation, we show that our conjecture holds not only for abelian topological orders like the toric code and the double semion model, but for nonabelian ones like the doubled Fibonacci as well. 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[19]) and later generalized to twisted quantum doubles (twisted G-injectivity[14] and MPO-injectivity[2]) and general string-net models (MPO-injectivity[2]).

Finally, a summary of the results is given in section VI and open questions are discussed. Some details of our analysis are explained in the appendices, including a brief review of string-net models, their tensor network representation and their transformation under the application of string-operators, 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 condition in our calculation.

II. TORIC CODE

We start from the simplest illustrative example of nonchiral intrinsic topological order: the toric code[18]. We work on a hexagonal lattice and assign local degrees of freedom, i.e. 0-spin down- or 1-spin up, on the edges of the lattice. The Hamiltonian is a sum of local commuting

projectors, given as

$$H = -\sum_v A_v - \sum_p B_p$$
$$= -\sum_v \prod_{l \in v} Z_l - \sum_p \prod_{l \in p} X_l \qquad (2)$$

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. 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

$$|\Psi_{\text{gs}}\rangle = \sum_{X \in \text{closed}} |X\rangle \qquad (3)$$

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 $S_{\text{topo}} = \log 2$.

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

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

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

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

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 neighbouring a vertex, i.e., we satisfy the vertex condition. The plaquette condition is also satisfied since every configuration is of

(a)

(b)
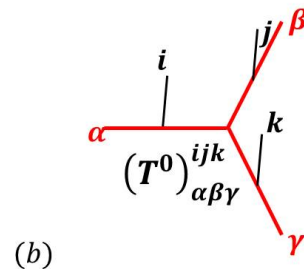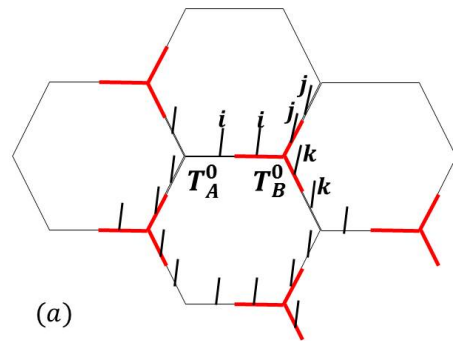
FIG. 1. (a) Single-line TNR of the toric code state. We make a copy of each qubit on every edge, $i \to ii$, and assign each to two respective sublattices, A and B. So local physical Hilbert space at each vertex is made of 3 qubits, corresponding to the three out-of-plane legs attached to the vertex. We assign a tensor, $(T^0)^{ijk}_{\alpha\beta\gamma}$ (shown in (b)), to each vertex. The out-of-plane legs, i, j, k, correspond to the 3 physical indices, while in-plane legs, α, β, γ, correspond to the 3 virtual indices. Contracting virtual indices on the shared edges give us the toric code state.

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 Chen *et al.*[1] that single-line TNR of the toric code state is not stable in certain directions of variation. Before we explain what these unstable directions of variation are, we first note that the single-line TNR explained above has a *virtual symmetry*. If an operation on the virtual indices leaves the tensor invariant, we will call it a svirtual symmetry of the tensor. Because the single-line tensor is non-zero only when virtual legs have even number of 1s, it has a natural $Z \otimes Z \otimes Z$ virtual symmetry (Fig. 2)(See Schuch *et al.*[19] for TNR virtual symmetries of the quantum double models). It
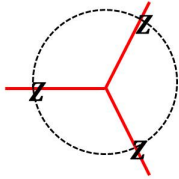
FIG. 2. Single-line RG fixed point tensor has a Z_2 symmetry on the virtual space. Since, for non-zero tensor values, only even number of virtual indices take value 1, the tensor is invariant under $Z \otimes Z \otimes Z$ on the virtual indices.
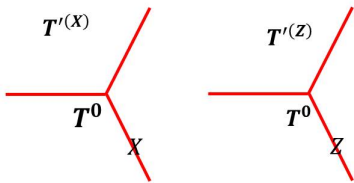
FIG. 3. Illustrative examples of two kinds of variations to the single line toric code fixed point tensor. The X variation breaks the $Z \otimes Z \otimes Z$ virtual symmetry, while the Z variation does not.

is a Z_2 symmetry with group elements $\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}$ and $Z \otimes Z \otimes Z$ acting on the virtual legs of the local tensor. Chen *et al.*[1] showed that topological order is stable with *any* Z_2 respecting variations and unstable with any Z_2 violating variation. To illustrate this, we can consider two different directions of variation in single-line TNR. We can add an X or Z variation on one of the virtual indices of the tensor (Fig. 3). The tensor is varied as $T^0 \rightarrow T^0 + \epsilon T$ with

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

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

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

We 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. Before we move on to other TNRs, we would like to explain the algorithm used here. Readers can skip this section if they are not interested in the details of the algorithm.

B. Algorithm for calculating S_{topo}

Here we explain the algorithm we use to calculate the topological entanglement entropy of any translation invariant tensor network state. We use the idea presented by Cirac *et al.*[20] to calculate reduced density matrix on a region and hence its entanglement entropy. We consider honeycomb lattice, though it can easily be extended to other lattices. By translation invariant we mean that all vertices on the sublattice A and sublattice B are attached with the same tensors, T_A and T_B, respectively. First we define certain notations for convenience of later discussion. The starting objects are given tensors T^I_α, where I and α denote the set of physical and virtual indices, respectively: $I = (i_1, i_2, ..), \alpha = (\alpha_1, \alpha_2, ..)$. The state represented by these tensors can be written as

$$|\Psi\rangle = \sum_{I_1, I_2, ..} \text{Tr}(T^{I_1} T^{I_2} \ldots) |I_1, I_2, ...\rangle. \tag{5}$$

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 the Fig. 4(a). We denote the tensors on the left and right boundaries as T_l and T_r.

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

$$\sigma_L = \mathbb{T}_l(\partial L)\mathbb{T}(L), \quad \sigma_R = \mathbb{T}(R)\mathbb{T}_r(\partial R). \tag{6}$$

Cirac *et al.*[20] showed that the physical reduced density matrix on one of these halves, let's say the left one, is related to the density operator on the virtual indices as,

$$\rho_L = U\sqrt{\sigma_L^T}\sigma_R\sqrt{\sigma_L^T}U^\dagger \tag{7}$$

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

$$\rho_L = \frac{\sigma_b^2}{\text{Tr}(\sigma_b^2)}. \tag{8}$$

It is known that the Rényi entropy with any Rényi index gives the same topological entanglement entropy[21]. So we calculate Rényi entropy with Rényi index $\frac{1}{2}$,

$$S_{1/2}(\rho_L) = \frac{1}{1-1/2} \log \mathrm{Tr}(\rho_L^{1/2})$$
$$= 2\log \mathrm{Tr}(\sigma_b) - \log \mathrm{Tr}(\sigma_b^2). \quad (9)$$

In the limit of large cylinder, it should behave like

$$S_{1/2}(\rho_L) = \alpha_0 |C| - S_{\mathrm{topo}} \quad (10)$$

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

Before we move on to the next step, we would like to mention an important subtlety regarding computation of S_{topo} on a cylinder. In Ref.22 and 23 it has been shown that S_{topo} calculated this way on a cylinder, in general, might depend on the boundary conditions. We choose a particular boundary condition for all our calculations and examine the dependence of S_{topo} on boundary condition in the appendix C. Our findings are consistent with the conclusion in Ref.23.

We first have to calculate $\mathbb{T}(R)\mathbb{T}_r(\partial R)$ for the above setup. The problem is, the computational complexity of exact tensor contraction grows exponentially with the size of R, so we need to use some approximate renormalization algorithm. We use an algorithm which is a slight modification of known tensor renormalization algorithms[6,17,24]. Consider double tensors contracted along a thin strip on the cylinder giving us a *transfer matrix operator*, $\mathbb{S}$. If R includes n of such strips, we have $\mathbb{T}(R) = \mathbb{S}^n$. Since the tensor network state under consideration are short range correlated along the cylinder, the spectrum of $\mathbb{S}$ is gapped. Consequently, for large n, only the highest eigenvalue and the corresponding eigenvector of $\mathbb{S}$ dominates. That is, in thermodynamic limit, $\mathbb{T}(R)$ only depends on the highest eigenvalue/eigenvector of the transfer matrix operator, $\mathbb{S}$. Moreover, we expect to approximate the eigenvector of highest eigenvalue with a *Matrix Product State (MPS)* with finite bond dimensions, since the tensor network state is short range correlated along the circumference of the cylinder. So we can start with a boundary MPS, apply the transfer matrix operator, and approximate the resulting state as an MPS with a fixed, finite bond dimensions. With each step, approximation to the eigenvector with highest eigenvalue improves and we do this recursively until we reach the fixed point giving us the desired eigenvector. Note that we require transfer matrix operators to be reflection symmetric for the condition $\sigma_L^T = \sigma_R = \sigma_b \Rightarrow \rho_L \propto \sigma_b^2$ to hold true.

The recursive algorithm is as following:

1. Initiate the boundary double tensor $\mathbb{T}_{A'} = \mathbb{T}_{r,A'}$ and $\mathbb{T}_{B'} = \mathbb{T}_{r,B'}$ (Fig. 4(b)).

2. Contract the bulk double tensors, $\mathbb{T}_A$ and $\mathbb{T}_B$ with each other giving a 4 leg tensor $\mathbb{T}_{AB}$. Contract $\mathbb{T}_{B'}$
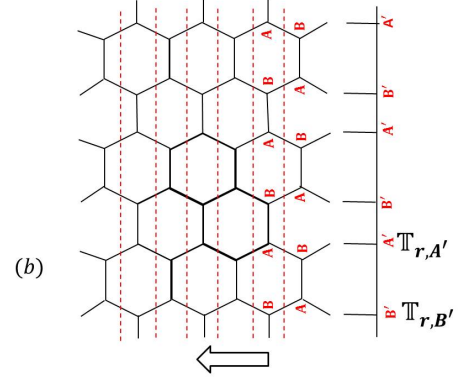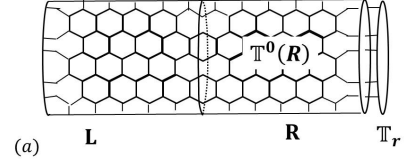
FIG. 4. (a) The honeycomb lattice is put on a cylinder with some boundary tensors, T_r. We calculate the topological entanglement entropy by calculating the entanglement entropy of the right half of the cylinder. (b) We contract the bulk tensors with the boundary ones layer by layer (from right to left) in a recursive way. In each recursion step, a layer between the two red dotted lines is contracted with the boundary tensors.

and $\mathbb{T}_{A'}$ with each other giving a 4 leg tensor $\mathbb{T}_{B'A'}$. Contract $\mathbb{T}_{AB}$ and $\mathbb{T}_{B'A'}$ with one another giving 4 leg tensor $\mathbb{T}_{AB'BA'}$ (first equality in Fig. 5).

3. Reshape the tensor $\mathbb{T}_{AB'BA'}$ into a matrix M where $M_{\alpha\beta',\beta\alpha'} = (\mathbb{T}_{AB'BA'})_{\alpha\beta'\beta\alpha'}[24]$. 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 $1, 2, \ldots, D_{\mathrm{cut}}$. $\mathbb{S}_{A'}$ and $\mathbb{S}_{B'}$ form an approximate decomposition of $\mathbb{T}_{AB'BA'}$,

$$\sum_{\gamma=1}^{D_{\mathrm{cut}}} (S_{A'})_{\alpha\beta'\gamma}(S_{B'}^\dagger)_{\gamma\beta\alpha'} \approx (\mathbb{T}_{AB'BA'})_{\alpha\beta'\beta\alpha'} \quad (11)$$

(approximating step in Fig. 5).

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

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

FIG. 5. Each recursion step is shown in detail. We first contract the two bulk tensors $\mathbb{T}_A$, $\mathbb{T}_B$ and two boundary tensors $\mathbb{T}_{A'}$, $\mathbb{T}_{B'}$ to produce $\mathbb{T}_{AB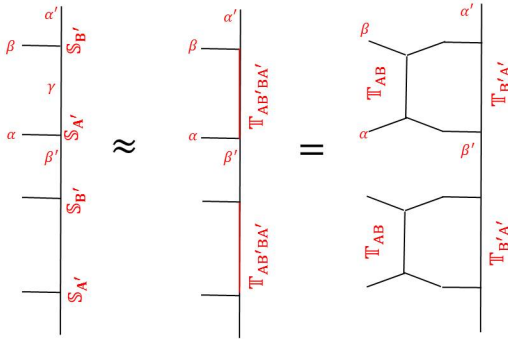'BA'}$. Then we use singular value decomposition to approximate $\mathbb{T}_{AB'BA'}$ as a contraction of two tensors $\mathbb{S}_{A'}$ and $\mathbb{S}_{B'}$.
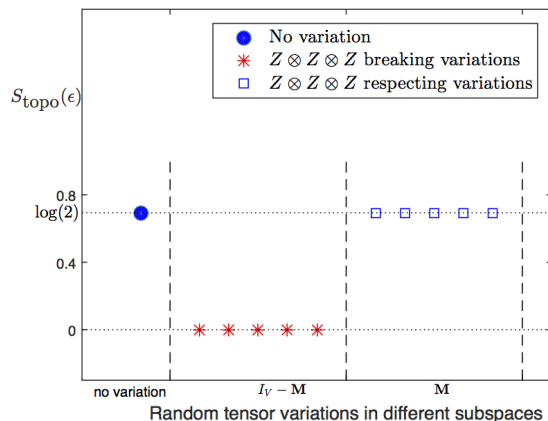

FIG. 6. Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by toric code fixed point single-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. $I_V = I^{\otimes 3}$ is the projector on to the full virtual space. $\mathbb{M} = \frac{1}{2}(I^\otimes + Z^{\otimes 3})$ is the projector on to the space of variations that respect the $Z^{\otimes 3}$ symmetries. So, $I_V - \mathbb{M}$ is a projector on to the space of variations that break $Z^{\otimes 3}$ symmetries. We see that variations in $I_V - \mathbb{M}$ subspace are unstable while variations in $\mathbb{M}$ are stable. Details of this numerical calculation are given in the appendix D 1

.

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

We use the algorithm described in the previous section 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 projector onto the full virtual space. $\mathbb{M} = \frac{1}{2}(I^\otimes + Z^{\otimes 3})$ is a pro-

jector on to the space of variations that respect the $Z^{\otimes 3}$ symmetries. So, $I_V - \mathbb{M}$ is a projector on to the space of variations that break $Z^{\otimes 3}$ symmetries. We first calculate S_{topo} in the state constructed by the fixed point tensor, T^0. Then we generate a random tensor T^r on the full space, project it on to the subspace $I_V - \mathbb{M}$, add it to the fixed point value, $T^0 \to T^0 + \epsilon(I_V - \mathbb{M})T^r$ and calculate $S_{\text{topo}}(\epsilon)$. Similarly, we generate a random tensor T^r on the full space, project it on to $Z^{\otimes 3}$ respecting subspace $\mathbb{M}$, add it to the fixed point value, $T^0 \to T^0 + \epsilon\mathbb{M}T^r$ and 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. 6.

We see that $Z^{\otimes 3}$ respecting variations lead to the same topological entanglement entropy as the fixed point state, while $Z^{\otimes 3}$ violating variations lead to zero topological entanglement entropy. This reproduces the result by Chen et al.[1].

D. Physical understanding of instability: symmetry breaking and boson condensation

How can we understand physically the fact that it is only $Z^{\otimes 3}$ symmetry breaking variations that drive the state to a trivial state ($S_{\text{topo}} = 0$)? For that we need to understand three separate but intricately connected concepts,
1- virtual symmetry and quasi-particle excitation,
2- zero-string operators
3- boson condensation
Before we move on to explain these concepts, we briefly recall the physical properties of the toric code model. It has 3 types of non-trivial quasi-particle excitations/anyons: e particle (electric charge), m particle (magnetic flux) and em particle (electric charge+magnetic flux). e and m are self-bosons, that is, braiding an e particle (m particle) with another e particle (m particle) doesn't produce any phase factor. But e and m have mutual semionic statistic with each other. That is, braiding e around m, or vice versa, produces a factor of -1. For the particular basis we have chosen in Eq. (2), a pair of e particles can be created by applying X operators on the ground state along a string. e particles appear at the end of this string. Similarly, a pair of m particles can be created by applying Z operators along a string path on the dual lattice. m particles appear on the end of this string.

1. Virtual symmetries and quasi-particle excitations

Let's first understand the relation between virtual symmetries of the tensor and quasi-particle excitations of the model. A tensor can be seen as a linear map from the local virtual Hilbert space to the local physical Hilbert space. Associated to this map, the tensor has certain

symmetries. We can count the number of symmetries required by looking at the dimension of local virtual and physical space. The virtual space of single-line TNR is made of 3 qubits, hence it is an 8 dimensional space. The physical space however is 4 dimensional, corresponding to the 4 configuration the physical legs are allowed to take in the toric code ground state. These are $000, 011, 101$ and 110. So to project an 8 dimensional space onto a 4 dimensional space, we need a Z_2 symmetry. Indeed the tensor has a $Z \otimes Z \otimes Z$ symmetry, where Z operators are applied on the 3 virtual indices. But is this symmetry of any physical significance? We analyze it now.

For a translational invariant state, the information of the many body state is encoded in a single tensor (or, two tensors on the two sublattices, T_A and T_B. But here we consider T_A and T_B to have the same components). Since the fixed point tensor corresponds to the ground state of toric code Hamiltonian, they have to encode the information that there are no quasi-particle excitations. This information is encoded in form of a symmetry of the tensor as we show below.

Consider the following physical process. We generate a particle a, antiparticle $\bar{a}$ pair, move a around a vertex 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 vertex. If there was another particle excitation b present at the vertex and if a and b have a non-trivial braiding statistics with each other, then this process produces a phase factor. Hence application of $W_a(C)$, where C is a loop around a vertex, can be used to detect if there is an excitation present at the vertex. It implies that $W_a(C)$ is a symmetry of the ground state.

$W_a(C)$ is an operator on the physical degrees of freedom, which induces an operator, $M_a(C)$, on the virtual degrees of freedom. Just as $W_a(C)$ is a symmetry on the physical level, $M_a(C)$ should be a symmetry on the virtual level. A tensor variation that breaks this symmetry would imply presence of an excitation. This is how some of the virtual symmetries of the ground state tensors are rooted in a physical requirement. At this point, one should note that the fixed point ground state tensor, such as given in Eq. (4), (14), have a lot of symmetries, but not all of them come from the same physical source. Some of them may just be a redundancy of the mathematical description of the tensor. For example, if a particular leg of the tensor is copied into two legs, the tensor obtains an additional symmetry. But clearly this symmetry cannot be of any physical consequence.

Let's illustrate the above discussion with the single-line TNR of toric code state. An m particle going around the vertex applies 3 Z operators on the 3 physical indices. But single-line tensor, T^0 in Eq. (4), has the property that applying 3 Z operators on the 3 physical indices is equal to applying 3 Z operators on the 3 virtual indices (Fig. 7(a)). So we have, $M_m(C) \equiv Z^{\otimes 3}$. This is precisely the physical origin of the $Z^{\otimes 3}$ symmetry of the single-line TNR. So, a tensor variation that breaks this symmetry
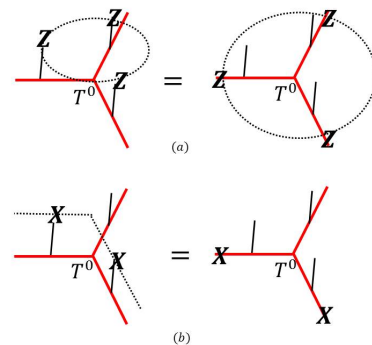
FIG. 7. Operators on virtual space induced by operators on physical space. (a) Applying a loop of Z operators on physical indices of T^0 in Eq. (4) is equal to applying a loop of Z operators on virtual indices. (b) Applying two X operators on two physical indices of T^0 is equal to applying two X operators on the corresponding virtual indices
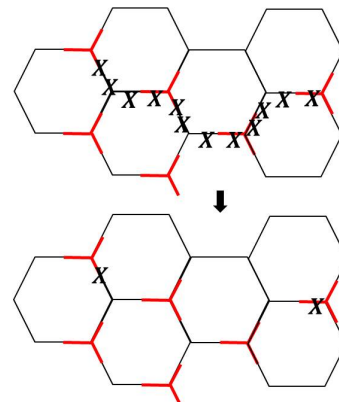
FIG. 8. The X-string operator, that creates a pair of e-particles, 'disappears' along the path on the single-line TNR. That is, the X-string operator when represented on the virtual indices, does not change the tensors along the path. We call such a string operator 'zero-string operator' of that TNR.

corresponds to the presence of an e particle.

2. Zero-string operators and bosons

Now a natural question arises: why isn't there an analogous symmetry constraint on the tensor corresponding to an e-string operator Wilson loop? To understand this, let us see how a string of X operators act on the fixed point tensors. As the X-string passes through a vertex, it applies X operator on two physical indices. But again, for T^0, this operation on physical level is equal to applying two X operators on the corresponding virtual indices (Fig. 7(b)). So, as the string passes through tensors along its path, it applies X operators to two of the three virtual indices of the tensor. But, when the virtual indices of two tensors are contracted, the two X operators coming from each tensors simply cancel each other.

It means, an X string simply disappears along the path on a contracted tensor network of single-line tensor (see Fig. 8). We call such string operators *zero-string operators*. It is a TNR-specific property that some of the string operators of the model might be zero-string operators on its virtual legs. Different TNRs of the same topological state can have different string operators as zero-string operators. Because X-string operator simply disappears on the single-line TNR along the path, a X-string operator Wilson loop completely disappears on this contracted tensor network, and it does not impose a virtual symmetry on the tensor.

We explained above how for single-line TNR, the e-string operator disappears along the path. We also know that e particles are self-bosons. Now we argue that this is not a coincidence; a string operator can be a zero-string operator for a given TNR only if it creates particles with trivial topological spins, i.e., bosons.

Let's say there is a TNR of a topologically ordered ground state and a string operator corresponding to particle a is a zero-string operator. Topological spin is calculated as the phase gained in the process shown in Fig. 9^{25}. We create two pairs of $a-\bar{a}$, where a is some anyon of the model. For convenience, to keep track of the anyons we have shown them in different colors, red and blue. Let's say the blue a and red a sit at sites 1 and 2 respectively. Now we apply the following procedure in this specific order: move red a from 2 to 3 (Fig. 9a), move blue a from 1 to 2 (Fig. 9b), move red a from 3 to 1 a (Fig. 9c). And finally annihilate blue a with red $\bar{a}$ and *vice-versa*. In general, the order in which each process is done is important. However, when the string operator is a zero-string operator, the order simply becomes irrelevant as one string operator does not see the presence of another string operator. So the whole process simply becomes creating and annihilating two pairs of $a-\bar{a}$ which has an amplitude 1, giving us a topological spin equal to 1. So a has to be a boson of the model.

3. Boson condensation in a TNR

We explained in the previous sections how a $Z^{\otimes 3}$ breaking variation corresponds to an e particle, and how the string-operator of e particle disappears along the path on the single-line TNR. Now we will see how these two facts combine to give an e particle condensate when a $Z^{\otimes 3}$ breaking variation is added to the fixed point tensor T^0.

If T^0 is the tensor corresponding to toric code ground state $|\Psi_0\rangle$, and T^X is an X variation, let's try to understand what state $|\Psi\rangle$ the tensor $T^0 + \epsilon T^X$, with ϵ
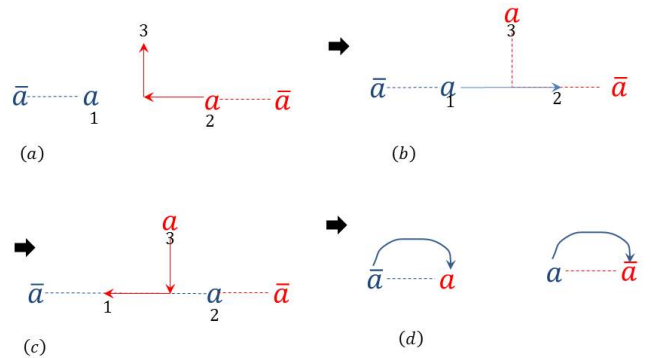
FIG. 9. Calculation of topological spin. We create two pairs (shown as red and blue) of particle, anti-particle pairs $a - \bar{a}$, with a situated at site 1 and 2. We apply the following procedure in this order: (a) Move first a (red) from 2 to 3, (b) move second a (blue) from 1 to 2, (c) move first a (red) from 3 to 1. Finally, (d) we annihilate each a with the anti-particles of the other anyon (i.e. red a with blue $\bar{a}$ and *vice versa*). When the propagation of a happens through a zero-string operator, this order of process becomes irrelevant, 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.

infinitesimally small, corresponds to.

$$|\Psi_0\rangle = \sum_{\{i_j\}} (T^0)^{i_1} (T^0)^{i_2} \ldots (T^0)^{i_n} |i_1 i_2 \ldots i_n\rangle,$$

$$|\Psi\rangle = \sum_{\{i_j\}} (T^0 + \epsilon T^X)^{i_1} (T^0 + \epsilon T^X)^{i_2} \ldots \times$$
$$(T^0 + \epsilon T^X)^{i_n} |i_1 i_2 \ldots i_n\rangle$$
$$= |\Psi_0\rangle + \epsilon^2 \sum_{v_1, v_2} |\Psi_{v_1, v_2}\rangle + \ldots \qquad (12)$$

where $\ldots$ denotes the higher order terms, and $|\Psi_{v_1, v_2}\rangle$ denotes the wavefunction that we get by replacing T^0 with T^X *only* at vertices v_1 and v_2. Now the key point is, if we apply a X-string operator between sites v_1 and v_2, we get exactly $|\Psi_{v_1, v_2}\rangle$ because this operator doesn't change tensors along the path and applies X on the virtual index of the tensors at both of its ends (Fig. 8). It means $|\Psi_{v_1, v_2}\rangle$ corresponds to a state with e particles sitting at v_1 and v_2. The weight of this state in the superposition remains the same, ϵ^2, irrespective of how far away v_1 and v_2 are from each other. It means the two e particles can appear at any length scale. Though ϵ^2 looks small compared to the weight of $|\Psi_0\rangle$, one has to bear in mind there are $\sim \binom{N}{2}$ such terms in the expansion, where N is the number of sites. So the contribution of these terms become exponentially larger than that of $|\Psi_0\rangle$ in the thermodynamic limit.

A similar argument can be applied to higher order terms, to show that they correspond to a state of e particles appearing wherever the variation term T^X appears. It means that $|\Psi\rangle$ is roughly a superposition of one e par-

ticle and no e particle at each site. This is nothing but an e particle condensate.

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

$$H = H_0 + \epsilon \sum_l X_l. \tag{13}$$

The ground state of this perturbed Hamiltonian is also a superposition of $|\Psi_0\rangle$ and terms like $|\Psi_{v_1,v_2}\rangle$. But the weight that appears with $|\Psi_{v_1,v_2}\rangle$ is of the order of $\epsilon^{\text{distance}(v_1,v_2)}$, that is, the separation between two e particles is exponentially suppressed. So, in thermodynamic limit, these excitations disappear. But this is not the case with state in Eq. (12). That is why the state in Eq. (12) cannot be produced by infinitesimal small local perturbation of the parent Hamiltonian.

So we see that the state in Eq. (12) is actually a 'condensate' of e particles and the transition $|\Psi_0\rangle \to |\Psi\rangle$ is a boson condensation transition. One should note that this is not a 'phase transition' in the traditional meaning of the word, as it is not brought about by physical perturbations to the system. It is merely a mathematical property of the TNR that even infinitesimal variations to the tensor might result in global changes in the state.

One should carefully note that analogous phenomena does not happen with Z variations to the tensor. Two Z variations sitting at v_1 and v_2 cannot be considered m particles at the ends of an invisible m-string operator simply because Z-string operator does not disappear along the path in single-line TNR (that is, it is not a zero-string operator of the single-line TNR). To create two m particles at v_1 and v_2, T^0 has to be changed along a path from v_1 and v_2 and, consequently, has a weight of the order of $\epsilon^{\text{distance}(v_1,v_2)}$. So the distance between two m particles is exponentially suppressed and, in the thermodynamic limit, m particles disappears altogether resulting in the same topological state as the original ground state. This is the reason why X variation results in instability while Z variation does not.

In the above discussion of boson condensation we used the exact eigenvector (with eigenvalue -1) of the $Z^{\otimes 3}$ symmetry, T^X. But what if we use a variation T', that has a finite component in T^X direction? It will simply result in a state that has a finite overlap with the state $|\Psi\rangle$ at all length scales. So it will again be an e particle condensate. Any variation T' has a finite component in T^X if and only if it violates $Z^{\otimes 3}$ virtual symmetry. So we reach the conclusion that any variation that breaks $Z^{\otimes 3}$ symmetry leads to a change in topological order due to e particle condensation.

Now we consider a different TNR of the toric code state that has the exact opposite properties to single-line TNR:

(a)

(b)
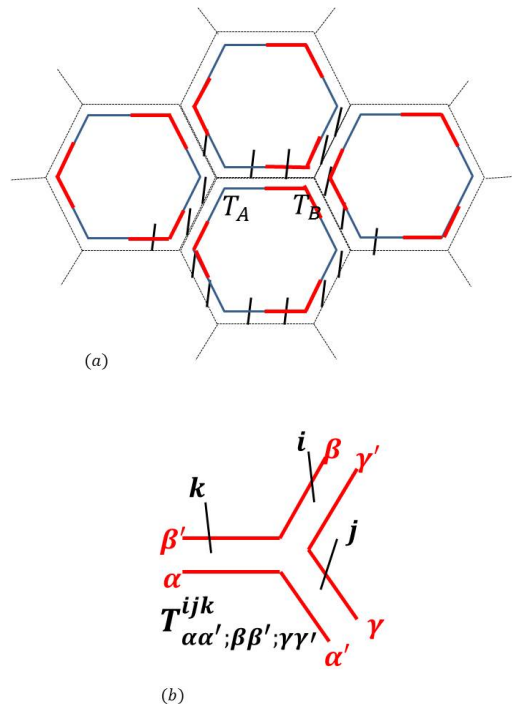
FIG. 10. Double-line TNR of the toric code state. We again split the qubit on each edge into two, and assign each to the two nearby vertices. The local physical Hilbert space consists of 3 qubits. We associate to each vertex a tensor $T^{i,j,k}_{\alpha,\alpha';\beta,\beta';\gamma,\gamma'}$ (shown in (b)), where out of plane legs, i, j, k, correspond to the 3 physical indices, and in-plane legs $\alpha, \alpha', \beta, \beta', \gamma, \gamma'$ are the virtual indices. Virtual indices of the tensors contract along the shared edges to produce the toric code state on the physical indices.

Z variations result in instabilities while X variation does not.

E. Double-line TNR of the toric code state

In the double-line TNR of the toric code state, we associate with each vertex a tensor with 3 physical legs and 6 virtual legs, $T^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'}$, (see Fig. 10). We will refer to these virtual indices as 'plaquette indices' or 'plaquette legs' sometimes, because they carry the plaquette degree of freedom that comes from the local Hamiltonian term. All indices take values 0 and 1. We denote the TNR corresponding to the RG fixed point state as T^0. (We use the same notation for different fixed point tensors, but it should be clear from the context which fixed point tensor we are discussing.) First property of T^0 is that $(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'} \propto \delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{\gamma\gamma'}$, that is, indices on the same plaquette assume the same values. 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
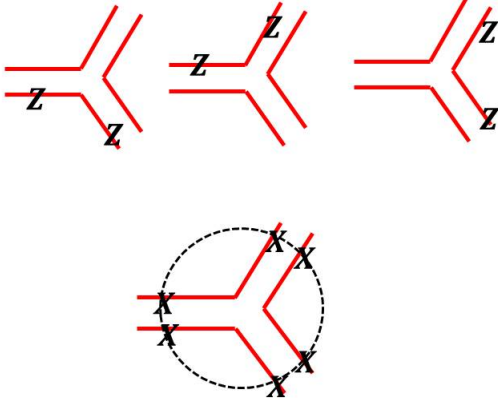
FIG. 11. Symmetries of the double-line TNR of toric code state. It has 4 inner/virtual Z_2 symmetries. Since the tensor is non-zero only for $s = s'$, $s = \alpha, \beta, \gamma$, it has 3 $Z \otimes Z$ symmetries. Also, since the physical indices are the domain wall values of the virtual indices (that is, $i = \beta + \gamma \bmod 2$ etc), the tensor remains the same when all virtual indices are flipped. That is, it has an additional $X^{\otimes 6}$ symmetry.

same values (both either 00 or 11) then the physical index in the middle has value 0, otherwise it is 1. That is, $i = \beta + \gamma, j = \gamma + \alpha, k = \alpha + \beta$ (all additions are modulo 2). So we can write T^0 as

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

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

We can write all non-zero components explicitly,

$$T^{000}_{00;00;00} = T^{000}_{11;11;11} = 1, \quad T^{011}_{00;11;11} = T^{011}_{11;00;00} = 1,$$
$$T^{101}_{11;00;11} = T^{101}_{00;11;00} = 1, \quad T^{110}_{11;11;00} = T^{110}_{00;00;11} = 1.$$
$$\tag{14}$$

Let's look at the symmetries of the double-line TNR (Fig. 11). 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 symmetry as shown in Fig. 11. First it has a X^6 symmetry. That is, if we flip all the six virtual indices, the tensor remains the same. Second, it has 3 $Z \otimes Z$ symmetry, where $Z \otimes Z$ are applied to the two virtual indices on the same plaquette. Single-line TNR had only one such Z_2 symmetry and it turned out that breaking it results in phase transition. For double-line we have four Z_2 symmetries. So the question is, are all of them important? That is, is it the case that breaking any of them with a variation leads to instability? Indeed many different possible kinds of variations are possible (Fig.
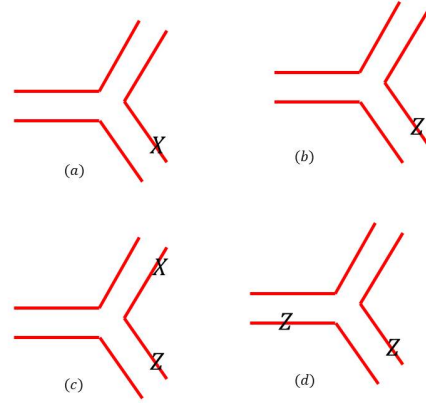
FIG. 12. Illustrative examples of different kinds of variations that can be added to double-line TNR. (a)An X variation violates $Z \otimes Z$ but not $X^{\otimes 6}$. (b) A Z variation violates $X^{\otimes 6}$ but not $Z \otimes Z$. (c) A ZX variation violates both and (d) a ZZ variation violates neither

12). A variation can violate $Z \otimes Z$ but not $X^{\otimes 6}$ (X variation), or it can violate both (ZX variation), or it can violate neither (ZZ variation), 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. 13)

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

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

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

To understand it physically, we first have to understand the notion of 'stand-alone' variations of a TNR.

1. Stand-alone subspace of a TNR

Double tensor $\mathbb{T}$ of a tensor T is defined as $\mathbb{T} = \sum_I T^I_\alpha (T^\dagger)^I_{\alpha'}$. It can be interpreted as a density matrix on the virtual space. Now consider the double tensor of a RG fixed point TNR, $\mathbb{T}^0$, contracted over some large region R. Let's say we remove $\mathbb{T}^0$ from one site and replace it with some other double tensor, $\mathbb{T}$. What do we get? In particular, are there tensors $\mathbb{T}$ such that this replacement collapses the whole tensor network? By collapse, we mean that we simply get zero upon contraction. The answer turns out to be, yes. In fact, as we see later, *most* tensors $\mathbb{T}$ will collapse the fixed point tensor network upon replacement. It turns out that only the tensors supported on a particular subspace of the full virtual space can replace the fix point tensor without collapsing the whole tensor network. We will call this space *the stand-alone subspace of the TNR*. Now we will give
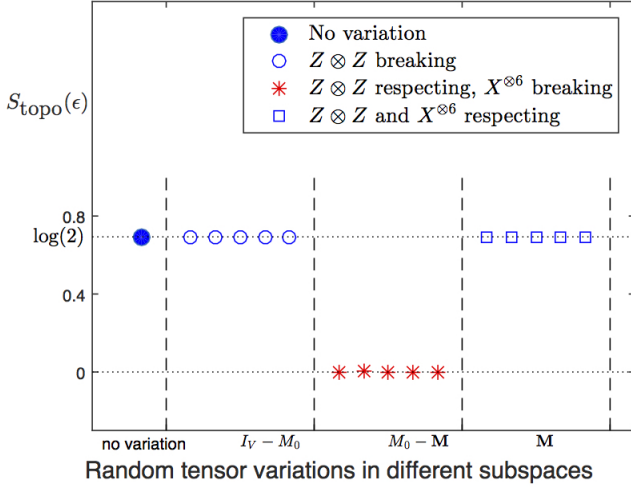
FIG. 13. Numerical calculation of topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of the states represented by toric code fixed point double-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Variations in $I_V - M_0$ violate $Z \otimes Z$ symmetry. Variations in $M_0 - \mathbb{M}$ violate $X^{\otimes 6}$ but not $Z \otimes Z$. Variations in $\mathbb{M}$ violate no virtual symmetry. Details of this numerical calculation are given in the appendix D 2.

a systematic way of calculating this subspace for a given fixed-point TNR.

Consider contracting the fixed-point double tensors $\mathbb{T}^0$ on a large disc with an open boundary. Now we remove the tensor at the origin. This tensor network will have dangling virtual indices at the origin and at the boundary of the disc. We want to find out the space of tensors that can be put on the origin without collapsing the tensor network. We do not care what tensor at the boundary we get. So we trace out the indices at the outer boundary (i.e. contract α and α' with each other). This leaves us with a tensor at the origin. The support space of this tensor will be precisely the stand-alone space. Any tensor supported on this subspace can stand alone with surrounding tensor being the fixed-point tensors.

Now let's calculate the stand-alone subspace of double-line TNR of toric code. The double tensor of T^0 in (14) can be written as (ignoring an overall normalization factor)

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

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. 11. We need to contract this tensor on a disc with a hole at the

origin. To contract two tensors given in an operator form, we need to multiply them and take a trace on the shared indices. A cumbersome but straight-forward calculation shows that double tensor contracted on a region R give (ignoring an overall normalization factor)

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

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

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

As explained in Fig. 16, X operators act on the two boundaries simultaneously, but Z operators act independently. Now to get the stand-alone space at the origin, we need to trace out the virtual legs at the boundary of D_{2m}. If we expand the expression for $\mathbb{T}(D_{2m} - D_6)$ above and apply trace on the operators on the outer boundary, only the terms with identity on the outer boundary survive. X operator does not have such a term, but Z does. So finally, tracing out the outer boundary leaves only $Z^{\otimes 2}$ on the inner boundary. That is, we get the following tensor on the 6 virtual indices incident on a singe vertex

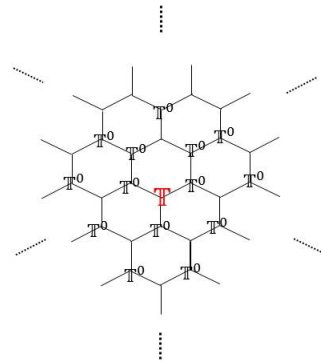$$B_0 = (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} \qquad (18)$$

$B_0^2 = 2B_0$, so

FIG. 14. Replacing a fixed point double tensor $\mathbb{T}^0$ in the tensor network with a tensor variation $\mathbb{T}$. A varied tensor network is a superposition of wavefunctions where fixed point tensors have been replaced by the variation. But for most variations, this tensor network collapses (i.e. becomes zero).
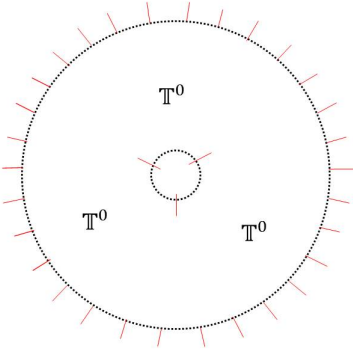
FIG. 15. Calculation of stand-alone space. We put the fixed point double tensor network on a large disc with a hole at the origin (one double tensor removed). This tensor network has dangling virtual indices (red legs) at the outer and inner boundaries. We trace out the virtual indices at the outer boundary, and the support space of the remaining tensor at the inner boundary gives us the stand-alone space.

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

Before we move back to explaining the instability of double-line TNR, let's calculate the stand-alone subspace of single-line TNR of toric code. One can calculate the double tensor of single-line TNR in Fig. 4 to be

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

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

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

Now contracting the outer circle gives us

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

So we see that, for single-line TNR, the stand-alone subspace is actually *all* of the virtual space. That is, there are no tensors that cannot stand alone. This was the reason why we did not need to discuss this concept for single-line TNR.

Why is the stand-alone subspace important? The stand-alone subspace plays a crucial role in deciding which tensor variations are stable/unstable. To see this, let's perform a similar analysis as done in section II D 3 but with a general tensor variation T. We have a fixed-point tensor T^0 that gives us the topological ground state $|\psi_0\rangle$. Now we vary this tensor, $T^0 \to T^0 + \epsilon T$. We want
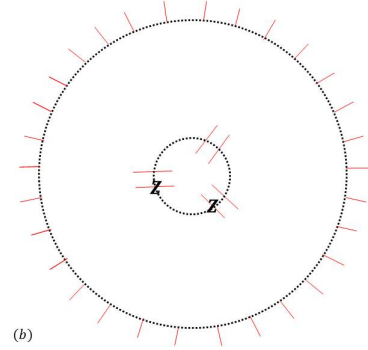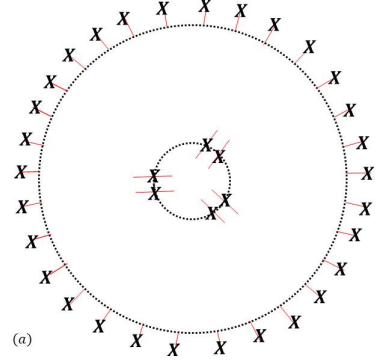
FIG. 16. X and Z operators appear differently in the toric code double-line double tensor contracted on a disc with a hole. X operators on inner boundary only appears with X operators on the outer boundary, which vanishes upon taking the trace. But Z operators on the inner boundary appear with identity on the outer boundary. So these terms survive the trace. That is why $Z^{\otimes 2}$ symmetry is imposed on the stand-alone space but not the $X^{\otimes 6}$ symmetry.

to know what the new state $|\psi\rangle$ looks like,

$$|\psi\rangle = \sum_{\{i_j\}} (T^0 + \epsilon T)^{i_1} (T^0 + \epsilon T)^{i_2} \dots |\{i_j\}\rangle$$

$$|\psi\rangle = |\psi_0\rangle + \epsilon^2 \sum_{v_1, v_2} |\psi_{v_1, v_2}\rangle + \dots \tag{23}$$

where $|\psi_{v_1, v_2}\rangle$ denotes the state we get by replacing T^0 with T at vertices v_1 and v_2. Higher order terms can be looked at in a similar way. Now the crucial point to note is, if v_1 and v_2 are away from each other, and if T does not belong to the stand-alone space, then $|\psi\rangle_{v_1, v_2} = 0$. In fact the only terms in the perturbative expansion above that can possibly survive are the ones where T appear next to each other. But the weight of these wavefunctions in the superposition will decay exponentially with the distance between v_1 and v_2, that is, as $\epsilon^{\text{distance}(v_1, v_2)}$. So such variations do not proliferate in the tensor network state. Since the variations cannot sit far away from each other, it can at most describe particles sitting close to each other. Such a state flows back to the original state under RG coarse-graining, which means that such a variation should not change the phase of the state.
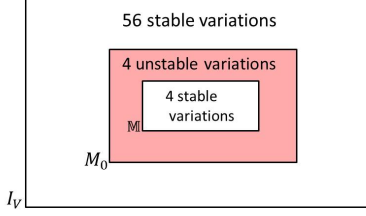
FIG. 17. Classification of the space of all variations to the toric code double-line tensor in terms of different subspaces. I_V is the full virtual space, M_0 is a virtual subspace projected by $M_0 = \frac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})$, and $\mathbb{M}$ is a subspace of M_0 projected by $\mathbb{M} = \frac{1}{2}(I^{\otimes 6} + X^{\otimes 6})M_0$. Since these are vector spaces, by 'N stable/unstable variations' in a subspace, we really mean there are N such linearly independent variations.

However, if the variation T does belong to the stand-alone space, then wavefunctions $|\psi_{v_1,v_2}\rangle$ do not vanish, and contribute to the quantum state with a finite weight. Or, in other words, the tensor network state is, roughly speaking, a superposition of wavefunctions where any T^0 can be replaced with T with a weight cost of ϵ. However this alone is not enough to cause a phase transition. But when the variation T corresponds to a boson, the state corresponds to bosons proliferation and it results in boson-condensation, causing the state to go through a phase transition.

2. Instability of double-line TNR

Now we are ready to explain the numerical results shown in Fig. 13. Let's first fix the notation that we will use throughout the paper. I_V denotes the projector on to the full virtual space. M_0 denotes the projector on to the stand-alone subspace. $\mathbb{M}$ denotes the projector on to the *MPO subspace*, which we will fully define in section IV B. For now, we understand it as the subspace of variations that respect both $Z \otimes Z$ and $X^{\otimes 6}$ symmetries. From their definitions, it should be clear that $\mathbb{M} \subseteq M_0 \subseteq I_V$ (See Fig. 17. For the double-line TNR of the toric code state, dimensions of these spaces can be easily seen to be $dim(I_V) = 2^6 = 64$, $dim(M_0) = 8$, $dim(\mathbb{M}) = 4$. Now we will try to understand the behavior of variations in each subspaces one by one.

Subspace $I_V - M_0$: We saw in section II E 1 that double-line stand-alone space, M_0, corresponds to tensors that respect the three $Z \otimes Z$ symmetries. So variations in $I_V - M_0$ are the ones that break $Z \otimes Z$ (regardless of whether they break $X^{\otimes 6}$ or not). Any variation in this subspace cannot stand alone. So they cannot proliferate and, consequently, they cannot cause a condensation. This explains the stability of variations in $I_V - M_0$ subspace in Fig. 13. Since $dim(I_V - M_0) = 56$, it means that out of 64 possible direction of variations, 56 are sta-
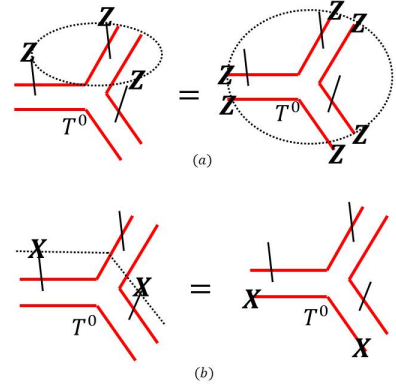
FIG. 18. (a) For T^0 in Eq. (14), applying Z operators on each physical index (an m particle Wilson loop) is equal to applying Z operators on each virtual index. (b) As a X-string passes through the tensor, it applies X operations on two physical indices, which for T^0, is equal to applying X operators on the plquette legs
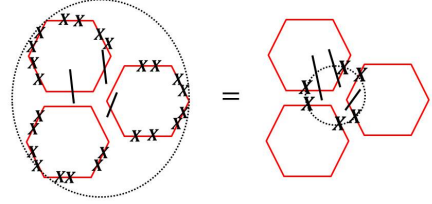
FIG. 19. An e particle Wilson loop encircling 3 plaquettes applies X operations on the virtual indices of the nearby tensors, as suggested by Fig. 18(b). But, since these indices are contracted with the tensor at the center, it is equal to applying X operation on all 6 virtual indices of this tensor and not changing the nearby tensors. So $X^{\otimes 6}$ is essentially an e particle Wilson loop.

ble simply because they cannot stand alone.

Subspace $M_0 - \mathbb{M}$: Variations in subspace $M_0 - \mathbb{M}$ are the ones that respect $Z \otimes Z$ but break $X^{\otimes 6}$ symmetry. Let's look at the sources of the symmetries of double-line TNR. Similar to single-line case, we consider the string-operators on the physical indices and see how they transfer to the virtual indices. An m particle going around a vertex (Z-string Wilson loop) induces a virtual $Z^{\otimes 6}$ symmetry (see Fig. 18(a)). And an e particle string operator (a X-string) passing through a tensor induces two X operator on the virtual indices (Fig. 18(b)). So, as explained in Fig. 19, the $X^{\otimes 6}$ symmetry can be understood as a virtual symmetry induced by an e particle Wilson loop that encircles the 3 plaquettes adjacent to the vertex. A variation breaking $Z^{\otimes 6}$ symmetry corresponds to the presence of an e particle and a variation breaking $X^{\otimes 6}$ symmetry corresponds to the presence of an m particle. But the key difference between the two is, $Z^{\otimes 6}$ symmetry *cannot* be broken within the stand-alone

subspace, and $X^{\otimes 6}$ *can* be. For example, a Z variation (Fig. 12b) is within the stand-alone space but breaks the $X^{\otimes 6}$ symmetry. Such variations correspond to an m particle that can proliferate, and that is why they lead to a phase transition. This explains the instability of random variations in $M_0 - \mathbb{M}$ in Fig. 22. $dim(M_0 - \mathbb{M}) = 4$. It means that out of 64 possible directions of variations, 4 are unstable.

Subspace $\mathbb{M}$: Finally we come to the variations in subspace $\mathbb{M}$. These variations break no virtual symmetries. So, at most, they might break a physical symmetry, which implies that all variations in this subspace are physical variations. But, since it is a gapped quantum phase, infinitesimal physical variations cannot lead to a phase transition, and that is why these variations are stable. ZZ variation in Fig. 12 is an example of such a variation. Though it looks like a variation on the virtual space, it actually corresponds a variation on the physical space: ZZ applied on two virtual indices is equal to a single Z applied on the physical index between them (which is basically an m-string operator applied between two plaquettes next to each other). For this reason we will sometime refer to $\mathbb{M}$ as the 'physical subspace' of the virtual space. Another way of looking at the stability of ZZ variation is that they correspond to two m particles next to each other; they cannot be separated and proliferated. So they disappear in the large scale limit. We conclude that out of 64 possible variations $dim(\mathbb{M}) = 4$ are stable since they are physical variations.

FIG. 20. The Z-string operator, that creates m-particles, dissappears along the path on double-line TNR of toric code. The reason is, it crosses each plaquette twice. The 6 virtual legs on a plaquette are contracted together, so the 6 indices are either all 1 or all 0. So two Z operators on them simply cancel each other.

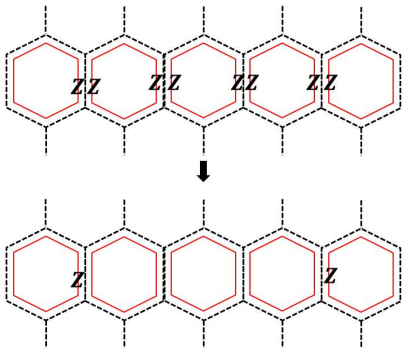An equivalent analysis can be done if we look at the string-operators of both quasi-particles. The string-operator of m-particle disappears along the path (see Fig. 20) in the stand-alone space, but the string-operator of e-particle does not. So the m-particle string operator is a zero-string operator of the double-line TNR. It means that two m-particles can be created at arbitrary distances from each other but not the e-particles, which is mathematically equivalent to saying that an m-particle can

exist alone but an e-particle cannot. So m-particles can condense but e-particles cannot. Hence the symmetry protecting against e-particles (the $Z^{\otimes 6}$ symmetries) can be broken but symmetries protection against m-particles (the $X^{\otimes 6}$ symmetry) cannot be broken (within the stand-alone space) without causing a condensation.

The classification of all possible variations can be summarized pictorially in Fig. 17.

F. Implications for the simulation of phase transitions

Projected Entangled Pair States (PEPS), one type of Tensor Network States (TNS), are often used as *ansatz* for different numerical simulations of gapped lattice topological models. In particular, TNS can be used to simulate phase transitions between different topological phases[17]. The fixed point Hamiltonian is perturbed with a local Hamiltonian $H_0 \rightarrow H_0 + \eta H_{\text{local}}$ and the perturbation strength, η is increased slowly. At some finite value of η the gap closes and the system goes through a phase transition. For many perturbations, this phase transition consists of boson condensation. For example, for the toric code Hamiltonian Eq. (2), two kinds of perturbations can be added

$$H_1 = -U \sum_v \prod_{l \in v} Z_l - g \sum_p \prod_{l \in p} X_l - \eta \sum_l Z_l, \quad (24)$$

$$H_2 = -U \sum_v \prod_{l \in v} Z_l - g \sum_p \prod_{l \in p} X_l - \eta \sum_l X_l. \quad (25)$$

Let's first discuss the first kind of perturbation. In the first Hamiltonian, we keep $U = \infty$ and study the ground state as the relative values of η and g change. At $\eta = 0$ the ground state is simply the fixed point toric code state given in Eq. (3). That is, it is an equal weight superposition of all closed string configuration. At $g = 0$, the state is the vacuum state, that is, all spins are 0. These two states are topologically different, 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 phase transition point, the system goes through a boson (m particle) condensation and the ground state becomes a trivial state. Boson condensation phase transitions are known to be second order phase transitions. That is, ground state energy and its first order derivative as a function of η/g are smooth functions, but its second order derivative is discontinuous at the phase transition point.

It was shown by Gu *et al.*[17] that an attempt to numerically simulate this phase transition point with single-

line tensor network state ansatz gives a transition that is wrong both quantitatively and qualitatively. It gives a wrong critical point value of η/g, and it gives a first order phase transition, not a second order one. But with double-line tensor network state ansatz, it gives the correct second order phase transition with correct critical point.

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

A similar analysis can be done for the second type of perturbation. We set $g = \infty$ and change relative value of U and η. For $\eta = 0$ the ground state is the toric code ground state in Eq. (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 single-line TNS ansatz and not the double-line TNS ansatz.

This is one of the important point 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)[17] and Tensor Network Renormalization (TNR[26])[27] 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.

III. DOUBLE SEMION

Double Semion model can be understood as a 'twisted' Z_2 quantum double model[16,28]. Its Hamiltonian is almost the same as that of toric code, except for the phase factor associated to the plaquette term

$$H_0 = -\sum_v \prod_{l \in v} Z_l - \sum_p \prod_{l \in p} X_l \prod_{r \in \text{legs of } p} i^{\frac{1-Z_r}{2}} \quad (26)$$
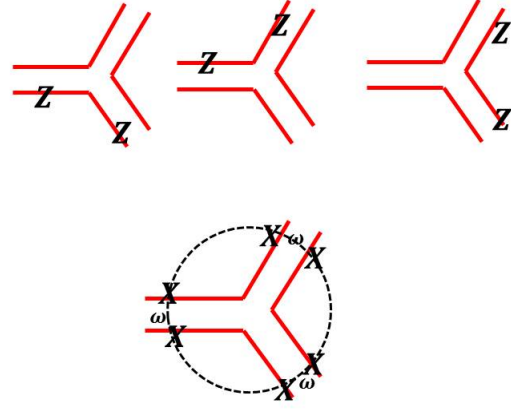
FIG. 21. Virtual symmetries of the double-semion double-line tensor. It has the same 3 Z_2 symmetries of the form $Z \otimes Z$ as that for the toric code double-line tensor. It also has a $X^{\otimes 6}$ symmetry but with an added phase factor of $i^{n(d)}$, where $n(d)$ is the number of domain walls between the inner indices (Eq. 29).

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

$$|\psi\rangle = \sum_{X \in \text{closed}} (-1)^{n(X)} |X\rangle \quad (27)$$

where X again refers to string configurations on the hexagonal lattice. $n(X)$ denotes the number of loops in a given string configuration. The ground state, like that of toric code, is again a superposition of all closed string configurations. But it has a phase factor of $(-1)^{n(X)}$ which is 1 for even number of loops and -1 for odd number of loops. It has 3 quasi-particle excitations, a semion, an anti-semion, and a self-boson. So, unlike the toric code, it has only one boson. There is a known double-line TNR of this state[8,9], $(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'}$, with the same structure as that of toric code. So, $(T^0)^{ijk}_{\alpha\alpha';\beta\beta';\gamma\gamma'} = S_{\alpha\beta\gamma}\delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{\gamma\gamma'}\delta_{i,\beta+\gamma}\delta_{j,\alpha+\gamma}\delta_{k,\alpha+\beta}$ But now the values are

$$S_{\alpha\beta\gamma} = \begin{cases} 1 & \text{if } \alpha + \beta + \gamma = 0, 3 \\ i & \text{if } \alpha + \beta + \gamma = 1 \\ -i & \text{if } \alpha + \beta + \gamma = 2. \end{cases} \quad (28)$$

Clearly, it has the same $Z^{\otimes 2}$ symmetry, but does not have the $X^{\otimes 6}$ symmetry as that of toric code double-line TNR. By looking at the tensor values, it can be seen that it has a 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,

$$n(d) = \begin{cases} 0 & \text{if } \alpha + \beta + \gamma = 0, 3 \\ 2 & \text{if } \alpha + \beta + \gamma = 1, 2. \end{cases} \quad (29)$$

We see that there is an additional phases factor with $X^{\otimes 6}$ symmetry. One can show that this symmetry is nothing
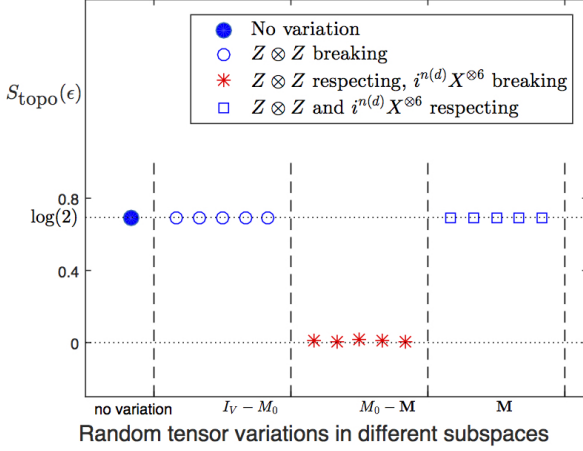
FIG. 22. Numerical calculation of the topological entanglement entropy $S_{\text{topo}}(\epsilon)$ of the states represented by double semion fixed point double-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.01$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Variations in $I_V - M_0$ violate $Z \otimes Z$ symmetry. Variations in $M_0 - \mathbb{M}$ violate $i^{n(d)} X^{\otimes 6}$ but not $Z \otimes Z$. Variations in $\mathbb{M}$ violate no virtual symmetry.. Details of this numerical calculation are given in the appendix D 2.

.

but a Wilson loop operator corresponding to semion (or anti-semion). A calculation similar to that of double-line toric code shows that the double-line double semion tensors also have the same stand-alone space: only the variations that respect the $Z \otimes Z$ symmetries are within the stand-alone space. It leads to the following prediction of stable/unstable variations:

1. Variations that violate $Z \otimes Z$ symmetries are stable.

2. Variations that respect $Z \otimes Z$ but break $(i)^{n(d)} X^{\otimes 6}$ are unstable.

3. Variations that respect both $Z \otimes Z$ and $(i)^{n(d)} X^{\otimes 6}$ are stable.

These predictions are confirmed by numerical calculation as shown in Fig. 22. To see this through string-operators of the model, we note that the string operator corresponding to the boson in the model is again a Z-string operator, and it again disappears along the path, as it was discussed for the toric code (Fig. 20). So it makes sense that bosons can stand alone and cause phase transition.

IV. GENERAL STRING-NET MODELS AND TENSOR INSTABILITY CONJECTURE

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 TNR discussed so far (single-line and double-line) are reduced versions of the a general *triple-line* TNR of the string-net states[8,9].

A string-net construction defines a topological model on a honeycomb lattice for any arbitrary *unitary tensor fusion category*[16,18]. The local Hilbert space has spins sitting on the edges. These spins can take $i = 0, 1, .., N - 1$ values called *string-types*. $i = 0$ corresponds to the vacuum state. 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 $P_{gs} = \left(\sum_p B_p\right)\left(\sum_v A_v\right)$ on the vacuum state. A brief review of the string-net models has been given in the appendix A. Readers can refer to the original papers for more details on the subject[2,8,9,16].

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

As shown by Gu *et al.*[8], Buerschaper *et al.*[9], 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 the appendix A 2. An interested reader may refer to the original papers[7–9] for more details.

A string-net fixed point state has a triple line TNR (Fig. 23), with components given by

$$(T^0)^{i_{23}i_{31}i_{12}}_{i'_{12}i'_{23}i'_{31};a_1a'_1;a_2a'_2;a_3a'_3} = S^{i_{23}i_{31}i_{12}}_{a_1a_2a_3}\delta_{i_{23}i'_{23}}\delta_{i'_{31}i_{31}}\delta_{i'_{12}i_{12}} \times$$
$$\delta_{a'_1a_1}\delta_{a'_2a_2}\delta_{a'_3a_3}$$
$$\text{where } S^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} = G^{i_{23}i_{31}i_{12}}_{a_1a_2a_3} \times$$
$$\sqrt[6]{d_{a_1}d_{a_2}d_{a_3}}\sqrt[4]{d_{i_{23}}d_{i_{31}}d_{i_{12}}}$$

(30)

where d_a etc are the quantum dimensions and G is the normalized F-symbol with full tetrahedral symmetry. We will refer to i_{12}, i_{23}, i_{31} as the 'physical indices/legs', $i'_{12}, i'_{23}, i'_{31}$ as the 'middle indices/legs' and $a_1, a'_1, a_2, a'_2, a_3, a'_3$ as the 'plaquette indices/legs' of the triple-line tensor. As is clear from Eq. 30, the fixed point
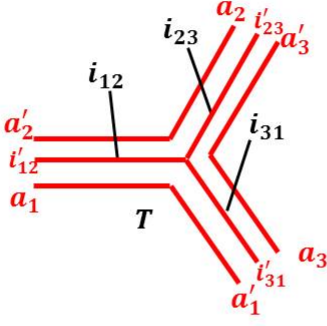
FIG. 23. Triple-line TNR of general string-net states. Black, out of plane legs carry the 3 physical indices, i_{12}, i_{23}, i_{31} corresponding to the 3 spins on the 3 edges adjacent to a vertex. In-plane legs carry the 9 virtual indices, $a_1, i'_{12}, a'_2, a_2, i'_{23}, a'_3, a_3, i_{31}, a'_1$. For RG fixed point, tensor takes non-zero values only when $i'_{12} = i_{12}, i'_{23} = i_{23}, i'_{31} = i_{31}$ and $a_1 = a'_1, a_2 = a'_2, a_3 = a'_3$.

tensor is non-zero only when the middle legs are equal to their respective physical legs, and plaquette legs on the same plaquette are equal to one another.

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

Here onward, statements about string-net models and their TNR should be understood as statements about all models and TNR that are within the string-net paradigm.
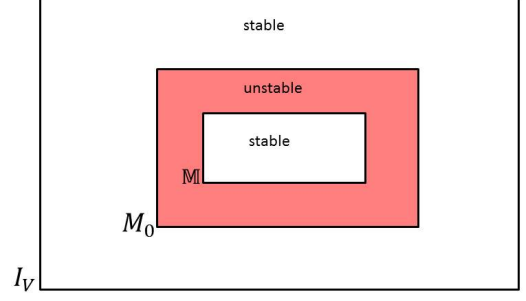
FIG. 24. Range of MPO projector $\mathbb{M}$ and stand-alone projector M_0 decompose the full virtual space into disjoint subspaces. I_V is the identity onto the full virtual space. White regions are stable, while the red region is unstable.

Now we will present the central conjecture of the paper.

B. MPO and Tensor instability conjecture

Let $(T^0)^I_\alpha : V \to P$ be the tensor network representation of an RG fixed point 2+1D topological ground state within the string-net type models. V is the virtual space and P is the local ground state physical space. α and I, respectively, denote the set of virtual and physical indices. We define the support space of a tensor T^I_α as the support space of the virtual density matrix $\sigma_{\alpha,\alpha'} = \sum_I T^I_\alpha (T^*)^I_{\alpha'}$, or simply, $\sigma = TT^\dagger$. Let MPO $\mathbb{M}$ be the projector on to the support space of T^0. Let $T^0 \to T^0 + \epsilon T^r$ be an infinitesimal variation to T^0. Let M_0 be the projector onto the subspace of stand-alone variations as described in section II E 1. Finally, let I_V be the identity operator on the full virtual space. We say a tensor T is supported by a projector P if $PT = T$. Note that if projector P_1 and P_2 project onto vector spaces V_1 and $V_2 \subseteq V_1$ respectively, then $P_1 - P_2$ is also a projector and it projects onto space $V_1 - V_2$. Now we present the central conjecture of our work.

Instability Conjecture:

1. $I_V \supseteq M_0 \supset \mathbb{M}$

2. All tensor variations in the projected space of $I_V - M_0$ are *stable*.

3. All tensor variations in the projected space of $M_0 - \mathbb{M}$ are *unstable*.

4. All tensor variations in the projected space of $\mathbb{M}$ are *stable*.

Or in simple words, *the only unstable variations are the stand-alone variations that do not respect the MPO symmetry*. A venn diagram of the decomposition of the virtual space through these projectors is shown in Fig. 24.

Note that conjecture 1 and conjecture 3 above together imply that *all* string-net TNR have unstable directions.

MPO subspace $\mathbb{M}$ can be interpreted in multiple but equivalent ways. It is defined as the support space of $TT^\dagger$ above. It can be understood as the virtual subspace which is isomorphic to the local physical space. It can also be understood as the space of variations that do not break *any* virtual symmetries. Finally, it can be defined as the space of variations that respect all virtual symmetries induced by Wilson loop operators within the stand-alone space. That is, in the general string-net formalism, the MPO projector can be written as,

$$\mathbb{M} \equiv \sum_f \frac{d_f}{D} M_f \qquad (31)$$

where the sum is over the 'simple' string-operators of the model, and d_f are their respective quantum dimensions. All other string-operators of the model can be obtained through products of simple string-operators. $M_f \equiv M_0 W_f M_0$, represents the virtual symmetry induced by a f-type Wilson loop operator , W_f, within the stand-alone space, M_0. Wilson loop W_f is applied covering the three adjacent plaquettes. In simple words, M_f is an inner symmetry of a single tensor which, when violated (and no other nearby tensors changed), results in a violation of Wilson loop on a larger space enclosing the given tensor. We include only the simple string-operators, since a symmetry constraints induced by other string-operators are simply a product of these ones. A string operator W_f is called a zero-string operator if $M_0 W_f M_0 = M_0$, that is, when it disappears trivially on the stand-alone space. The instability physically corresponds to the condensation of boson b of the model whose corresponding Wilson loop operator projection $M_b = M_0 W_b M_0$ is either a zero-string operator, or, as we will see in the case of Fibonacci model, has at least a finite component in M_0. That is, if $\text{Tr}(M_b M_0) \neq 0$ in the large loop limit, then the corresponding boson can condense.

Let's first understand the results from previous sections as an example of the above conjecture.

1- Single-line TNR of the toric code: Using Eq. (4), we have

$$\sigma = \sum_{i,j,k} (T^0)^{i,j,k} (T^{0;*})^{i,j,k}$$
$$= |000\rangle \langle 000| + |011\rangle \langle 011| + |101\rangle \langle 101| + |110\rangle \langle 110|$$
$$= \frac{1}{2} \left(I^{\otimes 3} + Z^{\otimes 3} \right) = \mathbb{M}. \qquad (32)$$

So virtual density matrix happens to be the MPO projector $\mathbb{M}$ itself. As shown in section II E 1, for single-line TNR, stand-alone space is actually the full virtual space. So $M_0 = I^{\otimes 3} = I_V$. Tensor variations that are supported on $\mathbb{M}$ are the ones that respect $Z^{\otimes 3}$ symmetry. Similarly the projector onto the unstable space is

$$P_U = M_0 - \mathbb{M} = \frac{1}{2} \left(I^{\otimes 3} - Z^{\otimes 3} \right). \qquad (33)$$

Now let's derive the same using Wilson loop operators.

$$\mathbb{M} = \frac{1}{4} M_0 (I + W_e + W_m + W_{em}) M_0 \qquad (34)$$
$$= \frac{1}{4} (I^{\otimes 3} + M_e + M_m + M_{em})$$
$$= \frac{1}{2} (I^{\otimes 3} + Z^{\otimes 3}) \qquad (35)$$

where $W_{m(e)}$ is Wilson loop operator of $m(e)$ particle. As shown in section II D 2, $M_m = Z^{\otimes 3}$, $M_e = I^{\otimes 3}$. And $M_{em} = M_e M_m$.

2- Double-line TNR of the toric code: Using Eq. (14), we calculate the virtual density matrix as

$$\sigma = \sum_{i,j,k} (T^0)^{i,j,k} (T^{0;*})^{i,j,k}$$
$$= \frac{1}{8} (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} (I^{\otimes 6} + X^{\otimes 6}) \qquad (36)$$

where the operators are acting as shown in Fig. (11). Since $\sigma^2 = 2\sigma$, projector on to the support of σ is $\mathbb{M} = \frac{1}{2}\sigma$ while, as calculated in section II E 1, projector on to the stand-alone space is $M_0 = \frac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}$. So the projector on to the unstable space is

$$P_U = M_0 - \mathbb{M} = \frac{1}{16}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes 6} - X^{\otimes 6}). (37)$$

So the only unstable variations are the ones that respect the $Z^{\otimes 2}$ symmetries but break the $X^{\otimes 6}$ symmetries. We can reproduce this using Wilson loop operators. For the double-line toric code we have $W_m = Z^{\otimes 6}$ and $W_e = X^{\otimes 6}$. $M_0 W_m M_0 = M_0$ while $M_0 W_e M_0 = M_0 W_e$. Hence W_m is a zero-string operator of double-line TNR. Putting these values in Eq. (31) we get

$$\mathbb{M} = \frac{1}{4} M_0 (I + W_e + W_m + W_{em}) M_0 \qquad (38)$$
$$= \frac{1}{4} (M_0 + M_0 W_e + M_0 + M_0 W_e)$$
$$= \frac{1}{16} (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} (I^{\otimes 6} + X^{\otimes 6}) \qquad (39)$$

which is the same as $\frac{1}{2}\sigma$.

3- Double-line TNR of the double semion model: Similarly, using (28) we calculate the projector on to the support space of the virtual density matrix for the double-line TNR of the double semion model:

$$\sigma = \sum_{i,j,k} (T^0)^{i,j,k} (T^{0;*})^{i,j,k}$$
$$= \frac{1}{8} (I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3} (I^{\otimes 6} + i^{n(d)} X^{\otimes 6}) \qquad (40)$$

where $n(d)$ is defined in Eq. (29). Again, $\mathbb{M} = \frac{1}{2}\sigma$. It has the same stand-alone space as double-line toric code, $M_0 = \frac{1}{8}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}$. So the projector on to the unstable space is

$$P_U = M_0 - \mathbb{M} \qquad (41)$$
$$= \frac{1}{16}(I^{\otimes 2} + Z^{\otimes 2})^{\otimes 3}(I^{\otimes 6} - i^{n(d)} X^{\otimes 6}). \qquad (42)$$

Using Wilson loop operators,

$$\mathbb{M} = \frac{1}{4} M_0 (W_0 + W_s + W_{\bar{s}} + W_{s\bar{s}}) M_0$$

$$= \frac{1}{4} (M_0 + M_s + M_{\bar{s}} + M_s M_{\bar{s}}) \tag{43}$$

where s and $\bar{s}$ denote the semion and anti-semion quasi-particles. $s\bar{s}$ is the boson. For double-line TNR, we saw that $s\bar{s}$-string operator is the zero-string operator, $M_{s\bar{s}} = M_0 W_{s\bar{s}} M_0 = M_0$, and $M_s = M_0 W_s M_0 = M_0 i^{n(d)} X^{\otimes 6}$. Note that $M_{\bar{s}} = M_{s\bar{s}} M_s = M_s$. So we get,

$$\mathbb{M} = \frac{1}{4} (2M_0 + 2M_0 i^{n(d)} X^{\otimes 6})$$

$$= \frac{1}{2} M_0 (I^{\otimes 6} + i^{n(d)} X^{\otimes 6}), \tag{44}$$

which is the same as $\frac{1}{2}\sigma$.

To apply the conjecture to the triple-line TNR of general string-net model, we now first calculate its MPO and stand-alone subspaces.

C. String-net MPO projector

Using the triple-line TNR T^0 of string-net states given in Eq. (30), the virtual density matrix is found to be

$$\sigma = \sum_I (T^0)_\alpha^I (T^{0;*})_\alpha^I$$

$$= \sum_{\{a_k, b_k; i_{k,k+1}\}} G_{a_1 a_2 a_3}^{i_{23} i_{31} i_{12}} G_{b_1 b_2 b_3}^{i_{23} i_{31} i_{12}} \prod_j (d_{a_j}^{\frac{1}{6}} d_{b_j}^{\frac{1}{6}} d_{i_{j,j+1}}^{\frac{1}{2}})$$

$$|\{a_k; i_{k,k+1}\}\rangle \langle \{b_k; i_{k,k+1}\}| . \tag{45}$$

Clearly, this density matrix can simply be written as

$$\sigma = \sum_{i,j,k} |v_{i,j,k}\rangle \langle v_{i,j,k}| \tag{46}$$

where $|v_{i,j,k}\rangle = (d_i d_j d_k)^{\frac{1}{4}} \sum_{a_1, a_2, a_3} G_{a_1, a_2, a_3}^{i,j,k} (d_{a_1} d_{a_2} d_{a_3})^{\frac{1}{6}}$

$$|a_1, a_2, a_3; i, j, k\rangle \tag{47}$$

So σ has a diagonal form in terms of vectors $v_{i,j,k}$. To get the projector on to its support space, we simply need to use the unit vectos $\frac{1}{N_{i,j,k}} |v_{i,j,k}\rangle$, 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

$$\mathbb{M} = \sum_{i,j,k} \frac{1}{N_{i,j,k}^2} |v_{i,j,k}\rangle \langle v_{i,j,k}| . \tag{48}$$

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

$$dim(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k}. \tag{49}$$

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

To calculate the stand-alone space, we need to contract the triple-line double tensor on an infinite disc with a hole at the origin. To do that, let's first rewrite the double tensor $\mathbb{T} = \sigma$ in another form. Using the pentagon identity in Eq. (I.1), $G_{a_1 a_2 a_3}^{i_{23} i_{31} i_{12}} G_{b_1 b_2 b_3}^{i_{23} i_{31} i_{12}} = \sum_f d_f \prod_j (G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}})$ in (45), we can rewrite $\mathbb{T}$ as

$$\mathbb{T} = \sum_{f=0}^{N-1} d_f B_f \tag{50}$$

where

$$B_f = \sum_{\{a_k, b_k; i_{k,k+1}\}} \prod_{j=1}^{3} (G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}} d_{a_j}^{\frac{1}{6}} d_{b_j}^{\frac{1}{6}} d_{i_{j,j+1}}^{\frac{1}{2}})$$

$$|\{a_k; i_{k,k+1}\}\rangle \langle \{b_k; i_{k,k+1}\}| . \tag{51}$$

B_f are loop operators on the dual lattice of honeycomb, the triangular lattice. The above expression is the boundary operator on a single triangle (Fig. 38). A general expression on the boundary ∂R of a region R can be written

$$B_f(\partial R) = \sum_{\{a_k, b_k; i_{k,k+1}\}} \prod_{j=1}^{n} (G_{a_{j+1} a_j f}^{b_j b_{j+1} i_{j,j+1}} d_{a_j}^{\frac{\theta_j}{2\pi}} d_{b_j}^{\frac{\theta_j}{2\pi}} d_{i_{j,j+1}}^{\frac{1}{2}})$$

$$|\{a_k; i_{k,k+1}\}\rangle \langle \{b_k; i_{k,k+1}\}| \tag{52}$$

where $j = 1, 2, \ldots, n$ label the vertices on the triangular lattice and (j, k) label the edges between vertices j and k and θ_j is the angle that the boundary ∂R makes on the vertex j. We will call these operators *f-type boundary operators* (Fig. 39). We will denote this boundary operator on the boundary ∂R of a region R as $B_f(\partial R)$. So to contract double-tensors on a region all we need to know is how the boundary operators contract with each other. The detailed calculation is done in the appendix B 1. We will mention the main results here. B_f operators have a very nice contraction property with each other, and they follow two main rules

1. Different f-type boundary operators do not contract with each other. That is, $\text{Ev}(B_f B_{f'}) = 0$ if $f \neq f'$. Where Ev denotes the operation of contracting two tensors along shared indices.

2. $\text{Ev}(B_f(\partial R_1) B_f(\partial R_2)) = D^V d_f^{E-V} B_f(\partial(R_1 + R_2))$, where D is the total quantum dimension and E and V are number of contracted edges and vertices, respectively.

It follows from these two observations, that the double tensor in Eq. (50) contracted on a region R produces

$$\mathbb{T}(R) = D^V \sum_f d_f^{V-E+F} B_f(\partial R) \tag{53}$$

where V, E, F are the number of contracted vertices, edges, and faces respectively. D^V is just an overall factor

so we will drop it from now on. A disc with a hole at the origin has two disconnected boundaries, one on the outer edge, other on the origin. $V - E + F = 0$ for such a region, so we get

$$\mathbb{T} = \sum_f (B_f)_{\text{inner}} \otimes (B_f)_{\text{outer}}. \tag{54}$$

Now we trace out the indices on the outer boundary. Here we use the property $\lim_{|\partial R| \to \infty} Tr(B_f(\partial R)) = \delta_{f,0}$ (see appendix B 2 for the proof). We see that only the B_0 term survives on the inner boundary. So the stand-alone subspace is the support space of B_0. Let's write B_0 explicitly

$$B_0 = \sum_{\{a_k, b_k; i_{k,k+1}\}} \prod_{j=1}^{3} (G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j 0} d^{\frac{5}{6}}_{a_j} d^{\frac{5}{6}}_{b_j} d^{\frac{1}{2}}_{i_{j,j+1}})$$
$$|\{a_k; i_{k,k+1}\}\rangle \langle\{b_k; i_{k,k+1}\}| \tag{55}$$

using normalization of G-symbols as given in Eq. (I.3)

$$G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j 0} = \frac{1}{\sqrt{d_{b_j} d_{b_{j+1}}}} \delta_{b_j, b_{j+1}, i_{j,j+1}} \delta_{a_j, b_j} \delta_{a_{j+1}, b_{j+1}},$$

hence we get

$$B_0 = \sum_{\{b_k; i_{k,k+1}\}} \prod_j^{3} (\delta_{b_j, b_{j+1}, i_{j,j+1}} d^{\frac{2}{3}}_{b_j} i^{\frac{1}{2}}_{j,j+1})$$
$$|\{b_k; i_{k,k+1}\}\rangle \langle\{b_k; i_{k,k+1}\}|. \tag{56}$$

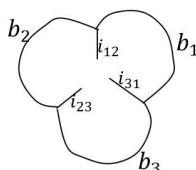B_0 is already written in eigenvalue decomposition form with $d^{\frac{2}{3}}_{b_j} i^{\frac{1}{2}}_{j,j+1}$ as the eigenvalues and $\prod_j \delta_{b_j, b_{j+1}, i_{j,j+1}} |\{b_k; i_{k,k+1}\}\rangle$ as the eigenvectors. So the projector on to the support basis of B_0 is simply

$$M_0 = \sum_{b_1, b_2, b_3; i_{12}, i_{23}, i_{31}} \delta_{b_1, b_2, i_{12}} \delta_{b_2, b_3, i_{23}} \delta_{b_3, b_1, i_{31}}$$
$$|b_1, b_2, b_3; i_{12}, i_{23}, i_{31}\rangle \langle b_1, b_2, b_3; i_{12}, i_{23}, i_{31}| \tag{57}$$

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

$$\left|\{\prod_{k=1}^{3} \delta_{b_k, b_{k+1}, i_{k,k+1}}\}\right\rangle = \delta_{b_1, b_2, i_{12}} \delta_{b_2, b_3, i_{23}} \delta_{b_3, b_1, i_{31}}$$
$$|b_1, b_2, b_3; i_{12}, i_{23}, i_{31}\rangle. \tag{58}$$

These basis vectors configurations,

$$\left|\{\prod_{k=1}^{3} \delta_{b_k, b_{k+1}, i_{k,k+1}}\}\right\rangle \tag{59}$$

So we get

$$dim(M_0) = \sum_{b_1, b_2, b_3; i_{12}, i_{23}, i_{31}} \delta_{b_1, b_2, i_{12}} \delta_{b_2, b_3, i_{23}} \delta_{b_3, b_1, i_{31}}. \tag{60}$$

Comparing Eq. 60 with Eq. 49, 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 in the appendix B and prove that topological entanglement entropy changes abruptly with such variations.

E. String-net MPO from Wilson loop operators

Now we calculate the MPO projector using Wilson loop operators. When an f-type simple string operator passes through the tensor T^0 on the physical level, it induces an operation on the virtual level in the way shown in Fig. 25. That is, it simply be-
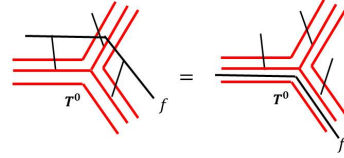
FIG. 25. An f-type simple string added on the physical legs (LHS) simply transforms to an f-type simple string added on the virtual legs (RHS).

comes a f-type string which is then fused with the plaquette legs. We consider the Wilson loop that encircles the 3 plaquettes of the tensor. This Wilson loop creates a f-type string that then fuses with the 3 plaquette loops. Remember that we need to calculate $M_f = M_0 W_f M_0$, 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 3 plaquettes and calculate induced operator on the stand-alone basis. It can be done in a convenient way using string-net diagrams in Eq. (59). 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 as shown in Fig. 25. Since these plaquette legs are contracted with the plaquette legs of the stand-alone tensor, it is equivalent to fusing f-string loop with the 3 plaquette legs of the stand-alone tensor. So essentially we get Fig. 26(a).

Step 2: We fuse these strings with the three nearby strings a_1, a_2, a_3. Let's say they fuse to make strings
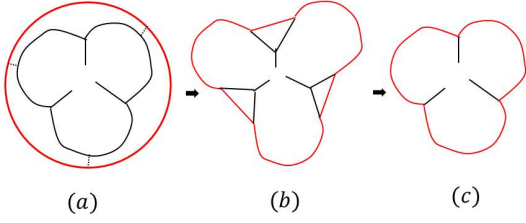
FIG. 26. Calculating f-type MPO, M_f, diagrammatically. A Wilson loop is applied on the physical level encircling 3 plaquettes. It essentially acts on the stand-alone basis as shown in (a). We then fuse it with plaquette legs leading to (b) and then finally remove the 3 bubbles, leading to (c). (c) is another basis state of the stand-alone space.

b_1, b_2, b_3. We gain factors $F^{a_j, a_j, 0}_{f, f, b_j} = \frac{\sqrt{d_{b_j}}}{\sqrt{d_f d_{a_j}}}$, $j = 1, 2, 3$ for each fusion.

Step 3: In the last step we remove the three bubbles created in the previous step. Each bubble removal produces a factor of $\sqrt{d_f d_{a_j} d_{a_{j+1}}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}$, $j = 1, 2, 3$.

Collecting the factors from step 2 and step 3 we get

$$\left\langle \{\prod_{k=1}^{3} \delta_{b_k, b_{k+1}, i_{k,k+1}}\} \middle| W_f \middle| \{\prod_{k=1}^{3} \delta_{a_k, a_{k+1}, i_{k,k+1}}\} \right\rangle$$
$$= \prod_{j=1}^{3} d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}. \quad (61)$$

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

$$\mathbb{M} = \sum_f \frac{d_f}{D} \prod_{j=1}^{3} d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \quad (62)$$

$$= \frac{1}{D} d_{b_j}^{\frac{1}{6}} d_{a_j}^{\frac{1}{6}} G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \quad (63)$$

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 pentagon identity in the second step. We can see that it projects on to the space with $G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} \neq 0$, that is $\delta_{i_{23}, i_{31}, i_{12}} \neq 0$.

There is a small technical issue though. The factor $d_{b_j}^{\frac{1}{2}} d_{a_j}^{\frac{1}{2}}$ does not exactly match the factors in the $TT^\dagger$ support space given in Eq. (47)). 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 Wilson loop around a single vertex is somewhat ill-defined. But we are only trying to get a symmetry condition on the individual tensors which makes sure that Wilson loop on a larger region are a symmetry of the state. We can show that this factor has to be exactly $d_{b_j}^{\frac{1}{6}} d_{a_j}^{\frac{1}{6}}$ if the Wilson loop is to be a symmetry of the

state. The reason is simply, as concluded in the original string-net paper, a Wilson loop commutes with the plaquette term $B_p = \sum_s a_s B_p^s$ only when $a_s = d_s$. In tensor network language, it translates to the fact that every tensor must contribute a factor of $d_s^{\frac{1}{6}}$ for the Wilson loop to be a symmetry. Also we know that an f-type Wilson loop applied to the ground state, produces a factor of d_f. Combining all these we can write the exact Wilson loop operator on a single tensor as (a detailed calculation of this expression is done in the appendix A 3)

$$M_f = \prod_{j=1}^{3} d_{a_j} (d_{b_j} d_{a_j}^{-1})^{\frac{1}{6}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \quad (64)$$

$$\Rightarrow \mathbb{M} = \frac{1}{D} d_{a_j} (d_{b_j} d_{a_j}^{-1})^{\frac{1}{6}} G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3} \quad (65)$$

The fixed point triple-line tensor satisfies

$$M_f T^0 = d_f T^0, \quad (66)$$
$$\mathbb{M} T^0 = T^0. \quad (67)$$

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

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

$$M_f(\partial R) = \prod_{j=1}^{n} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} d_{a_j} (d_{b_j} d_{a_j}^{-1})^{\frac{\theta_j}{2\pi}} \quad (68)$$

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

F. String-net zero-string operators

Zero-string operators of a model are the string operators that act trivially on the tensors along the path. We saw earlier that, for the single-line and double-line TNR of toric code, X-string and Z-string operators were the zero-string operators respectively. 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. (68) along the path with some changes at the end. But we don't worry too much about the details of how this operator looks at its ends, since those details can always be changed using local unitary operators at its ends. Looking at the Wilson-loop operators in Eq. (68), it is immediately clear what the zero-string operators are for the triple-line TNR of general string-nets. For $f = 0$, (using identity (I.3))

$$M_0(\partial R) = \prod_{j=1}^{n} \left(G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j 0} d_{b_j} (d_{a_j} d_{b_j}^{-1})^{\frac{m_j}{6}} \right)$$
$$= \prod_{j=1}^{n} \left(\delta_{a_j, b_j} (d_{a_j} d_{b_j})^{-\frac{1}{2}} d_{b_j} (d_{a_j} d_{b_j}^{-1})^{\frac{m_j}{6}} \right)$$
$$\equiv I. \quad (69)$$

So, a 0-type simple string operator is a zero-string operator.

It is also clear that for $f \neq 0$, M_f acts necessarily non-trivially on the tensors along the path. One should carefully note that, though non-zero-string operators change tensors along the path, it does not mean that this path is a physical observable. These paths can always be deformed as M_f passes through T^0 without any phase accumulation. It is called the 'pulling-through condition'[2]. 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.'

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

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

$$\delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} |a, b, c; i, j, k\rangle. \tag{70}$$

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

$$|v_{i,j,k}\rangle = \sum_{a,b,c} G_{a,b,c}^{i,j,k}(d_a d_b d_c)^{\frac{1}{6}} |a, b, c; i, j, k\rangle. \tag{71}$$

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

1- $\delta_{i,j,k} = 0$: This consists of all the string-configurations in Fig. 59 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 $v_{i,j,k}$. $\dim(\mathbb{V}_{i,j,k}) = \sum_{a,b,c} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = \sum_{a,b}[G_{c,a,b}^{i,j,k}]$ where $[G_{c,a,b}^{i,j,k}] = 1$ if $G_{c,a,b}^{i,j,k} \neq 0$ and 0 otherwise. Note that since $\mathbb{V}_{i,j,k}$ are orthogonal for different values of i, j, k, we just need to find vectors in individual $\mathbb{V}$ subspaces. To find these we will use the orthogonality of G (I.2)

$$\sum_c G_{a,b,c}^{i,j,k} G_{a,b,c}^{i,j,k} d_c = \frac{1}{d_k}\delta_{a,b,k} \tag{72}$$

and the fact that matrices N^k defined by $N_{a,b}^k = \delta_{a,b,k}$ can be simultaneously diagonalized $\forall k$. Let's say $|s_q\rangle =$

$s_{q;a} |a\rangle$ is its qth such simultaneous eigenvector. As discussed in the appendix A 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

$$\left|v_{i,j,k}^{q;a}\right\rangle = \sum_{a,b,c} \frac{s_{q;a}}{d_a} G_{a,b,c}^{i,j,k}(d_a d_b d_c)^{\frac{5}{6}} |a, b, c; i, j, k\rangle \tag{73}$$

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

$$\left\langle v_{i,j,k} \,\middle|\, v_{i,j,k}^{q;a}\right\rangle = \sum_{a,b,c} s_{q;a}d_b d_c G_{a,b,c}^{i,j,k} G_{a,b,c}^{i,j,k} \tag{74}$$

$$= \sum_{a,b} s_{q;a}\delta_{a,b,k}d_b \tag{75}$$

$$= \sum_{a,b} s_{q;a}\delta_{a,b,k}s_{0;b} \tag{76}$$

$$\propto \delta_{q,0}. \tag{77}$$

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

H. Triple-line TNR of the toric code and double semion states

Being a string-net model, toric code and double semion states also have a triple-line TNR. One can get the triple-line TNR for them by plugging in the relevant string-net data into Eq. (30). We will apply the results about general-string net models developed in previous sections to the two cases.

toric code string-net data is,

$$N = 1, \ d_0 = 1, \ d_1 = 1;$$
$$\delta_{000} = \delta_{110} = \delta_{101} = \delta_{110} = 1;$$
$$G_{000}^{000} = G_{111}^{000} = 1;$$
$$G_{011}^{011} = G_{100}^{011} = G_{101}^{101} = G_{010}^{101} = G_{110}^{110} = G_{001}^{110} = 1. \tag{78}$$

The triple-line TNR of toric code can be built by plugging in this data into the general expression in Eq. (30). This tensor has 9 virtual indices, each of which takes 2 values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$ dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 8, \tag{79}$$
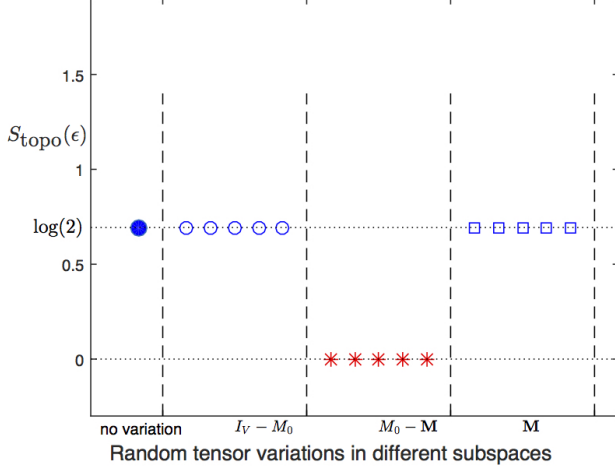
FIG. 27. Numerical calculation of topologiccal entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by toric code fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Details of this numerical calculation are given in appendix D 4.

and the dimension of the MPO subspace is

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 4. \qquad (80)$$

These imply that $\text{rank}(I_V - M_0) = 512 - 8 = 504$ and $\text{rank}(M_0 - \mathbb{M}) = 8 - 4 = 4$. So we reach the conclusion that out of 512 possible variations, 504 are stable since they are outside the stand-alone space. In the remaining 8 dimensional subspace, perturbations in a 4 dimensional subspace are in stable whereas the ones in the other 4 dimensional subspace are unstable. The numerical calculation supporting this conclusion is shown in Fig. 27. Also note that all unstable variations are flux variations, that is, it happens through the condensation of m-particle. It is not possible for the e-particle to condense in this way. The classification of all variations is shown in Fig. 28.

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

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

$$(81)$$

The triple-line TNR of the double semion model can be built by plugging in this data into the general expression in Eq. (30). This tensor has 9 virtual indices, each of which takes 2 values. So the full virtual space
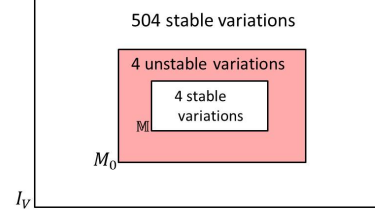

FIG. 28. Classification of the space of all variations to the toric code triple-line tensor into different subspaces. We see that stand-alone subspace is very small compared to the full virtual space, and half of it is unstable and the other half is stable.

is $\text{rank}(I_V) = 2^9 = 512$ dimensional. Dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 8, \qquad (82)$$

and the dimension of the MPO subspace is

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 4. \qquad (83)$$

These imply that $\text{rank}(I_V - M_0) = 512 - 8 = 504$ and $\text{rank}(M_0 - \mathbb{M}) = 8 - 4 = 4$. So we reach the conclusion that out of 512 possible variations, 504 are stable since they are outside the stand-alone space. In the remaining 8, 4 are in stable and 4 are unstable. The numerical calculation supporting this conclusion is shown in Fig. 29. 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 shown in Fig. 30.

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.

V. DOUBLE-FIBONACCI MODEL

Toric code and double-semion models are abelian models. Now we will discuss a non-abelian model: (double) Fibonacci model. Ground state of non-abelian string net models cannot be described by a single-line or a double-line TNR, it only accepts a triple line TNR (Fig. 23). Let's first describe the model briefly. The data for this
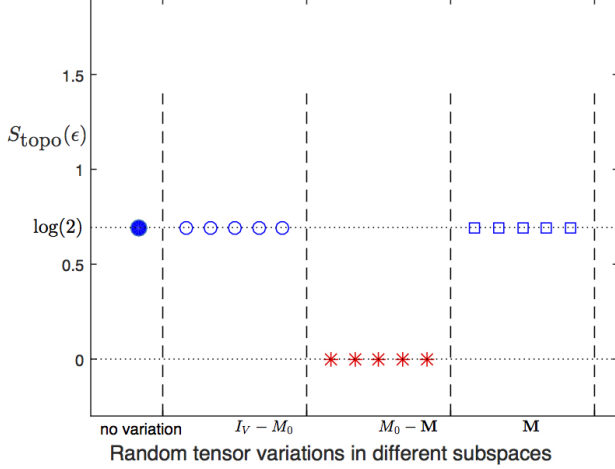
FIG. 29. Numerical calculation of topologiccal entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by double semion model fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Details of this numerical calculation are given in the appendix D 5.
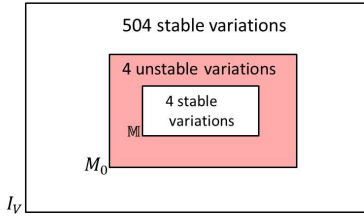

FIG. 30. Classification of the space of all variations to double semion triple-line tensor into different subspaces. We see that stand-alone subspace is very small compared to the full virtual space, and half of it is unstable and the other half is stable.

can be found in section IV.B of Levin and Wen[7]. There is one type of string ($N = 1$). Its quantum dimension is, $d_1 = \gamma = \frac{1+\sqrt{5}}{2}$. Its branching rules are,

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

$$d_0 = 1, \quad d_1 = \gamma, \quad \text{where } \gamma^2 = \gamma + 1 \qquad (84)$$

$$G^{111}_{111} = -\frac{1}{\gamma^2}; G^{110}_{111} = \frac{1}{\gamma}; G^{110}_{110} = \frac{1}{\gamma}; G^{000}_{111} = \frac{1}{\sqrt{\gamma}}; G^{000}_{000} = 1. \qquad (85)$$
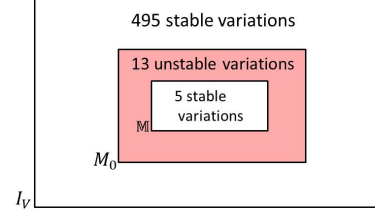

FIG. 31. Classification of the space of all variations to fibonacci triple-line tensor into different subspaces. We see that stand-alone subspace is 18 dimensional. It is again very small compared to the full virtual space. Out of 18 directions, 13 are unstable and 5 are stable.

The branching rules tells us that one string is allowed to branch into two, unlike the abelian models we have studied until now. First, let's apply our conjecture to find out how many unstable directions we should expect. The triple-line TNR of the fibonacci model can be built by plugging in this data into the general expression in Eq. (30). This tensor has 9 virtual indices, each of which takes 2 values. So the full virtual space is $\text{rank}(I_V) = 2^9 = 512$ dimensional. The dimension of the stand-alone space is

$$\text{rank}(M_0) = \sum_{a,b,c;i,j,k} \delta_{i,b,c}\delta_{j,c,a}\delta_{k,a,b} = 18, \qquad (86)$$

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

$$\text{rank}(\mathbb{M}) = \sum_{i,j,k} \delta_{i,j,k} = 5 \qquad (87)$$

which implies that $\text{rank}(I_V - M_0) = 512 - 18 = 496$ and $\text{rank}(M_0 - \mathbb{M}) = 18 - 5 = 13$. So we reach the conclusion that out of 512 possible variations, 496 are stable since they are outside the stand-alone space. In the remaining 18, 5 are in stable as they are in the MPO subspace and remaining 13 are unstable. The numerical calculation supporting this conclusion is shown in Fig. 32. The classification of all variations is given in Fig. 31.

Comparing it to the toric code and the double-semion models we see that the Fibonacci triple-line TNR is significantly more unstable. Another difference is that the stand-alone space does have vertex unstable variations in addition to plaquette ones. Out of 13 unstable variations in $M_0 - \mathbb{M}$ the following 3 are vertex variations and the rest 10 are plaquette variations:

$$|a,b,c;i,j,k\rangle = |1,1,1;1,0,0\rangle, |1,1,1;0,1,0\rangle, |1,1,1;0,0,1\rangle \qquad (88)$$

that is, the following 3 tensor components are allowed in the stand-alone space but not in the physical space. Since $\delta_{i,j,k} = \delta_{1,0,0} = \delta_{0,1,0} = \delta_{0,0,1} = 0$ these 3 vectors
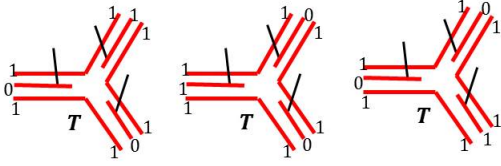
are not in the MPO subspace $\mathbb{M}$.

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

$$
T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-\frac{4}{5}\pi i} & 0 & 0 \\ 0 & 0 & e^{\frac{4}{5}\pi i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S = \frac{1}{1+\gamma^2} \begin{bmatrix} 1 & \gamma & \gamma & \gamma^2 \\ \gamma & -1 & \gamma^2 & -\gamma \\ \gamma & \gamma^2 & -1 & -\gamma \\ \gamma^2 & -\gamma & -\gamma & 1 \end{bmatrix}.
$$
(89)

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

$$
n_{4,0} = 1, n_{4,1} = 1, \Omega^0_{4,000} = 1, \Omega^1_{4,110} = 1,
$$
$$
\Omega^1_{4,001} = -\gamma^{-2}, \Omega^0_{4,111} = \gamma^{-1}, \Omega^1_{4,111} = \gamma^{-5/2},
$$
$$
\Omega^1_{4,101} = \Omega^{*1}_{4,011} = \gamma^{-11/4}(2 - e^{3\pi i/5} + \gamma e^{-3\pi i/5}). \quad (90)
$$

One can see that it is not a simple-string operator: when applied on the vacuum, it creates both 0-type and 1-type strings. So we see that the double-fibonacci model is different from the above two examples in one crucial aspect: the boson string operators in the toric code and the double-semion models were zero-string operators for the given TNRs. That is, the string operator 'disappeared' along the path (Fig. 820), not changing tensors along the path. This is why a single variations standing alone could be thought of as an operator sitting at the ends of an invisible string operator. But the same is not true for the double-fibonacci model. The string operator corresponding to the boson $\tau\bar{\tau}$ does not disappear in the middle.

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

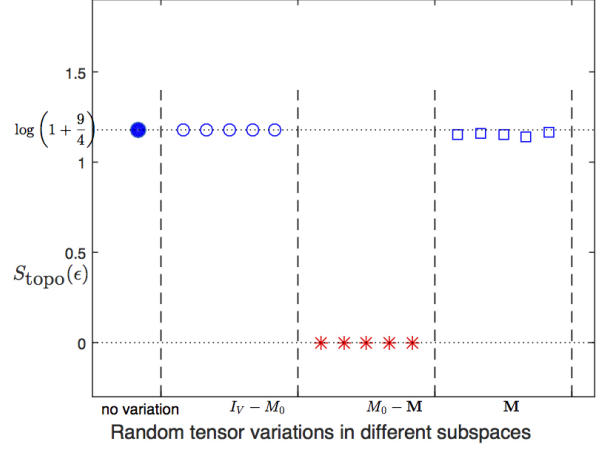We look at how the boson string-operator changes the tensors along the path. In Fig. 33, one can see that

FIG. 32. Numerical calculation of topologiccal entanglement entropy $S_{\text{topo}}(\epsilon)$ of states represented by fibonacci model fixed point triple-line tensors, T^0, varied with an infinitesimal random tensor in different subspaces. ϵ value is kept fixed at $\epsilon = 0.1$. Blue dot corresponds to S_{topo} with no variation. I_V is projector onto the full virtual space. M_0 is the projector on the stand-alone subspace. $\mathbb{M}$ is the MPO subspace projector. We take a random tensor and apply the projectors to generate random tensors in respective subspaces. Details of this numerical calculation are given in the appendix D 6.

a wavefunction corresponding to boson sitting at two places v_1 and v_2 is actually a superposition of many wavefunctions:

$$
|\Psi_{\text{boson}}\rangle = \sum_{t_1,s,t_2} n_s \Phi_{t_1,s,t_2} |\Psi_{\text{gs}}\rangle
$$
$$
= |\Psi_{0,0,0}\rangle + |\Psi_{1,0,0}\rangle + |\Psi_{0,0,1}\rangle + |\Psi_{1,0,1}\rangle
$$
$$
+ |\Psi_{0,1,0}\rangle + |\Psi_{1,1,0}\rangle + |\Psi_{0,1,1}\rangle + |\Psi_{1,1,1}\rangle (91)
$$

where the operator Φ_{t_1,s,t_2} is explained in Fig. 33. Φ_{t_1,s,t_2} is equivalent to applying $\Omega^{s'_1}_{4;t_1,s,s_1}$ and $\bar{\Omega}^{s'_n}_{4;s,t_2,s_n}$ on the loops at the ends of the string operator, and creating a s type string along the path. Fusing the loops with each other and with the s string along path P gives the final state. The important thing to note is that, though a TNR of the full state $|\Psi_{\text{boson}}\rangle$ involves changing tensors along the path, the TNR of $|\Psi_{t_1,0,t_2}\rangle$, $t_1, t_2 = 0, 1$, 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-
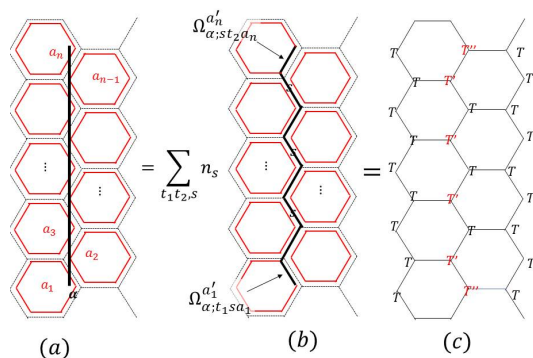
FIG. 33. Action of a generic (simple and non-simple) open-end string operators corresponding to anyon α on tensors can be calculated in a similar fashion as that of simple-string operator Wilson loops. (a) We start with applying the string operator on the 'loop state' on the fattened lattice. (b) The string operator becomes a superposition of operations $\sum_s n_s \Phi_{t_1 s t_2}$. $\Phi_{t_1 s t_2}$ acts as follows: at the ends, the string operator acts as $\Omega_{\alpha;t_1 s}$ and $\omega_{\alpha;s t_2}$ matrices on the plaquette-loops, while in the middle, it is simply a s-type string to be fused with the nearby plaquette loops. (c) We fuse all strings in the previous step to get the physical state. The effect of the string operator can be absorbed into redefining the tensors along the path. A generic string operator changes the tensors along its path. The only case where it doesn't change the tensors is for simple-string operators of type 0.

condensed state.

VI. CONCLUSIONS AND DISCUSSION

In this paper, we try to answer the following question: are the tensor network representation 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 topological order of a tensor network state may be numerically 'ill-posed'. That is, arbitrarily small numerical error in the process may change our conclusion in a qualitative way. Previous work[1] 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.1 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

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 'stand-alone' variations that break the Matrix-Product-Operator(MPO) symmetry introduced in Ref.2 and 14.

3. The physical reason for the instability is that 'stand-alone' variations which violate these symmetries induce condensation of bosonic quasi-particles and hence destroys (totally or partially) the topological order.

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.17.

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

On the other hand, if we want to properly simulate a topological phase transition induced by boson condensation, we need to put in the proper variational parameter. However, as we have seen in the case of the toric code, different representations (single line, double line, triple line) contain parameters corresponding to the condensation of different bosons (e or m). In fact, none of the representations contain parameters which correspond to the condensation of both bosons. Therefore, it is not possible to use any of them to correctly obtain the full phase

diagram. It implies that, if we want to study a topological phase transition whose nature is unknown, we need to try different ansätze. How to do that in an efficient and unbiased way is again an interesting open problem. We leave these problems to future study.

Appendix A: A brief review of string-net models

String-net models, which are Hamiltonian realizations of Turaev-Viro TQFTs, are introduced by Levin and Wen[16] as RG fixed point models that describe topological order in $2 + 1$ spacetime dimensions. Following are the defining data of the string-net states:

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

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

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

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

$$|\Psi\rangle = \sum_X \Phi_X |X\rangle \tag{A1}$$

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

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

$$\tag{A2}$$

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

$$F_{j^\star i^\star 0}^{ijk} = \frac{\sqrt{d_k}}{\sqrt{d_i}\sqrt{d_j}}\delta_{ijk}, \tag{A3}$$

$$F_{kln}^{ijm} = F_{jin}^{lkm^\star} = F_{lkn^\star}^{jim} = F_{k^\star nl}^{imj}\frac{\sqrt{d_m d_n}}{\sqrt{d_j d_l}}. \tag{A4}$$

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

$$\sum_f d_f G_{a_2 a_1 f}^{b_1 b_2 i_{12}} G_{a_3 a_2 f}^{b_2 b_3 i_{23}} G_{a_1 a_3 f}^{b_3 b_1 i_{31}} = G_{a_1 a_2 a_3}^{i_{23} i_{31} i_{12}} G_{b_1 b_2 b_3}^{i_{23} i_{31} i_{12}}. \tag{A5}$$

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

$$H = -\sum_v A_v - \sum_p B_p \tag{A6}$$

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

$$A_v = \sum_{i,j,k} \delta_{ijk} |ijk\rangle \langle ijk|. \tag{A7}$$

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

$$B_p = \sum_s \frac{d_s}{D} B_p^s, \tag{A8}$$

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

Finally putting all of it together, we see that the data $(N, d_i, \delta_{ijk}, F_{klm}^{ijk})$ describes 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',

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

G symbols also satisfy an 'orthogonality identity',

$$\sum_{i_{12}} G_{a_2 a_1 f}^{b_1 b_2 i_{12}} G_{a_2 a_1 f'}^{b_1 b_2 i_{12}} d_{i_{12}} = \frac{1}{d_f} \delta_{f,f'} \delta_{a_1 a_2 f} \delta_{b_1 b_2 f}. \quad (I.2)$$

G-symbols are normalized as

$$G_{a_2 a_1 0}^{b_1 b_2 i_{12}} = \delta_{a_1,b_1} \delta_{a_2,b_2} \delta_{a_1 b_1 i_{12}} (d_{a_1} d_{a_2})^{-\frac{1}{2}}. \quad (I.3)$$

Cyclic products of G symbols satisfy the following equation

$$\sum_{\{b_j\}} \prod_{j=1}^{n}(G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}} G_{b_{j+1} b_j f}^{c_j c_{j+1} i_{j,j+1}} d_{b_j}) \quad (I.4)$$

$$= \sum_s \delta_{ff's} \prod_{j=1}^{n}(G_{a_{j+1}a_j s}^{c_j c_{j+1} i_{j,j+1}}). \quad (A9)$$

Plaquette operators B_p^f correspondingly satisfy

$$B_p^f B_p^{f'} = \sum_s \delta_{ff's} B_p^s. \quad (I.5)$$

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

$$d_f d_{f'} = \sum_s \delta_{ff's} d_s \quad (I.6)$$

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

Define matrix N^k as $N_{a,b}^k = \delta_{k,a,b}$. 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

$$|s_q\rangle = \sum_a s_{q;a} |a\rangle. \quad (A10)$$
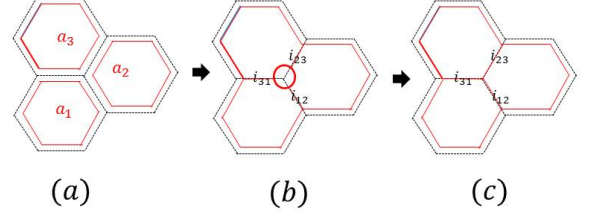

(a)   (b)   (c)

FIG. 34. Diagrammatic representation of derivation of the triple-line TNR of the string-net condensed state. We start with the 'loop state' on the fattened lattice given in Eq. (A13). We fuse loops with each, giving us the physical spins on the edges. Finally we remove the bubble on each vertex.

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

$$\langle s_q | s_{q'}\rangle \propto \delta_{q,q'}, \quad (A11)$$

$$\langle s_q | N^k | s_{q'}\rangle = \sum_{a,b} s_{q;a} \delta_{k,a,b} s_{q';b} \propto \delta_{q,q'}. \quad (A12)$$

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

2. Triple-line TNR of string-net states

We now briefly describe the derivation of triple-line TNR along the lines described in the original paper by Gu $et\ al.$[8]. It is important to understand this derivation as it gives us a way to apply string-operators on triple-line TNR.

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

$$|\Psi_{\text{gs}}\rangle = \prod_p B_p |0\rangle = \prod_p \sum_s d_s B_p^s |0\rangle$$

$$= \sum_{s_1,s_2,..} d_{s_1} d_{s_2}.. |s_1, s_2, ...\rangle \quad (A13)$$

where

$$|s_1, s_2, ...\rangle = B_{p_1}^{s_1} B_{p_2}^{s_2} \ldots |0\rangle. \quad (A14)$$

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

the final physical state,

$$|s_1, s_2, ...\rangle = \sum_{i_{12}, i_{23}...} \Phi^{i_{12} i_{23}..}_{s_1 s_2 s_3..} |i_{12}, i_{23}, ..\rangle . \quad (A15)$$

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

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

$$|\Psi_0\rangle = \sum_{s_1, s_2,..} d_{s_1} d_{s_2}.. |s_1, s_2, ..\rangle . \quad (A16)$$

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

Step 2: We fuse all loops with nearby loops producing a string on the links. We assume a 0-string between them and perform an F-move. It produces a factor of $\sum_{i_{j,k}} \sqrt{\frac{d_{i_{j,k}}}{d_{s_j} d_{s_k}}}$ on each link between plaquette j and k. A link is shared between two vertices, so each vertex gets a factor of $\sqrt[4]{\frac{d_{i_{j,k}}}{d_{s_j} d_{s_k}}}$.

Step 3: After the previous step, we are left with a 'bubble' on the vertex. Removing it produces a factor of $\sqrt{d_{s_j} d_{s_k} d_{s_l}} G^{i_{kl} i_{lj} i_{jk}}_{s_j s_k s_l}$.

Putting the 3 steps together, we get

$$(T^0)^{i_{jk} i_{kl} i_{lj}}_{s_l s_j s_k} = \sqrt[4]{d_{i_{jk}} d_{i_{kl}} d_{i_{lj}}} G^{i_{jk} i_{kl} i_{lj}}_{s_j s_k s_l} \sqrt[6]{d_{s_j} d_{s_k} d_{s_l}} \quad (A17)$$

3. Virtual symmetry of the triple-line TNR through Wilson loop

To be precise, we want to apply a Wilson loop operator on a slightly larger region (encircling 3 plaquettes) but want to finally see the induced operation on the virtual legs of a single tensor, without changing the nearby tensors. It should be a symmetry of the fixed point tensors. Wilson loop operators W_f represented on the inner legs of fixed point tensor T^0 as

$$\sum_\alpha (W_f)_{\alpha', \alpha} (T^0)^I_\alpha = d_f (T^0)^I_{\alpha'} \quad (A18)$$

We will use the same method as in A 2: we will start with the 'loop state' but with an additional f-type loop around encircling 3 plaquettes (Fig. 35(a)).

Step 1: Our state on the fattened lattice is

$$W_f |\Psi_0\rangle = W_f \sum_{\{a_j\}} d_{a_1} d_{a_2}.. |a_1, a_2, ...\rangle . \quad (A19)$$

We slide the Wilson loop in between the loops as shown in Fig. 35(a)→(b). Wilson loop simply becomes an f-type string in the space between the loops. But due to
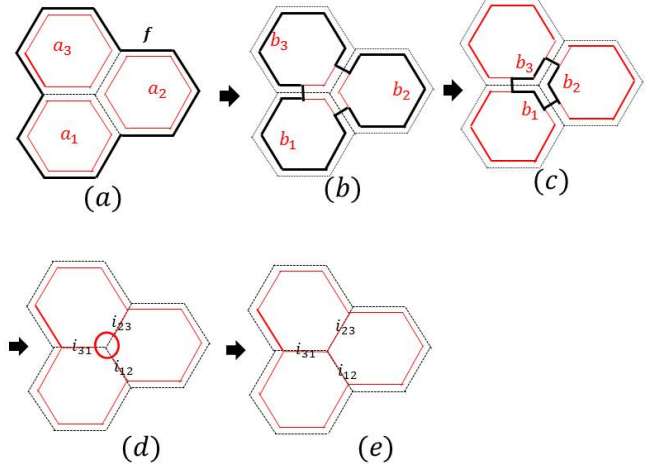

FIG. 35. Diagrammatic representation of applying f-type simply string operator Wilson loop on virtual indices of triple-line tensor. (a) We start with the ground state represented as the 'loop state' (Eq. (A13)) along with Wilson loop encircling 3 plaquettes . (b)Using the deformability of string-operators, we deform it to the space between the loops, where it becomes an f-type string. (c) Again using deformability across plaquettes, we deform it to a loop acting on a single tensor. We fuse f-type loop with the 3 plaquette loops. (d) We do a F-move to produce physical degrees of freedom on the links. (e) We remove the bubble at the vertex. Finally we collect all factors generated by this process into a single tensor to make sure nearby tensors don't change.

deformability of strings in ground state subspace, we can consider the equivalent configuration in which the f-loop acts only on a single tensor (35(b)→(c)). Each plaquette contributes a factor of $d_{a_j}^{1/6}$ to the tensor at the center.

Step 2: We first fuse loops with the f-type string. It produces a factor of

$$\prod_{j=1}^3 F^{a_j a_j 0}_{f f b_j} = \prod_{j=1}^3 (d_f^{-\frac{1}{2}} d_{b_j}^{\frac{1}{2}} d_{a_j}^{-\frac{1}{2}}) \quad (A20)$$

where b_1, b_2, b_3 are the final values. There is a sum over b_1, b_2, b_3 which we haven't written for brevity.

Step 3: Now we perform F-moves on each link (Fig. 35(c)→(d)). It produces a factor of

$$\prod_{j=1}^3 F^{b_j a_j f}_{a_{j+1} b_{j+1} i_{j,j+1}} = \prod_{j=1}^3 (d_f^{\frac{1}{2}} d_{i_{j,j+1}}^{\frac{1}{2}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} b_j f}) \quad (A21)$$

Note that in step 2 of the appendix A 2, vertices get a factor of $\sqrt[4]{\frac{d_{i_{jk}}}{d_{a_j} d_{a_k}}}$ from each link. So in order for the nearby tensors not to change, we get an additional factor of $\sqrt[4]{\frac{d_{a_j} d_{a_k}}{d_{i_{jk}}}}$ on each of the three links.

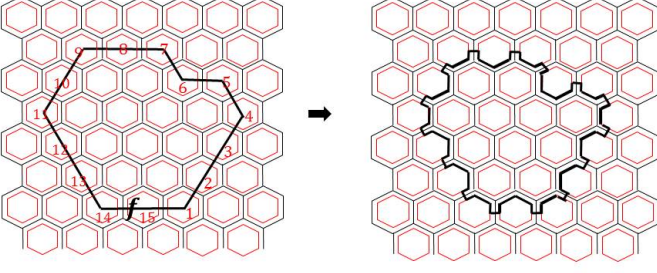**Step 4:** Finally we remove the bubble on the vertex

FIG. 36. Wilson loop of a simple-string operator of any size can be reduced to an operator that acts on the virtual indices of the boundary tensors. The calculation is the same as done for a Wilson loop around a single vertex. We first deform the Wilson loop to the space between the loops of the loop-state, fuse it with the loops, and finally perform F-moves to get the physical state on the lattice.

(Fig. 35(d)→(e)), which produces a factor of

$$\sqrt{d_{b_1} d_{b_2} d_{b_3}} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{A22}$$

Step 5: Accumulating all the factors in the previous 4 steps, we finally get an expression for the tensor at the vertex,

$$\sum_{b_1, b_2, b_3} \prod_{j=1}^{3} (G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} d_{b_j} d^{\frac{1}{4}}_{i_{j,j+1}} d^{\frac{1}{6}}_{a_j}) G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{A23}$$

Now using the expression for M_f in Eq. (65) and expression for T^0 in Eq. (30), we see that the above expression is equal to

$$\sum_{b_1, b_2, b_3} (M_f)_{a_1, a_2, a_3; b_1 b_2 b_3} (T^0)^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3} = M_f T^0 \tag{A24}$$

where

$$M_f = \prod_{j}^{3} d_{a_j} (d_{b_j} d^{-1}_{a_j})^{\frac{1}{6}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}. \tag{A25}$$

We can now immediately see why M_f is a symmetry of T^0. Instead of going through the above steps, we could just collapse the f-type string, which would produce a factor of d_f and go one with the usual procedure as done in appendix A 2. So we will end up with the tensor $d_f T^0$. The two ways of fusing strings should be equivalent, hence we get $M_f T^0 = d_f T^0$.

Though here we have computed MPO for a single tensor, we can apply the same definition to $T^0(R)$, that is, the tensor that results from contracting T^0 over some finite region R. In this case, M_f correspond to Wilson loop operators that encircle the region R on its boundary ∂R. Let's say $T^0(R)$ has dangling virtual legs on the

boundary as a_j, $j = 1, 2, \ldots, n$. $i_{j,j+1}$ again will denote the middle legs. M_f on the space spanned by $a_j, i_{j,j+1}$ is given by,

$$M_f(\partial R) = \prod_{j=1}^{n} \left(G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} d_{a_j} (d_{b_j} d^{-1}_{a_j})^{\frac{m_j}{6}} \right) \tag{A26}$$

where m_j takes values between 1 and 6, depending on how many vertices of the jth plaquette are inside the region R that has been contracted. If one vertex is inside, $m_j = 1$, if two are inside, $m_j = 2$ and so on. For a single tensor, one vertex of each of the 3 plaquettes is inside the region, that is why $m_j = 1$ giving us the factor of $d^{\frac{1}{6}}_{a_j}$ that appears in Eq. (65). Note that the identity in Eq. (A9) implies that M_f satisfies the string-algebra

$$M_f M_{f'} = \sum_{s} \delta_{f f' s} M_s \tag{A27}$$

which makes sense because M_f is nothing but adding f-type loop to the condensate and as such should satisfy the fusion rule.

Appendix B: Proof of the existence of instability in general string-net triple-line TNR

We will give an analytical proof of why all string-net triple-line TNR have at least one unstable direction which comes from the $M_0 - \mathbb{M}$ subspace. We will do so by directly calculating $S_{\text{topo}}(\epsilon)$. It is more convenient to work with the dual lattice of honeycomb lattice. The dual lattice is a 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, see Fig. 37. 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

$$(T^0)^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} = \prod_{j=1}^{3} \left(d^{\frac{1}{4}}_{i_{j,j+1}} d^{\frac{1}{6}}_{a_j} \right) G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3}. \tag{B1}$$

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 (see Fig. 38). 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

$$\mathbb{T}^0 = \prod_{j=1}^{n} \left(d^{\frac{1}{2}}_{i_{j,j+1}} (d_{a_j} d_{b_j})^{\frac{1}{6}} \right) G^{i_{23} i_{31} i_{12}}_{a_1 a_2 a_3} G^{i_{23} i_{31} i_{12}}_{b_1 b_2 b_3}. \tag{B2}$$
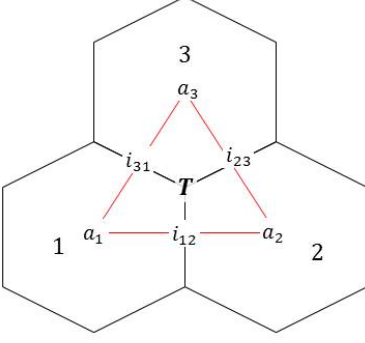
FIG. 37. Triple-line tensors as triangles on the dual lattice of the honeycomb lattice. The dual of honeycomb lattice is a triangular lattice. We label the vertices of the triangular lattice with integers, $j = 1, 2...$ Edges from vertex j to k are labeled as (j, k). We can associate a triple-line tensor to each triangle. The virtual indices, a_j live on the vertex j. The physical index and the middle leg virtual index, $i_{j,k}$ live on the edge (j, k). A simple example with $j = 1, 2, 3$ is shown in the figure.

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

$$\mathbb{T}^0 = \sum_f d_f B_f \tag{B3}$$

$$B_f = \prod_{j=1}^3 \left(d^{\frac{1}{2}}_{i_{j,j+1}} (d_{a_j} d_{b_j})^{\frac{1}{6}}, G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \right). \tag{B4}$$

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

$$B_f = \prod_{j=1}^n \left(d^{\frac{1}{2}}_{i_{j,j+1}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \right) \prod_{j=1}^n \left((d_{a_j} d_{b_j})^{\frac{1}{6}} \right). \tag{B5}$$

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

$$\mathbb{T}^0(\Delta) = \sum_f d_f B_f(\partial \Delta). \tag{B6}$$

The tensor resulting from contracting tensors $\mathbb{T}$ on a region R will be denoted as $\mathbb{T}(R)$. B_f is a f-type boundary operator. It lives on the boundary ∂R of a region R (not to be confused with the plaquette term B_p of string-net Hamiltonian). See Fig. 39. Let's say the vertices on the boundary of a region R on the triangluar lattice are labeled as $j = 1, 2, .., n$. We associate with each vertex a factor of $(a_j b_j)^{\frac{m_j}{6}}$. m_j denotes the number of the triangles inside R meeting at vertex j. It can simply be
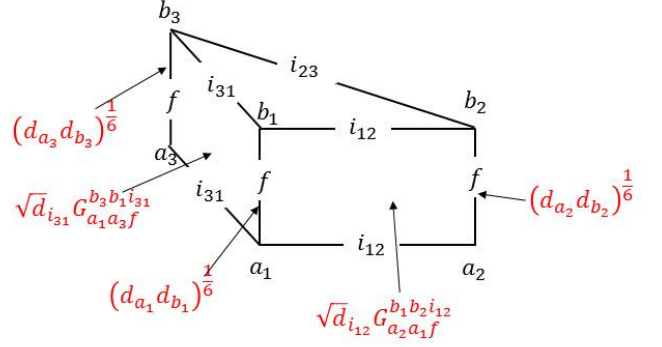

FIG. 38. Diagrammatic representation of boundary operator on a single triangle, $B_f(\partial \Delta)$. Different factors sit on different part of the double-triangle as shown.
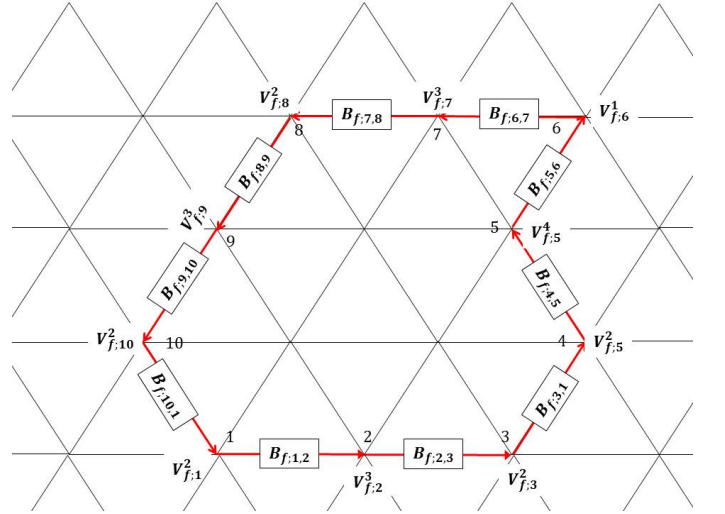
FIG. 39. Diagrammatic representation of the f-type boundary operator on a region R, $B_f(\partial R)$. It is a generalization of the boundary operator on a single triangle, as shown in Fig. 38; it sits on the boundary of a double layered triangular lattice. As is clear from its expression in Eq. (B7), it has two kinds of factors. Factors of the form $B_{f;j,j+1} = d^{\frac{1}{2}}_{i_{j,j+1}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}$ sit on the edge $(j, j+1)$, while the remaining factors, $V^{m_j}_j = (d_{a_j} d_{b_j})^{\frac{m_j}{6}}$ sit on the vertex j. f-strings connect the top and the bottom vertices as shown in Fig. 38.

written as $m_j = \theta_j/(2\pi/6)$, where θ_j is the angle the loop makes on vertex j. Finally, on every edge $(j, j + 1)$ we associate an operator $d^{\frac{1}{2}}_{i_{j,j+1}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f}$. With this construction, $B_f(\partial R)$ can be written as,

$$B_f(\partial R) = \prod_{j=1}^n \left(d^{\frac{1}{2}}_{i_{j,j+1}} G^{b_j b_{j+1} i_{j,j+1}}_{a_{j+1} a_j f} \right) \prod_{j=1}^n \left((d_{a_j} d_{b_j})^{\frac{m_j}{6}} \right). \tag{B7}$$

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

1. Result I: Boundary operators

Result: We find that the double tensor $\mathbb{T}^0(R)$ satisfies the general version of Eq. (B6):

$$\mathbb{T}^0(R) = D^V \sum_f d_f^{\chi_R} B_f(\partial R) \tag{B8}$$

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

$$\mathbb{T}^0(R) = \mathrm{Ev}\left(\mathbb{T}^0(\Delta_1)\mathbb{T}^0(\Delta_2)\ldots\right)$$
$$= \sum_{f_1, f_2, \ldots} (d_{f_1} d_{f_2} \ldots) B_{f_1}(\partial \Delta_1) B_{f_2}(\partial \Delta_2) \ldots \tag{B9}$$

where, as defined before, Ev 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. (I.2), edge contraction on the edge $(j, j+1)$ between B_f and $B_{f'}$ gives

$$\sum_{i_{j,j+1}} d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}} d_{i_{j,j+1}}^{\frac{1}{2}} G_{a_{j+1}a_j f'}^{b_j b_{j+1} i_{j,j+1}}$$
$$= \frac{1}{d_f} \delta_{f,f'} \delta_{a_j a_{j+1} f} \delta_{b_j b_{j+1} f}. \tag{B10}$$

The factor $\delta_{f,f'}$ implies that B_f only contracts with B_f. So the expression in Eq. (B9) is only non-zero for $f_1 = f_2 = \ldots f$. So we have

$$\mathbb{T}^0(R) = \sum_f d_f^F B_f(\partial \Delta_1) B_f(\partial \Delta_2) \ldots \tag{B11}$$

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. (B10) 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 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 (I.6)

$$\sum_{a_j, b_j} d_{a_j} d_{b_j} \delta_{a_j b_j f} = \sum_{a_j} d_f d_{a_j} d_{a_j} = D d_f. \tag{B12}$$

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 $(D d_f)^V = D^V d_f^V$ factor. Putting all the factors together, we get

$$\mathbb{T}^0(R) = \sum_f d_f^F d_f^{-E} (D d_f)^V B_f(\partial R)$$
$$= D^V \sum_f d_f^{\chi_R} B_f. \tag{B13}$$

This completes the proof of result I.

2. Result II: String-net stand-alone subspace

Result: The stand alone space of the triple-line string net TNR is given by

$$M_0 = \delta_{a_1, a_2, i_{12}} \delta_{a_2, a_3, i_{23}} \delta_{a_3, a_1, i_{31}}. \tag{B14}$$

Proof: First we note that though c_f are written as constants independent of the size of region R, they can depend on the topology of R. For example, for the fixed point tensors $\mathbb{T}^0$, we find $c_f = d_f^{\chi_R}$ where χ_R is the Euler characteristic of R. For a cylinderical topology, c_f will also depend on the boundary tensors. So we keep simple lattices in mind, such as sphere or torus or cylinder with boundaries. We divide it in symmetric two halves, let's say R and L, and calculate $\mathbb{T}(R)$. So the region of interest on which c_f should be calculated are either of the two halves. We assume the state has the appropriate symmetry such that $\sigma_L^T = \sigma_R = \sigma_b = \mathbb{T}(R)$. Using the result by[20], we know that the physical density matrix ρ_R has the same spectrum as σ_b^2, that is, $\rho_R \propto \sigma_b^2$. Let's say $\rho_R = N \sigma_b^2$, where N is the normalization factor. We first calculate N. To do that, we first need to calculate the algebra and the trace of B_f.

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

$$B_f(\partial R) = \prod_{j=1}^{n} \left(d_{i_{j,j+1}}^{\frac{1}{2}} (d_{a_j} d_{b_j})^{\frac{m_j}{6}} G_{a_{j+1}a_j f}^{b_j b_{j+1} i_{j,j+1}} \right) \tag{B15}$$

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

get,

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

$$= M_f \times \prod_{j}^{n} (d_{i_{j,j+1}}^{\frac{1}{2}}). \tag{B16}$$

This is how the boundary operator B_f and the MPO M_f are related. Since for abelian theories $d_i = 1, \forall i$, $B_f = M_f$. We know that

$$M_0 = \prod_{j=1}^{n} \delta_{i_{j,j+1},a_j,b_j} \tag{B17}$$

$$\Rightarrow B_0 = \prod_{j=1}^{n} d_{i_{j,j+1}}^{\frac{1}{2}} \delta_{i_{j,j+1},a_j,b_j}. \tag{B18}$$

We know the algebra of M_f operators, from Eq. (A27),

$$M_f M_{f'} = \sum_s \delta_{ff's} M_s. \tag{B19}$$

This essentially tells us the algebra of B_f operators as well,

$$B_f B_{f'} = \sum_s \delta_{ff's} B_s \times \prod_{j=1}^{n} (d_{i_{j,j+1}}^{\frac{1}{2}}) \tag{B20}$$

$$= \sum_s \delta_{ff's} B_s B_0. \tag{B21}$$

We also know how to contract B_f with each other through the calculations done in the appendix B 1. 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 $D d_f$ for every vertex. On a loop the number of vertices is equal to number of edges. So we get,

$$\mathrm{Tr}(B_f B_{f'}) = \delta_{f,f'} D^n. \tag{B22}$$

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

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

$$= \mathrm{Tr}(A_f^n) \tag{B23}$$

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 non-degenerate highest eigen-value λ_f, for large n, $\mathrm{Tr}(A_f^n) \approx \lambda_f^n$. Note that Perron-Frobenius theorem makes sure that λ_0, highest eigen-value of A_0, will be non-degenerate. So we have

$$\lim_{n\to\infty} \mathrm{Tr}(B_0) = \lambda_0^n. \tag{B24}$$

For abelian models, $\mathrm{Tr}(B_{f\neq 0} = 0$ since $G_{baf}^{abi} = 0, f \neq 0$. For fibonacci model, 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 $\mathrm{Tr})(B_1) \approx \mathrm{Tr}(A_1^n) \approx 0$.

Now we will now prove an important result for all string-net models:

$$\lim_{n\to\infty} \frac{\mathrm{Tr}(B_{f\neq 0})}{\mathrm{Tr}(B_0)} = \lim_{n\to\infty} \frac{\mathrm{Tr}(B_{f\neq 0})}{\lambda_0^n} = 0. \tag{B25}$$

To prove this, we would calculate S_{topo} on a sphere and compare it to the known result, $S_{\mathrm{topo}} = \log D$. On a hemisphere, $\chi_R = 1$, so 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,

$$\mathrm{Tr}(\sigma_b^2) = \mathrm{Tr}(\sum_f d_f B_f)^2$$

$$= \sum_{f,f'} d_f d_{f'} \mathrm{Tr}(B_f B_{f'})$$

$$= \sum_{f,f'} d_f d_{f'} \delta_{f,f'} D^n$$

$$= D^n (\sum_f d_f^2). \tag{B26}$$

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

$$S_{1/2}(\rho_R) = \frac{1}{1-1/2} \log \mathrm{Tr}(\rho_R^{\frac{1}{2}})$$

$$= 2 \log \frac{\mathrm{Tr}(\sum_f d_f B_f)}{\sqrt{D^n \sum_f d_f^2}}$$

$$= -n \log D - 2 \log \sum_f (d_f \mathrm{Tr} B_f) - \log \sum_f d_f^2$$

$$= -n \log D - 2n \log \lambda_0 - 2 \log \left(1 + \sum_{f>0} \frac{\mathrm{Tr} B_f}{\lambda_0^n} \right)$$

$$- \log \sum_f d_f^2. \tag{B27}$$

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

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

$$\mathbb{T}(D - \Delta) = \sum_f B_f(\partial(D - \Delta)) \tag{B28}$$

$$= \sum_f B_f(\partial\Delta) \otimes B_f(\partial D). \tag{B29}$$

To get the stand-alone space, we simply trace out the inner indices on the outer edge. Using Eq. (B25), we simply get (up to an overall normalization factor which

we ignore) B_0. As noted earlier M_0 is the projector onto the support space of B_0, so we have proved that M_0 given in Eq. (B14) is the stand-alone space:

$$M_0 = \delta_{a_1,a_2,i_{12}} \delta_{a_2,a_3,i_{23}} \delta_{a_3,a_1,i_{31}}. \qquad (B30)$$

This completes the proof of result II.

3. Result III: Topological entanglement entropy on a cylinder with non-RG fixed point tensor

Result: Lets say we divide the cylinder in two halves (Fig. 4(a)). We denote the right half as R. If any given tensor network on this cylinder satisfies,

$$\lim_{|R|\to\infty} \mathbb{T}(R) = C^{|R|} \sum_f c_f B_f(\partial R) \qquad (B31)$$

where C is some constant, then, S_{topo}, as given in Eq. (10), is

$$S_{\text{topo}} = \log \sum_f \left(\frac{c_f^2}{c_0^2} \right). \qquad (B32)$$

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

$$
\begin{aligned}
\mathrm{Tr}(\sigma_b^2) &= \mathrm{Tr}(\sum_f c_f B_f)^2 \\
&= \sum_{f,f'} c_f c_{f'} \mathrm{Tr}(B_f B_{f'}) \\
&= \sum_{f,f'} c_f c_{f'} \delta_{f,f'} D^n \\
&= D^n (\sum_f c_f^2). \qquad (B33)
\end{aligned}
$$

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

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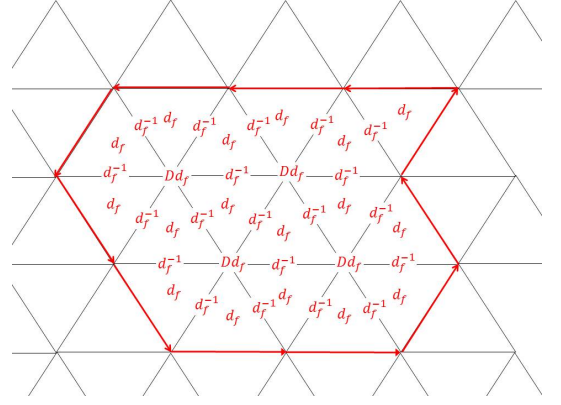&\quad + 2 \log c_0 - \log \sum_f c_f^2.
\end{aligned}
$$

FIG. 40. As we contract double-tensors with each other on the triangular lattice, each face contributes a factor of d_f, each edge contributes a factor of d_f^{-1}, and each vertex contributes a factor of $D d_f$. So, overall, contracting double tensors on a region R produces a factor of $D^V d_f^{\chi_R}$, where $\chi_R = F - E + V$, and F, E and V are the number of faces, edges and vertices inside the region R, respectively.

When we let $n \to \infty$ and using Eq. (B25)

$$S_{1/2}(\rho_R) = n \log \frac{\lambda_0^2}{D} - \log \sum_f \left(\frac{c_f^2}{c_0^2} \right) \qquad (B34)$$

$$\Rightarrow S_{\text{topo}} = \log \sum_f \left(\frac{c_f^2}{c_0^2} \right). \qquad (B35)$$

This completes the the proof of result III.

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

4. Result IV: Instability in string-net

Result: 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,$(M_0 - \mathbb{M})T^q = T^q$) that for the variation $T^0 \to T^0 + \epsilon T^q$, $\lim_{\epsilon \to 0} S_{\text{topo}}(\epsilon) \neq S_{\text{topo}}(0)$.

Proof: Combination of the results I and III 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 section IV G that there are two kinds of variations in $M_0 - \mathbb{M}$, vertex variations (that violate the vertex term) and plaquette variations (that violate the plaquette term). We will treat them one by one.

Before we do any analytical calculation, let us describe in simple words what the reason for instability is. We saw in the appendix B 1 that as fixed point tensors contract, every face, every edge and every vertex contributes a factor of d_f, d_f^{-1} and d_f respectively. 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 B 3 changes. We will now show that this precisely what variations in $M_0 - \mathbb{M}$ do. In particular, the vertex variations change the vertex factors , and the plaquette variations change the face factors.

Let's choose a particular tensor variation T^q such that

$$\mathbb{T}^q = T^q (T^q)^\dagger = B_0. \quad (B36)$$

This tensor is supported on the full M_0 space. Then, double tensor for the varied tensor is

$$\mathbb{T} = (T^0 + \epsilon T^q)(T^0 + \epsilon T^q)^\dagger \approx \mathbb{T}^0 + \epsilon^2 B_0 \quad (B37)$$

$$= (1 + \epsilon^2) B_0 + \sum_{f>0} d_f B_f. \quad (B38)$$

We have ignored the linear terms in ϵ as they are contained within the MPO subspace, and we don't need to worry about them. This double tensor will contract with itself in exactly the same way as $\mathbb{T}^0$ did, but the only difference is, now every face will contribute a factor of r_f, where, $r_0 = (1 + \epsilon^2)$, and $r_{f>0} = d_f$. The vertex factors and edge factors will remain to be d_f and d_f^{-1}, respectively. After contracting it on a large region we will get a double tensor $\mathbb{T}(R) = \sum_f c_f B_f(\partial R)$, where $c_f = r_f^F d_f^{V-E}$. So $c_0 = (1 + \epsilon^2)^F$ and $c_{f>0} = d_f^{\chi_R}$. So we see that c_0 is exponentially larger than $c_{f>0}$ even for an infinitesimal ϵ, hence, using Eq. (B35), $S_{\text{topo}} = 0$. This ends the proof.

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 the appendix A 1. The double tensor will again produce a factor of d_f on the faces, and d_f^{-1} on the edges upon contraction. But now the factors on the vertices would be

$$\sum_{a,b} \delta_{a,b,f}(d_a + \epsilon s_{q;a})(d_b + \epsilon s_{q;b}) = D(d_f + e_{q;f}\epsilon^2) \quad (B39)$$

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

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

5. Variations in the MPO space $\mathbb{M}$

Now we will show that any infinitesimal variation contained within $\mathbb{M}$ corresponds to an infinitesimal physical variation. And since string-net models are gapped, we do not expect them to cause a phase transition.

A singular value decomposition of $(T^0)_\alpha^I = U_{\alpha,\alpha'} S_{\alpha',I'} V_{I',I}^\dagger$ gives us

$$T^0 = \sum_i s_{i,i} |u_i\rangle \langle v_i| \quad (B40)$$

where u_i are the columns of U, $s_{i,i}$ are the singular values in S and v_i are the columns of V. u_i form a basis of the virtual space, while v_i form a basis of the physical space. Then the support space of T^0 is nothing but

$$\mathbb{M} = \eta_i |u_i\rangle \langle u_i| \quad (B41)$$

where $\eta_i = 1$, if $|s_{i,i}| > 0$ otherwise $\eta_i = 0$. Now consider a random tensor $(T^r)_\alpha^I$. It can always be written as

$$T^r = \sum_{i,j} r_{i,j} |u_i\rangle \langle v_j|. \quad (B42)$$

Then,

$$\mathbb{M}T^r = \sum_{i,j} \eta_i r_{i,j} |u_i\rangle \langle v_j|$$

$$= \sum_{i,i} \eta_i |u_i\rangle \langle w_i| \quad (B43)$$

$$\text{where } \langle w_i| = \sum_j r_{i,j} \langle v_j|. \quad (B44)$$

So the varied tensor $T^0 \rightarrow T^0 + \epsilon \mathbb{M}T^r$ is

$$T^0 + \epsilon \mathbb{M}T^r = \sum_i s_{ii} |u_i\rangle (\langle v_i| + \epsilon \langle w_i|)$$

$$= US(V(I + O(\epsilon)))^\dagger \quad (B45)$$

where O is an infinitesimal operator with components $O_{i,j} = \epsilon \langle v_i \, | \, w_j \rangle$. So we have shown that any tensor variation in the MPO space corresponds to applying an operator on the physical legs infinitesimally close to identity. Since the system is gapped, we do not expect such an operator to change the phase of the system.

Appendix C: Dependence of S_{topo} on boundary conditions in cylindrical geometry

Topological entanglement entropy calculation is done by calculating the entanglement entropy of a subsystem A. When the boundary of A consists of topologically trivial loops, for example when A has a disc geometry, S_{topo} is known to depend only on the total quantum dimension D, $S_{\text{topo}} = \log D$. However when the boundary of A consists of non-contractible topologically non-trivial loops, for example when a torus or cylinder is divided into two cylinders, it has been shown by Zhang $et\ al.$ [23] that
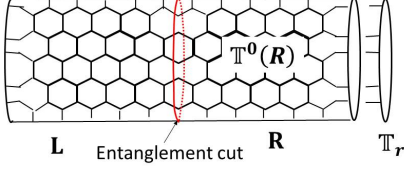
FIG. 41. We calculate entanglement entropy of the right-half of the cylinder with a certain boundary condition $\mathbb{T}_r$. The entanglement cut is in the middle of the cylinder.
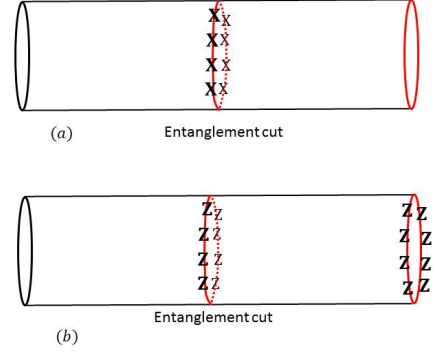

FIG. 42. MESs are eigenstates of different Wilson loop operators at the entanglement cut. (a)For fixed point single-line TNR, the state on the cylinder is always in $+1$ eignestate of X-loop, as it identically disappears. (b)The state is also in $+1$ eigenstate of simultaneous operation of two Z-loops, one at the entanglement cut, other at the right-most boundary. It implies, we can be in two MESs depending on the the boundary tensor choice. If the boundary tensor is in $+1$ eigenstate of the boundary Z loop, then the state is in $+1$ eigenstate of the entanglement-cut Z loop. Similarly, if the boundary tensor is in -1 eigenstate of the boundary Z loop, then the state is in -1 eigenstate of the entanglement-cut Z-loop.

S_{topo} also depends on the linear combination of ground states. For a ground state wave function on a torus

$$|\Psi\rangle = \sum_a c_a |\Xi_a\rangle \tag{C1}$$

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

$$S_n = \alpha_n L - S_{\text{topo}}, \tag{C2}$$

$$S_{\text{topo}} = 2\log D - \frac{1}{1-n}\log\left(\sum_a p_a^n d_a^{2(1-n)}\right) \tag{C3}$$

where d_a is the quantum dimension of ath quasi-particle and $p_a = |c_a|^2$. $|\Xi_a\rangle$ are special basis for which S_{topo} is maximal, or entanglement entropy is minimal. These states are called the *Minimum Entropic States (MES)*. It was shown that MES correspond to eigenstates of Wilson-loop operators along the entanglement cut.

This dependence of S_{topo} on the ground state is of crucial importance to us since we have used cylinder with a boundary for S_{topo} calculations. So, numerically obtained S_{topo} contain information about the boundary as well. For example, consider the toric code.

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

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

Let's first take the example of the single-line TNR of the toric code. See Fig. 42. 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. 41). The four MES correspond to four eigen states of e and m Wilson-loops on the entanglement cut. But, since e-Wilson loop is a zero-string operator, the state is always in its $+1$ eigenstate (Fig. 42(a)). So we have access to only two MES corresponding to ± 1 eigenstates of m-Wilson loop. We also know that the

state is in $+1$ eigenstate of the $Z_{\partial R}^{\otimes} = Z_{ec}^{\otimes L} \otimes Z_r^{\otimes L}$, 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 $+1$ eigenstate of $Z_{ec}^{\otimes L} \otimes Z_r^{\otimes L}$ (see Fig. 42)(b), the state can be either in $+1$ eigen-state of both $Z_r^{\otimes L}$ and $Z_{ec}^{\otimes L}$ or in -1 eigenstate of the both. The boundary tensor determines which eigenstate of $Z_r^{\otimes L}$ the wave function is in, and consequently also which eigenstate of $Z_{ec}^{\otimes L}$. This is how the boundary tensors and MES are connected. Since we have access to only two MES

$$S_{\text{topo}} = \log 2 - \frac{1}{1-n}\log(p_1^2 + p_2^2). \tag{C4}$$

A similar analysis follows in the double-line TNR, with the role of e and m Wilson loop operators reversed: now the state is always in the $+1$ eigen state of m-Wilson loop and the two MES correspond to the two eigenstates of e Wilson loop at the entanglement cut, which in turn depends on the boundary tensors.

We saw in the appendix B 3 $\rho_R = N\sigma_b^2$ where

$$\sigma_b = \mathbb{T}^0(R)\mathbb{T}_r \tag{C5}$$

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

$$\mathbb{T}^0(R) = \sum_f d_f^{\chi_R} B_f(\partial R)$$

$$= (B_0)_{ec} \otimes (B_0)_r + (B_1)_{ec} \otimes (B_1)_r, \tag{C6}$$

where $B_0 = I^{\otimes L}$ and $B_1 = Z^{\otimes L}$ for the single-line TNR and $B_1 = X^{\otimes L}$ for the double-line TNR. Let's say the
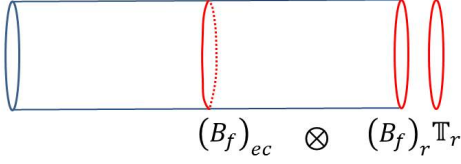
FIG. 43. Bulk double tensor is a sum of tensor product between $(B_f)_{ec}$ (B_f on the entanglement cut) and $(B_f)_r$ (B_f on the right boundary). So, when we contract a boundary tensor $\mathbb{T}_r$ with the bulk tensor, it contract with $(B_f)_r$ giving a scalar c_f. So resulting tensor is $\text{Ev}(\mathbb{T}(R)\mathbb{T}_r) = \sum_f c_f (B_f)_{ec}$. Consequently, S_{topo} using Eq. (B35) is simply $\log\left(\sum_f \frac{c_f^2}{c_0^2}\right)$.

boundary double tensor $\mathbb{T}_r$ contracts with $(B_f)_r$ to produce the constants c_f (see Fig. 43)

$$\begin{aligned}
\sigma_b &= ((B_0)_{ec} \otimes (B_0)_r + (B_1)_{ec} \otimes (B_1)_r)\,\mathbb{T}_r \\
&= c_0 (B_0)_{ec} + c_1 (B_1)_{ec} \\
&= c_- B_- + c_+ B_+.
\end{aligned} \quad (C7)$$

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,

$$B_\pm^2 = B_\pm, \quad \text{Tr}(B_\pm) = 2^{L-1}. \quad (C8)$$

With this, we get the normalized density matrix as,

$$\rho_R = \frac{1}{2^L}\left(\frac{c_-^2}{c_-^2 + c_+^2} B_- + \frac{c_+^2}{c_-^2 + c_+^2} B_+\right) \quad (C9)$$

$$= \frac{1}{2^L}\left(p_- B_- + p_+ B_+\right). \quad (C10)$$

The nth Renyi entropy is,

$$\begin{aligned}
S_n(\rho_R) &= \frac{1}{1-n}\log\text{Tr}(\rho_R^n) \\
&= \frac{1}{1-n}\log\text{Tr}\left(\frac{1}{2^{nL}}(p_-^n B_- + p_+^n B_+)\right) \\
&= \frac{1}{1-n}\log\left(\frac{1}{2^{nL}}(p_-^n 2^{L-1} + p_+^n 2^{L-1})\right) \\
&= L\log 2 - \left(\log 2 - \frac{1}{1-n}\log(p_-^n + p_+^n)\right).
\end{aligned} \quad (C11)$$

Comparing it with the MES formula in Eq. (C4), we see that $p_1 = p_- = c_0 - c_1$ and $p_2 = p_+ = c_- + c_+$. So the state is an MES if $p_\pm = 0 \Rightarrow c_0 = \pm c_1$ for which we get maximal topological entanglement entropy, $S_{\text{topo}} = \log 2$. This illustrates the direct dependence of S_{topo} on $\mathbb{T}_r$.

Of course the above analysis is done for the RG fixed point tensors only. We have to choose a boundary double tensor $\mathbb{T}_r$ such that S_{topo} is truly indicative for topological order, or lack of it, for both RG fixed point and varied tensors. We choose the following boundary tensor for our numerical calculations: For any tensor network,
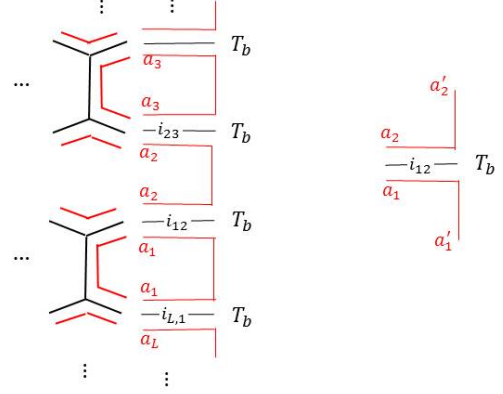

FIG. 44. Smooth boundary condition for triple-line tensor network. Tensors T_b are used on the boundary. T_b has 5 virtual legs, $a_1, a_1', a_2, a_2', i_{12}$ and 1 physical leg, i_{12}. Physical leg and the middle leg take the same values. We assign a particular value to the components of this tensor, $(T_b)^{i_{12}}_{i_{12}a_1a_1';a_2a_2'} = \delta_{i_{12},0}\delta_{a_1,a_1'}\delta_{a_2,a_2'}\delta_{a_1a_2i_{12}}$.

fixed point or varied, We use a 'smooth boundary condition'. It is explained in the Fig. 44. First we will explain it for the triple-line tensors. For double-line and single-line an appropriately reduced version of T_b will be used. Note that we haven't drawn the physical index explicitly and it should be understood the same as the middle index (the index in black color). So the boundary tensor T_b has four virtual indices, and we fix its components to be,

$$(T_b)^{i_{12}}_{i_{12}a_1a_1';a_2a_2'} = \delta_{i_{12},0}\delta_{a_1,a_1'}\delta_{a_2,a_2'}\delta_{a_1a_2i_{12}} \quad (C12)$$

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

Before we discuss why we choose this particular boundary, let us calculate what S_{topo} we are supposed to get with this particular choice of boundary tensor. For that, we need to calculate $c_f = B_f \mathbb{T}_r$. Note that $\delta_{a_j,a_{j+1},0}$ implies $a_j = a_{j+1}$. So the double tensor $\mathbb{T}_r$ is

$$\mathbb{T}_r = \sum_{a,b} |a,a,a..;000..\rangle \langle b,b,b,...;000...|. \quad (C13)$$

So

$$c_f = \text{Ev}(B_f \mathbb{T}_r)$$
$$= \sum_{a,b} \prod_{j=1}^{m} G_{a,a,f}^{b,b,0}(d_a d_b)^{\frac{1}{2}}$$
$$= \sum_{a,b} \prod_{j=1}^{m} \delta_{a,b,f}$$
$$= \sum_{a,b} \delta_{a,b,f}. \qquad (C14)$$

Then using Eq. (B35), S_{topo} is simply $\log(\sum_f \frac{c_f^2}{c_0^2})$. For the toric code, and double semion models $c_0 = c_1 = 2$, so we get $S_{\text{topo}} = \log 2$. For the double Fibonacci model, however, we get

$$c_0 = \sum_{a,b} \delta_{a,b,0} = \delta_{0,0,0} + \delta_{1,1,0} = 2 \quad (C15)$$

$$c_1 = \sum_{a,b} \delta_{a,b,1} = \delta_{1,0,1} + \delta_{0,1,1} + \delta_{1,1,1} = 3. \quad (C16)$$

$$(C17)$$

So we get $S_{\text{topo}} = \log(1 + \frac{3^2}{2^2}) = \log(1 + \frac{9}{4})$, which is consistent with our numerical result.

There are mainly two reasons why we choose this particular boundary condition

1- This is a very simple boundary condition which gives us a precise analytical value of the topological entanglement entropy (namely, $\log(\sum_f \frac{c_f^2}{c_0^2})$, with c_f given in Eq. (C14)) against which numerical calculations can be checked.

2- Though situation for non-abelian cases is more complicated, this boundary is definitely MPO symmetric for abelian models. That is, we expect the tensor network state to be an MES with maximal S_{topo} (=$\log D$).

Numerical calculations of S_{topo} will be checked against the analytical result in Eq. C14. 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. But $S_{\text{topo}}(\epsilon) = 0$ needs to be further verified as it might be because of the particular boundary conditions imposed. To verify, we will test for S_{topo} dependence on infinitesimal variation on the boundary tensors. The reason for this is 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. 45. We first fix the boundary tensor to be T_b given in Eq. 44 and calculate the S_{topo} for variations in $I_V - M_0, M_0 - \mathbb{M}$ and $\mathbb{M}$
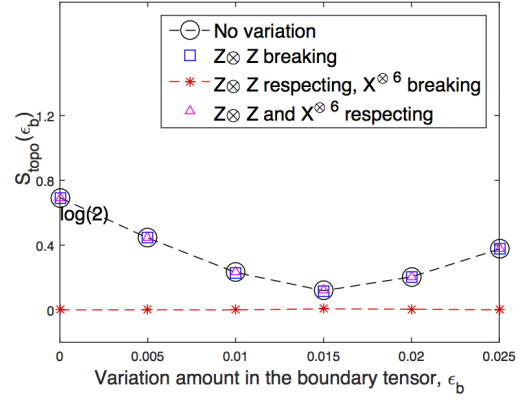

FIG. 45. Dependence of S_{topo} on boundary condition for toric-code double line TNR. We start with the boundary tensor, T_b, shown in Fig. 44. We add a random variation $\epsilon_b T_b^r$ to T_b and calculate $S_{\text{topo}}(\epsilon_b)$ for random bulk variations in different subspaces. We keep T_b^r 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.

subspaces added to the fixed point bulk tensor. Now we add an infinitesimal random variation to the boundary tensor, $T_b \rightarrow T_b + \epsilon_b T_b^r$. ϵ_b (different from ϵ, which the bulk variation strength) is the strength of the boundary variation. We increase ϵ_b slowly and for each value of the ϵ_b we calculate $S_{\text{topo}}(\epsilon)$ for random bulk variations in different subspaces. Fig. 45 shows S_{topo} as a function of ϵ_b for bulk variations in different subspaces. (the bulk variation strength ϵ is kept fixed throughout). We observe that

1- The variations which are unstable (i.e. $S_{\text{topo}} = 0$) for T_b, continue to be unstable for $T_b + \epsilon_b T_b^r$ for all values of ϵ_b. It implies that we get $S_{\text{topo}} = 0$ for these variation because the bulk topological order is indeed destroyed and not because of a specific boundary tensor chosen which gave an accidental zero.

2-The variations which are stable (i.e. $S_{\text{topo}} = \log 2$) for T_b, have the same value of S_{topo} as the fixed point tensor for all boundary tensors. It implies that tensor network state with these variations indeed have the same topological order as the fixed point tensor network state. Though this verification is shown for double-line toric code only, we find the same behavior for all numerical calculations presented in this paper.

It should be noted that any strictly positive value of S_{topo} (assuming sufficiently large cylinder was considered) is a sufficient condition for topological order but it is not a necessary condition. So all we need to do is to avoid getting accidental zeros.
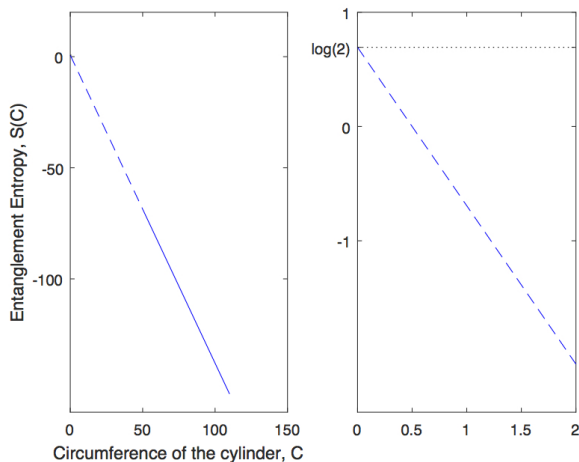
FIG. 46. Calculation of S_{topo} for single-line toric code fixed point tensor network state. We fix half cylinder length as $L = 500$. Circumference is varied from 50 to 110. S varies linearly with C. This line is extrapolated back to $C = 0$. Its intersection with the y-axis gives S_{topo}. Right figure is a zoomed in version of the left figure to show the intersection point clearly. We find $S_{\text{topo}} \approx \log(2)$
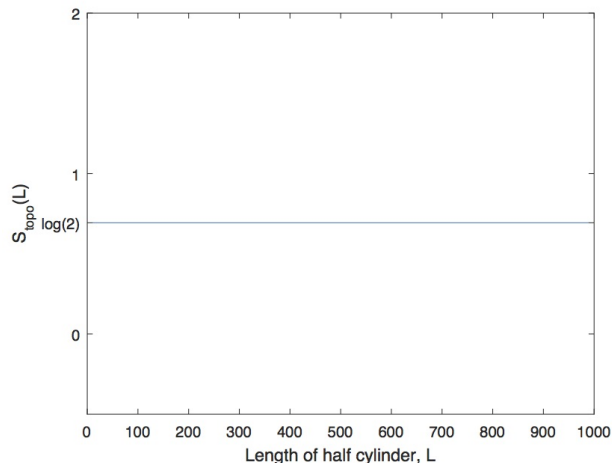

FIG. 47. S_{topo} was calculated for a fixed half cylinder length, $L = 500$, in Fig. 46. We now vary L from 10 to 1000. We see that S_{topo} is converged even for small values of L. So one does not need to large cylinder length to get the right S_{topo} value. It is expected as it is an RG fixed point tensor network state.

Appendix D: Details of numerical calculations

Here we will provide the various numerical details and data regarding the numerical calculations whose results were presented in the main text.

First, we will show convergence of numerical calculation of S_{topo}. We choose the simplest case, the single-line TNR of toric code. We first repeat the algorithm described in section II B in simple words here for conve-

nience. In the first step, the transfer matrix is calculated using the tensor given (fixed point or varied). Then we choose a specific boundary double tensor as explained in the appendix C. We apply the transfer matrix on this boundary double tensor and approximate the resulting tensor as an MPS of bond dimensions $D_{\text{cut}} = 8$. We apply transfer matrix again and approximate the resulting tensor as an MPS of bond dimension 8. We repeat this process and each repetition physically corresponds to increasing the longitudinal length of the our cylindrical subsystem by one unit. Let's say we repeat this process until the length of the half cylinder subsystem is equal to L. This process gives us the virtual density matrix σ, and assuming the mirror symmetry of transfer matrix, the physical reduced density matrix of the half cylinder is $\rho_L \propto \sigma^2$. With this reduced density matrix we calculate the entanglement entropy S of the half cylinder subsystem for different circumferences C. We plot $-S$ vs C and extrapolate it to $C = 0$ which gives us the topological entanglement entropy $S_{\text{topo}} = S(C = 0)$. In principle, one needs to take infinitely large cylinder to achieve the precise value of S_{topo}. Practically, we need to keep increasing L until we get a fixed point MPS and keep increasing C until the S_{topo} value converges to a fixed point.

Let's first look at the calculation for the single-line toric code fixed point tensor in Eq. 4. Half cylinder length is fixed at $L = 500$. C is varied from 50 to 110. Fig. 46 shows the entanglement entropy S vs the circumference C. We get a straight line which is extrapolated to $C = 0$. The right figure is a zoomed in version of the left figure to see clearly where the extrapolated line crosses the y-axis. We get $S_{\text{topo}} = S(C = 0) \approx \log(2)$ as expected. Fig. 47 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 $\epsilon = 0.01$. Half cylinder length is fixed at $L = 500$. C is varied from 50 to 110. Fig. 48 shows entanglement entropy S vs the circumference C. We get a straight line which is extrapolated to $C = 0$. The right figure is a zoomed in version of the left figure to show clearly where the extrapolated line crosses the y-axis. We see $S_{\text{topo}} \approx 0$. To see the effect of cylinder length we calculate S_{topo} again but with different cylinder lengths. The results are shown in Fig. 49. We see that S_{topo} is log(2) for small cylinders but converges to zero as the length is increased. Comparing it to Fig. 47 we see that, unlike the fixed point case, we need to consider large enough cylinder ($L > 600$ in this case) to calculate the correct S_{topo} value for the non-fixed point tensor network state.

Finally we show the effect of variation strength, ϵ, on the convergence. In above calculation we fixed $\epsilon = 0.01$. Now we vary ϵ from 0.01 to 0.02 (making sure it is well be-
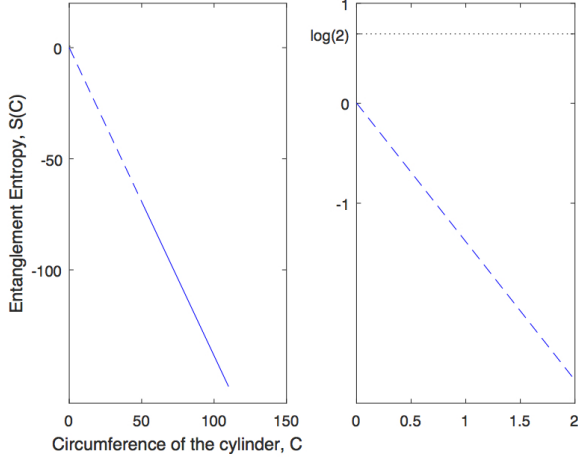
FIG. 48. Calculation of S_{topo} for a state represented by single-line toric code fixed point tensor varied with an MPO violating tensor. The strength of the variation is fixed at $\epsilon = 0.01$. We fix half cylinder length as $L = 500$. Circumference is varied from 50 to 110. S varies linearly with C. This line is extrapolated back to $C = 0$. Its intersection with the y-axis gives S_{topo}. Right figure is a zoomed in version of the left figure to show the intersection point clearly. We find $S_{\text{topo}} \approx 0$, that is, it is a trivial state.

FIG. 49. S_{topo} was calculated for a fixed half cylinder length, $L = 500$, in Fig. 48. We now vary L from 10 to 1000. We see that S_{topo} is close to $\log(2)$ for small cylinders but converges to zero cylinder length L is increased from 1 to 1000. So it is indeed a topologically trivial state.

low any critical points) and calculate corresponding convergence plots similar to Fig. 49. The results are shown in Fig. 50. We see that the strength of the variation has a huge effect on convergence. Bigger variations lead to faster convergence.

Though we have presented details of calculation only for one case (single-line toric code TNR), it should be understood that similar patterns are followed in all other cases. For completeness, we present the numerical data plotted in the main text and the relevant parameters used in each case.

FIG. 50. The variation strength ϵ affects convergence. Higher the variation strength (as long as it is below any critical points) faster is the convergence with the length of the size of the system.

1. Single-line TNR toric code

The bond dimension of the MPS is kept fixed at $D_{\text{cut}} = 8$ at each step of the iteration. The starting MPS is as explained in the appendix C. The strength of the variations is fixed at $\epsilon = 0.01$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110.

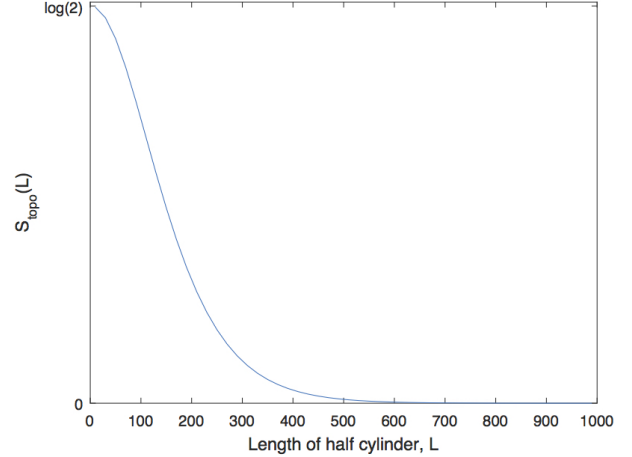Following table contains the exact values of the S_{topo} plotted in Fig. 6.

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 the appendix C. The strength of the variations is fixed at $\epsilon = 0.01$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 13.

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 the appendix C. The strength of the variations is fixed at $\epsilon = 0.01$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 22.

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 the appendix C. The strength of the variations is fixed at $\epsilon = 0.2$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 27.

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 the appendix C. The strength of the variations is fixed at $\epsilon = 0.2$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 1000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 29.

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 the appendix C. The strength of the variations is fixed at $\epsilon = 0.1$. Half cylinder length is either the length at which convergence of S_{topo} is reached (convergence is reached when S_{topo} value in two successive steps differ by less than 10^{-7}) or $L = 2000$, whichever is smaller. The circumference is varied from 50 to 110. Following table contains the exact values of the S_{topo} plotted in Fig. 32.

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] X. Chen, B. Zeng, Z.-C. Gu, I. L. Chuang, and X.-G. Wen, Phys. Rev. B **82**, 165119 (2010).

[2] M. B. Şahinoğlu, D. Williamson, N. Bultinck, M. Mariën, J. Haegeman, N. Schuch, and F. Verstraete, ArXiv e-prints (2014), arXiv:1409.2150 [quant-ph].

[3] M. Fannes, B. Nachtergaele, and R. Werner, Communications in Mathematical Physics **144**, 443 (1992), 10.1007/BF02099178.

[4] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[5] F. Verstraete, V. Murg, and J. Cirac, Advances in Physics **57**, 143 (2008).

[6] G. Vidal, ArXiv e-prints 0912.1651 (2009), arXiv:0912.1651 [cond-mat.str-el].

[7] M. A. Levin and X.-G. Wen, Phys. Rev. B **71**, 045110 (2005), cond-mat/0404617.

[8] Z.-C. Gu, M. Levin, B. Swingle, and X.-G. Wen, Phys. Rev. B **79**, 085118 (2009), arXiv:0809.2821 [cond-mat.str-el].

[9] O. Buerschaper, M. Aguado, and G. Vidal, Phys. Rev. B **79**, 085119 (2009).

[10] S. Yan, D. A. Huse, and S. R. White, Science **332**, 1173 (2011).

[11] H.-C. Jiang, Z. Wang, and L. Balents, Nat Phys **8**, 902 (2012).

[12] S. Depenbrock, I. P. McCulloch, and U. Schollwöck, Phys. Rev. Lett. **109**, 067201 (2012).

[13] S. Bravyi, M. B. Hastings, and S. Michalakis, Journal of Mathematical Physics **51**, 093512 (2010).

[14] O. Buerschaper, Annals of Physics **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] Z.-C. Gu, M. Levin, and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008), arXiv:0806.3509 [cond-mat.str-el].

[18] A. Kitaev, Annals of Physics **303**, 2 (2003).

[19] N. Schuch, I. Cirac, and D. Prez-Garca, Annals of Physics **325**, 2153 (2010).

[20] J. I. Cirac, D. Poilblanc, N. Schuch, and F. Verstraete, Phys. Rev. B **83**, 245134 (2011).

[21] S. T. Flammia, A. Hamma, T. L. Hughes, and X.-G. Wen, Phys. Rev. Lett. **103**, 261601 (2009).

[22] S. Dong, E. Fradkin, R. G. Leigh, and S. Nowling, Journal of High Energy Physics **5**, 016 (2008), arXiv:0802.3231 [hep-th].

[23] Y. Zhang, T. Grover, A. Turner, M. Oshikawa, and A. Vishwanath, Phys. Rev. **B85**, 235151 (2012), arXiv:1111.2342 [cond-mat.str-el].

[24] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[25] A. Kitaev, *January Special Issue*, Annals of Physics **321**, 2 (2006).

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

[27] G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015).

[28] M. Freedman, C. Nayak, K. Shtengel, K. Walker, and Z. Wang, Annals of Physics **310**, 428 (2004).