Quantum error correction with the semion code

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
23 December 2018

REVISED
19 April 2019

ACCEPTED FOR PUBLICATION
2 May 2019

PUBLISHED
29 May 2019

PAPER

Quantum error correction with the semion code

G Dauphinais, L Ortiz, S Varona and M A Martin-Delgado

Departamento de Física Teórica, Universidad Complutense, E-28040 Madrid, Spain

E-mail: gdauphin@ucm.es

Keywords: quantum information, quantum error correction, topological codes, double semion model

Abstract

We present a full quantum error correcting procedure with the semion code: an off-shell extension of the double-semion model. We construct open-string operators that recover the quantum memory from arbitrary errors and closed-string operators that implement the basic logical operations for information processing. Physically, the new open-string operators provide a detailed microscopic description of the creation of semions at their end-points. Remarkably, topological properties of the string operators are determined using fundamental properties of the Hamiltonian, namely, the fact that it is composed of commuting local terms squaring to the identity. In all, the semion code is a topological code that, unlike previously studied topological codes, it is of non-CSS type and fits into the stabilizer formalism. This is in sharp contrast with previous attempts yielding non-commutative codes.

1. Introduction

Topological properties of quantum systems have become a resource of paramount importance to construct quantum memories that are more robust to external noise and decoherence [1–7] than standard quantum error correcting codes [8–15]. The latter are based on a special class of codes—concatenated codes—which enable us to perform longer quantum computations reliably, as we increase the block size.

The Kitaev code is the simplest topological code yielding a quantum memory [1]. It can be thought of as a simple two-dimensional lattice gauge theory with gauge group $G = \mathbb{Z}_2$. In $D = 2$ spatial dimensions, there is another lattice gauge theory with the same gauge group but different topological properties: the double semion (DS) model [16–18, 22–25]. Although the Kitaev and the DS models are lattice gauge theories sharing the same gauge group, $G = \mathbb{Z}_2$, the braiding properties of their quasiparticle excitations are radically different. For example, whereas braiding two elementary quasiparticle excitations (either an electric or a magnetic charge) gives a $+1$ phase in the Kitaev code, doing so in the DS model yields $\pm i$ phase factors.

The DS model was introduced in the context of the search of new topological orders in strongly correlated systems, gapped, non-chiral and based on string-net mechanisms in $D = 2$ dimensions [16, 17]. Generalizations of the DS model to $D = 3$ and to higher dimensions have appeared recently [18]. While the properties of the Kitaev code has been extensively studied in quantum computation and condensed matter, barely nothing is known about the quantum error correcting properties of the DS model despite recent efforts towards realizing such models [19–21]. In this work we remedy this situation by introducing a new formulation of the DS model that is suitable for a complete treatment as a quantum memory with topological properties.

The first obstacle to tackle the DS model as a quantum memory is the original formulation as a string-net model [16]. In this formulation, the Hamiltonian is only Hermitian and exactly solvable in a particular subspace, where plaquette operators are Hermitian and commute. Only linear combinations of closed-string configurations, implying the absence of vertex excitations, are allowed in this subspace [16, 26, 27]. The microscopic formulation of the original DS model starts with a hexagonal lattice Λ with qubits placed at links e. Vertex operators Q_v are attached to the three links meeting at a vertex $v \in \Lambda$. Plaquette operators $\tilde{B}_p$ are attached to hexagons $p \in \Lambda$ with the novel feature that their outer links carry additional phase factors that are missing in the corresponding Kitaev model. Explicitly

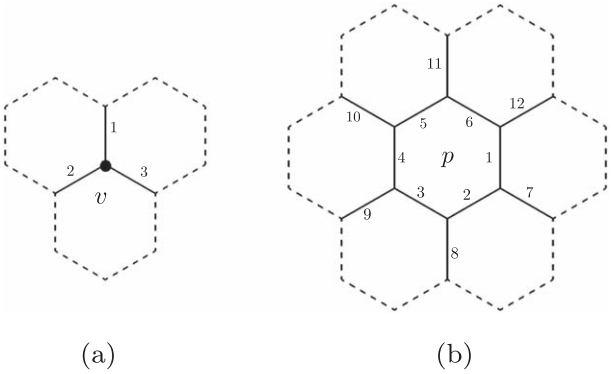

Figure 1. The support of the vertex operator $Q_v = \sigma_1^z\sigma_2^z\sigma_3^z$ and plaquette operator $\tilde{B}_p$. The qubits are placed on the edges.

$$Q_v \coloneqq \sigma_i^z\sigma_j^z\sigma_k^z, \tag{1}$$

with i, j, k, being the three qubits belonging to vertex v (see figure 1(a)) and

$$\tilde{B}_p \coloneqq \left(\prod_{i\in\partial p}\sigma_i^x\right)\prod_{j\in o(p)} i^{\frac{1}{2}(1-\sigma_j^z)}, \tag{2}$$

where ∂p are the six links of the hexagon and $o(p)$ is the set of six edges outgoing from each plaquette p, as it is shown in figure 1(b). Unlike the Kitaev code, these plaquette operators $\tilde{B}_p$ are Hermitian and commute among themselves only in a subspace of the whole Hilbert space, defined by the so-called zero-flux rule [16, 26, 27]. This is given by a vertex-free condition on states

$$\mathcal{H}_v \coloneqq \{|\psi\rangle\colon Q_v|\psi\rangle = +|\psi\rangle, \forall v \in \Lambda\}. \tag{3}$$

The set of vertex and plaquette operators defines a Hamiltonian

$$\tilde{H}_{\mathrm{DS}} \coloneqq -\sum_{v\in\Lambda} Q_v + \sum_{p\in\Lambda}\tilde{B}_p. \tag{4}$$

Due to the involved structure of phases that plaquette operators in equation (2) have, it was aforementioned that these operators do not commute out of the vertex-free subspace. This implies that the model is only well-defined when there are no vertex excitations.

Therefore, in order to treat the DS model as a quantum error correcting code [11, 28–31], it is necessary to have a formulation of the model that is valid in the whole Hilbert space and not just for the vertex-free subspace (3), since generic noise processes will make the system leave the mentioned subspace. To this end, we introduce an off-shell DS model that we call the semion code. This new model is achieved by making a deformation of the original plaquette operators (2) such that they become commuting and Hermitian operators without imposing the vertex-free condition [27]. In addition, we are able to develop the complete program of quantum error correction with the semion code.

1.1. Summary of main results

In order to summarize the main contributions that we present in this paper, we hereby advance a list of some of our most relevant results.

(i) We perform a thorough analysis of a formulation of the plaquette operators which commute in the whole Hilbert space [27]. This construction consists in adding extra phases to the plaquette operators, which depend on the configuration of the three edges at each vertex. When the vertex-free condition is imposed, we recover the standard definition of the DS model.

(ii) We give an explicit construction for string operators along arbitrary paths. They are complete in the sense that any operator acting on the system can be decomposed as a linear superposition of such operators. Additionally, the string operators can be constructed efficiently despite the complex structure of the plaquette operators. Remarkably, a microscopic formulation to create semions was not proven until now.

(iii) We analytically show that the excitations of the system behave as semions, via the detailed study of the constructed string operators, which allows us to explicitly calculate the topological S-matrix. Interestingly

enough, most of the string operator properties rely on very generic arguments about the structure of the local operators making up the Hamiltonian, namely that they commute and square to the identity.

(iv) Closed-string operators are constructed, which allow us to perform logical operations on the quantum memory built from the semion code. Logical operators are closed-string operators whose paths are homologically non-trivial and act non-trivially on the degenerate ground space.

(v) Given the above properties, we define a topological quantum error correcting code based on a non-trivial extension of the DS model. We build a code, which is characterized by the following key properties: is topological, satisfies the stabilizer formalism, is non-CSS, non-Pauli and additive.

The topological nature of the code becomes apparent through the fact that global degrees of freedom are used to encode information and only local interactions are considered [1, 4]. Another remarkable feature of the semion code is that is of non-CSS type since in the plaquette operators both Pauli X and Z operators enter in the definition [9, 28]. Consequently, errors in an uncorrelated error model such as independent bit-flip and phase-flip errors cannot be treated separately. However, they can be decomposed as a linear superpositions of fundamental *anyonic* errors (string operators creating pairs of anyons) with a known effect on the Hilbert space. Moreover, the semion code is not a subgroup of the Pauli group since the complex phases entering its definition (see equation (2)) makes impossible to express its generators in terms of tensor products of Pauli matrices [32, 33]. Nevertheless, the semion code is still an additive code: the sum of quantum codewords is also a codeword [34]. This last fact is intimitely related to the abelian nature of the semionic excitations [35, 36].

1.2. Outline

The article is organized as follows. In section 2 we introduce the off-shell DS model, which is suitable for quantum error correction. In section 3 we build string operators creating vertex excitations at their endpoints. Section 4 is devoted to logical operators and quantum error correction. We conclude in section 5. Appendices deserve special attention since they contain the detailed explanations of all the constructions used throughout the text. Specifically, appendix A presents an explicit example of a string operator, appendix B gives the detailed proof of theorem 1, which presents a systematic way to construct string operators, as well as several key properties of the string operators and finally appendix C is devoted to the proof of theorem 2, which gives the commutation relations among string operators.

2. Off-shell DS: microscopic model

We begin by considering a microscopic description of the DS model on the entire Hilbert space of states, since it is much better suited for quantum error correction. We call this an off-shell DS code by borrowing the terminology from quantum field theory and other instances in physics where a shell condition amounts to a constraint on the phase space of a system. For instance, the equation of motion is a shell condition for quantum particles but the phase space is more general. In our case, the shell condition is the vertex-free subspace or zero-flux rule introduced in equation (3).

2.1. DS model in the vertex-free subspace

Let us start by introducing the DS model in a new presentation that is more suitable for building an off-shell formulation of it. We consider the same hexagonal lattice Λ with qubits attached to the edges $e \in \Lambda$. The vertex operators Q_v will remain the same as in equation (1), but the plaquette operators in the zero-flux subspace can be rewritten in an equivalent form to that shown in equation (2), i.e.

$$\tilde{B}_p = \left(\prod_{j=1}^{6} \sigma_j^x\right)\left(\prod_{j=1}^{6} (-1)^{n_{j-1}^- n_j^+}\right), \tag{5}$$

in which $n_i^\pm := \frac{1}{2}(1 \pm \sigma_i^z)$ is the projector on the state $|0\rangle$ (n^+) or $|1\rangle$ (n^-) of qubit i, qubits are labeled as shown in figure 1 and we use the convention that $n_0^\pm$ refers to qubit '6' for simplicity. Remarkably, this expression avoids any reference to the outgoing links of hexagonal plaquettes p. One readily sees that the vertex operators fulfill the following relations:

$$Q_v^2 = 1, \quad [Q_v, Q_{v'}] = 0, \quad [Q_v, \tilde{B}_p] = 0; \tag{6}$$

$\forall\, v, v', p \in \Lambda$. As for the plaquette operators $\tilde{B}_p$, they also satisfy

$$\tilde{B}_p^2 = 1, \quad \tilde{B}_p^\dagger = \tilde{B}_p, \quad [\tilde{B}_p, \tilde{B}_{p'}] = 0; \tag{7}$$

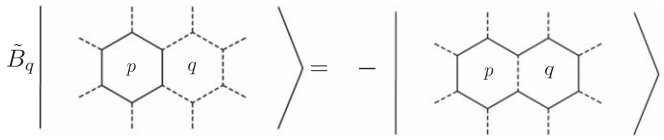

Figure 2. Applying plaquette operator $\tilde{B}_q$ adds a closed loop around plaquette q and multiply the wave function by a -1 phase. Each qubit in a given configuration is represented by a filled link of the lattice if it is in state $|1\rangle$, whereas links in $|0\rangle$ are left empty (dashed line).

$\forall\, p,\, p' \in \Lambda$, but only in the vertex-free subspace (3). Furthermore, the product of all the vertex and plaquette operators is the identity. A simple counting argument reveals that the ground space is 4^g degenerate[1], which g being the genus of the orientable compact surface onto which the lattice is placed.

An explicit unnormalized wavefunction belonging to the ground space is obtained in the following way: we start from the vacuum, i.e. $|0\rangle^{\otimes N}$, which has $+1$ eigenvalue for all vertex operators. Then, plaquette operators are used to build projectors and apply them onto the vacuum

$$|\Omega\rangle = \prod_{p\in\Lambda} \frac{1-\tilde{B}_p}{2}\, |0\rangle^{\otimes N}. \tag{8}$$

It is straightforward to check that this state fulfills the lowest energy condition for the Hamiltonian (4) within the vertex-free subspace. Expanding the product in equation (8), one can see that the ground state is a superposition of closed loops configurations. Due to the condition $\tilde{B}_p = -1$ for the ground state, the coefficients of this superposition of closed loops alternate sign. Thus, we can write the ground state in a different way:

$$|\Omega\rangle = \sum_{\vec{i}\ \in\{\text{C-S conf.}\}} (-1)^{N_L(\vec{i})} |\vec{i}\rangle, \tag{9}$$

where $\vec{i}$ is a bitstring representing a qubit configuration and $\{\text{C-S conf.}\}$ is the set of all possible closed-string configurations. Each configuration in this set has a certain number of closed loops, $N_L(\vec{i})$, whose parity determines the sign of the coefficient in the ground state superposition.

Of course the above construction only gives rise to one of the ground states. To find the other ones, the starting configuration can simply be replaced by a configuration containing an homologically non-trivial closed loop (which necessarily belongs to the vertex-free subspace), and proceed with the same construction. Every different homological class for the closed loop corresponds to a different ground state.

Applying the $\tilde{B}_p$ operator on a specific loop configuration flips the string occupancy of the interior edges of plaquette p while acquiring a phase that depends on the specific configuration under consideration. Applying $\tilde{B}_p$ on the vacuum simply adds a closed loop around plaquette p, while applying $\tilde{B}_p$ next to a closed loop either enlarges (or shrinks) the existing loop to include (exclude) plaquette p, while multiplying the wave function by -1 factor (see figure 2).

Due to the lack of commutativity of the plaquette operators and the fact that they are not Hermitian, the original DS model is only well-defined when there are no vertex excitations. Moreover, the strings creating vertex excitations are not properly defined either. A naive attempt to construct these strings as a chain of σ^x operators, following the similarities with the Kitaev code, resoundingly fails. As a consequence of the phases on the external legs of plaquette operators, σ^x operators create vertex excitations but also plaquette excitations. In order to get a string that creates only two vertex excitations at the endpoints but commutes with all the plaquette operators along the path $\mathcal{P}$, it is necessary to add some extra phases to the chain of σ^x on the outer legs. An approach to this problem is described in [27] but it is not successfully solved since the strings are only well-defined in the vertex-free subspace.

The DS model gives rise to quasiparticle excitations behaving like anyons. They are called semions, due to the fact that their topological charge is 'half' of that of a fermion, i.e. $\pm i$. There exist two types of semions in the model, one corresponding to a vertex excitation, while the other corresponds to both vertex and adjacent plaquettes excitations. From now on, we name these two different possibilities as *chiralities*, positive chirality for the former kind, and negative chirality for the latter one. We warn the reader that this choice has been made arbitrarily, and does not necessarily reflect the topological charge of a given specie.

[1] This is strictly true only if the total number of plaquettes of the system is even. If it is odd, then the ground state must contain a single flux excitation, which can be placed in any of the plaquettes. In that latter case the ground space degeneracy is an extensive quantity. However, any given flux configuration is 4^g-degenerate. For simplicity, we assume in this work that the system contains an even number of plaquettes.

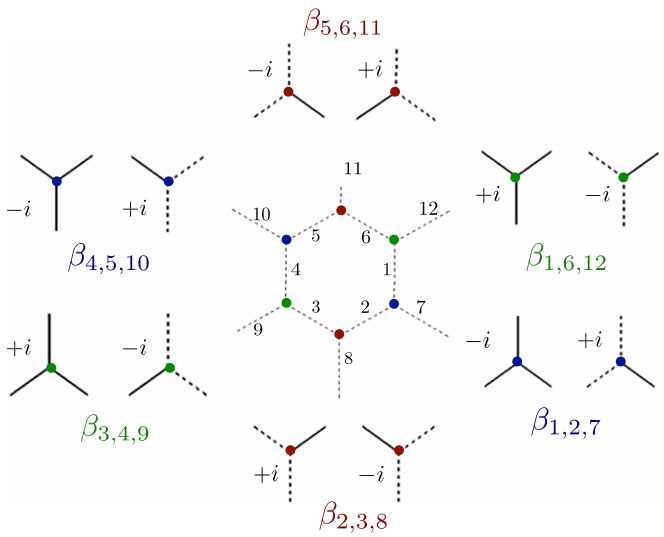

Figure 3. A graphical representation of the various non-trivial phase factors of the form $\beta_{(i,j,k)}$. A trivial phase of $+1$ is applied for any configuration not shown on this picture. The labeling of the edges have been omitted for simplicity, and corresponds to the same as the one on figure 1.

Taking into account all these caveats, we present in the following a formulation of the DS model which gives a microscopic approach to this interesting topological order, fulfilling all the necessary properties in the whole Hilbert space.

2.2. Exactly solvable model in the whole Hilbert space

If we want to consider encoding quantum information in the degenerate ground-state manifold of the standard (on-shell) DS model in equation (4), we immediately run into major problems: X Pauli errors make the state of the system leave the vertex-free subspace. The non-commutativity of the $\tilde{B}_p$ operators poses difficulties when interpreting the DS model as a stabilizer code.

To avoid such difficulties, we consider a modified version of the plaquette operators in equation (2), which we call the off-shell DS model or semion code

$$H_{\mathrm{DS}} = -\sum_{v} Q_v + \sum_{p} B_p, \tag{10}$$

where the *generalized* plaquette operator B_p is a modification of $\tilde{B}_p$ obtained by multiplying it by a phase factor that depends on the configuration on which it is applied. More specifically, we have

$$B_p \coloneqq \tilde{B}_p \times \tilde{\beta}_p, \tag{11}$$

with

$$\tilde{\beta}_p \coloneqq \sum_{\vec{i}} \tilde{b}_p(\vec{i})|\vec{i}\rangle\langle\vec{i}|, \tag{12}$$

where the sum runs over all possible configurations of edges 1 through 12 shown in figure 1. $\tilde{b}_p(\vec{i}\,)$ is the phase factor corresponding to the string configuration $\vec{i}$. $\vec{i}$ represents a state in the computational basis. A qubit in the state $|0\rangle$ is interpreted as the absence of a string on its corresponding edge, while the state $|1\rangle$ reflects the presence of a string. The phase operator β_p can be decomposed as

$$\tilde{\beta}_p = \beta_{(6,1,12)}\,\beta_{(1,2,7)}\,\beta_{(2,3,8)}\,\beta_{(3,4,9)}\,\beta_{(4,5,10)}\,\beta_{(5,6,11)}, \tag{13}$$

where $\beta_{(i,j,k)}$ is a function of the string configuration of edges i, j and k connected to vertex $v(i, j, k)$. The specific values for each factor $\beta_{(i,\,j,\,k)}$ are shown graphically in figure 3. Note that their specific form differ depending on their position on the plaquette.

For future reference, notice that the generalized plaquette operator B_p can be written as

$$B_p = \prod_{i\in\partial p} \sigma_i^x \left(\prod_{j\in\partial p} (-1)^{n_{j-1}^- n_j^+} \right) \prod_{v\in p} \beta_v, \tag{14}$$

where we use the notation $j \in \partial p$ to indicate the qubit associated to edge j in plaquette p. $v \in p$ identify the vertices belonging to plaquette p. Notice that this last expression clearly shows that the phase factor appearing in B_p is a product of phases, β_v, depending on the string configuration of the three edges connected to each vertex of

plaquette p. The complete algebraic expression of the product of all β_v in a plaquette p is [27]

$$\prod_{v\in p}\beta_v = i^{n_{12}^-(n_1^-n_6^- - n_1^+n_6^+)} i^{n_7^-(n_1^+n_2^+ - n_1^-n_2^-)} \times i^{n_8^+(n_2^-n_3^+ - n_2^+n_3^-)} i^{n_9^-(n_3^-n_4^- - n_3^+n_4^+)} \times i^{n_{10}^-(n_4^+n_5^+ - n_4^-n_5^-)} i^{n_{11}^+(n_5^-n_6^+ - n_5^+n_6^-)}. \tag{15}$$

We can easily check that in the zero-flux rule the factors in equation (15) reduce to 1, recovering expression (5) for the plaquette operators.

The crucial point now is that the new generalized plaquette operators, B_p, satisfy the desired properties needed by the stabilizer formalism of quantum error correction. Namely,

$$B_p^2 = 1,\ \ B_p^\dagger = B_p,\ \ [B_p, B_{p'}] = 0,\ \ [Q_v, B_p] = 0; \tag{16}$$

$\forall\, p,\, p',\, v \in \Lambda$, regardless of the vertex-free condition (3). The study of $H_{\rm DS}$ is rendered much simpler than that of $\tilde{H}_{\rm DS}$ on the whole Hilbert space of the qubits by the fact that the new plaquette operators commute.

3. String operators

We seek open-string operators creating excitations at their endpoints without affecting the rest of vertex and plaquette operators, as well as closed-string operators that commute with vertex and plaquette operators. In our case, excited states correspond to states in a -1 eigenstate for a vertex operator or a $+1$ eigenstate of a plaquette operator. We say that an excitation is present at vertex v (plaquette p) if the state of the system is in a -1 ($+1$) eigenstate of Q_v (B_p). Since we have that $\prod_{v\in\Lambda} Q_v = \prod_{p\in\Lambda} B_p = 1$, excitations are always created in pairs.

In order to find such string operators, it is convenient to reexpress the generalized plaquette operators as

$$B_p = \prod_{i\in p}\sigma_i^x \sum_{\vec{i}} b_p(\vec{i})|\vec{i}\rangle\langle\vec{i}|, \tag{17}$$

where p denotes the interior edges of a plaquette (edges 1 through 6 in figure 1) and the string configurations in the sum are taken on edges 1 through 12. $b_p(\vec{i})$ denotes the complex phase picked up when applying operator B_p to the configuration $\vec{i}$. Note that $\sum_{\vec{i}} b_p(\vec{i})|\vec{i}\rangle\langle\vec{i}|$ differs from the product of the β_p's in the -1 factors appearing in equation (14). $\sum_{\vec{i}} b_p(\vec{i})|\vec{i}\rangle\langle\vec{i}|$ includes the -1 factors as well as the product of β_v.

Given two string configurations $\vec{i}$ and $\vec{\alpha}$ on a set of edges, it is useful to define the string configuration $\vec{i}\,\oplus\,\vec{\alpha}$ to be the configuration $\vec{i}\,'$ where the edges occupied in configuration $\vec{\alpha}$ has been flipped. It is equivalent to sum (mod 2) the two bitstrings. Additionally, we define the configuration $\vec{\alpha}^p$ of plaquette p as the string configuration which is empty everywhere except for the six edges in the interior of plaquette p, corresponding to edges 1 through 6 in figure 1. Likewise, $\vec{\alpha}^{\mathcal{P}}$, will be the configuration which is only occupied for edges of path $\mathcal{P}$.

Given a path $\mathcal{P}$, we construct string operators $S_{\mathcal{P}}^+$ creating vertex excitations at its endpoints and commuting with every other operators in Hamiltonian (10). Negative chirality strings are defined as $S_{\mathcal{P}}^- := S_{\mathcal{P}}^+ S^z_{\mathcal{P}'_{\rm dual}}$, where $S^z_{\mathcal{P}'_{\rm dual}}$ is a product of σ^z operators forming a path $\mathcal{P}'_{\rm dual}$ in the dual lattice which is contained in the support of $S_{\mathcal{P}}^+$. If $\mathcal{P}$ is open, $S^z_{\mathcal{P}'_{\rm dual}}$ creates excitations at plaquettes containing the vertices at the endpoints of $\mathcal{P}$ and opposite to the first and last edges of $\mathcal{P}$ (p_1 and p_9 in figure 4), while if $\mathcal{P}$ is closed, $S^z_{\mathcal{P}'_{\rm dual}}$ forms a closed path in the same homological class as $\mathcal{P}$. Note that for a given path $\mathcal{P}$, various paths $\mathcal{P}'_{\rm dual}$ are possible, and each one gives rise to a different string operator $S_{\mathcal{P}}^-$.

3.1. An algorithm to generate string operators

In order to find these string operators, we consider the following ansatz:

$$S_{\mathcal{P}}^+ := \prod_{i\in\mathcal{P}}\sigma_i^x \sum_{\vec{i}\in{\rm Conn}(\mathcal{P})} F_{\mathcal{P}}(\vec{i})|\vec{i}\rangle\langle\vec{i}|, \tag{18}$$

where $F_{\mathcal{P}}(\vec{i})$ is a phase factor acquired when $S_{\mathcal{P}}^+$ is applied on configuration $\vec{i}$. $F_{\mathcal{P}}(\vec{i})$ only depends on qubits belonging to ${\rm Conn}(\mathcal{P})$, defined as:

$${\rm Conn}(\mathcal{P}) := \{\text{links of } \mathcal{P} \text{ and its external legs}\}. \tag{19}$$

It is also useful to define the set of plaquettes

$$\mathcal{B}_{\mathcal{P}} := \{p\colon \partial p \cap {\rm Conn}(\mathcal{P}) \neq \varnothing\}, \tag{20}$$

which is the set of plaquettes that have at least one of their interior edges contained in ${\rm Conn}(\mathcal{P})$. Equivalently, one can define $\mathcal{B}_{\mathcal{P}}$ to be the set of plaquettes such that for at least one string configuration $\vec{i}$, $b_p(\vec{i}\,\oplus\,\vec{\alpha}^{\mathcal{P}}) \neq b_p(\vec{i})$. The structure of $S_{\mathcal{P}}^+$ is illustrated in figure 4. Note that depending on the context, a configuration $\vec{i}$ is

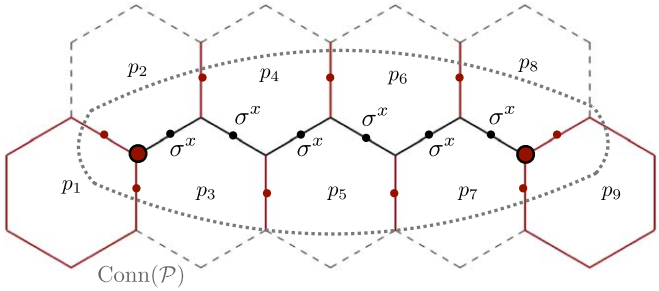

Figure 4. The structure of $S^+_{\mathcal{P}}$, where the path $\mathcal{P}$ is indicated by the full black edges where the σ^x operators are applied. The phase factors $F^+_{\mathcal{P}}$ depend on the configuration of all the edges identified with a dot, which are collectively denoted by Conn($\mathcal{P}$). The effect of $S^+_{\mathcal{P}}$ on the ground state of the system is to create a pair of vertex excitations at the vertices located at the endpoints of the path $\mathcal{P}$, which are identified by big red dots. Plaquettes p_1 to p_9 constitute the set $\mathcal{B}_{\mathcal{P}}$.

either understood to be on the full system, or is the configuration restricted to Conn($\mathcal{P}$). The specific case considered is explicitly stated in each case.

Ansatz (18) should satisfy the following properties.

(i) Anticommutes with vertex operators at the endpoints of $\mathcal{P}$ if it is open, while it commutes with every other vertex and plaquette operators.

(ii) Acts trivially on edges outside Conn($\mathcal{P}$).

Operators satisfying (i) and (ii) are called *string operators*. Additionally, we may be interested in the properties:

Squares to the identity.

Hermitian.

If these are satisfied, they will be called *canonical* string operators.

Properties (iii) and (iv) are satisfied (see lemmas 5 and 6 in appendix B) if

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*. \tag{21}$$

Since we want $S^+_{\mathcal{P}}$ to commute with all plaquettes in $\mathcal{B}_{\mathcal{P}}$ (for the rest of plaquettes, it commutes by construction), we impose that the commutator vanishes, $[S^+_{\mathcal{P}}, B_p] = 0$, which yields the equation

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p}) = \frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i})} F_{\mathcal{P}}(\vec{i}). \tag{22}$$

It is useful to define the function $\theta_{\mathcal{P}}(\vec{i}, p) \coloneqq b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})/b_p(\vec{i})$, which relates $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p})$ to $F_{\mathcal{P}}(\vec{i})$. We can generalize equation (22) and function $\theta_{\mathcal{P}}(\vec{i}, p)$ for an arbitrary number of plaquettes, namely

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}) = \theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) F_{\mathcal{P}}(\vec{i}), \tag{23}$$

where $\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m)$ can be expressed as

$$\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) = \prod_{i=1}^{m} \frac{b_{p_i}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{p_j}\right)}{b_{p_i}\left(\vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{p_j}\right)}. \tag{24}$$

Note that while we use the same symbol for the configurations $\vec{i}$ in $\theta_{\mathcal{P}}$ and in $F_{\mathcal{P}}$, the one in $\theta_{\mathcal{P}}$ is over the whole system in order for equation (24) to be well-defined. The specific way that the configuration $\vec{i}$ is extended over the whole system (i.e. which configuration on the rest of the system is appended to it) does not matter, since as a consequence of the structure of the plaquette operators, it does not affect the value of $\theta_{\mathcal{P}}$. As a consequence of the fact that plaquette operators commute, the order of the plaquettes $p_1, \ldots, p_m$ in $\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m)$ does not matter (see lemma 1). The function $\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m)$ relates the value of $F_{\mathcal{P}}$ for configuration $\vec{i}$ to that of configuration $\vec{i} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}$. These two configurations differ by a sum of plaquettes and may be considered part of the same configuration class $\mathcal{C}_{\mathcal{P}}(\vec{i})$, defined as

$$\mathcal{C}_{\mathcal{P}}(\vec{i}) \coloneqq \left\{ \vec{j} : \vec{j} = \vec{i} \bigoplus_{p \in \text{subset}(\mathcal{B}_{\mathcal{P}})} \vec{\alpha}^{p} \right\}, \tag{25}$$

where the configurations are restricted to $\text{Conn}(\mathcal{P})$. These configurations can be regarded as the set of configurations related to $\vec{i}$ by adding loops associated with plaquettes in $\mathcal{B}_{\mathcal{P}}$, only in the region where $S^{+}_{\mathcal{P}}$ acts non-trivially. Taking all this into account, one can obtain an algorithm to compute the phases $F^{+}_{\mathcal{P}}$ of the ansatz in equation (18). This is given by algorithm 1.

Algorithm 1. Determination of the $F_{\mathcal{P}}$ functions for a path $\mathcal{P}$. $e^{i\phi(\vec{i})}$ are initial phases that can take any value.

for every configuration class $\mathcal{C}_{\mathcal{P}}$ **do**
  Pick a class representative $\vec{i} \in \mathcal{C}_{\mathcal{P}}$
  Set $F_{\mathcal{P}}(\vec{i}) = e^{i\phi(\vec{i})}$
  for every subset $(p_1, \ldots, p_m) \subseteq \mathcal{B}_{\mathcal{P}}$ **do**
    $F_{\mathcal{P}}(\vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i}) = \theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) F_{\mathcal{P}}(\vec{i})$
  end for
end for

Algorithm 1 begins by picking up a configuration $\vec{i}$, which we call its class representative, and by setting its value to an arbitrary phase $e^{i\phi(\vec{i})}$. Any phase picked up by the algorithm yields a valid string operator. Once the value for the class representative $\vec{i}$ is fixed, the algorithm assigns values to the rest of the configurations in the same configuration class by making use of equation (23). Afterwards, a configuration, belonging to a different configuration class, where the values have not yet been fixed, is chosen and the same procedure is repeated until $F_{\mathcal{P}}$ has been fixed for all possible configurations. An explicit example of this can be seen in appendix A.

As it is shown in appendix B, it is always possible to determine $F_{\mathcal{P}}$ using algorithm 1 such that the resulting $S^{+}_{\mathcal{P}}$ is a string operator. Furthermore, it is also possible to enforce the constraint given in equation (21) such that we obtain canonical string operators. Those important results are summarized in the following theorem:

Theorem 1. *Let $\mathcal{P}$ be a path. Any function $F_{\mathcal{P}}$ defined by algorithm 1 is such that $S^{+}_{\mathcal{P}}$ is a string operator. Furthermore, it is possible to choose the phases $e^{i\phi}$ such that the string operator is canonical.*

Note that the open-string operators generated by algorithm 1, $S^{+}_{\mathcal{P}}$, have positive chirality, because they anticommute with vertex operators at the endpoints of $\mathcal{P}$ and commute with every other vertex and plaquette operators, satisfying property (i). However, the same does not apply to closed-string operators, since closed strings do not have endpoints. Algorithm 1 produces, in general, closed-string operators without a definite chirality.

3.1.1. Concatenation of open-string operators

It is very useful to build strings as a concatenation of smaller strings. This is specially relevant for constructing non-trivial closed strings, since, as it was mentioned before, algorithm 1 yields, in general, closed strings which have no definite chirality. By building these closed strings out of a multiplication of open strings, which have definite chirality, we obtain positive- and negative-chirality closed strings. Observe that given two paths, $\mathcal{P}_1$ and $\mathcal{P}_2$, meeting at one endpoint or forming a closed path, the multiplication of both $S^{+}_{\mathcal{P}_1 \# \mathcal{P}_2} = S^{+}_{\mathcal{P}_2} S^{+}_{\mathcal{P}_1}$ is a string operator satisfying properties (i) and (ii). In this way we can build long string operators by concatenating short ones.

3.2. Crossing string operators

In order to understand the algebra of the string operators of the semion code that we build in section 4.1, it is essential to know the commutation relations between them acting on different paths.

The notion of crossing paths need to be precisely defined since the region on which the string operators act non-trivially, $\text{Conn}(\mathcal{P})$, has a finite thickness. Heuristically, in order to consider that two paths are crossing, the commun edges to both paths must not contain the first nor last vertex of neither of the paths. Note that two paths can cross more than once.

We further need to define the notions of *self-crossing* and *self-overlapping* paths. Essentially, a path is self-crossing if an observer moving on the path passes more than once on any given edge. A path is said to be *self-overlapping* if some regions of the support of the string operator but not the path itself overlap and connect some distant parts of the paths. Figure 5 illustrates the previous concepts. We refer the reader to appendix C for rigorous definitions, as well as the proof of theorem 2. This theorem summarizes the commutation relations among the string operators.

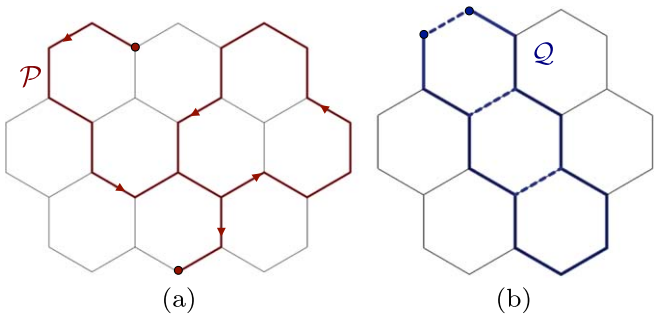

Figure 5. Examples of (a) a self-crossing path $\mathcal{P}$ and of (b) a self-overlapping path $\mathcal{Q}$. The dashed blue edges indicate the links where the string $\mathcal{Q}$ self-overlaps, since some distant parts of the path are connected.

Theorem 2. *Let $\mathcal{P}$ and $\mathcal{Q}$ be two paths crossing n times, composed of non self-overlapping nor self-crossing individual open paths, i.e. $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$ and $\mathcal{Q} = \mathcal{Q}_1\#...\#\mathcal{Q}_{m'}$ for some integers m and m′. We have that*

$$[S^+_{\mathcal{P}}, S^+_{\mathcal{Q}}] = 0 \text{ if} n \text{ is even},$$
$$\{S^+_{\mathcal{P}}, S^+_{\mathcal{Q}}\} = 0 \text{ if} n \text{ is odd}. \tag{26}$$

Given the definition of negative chirality string operators, and using the above results, we find that $[S^-_{\mathcal{P}}, S^-_{\mathcal{Q}}] = 0$ if n is even, while $\{S^-_{\mathcal{P}}, S^-_{\mathcal{Q}}\} = 0$ if n is odd. We also find $[S^\pm_{\mathcal{P}}, S^\mp_{\mathcal{Q}}] = 0$ for any n.

Interpreting a vertex (in the case of $S^+_{\mathcal{P}}$) or the combination of a vertex and plaquette excitations (in the case of $S^-_{\mathcal{P}}$) as the presence of a quasiparticle labeled by s^+ and s^- respectively, the topological S matrix [35], written in the basis $(1, s^+, s^-, s^+s^-)$ where 1 denotes the vacuum, i.e. the absence of excitation, and s^+s^- the composite object excitation, is found to be

$$S = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \tag{27}$$

We can thus interpret the string operators $S^+_{\mathcal{P}}$ and $S^-_{\mathcal{P}}$ as creating pairs of semions of different chirality at their endpoints.

3.2.1. The need for path concatenation

Notice that theorem 2 does not state anything about closed paths (homologically trivial or not) which are composed of a single path. One can check that when such a path crosses another one, in general they do not commute nor anti-commute. Such paths thus cannot be considered as 'fundamental' string operators in the sense that they do not possess a definite chirality.

Algorithm 1 enforces that the $\sum_{\vec{i}} F_{\mathcal{P}}(\vec{i})|\vec{i}\rangle\langle\vec{i}|$ operator does not contain any open $S^z_{\mathcal{P}'_{\text{dual}}}$ operator for an open path, since by construction, Algortihm 1 builds a string operator which commutes with every plaquette operator. For a closed path however, one can add $S^z_{\mathcal{P}'_{\text{dual}}}$ to $\sum_{\vec{i}} F_{\mathcal{P}}(\vec{i})|\vec{i}\rangle\langle\vec{i}| \rightarrow (\sum_{\vec{i}} F_{\mathcal{P}}(\vec{i})|\vec{i}\rangle\langle\vec{i}|)S^z_{\mathcal{P}'_{\text{dual}}}$, which is also a valid output of algorithm 1. In fact, such a $S^z_{\mathcal{P}'_{\text{dual}}}$ operator can be added selectively to only a subset of configuration classes, causing a 'mixing' of the chiralities. This is explained in detail in appendix C.3.

By producing closed paths starting from smaller open paths as their basic constituents, one can enforce the production of strings of a definite chirality. This is caused by the fact that the individual components cannot carry flux excitations by construction, and so neither can their concatenation. The physical intuition is that each small open string operator creates a pair of semions of positive chirality at their endpoints. Since the created semions are their own anti-particles, they subsequently all fuse to the vacuum, returning the system to the ground space.

3.3. Completeness of the string operators

In this section, we seek to decompose a string of σ^x operators into the strings operators defined in our model, S^+ and S^z.

We first note that any matrix ρ of size $2^n \times 2^n$ can be written as a linear combination of Pauli operators, i.e.

$$\rho = \sum_{P_x, P_z} c(P_x, P_z) P_x P_z, \tag{28}$$

where P_x (P_z) are Pauli operators acting on n qubits and formed of products of identities and σ^x (σ^z) operators only and where $c(P_x, P_z)$ are non-zero complex numbers. Given the matrix ρ, one can recover the coefficients $c(P_x, P_z)$ using the formula

$$c(P_x, P_z) = \frac{1}{2^n}\mathrm{Tr}(P_x P_z \rho). \tag{29}$$

In our case, a chain of σ^x operators on path $\mathcal{P}$, denoted by $X_\mathcal{P}$, can be written as

$$X_\mathcal{P} = S_\mathcal{P}^+ \times \sum_{\vec{i}} [F_\mathcal{P}(\vec{i})]^* |\vec{i}\rangle\langle\vec{i}|. \tag{30}$$

Given the form of $\sum_{\vec{i}} [F_\mathcal{P}(\vec{i})]^* |\vec{i}\rangle\langle\vec{i}|$, we can write

$$X_\mathcal{P} = S_\mathcal{P}^+ \times \sum_{P_z \in \mathrm{Conn}(\mathcal{P})} c(P_z) P_z, \tag{31}$$

where P_z are Pauli operators containing only identities and σ^z and where we write $P_z \in \mathrm{Conn}(\mathcal{P})$ in an abuse of notation to signify that P_z acts non-trivially only on the qubits in $\mathrm{Conn}(\mathcal{P})$. The coefficients $c(P_z)$ are given by

$$c(P_z) = \frac{1}{2^{|\mathrm{Conn}(\mathcal{P})|}} \sum_{\vec{i}} \langle\vec{i}|P_z|\vec{i}\rangle [F_\mathcal{P}(\vec{i})]^*, \tag{32}$$

where $|\mathrm{Conn}(\mathcal{P})|$ is the number of edges in $\mathrm{Conn}(\mathcal{P})$. Chains of σ^z operators form valid string operators S^z, which create flux excitations at their endpoints. This clearly shows that any Pauli operator acting on the system can be written in terms of string operators. Since any operator acting on the system can be decomposed as a linear combination of Pauli operators, we find that the string operators are complete in the sense that any operator can be expressed in terms of them.

4. The semion code

The tools we have developed in previous sections can be used to build a quantum error correction code using as code space the ground space of the off-shell DS model, given by Hamiltonian (4). The information encoded is topologically protected since we are using global degrees of freedom that cannot be affected by local errors. Additionally, we perform quantum error correction using the DS model as a stabilizer code, where plaquette and vertex operators are our stabilizers.

4.1. Logical operators

Recalling that a surface of genus g can be seen as the connected sum of g tori [37], we can define two pairs of anti-commutating logical operators for every torus in the connected sum [38]. One pair consists of string operators S_V^+ and S_H^+, with $V(H)$ any homologically non-trivial path along the vertical (horizontal) direction,while the other pair consists of S_V^- and S_H^-, Both pairs are made up of open non self-crossing nor self-overlapping individual paths, as prescribed by theorem 2. Figure 6 illustrates two such pairs for a genus 1 torus.

4.2. Quantum error correction

The stabilizer operators, vertices and plaquette defined in equations (1) and (11), can be periodically measured to detect any errors occurring in the system. Once the syndrome pattern is obtained, assuming a given noise model, it is fed into a decoder. It outputs a recovery operation using the string operators developed in this work in order to bring the system back to the encoded subspace, where the probability of applying a non-trivial logical operation is minimized. While we leave the development of decoders specifically designed for the semion code for future work, one could imagine adapting some of the various existing decoders developed for topological codes [2, 39–61].

Table 1 shows the probabilities of measuring a given flux configurations after applying a single σ^x on the ground state for the three possible edge orientations shown in figure 7. Note that as equations (30) and (31) suggest, the probabilities in table 1 do not depend of the phases used to initialize the $F_{\{e\}}$ function in algorithm 1. More details giving a deeper understanding on the structure of the string operators $S_\mathcal{P}^+$ can be found in appendix B.3.

A distinctive feature of the probability distributions in table 1 is that there is a directionality in the error pattern. A σ^x error affecting a vertical edge (orientation (b)) is much more likely to leave flux excitations behind than for the other two orientations. This is clearly due to the specific structure of the plaquette operators, and could be used advantageously when dealing with asymmetric noise [62, 63]. Another major difference with the toric code is the fact that chains of σ^x errors are likely to leave flux excitations along their path. This additional information could be used by the decoder and may lead to a higher threshold value.

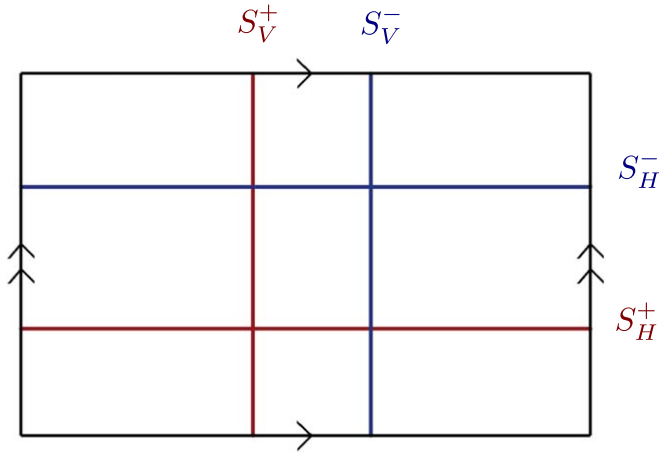

Figure 6. A cartoon example of 2 sets of logical operators on a 1-torus. $\{S_V^+, S_H^+\} = \{S_V^-, S_H^-\} = 0$, while $[S_V^+, S_H^-] = [S_V^-, S_H^+] = 0$. Arrows indicate identified boundaries.

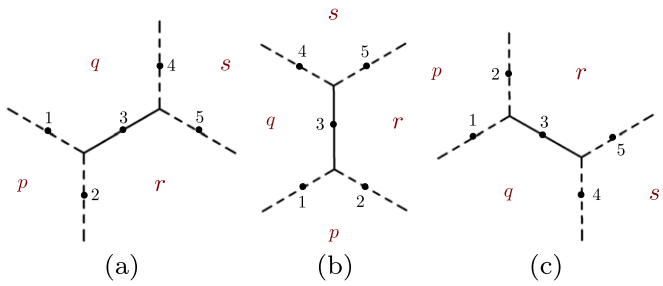

Figure 7. The three possible edge orientations on which the σ^x operator can be applied. Qubit 3 is affected in all cases, and may leave flux excitations on the four surrounding plaquettes labeled by p, q, r and s. The probabilities of measuring a given flux pattern are given in table 1.

Table 1. The various probabilities of getting a given flux excitation configuration after the application of the operator σ^x on a qubit, for the three possible orientations. The plaquettes label correspond to the ones in figure 7.

	Probability		
(b_p, b_q, b_r, b_s)	Orientation (a)	Orientation (b)	Orientation (c)
(0, 0, 0, 0)	9/16	1/16	9/16
(1, 1, 0, 0)	1/16	1/16	1/16
(1, 0, 1, 0)	1/16	1/16	1/16
(0, 1, 1, 0)	1/16	9/16	1/16
(1, 0, 0, 1)	1/16	1/16	1/16
(0, 1, 0, 1)	1/16	1/16	1/16
(0, 0, 1, 1)	1/16	1/16	1/16
(1, 1, 1, 1)	1/16	1/16	1/16

5. Conclusions and outlook

One of the key features of the off-shell DS model developed here for error correction is that it is a non-CSS code [64, 65]. This is not novel in the theory of quantum error correcting codes. In fact, the answer to the important question of what is the minimal complete error correction code that is able to encode one logical qubit and correct for an arbitrary error was precisely a non-CSS code of five qubits [64, 65]. This is consistent with the quantum Hamming bound [32], and is in sharp contrast with the classical case where the solution is the repetition code of three bits. However, what is peculiar of the off-shell DS code is that, to our knowledge, this is

the first non-CSS topological quantum memory that being a stabilizer code, it is also a topological code throughout the whole Hilbert space. In a sense, this was a missing link in the theory of topological quantum error correction codes and we have filled this gap with the tools introduced in our work.

We notice that a previous study [33] attempted to construct a quantum error correction code using the DS model as the starting point. The main difference with our work is that they construct a non-commuting quantum correcting code, whereas we have succeeded in constructing an extension that belongs to the stabilizer formalism. As a consequence of this, the whole error correction procedure of the off-shell DS code is topological. On the contrary, the topological nature of the non-commuting code in [33] is unproven. Both constructions share the feature of using non-Pauli operators to construct the basic string operators of the model.

The outcome of our work is a complete characterization of the error correction procedure for a quantum memory based on a topological non-CSS stabilizer code. This is a major step for the reason explained above. However, a fully-fledged quantum computer will demand more, namely, a universal gate set and a fault-tolerant procedure to battle errors dynamically [66]. With the tools deployed here, it is conceivable that this goal will be achieved elsewhere.

A new way of constructing quantum codes opens up with this work. The tools introduced here for models like the DS based on Abelian lattice gauge theories can be generalized to other Levin–Wen models [16, 67, 68], like doubled Fibonacci models, or twisted versions of fracton models [69]. This is the subject of further study.

Acknowledgments

We thank Fiona Burnell and Juan Miguel Nieto for helpful discussions. We acknowledge financial support from the Spanish MINECO grants FIS2012-33152, FIS2015-67411, and the CAM research consortium QUITEMAD +, Grant No. S2013/ICE-2801. The research of MAM-D has been supported in part by the US Army Research Office through Grant No. W911N F-14-1-0103. SV thanks FPU MECD Grant.

Appendix A. Example of an open-string operator

It is instructive to illustrate the workings of algorithm 1 to find string operators $S^+_{\mathcal{P}}$ in order to gain a more intuitive understanding. Consider the very simple path $\mathcal{P} = \{e\}$, consisting only of edge e shown in figure A1. Any given configuration on the five edges included in Conn($\{e\}$) is represented by a bit string of length 5, for which a 0 indicates the absence of a string, while a 1 indicates that it is occupied. One can also interpret the bit string as a state in the computational basis, a 0 indicating a $+1$ eigenstate of the corresponding σ^z, while a 1 indicates a -1 eigenstate of σ^z. We compute the function $F^+_{\{e\}}$ so that $S^+_{\{e\}}$ is a canonical string operator, i.e. it also fulfills equation (21). Following algorithm 1, the configuration (0, 0, 0, 0, 0) is first chosen and we set $F^+_{\{e\}}(0,\ 0,\ 0,\ 0,\ 0) = 1$. Noting that $\mathcal{B}_{\{e\}}$ contains the 4 plaquettes identified as p, q, r and s in figure A1, we find the following values for $F^+_{\{e\}}$:

$$
\begin{aligned}
&F_{\{e\}}(0,\ 0,\ 0,\ 0,\ 0) = 1, \quad F_{\{e\}}(1,\ 1,\ 0,\ 0,\ 0) = -i,\\
&F_{\{e\}}(1,\ 0,\ 1,\ 1,\ 0) = -i, \quad F_{\{e\}}(0,\ 0,\ 0,\ 1,\ 1) = -i,\\
&F_{\{e\}}(0,\ 0,\ 1,\ 0,\ 0) = 1, \quad F_{\{e\}}(1,\ 1,\ 1,\ 0,\ 0) = i,\\
&F_{\{e\}}(1,\ 0,\ 0,\ 1,\ 0) = i, \quad F_{\{e\}}(0,\ 0,\ 1,\ 1,\ 1) = i,\\
&F_{\{e\}}(0,\ 1,\ 1,\ 0,\ 1) = -i, \quad F_{\{e\}}(1,\ 1,\ 1,\ 1,\ 1) = -1,\\
&F_{\{e\}}(0,\ 1,\ 1,\ 1,\ 0) = -i, \quad F_{\{e\}}(1,\ 0,\ 0,\ 0,\ 1) = i,\\
&F_{\{e\}}(0,\ 1,\ 0,\ 0,\ 1) = i, \quad F_{\{e\}}(1,\ 1,\ 0,\ 1,\ 1) = -1,\\
&F_{\{e\}}(0,\ 1,\ 0,\ 1,\ 0) = i, \quad F_{\{e\}}(1,\ 0,\ 1,\ 0,\ 1) = -i.
\end{aligned}
$$

Here we are not only computing the values for configuration class $\mathcal{C}_{\{e\}}(\vec{0})$, but also for configuration class $\mathcal{C}_{\{e\}}(\vec{\alpha}^{\{e\}})$, since these two are related by equation (21)and therefore algorithm 1 makes the assignment $F_{\{e\}}(i_1,\ i_2,\ i_3 \oplus 1,\ i_4,\ i_5) = [F_{\{e\}}(i_1,\ i_2,\ i_3,\ i_4,\ i_5)]^*$.

Choosing next the configuration (0, 0, 0, 0, 1) and setting $F_{\{e\}}(0,\ 0,\ 0,\ 0,\ 1) = 1$, we can fix the following values of $F_{\{e\}}$:

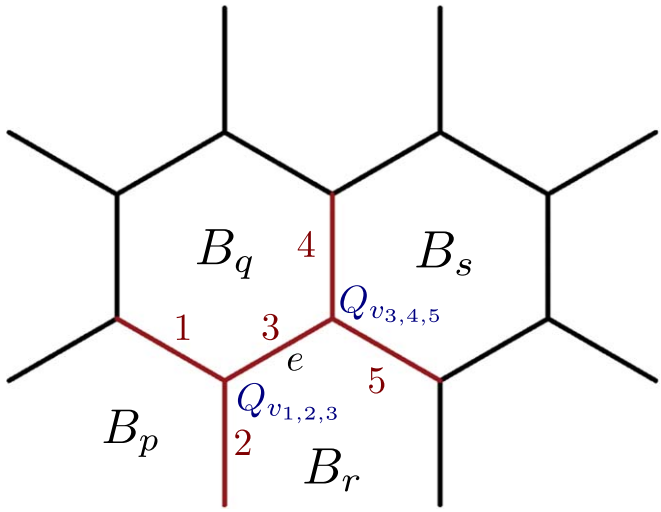

Figure A1. The layout we consider to find a possible string operator S_e^+ creating a pair of vertex excitations at $v_{1,2,3}$ and $v_{3,4,5}$. The red edges correspond to the qubits in Conn(e), the ones on which S_e^+ acts non-trivially, with edge e corresponding to qubit 3. S_e^+ anti commutes with both vertex operators $Q_{v_{1,2,3}} = \sigma_1^z\sigma_2^z\sigma_3^z$ and $Q_{v_{3,4,5}} = \sigma_3^z\sigma_4^z\sigma_5^z$, while commuting with every other vertex operators as well as every generalized plaquette operators.

$$
\begin{aligned}
&F_{\{e\}}(0,0,0,0,1) = 1, \quad F_{\{e\}}(0,0,0,1,0) = 1,\\
&F_{\{e\}}(1,1,0,0,1) = -i, \quad F_{\{e\}}(1,1,0,1,0) = -i,\\
&F_{\{e\}}(0,0,1,0,1) = 1, \quad F_{\{e\}}(0,0,1,1,0) = 1,\\
&F_{\{e\}}(1,1,1,0,1) = i, \quad F_{\{e\}}(1,1,1,1,0) = i,\\
&F_{\{e\}}(1,0,1,1,1) = -1, \quad F_{\{e\}}(0,1,1,1,1) = -1,\\
&F_{\{e\}}(0,1,1,0,0) = i, \quad F_{\{e\}}(1,0,1,0,0) = i,\\
&F_{\{e\}}(1,0,0,1,1) = -1, \quad F_{\{e\}}(0,1,0,1,1) = -1,\\
&F_{\{e\}}(0,1,0,0,0) = -i, \quad F_{\{e\}}(1,0,0,0,0) = -i.
\end{aligned}
$$

Again, two class of configurations are fixed due to the fact that the string is canonical. We keep doing this till all configuration classes have been fixed. As a result, we obtain $S_{\{e\}}^+$, which commutes with the four neighboring plaquette operators B_p, B_q, B_r and B_s shown in figure A1, as well as all the other plaquette operators which are farther away.

Appendix B. Proofs regarding string operators produced by algorithm 1

In order to prove theorem 1, we begin by stating several technical results in the following section.

B.1. Useful technical lemmas

Lemma 1. *Let* $\{p_1, p_2, \ldots, p_n\}$ *an ordered set of plaquettes and let* $\{q_1, q_2, \ldots, q_n\}$ *be a permutation of it. For any configuration* $\vec{i}$, *we have that* $\theta_{\mathcal{P}}(\vec{i}, p_1, p_2, \ldots, p_n) = \theta_{\mathcal{P}}(\vec{i}, q_1, q_2, \ldots, q_n)$.

Proof. According to equation (24), we have that

$$
\begin{aligned}
\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) &= \frac{\left\langle \vec{i}\bigoplus_{i=1}^m \vec{\alpha}^{p_i} \oplus \vec{\alpha}^{\mathcal{P}} | B_{p_m} \ldots B_{p_1} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i}\bigoplus_{i=1}^m \vec{\alpha}^{p_i} | B_{p_m} \ldots B_{p_1} | \vec{i} \right\rangle}\\
&= \frac{\left\langle \vec{i}\bigoplus_{i=1}^m \vec{\alpha}^{q_i} \oplus \vec{\alpha}^{\mathcal{P}} | B_{q(p_m)} \ldots B_{q(p_1)} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i}\bigoplus_{i=1}^m \vec{\alpha}^{q_i} | B_{q(p_m)} \ldots B_{q(p_1)} | \vec{i} \right\rangle} = \theta_{\mathcal{P}}(q_1, \ldots, q_m),
\end{aligned} \tag{B1}
$$

where $q(p_i)$ denotes the permutation of plaquettes that exchange $\{p_1, \ldots, p_m\}$ to $\{q_1, \ldots, q_m\}$, and where we used the fact that the plaquette operators all commute. □

Lemma 2. *Let* $\{p_1, p_2, \ldots, p_m\}$ *and* $\{q_1, q_2, \ldots, q_k\}$ *be two different set of plaquettes in* $\mathcal{B}_{\mathcal{P}}$ *such that* $\bigoplus_{i=1}^m \vec{\alpha}^{p_i} = \bigoplus_{i=1}^k \vec{\alpha}^{q_i}$ *on the string configuration of* Conn($\mathcal{P}$). *Then, for any configuration* $\vec{i}$, *we find that*

$$\theta_{\mathcal{P}}(\vec{i}, p_1, p_2, \ldots, p_m) = \theta_{\mathcal{P}}(\vec{i}, q_1, q_2, \ldots, q_k). \tag{B2}$$

Proof. First notice that given the structure of the plaquette operators, $\bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} = \bigoplus_{i=1}^{k} \vec{\alpha}^{q_i}$ on $\mathrm{Conn}(\mathcal{P})$ implies that $\prod_{i=1}^{m} B_{p_i} \prod_{j=1}^{k} B_{q_j} \prod_{l=1}^{s} B_{r_l} = 1$, where $\{r_1, \ldots, r_s\}$ are all the plaquettes outside of $\mathcal{B}_{\mathcal{P}}$. We also have that $\bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} = (\bigoplus_{j=1}^{k} \vec{\alpha}^{q_j}) \oplus (\bigoplus_{l=1}^{s} \vec{\alpha}^{r_l})$, where the configurations are taken over the whole system. Using those facts, we find

$$\begin{aligned}
\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) &= \frac{\left\langle \vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} \oplus \vec{\alpha}^{\mathcal{P}} | B_{p_m} \ldots B_{p_1} \prod_{i=1}^{m} B_{p_i} \prod_{j=1}^{k} B_{q_j} \prod_{l=1}^{s} B_{r_l} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} | B_{p_m} \ldots B_{p_1} \prod_{i=1}^{m} B_{p_i} \prod_{j=1}^{k} B_{q_j} \prod_{l=1}^{s} B_{r_l} | \vec{i} \right\rangle} \\
&= \frac{\left\langle \vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} \oplus \vec{\alpha}^{\mathcal{P}} | \prod_{j=1}^{k} B_{q_j} \prod_{l=1}^{s} B_{r_l} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} | \prod_{j=1}^{k} B_{q_j} \prod_{l=1}^{s} B_{r_l} | \vec{i} \right\rangle} \\
&= \frac{\left\langle \vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} \oplus \vec{\alpha}^{\mathcal{P}} | \prod_{l=1}^{s} B_{r_l} \left(\sum_{\vec{j}} |\vec{j}\rangle\langle\vec{j}| \right) \prod_{j=1}^{k} B_{q_j} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i} \bigoplus_{i=1}^{m} \vec{\alpha}^{p_i} | \prod_{l=1}^{s} B_{r_l} \left(\sum_{\vec{j}} |\vec{j}\rangle\langle\vec{j}| \right) \prod_{j=1}^{k} B_{q_j} | \vec{i} \right\rangle} \\
&= \frac{\left\langle \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \oplus (\bigoplus_{l=1}^{s} \vec{\alpha}^{r_l}) \oplus \vec{\alpha}^{\mathcal{P}} | \prod_{l=1}^{s} B_{r_l} | \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \oplus (\bigoplus_{l=1}^{s} \vec{\alpha}^{r_l}) | \prod_{l=1}^{s} B_{r_l} | \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \right\rangle} \\
&\quad \times \frac{\left\langle \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \oplus \vec{\alpha}^{\mathcal{P}} | \prod_{j=1}^{k} B_{q_j} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} | \prod_{j=1}^{k} B_{q_j} | \vec{i} \right\rangle} \\
&= \frac{\left\langle \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \oplus (\bigoplus_{l=1}^{s} \vec{\alpha}^{r_l}) | X_{\mathcal{P}} \prod_{l=1}^{s} B_{r_l} X_{\mathcal{P}} | \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \oplus (\bigoplus_{l=1}^{s} \vec{\alpha}^{r_l}) | \prod_{l=1}^{s} B_{r_l} | \vec{i} \bigoplus_{j=1}^{k} \vec{\alpha}^{q_j} \right\rangle} \theta_{\mathcal{P}}(\vec{i}, q_1, \ldots, q_k), \\
&= \theta_{\mathcal{P}}(\vec{i}, q_1, \ldots, q_k),
\end{aligned} \tag{B3}$$

where we used the fact that $X_{\mathcal{P}}$ commutes with the plaquette operators B_{r_l}, since the plaquettes in $\{r_1, \ldots, r_s\}$ are not in $\mathrm{Conn}(\mathcal{P})$. $X_{\mathcal{P}}$ is the string of σ_x corresponding to the string operator defined on $\mathcal{P}$. □

Lemma 3. *The functions $F_{\mathcal{P}}$ constructed by algorithm 1 are well-defined.*

Proof. First note that lemma 1 states that the order in which the plaquettes $\{p_1, \ldots, p_n\}$ appear in a specific subset of $\mathcal{B}_{\mathcal{P}}$ and the order into which the subsets are chosen do not affect its value.

Next, we show that if there are two different sets of plaquettes $\{p_1, \ldots, p_m\} \subset \mathcal{B}_{\mathcal{P}}$ and $\{q_1, \ldots, q_k\} \subset \mathcal{B}_{\mathcal{P}}$ such that $\vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m} = \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_k}$, where it is understood that the configurations are equal on $\mathrm{Conn}(\mathcal{P})$ (as opposed to the whole lattice), then algorithm 1 ensures that $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_n}) = F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_m})$, for any configurations $\vec{i}$ (and for configuration $\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}$ as well). This is a simple consequence of lemma B2, which tells us that $\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) = \theta_{\mathcal{P}}(\vec{i}, q_1, \ldots, q_k)$, and of the way that the $F_{\mathcal{P}}$ functions are built;

$$\begin{aligned}
F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}) &= \theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) F_{\mathcal{P}}(\vec{i}) \\
&= \theta_{\mathcal{P}}(\vec{i}, q_1, \ldots, q_k) F_{\mathcal{P}}(\vec{i}) \\
&= F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_k}).
\end{aligned}$$

□

Lemma 4. *Let $F_{\mathcal{P}}$ be a function determined by algorithm 1. Then $F_{\mathcal{P}}$ simultaneously satisfies all the constraints (22).*

Proof. Consider an arbitrary configuration $\vec{i}$ for which the value of $F_{\mathcal{P}}(\vec{i})$ has been determined using algorithm 1. Two possible cases need to be analyzed:

1. If $\vec{i}$ is one of the configuration picked to set an unknown value of $F_{\mathcal{P}}(\vec{i})$, then we find that for any plaquette p,

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p}) = \frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i})} F_{\mathcal{P}}(\vec{i}), \tag{B4}$$

by definition of $\theta_{\mathcal{P}}$.

2. If $\vec{i}$ is not one of the configurations picked, then there is another configuration $\vec{i}\,'$ which was picked as a class representative and a set of plaquettes $\{p_1, \ldots, p_m\} \subset \mathcal{B}_{\mathcal{P}}$ such that $\vec{i} = \vec{i}\,' \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}$. For any plaquette p, according to algorithm 1, we find that

$$\frac{F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p})}{F_{\mathcal{P}}(\vec{i})} = \frac{\theta_{\mathcal{P}}(\vec{i}\,', p_1, \ldots, p_m, p)}{\theta_{\mathcal{P}}(\vec{i}\,', p_1, \ldots, p_m)}. \tag{B5}$$

By using definition (24), we find

$$\frac{\theta_{\mathcal{P}}(\vec{i}\,', p_1, \ldots, p_m, p)}{\theta_{\mathcal{P}}(\vec{i}\,', p_1, \ldots, p_m)} = \frac{b_p(\vec{i}\,' \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i}\,' \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m})} = \frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i})}.$$

We thus have that in both cases, all the constraints (22) are satisfied. □

Lemma 5. *(Canonical strings)* $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$ *for any string configuration* $\vec{i}$ *if and only if* $[S_{\mathcal{P}}^{+}]^2 = 1$.

Proof. Explicit calculation of $[S_{\mathcal{P}}^{+}]^2$ gives

$$\begin{aligned}
[S_{\mathcal{P}}^{+}]^2 &= \left(\prod_{i \in \mathcal{P}} \sigma_i^x \sum_{\text{String conf.}\vec{i}} F_{\mathcal{P}}(\vec{i}) |\vec{i}\rangle \langle \vec{i}| \right) \\
&\quad \times \left(\prod_{i' \in \mathcal{P}} \sigma_{i'}^x \sum_{\text{String conf.}\vec{i}\,'} F_{\mathcal{P}}(\vec{i}\,') |\vec{i}\,'\rangle \langle \vec{i}\,'| \right) \\
&= \sum_{\text{String conf.}\vec{i}} F_{\mathcal{P}}(\vec{i}) |\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}\rangle \langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}}| \\
&\quad \times \sum_{\text{String conf.}\vec{i}\,'} F_{\mathcal{P}}(\vec{i}\,') |\vec{i}\,'\rangle \langle \vec{i}\,'| \\
&= \sum_{\text{String conf.}\vec{i}} F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) F_{\mathcal{P}}(\vec{i}) |\vec{i}\rangle \langle \vec{i}|.
\end{aligned}$$

Clearly, $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$ implies that $[S_{\mathcal{P}}^{+}]^2 = 1$, since $F_{\mathcal{P}}(\vec{i})$ is a complex number lying on the unit circle. On the other hand, $[S_{\mathcal{P}}^{+}]^2 = 1$ implies that $F_{\mathcal{P}}(\vec{i}) F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = 1$, which means that $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$, once again because $F_{\mathcal{P}}(\vec{i})$ is a complex number lying on the unit circle. □

Lemma 6. *(Canonical strings)* $[S_{\mathcal{P}}^{+}]^{\dagger} = S_{\mathcal{P}}^{+}$ *if and only if* $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$ *for any configuration* $\vec{i}$.

Proof. Explicit calculation of $[S_{\mathcal{P}}^{+}]^{\dagger}$ gives

$$\begin{aligned}
[S_{\mathcal{P}}^{+}]^{\dagger} &= \sum_{\text{String conf.}\vec{i}} [F_{\mathcal{P}}(\vec{i})]^* |\vec{i}\rangle \langle \vec{i}| \prod_{i \in \mathcal{P}} \sigma_i^x \\
&= \sum_{\text{String conf.}\vec{i}} [F_{\mathcal{P}}(\vec{i})]^* \left(\prod_{j \in \mathcal{P}} \sigma_j^x \right)^2 |\vec{i}\rangle \langle \vec{i}| \prod_{i \in \mathcal{P}} \sigma_i^x \\
&= \prod_{i \in \mathcal{P}} \sigma_i^x \sum_{\text{String conf.}\vec{i}} [F_{\mathcal{P}}(\vec{i})]^* |\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}\rangle \langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}}|,
\end{aligned}$$

Suppose that $[S_{\mathcal{P}}^{+}]^{\dagger} = S_{\mathcal{P}}^{+}$. In that case, it is clear that $[F_{\mathcal{P}}(\vec{i})]^* = F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})$.

It is also clear that $[F_{\mathcal{P}}(\vec{i})]^* = F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})$ implies that $[S_{\mathcal{P}}^{+}]^{\dagger} = S_{\mathcal{P}}^{+}$. □

Lemma 7. *(Canonical strings) Suppose that for any set of plaquettes* $\{p_1, p_2, \ldots, p_m\} \subset \mathcal{B}_{\mathcal{P}}$ *and for a string configuration* $\vec{i}$, *we have that* $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}) = \theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m) F_{\mathcal{P}}(\vec{i})$ *and* $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}) = \theta_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}, p_1, \ldots, p_m) F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})$. *If* $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$, *then for any set of plaquettes* $\{q_1, \ldots, q_k\} \subset \mathcal{B}_{\mathcal{P}}$, *we have that* $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_k}) = [F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_k})]^*$.

Proof. By hypothesis, we have that

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1}) = \theta_{\mathcal{P}}(\vec{i}, q_1) F_{\mathcal{P}}(\vec{i}), \tag{B6}$$

as well as

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{q_1}) = [\theta_{\mathcal{P}}(\vec{i}, q_1)]^{-1} F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}), \tag{B7}$$

where we took advantage of the fact that $\theta_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}, q_1) = [\theta_{\mathcal{P}}(\vec{i}, q_1)]^{-1}$, as is clear from equation (21).

Using the fact that $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$ in equation (B6), we find

$$\begin{aligned}[F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})]^* &= [\theta_{\mathcal{P}}(\vec{i}, q_1)]^{-1} F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1}) \\ &= \frac{F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{q_1})}{F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})} F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1}).\end{aligned} \tag{B8}$$

Since $F_{\mathcal{P}}$ lies on the unit circle, we find that

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{q_1}) F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1}) = 1, \tag{B9}$$

which in turn implies that

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{q_1}) = [F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1})]^*. \tag{B10}$$

The same reasoning can be recursively employed to show that

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_k}) = [F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{q_1} \oplus \ldots \oplus \vec{\alpha}^{q_k})]^*. \tag{B11}$$

□

Lemma 8. *(Canonical strings) Let $F_{\mathcal{P}}$ be a function determined by algorithm 1. If the phases $e^{i\phi}$ are assigned to the class representatives $\vec{i}$ in such a way that $F_{\mathcal{P}}(\vec{i}) = [F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})]^*$, then $F_{\mathcal{P}}$ simultaneously satisfies all the constraints in equation (21).*

Proof. As it was done before, two possible cases are considered.

1. If $\vec{i}$ is one of the configuration picked as a class representative to set the value of $F_{\mathcal{P}}(\vec{i})$, then we trivially have that $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}(\vec{i})]^*$.
2. If $\vec{i}$ is not one of the configurations picked, then once again we find that $\vec{i} = \vec{i}\,' \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}$, as mentioned above. We find

$$F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = F_{\mathcal{P}}(\vec{i}\,' \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}). \tag{B12}$$

Using algorithm 1, we find that

$$\begin{aligned}F_{\mathcal{P}}(\vec{i}\,' \bigoplus_{j=1}^{m} \vec{\alpha}^{p_j}) &= \theta_{\mathcal{P}}(\vec{i}\,', p_1, \ldots, p_m) F^{\pm}_{\mathcal{P}}(\vec{i}\,'), \\ F_{\mathcal{P}}(\vec{i}\,' \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{j=1}^{m} \vec{\alpha}^{p_j}) &= \theta_{\mathcal{P}}(\vec{i}\,' \oplus \vec{\alpha}^{\mathcal{P}}, p_1, \ldots, p_m) F^{\pm}_{\mathcal{P}}(\vec{i}\,' \oplus \vec{\alpha}^{\mathcal{P}}).\end{aligned}$$

Since all the conditions of lemma 7 are satisfied, we have that

$$F_{\mathcal{P}}(\vec{i}\,' \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}) = [F_{\mathcal{P}}(\vec{i}\,' \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m})]^* = [F_{\mathcal{P}}(\vec{i})]^*. \tag{B13}$$

□

B.2. Proof of theorem 1

Given all the previous technical results, it is straightforward to give the proof of theorem 1, which we restate here for ease of reading:

Theorem 1. *Let $\mathcal{P}$ be a path. Any function $F_{\mathcal{P}}$ defined by algorithm 1 is such that $S^{\pm}_{\mathcal{P}}$ is a string operator. Furthermore, it is possible to choose the phases $e^{i\phi(i)}$ such that the string operator is canonical.*

Proof. First note that according to lemma 3, $F_{\mathcal{P}}$ is well-defined. By construction, $F_{\mathcal{P}}$ has non-trivial support only in Conn($\mathcal{P}$),thus any operator $S_{\mathcal{P}}^{\pm}$ built from it satisfies condition (ii). Furthermore,lemma 4 states that $S_{\mathcal{P}}^{+}$ satisfies conditions (22). We thus have that condition (i) is satisfied as well,proving that $S_{\mathcal{P}}^{\pm}$ is a string operator.

In order to show that it is always possible to choose the phases $e^{i\phi(\vec{i})}$'s so that $S_{\mathcal{P}}^{\pm}$ is canonical,we must consider two cases,depending on whether $\mathcal{P}$ is open or close.

Suppose first that the path $\mathcal{P}$ is open. In that case,we have that configurations $\vec{i}$ and $\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}$ are in two distinct configuration classes. Given the class representative $\vec{i}$ for which we set $F_{\mathcal{P}}(\vec{i}) = e^{i\phi}$,we simply pick class representative $\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}$ as representative for its corresponding class, and set $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = e^{-i\phi}$.

If $\mathcal{P}$ is close, then we find that $\vec{i}$ and $\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}$ belong to the same class of configurations, since there exists a set of plaquettes $\{p_1, \ldots, p_m\} \subseteq \mathcal{B}_{\mathcal{P}}$ such that on Conn($\mathcal{P}$), $\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} = \vec{i} \oplus \vec{\alpha}^{p_1} \oplus \ldots \oplus \vec{\alpha}^{p_m}$. Setting $F_{\mathcal{P}}(\vec{i}) = [\theta_{\mathcal{P}}(\vec{i}, p_1, \ldots, p_m)]^{-\frac{1}{2}}$, we find that $F_{\mathcal{P}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}}) = [F_{\mathcal{P}}]^*$.

In both cases, we can use lemma 8 to find that all constraints of equation (21) are fulfilled and therefore, conditions (iii) and (iv) are also satisfied. □

B.3. Consistency of the probability of measuring an excitation configuration

The decomposition of $X_{\mathcal{P}}$ in terms of string operators given by equation (31) is not unique given that in algorithm 1, we are free to choose different initial phases for the various class representatives. However, we show here that the probabilities associated with finding a given excitation pattern after the application $X_{\mathcal{P}}$ are insensitive to those initial phases. For simplicity, we assume that $\mathcal{P}$ is a single non-overlapping and non-crossing open path. The arguments below generalize in a straightforward manner to the case where we need to consider $\mathcal{P} = \mathcal{P}_1 \# \ldots \# \mathcal{P}_m$ for some $m > 1$. First note that

$$|\vec{i}\rangle\langle\vec{i}| = \prod_i \frac{1}{2}(1 + (-1)^{\alpha_i}\sigma_i^z), \tag{B14}$$

where $\vec{i} = (\alpha_1, \ldots, \alpha_n)$. We can thus write $F_{\mathcal{P}}(\vec{i})|\vec{i}\rangle\langle\vec{i}| = \sum_{P_z} c_{P_z}(\vec{i})P_z$, where P_z are all the possible Pauli operators acting on the qubits in Conn($\mathcal{P}$) and composed of σ^z and identities only, and where $c_{P_z}(\vec{i})$ are complex coefficients given by equation (B14) multiplied by $F_{\mathcal{P}}(\vec{i})$. Given $\tilde{F}_{\mathcal{P}}$ differing of $F_{\mathcal{P}}$ by the choice of phases associated with the different class representatives, we have that $\tilde{F}_{\mathcal{P}}(\vec{i})|\vec{i}\rangle\langle\vec{i}| = \sum_{P_z} c_{P_z}(\vec{i})e^{i\varphi(\vec{j})}P_z$, where $\vec{j}$ is the representative of the configuration class into which $\vec{i}$ belongs, and $\varphi(\vec{j})$ is the phase difference used between $F_{\mathcal{P}}$ and $\tilde{F}_{\mathcal{P}}$ to initialize algorithm 1.

We define the orthonormal basis $\{|L, C\rangle\}$ where C labels the vertex and flux excitations configuration while L is a label for the 4^g degenerate states corresponding to a given configuration. The probability of the transition $|L, C\rangle \to |L', C'\rangle$ caused by the application of $X_{\mathcal{P}}$ is given by

$$\begin{aligned} P(|L, C\rangle \to |L', C'\rangle) &= |\langle L', C'|S_{\mathcal{P}}^{+}\sum_{\vec{j}}\sum_{\vec{i}\in\mathcal{C}_{\mathcal{P}}(\vec{j})}\sum_{P_z} c_{P_z}^*(\vec{i})e^{-i\varphi(\vec{j})}P_z|L, C\rangle|^2 \\ &= \sum_{\substack{\vec{j},\vec{j}\,'}}\sum_{\substack{\vec{i}\in\mathcal{C}_{\mathcal{P}}(\vec{j}) \\ \vec{i}\,'\in\mathcal{C}_{\mathcal{P}}(\vec{j}\,')}}\sum_{\substack{P_z \text{ s.t.}|\langle L', C'|S_{\mathcal{P}}^{+}P_z|L, C\rangle|=1 \\ P_z' \text{ s.t.}|\langle L', C'|S_{\mathcal{P}}^{+}P_z'|L, C\rangle|=1}} c_{P_z}^*(\vec{i})c_{P_z'}(\vec{i}\,')e^{i(\varphi(\vec{j})-\varphi(\vec{j}\,'))}. \end{aligned} \tag{B15}$$

Clearly, P_z and P_z' share the same endpoints and belong to the same homological class, i.e. P_zP_z' form a trivial closed loop. Given the decomposition in equation (B14), we find that for $\vec{j} \neq \vec{j}\,'$

$$\sum_{\substack{P_z \text{ s.t.}|\langle L', C'|S_{\mathcal{P}}^{+}P_z|L, C\rangle|=1 \\ P_z' \text{ s.t.}|\langle L', C'|S_{\mathcal{P}}^{+}P_z'|L, C\rangle|=1}} c_{P_z}^*(\vec{i})c_{P_z'}(\vec{i}\,') = 0. \tag{B16}$$

To see this, we rewrite equation (B16) as

$$\sum_{P_z \text{ s.t.}|\langle L', C'|S_{\mathcal{P}}^{+}P_z|L, C\rangle|=1}\sum_{Q_z\subseteq\{Q_\nu\}} c_{P_z}^*(\vec{i})c_{P_z\cdot Q_z}(\vec{i}\,'), \tag{B17}$$

where $\{Q_\nu\}$ denotes the set of all subsets of products of charge operators associated with the vertices in path $\mathcal{P}$. Noting that since $\vec{i}$ and $\vec{i}\,'$ belongs to different configuration classes, there exists a vertex ν' such that $c_{P_z}(\vec{i}\,') = -c_{P_z\cdot Q_{\nu'}}(\vec{i}\,')$. Using this last fact, we get

$$\sum_{P_z \text{ s.t.}|\langle L', C'|S_{\mathcal{P}}^{+}P_z|L, C\rangle|=1}\sum_{Q_z\subseteq\{Q_\nu\}\setminus Q_{\nu'}} [c_{P_z}^*(\vec{i})c_{P_z\cdot Q_z}(\vec{i}\,') + c_{P_z}^*(\vec{i})c_{P_z\cdot Q_z Q_{\nu'}}(\vec{i}\,')], \tag{B18}$$

which clearly equals 0.

The probability transition is thus given by

$$P(|L,\,C\rangle \to |L',\,C'\rangle) = \sum_{j} \sum_{i,i' \in \mathcal{C}_{\mathcal{P}}(\vec{j})} \sum_{\substack{P_z \text{s.t.} |\langle L',C'|P_z|L,C\rangle|=1 \\ P'_z \text{s.t.} |\langle L',C'|P'_z|L,C\rangle|=1}} c^{*}_{P_z}(\vec{i}\,')\, c_{P'_z}(\vec{i})\, \langle L,\, C|P_z P'_z|L,\, C\rangle, \tag{B19}$$

which is independent of the phases $e^{i\varphi(\vec{j})}$.

Appendix C. Topological properties of strings operators

Before presenting various technical results, we begin by precisely defining what we mean by *crossing paths*.

Definition 1. Let a path $\mathcal{P} = \{e_1, e_2, \ldots, e_n\}$ be a sequence of edges such that edge e_i connects vertices v_{i-1} to v_i. If $v_0 = v_n$, we say that the path $\mathcal{P}$ is closed; otherwise it is open. If there exists $0 < i < n$ and $j \neq i$, $0 \leqslant j \leqslant n$ such that $v_i = v_j$ in the sequence of vertices it contains, $v_{\mathcal{P}} = \{v_0, v_1, \ldots, v_n\}$, then $\mathcal{P}$ is said to be *self-crossing*. If there exists a plaquette containing two or more vertices of $v_{\mathcal{P}}$ that cannot form a single consecutive sequence, then path $\mathcal{P}$ is said to be self-overlapping. Note that the notions of self-overlapping and self-crossing do not imply each other (see figure 5).

Definition 2. Consider two paths $\mathcal{P} = \{e_1^{\mathcal{P}}, \ldots, e_n^{\mathcal{P}}\}$ and $\mathcal{Q} = \{e_1^{\mathcal{Q}}, \ldots, e_{n'}^{\mathcal{Q}}\}$ connecting vertices $\{v_0^{\mathcal{P}}, \ldots, v_n^{\mathcal{P}}\}$ and $\{v_0^{\mathcal{Q}}, \ldots, v_{n'}^{\mathcal{Q}}\}$ respectively. Consider a sequence of edges in common of both paths $\mathcal{P}$ and $\mathcal{Q}$, and consider the largest (possibly empty) such sequence (supposing for now that it is unique), $E_{\mathcal{P},\mathcal{Q}} = \{e_i^{\mathcal{P}} = e_j^{\mathcal{Q}}, \ldots, e_{i'}^{\mathcal{P}} = e_{j'}^{\mathcal{Q}}\}$ (with both se-quences ordered in increasing order of simplicity, i.e. $i' > i$ and $j' > j$) connecting the common vertices in $\mathcal{P}$ and $\mathcal{Q}$, denoted by $\Lambda_{\mathcal{P},\mathcal{Q}} = \{v_{i-1}^{\mathcal{P}} = v_{j-1}^{\mathcal{Q}}, \ldots, v_{i'}^{\mathcal{P}} = v_{j'}^{\mathcal{Q}}\}$. Consider the following properties:

(i) (in the case where both paths open) none of the vertices $v_0^{\mathcal{P}}$, $v_n^{\mathcal{P}}$, $v_0^{\mathcal{Q}}$ and $v_{n'}^{\mathcal{Q}}$ are in $\Lambda_{\mathcal{P},\mathcal{Q}}$,

(ii) (in the case where one path is open ($\mathcal{P}$), the other is closed ($\mathcal{Q}$)) none of the vertices $v_0^{\mathcal{P}}$ and $v_n^{\mathcal{P}}$ are in $\Lambda_{\mathcal{P},\mathcal{Q}}$,

(iii) both pairs of edges $(e_{i-1}^{\mathcal{P}}, e_{j-1}^{\mathcal{Q}})$ and $(e_{i'+1}^{\mathcal{P}}, e_{j'+1}^{\mathcal{Q}})$ have the same relative orientation, i.e. clockwise or counter-clockwise.

If condition (i) is satisfied, we say that paths $\mathcal{P}$ and $\mathcal{Q}$ cross over the edges $E_{\mathcal{P},\mathcal{Q}}$. Note that for the case of two closed paths, we always say that they cross. If condition (ii) is not satisfied (including the case where $E_{\mathcal{P},\mathcal{Q}}$ is the empty set), we say that $\mathcal{P}$ and $\mathcal{Q}$ cross 0 times, otherwise we say that paths $\mathcal{P}$ and $\mathcal{Q}$ cross once. Finally, if there is more than one pair of sequences $E^{i}_{\mathcal{P},\mathcal{Q}}$ and $\Lambda^{i}_{\mathcal{P},\mathcal{Q}}$ where i runs from 1 through m which *all* satisfy conditions (i), we say that paths $\mathcal{P}$ and $\mathcal{Q}$ cross. For all the regions $E^{i_1}_{\mathcal{P},\mathcal{Q}}, \ldots, E^{i_{m'}}_{\mathcal{P},\mathcal{Q}}$ and $\Lambda^{i_1}_{\mathcal{P},\mathcal{Q}}, \ldots, \Lambda^{i_{m'}}_{\mathcal{P},\mathcal{Q}}$ for which (ii) is satisfied, with $m' \leqslant m$, we say that paths $\mathcal{P}$ and $\mathcal{Q}$ cross over the relevant region, and we additionally say that paths $\mathcal{P}$ and $\mathcal{Q}$ cross m' times. See figure C1 for explicit examples.

Note that in the previous definition, paths $\mathcal{P}$ and $\mathcal{Q}$ can be formed of smaller paths, i.e. $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$ and $\mathcal{Q} = \mathcal{Q}_1\#...\#\mathcal{Q}_n$. Notice that when $\mathcal{P}$ and $\mathcal{Q}$ cross, we may define a reference frame such that one of the paths plays the role of the horizontal and the other the vertical. In the following, we consider that path $\mathcal{P}$ is the horizontal and $\mathcal{Q}$ the vertical.

It is useful to define $\mathcal{B}_{\mathcal{P}}^{\mathcal{Q},\text{left}} = \{p_1^{\text{left}}, \ldots, p_m^{\text{left}}\}$, the set of plaquettes in $\mathcal{B}_{\mathcal{P}}$ such that $\vec{\alpha}^{p_1^{\text{left}}} \oplus ... \oplus \vec{\alpha}^{p_m^{\text{left}}} = \vec{\alpha}^{\mathcal{Q}}$ when restricted to Conn($\mathcal{P}$) and containing the left-most plaquette of $\mathcal{B}_{\mathcal{P}}$. In a complementary way, we define $\mathcal{B}_{\mathcal{P}}^{\mathcal{Q},\text{right}} = \mathcal{B}_{\mathcal{P}} \backslash \mathcal{B}_{\mathcal{P}}^{\mathcal{Q},\text{left}}$. In a similar way, $\mathcal{B}_{\mathcal{Q}}^{\mathcal{P},\text{up}}$ and $\mathcal{B}_{\mathcal{Q}}^{\mathcal{P},\text{down}}$ can be defined for a suitable path $\mathcal{Q}$. The nomenclature of *left* versus *right* is an arbitrary choice (just as is the case for *up* versus *down*).

Note that we implicitly used the fact that paths $\mathcal{P}$ and $\mathcal{Q}$ cross, are open, and that they are not self-overlapping nor self-crossing in the above definition. If it were not the case, then it would not be possible to find a set of plaquettes such that the associated configuration corresponds to the configuration of the other path on its connected region.

Furthermore, for a general path $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$ with every path $\{\mathcal{P}_i\}$ open, not self-crossing nor self-overlapping, but for which it is not necessarily true for the *whole* path $\mathcal{P}$, and for a path $\mathcal{Q}$ such that $\mathcal{P}$ and $\mathcal{Q}$ are crossing, it is always possible to similarly define $\mathcal{B}_{\mathcal{P}_i}^{\mathcal{Q},\text{left (right)}}$ for $i \in \{1, \ldots, m\}$.

Consider two crossing paths $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$ and $\mathcal{Q} = \mathcal{Q}_1\#...\#\mathcal{Q}_n$ such that every individual path $\mathcal{P}_i$ and $\mathcal{Q}_j$ is open, is not self-crossing nor self-overlapping, and we are interested in computing the commutation relations between $S^{+}_{\mathcal{P}_1\#...\#\mathcal{P}_m}$ and $S^{+}_{\mathcal{Q}_1\#...\#\mathcal{Q}_n}$. Explicit calculations give

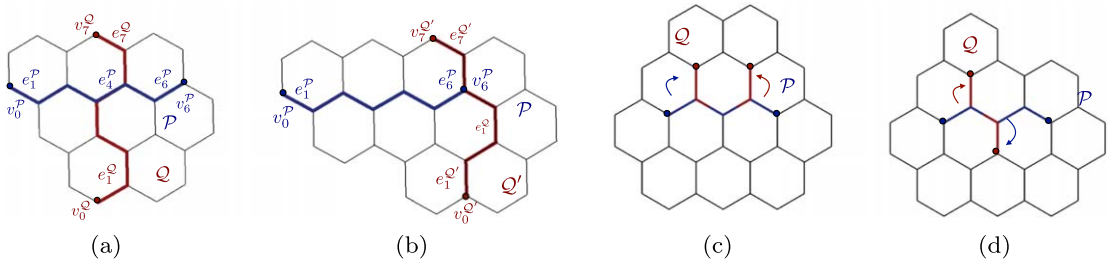

Figure C1. Example of two open paths $\mathcal{P}$ and $\mathcal{Q}$ which, according to criteria (i) of definition 2 are crossing over the edge $E_{\mathcal{P},\mathcal{Q}} = \{e_4^{\mathcal{P}} = e_4^{\mathcal{Q}}\}$(a), as well as paths $\mathcal{P}$ and $\mathcal{Q}'$ which are not crossing (b), since path $\mathcal{Q}$ contains vertex $v_6^{\mathcal{P}}$. The two open paths $\mathcal{P}$ and $\mathcal{Q}$ cross 0 times in (c), according to criteria (ii), while they cross once in (d). The arrows indicate the relative orientation among the strings.

$$S^+_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m} S^+_{\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} = \prod_{i\in\mathcal{P}} \sigma_i^x \prod_{j\in\mathcal{Q}} \sigma_j^x \sum_{\text{string conf.}\,\vec{i}} \prod_{i=1}^{m} F_{\mathcal{P}_i}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j}\right) \prod_{i'=1}^{n} F_{\mathcal{Q}_{i'}}\left(\vec{i} \bigoplus_{j'=1}^{i'-1} \vec{\alpha}^{\mathcal{Q}_{j'}}\right) |\vec{i}\rangle\langle\vec{i}|, \quad \text{(C1)}$$

where the product over the small strings are taken in the reverse order, and where the string configurations are considered over the whole system.

Similarly computing the product of the string operators in the reverse order, we get

$$S^+_{\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} S^+_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m} = \prod_{i\in\mathcal{P}} \sigma_i^x \prod_{j\in\mathcal{Q}} \sigma_j^x \sum_{\text{string conf.}\,\vec{i}} \prod_{i=1}^{m} F_{\mathcal{P}_i}\left(\vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j}\right) \prod_{i'=1}^{n} F_{\mathcal{Q}_{i'}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{j'=1}^{i'-1} \vec{\alpha}^{\mathcal{Q}_{j'}}\right) |\vec{i}\rangle\langle\vec{i}|. \quad \text{(C2)}$$

Considering equations (C1) and (C2), we define the quantity

$$\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i}) = \frac{\prod_{i=1}^{m} F_{\mathcal{P}_i}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j}\right) \prod_{i'=1}^{n} F_{\mathcal{Q}_{i'}}\left(\vec{i} \bigoplus_{j'=1}^{i'-1} \vec{\alpha}^{\mathcal{Q}_{j'}}\right)}{\prod_{i=1}^{m} F_{\mathcal{P}_i}\left(\vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j}\right) \prod_{i'=1}^{n} F_{\mathcal{Q}_{i'}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{j'=1}^{i'-1} \vec{\alpha}^{\mathcal{Q}_{j'}}\right)}, \quad \text{(C3)}$$

which gives the commutation relations between $S^+_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m}$ and $S^+_{\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}$.The quantity defined in equation (C3) is independent on the specific string configuration $\vec{i}$, which is shown in lemmas 9 and 10, and that it does not depend on the specific way that the paths $\mathcal{P}$ and $\mathcal{Q}$ are partitioned in the smaller paths $\{\mathcal{P}_i\}$ and $\{\mathcal{Q}_j\}$, as long as those are not self-crossing nor self-overlapping. It can also be shown that if paths $\mathcal{P}$ and $\mathcal{Q}$ are transformed to paths $\mathcal{P}'$ and $\mathcal{Q}'$ using a series of elongations, reductions and valid deformations such that none of the elementary step makes a path crossing an endpoint of the other path, then we find $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \mathcal{R}_{\mathcal{P}',\mathcal{Q}'}$. This is done in lemmas 11 through 15 as well as in corollary 1. If paths $\mathcal{P}$ and $\mathcal{Q}$ cross an odd number of times, one can then proceed to transforms paths $\mathcal{P}$ and $\mathcal{Q}$ to minimal configurations $\mathcal{P}_{\min}$ and $\mathcal{Q}_{\min}$ such that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \mathcal{R}_{\mathcal{P}_{\min},\mathcal{Q}_{\min}}$ (see figure C4). Explicitly computing this last quantity for a given string configuration yields $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = -1$. If, on the other hand, $\mathcal{P}$ and $\mathcal{Q}$ cross an even number of times, one can consider the deformed paths $\mathcal{P}_0$ and $\mathcal{Q}_0$ such that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \mathcal{R}_{\mathcal{P}_0,\mathcal{Q}_0}$ and such that $\mathcal{P}_0$ and $\mathcal{Q}_0$ supports are disjoint, showing that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = +1$. Remarkably, all those previous results are essentially due to the fact that the plaquette operators commute and square to the identity.

C.1. Useful technical lemmas

Lemmas mentioned before are introduced here. They are necessary to show theorem 2.

Lemma 9. *Let* $\{p_1^i, \ldots, p_{m^i}^i\} = \mathcal{B}_{\mathcal{P}_i}^{\mathcal{Q},a_i}$ *and* $\{q_1^j, \ldots, q_{n^j}^j\} = \mathcal{B}_{\mathcal{Q}_j}^{\mathcal{P},b_j}$ *with* $a_i =$ left *or* $a_i =$ right *and* $b_j =$ up *or* $b_j =$ down, *as it is shown in figure C2. Then,* $\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i})$ *can be written as:*

$$\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i}) = \prod_{i=1}^{m} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}. \quad \text{(C4)}$$

Proof. We begin by considering the quantity

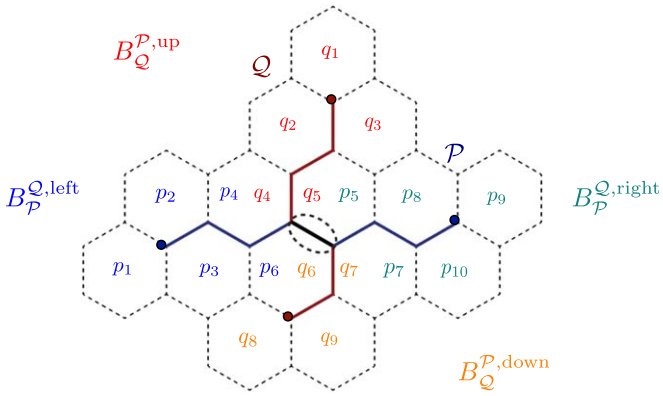

Figure C2. An example of the structure of string operators $S_{\mathcal{P}}^{+}$ and $S_{\mathcal{Q}}^{+}$ on crossing paths $\mathcal{P}$ and $\mathcal{Q}$, shown in dark blue and red, respectively. The various sets of plaquettes $\mathcal{B}_{\mathcal{P}}^{\mathcal{Q},\mathrm{left\ (right)}}$ and $\mathcal{B}_{\mathcal{Q}}^{\mathcal{P},\mathrm{up\ (down)}}$ are presented in various matching colors. The edge over which both paths $\mathcal{P}$ and $\mathcal{Q}$ are overlapping is circled by a dashed line.

$$\frac{F_{\mathcal{P}_i}\left(\vec{i}\,\oplus\,\vec{\alpha}^{\mathcal{Q}}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}{F_{\mathcal{P}_i}\left(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}, \tag{C5}$$

which appears in equation (C3). Using the definition of $\mathcal{B}_{\mathcal{P}_i}^{\mathcal{Q},a_i}$, we find

$$\frac{F_{\mathcal{P}_i}\left(\vec{i}\,\oplus\,\vec{\alpha}^{\mathcal{Q}}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}{F_{\mathcal{P}_i}\left(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}=\frac{F_{\mathcal{P}_i}(\vec{i}\bigoplus_{j=1}^{m^i}\vec{\alpha}^{p_j^i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j})}{F_{\mathcal{P}_i}\left(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}. \tag{C6}$$

Using the structure of the $F_{\mathcal{P}_i}$ as defined by algorithm 1, we find that there exists a configuration $\vec{i}\,'$ and a set of plaquettes $\{p_1', \ldots, p_{m'}'\}$ (possibly empty) such that $\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}=\vec{i}\,'\bigoplus_{j=1}^{m'}\vec{\alpha}^{p_j'}$ when restricted to $\mathrm{Conn}(\mathcal{P}_i)$. We can thus write

$$\frac{F_{\mathcal{P}_i}\left(\vec{i}\,\oplus\,\vec{\alpha}^{\mathcal{Q}}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}{F_{\mathcal{P}_i}\left(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}=\frac{\theta_{\mathcal{P}_i}(\vec{i}\,', p_1', \ldots, p_{m'}', p_1^i, \ldots, p_{m^i}^i)}{\theta_{\mathcal{P}_i}(\vec{i}\,', p_1', \ldots, p_{m'}')}=\theta_{\mathcal{P}_i}(\vec{i}, p_1^i, \ldots, p_{m^i}^i), \tag{C7}$$

where we made use of equation (24). Note that using lemma 2, we are free to choose the values for $a_i = \mathrm{left/right}$ and $b_j = \mathrm{up/down}$, as we please. This will turn out to be very useful later on.

Using equation (24), we find

$$\frac{F_{\mathcal{P}_i}^{+}\left(\vec{i}\,\oplus\,\vec{\alpha}^{\mathcal{Q}}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}{F_{\mathcal{P}_i}^{+}\left(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right)}=\frac{\left\langle\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\middle|\prod_{j=1}^{m^i}B_{p_j^i}\middle|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\right\rangle}{\left\langle\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\middle|\prod_{j=1}^{m^i}B_{p_j^i}\middle|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right\rangle}. \tag{C8}$$

A similar reasoning holding for the quantity

$$\left(\frac{F_{\mathcal{Q}_{i'}}^{+}\left(\vec{i}\bigoplus_{j'=1}^{i'-1}\vec{\alpha}^{\mathcal{Q}_{j'}}\right)}{F_{\mathcal{Q}_{i'}}^{+}(\vec{i}\,\oplus\,\vec{\alpha}^{\mathcal{P}}\bigoplus_{j'=1}^{i'-1}\vec{\alpha}^{\mathcal{Q}_{j'}})}\right)^{-1}, \tag{C9}$$

this completes the proof. □

The precedent lemma stipulates that $\mathcal{R}_{\mathcal{P},\mathcal{Q}}(\vec{i}\,)$ can be written in terms of the phases acquired by the product of the plaquette operators of the plaquettes contained in $\mathcal{B}_{\mathcal{P}}^{\mathcal{Q},a}$ and $\mathcal{B}_{\mathcal{Q}}^{\mathcal{P},b}$, on the appropriate string configurations.

Lemma 10. *The quantity $\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i}\,)$ is independent of the string configuration $\vec{i}$.*

Proof. Consider an arbitrary edge e and its associated canonical string operator S_e^{+}, which we can always find according to theorem 1. Further let $\{p_1^i, \ldots, p_{m^i}^i\} = \mathcal{B}_{\mathcal{P}_i}^{\mathcal{Q},a_i}$ and $\{q_1^j, \ldots, q_{n^j}^j\} = \mathcal{B}_{\mathcal{Q}_j}^{\mathcal{P},b_j}$ with $a_i =$ left or $a_i =$ right and $b_j =$ up or $b_j =$ down, and which we are always free to choose according to lemma 2. Using lemma 9 and the fact that S_e^{+} is canonical and therefore squares to one, the quantity $\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_m}(\vec{i}\,)$ is given by

$$\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i}) = \prod_{i=1}^{m}\left(\frac{\left\langle \vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}|S_e^+\prod_{j=1}^{m^i}B_{p_j^i}S_e^+|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\right\rangle}{\left\langle \vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}|S_e^+\prod_{j=1}^{m^i}B_{p_j^i}S_e^+|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right\rangle}\right)$$
$$\times\prod_{i=1}^{n}\left(\frac{\left\langle \vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}|S_e^+\prod_{j=1}^{n^i}B_{q_j^i}S_e^+|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\right\rangle}{\left\langle \vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}|S_e^+\prod_{j=1}^{n^i}B_{q_j^i}S_e^+|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\right\rangle}\right)$$
$$=\prod_{i=1}^{m}\left(\frac{\left\langle \vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e}|\prod_{j=1}^{m^i}B_{p_j^i}|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\oplus\vec{\alpha}^{e}\right\rangle}{\left\langle \vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e}|\prod_{j=1}^{m^i}B_{p_j^i}|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\oplus\vec{\alpha}^{e}\right\rangle}\right)$$
$$\times\prod_{i=1}^{n}\left(\frac{\left\langle \vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e}|\prod_{j=1}^{n^i}B_{q_j^i}|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\oplus\vec{\alpha}^{e}\right\rangle}{\left\langle \vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e}|\prod_{j=1}^{n^i}B_{q_j^i}|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\oplus\vec{\alpha}^{e}\right\rangle}\right)$$
$$\times\prod_{i=1}^{m}\left(\frac{F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e})F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j})}{F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j})F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e})}\right)$$
$$\times\prod_{i=1}^{n}\left(\frac{F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e})F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j})}{F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j})F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e})}\right). \tag{C10}$$

Carefully looking at the right-hand side of the equality, we find that

$$\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i}) = \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i}\oplus\vec{\alpha}^{e})$$
$$\times\prod_{i=1}^{m}\left(\frac{F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e})F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j})}{F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j})F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e})}\right)$$
$$\times\prod_{i=1}^{n}\left(\frac{F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e})F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j})}{F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j})F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e})}\right). \tag{C11}$$

Note that it is always possible to choose the a_i's and the b_j's so that, in case of need, we can add some additional plaquettes in $\mathcal{B}_{\mathcal{P}_i}^{\mathcal{Q},a_i}$ and $\mathcal{B}_{\mathcal{Q}_j}^{\mathcal{P},b_j}$ respectively without affecting $\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}(\vec{i})$, in order to have that on $\mathrm{Conn}(\{e\})$, we find $\vec{\alpha}^{p_1^i}\oplus\ldots\oplus\vec{\alpha}^{p_{m^i}^i}=\vec{\alpha}^{\mathcal{Q}}$ for any i, and $\vec{\alpha}^{q_1^j}\oplus\ldots\oplus\vec{\alpha}^{q_{n^j}^j}=\vec{\alpha}^{\mathcal{P}}$ for any j. Using this, we find

$$\prod_{i=1}^{m}\left(\frac{F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e})F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j})}{F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j})F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}\oplus\vec{\alpha}^{e})}\right)$$
$$\times\prod_{i=1}^{n}\left(\frac{F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e})F_e(\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j})}{F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j})F_e(\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}\oplus\vec{\alpha}^{e})}\right)$$
$$=\frac{F_e(\vec{i}\oplus\vec{\alpha}^{\mathcal{P}}\oplus\vec{\alpha}^{\mathcal{Q}}\oplus\vec{\alpha}^{e})F_e(\vec{i}\oplus\vec{\alpha}^{\mathcal{P}})}{F_e(\vec{i})F_e(\vec{i}\oplus\vec{\alpha}^{\mathcal{Q}}\oplus\vec{\alpha}^{e})}\times\frac{F_e(\vec{i}\oplus\vec{\alpha}^{\mathcal{P}}\oplus\vec{\alpha}^{e})F_e(\vec{i})}{F_e(\vec{i}\oplus\vec{\alpha}^{\mathcal{Q}})F_e(\vec{i}\oplus\vec{\alpha}^{\mathcal{Q}}\oplus\vec{\alpha}^{\mathcal{P}}\oplus\vec{\alpha}^{e})}=1, \tag{C12}$$

where we used that $F_e(\vec{i}\,'\oplus\vec{\alpha}^{e})=[F_e(\vec{i}\,')]^{-1}$ for any string configuration $\vec{i}\,'$, since S_e^+ is canonical. □

Lemma 11. *Let $\mathcal{P}=\mathcal{P}_1\#\ldots\#\mathcal{P}_i\#\mathcal{P}_{i+1}\#\ldots\#\mathcal{P}_m$ and $\mathcal{Q}=\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n$ be two crossing paths. For any $i\in\{1,\ldots,m-1\}$, we denote $\mathcal{P}_i=\{e_1^{\mathcal{P}_i},\ldots,e_{|\mathcal{P}_i|}^{\mathcal{P}_i}\}$, we define $\mathcal{P}_i^+=\{e_1^{\mathcal{P}_i},\ldots,e_{|\mathcal{P}_i|}^{\mathcal{P}_i},e_1^{\mathcal{P}_{i+1}}\}$, ${}^+\mathcal{P}_i=\{e_{|\mathcal{P}_{i-1}|}^{\mathcal{P}_{i-1}},e_1^{\mathcal{P}_i},\ldots,e_{|\mathcal{P}_i|}^{\mathcal{P}_i}\}$, and similarly for $\mathcal{P}_i^-$ and ${}^-\mathcal{P}_i$, this time* removing *the first or last edge, depending on the case. If the paths $\mathcal{P}_i^{\pm}$ and ${}^{\pm}\mathcal{P}_{i+1}$ are not self-overlapping nor self-crossing, we have that*

$$\begin{aligned}\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_i\#\mathcal{P}_{i+1}\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} &= \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_i^+\#^-\mathcal{P}_{i+1}\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}\\ &= \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_i^-\#^+\mathcal{P}_{i+1}\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}.\end{aligned} \tag{C13}$$

Proof. Consider the edge $e_1^{\mathcal{P}_{x+1}}$ and the corresponding *canonical* string operator, written as S_e^+. We then have that

$$\begin{aligned}\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x\#\mathcal{P}_{x+1}\#\ldots\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} = &\prod_{i=1}^{m}\frac{\left\langle\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}|[S_e^+]^{\delta_{x,i}}\prod_{j=1}^{m^i}B_{p_j^i}[S_e^+]^{\delta_{x,i}}|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{P}_j}\right\rangle}{\left\langle\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^i}\vec{\alpha}^{p_k^i}|[S_e^+]^{\delta_{x,i+1}}\prod_{j=1}^{m^i}B_{p_j^i}[S_e^+]^{\delta_{x,i+1}}|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{P}_j}\right\rangle}\\ &\times\prod_{i=1}^{n}\frac{\left\langle\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}|\prod_{j=1}^{n^i}B_{q_j^i}|\vec{i}\bigoplus_{j=1}^{i-1}\vec{\alpha}^{\mathcal{Q}_j}\right\rangle}{\left\langle\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\bigoplus_{k=1}^{n^i}\vec{\alpha}^{q_k^i}|\prod_{j=1}^{n^i}B_{q_j^i}|\vec{i}\bigoplus_{j=1}^{i}\vec{\alpha}^{\mathcal{Q}_j}\right\rangle}.\end{aligned} \tag{C14}$$

We thus get

$$\begin{aligned}\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x\#\mathcal{P}_{x+1}\#\ldots\#P_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} = &\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x^+\#^-P_{x+1}\#\ldots\#P_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}\\ &\times\frac{F_e(\vec{i}\bigoplus_{j=1}^{x}\vec{\alpha}^{\mathcal{P}_j})F_e(\vec{i}\bigoplus_{j}^{x}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^x}\vec{\alpha}^{p_k^x}\oplus\vec{\alpha}^{e})}{F_e(\vec{i}\bigoplus_{j=1}^{x}\vec{\alpha}^{\mathcal{P}_j})F_e(\vec{i}\bigoplus_{j}^{x}\vec{\alpha}^{\mathcal{P}_j}\bigoplus_{k=1}^{m^{x+1}}\vec{\alpha}^{p_k^{x+1}}\oplus\vec{\alpha}^{e})}.\end{aligned} \tag{C15}$$

As in the reasoning of the proof of lemma 10, we used our freedom to add some plaquettes to $\mathcal{B}_{\mathcal{P}_x}^{\mathcal{Q},a_x}$ and $\mathcal{B}_{\mathcal{P}_{x+1}}^{\mathcal{Q},a_{x+1}}$, so that we have $\mathcal{B}_{\mathcal{P}_x^+}^{\mathcal{Q},a_x}\subset\mathcal{B}_{\mathcal{P}_x}^{\mathcal{Q},a_x}$ and $\mathcal{B}_{^-\mathcal{P}_{x+1}}^{\mathcal{Q},a_{x+1}}\subset\mathcal{B}_{\mathcal{P}_{x+1}}^{\mathcal{Q},a_{x+1}}$. Additionally, when can choose them such that when restricted to $\mathrm{Conn}(\{e\})$, we have that $\bigoplus_{k=1}^{m^x}\vec{\alpha}^{p_k^x}=\bigoplus_{k=1}^{m^{x+1}}\vec{\alpha}^{p_k^{x+1}}=\mathcal{Q}$. We thus find that

$$\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x\#\mathcal{P}_{x+1}\#\ldots\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} = \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x^+\#^-P_{x+1}\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}. \tag{C16}$$

A similar reasoning shows that

$$\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x\#\mathcal{P}_{x+1}\#\ldots\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n} = \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_x^-\#^+P_{x+1}\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}. \tag{C17}$$

□

Corollary 1. *Let $\mathcal{P}=\mathcal{P}_1\#\ldots\#\mathcal{P}_m$ and $\mathcal{Q}=\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n$ be two crossing paths. For any $i\in\{1,\ldots,n-1\}$, we have that*

$$\begin{aligned}\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_i\#\mathcal{Q}_{i+1}\#\ldots\#\mathcal{Q}_n} &= \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_i^+\#^-\mathcal{Q}_{i+1}\#\ldots\#\mathcal{Q}_n}\\ &= \mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_i^-\#^+\mathcal{Q}_{i+1}\#\ldots\#\mathcal{Q}_n}.\end{aligned} \tag{C18}$$

Proof. To see that equation (C18) holds, it suffices to use the same reasoning than in lemma 11, this time for the appropriate path $\mathcal{Q}_x$. □

Given the fact that the quantity $\mathcal{R}_{\mathcal{P}_1\#\ldots\#\mathcal{P}_m,\mathcal{Q}_1\#\ldots\#\mathcal{Q}_n}$ is insensitive to the specific decomposition of the paths $\mathcal{P}$ and $\mathcal{Q}$ as long as they are composed of simple paths which are not self-crossing nor self-overlapping, from now on we will simply write $\mathcal{R}_{\mathcal{P},\mathcal{Q}}$.

Lemma 12. *Consider the paths $\mathcal{P}$, $\mathcal{P}'=\mathcal{P}_0\#\mathcal{P}\#\mathcal{P}_{m+1}$, $\mathcal{Q}$ and $\mathcal{Q}'=\mathcal{Q}_0\#\mathcal{Q}\#\mathcal{Q}_{n+1}$, such that $\mathcal{P}$ and $\mathcal{Q}$ are crossing, and such that $\mathcal{P}_0$, $\mathcal{P}_{m+1}$ do not contain edges in* $\mathrm{Conn}(\mathcal{Q}')$*, as well as $\mathcal{Q}_0$, $\mathcal{Q}_{n+1}$ do not contain edges in* $\mathrm{Conn}(\mathcal{P}')$*. Then, we have that*

$$\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \mathcal{R}_{\mathcal{P}',\mathcal{Q}'}. \tag{C19}$$

Proof. We begin by considering $\mathcal{R}_{\mathcal{P}_0\#\mathcal{P},\mathcal{Q}}$, given by

$$\mathcal{R}_{\mathcal{P}_0\#\mathcal{P},\mathcal{Q}} = \prod_{i=0}^{m} \frac{\left\langle \vec{i} \bigoplus_{j=0}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \bigoplus_{j=0}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=0}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \bigoplus_{j=0}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}. \tag{C20}$$

Inserting $[S_{\mathcal{P}_0}^{+}]^2$ at various appropriate locations, with $S_{\mathcal{P}_0}^{+}$ a canonical string operator, and rearranging the terms, we find

$$\begin{aligned}\mathcal{R}_{\mathcal{P}_0\#\mathcal{P},\mathcal{Q}} = \prod_{i=1}^{m} & \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \\ & \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | S_{\mathcal{P}_0}^{+} \prod_{j=1}^{n^i} B_{q_j^i} S_{\mathcal{P}_0}^{+} | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | S_{\mathcal{P}_0}^{+} \prod_{j=1}^{n^i} B_{q_j^i} S_{\mathcal{P}_0}^{+} | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle} \\ & \times \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{k=1}^{m^0} \vec{\alpha}^{p_k^0} | S_{\mathcal{P}_0}^{+} \prod_{j=1}^{m^0} B_{p_j^0} S_{\mathcal{P}_0}^{+} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \right\rangle}{\left\langle \vec{i} \bigoplus_{k=1}^{m^0} \vec{\alpha}^{p_k^0} | \prod_{j=1}^{m^0} B_{p_j^0} | \vec{i} \right\rangle}. \end{aligned} \tag{C21}$$

Using the structure of $S_{\mathcal{P}_0}^{+}$, we find that

$$\begin{aligned}\mathcal{R}_{\mathcal{P}_0\#\mathcal{P},\mathcal{Q}} = \prod_{i=1}^{m} & \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \\ & \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle} \\ & \times \frac{F_{\mathcal{P}_0}\left(\vec{i} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} \oplus \vec{\alpha}^{\mathcal{P}_0}\right) F_{\mathcal{P}_0}(\vec{i})}{F_{\mathcal{P}_0}(\vec{i} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} \oplus \vec{\alpha}^{\mathcal{P}_0}) F_{\mathcal{P}_0}(\vec{i} \oplus \vec{\alpha}^{\mathcal{Q}})} F_{\mathcal{P}_0}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0}) F_{\mathcal{P}_0}\left(\vec{i} \bigoplus_{k=1}^{m^0} \vec{\alpha}^{p_k^0}\right). \end{aligned} \tag{C22}$$

Since we have that $\mathcal{P}_0 \cap \mathrm{Conn}(\mathcal{Q}) = \varnothing$ and since paths $\mathcal{P}$ and $\mathcal{Q}$ are crossing, we find that for any string configuration $\vec{i}\,'$, $F_{\mathcal{P}_0}(\vec{i}\,' \oplus \vec{\alpha}^{\mathcal{Q}}) = F_{\mathcal{P}_0}(\vec{i}\,')$. We thus conclude that

$$\frac{F_{\mathcal{P}_0}\left(\vec{i} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} \oplus \vec{\alpha}^{\mathcal{P}_0}\right) F_{\mathcal{P}_0}(\vec{i})}{F_{\mathcal{P}_0}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} \oplus \vec{\alpha}^{\mathcal{P}_0}\right) F_{\mathcal{P}_0}(\vec{i} \oplus \vec{\alpha}^{\mathcal{Q}})} F_{\mathcal{P}_0}(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0}) F_{\mathcal{P}_0}\left(\vec{i} \bigoplus_{k=1}^{m^0} \vec{\alpha}^{p_k^0}\right) = 1. \tag{C23}$$

Furthermore, using lemma 10, we see that

$$\begin{aligned}\prod_{i=1}^{m} & \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p_k^i} | \prod_{j=1}^{m^i} B_{p_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \\ & \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q_k^i} | \prod_{j=1}^{n^i} B_{q_j^i} | \vec{i} \oplus \vec{\alpha}^{\mathcal{P}_0} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle} = \mathcal{R}_{\mathcal{P},\mathcal{Q}}. \end{aligned} \tag{C24}$$

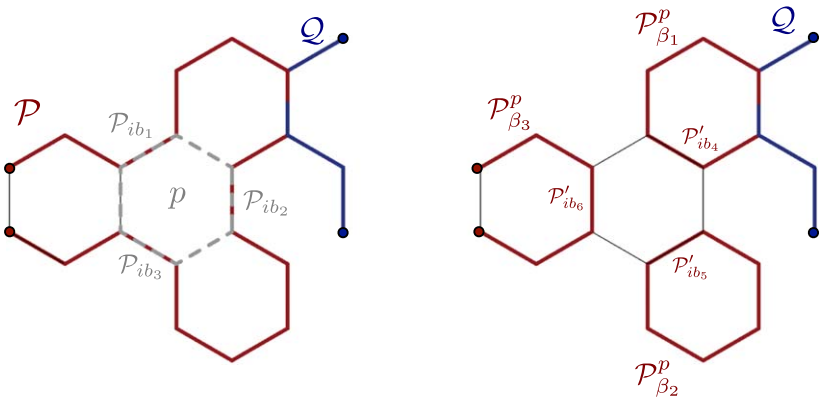

Figure C3. This illustration shows a specific example on path deformation. Three distinct paths ($\{\mathcal{P}_{\beta_1}, \mathcal{P}_{\beta_2}, \mathcal{P}_{\beta_3}\}$, in dark red) are created by applying a plaquette p (in dark blue) to the original path $\mathcal{P}$(dark red). $\mathcal{P}_{ib_1}$, $\mathcal{P}_{ib_2}$, $\mathcal{P}_{ib_3}$ are individual paths which give the parts of path $\mathcal{P}$ that overlaps with plaquette p. $\mathcal{P}'_{ib_4}$, $\mathcal{P}'_{ib_5}$, $\mathcal{P}'_{ib_6}$ are the individual paths which complete plaquette p and are contained in the newly formed paths.

A similar reasoning allows one to add the remaining paths $\mathcal{P}_{m+1}$, $\mathcal{Q}_0$ and $\mathcal{Q}_{n+1}$, to finally find that

$$\mathcal{R}_{\mathcal{P}',\mathcal{Q}'} = \mathcal{R}_{\mathcal{P},\mathcal{Q}}. \tag{C25}$$

□

Lemma 13. *Let $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$ and $\mathcal{Q} = \mathcal{Q}_1\#...\#\mathcal{Q}_n$ be two crossing paths. Assume further that path $\mathcal{P}$ is not self-crossing. Consider a set of distinct paths $\{\mathcal{P}^p_\beta\}$ differing from path $\mathcal{P}$ by a plaquette p, i.e. $\vec{\alpha}^{\mathcal{P}} = \bigoplus_\beta \vec{\alpha}^{\mathcal{P}^p_\beta} \oplus \vec{\alpha}^p$, with the plaquette p not containing the vertices at the endpoint of path $\mathcal{Q}$ and such that every $\mathcal{P}^p_\beta$ is composed of non self-overlapping nor self-crossing individual open paths. Then we have that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_\beta \mathcal{R}_{\mathcal{P}^p_\beta,\mathcal{Q}}$.*

Proof. First notice that since paths $\mathcal{P}$ and $\mathcal{Q}$ cross, we have that paths $\mathcal{P}^p_\beta$ and $\mathcal{Q}$ also cross, since the set of paths $\{\mathcal{P}^p_\beta\}$ and $\mathcal{P}$ differs only by a plaquette, which cannot change its endpoints, and since the plaquette p does not contain the endpoints of path $\mathcal{Q}$.

We can decompose the path $\mathcal{P}_p$ of the plaquette in a series of small paths $\mathcal{P}_{b_1}\#...\#\mathcal{P}_{b_k}$ with $k \leqslant 6$ which are not self-overlapping nor self-crossing. Furthermore, using lemma 11, we can assume without loss of generality that the parts of path $\mathcal{P}$ that overlap with plaquette p are given by individuals paths $\mathcal{P}_{i_{b_1}}, \ldots, \mathcal{P}_{i_{b_l}}$, while the rest of the plaquette is given by individual paths $\mathcal{P}'_{b_{l+1}}, \ldots, \mathcal{P}'_{b_k}$ (see figure C3). Note that the various individual paths $\mathcal{P}_{i_{b_1}}, \ldots, P_{i_{b_l}}$ (as well as $\mathcal{P}'_{i_{b_{l+1}}}, \ldots, \mathcal{P}'_{i_{b_k}}$)need not be adjacent to each other. We note that in order for $\{\mathcal{P}^p_\beta\}$ to contain more than one path, the path $\mathcal{P}$ must have at least two different sequences of individual paths in $\{\mathcal{P}_{i_{b_1}}, \ldots, \mathcal{P}_{i_{b_l}}\}$ such that they are separated by some paths in $\{\mathcal{P}'_{i_{b_{l+1}}}, \ldots, \mathcal{P}'_{i_{b_k}}\}$. Given the structure of a plaquette and of the connected region of a path, and using the fact that $\mathcal{P}$ is not self-crossing, we find that the resulting set of paths $\{\mathcal{P}^p_\beta\}$ is such that any path in it does not contain any edge which is in the connected regions of the other ones.

For convenience, we denote the newly formed paths $\mathcal{P}^p_\beta = \mathcal{P}_{\beta,1}\#...\#\mathcal{P}_{\beta,m^\beta}$, and the corresponding paths $\mathcal{P}'_{b_{l+1}}, \ldots, \mathcal{P}'_{b_k}$ that are in $\mathcal{P}^p_\beta$ as $\{\mathcal{P}_{\beta,i_{\beta_1}}, \ldots, \mathcal{P}_{\beta,i_{\beta_{l'}}}\}$, appearing in order. Note that $l' \leqslant k - l$, and its dependance on β has been omitted for the sake of clarity.

We begin by considering $\mathcal{R}^{+,+}_{\mathcal{P},\mathcal{Q}}$ which is given by

$$\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_{i=1}^{m} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} | S^+_{\mathcal{P}[i]} \prod_{j=1}^{m^i} B_{p^i_j} [S^+_{\mathcal{P}[i]}]^\dagger | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} | S^+_{\mathcal{P}[i-1]} \prod_{j=1}^{m^i} B_{p^i_j} [S^+_{\mathcal{P}[i-1]}]^\dagger | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q^i_k} | B_p (\prod_{j=1}^{n^i} B_{q^i_j} B_p) | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q^i_k} | B_p (\prod_{j=1}^{n^i} B_{q^i_j} B_p) | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}, \tag{C26}$$

where we have defined

$$S^{+}_{\mathcal{P}[i]} = \begin{cases} 1 & \text{for} \quad i < i_{b_1}, \\ S^{+}_{\mathcal{P}_{i_{b_1}}} \ldots S^{+}_{\mathcal{P}_{i_{b_j}}} & \text{for} \quad i_{b_j} \leqslant i < i_{b_{j+1}}, j < l \\ S^{+}_{\mathcal{P}_{i_{b_1}}} \ldots S^{+}_{\mathcal{P}_{i_{b_l}}} & \text{for} \quad i \geqslant i_{b_l}, \end{cases} \tag{C27}$$

and where B_p is the plaquette operator associated to plaquette p. We thus get

$$\begin{aligned} \mathcal{R}_{\mathcal{P},\mathcal{Q}} = & \prod_{\substack{i=1 \\ i\notin\{i_{b_1},\ldots,i_{b_l}\}}}^{m} \frac{\left\langle \vec{i} \bigoplus^{i}_{\substack{j=1 \\ j\notin\{i_{b_1},\ldots,i_{b_l}\}}} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} \middle| \prod_{j=1}^{m^i} B_{p^i_j} \middle| \vec{i} \bigoplus^{i}_{\substack{j=1 \\ j\notin\{i_{b_1},\ldots,i_{b_l}\}}} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \bigoplus^{i-1}_{\substack{j=1 \\ j\notin\{i_{b_1},\ldots,i_{b_l}\}}} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} \middle| \prod_{j=1}^{m^i} B_{p^i_j} \middle| \vec{i} \bigoplus^{i-1}_{\substack{j=1 \\ j\notin\{i_{b_1},\ldots,i_{b_l}\}}} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \\ & \times \prod_{i=1}^{n} \frac{\left\langle \vec{i} \bigoplus^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q^i_k} \oplus \vec{\alpha}^{p} \middle| \prod_{j=1}^{n^i} B_{q^i_j} B_p \middle| \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^i} \vec{\alpha}^{q^i_k} \oplus \vec{\alpha}^{p} \middle| \prod_{j=1}^{n^i} B_{q^i_j} B_p \middle| \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle} \\ & \times \prod_{j=1}^{l} F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right) F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{j} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right) \\ & \times \frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}})}, \end{aligned} \tag{C28}$$

where we have implicitly used our freedom in choosing the sets of plaquettes $\mathcal{B}^{\mathcal{Q},a_i}_{\mathcal{P}_i}$, $\mathcal{B}^{\mathcal{P},b_j}_{\mathcal{Q}_j}$ and of adding plaquettes if necessary without affecting the value of $\mathcal{R}_{\mathcal{P},\mathcal{Q}}$ in order to have that on $\mathrm{Conn}(\mathcal{P}_{i_{b_j}})$, we have that $\bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} = \vec{\alpha}^{\mathcal{Q}}$ for any $i \in \{1, \ldots, m\}$ and for any $j \in \{1, \ldots, l\}$, and where we also have modified the set of plaquettes $\mathcal{B}^{\mathcal{P},b_j}_{\mathcal{Q}_j}$ so that on $\mathrm{Conn}(\mathcal{P}_p)$, $\bigoplus_{k=1}^{n^j} \vec{\alpha}^{q^j_k} = \vec{\alpha}^{\mathcal{P}}$ for any $j \in \{1, \ldots, n\}$ and such that $\bigcup_\beta \mathcal{B}^{\mathcal{P}^p_\beta,b_{\beta,j}}_{\mathcal{Q}_j} = \mathcal{B}^{\mathcal{P},b_j}_{\mathcal{Q}_j} \cup \{p\}$.

We next introduce the various individual paths $\mathcal{P}'_{i_{b_{l+1}}}, \ldots, \mathcal{P}'_{i_{b_k}}$ and rearrange the terms in an appropriate order so as to explicitly make appear the various $\mathcal{P}^p_\beta$'s. Note that in order to do so, we used the fact that the various paths do not have support on the other path's connected regions. We get

$$\begin{aligned} \mathcal{R}_{\mathcal{P},\mathcal{Q}} = & \prod_{\beta} \left(\prod_{i=1}^{m^\beta} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}^p_\beta j} \bigoplus_{k=1}^{m^{\beta,i}} \vec{\alpha}^{p^{\beta,i}_k} \middle| S^{+}_{\mathcal{P}^p_\beta[i]} \prod_{j=1}^{m^{\beta,i}} B_{p^{\beta,i}_j} [S^{+}_{\mathcal{P}^p_\beta[i]}]^\dagger \middle| \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}^p_\beta j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}^p_\beta j} \bigoplus_{k=1}^{m^{\beta,i}} \vec{\alpha}^{p^{\beta,i}_k} \middle| S^{+}_{\mathcal{P}^p_\beta[i-1]} \prod_{j=1}^{m^{i}} B_{p^{\beta,i}_j} [S^{+}_{\mathcal{P}^p_\beta[i-1]}]^\dagger \middle| \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}^p_\beta j} \right\rangle} \right) \\ & \times \prod_{\beta} \left(\prod_{i=1}^{n^\beta} \frac{\left\langle \vec{i} \bigoplus^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^{\beta,i}} \vec{\alpha}^{q^{\beta,i}_k} \middle| \prod_{j=1}^{n^{\beta,i}} B_{q^{\beta,i}_j} \middle| \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \bigoplus_{k=1}^{n^{\beta,i}} \vec{\alpha}^{q^{\beta,i}_k} \middle| \prod_{j=1}^{n^{\beta,i}} B_{q^{\beta,i}_j} \middle| \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{Q}_j} \right\rangle} \right) \\ & \times \left(\prod_{j=1}^{l} \frac{F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right)}{F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right)} \right) \frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}})}, \end{aligned} \tag{C29}$$

where we have defined

$$S^{+}_{\mathcal{P}^p_\beta[i]} = \begin{cases} 1 & \text{for} i < i_{\beta_1} \\ S^{+}_{\mathcal{P}_\beta,i_{\beta_1}} \ldots S^{+}_{\mathcal{P}_\beta,i_{\beta_j}} & \text{for} i_{\beta_j} \leqslant i < i_{\beta_{j+1}}, j < l' \\ S^{+}_{\mathcal{P}_\beta,i_{\beta_1}} \ldots S^{+}_{\mathcal{P}_\beta,i_{\beta_l}} & \text{for} i \geqslant i_{\beta_{l'}}, \end{cases} \tag{C30}$$

and where we have $\{p^{\beta,i}_j\} = \mathcal{B}^{\mathcal{Q},a_\beta,i}_{\mathcal{P}_\beta,i}$, $\{q^{\beta,i}_j\} = \mathcal{B}^{\mathcal{P}_{\beta,i},b_{\beta,i}}_{\mathcal{Q}}$.

Under close inspection, it thus becomes clear that

$$\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_{\beta} \mathcal{R}_{\mathcal{P}^p_\beta} \times \left(\prod_{j=1}^{k} \frac{F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right)}{F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right)} \right) \frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}})}. \tag{C31}$$

We first notice that since all the paths $\mathcal{P}_\beta^{p_p}$ and $\mathcal{P}$ cross with path $\mathcal{Q}$, we can define $\mathcal{B}_{\mathcal{P}_p}^{\mathcal{Q}} = \{p_{\mathcal{Q},1}, \ldots, p_{\mathcal{Q},o}\}$, a set of plaquettes in $\mathcal{B}_{\mathcal{P}_p}$ such that on Conn($\mathcal{P}_b$), we have that $\bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} = \vec{\alpha}^{\mathcal{Q}}$. Using the same reasoning as in lemma 9, we find that

$$\frac{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}})}{b_p(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}})} = \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} \middle| \prod_{i=1}^{o} B_{p_{\mathcal{Q},i}} \middle| \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{P}_p} \bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} \middle| \prod_{i=1}^{o} B_{p_{\mathcal{Q},i}} \middle| \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{P}_p} \right\rangle}. \tag{C32}$$

On the other hand, we find that

$$\begin{aligned} \prod_{j=1}^{k} \frac{F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{Q}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right)}{F_{\mathcal{P}_{i_{b_j}}}\left(\vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}}\right)} &= \prod_{j=1}^{k} \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} \bigoplus_{i=1}^{j} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}} \middle| \prod_{i=1}^{o} B_{p_{\mathcal{Q},i}} \middle| \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{j} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}} \middle| \prod_{i=1}^{o} B_{p_{\mathcal{Q},i}} \middle| \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{j-1} \vec{\alpha}^{\mathcal{P}_{i_{b_i}}} \right\rangle} \\ &= \frac{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{P}_p} \bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} \middle| \prod_{i=1}^{o} B_{p_{\mathcal{Q},i}} \middle| \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \oplus \vec{\alpha}^{\mathcal{P}_p} \right\rangle}{\left\langle \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \bigoplus_{i=1}^{o} \vec{\alpha}^{p_{\mathcal{Q},i}} \middle| \prod_{i=1}^{o} B_{p_{\mathcal{Q},i}} \middle| \vec{i} \oplus \vec{\alpha}^{\mathcal{P}} \right\rangle}. \end{aligned} \tag{C33}$$

Putting equations (C31)–(C33) together, we conclude that

$$\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_{\beta} \mathcal{R}_{\mathcal{P}_\beta^{p_p}}. \tag{C34}$$

To finally conclude the proof, we remark that according to lemma 11 we are free to modify the composition in terms of individual paths of the various $\{\mathcal{P}_\beta^{p_p}\}$'s as long as the their individual paths remain non self-overlapping nor self-crossing. □

Lemma 14. *Let $\mathcal{P}$ and $\mathcal{Q}$ be two crossing paths such that each of them is made of simple non self-overlapping nor self-crossing open paths, and such that $\mathcal{P}$ is not self-crossing. If they cross an even number of times, then $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = 1$.*

Proof. The idea of the proof is to deform the path $\mathcal{P}$ using the results of lemma 13, as well as path $\mathcal{Q}$ by some elongation and reductions using lemma 12, in order to get a set of paths $\{\mathcal{P}_\beta\}$ and path $\mathcal{Q}'$ such that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_\beta \mathcal{R}_{\mathcal{P}_\beta,\mathcal{Q}'}$, and such that all of them are outside of Conn($\mathcal{Q}'$), thus implying that $R_{\mathcal{P}_\beta,\mathcal{Q}} = 1$.

We first note that if $\mathcal{P}$ and $\mathcal{Q}$ do not have any edges in common, then they trivially commute, since by supposition they are crossing each other. This implies that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = 1$.

Consider the case where $\mathcal{P}$ and $\mathcal{Q}$ have some edges in common. Suppose first that $\mathcal{P}$ and $\mathcal{Q}$ have a single contiguous set of common edges, $\mathcal{E}$. In that case, the path $\mathcal{P}$ can be sequentially deformed along the set of contiguous plaquettes in Conn($\mathcal{Q}$) containing the edges in $\mathcal{E}$ as well as the two edges in $\mathcal{P}\backslash\mathcal{E}$ sharing vertices with the edges in $\mathcal{E}$ in order to give a new sets of paths $\{\mathcal{P}_\beta\}$. Note that since $\mathcal{P}$ and $\mathcal{Q}$ cross 0 times, we can assume that none of the paths in $\{\mathcal{P}_\beta\}$ contains edges in Conn($\mathcal{Q}$). If it is not the case, then we can use lemma 12 to first find a shorter path $\mathcal{Q}'$ such that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \mathcal{R}_{\mathcal{P},\mathcal{Q}'}$ and for which it is true. By suitably choosing a decomposition of those plaquettes such that they are non self-overlapping nor self-crossing, we can use lemma 13, to find that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_\beta \mathcal{R}_{\mathcal{P}_\beta,\mathcal{Q}} = 1$.

Suppose next that $\mathcal{P}$ and $\mathcal{Q}$ have two or more different contiguous sets of common edges $\{\mathcal{E}_i\}$. In that case, we can deform $\mathcal{P}$ by a subset of plaquettes in $\mathcal{B}_\mathcal{Q}$ so as to form a single contiguous set of common edges $\mathcal{E}$ such that $\mathcal{E} \cup \delta\mathcal{E} \supset \bigcup_i \mathcal{E}_i$, where $\delta\mathcal{E}$ denotes the first and last edges in $\mathcal{Q}$ which are also in $\mathcal{E}$, and such that all newly formed paths $\{\mathcal{P}_\beta\}$ cross $\mathcal{Q}$, where we have again chosen a suitable decomposition of the paths along the plaquettes. Again using lemma 13 and considering a shortened path $\mathcal{Q}'$ using lemma 12 if necessary, we find that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_\beta \mathcal{R}_{\mathcal{P}_\beta,\mathcal{Q}'}$, where one of the $\mathcal{P}_\beta$'s have a single set of edges ($\mathcal{E}$) in common with $\mathcal{Q}'$, and where the other paths cross 0 times with $\mathcal{Q}'$. Since modifying the path $\mathcal{P}$ by a set of plaquettes cannot change the parity of the number of crossing, we have that the former path crosses 0 times with $\mathcal{Q}'$ as well. By the previous reasoning, we thus find that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = 1$. □

Lemma 15. *Let $\mathcal{P}$ and $\mathcal{Q}$ be two crossing paths such that each of them is made of simple non self-overlapping nor self-crossing open paths, and such that $\mathcal{P}$ is not self-crossing. If they cross an* odd *number of times, then $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = -1$.*

Proof. The idea of the proof closely follows that of lemma 14. We begin by sequentially deforming the path $\mathcal{P}$ into a set of paths $\{\mathcal{P}_\beta\}$ such that all of them are crossing with path $\mathcal{Q}$, and such that there is a single set of contiguous edges between one of the path $\mathcal{P}_o \in \{\mathcal{P}_\beta\}$ and $\mathcal{Q}$, and such that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \prod_\beta \mathcal{R}_{\mathcal{P}_\beta,\mathcal{Q}'}$, where $\mathcal{Q}'$ may be a shortened path of $\mathcal{Q}$, as described in the proof of lemma 14. Since any path $\mathcal{P}_\beta \neq \mathcal{P}_o$ crosses path $\mathcal{Q}'$ 0 times, we have that $\mathcal{R}_{\mathcal{P}_\beta,\mathcal{Q}} = 1$ for $\beta \neq o$. We thus have that $\mathcal{R}_{\mathcal{P},\mathcal{Q}} = \mathcal{R}_{\mathcal{P}_o,\mathcal{Q}'}$.

In order to calculate $\mathcal{R}_{\mathcal{P}_o,\mathcal{Q}'}$, we note that we can first sequentially deform path $\mathcal{P}_o$ as described previously so that there is a single common edge between the two paths, and we can use lemma 12 to bring the endpoints of the

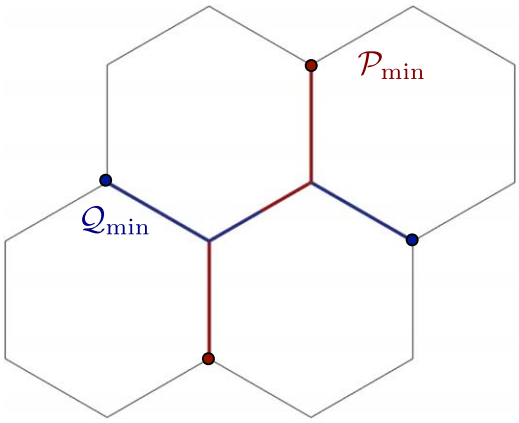

Figure C4. The quantity $\mathcal{R}_{\mathcal{P},\mathcal{Q}}$ for 2 paths $\mathcal{P}$ and $\mathcal{Q}$ crossing an odd number of times is equal to $\mathcal{R}_{\mathcal{P}_{\min},\mathcal{Q}_{\min}}$, for the minimal paths $\mathcal{P}_{\min}$ and $\mathcal{Q}_{\min}$ shown above.

two paths as close as possible in order to minimize the length of the paths. We thus find that computing $\mathcal{R}_{\mathcal{P},\mathcal{Q}}$ reduces to computing this quantity for a single minimal path configurations $\mathcal{P}_{\min}$ and $\mathcal{Q}_{\min}$ illustrated in figure C4. Explicit calculations using equation (C3) for a single underlying string configuration (lemma 10 ensures that its value is independent of the configuration), gives that $\mathcal{R}_{\mathcal{P}_{\min},\mathcal{Q}_{\min}} = -1$. □

C.2. Proof of theorem 2

Having introduced all previous technical lemmas, we are in a position to complete the demonstration of theorem 2.

Theorem 2. *Let $\mathcal{P}$ and $\mathcal{Q}$ be two paths crossing n times, composed of non self-overlapping nor self-crossing individual open paths. We have that*

$$[S^+_{\mathcal{P}},\, S^+_{\mathcal{Q}}] = 0 \text{ if } n \text{ is even},$$
$$\{S^+_{\mathcal{P}},\, S^+_{\mathcal{Q}}\} = 0 \text{ if } n \text{ is odd}. \tag{C35}$$

Proof. Consider first the case where path $\mathcal{P}$ is not self-crossing. Given the definition in equation (C3) of $\mathcal{R}_{\mathcal{P},\mathcal{Q}}$, lemma 14 shows that $[S^+_{\mathcal{P}},\, S^+_{\mathcal{Q}}] = 0$ if n is even, while lemma 15 shows that $\{S^+_{\mathcal{P}},\, S^+_{\mathcal{Q}}\} = 0$ is n is odd.

Consider next the case where $\mathcal{P}$ is self-crossing. Using lemma 11, we can always find paths $\mathcal{P}_1, \ldots, \mathcal{P}_m$ for which $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$, such that none of those are self-crossing, with $\mathcal{R}_{\mathcal{P}_1\#...\#\mathcal{P}_m,\mathcal{Q}} = \mathcal{R}_{\mathcal{P},\mathcal{Q}}$. Suppose that all those paths cross with $\mathcal{Q}$. Since $S^+_{\mathcal{P}_1\#...\#\mathcal{P}_m} = S^+_{\mathcal{P}_m}...S^+_{\mathcal{P}_1}$, it suffices to know the commutation relations between every of the operators $S^+_{\mathcal{P}_1}, \ldots, S^+_{\mathcal{P}_m}$ and $S^+_{\mathcal{Q}}$. Since none of the corresponding paths are self-overlapping, the reasoning of the above paragraph can be used to find the same result.

It may be impossible to decompose path $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_m$ such that all of its components cross with $\mathcal{Q}$. This happens only in the case where some edges in path $\mathcal{P}$ appear at more than one position. In that case, suppose for simplicity that path $\mathcal{P} = \mathcal{P}_1\#...\#\mathcal{P}_{l-1}\#\mathcal{P}_l\#\mathcal{P}_{l+1}\#...\#\mathcal{P}_{o-1}\#\mathcal{P}_o\#\mathcal{P}_{o+1}\#...\#\mathcal{P}_m$ can be decomposed such that pahts $\mathcal{P}_l = \mathcal{P}_o$ are the only ones with edges in common (possibly in reversed order). The following reasoning works in the same way if there are more than a single pair of such paths. Consider the quantity

$$\begin{aligned}
&\prod_{i=1}^{m} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} | S^+_{\mathcal{P}[i]} \prod_{j=1}^{m^i} B_{p^i_j} S^+_{\mathcal{P}[i]} | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} | S^+_{\mathcal{P}[i-1]} \prod_{j=1}^{m^i} B_{p^i_j} S^+_{\mathcal{P}[i-1]} | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \\
&= \prod_{\substack{i=1 \\ i\neq l,o}}^{m} \frac{\left\langle \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} | \prod_{j=1}^{m^i} B_{p^i_j} | \vec{i} \bigoplus_{j=1}^{i} \vec{\alpha}^{\mathcal{P}_j} \right\rangle}{\left\langle \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} | \prod_{j=1}^{m^i} B_{p^i_j} | \vec{i} \bigoplus_{j=1}^{i-1} \vec{\alpha}^{\mathcal{P}_j} \right\rangle} \\
&\times \prod_{i=l}^{o-1} \frac{F_{\mathcal{P}_l}(\vec{i} \bigoplus_{j=1}^{i} \alpha^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} \oplus \vec{\alpha}^{\mathcal{P}_l})}{F_{\mathcal{P}_l}(\vec{i} \bigoplus_{j=1}^{i} \alpha^{\mathcal{P}_j} \bigoplus_{k=1}^{m^{i+1}} \vec{\alpha}^{p^{i+1}_k} \oplus \vec{\alpha}^{\mathcal{P}_l})},
\end{aligned} \tag{C36}$$

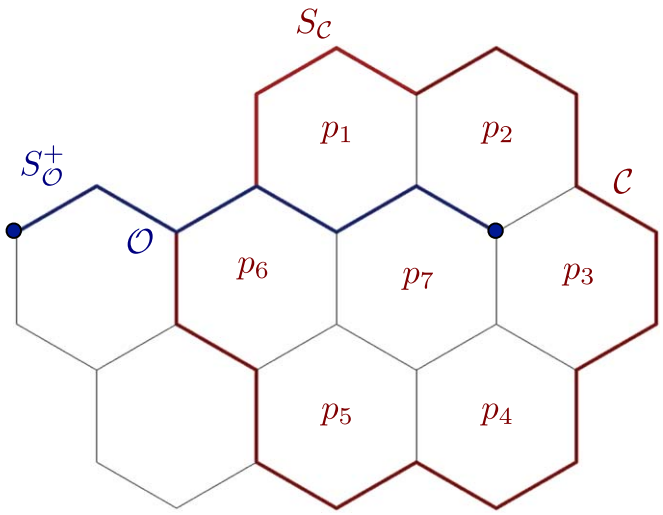

Figure C5. A closed string, $S_{\mathcal{C}}$, and a positive-chirality open string, $S^+_{\mathcal{O}}$, where paths $\mathcal{O}$ and $\mathcal{C}$ cross once. $\vec{\alpha}^{\mathcal{C}}$ can be written in this particular example as $\vec{\alpha}^{\mathcal{C}} = \vec{\alpha}^{p_1} \oplus ... \oplus \vec{\alpha}^{p_6}$, restricted to Conn$(\mathcal{C})$. The multiplication of plaquettes inside $\mathcal{C}$, i.e. $B_{p_1} ... B_{p_7}$ is a negative-chirality string in Conn$(\mathcal{C})$.

where

$$S^+_{\mathcal{P}[i]} = \begin{cases} S^+_{\mathcal{P}_l} & \text{if} l \leqslant i < o \\ 1 & \text{otherwise.} \end{cases} \tag{C37}$$

Using lemma 11, for any $i \in \{l, \ldots, o\}$ it is always possible to find a path decomposition and sets of plaquettes $\mathcal{B}^{\mathcal{Q},a_i}_{\mathcal{P}_i}$ such that on Conn$(\mathcal{P}_l)$, $\bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} = \vec{\alpha}^{\mathcal{Q}}$. This can simply be achieved by taking individual paths of lenght 1 in the decomposition of path $\mathcal{P}$. We thus find that

$$\prod_{i=l}^{o-1} \frac{F_{\mathcal{P}_l}(\vec{i} \bigoplus_{j=1}^{i} \alpha^{\mathcal{P}_j} \bigoplus_{k=1}^{m^i} \vec{\alpha}^{p^i_k} \oplus \vec{\alpha}^{\mathcal{P}_l})}{F_{\mathcal{P}_l}(\vec{i} \bigoplus_{j=1}^{i} \alpha^{\mathcal{P}_j} \bigoplus_{k=1}^{m^{i+1}} \vec{\alpha}^{p^{i+1}_k} \oplus \vec{\alpha}^{\mathcal{P}_l})} = 1, \tag{C38}$$

which, given equation (C4), leads us to the conclusion that

$$\mathcal{R}_{\mathcal{P}_1\#...\#\mathcal{P}_{l-1}\#\mathcal{P}_l\#\mathcal{P}_{l+1}\#...\#\mathcal{P}_{o-1}\#\mathcal{P}_o\#\mathcal{P}_{o+1}\#...\#\mathcal{P}_m,\mathcal{Q}} = \mathcal{R}_{\mathcal{P}_1\#...\#\mathcal{P}_{l-1}\#\mathcal{P}_{l+1}\#...\#\mathcal{P}_{o-1}\#\mathcal{P}_{o+1}\#...\#\mathcal{P}_m,\mathcal{Q}}. \tag{C39}$$

To complete the proof, it suffices to notice that we can now apply the reasoning of the first two paragraphs of this proof. □

C.3. The need for path concatenation

Each time algorithm 1 picks a configuration representative $\vec{i}$, an initial phase must be chosen. All of these choices yield valid string operators. One may wonder what is the physical difference between different choices. To answer this question, note that two string operators $S^+_{\mathcal{P}}$ and $\tilde{S}^+_{\mathcal{P}}$ obtained by a different choice of phases in algorithm 1 are related through

$$\tilde{S}^+_{\mathcal{P}} = S^+_{\mathcal{P}} \sum_{c \in \mathcal{C}_{\mathcal{P}}} e^{i\theta_c} P_c, \tag{C40}$$

where $\mathcal{C}_{\mathcal{P}}$ denotes the set of all configuration classes of path $\mathcal{P}$, $P_c = \sum_{\vec{i} \in c} |\vec{i}\rangle \langle \vec{i}|$ is the projector on the states of configuration class c, and $e^{i\theta_c}$ is an independent arbitrary complex phase for every class configuration.

Equivalently, given a string operator $S^+_{\mathcal{P}}$, we can obtain another string operator $\tilde{S}^+_{\mathcal{P}}$ multiplying $S^+_{\mathcal{P}}$ by an operator $S^z_{\mathcal{C}_{\text{dual}}}$, where $\mathcal{C}_{\text{dual}}$ is a closed loop in the dual lattice affecting only qubits in Conn$(\mathcal{P})$. Since $S^z_{\mathcal{C}_{\text{dual}}}$ is a loop, it may be expressed as a multiplication of vertex operators (unless it is a non-trivial loop, which may happen for closed strings). Therefore $\tilde{S}^+_{\mathcal{P}}$ still commutes with all plaquette and vertex operators and it is contained in Conn$(\mathcal{P})$, satisfying properties (i) and (ii). It is also possible to obtain another string operator multiplying $S^+_{\mathcal{P}}$ by a linear combination of S^z operators on closed loops, i.e.

$$\tilde{S}^+_{\mathcal{P}} = S^+_{\mathcal{P}}(\mathbb{I} + c(\vec{\alpha}^{\mathcal{C}_1}) S^z_{\mathcal{C}_1} + c(\vec{\alpha}^{\mathcal{C}_2}) S^z_{\mathcal{C}_2} + ...), \tag{C41}$$

where $c(\vec{\alpha}^{\mathcal{C}}_i)$ are coefficients associated with the closed-string operator $S^z_{\mathcal{C}}$ and $\mathcal{C}_1, ..., \mathcal{C}_n$ are closed paths in the dual lattice contained in Conn$(\mathcal{P})$. For any two strings generated by algorithm 1 differing only in the choice of initial phases, we can always find a relation of the form given by equation (C41).

C.3.1. Closed-string operators. If we consider now a closed path, $\mathcal{C}$, and we use algorithm 1 to find a closed-string operator, we will not find in general a positive-chirality nor a negative-chirality string, but some mixing of both. Physically, this is caused by the fact that for a closed string there is no difference in the pattern of plaquette violations between positive- and negative-chirality strings, since there are no endpoints. From equation (C41) we can see that starting from a positive-chirality string, $S_{\mathcal{C}}^{+}$, it is possible to add S^z operators forming a loop which cannot be expressed as the product of vertex operators (which we call a non trivial loop in the remainder of the section) to the linear superposition. Once this is done, the resulting operator, $\tilde{S}_{\mathcal{C}}$, does not have a well-defined chirality. Remember, that for an open string, we may obtain the negative-chirality string by multiplying it by an open string S^z, violating the plaquettes at the endpoints. For closed strings we may proceed analogously to obtain the opposite chirality by multiplying by a non-trivial closed string $S_{\mathcal{C}'}^{z}$. Since we are multiplying $S_{\mathcal{C}}^{+}$ by a linear combination of trivial and non-trivial loops of S^z, the chirality of $\tilde{S}_{\mathcal{C}}$ is no longer positive nor negative. Thus, we drop the '+' superscript in $\tilde{S}_{\mathcal{C}}$.

The mixing of chiralities becomes apparent when computing the commutator between a closed string, $S_{\mathcal{C}}$, and a positive-chirality open string, $S_{\mathcal{O}}^{+}$, where paths $\mathcal{C}$ and $\mathcal{O}$ cross once (see figure C5):

$$\mathcal{R}_{\mathcal{O},\mathcal{C}} = \langle \vec{i} \; | [S_{\mathcal{O}}^{+}, S_{\mathcal{C}}] | \vec{i} \rangle = \frac{F_{\mathcal{O}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{C}}) F_{\mathcal{C}}(\vec{i})}{F_{\mathcal{O}}(\vec{i}) F_{\mathcal{C}}(\vec{i} \oplus \vec{\alpha}^{\mathcal{O}})}, \tag{C42}$$

where here $[\cdot,\cdot]$ is the commutator in the group sense, i.e. $[g,\ h] = g^{-1}h^{-1}gh$. In this case, notice that on $\mathrm{Conn}(\mathcal{C})$, the configurations $\vec{i}$ and $\vec{i} \oplus \vec{\alpha}^{\mathcal{O}}$ do not belong to the same class. Using equation (C40), it is thus clear that $\mathcal{R}_{\mathcal{O},\mathcal{C}}$ can be modified by selecting different phases to initialize $F_{\mathcal{C}}$ in algorithm 1.

References

[1] Kitaev A Y 2003 Fault-tolerant quantum computation by anyons *Ann. Phys.* **303** 2–30
[2] Dennis E, Kitaev A, Landahl A and Preskill J 2002 Topological quantum memory *J. Math. Phys.* **43** 4452–505
[3] Bravyi S B and Kitaev A Y 1998 Quantum codes on a lattice with boundary arXiv:quant-ph/9811052
[4] Bombin H and Martin-Delgado M A 2006 Topological quantum distillation *Phys. Rev. Lett.* **97** 180501
[5] Bombin H and Martin-Delgado M A 2007 Topological computation without braiding *Phys. Rev. Lett.* **98** 160502
[6] Bombin H and Martin-Delgado M A 2007 Homological error correction: classical and quantum codes *J. Math. Phys.* **48** 052105
[7] Bombin H and Martin-Delgado M A 2007 Optimal resources for topological two-dimensional stabilizer codes: comparative study *Phys. Rev.* A **76** 012305
[8] Shor P W 1995 Scheme for reducing decoherence in quantum computer memory *Phys. Rev.* A **52** R2493–6
[9] Steane A M 1996 Error correcting codes in quantum theory *Phys. Rev. Lett.* **77** 793–7
[10] Shor P W 1996 Fault-tolerant quantum computation *37th Symposium on Foundations of Computing* (Piscataway, NJ: IEEE) pp 56–65
[11] Knill W Z E and Laflamme R 1996 Threshold accuracy for quantum computation arXiv:quant-ph/9610011
[12] Kitaev A Y 1997 Quantum computations: algorithms and error correction *Russ. Math. Surv.* **52** 1191–249
[13] Aharonov M B-O D 1999 Fault tolerant quantum computation with constant error *SIAM J. Comp.* **38** 1207–82
[14] Nielsen M A and Chuang I L 2000 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press)
[15] Galindo A and Martín-Delgado M A 2002 Information and computation: classical and quantum aspects *Rev. Mod. Phys.* **74** 347–423
[16] Levin M A and Wen X-G 2005 String-net condensation: a physical mechanism for topological phases *Phys. Rev.* B **71** 045110
[17] Freedman M, Nayak C, Shtengel K, Walker K and Wang Z 2004 A class of p,t-invariant topological phases of interacting electrons *Ann. Phys.* **310** 428–92
[18] Freeman M H and Hastings M B 2016 Double semions in arbitrary dimension *Commun. Math. Phys.* **347** 389–419
[19] Furusaki A 2017 Weyl points and Dirac lines protected by multiple screw rotations *Sci. Bull.* **62** 788–994
[20] Li K, Wan Y, Hung L-Y, Lan T, Long G, Lu D, Zeng B and Laflamme R 2017 Experimental identification of non-abelian topological orders on a quantum simulator *Phys. Rev. Lett.* **118** 080502
[21] Syed R, Sirota A and Teo J C Y 2019 From Dirac semimetrals to topological phases in three dimensions: a coupled-wire construction *Phys. Rev.* X **9** 011039
[22] Buerscher O, Morampudi S C and Pollmann F 2014 Double semion phase in an exactly solvable quantum dimer model on the kagome lattice *Phys. Rev.* B **90** 195148
[23] Morampudi S C, von Keyserlingk C and Pollmann F 2014 Numerical study of a transition between $\mathbb{Z}_2$ topologically ordered phases *Phys. Rev.* B **90** 035117
[24] Ors R, Wei T-C, Buerschaper O and Van den Nest M 2014 Geometric entanglement in topologically ordered states *New J. Phys.* **16** 013015
[25] Qi Y, Gu Z-C and Yao H 2015 Double-semion topological order from exactly solvable quantum dimer models *Phys. Rev.* B **92** 155105
[26] Mesaros A and Ran Y 2013 Classification of symmetry enriched topological phases with exactly solvable models *Phys. Rev.* B **87** 155115
[27] von Keyserlingk C W, Burnell F J and Simon S H 2013 Three-dimensional topological lattice models with surface anyons *Phys. Rev.* B **87** 045107
[28] Calderbank A R and Shor P W 1996 Good quantum error-correcting codes exist *Phys. Rev.* A **54** 1098–105
[29] Steane A 1996 Multiple-particle interference and quantum error correction *Proc. R. Soc.* A **452** 2551–77
[30] Preskill J 2004 *Lecture Notes for Physics 219: Quantum Computation*
[31] Terhal B M 2015 Quantum error correction for quantum memories *Rev. Mod. Phys.* **87** 307–46
[32] Gottesman D 1996 Class of quantum error-correcting codes saturating the quantum hamming bound *Phys. Rev.* A **54** 1862–8
[33] Ni X, Buerschaper O and Van den Nest M 2015 A non-commuting stabilizer formalism *J. Math. Phys.* **56** 052201
[34] Calderbank A R, Rains E M, Shor P W and Sloane N J A 1997 Quantum error correction and orthogonal geometry *Phys. Rev. Lett.* **78** 405–8
[35] Kitaev A 2006 Anyons in an exactly solved model and beyond *Ann. Phys.* **321** 2–111

[36] Pfeifer R N, Buerscher O, Trebst S, Ludwig A W W, Troyer M and Vidal G 2012 Translation invariance, topology, and protection of criticality un chains of interacting anyons *Phys. Rev.* B **86** 155111
[37] Nakahara N 2003 *Geometry, Topology and Physics* (London: Taylor and Francis)
[38] Bombin H and Martin-Delgado M A 2007 Homological error correction: classical and quantum codes *J. Math. Phys.* **48** 052105
[39] Wang D S, Fowler A G, Stephens A M and Hollenberg L C L 2010 Threshold error rates for the toric and planar codes *Quantum Inf. Comput.* **10** 456–69
[40] Duclos-Cianci G and Poulin D 2010 Fast decoders for topological quantum codes *Phys. Rev. Lett.* **104** 050504
[41] Bravyi S and Haah J 2013 Quantum self-correction in the 3d cubic code model *Phys. Rev. Lett.* **111** 200501
[42] Fowler A G, Whiteside A C and Hollenberg L C L 2012 Towards practical classical processing for the surface code *Phys. Rev. Lett.* **108** 180501
[43] Anwar H, Brown B J, Campbell E T and Browne D E 2014 Fast decoders for qudit topological codes *New J. Phys.* **16** 063038
[44] Delfosse N and Zmor G 2017 Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel arXiv:1703.01517
[45] Delfosse N and Nickerson N H 2017 Almost-linear time decoding algorithm for topological codes arXiv:1709.06218
[46] Maskara N, Kubica A and Jochym-O'Connor T 2018 Advantages of versatile neural-network decoding for topological codes arXiv:1802.08680
[47] Kubica A and Preskill J 2018 Cellular-automaton decoders with provable thresholds for topological codes arXiv:1809.10145
[48] Herold M, Campbell E T, Eisert J and Kastoryano M J 2015 Cellular-automaton decoders for topological quantum memories *Npj Quantum Inf.* **1** 15010
[49] Wootton J 2015 A simple decoder for topological codes *Entropy* **17** 1946–57
[50] Sarvepalli P and Raussendorf R 2012 Efficient decoding of topological color codes *Phys. Rev.* A **85** 022317
[51] Harrington J 2004 Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes *PhD Thesis* Caltech
[52] Chamberland C and Ronagh P 2018 Deep neural decoders for near term fault-tolerant experiments *Quantum Sci. Technol.* **3** 044002
[53] Herold M, Kastoryano M J, Campbell E T and Eisert J 2017 Cellular automaton decoders of topological quantum memories in the fault tolerant setting *New J. Phys.* **19** 063012
[54] Dauphinais G and Poulin D 2017 Fault-tolerant quantum error correction for non-abelian anyons *Commun. Math. Phys.* **355** 519–60
[55] Sweke R, Kesselring M S, van Nieuwenburg E P L and Eisert J 2018 Reinforcement learning decoders for fault-tolerant quantum computation arXiv:1810.07207
[56] Breuckmann N P, Duivenvoorden K, Michels D and Terhal B M 2017 Local decoders for the 2d and 4d toric code *Quantum Inf. Comput.* **17** 0181
[57] Breuckmann N P and Ni X 2018 Scalable neural network decoders for higher dimensional quantum codes *Quantum* **2** 68
[58] Wootton J R and Loss D 2012 High threshold error correction for the surface code *Phys. Rev. Lett.* **109** 160503
[59] Hutter A, Wootton J R and Loss D 2014 Efficient Markov chain Monte Carlo algorithm for the surface code *Phys. Rev.* A **89** 022326
[60] Bravyi S, Suchara M and Vargo A 2014 Efficient algorithms for maximum likelihood decoding in the surface code *Phys. Rev.* A **90** 032326
[61] Darmawan A S and Poulin D 2018 Linear-time general decoding algorithm for the surface code *Phys. Rev.* E **97** 051302
[62] Bombin H, Andrist R S, Ohzeki M, Katzgraber H G and Martin-Delgado M A 2012 Strong resilience of topological codes to depolarization *Phys. Rev.* X **2** 021004
[63] Tuckett D K, Bartlett S D and Flammia S T 2018 Ultrahigh error threshold for surface codes with biased noise *Phys. Rev. Lett.* **120** 050505
[64] Laflamme R, Miquel C, Paz J P and Zurek W H 1996 Perfect quantum error correcting code *Phys. Rev. Lett.* **77** 198–201
[65] DiVincenzo D P and Shor P W 1996 Fault-tolerant error correction with efficient quantum codes *Phys. Rev. Lett.* **77** 3260–3
[66] Preskill J 1997 Fault-tolerant quantum computation *Introduction to Quantum Computation and Information* ed H-K Lo, T Spiller and S Popescu (Singapore: World Scientific)
[67] Levin M and Gu Z-C 2012 Braiding statistics approach to symmetry-protected topological phases *Phys. Rev.* B **86** 115109
[68] Ortiz L and Martin-Delgado M A 2016 A bilayer double semion model with symmetry-enriched topological order *Ann. Phys.* **375** 193–226
[69] Song H, Prem A, Huang S-J and Martin-Delgado M A 2018 Twisted fracton models in three dimensions *Phys. Rev.* B **99** 155118