

# Dynamical Logical Qubits in the Bacon-Shor Code

M. Sohaib Alam\*

*Quantum Artificial Intelligence Laboratory (QuAIL),  
NASA Ames Research Center, Moffett Field, CA, 94035, USA and  
USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, CA, 94043, USA*

Eleanor Rieffel†

*Quantum Artificial Intelligence Laboratory (QuAIL),  
NASA Ames Research Center, Moffett Field, CA, 94035, USA  
(Dated: March 7, 2024)*

The Bacon-Shor code is a quantum error correcting subsystem code composed of weight 2 check operators that admits a single logical qubit, and has distance  $d$  on a  $d \times d$  square lattice. We show that when viewed as a Floquet code, by choosing an appropriate measurement schedule of the check operators, it can additionally host several dynamical logical qubits. Specifically, we identify a period 4 measurement schedule of the check operators that preserves logical information between the instantaneous stabilizer groups. Such a schedule measures not only the usual stabilizers of the Bacon-Shor code, but also additional stabilizers that protect the dynamical logical qubits against errors. We show that the code distance of these Floquet-Bacon-Shor codes scales as  $\Theta(d/\sqrt{k})$  on a  $d \times d$  lattice with  $k$  dynamical logical qubits, along with the logical qubit of the parent subsystem code. Moreover, several errors are shown to be self-corrected purely by the measurement schedule itself.

## I. INTRODUCTION

Quantum computers offer a promising avenue for solving computational problems that may be intractable for classical computers. However, in order to be practically useful, one must correct for errors that accrue over the course of the computation. Quantum error correction (QEC) provides a broad framework in which the logical information is encoded on part of the full physical Hilbert space, and can be protected against errors through measurements of appropriate syndrome observables followed by post-measurement correction operations. A leading candidate for QEC has been the surface code [1–3], for its relatively high threshold as well as a simple square lattice architecture.

Recently, a remarkable class of QEC codes was introduced by Hastings and Haah [4] in which the logical degrees of freedom do not form a fixed subspace of the physical Hilbert space, but rather evolve dynamically. The first such example involved the Kitaev honeycomb model [5], which when viewed as a subsystem code, encodes no logical qubits. However, by choosing an appropriate measurement schedule of the weight-2 check operators, it was shown that this model leads to dynamical logical degrees of freedom. Due to the periodicity of the measurement schedule and the induced instantaneous stabilizer groups (ISGs), such codes are typically referred to as Floquet codes.

The original honeycomb Floquet code [4] was defined on a hexagonal torus, while generalizations to 3-colorable and 3-valent planar graphs were explored in

[6]. Due to a non-trivial automorphism between electric and magnetic operators across measurement cycles, adding boundaries to the model is non-trivial, but were introduced in [7]. Benchmarking studies [8–10] showed a competitive threshold for the model, with at least one small scale experiment[11] demonstrating stabilizer measurements. Since then, several examples of Floquet codes have been constructed, including those without parent subsystem codes[12], those involving the color code[13, 14], constructions based on hyperbolic geometries[15, 16], in 3D Euclidean space based on fracton order[17], rewinding the measurement schedule[18], as well as introducing twists in the geometry[19]. Generalizations of Floquet code constructions have also been explored using the ZX calculus[20, 21], adiabatic paths[22], anyon condensation[13], as well as aperiodic measurement schedules in the form of dynamic automorphism codes[23].

While the dynamics of previously constructed Floquet codes typically requires a description of the embedded toric codes in its ISGs, and may perhaps be intuitively understood through adiabatic paths [22] or anyon condensation [13], the construction of the Floquet code we describe here follows directly from the subsystem structure of the parent Bacon-Shor code, without the need to move to a condensed matter perspective. In this sense, it hopefully offers a comparatively simpler example of a Floquet code. It also comes naturally equipped with a boundary, without the complication of non-trivial automorphisms between electric and magnetic operators [4, 6, 7]. Furthermore, this Floquet code is naturally defined on a square lattice, whose simple lattice structure may be appealing from an experimental perspective. To our knowledge, the model introduced here provides a first example of a Floquet code on a square lattice, though it

\* sohaib.alam@nasa.gov

† eleanor.rieffel@nasa.gov

has been noted that a square lattice construction may also be possible using an alternative approach[24]. It is an example of a CSS Floquet code[12, 13], since all the stabilizers are of either  $X$  or  $Z$  type. Moreover, unlike the original honeycomb code[4], it also serves as an example of a Floquet code whose parent subsystem code hosts a non-zero number of logical qubits, as well as one which may host several dynamical logical qubits. Perhaps most importantly, it demonstrates the construction of a Floquet code using essentially only the subsystem structure of the parent code through the addition of gauge defects. We strongly suspect that this approach may be much more widely applicable in constructing more examples of Floquet codes.

This paper is organized as follows. In Section II, we recall and describe the usual Bacon-Shor code[25] using the virtual qubit framework that sets up the notation and makes the discussion in the following sections simpler. In Section III, we show how to introduce logical dynamical degrees of freedom through gauge defects, in addition to the usual Bacon-Shor logical qubit. In Section IV, we discuss error correction for such Floquet-Bacon-Shor codes. We conclude in Section V, with some thoughts on open problems and future work.

## II. BACON-SHOR CODE

Here, we briefly review the Bacon-Shor code[25]. We start by first recalling some basic facts about subspace, or stabilizer, codes as well as subsystem codes, using the formalism of ‘virtual’ qubits [26, 27]. First, note that the Pauli group of  $n$  qubits  $\mathcal{P}_n = \langle i\mathbb{I}, X_1, Z_1, \dots, X_n, Z_n \rangle$  can be transformed under an automorphism  $X_i, Z_j \rightarrow \bar{X}_i, \bar{Z}_j$  as  $\langle i\mathbb{I}, \bar{X}_1, \bar{Z}_1, \dots, \bar{X}_n, \bar{Z}_n \rangle$  as long as the transformed qubit operators satisfy the canonical commutation relations  $[\bar{X}_i, \bar{Z}_j] = 2\delta_{ij}\bar{X}_i\bar{Z}_j$  and  $\{\bar{X}_i, \bar{Z}_j\} = 2(1 - \delta_{ij})\bar{X}_i\bar{Z}_j$ . These ‘virtual qubit’ operators need not be operators on individual physical qubits, but rather describe collective degrees of freedom.

### A. Stabilizer and subsystem codes

In stabilizer, or subspace, codes, one promotes the first  $s$  virtual  $\bar{Z}$  operators as stabilizers, and identifies the group they generate  $\mathcal{S} = \langle \bar{Z}_1, \dots, \bar{Z}_s \rangle$  as the stabilizer group. Since these generators all mutually commute, they admit a simultaneous eigenbasis. Conventionally, an eigenbasis in which all the  $\{\bar{Z}_j\}_{j=1}^s$  take a value  $+1$  is chosen to be the code space. Notably,  $-\mathbb{I} \notin \mathcal{S}$ . Now, the centralizer of some arbitrary group  $G$  in the Pauli group is defined as  $\mathcal{Z}(G) = \{p \mid p \in \mathcal{P}_n, pgp^\dagger = g \ \forall g \in G\}$ , the set of all Pauli operators that commute with every element of the group  $G$ . The centralizer of this stabilizer group is given by  $\mathcal{Z}(\mathcal{S}) = \langle i\mathbb{I}, \bar{Z}_1, \dots, \bar{Z}_s, \bar{X}_{s+1}, \bar{Z}_{s+1}, \dots, \bar{X}_n, \bar{Z}_n \rangle$ .

The centralizer contains the logical operators of this code, as well as undetectable logical errors, in the subspace  $\mathcal{L}(\mathcal{S}) = \mathcal{Z}(\mathcal{S}) \setminus \mathcal{S}$ , where the notation  $A \setminus B$  denotes the set of all the elements in  $A$  that are not in  $B$ . The set of operators  $\{\bar{X}_{s+1}, \bar{Z}_{s+1}, \dots, \bar{X}_n, \bar{Z}_n\}$  provides the logical operators for  $k = n - s$  logical qubits. The smallest weight of any operator in  $\mathcal{Z}(\mathcal{S}) \setminus \mathcal{S}$  gives the code distance  $d$ , where by weight we mean the number of physical qubits an operator non-trivially acts on. Detectable errors live in the space  $\mathcal{P}_n \setminus \mathcal{Z}(\mathcal{S})$ , and all such operators with weight  $\lfloor (d-1)/2 \rfloor$  are correctable. A code described in this manner is succinctly referred to as an  $[[n, k, d]]$  stabilizer code.

Note that due to the abelian nature of stabilizer groups, and the fact that any pair of Pauli operators either commutes or anti-commutes, the normalizer of any stabilizer group  $\mathcal{S}$ , defined as  $\mathcal{N}(\mathcal{S}) = \{p \mid p \in \mathcal{P}_n, p\mathcal{S}p^\dagger \in \mathcal{S}\}$ , the set of all Pauli operators that leave the stabilizer group fixed upon conjugation, exactly coincides with the centralizer. However, as noted below, the normalizer and centralizer of non-abelian gauge groups are generally different, so we use the latter throughout this paper for consistency.

In subsystem codes, we identify a non-abelian subgroup of the Pauli group  $\mathcal{G} \subset \mathcal{P}_n$  as the gauge group. Due to its non-abelian nature, we now have  $-\mathbb{I} \in \mathcal{G}$ , unlike the case for stabilizer groups. This also means that the normalizer of the gauge group is the entire Pauli group  $\mathcal{N}(\mathcal{G}) = \mathcal{P}_n$ , so that we now define relevant quantities in terms of the centralizer  $\mathcal{Z}(\mathcal{G})$  instead. The generators of this gauge group are typically chosen to be low-weight operators that can be measured with relative ease. We first rewrite this non-abelian gauge group in terms of virtual qubit generators  $\mathcal{G} = \langle i\mathbb{I}, \bar{Z}_1, \dots, \bar{Z}_s, \bar{X}_{s+1}, \bar{Z}_{s+1}, \dots, \bar{X}_{s+g}, \bar{Z}_{s+g} \rangle$ .

The center of this gauge group, defined as the set of all elements of the gauge group that commute with every other element of the gauge group, is identified as the stabilizer subgroup  $\mathcal{C}(\mathcal{G}) = \mathcal{Z}(\mathcal{G}) \cap \mathcal{G} = \mathcal{S}$ , and is given in terms of the virtual qubit operators as  $\mathcal{S} = \langle \bar{Z}_1, \dots, \bar{Z}_s \rangle$ . Meanwhile, the remaining non-trivial generators of the gauge group  $\{\bar{X}_{s+1}, \bar{Z}_{s+1}, \dots, \bar{X}_{s+g}, \bar{Z}_{s+g}\}$  are the logical operators for gauge degrees of freedom, that we refer to as simply gauge qubits. The logical operators associated with these gauge qubits represent transformations that do not affect the codespace. Altogether, this gives us  $s = |\mathcal{S}|$  many stabilizer qubits, and  $g = (|G| - |\mathcal{S}|)/2$  many gauge qubits, where  $|A|$  denotes the cardinality, or the number of independent generators, of some group  $A$ .

In the case of the nonabelian gauge groups for subsystem codes, the set of all operators  $\mathcal{Z}(\mathcal{G}) \setminus \mathcal{S}$  contains the bare logical operators for this subsystem code, which leave both the stabilizer as well as the gauge qubits intact, and only affect the logical qubits of the subsystem code. However, owing to the gauge freedom in a subsystem code, we are also free to multiply these bare logical operators by any gauge operator, so that they can non-trivially operate on both the gauge and logi-

cal qubits, without impacting the state of the stabilizer qubits. The set of all such dressed logical operators operators is given by the subset of the centralizer of the stabilizer subgroup that excludes all elements in the gauge group,  $\mathcal{L}(\mathcal{G}) = \mathcal{Z}(\mathcal{S}) \setminus \mathcal{G}$ . The weight of the smallest operator in the space  $\mathcal{L}(\mathcal{G})$  gives the code distance  $d$ . As with stabilizer codes, detectable errors live in the space  $\mathcal{P}_n \setminus \mathcal{Z}(\mathcal{S})$ , and all operators in this space with weight  $\lfloor (d-1)/2 \rfloor$  are correctable. A code with gauge degrees of freedom described in this manner is referred to as an  $[[n, k, g, d]]$  subsystem code, where  $k = n - s - g$  is the number of logical qubits.

## B. Virtual qubit operators for the Bacon-Shor code

The Bacon-Shor code is a prototypical subsystem code defined on an  $L \times M$  square lattice. The gauge group associated with this subsystem code is generated by all nearest-neighbor horizontal XX checks and vertical ZZ checks, i.e.  $\mathcal{G} = \langle X_{i,j}X_{i,j+1}, Z_{i,j}Z_{i+1,j} \mid i \in [L], j \in [M] \rangle$ , where we use the notation  $[K] = \{0, \dots, K-1\}$ . The stabilizer subgroup of this gauge group is generated by operators that are either the product of all horizontal XX checks along two neighboring columns, or the product of all vertical ZZ checks along two neighboring rows.

The rank of this gauge group is simply the number of edges on the square lattice,  $|G| = L(M-1) + M(L-1)$ . Meanwhile, the rank of the stabilizer subgroup described above is given by  $|S| = (L-1) + (M-1)$ . This leaves us with  $g = (|G| - |S|)/2 = (M-1)(L-1)$  many gauge qubits, which is exactly the number of square plaquettes in the  $L \times M$  lattice. Thus, we can associate each gauge qubit with a square plaquette of the lattice. Since the number of physical qubits is given by  $n = ML$ , we are left with  $k = n - g - s = 1$  logical qubit in this subsystem code.

The pair of bare logical operators for this subsystem code can be taken to be the product of  $X$  operators along

the left-most column, and similarly the product of  $Z$  operators along the bottom-most row, which we choose to identify as the logical  $\bar{X}$  and  $\bar{Z}$  operators respectively. Upto stabilizer transformations, the logical  $\bar{X}$  ( $\bar{Z}$ ) operator is equivalent to the product of  $X$  ( $Z$ ) operators along any column (row). Of course, these bare logical operators can also be multiplied by any gauge operator to yield (gauge) equivalent dressed logical operators.

We also identify the operators for the gauge qubits in this subsystem code, which we noted above can be associated with the square plaquettes of the lattice. We can take the  $\bar{X}$  operator for a given gauge qubit to be the product of  $XX$  checks on all the edges above the square plaquette, including the top edge of the square plaquette, associated with this gauge qubit. Similarly, the  $\bar{Z}$  operator for this gauge qubit may be taken to be the product of  $ZZ$  checks on all the edges to the right of its square plaquette, including the right edge of the square plaquette.

The identification of gauge qubits with square plaquettes also motivates the identification of the stabilizer qubits and the logical qubit with some part of the physical lattice. With the definitions of the stabilizer and logical qubit operators given above, it is intuitive to identify the horizontal  $Z$ -type stabilizer qubits with imaginary square plaquettes sharing right-side edges with the left-most column, and the vertical  $X$ -type stabilizer qubits with similar imaginary square plaquettes sharing top-side edges with the bottom-most row.

Similarly, the logical qubit can be associated with an imaginary square plaquette with its top-right corner at the origin of the lattice residing in its bottom-left most corner. With such an identification of all virtual qubits with square plaquettes on the lattice, we denote e.g.  $\bar{X}_{i,j}$  to be the logical  $X$  operator of the virtual qubit associated with the square plaquette with its top-right corner at the  $(i, j)$  coordinate of the  $L \times M$  square lattice, where  $i \in [M]$  and  $j \in [L]$ . This situation is depicted in Fig. 1. The explicit expressions for the logical operators of the various virtual qubits, in terms of the operators of the physical qubits, which reside on the nodes of the square lattice, is given in Eq. (1).

---

Horizontal $Z$ -type stabilizers:	$\bar{Z}_{0,i} = \prod_{j=0}^{M-1} Z_{j,i} Z_{j,i-1}$	$\bar{X}_{0,i} = \prod_{j=1}^{L-1} X_{0,j}$	$(1 \leq i \leq L-1)$
Vertical $X$ -type stabilizers:	$\bar{Z}_{i,0} = \prod_{j=0}^{L-1} X_{i,j} X_{i-1,j}$	$\bar{X}_{i,0} = \prod_{j=i}^{M-1} Z_{j,0}$	$(1 \leq i \leq M-1)$
Gauge qubits:	$\bar{Z}_{i,j} = \prod_{k=i}^{M-1} Z_{k,j} Z_{k,j-1}$	$\bar{X}_{i,j} = \prod_{k=j}^{L-1} X_{i,k} X_{i-1,k}$	$(1 \leq i, j \leq M-1, L-1)$
Logical qubit:	$\bar{Z}_{0,0} = \prod_{i=0}^{M-1} Z_{i,0}$	$\bar{X}_{0,0} = \prod_{i=0}^{L-1} X_{0,i}$	

(1)

One way to fix a gauge of the Bacon-Shor code is to fix the values of some of the gauge qubit logical operators. Threshold behavior in such gauge fixings of the Bacon-Shor code have been previously explored in the context of 2D compass codes[28].

### III. FLOQUET-BACON-SHOR CODE

In order to measure the stabilizers of the Bacon-Shor subsystem code, it suffices to use a period 2 measurement schedule, such that we measure all the horizontal  $XX$  checks in the first round, followed by all the vertical  $ZZ$  checks. From a Floquet perspective, this corresponds to periodically moving between two different gauge fixings of the Bacon-Shor code, one where the  $\bar{X}$  operators of all the gauge qubits has been fixed, followed by fixing all the  $\bar{Z}$  gauge qubit operators. We can view either gauge fixing as a stabilizer code that contains the Bacon-Shor stabilizers as a subset, but also includes the  $\bar{X}$  or  $\bar{Z}$  gauge qubit operators. It does not matter whether these gauge qubit operators are fixed to  $\pm 1$ , and in general they will take random values upon each measurement, with the only constraint being the values of the stabilizers.

In order to free up space for an additional logical qubit, we must unfix one of the gauge degrees of freedom. In general, such an unfixing, or gauge defect, is difficult to maintain while ensuring all the Bacon-Shor stabilizers also get measured. However, as we show here, it is possible to maintain such a gauge defect in a dynamical sense, such that the encoded space for the corresponding logical qubit changes across measurement rounds, similar to the examples of previously constructed Floquet codes.

#### A. Measurement schedule and ISGs

In particular, we can adopt the period 4 measurement schedule shown in the top part of Fig. 2. The core idea here is that we distribute the measurement of the Bacon-Shor stabilizers into four, instead of two, rounds, allowing us to prevent one of the gauge degrees of freedom from being fixed, thereby introducing a gauge defect that evolves across the measurement rounds and serves as the dynamical logical degree of freedom.

In order to describe the measurement process, it is helpful to label four of the square plaquettes that serve a special role in identifying the dynamical logical qubit. For simplicity, we will assume a  $d \times d$  square lattice. As in Section II, we denote the location of square plaquettes by the location of their top-right corner. Since square plaquettes are in one-to-one correspondence with gauge qubits, we will use the two terms interchangeably. For odd  $d$ , the gauge qubit A is identified as the plaquette at location  $(\frac{d-1}{2}, \frac{d+1}{2})$ , B at  $(\frac{d+1}{2}, \frac{d+1}{2})$ , C at  $(\frac{d+1}{2}, \frac{d-1}{2})$ , and D at  $(\frac{d-1}{2}, \frac{d-1}{2})$ . Similarly, for even  $d$ , A is identified as the plaquette at  $(\frac{d}{2} - 1, \frac{d}{2} + 1)$ , B at  $(\frac{d}{2}, \frac{d}{2} + 1)$ , C at

$(\frac{d}{2}, \frac{d}{2})$ , and D at  $(\frac{d}{2} - 1, \frac{d}{2})$ . In either case, these four plaquettes all share a single corner that lies roughly at the center of the square lattice. We also denote columns AD and BC, and rows AB and CD as the unique columns and rows containing the obvious choice of plaquettes, as well as edges AB, BC, CD and AD as the edges between the respective plaquettes.

A complete period of the measurement cycle consists of 4 measurement rounds, which are subsequently repeated. In each measurement round, we measure either all  $XX$  or  $ZZ$  checks except along a particular column or row of plaquettes, that we refer to as the defect column or row. The defect column is column AD at round 0, and column BC at round 2, while the defect row is row AB at round 1, and row CD at round 3. In the defect column or row, we measure a single  $XX$  or  $ZZ$  check on the edge that shares the same label as the column. For instance, at round 0, we measure all the  $XX$  checks except those in column AD, in which we only measure the  $XX$  check on edge AD. With this measurement schedule, the Bacon-Shor stabilizers associated with the defect columns and rows get measured only once in every period, while all other stabilizers get measured twice. The gauge defect is identified with the pair of gauge qubits that straddle the unique edge in the defect column or row that we measure a check operator on. It evolves from the pair AD at round 0, to AB at round 1, to BC at round 2, to CD at round 3, and then again to AD, repeating the cycle. In essence, it is this gauge defect that serves as the dynamical logical qubit. Note that an  $XX$  ( $ZZ$ ) check on any edge equals the product of  $\bar{X}$  ( $\bar{Z}$ ) operators for the two square plaquettes sharing that edge, fixing which removes precisely one gauge degree of freedom.

After one complete period, the instantaneous stabilizer groups (ISGs) induced by the measurement schedule settles into a steady state, and is shown in the bottom part of Fig. 2. Each of the four ISGs contains all the check operators measured in the current round, or equivalently all the virtual gauge qubit  $\bar{X}$  ( $\bar{Z}$ ) operators in the plaquette columns (rows) along which we measure all the horizontal (vertical)  $XX$  ( $ZZ$ ) checks. Along the defect plaquette column (row), the ISG consists of  $\bar{Z}$  ( $\bar{X}$ ) operators for all the gauge qubits except those that straddle the edge whose  $XX$  ( $ZZ$ ) check we measure in the current round. For this pair of gauge qubits, the gauge defect, we fix only the product of their  $\bar{X}$  ( $\bar{Z}$ ) operators. In addition, all the ISGs also contain the Bacon-Shor stabilizers. The fixed gauge qubit operators randomly take either value  $\pm 1$  each time they are measured afresh. Only a subset of these operators survive across rounds, and can therefore serve as error syndromes. In particular, the  $\bar{X}$  operators along rows CD and AB serve as syndromes at rounds 0 and 2 respectively, while the  $\bar{Z}$  operators along columns AD and BC serve as syndromes at rounds 1 and 3 respectively. In addition, the Bacon-Shor stabilizers also serve as error syndromes just as they do in the parent subsystem code.

Let  $\mathcal{S}^{(r)}$  denote the ISG at round  $r$ . As an example of

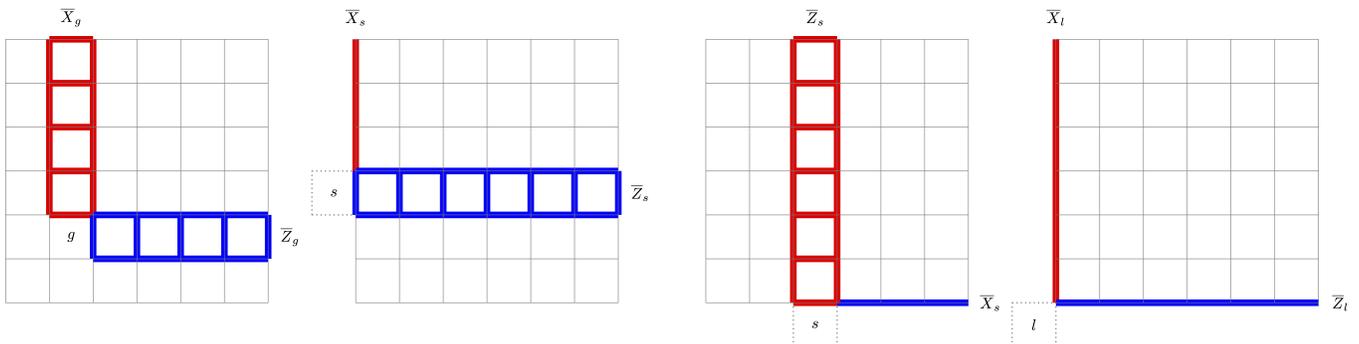


FIG. 1. Virtual qubit operators for the Bacon-Shor subsystem code. From left-right, we have respectively an example of a (i) gauge qubit, (ii) horizontal stabilizer qubit, (iii) vertical stabilizer qubit, and (iv) logical qubit. Note that while the gauge qubits are in one-to-one correspondence with the square plaquettes of the lattice, one can associate imaginary square plaquettes running along the left and bottom boundaries of the lattice for the latter three types of virtual qubits.

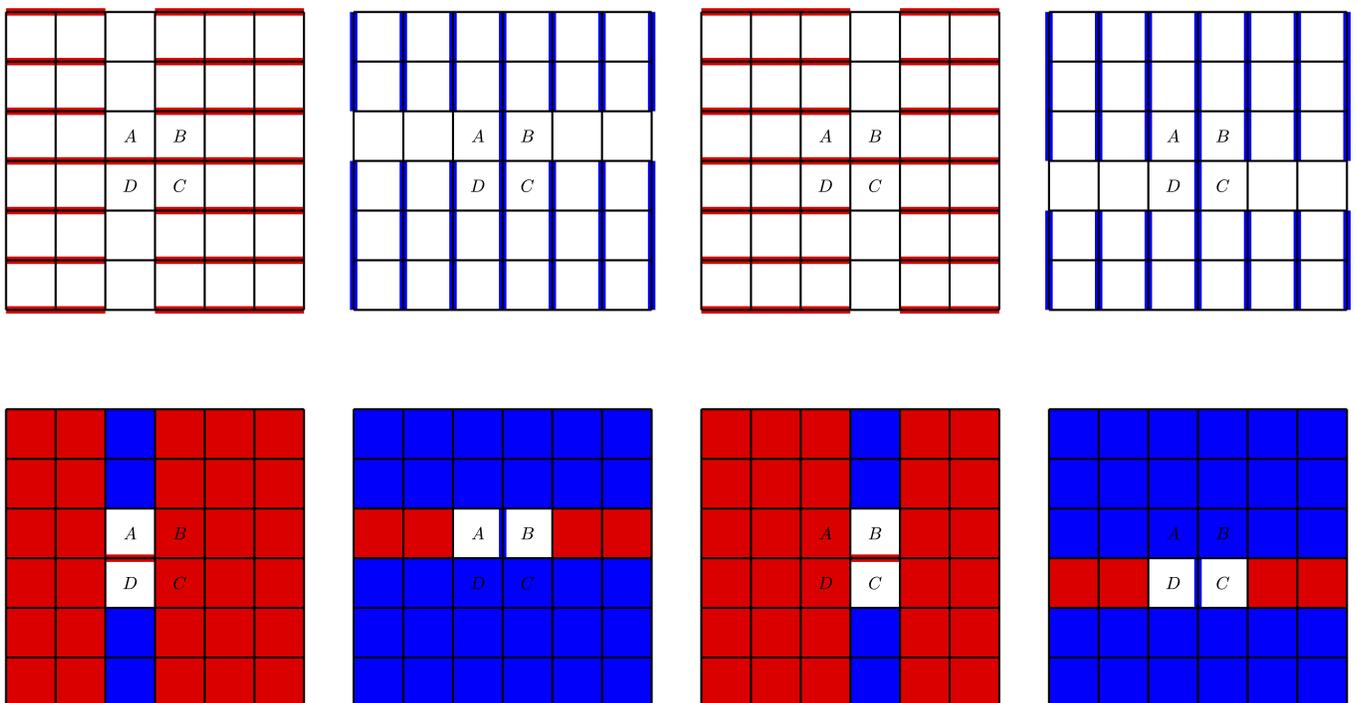


FIG. 2. Measurement schedule (top) with period 4 that maintains a gauge defect across the entire measurement cycle, thereby realizing a dynamical logical qubit in addition to the usual Bacon-Shor logical qubit. Horizontal  $XX$  checks are colored red, while vertical  $ZZ$  checks are colored blue. After one complete period of this measurement schedule, the induced instantaneous stabilizer groups (ISGs) (bottom) achieve steady state. In addition to the usual Bacon-Shor stabilizers, each of the ISGs contains the measured check operators, as well as any elements from the previous ISG that commute with the currently measured checks. Any cell, or equivalently a gauge qubit, colored either red or blue corresponds to gauge fixing its  $\bar{X}$  or  $\bar{Z}$  operator respectively.

the description above,  $\mathcal{S}^{(4k)}$  ( $k \geq 1$ ) contains the  $\bar{X}$  operators of all the gauge qubits except those that belong in the missed plaquette column containing gauge qubits A and D. This ISG contains the  $\bar{Z}$  operators of all the gauge qubits in this column except the gauge qubits A and D. These  $\bar{Z}$  operators were measured and have persisted from the previous round, and commute with all other elements of the current ISG. In addition, the product  $\bar{X}_A \bar{X}_D$ , which is simply the  $XX$  check measured on

the edge shared between qubits A and D, also exists as an element of the ISG  $\mathcal{S}^{(4k)}$ .

In any ISG, only gauge fixed  $\bar{Z}$  operators appear anywhere in the column(s), and similarly only gauge fixed  $\bar{X}$  operators appear anywhere in the row(s), in which the gauge defect belongs. This ensures that the code distance of any of the ISGs is maintained at  $\sim d$  on a  $d \times d$  square lattice. Of course, this also relies on the vertex shared by the plaquettes A, B, C and D lying roughly

$d/2$  away from any boundary of the lattice, and the code distance would shrink if the gauge defect was instead brought closer to any lattice boundary.

The logical operators for the Bacon-Shor qubit do not change across these ISGs, and are still given by the operators  $\overline{X}_{0,0}$  and  $\overline{Z}_{0,0}$  from Eq. (1). In the honeycomb Floquet code [4], there are similar static logical operators arising from non-trivial homological loops. These were referred to as inner logical operators since they belong to the center of the parent gauge group, though the measurement schedule prevents them from ever being measured. Here, the logical operators for the Bacon-Shor qubit are not inner logical operators in the sense that they do not belong to the center of the gauge group. So instead, we simply refer to them as static logical operators, even though of course, the equivalence class of such operators upto instantaneous stabilizer transformations does evolve from round to round.

The logical operators for the dynamical qubit introduced as a result of the measurement schedule introduced above do change across rounds. Roughly speaking, they correspond to the virtual qubit operators associated with the gauge defect that arises from fixing the product of operators of neighboring gauge qubits, and that evolves from round to round. Whenever we fix the product of neighboring gauge qubits of a particular type,  $\overline{X}$  or  $\overline{Z}$ , the logical operator of the same type for the dynamical qubit is given by the virtual qubit operator for either gauge qubit, while that of the opposite type is given by the product of the virtual qubit operators of the two gauge qubits. In particular, using the definitions of the gauge qubits A, B, C and D described above, and depicted in Fig. 2, the pair  $(\overline{X}, \overline{Z})$  of logical operators for the dynamical qubit evolves as  $(\overline{X}_A, \overline{Z}_A \overline{Z}_D)$  at round 0,  $(\overline{X}_A \overline{X}_B, \overline{Z}_B)$  at round 1,  $(\overline{X}_B, \overline{Z}_B \overline{Z}_C)$  at round 2, and  $(\overline{X}_C \overline{X}_D, \overline{Z}_C)$  at round 3.

In order to preserve logical information across these four ISGs, it is also necessary that some representative, upto instantaneous stabilizer operations, of a logical operator at round  $r$  persists as a logical operator in the next round  $r + 1$ . This is formalized in Eq. (2). Roughly speaking, this condition ensures that if  $|\psi^{(r)}\rangle = \alpha |0^{(r)}\rangle + \beta |1^{(r)}\rangle$  is the state of the dynamical logical qubit at round  $r$ , then the measurement schedule projects it to  $|\psi^{(r)}\rangle \rightarrow \alpha |0^{(r+1)}\rangle + \beta |1^{(r+1)}\rangle = |\psi^{(r+1)}\rangle$  at round  $r + 1$ , where the codewords  $|0^{(r)}\rangle$  etc change across the ISGs, but the pair of coefficients  $\alpha$  and  $\beta$ , and therefore the logical information, does not. Specifically, this condition says that if  $S^A$  and  $S^B$  are two stabilizer groups associated with stabilizer codes  $A$  and  $B$  respectively, with associated logical operators  $O^A$  and  $O^B$ , where  $O^j \subset \mathcal{N}(S^j) \setminus S^j$ , then if there exist  $s^A \in S^A$  and  $s^B \in S^B$  such that

$$s^A O^A = s^B O^B \quad (2)$$

then logical information is preserved. This is also discussed more in the appendix.

Following Eq. (2), we must identify pairs of stabilizer

elements  $s^{(r)}, s^{(r+1)} \in \mathcal{S}^{(r)}, \mathcal{S}^{(r+1)}$  that connect a logical operator for the ISG  $\mathcal{S}^{(r)}$  to one for the ISG in the next round  $\mathcal{S}^{(r+1)}$  in order to preserve the logical information contained in the dynamical qubit across these rounds. These pairs are given in Table I.

Round	0	1	2	3
$\overline{X}^{(r)}$	$\overline{X}_A$	$\overline{X}_A \overline{X}_B$	$\overline{X}_B$	$\overline{X}_C \overline{X}_D$
$\overline{Z}^{(r)}$	$\overline{Z}_A \overline{Z}_D$	$\overline{Z}_B$	$\overline{Z}_B \overline{Z}_C$	$\overline{Z}_C$
$s_x^{(r)}$	$\overline{X}_A \overline{X}_D \cdot \overline{X}_C$	$\mathbb{I}$	$\overline{X}_A$	$\mathbb{I}$
$s_x^{(r-1)}$	$\mathbb{I}$	$\overline{X}_B$	$\mathbb{I}$	$\overline{X}_B \overline{X}_C \cdot \overline{X}_D$
$s_z^{(r)}$	$\mathbb{I}$	$\overline{Z}_D \cdot \overline{Z}_A \overline{Z}_B$	$\mathbb{I}$	$\overline{Z}_B$
$s_z^{(r-1)}$	$\overline{Z}_C \overline{Z}_D \cdot \overline{Z}_A$	$\mathbb{I}$	$\overline{Z}_C$	$\mathbb{I}$

TABLE I. Evolution of the dynamical logical operators  $\overline{X}^{(r)}$  and  $\overline{Z}^{(r)}$  across the four measurement rounds, where the locations of the gauge qubits/plaquettes A, B, C and D are described in the main text. The stabilizer element pairs  $s_j^{(r)}$  and  $s_j^{(r-1)}$  for type  $j$  connect the logical operators across subsequent rounds,  $s_x^{(r)} \overline{X}^{(r)} = s_x^{(r-1)} \overline{X}^{(r-1)}$ , and similarly for the  $\overline{Z}^{(r)}$  operators.

## B. Multiple Dynamical Qubits

Introducing additional dynamical logical degrees of freedom amounts to adding more gauge defects to the measurement schedule, and is straightforward to generalize from the construction above for a single dynamical logical qubit. Consider relabeling the set of plaquettes  $(A, B, C, D)$  described above as  $(A_0, B_0, C_0, D_0)$ , and identifying another such similar set of plaquettes  $(A_1, B_1, C_1, D_1)$  where we repeat the measurement schedule identified above. Thus, for instance, we would now have two defect columns,  $A_0 D_0$  and  $A_1 D_1$ , at round 0. The code distance would now be proportional to the shorter of the two distances from any edge of the square lattice to either of the two resultant gauge defects. To make one of these distances the same for both gauge defects, we can put them in the same plaquette column or row. Suppose, without loss of generality, that we put them in the same row. The optimal choice would be to place both gauge defects roughly  $d/3$  distance away from each other, as well as either vertical edge of a  $d \times d$  square lattice, but roughly  $d/2$  away from the horizontal edges. Since this already reduces the code distance to  $\sim d/3$ , one might as well add two more gauge defects such that all of these lie roughly  $d/3$  units away from any edge of a  $d \times d$  square lattice.

Generalizing this construction, we see that while one could in principle keep adding more dynamical logical qubits by simply adding more gauge defects, the overall

code distance decreases by the same amount as one goes from  $m^2$  to any number between  $m^2 + 1$  and  $(m + 1)^2$  dynamical logical qubits, where  $m$  is an integer. The optimal placement for  $m^2$  many gauge defects is such that we place  $m$  of these gauge defects in the same row and column, such that the boundaries of the resultant defect lattice are roughly  $d/(m + 1)$  away from the edges of the square lattice. Therefore, adding  $k$  dynamical logical qubits to the Bacon-Shor code reduces the code distance to roughly  $d/(\sqrt{k} + 1)$ , or  $\Theta(d/\sqrt{k})$ . This situation is depicted in Fig. 3 for the illustrative case of 9 dynamical logical qubits.

In the presence of biased noise, one may instead want to adjust the  $X$  and  $Z$  distances,  $d_X$  and  $d_Z$ , separately. For instance, in the hypothetical case of infinite  $Z$  biased noise, one could keep adding more dynamical gauge defects along a particular column, which progressively shortens  $d_X$  but maintains the same value of  $d_Z$ , as all these dynamical qubits lie the same distance away from the vertical boundaries of the square lattice. Alternatively, as with the Bacon-Shor parent subsystem code, we may elongate the horizontal dimension of the square lattice itself while keeping the vertical dimension fixed.

#### IV. ERROR DETECTION AND CORRECTION

Here, we assume an error model of independent  $X$  and  $Z$  errors on all the qubits between measurement rounds, and independent measurement errors of the  $XX$  and  $ZZ$  check operators. A similar error model was assumed in [4], but there the model also assumed perfect measurements of the check operators, with measurement errors of e.g.  $XX$  checks being equivalent to a single  $Z$  error on one of the qubits supported by this check before and after the measurement of this check operator. In our case, such an error pattern would flip two checks supported by that qubit, and would make it inequivalent to a measurement error on a single check. Since  $X$  operators can be commuted past the measurements of rounds 0 and 2, we assume that  $X$  errors only occur between either rounds 0 and 1, or between rounds 2 and 3. Similarly,  $Z$  errors can be commuted past the measurements of round 1 and 3, and we assume that these occur either between rounds 1 and 2, or between rounds 3 and 0 (mod 4).

The decoding graph is typically defined as the graph whose vertices are given by the stabilizer measurements, and edges are given by independent errors, or faults. In the Floquet-Bacon-Shor construction, described in previous sections of the paper, we can have independent errors that flip up to 4 stabilizers. Thus, we are naturally led to the slight generalization of a decoding hyper-graph. A hyper-graph is a generalization of a graph where instead of an edge connecting two vertices, we have a hyper-edge that connects  $n$  vertices. In the decoding hyper-graph, a hyper-edge connecting 4 vertices would correspond to precisely such an independent error. This makes decoding more complicated in principle, as standard decoding

algorithms involving Minimum Weight Perfect Matching (MWPM) or Union Finding (UF) typically work with edges, not hyper-edges. However, some recent work [29] has shown that one could nevertheless adapt such techniques to work with decoding hyper-graphs as well.

In the usual Bacon-Shor subsystem code, an odd number of  $X$  errors in any row are gauge equivalent to a single  $X$  error anywhere in that row. Similarly, an odd number of  $Z$  errors in any column are, upto multiplication by gauge group elements, equivalent to a single  $Z$  error anywhere in that column. A chain of such errors will anti-commute with, and therefore get detected by, the stabilizers that reside at the boundaries of such chains. For instance, a single  $X$  error on any qubit on the lattice will anti-commute with the horizontal  $Z$ -type stabilizers that straddle the row in which this  $X$  error occurred. Error correction then proceeds by applying a single correcting  $X$  or  $Z$  operator anywhere in the relevant row or column. Conventionally we can choose the left-most column to apply correcting  $X$  operators along, and the bottom-most row to apply correcting  $Z$  operators on. All such errors of weights  $< d/2$  can be corrected in this manner on a  $d \times d$  square lattice.

In the Floquet-Bacon-Shor code described above, we have additional stabilizers that protect the encoded dynamical logical qubits. These are given by the gauge fixed operators that lie along the defect columns and rows of the various ISGs. We will refer to these as transient stabilizers, to distinguish them from the usual Bacon-Shor stabilizers that run along the length of the entire lattice, which we shall refer to as permanent stabilizers. The collection of such gauge operators, or transient stabilizers, in any defect column or row form a domain wall across which single qubit errors cannot propagate, since we skip the measurements along that column or row. For instance, assuming temporarily a single dynamical logical qubit for simplicity, at round  $4k$  ( $k \geq 1$ ), we do not measure any of the  $XX$  checks in the defect column AD except on the edge AD, so that single qubit  $X$  errors are equivalent, up to instantaneous stabilizer operations, to any single  $X$  error on either side of the domain wall in the column AD, depending on which side of this wall the error occurred, except along the row that contains the edge AD. This means that unlike the Bacon-Shor code, an even number of  $X$  errors, for instance, can bring us out of the dynamical codespace if we have an odd number of  $X$  errors on both sides of the domain wall.

Measurement errors of the permanent stabilizers are given in the decoding hyper-graph as edges connecting the same stabilizer across consecutive time slices. A defect is referred to as a flip in the recorded value of a stabilizer. A measurement error of a permanent stabilizer will cause two defects in the decoding hyper-graph, once due to the erroneous flip in recorded value, and the second when its value flips back to the correct one, and can therefore be decoded by standard matching algorithms. In the case of measurement errors of transient stabilizers, one has to be a bit more careful. Suppose we have just

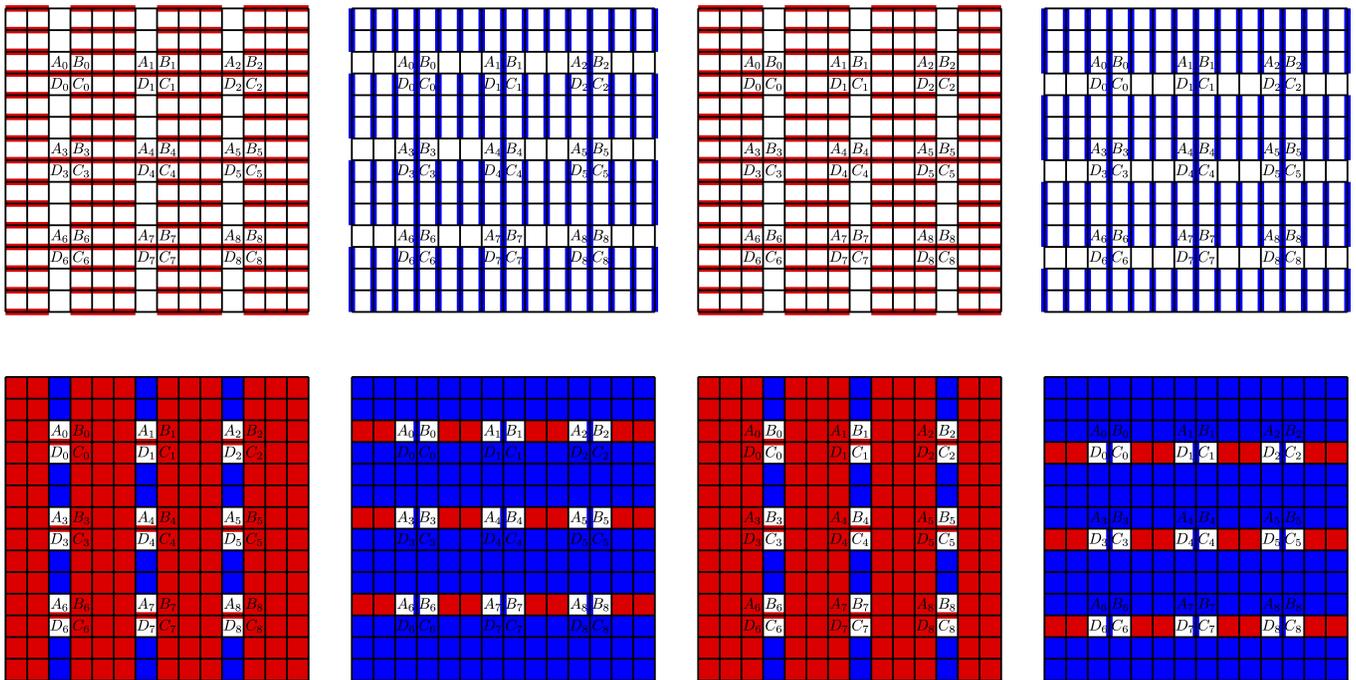


FIG. 3. Measurement schedule (top) for hosting 9 dynamical logical qubits in addition to the usual Bacon-Shor subsystem logical qubit. The sequence of instantaneous stabilizer groups (ISGs) this measurement schedule induces (below) maintains the introduced gauge defects throughout the measurement cycle.

carried out the measurements of round 1 mod 4, and one of the  $Z$ -type check operators along some plaquette row suffered a measurement error. If it occurred to the left of column AD, then it flips only the value of the permanent  $Z$ -type stabilizer along this row. If it occurs on the check between cells A and B, then it flips the value of permanent  $Z$ -type stabilizer along this row, as well as the transient stabilizer given by the gauge fixed  $\bar{Z}$  operator of the gauge qubit lying at the intersection of this row and column AD. If it instead occurs on any check further right along this row, then it flips the previous two stabilizers, as well as the gauge fixed  $\bar{Z}$  operator of the gauge qubit at the intersection of this row and column BC. Thus, measurement errors of transient stabilizer in general will lead to hyper-edges connecting several vertices in the decoding hyper-graph.

### A. Decoding and Code Distance

For simplicity, we will assume the case of a single dynamical logical qubit, as the extension to multiple logical qubits is straightforward, and briefly discussed towards the end of this section. In addition, note that each of the ISGs is equivalent to a clockwise  $\pi/2$  rotation of the previous ISG, together with interchanging  $X \leftrightarrow Z$ , so that it suffices to discuss errors occurring between any two consecutive rounds.

Next, suppose concretely that we are in the ISG  $S^{(4k)}$  ( $k \geq 1$ ), and suppose that a single qubit  $X$  error occurs

on some row in the bulk, far from the top and bottom boundaries of the square lattice, and far from the boundaries of the cells A, B, C and D. This lone  $X$  error will flip the two permanent  $Z$ -type stabilizers that straddle this row, and that are measured in the next row. If this  $X$  error occurred on the right of the blue domain wall along plaquette column AD, then it will also additionally flip the two transient stabilizers that exist along this defect column, and that also straddle this row and are measured in the next round. Thus, such errors give rise to hyper-edges connecting 4 vertices in the decoding hyper-graph.

At the top and bottom boundary rows, we may add additional imaginary stabilizers that get flipped in addition to the lone stabilizers, permanent and transient, in order to apply matching algorithms. This leaves us with three rows left to consider: the top row of cells AB, the row between cells AB and CD, and the bottom row of cells CD. Note that an  $X$  error on the row between cells AB and CD is ISG equivalent to a single  $X$  error anywhere on that row, so that we may treat it as occurring e.g. on the left-most column.

Between rounds  $4k$  and  $4k + 1$ , an  $X$  error on the bottom row of cells CD can be identified in a somewhat similar fashion, as it flips the pairs of permanent  $Z$ -type stabilizers that straddle this row and that all get measured in the round  $4k + 1$ . In addition, it also flips the transient  $Z$ -type stabilizer associated with the cell below cell D, which also gets measured in the next round  $4k + 1$ . Although the  $\bar{Z}$  operator of cell D gets measured in round  $4k + 1$ , it cannot be used as a syndrome bit as it has not

persisted through the previous round, nor does it persist in the subsequent round. In general, none of the  $Z$  operators of cells/gauge qubits A, B, C or D may be used as syndrome bits, even if they exist in some of the ISGs. We therefore add imaginary boundary transient  $Z$ -type stabilizers at cells A, B, C and D to allow for matching with neighboring transient  $Z$ -type stabilizers.

To handle the remaining two cases, we may adopt two different decoding strategies, one that detects and fixes errors as they accumulate, i.e. in the same time slice, and the second where non-trivial syndrome measurements, even with perfect measurements, are edge-connected across time slices separated by two measurement rounds. Using the first strategy, a single  $X$  error on either one of the above two rows, i.e. the row above cells AB, or the row between cells AB and CD, will flip a single permanent  $Z$ -type stabilizer that borders the plaquette row AB in the next round  $4k + 1$ . In addition, an  $X$  error on the row above cells AB will additionally flip a single transient  $Z$ -type stabilizer above cell A. In the first decoding strategy, we associate an imaginary permanent  $Z$ -type stabilizer to the plaquette row AB, as well as an imaginary transient  $Z$ -type stabilizer to the cell A, to enable pairing with the nearby stabilizers in a matching algorithm. This effectively reduces the distance by a factor of roughly  $1/2$ , as an error chain of weight  $\lesssim d/2$  consisting of  $X$  errors on all the rows above the top row of cells AB will produce the same syndrome information as a single  $X$  error on the row above cells AB.

Such a chain would however be distinguished from the single  $X$  error if one also includes the  $Z$ -type stabilizer along plaquette row AB, but this does not get measured until measurement round  $4k + 3$ . In the second decoding strategy, the decoding graph consists of edges that connect the two permanent  $Z$ -type stabilizers that border plaquette row AB, measured at round  $4k + 1$ , to the permanent  $Z$ -type stabilizer that runs along plaquette row AB. Together with the flipped transient  $Z$ -type stabilizer above cell B, as well as the flipped imaginary transient stabilizer at cell B, this enables distinguishing the lone  $X$  error from the error chain described above, and therefore restores the code distance to  $\sim d$ . In either case, the code distance remains  $\Theta(d)$ .

We note that there currently exists no general well-formed notion of the code distance of a Floquet code. In particular, it is not simply the minimum code distance of any of the ISGs considered as stabilizer codes in isolation, as the time dynamics may allow for small weight errors to evolve into logical errors, even if the ISGs have large code distances by themselves. For one thing, we can multiply instantaneous logical operators by not only instantaneous stabilizer elements, but also elements of the ISGs before and after the current round. This is so because an error of the form  $\mathcal{L}^{(r)} s^{(r-1)}$  between measurement rounds  $r - 1$  and  $r$ , where  $\mathcal{L}^{(r)}$  is a logical operator at round  $r$ , and  $s^{(r-1)} \in \mathcal{S}^{(r-1)}$ , has the same effect as an error of the form  $\mathcal{L}^{(r)}$  at round  $r$ . Similarly, an error of the form  $s^{(r+1)} \mathcal{L}^{(r)}$  between measurement rounds

$r$  and  $r + 1$  will also have the same effect. In the present case, such transformations do not affect the code distance scaling reported here.

Even more generally speaking, we may say that two Pauli operators  $\sigma_\alpha$  and  $\sigma_\beta$  are space-time connected in a Floquet code if their difference is a chain of instantaneous stabilizer elements, i.e.  $\sigma_\alpha = s_{r+t} \dots s_r \sigma_\beta$ . It may be the case that a low weight Pauli error  $\sigma_\beta$  traverses, in its spacetime path, an undetectable logical error, before evolving into a partially detectable error, whose correction does not fix the previous logical error, or even getting absorbed into the ISG. Such a logical error occurs in [6], for instance. It may therefore be desirable to keep the number of time slices between which stabilizers are measured and used as syndromes to be as minimal as possible when designing Floquet codes, which motivates the distance  $\sim d/2$  decoding strategy we described above. In the absence of a formal definition of the code distance of a Floquet code, we simply identify the code distance as the smallest weight of any error operator that does not produce any non-trivial syndromes, given the decoding hyper-graph, which in turn depends on the choice of the specific error model we consider.

We also note that like the parent Bacon-Shor subsystem code, the Floquet-Bacon-Shor family of codes introduced here do not possess a threshold. This is attributable to the fact that the weight of the stabilizers used in both examples of codes grows as the distance increases, so that erroneous stabilizer measurements get likelier with increasing distance for any fixed value of independent error probability.

## B. Self-correcting errors

Suppose that a 2-qubit  $XX$  error occurs on an edge somewhere above cell A between rounds 0 and 1 mod 4. This will flip the two transient, but not the two permanent,  $Z$ -type stabilizers that straddle this edge, lie along column AD, and are measured in round 1 mod 4. In the very next round 2 mod 4, we carry out a measurement of an  $XX$  check operator on precisely the same edge, and the error becomes part of the ISG. In other words, the appropriate correction operator automatically applies itself because of the measurement schedule itself. This occurs even if this 2-qubit error anti-commutes with a logical operator at round 0 mod 4, such as if it occurred on the top edge of cell A. We show this more explicitly in the appendix.

Furthermore, note that a single qubit  $X$  error occurring between rounds 0 and 1 mod 4 on any one row on say the right side of the blue domain wall AD is ISG equivalent to a single  $X$  error on the right boundary of the domain wall. It cannot be transported further left due to the existence of the domain wall, which in turn exists because we do not carry out any check measurements along the plaquette column of this wall. However, two measurement rounds later, we do carry out the measure-

ment of this check and the error can propagate further left. In turn, this implies that whether a single qubit  $X$  error occurs on the left or the right of the blue domain wall, the correction operator two measurement rounds later is to apply a single  $X$  operator somewhere on the left of the instantaneous blue domain wall, which has now shifted one column to the right and exists along plaquette column BC.

### C. Multiple Dynamical Qubits

The discussion above generalizes fairly straightforwardly to the case of adding multiple dynamical logical qubits. In this case, the main difference is that the hyper-edges in the decoding hyper-graph now connect more vertices. For instance, if we have 4 dynamical logical qubits arranged in a manner such that the pairs of cells  $(A_0, B_0)$  and  $(A_1, B_1)$  exist in the same plaquette row, and the pairs  $(D_0, C_0)$  and  $(D_1, C_1)$  exist in the plaquette row below it, then a single qubit  $X$  error on some row in the bulk towards the right of defect column  $A_1D_1$  between rounds 0 and 1 mod 4 will flip the values of the two permanent  $Z$ -type stabilizers, the two transient  $Z$ -type stabilizers along column  $A_1D_1$ , as well as the two transient  $Z$ -type stabilizers along column  $A_0D_0$ , that straddle this row. Thus, such an error gives rise to a hyper-edge connecting 6 vertices in the decoding hyper-graph.

The only other major difference is that an even number of  $X$  errors between gauge defects in the same plaquette row are now part of the ISG at round 0, and no longer count as errors. Only single qubit  $X$  errors between any consecutive pair of gauge defects, or a gauge defect and a lattice boundary count as errors, and similarly for  $Z$  errors.

### V. CONCLUSIONS

In this paper, we have described the introduction of dynamical logical qubits to the Bacon-Shor subsystem code by modifying the measurement schedule of the weight 2 check operators that generate the gauge group of the parent sub-system code. Such a measurement schedule

introduces and maintains gauge defects throughout the measurement cycle. This broad prescription of identifying the virtual qubit degrees of freedom in a subsystem code, and picking a measurement schedule that can maintain gauge defects across measurement cycles as a way to introduce dynamical logical qubits may be more broadly applicable in constructing more examples of Floquet codes. We showed that independent errors of a fairly simple noise model naturally lead us to consider a decoding hyper-graph, which is generally more complicated to work with than decoding graphs that arise in surface codes, for instance. This additional complication in the decoding task is also shared to some degree with LDPC codes, in which independent errors, or faults, also typically flip more than two stabilizer syndromes, and progress in constructing good decoders for LDPC codes may also benefit decoding Floquet codes, such as the one describe here. We leave a detailed consideration of fault tolerance, together with fault tolerant implementation of logical gates, in this code to future work. Lastly, we also note that skipping measurements of some check operators in some rounds was also used in [30] to introduce a threshold to the Bacon-Shor code. In our paper, we used such skipped measurements to introduce dynamical logical qubits to the Bacon-Shor code. We leave it as an open problem to investigate whether one could combine both sets of ideas to introduce a threshold to the Bacon-Shor code in the presence of dynamical logical qubits.

### VI. ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Superconducting Quantum Materials and Systems Center (SQMS) under contract No. DE-AC02-07CH11359. M.S.A. acknowledges support from USRA NASA Academic Mission Services under contract No. NNA16BD14C with NASA, with this work funded under the NASA-DOE interagency agreement SAA2-403602 governing NASA's work as part of the SQMS center. The authors wish to thank Jason Saied and Ryan LaRose for useful discussions on Floquet and error correcting codes in general, as well as Nikolas Breuckmann, Christophe Vuillot and Margarita Davydova for helpful comments during the final stages of this work.

- 
- [1] A. Kitaev, Fault-tolerant quantum computation by anyons, *Annals of Physics* **303**, 2 (2003).
  - [2] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary (1998), arXiv:quant-ph/9811052 [quant-ph].
  - [3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Phys. Rev. A* **86**, 032324 (2012).
  - [4] M. B. Hastings and J. Haah, Dynamically Generated Logical Qubits, *Quantum* **5**, 564 (2021).
  - [5] A. Kitaev, Anyons in an exactly solved model and beyond, *Annals of Physics* **321**, 2 (2006), january Special Issue.
  - [6] C. Vuillot, Planar floquet codes (2021), arXiv:2110.05348 [quant-ph].

- [7] J. Haah and M. B. Hastings, Boundaries for the Honeycomb Code, *Quantum* **6**, 693 (2022).
- [8] C. Gidney, M. Newman, A. Fowler, and M. Broughton, A Fault-Tolerant Honeycomb Memory, *Quantum* **5**, 605 (2021).
- [9] C. Gidney, M. Newman, and M. McEwen, Benchmarking the Planar Honeycomb Code, *Quantum* **6**, 813 (2022).
- [10] A. Paetznick, C. Knapp, N. Delfosse, B. Bauer, J. Haah, M. B. Hastings, and M. P. da Silva, Performance of planar floquet codes with majorana-based qubits, *PRX Quantum* **4**, 010310 (2023).
- [11] J. R. Wootton, Measurements of floquet code plaquette stabilizers (2022), arXiv:2210.13154 [quant-ph].
- [12] M. Davydova, N. Tantivasadakarn, and S. Balasubramanian, Floquet codes without parent subsystem codes, *PRX Quantum* **4**, 020341 (2023).
- [13] M. S. Kesselring, J. C. M. de la Fuente, F. Thomsen, J. Eisert, S. D. Bartlett, and B. J. Brown, Anyon condensation and the color code (2022), arXiv:2212.00042 [quant-ph].
- [14] A. Townsend-Teague, J. Magdalena de la Fuente, and M. Kesselring, Floquetifying the colour code, *Electronic Proceedings in Theoretical Computer Science* **384**, 265–303 (2023).
- [15] O. Higgott and N. P. Breuckmann, Constructions and performance of hyperbolic and semi-hyperbolic floquet codes (2023), arXiv:2308.03750 [quant-ph].
- [16] A. Fahimniya, H. Dehghani, K. Bharti, S. Mathew, A. J. Kollár, A. V. Gorshkov, and M. J. Gullans, Fault-tolerant hyperbolic floquet quantum error correcting codes (2023), arXiv:2309.10033 [quant-ph].
- [17] Z. Zhang, D. Aasen, and S. Vijay,  $x$ -cube floquet code: A dynamical quantum error correcting code with a subextensive number of logical qubits, *Phys. Rev. B* **108**, 205116 (2023).
- [18] A. Dua, N. Tantivasadakarn, J. Sullivan, and T. D. Ellison, Engineering floquet codes by rewinding (2023), arXiv:2307.13668 [quant-ph].
- [19] T. D. Ellison, J. Sullivan, and A. Dua, Floquet codes with a twist (2023), arXiv:2306.08027 [quant-ph].
- [20] H. Bombin, D. Litinski, N. Nickerson, F. Pastawski, and S. Roberts, Unifying flavors of fault tolerance with the zx calculus (2023), arXiv:2303.08829 [quant-ph].
- [21] B. Coecke and R. Duncan, Interacting quantum observables: categorical algebra and diagrammatics, *New Journal of Physics* **13**, 043016 (2011).
- [22] D. Aasen, Z. Wang, and M. B. Hastings, Adiabatic paths of hamiltonians, symmetries of topological order, and automorphism codes, *Phys. Rev. B* **106**, 085122 (2022).
- [23] M. Davydova, N. Tantivasadakarn, S. Balasubramanian, and D. Aasen, Quantum computation from dynamic automorphism codes (2023), arXiv:2307.10353 [quant-ph].
- [24] M. Davydova, Logical operations in floquet codes (talk at simons institute, berkeley) (2023).
- [25] D. Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, *Phys. Rev. A* **73**, 012340 (2006).
- [26] D. Poulin, Stabilizer formalism for operator quantum error correction, *Phys. Rev. Lett.* **95**, 230504 (2005).
- [27] N. P. Breuckmann, Quantum subsystem codes: Their theory and use (bachelor thesis) (2010).
- [28] M. Li, D. Miller, M. Newman, Y. Wu, and K. R. Brown, 2d compass codes, *Phys. Rev. X* **9**, 021041 (2019).
- [29] N. Delfosse, A. Paetznick, J. Haah, and M. B. Hastings, Splitting decoders for correcting hypergraph faults (2023), arXiv:2309.15354 [quant-ph].
- [30] C. Gidney and D. Bacon, Less bacon more threshold (2023), arXiv:2305.12046 [quant-ph].

## Appendix A: Preservation of logical information

Here, we expand on the discussion on logical preservation between any two stabilizer codes. In particular, this discussion applies to the ISGs of the Floquet-Bacon-Shor family of codes described above. Let us suppose that  $|b^A\rangle$  is a codeword such that it is an eigenvector of the logical operator  $Z^A$  of code  $A$  with eigenvalue  $\alpha$ . The following short calculation shows that the projected codeword  $|b'^A\rangle = \prod_{s \in \mathcal{S}^B} \left(\frac{\mathbb{I}+s}{2}\right) |b^A\rangle$  is an eigenvector of the logical operator  $Z^B$  of code  $B$  with the same eigenvalue  $\alpha$  whenever  $s^A Z^A = s^B Z^B$  for some  $s^A \in \mathcal{S}^A$  and  $s^B \in \mathcal{S}^B$ .

$$\begin{aligned}
 Z^B |b'^A\rangle &= Z^B \prod_{s \in \mathcal{S}^B} \left(\frac{\mathbb{I}+s}{2}\right) |b^A\rangle \\
 &= \prod_{s \in \mathcal{S}^B} \left(\frac{\mathbb{I}+s}{2}\right) s^B Z^B |b^A\rangle \\
 &= \prod_{s \in \mathcal{S}^B} \left(\frac{\mathbb{I}+s}{2}\right) s^A Z^A |b^A\rangle \\
 &= \prod_{s \in \mathcal{S}^B} \left(\frac{\mathbb{I}+s}{2}\right) \alpha |b^A\rangle \\
 &= \alpha |b'^A\rangle
 \end{aligned} \tag{A1}$$

In other words, the projected codeword  $|b'^A\rangle$  behaves exactly like a codeword of code  $B$ , i.e.  $|b'^A\rangle = |b^B\rangle$ . In turn, this implies that when we project the state  $|\psi^A\rangle = \alpha |0^A\rangle + \beta |1^A\rangle$  to  $|\psi^B\rangle = \alpha |0'^A\rangle + \beta |1'^A\rangle$ , we will ensure that  $Z^B |0'^A\rangle = |0'^A\rangle$  and  $Z^B |1'^A\rangle = -|1'^A\rangle$ , so that while the logical codewords  $|0^B\rangle = |0'^A\rangle$  and  $|1^B\rangle = |1'^A\rangle$  have deformed, the information has remained intact.

As a simple illustration of this, consider the two familiar 3-qubit repetition codes

$$\begin{aligned}
 \mathcal{S}^{(0)} &= \langle Z_1 Z_2, Z_2 Z_3 \rangle \\
 \mathcal{S}^{(1)} &= \langle X_1 X_2, X_2 X_3 \rangle
 \end{aligned} \tag{A2}$$

with the same logical operators for both codes

$$\begin{aligned}
 \bar{X} &= X_1 X_2 X_3 \\
 \bar{Z} &= Z_1 Z_2 Z_3
 \end{aligned} \tag{A3}$$

and codewords

$$\begin{aligned}
 |\bar{0}\rangle^{(0)} &= |000\rangle, \quad |\bar{0}\rangle^{(1)} = \frac{1}{\sqrt{2}} (|+++ \rangle + |-- - \rangle) \\
 |\bar{1}\rangle^{(0)} &= |111\rangle, \quad |\bar{1}\rangle^{(1)} = \frac{1}{\sqrt{2}} (|+++ \rangle - |-- - \rangle)
 \end{aligned} \tag{A4}$$

It is clear that the logical operators defined above satisfy the requirements of preserving logical information from one round to the next, by simply picking  $s^{(0)} = s^{(1)} = \mathbb{I}$ . Note that this is why we require the unconventional definition of the logical operators for  $\mathcal{S}^{(1)}$  which are usually defined the other way round, i.e.  $\bar{X} \Leftrightarrow \bar{Z}$ . Similarly, the codewords for  $\mathcal{S}^{(1)}$  are different from the conventional  $|\bar{0}\rangle = |+++ \rangle$  and  $|\bar{1}\rangle = |-- - \rangle$ .

If we view the two repetition codes as ISGs of a 3-qubit Floquet code, then even though logical information is conserved from one round to the next, the code distance is 1 since no stabilizers remain preserved between rounds and this cannot even detect single-qubit  $Z$  ( $X$ ) errors, so that such errors become logical errors that persist once they occur.

We also note that Eq. (2) guarantees that the commutation structure of the logical operators remains intact. Using identities  $[A, BC] = B[A, C] + [A, B]C$  and  $[AB, C] = A[B, C] + [A, C]B$ , we can quickly prove that  $[O_1^A, O_2^A] = 0 \Rightarrow [s_1^A O_1^A, s_2^A O_2^A] = 0$ , for any  $s_1^A, s_2^A \in \mathcal{S}^A$ . Under the assumption of Eq. (2), this implies that

$$[s_1^B O_1^B, s_2^B O_2^B] = 0 \quad (\text{A5})$$

Again, using commutator identities, and the facts that  $O_{1,2}^B$  are logical operators at round  $B$ , i.e.  $[O_{1,2}^B, s^B] = 0$  for all  $s^B \in \mathcal{S}^B$ , we find that this implies  $[O_1^B, O_2^B] = 0$ . The case for the anti-commutator is proved similarly,

using anti-commutator identities  $\{A, BC\} = [A, B]C + B\{A, C\}$  and  $\{AB, C\} = A\{B, C\} - [A, C]B = A[B, C] + \{A, C\}B$ .

## Appendix B: Self-correction

Suppose that  $X$  errors occur on two qubits that lie on the row above cell A on either side of the plaquette column AD between rounds 0 and 1 mod 4. Such a 2-qubit error is equivalent, upto stabilizer transformations of the ISG at round 0 mod 4 to a 2-qubit  $XX$  error that occurs on the top edge of cell A. This anti-commutes with the logical  $\bar{Z}$  error of the dynamical qubit at round 0 mod 4, and thus contains a factor of the logical  $\bar{X}$  operator. Specifically, it is given by  $\bar{X}_A \bar{X}_{A'}$ , where cell  $A'$  is defined as the cell on top of cell A. In the next round 1 mod 4, we would measure a subset of all the  $ZZ$  checks which, in addition to other transient stabilizers, would measure the gauge qubit operator  $\bar{Z}_{A'}$ , as well as the check operator  $\bar{Z}_A \bar{Z}_B$ . Suppose, without loss of generality, that in the absence of any errors, the measurements of both would have yielded result +1. Due to the error, their values will flip. In the next round 2 mod 4, we measure the 2-qubit check  $\bar{X}_A \bar{X}_{A'}$ , and use the property that  $(\mathbb{I} \pm \sigma) = \pm(\mathbb{I} \pm \sigma)\sigma$  for any Pauli operator  $\sigma$ . In the measurement round 2 mod 4, this check takes on some random value  $\pm 1$ . The entire sequence of steps can be followed as

$$\begin{aligned} |\psi^{(0)}\rangle &\xrightarrow{\text{error}} \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\ &\xrightarrow{\text{detection}} [\dots (\mathbb{I} - \bar{Z}_A \bar{Z}_B) (\mathbb{I} - \bar{Z}_{A'})] \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\ &\xrightarrow{\text{measurement}} [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] [\dots (\mathbb{I} - \bar{Z}_A \bar{Z}_B) (\mathbb{I} - \bar{Z}_{A'})] \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\ &= \pm [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] \bar{X}_A \bar{X}_{A'} [\dots (\mathbb{I} - \bar{Z}_A \bar{Z}_B) (\mathbb{I} - \bar{Z}_{A'})] \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\ &= \pm [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] [\dots (\mathbb{I} + \bar{Z}_A \bar{Z}_B) (\mathbb{I} + \bar{Z}_{A'})] |\psi^{(0)}\rangle \\ &= \pm [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] |\psi^{(1)}\rangle \\ &= \pm |\psi^{(2)}\rangle \end{aligned} \quad (\text{B1})$$

so that the entire situation is indistinguishable from reaching the correct logical state  $|\psi^{(2)}\rangle$  at round 2 mod 4 without any errors.

The above discussion also implies that the correcting operator for a single  $X$  error on any single row is the same whether it lies on the left or the right of the blue domain wall, even though they produce distinct syndromes. Suppose that such a single qubit  $X$  error occurred on the right side of the blue domain wall along column AD on the row above cell A between rounds 0 and 1 mod 4. Let  $B'$  be the cell above B, and  $s_{AB}$  and  $s_{A'B'}$  denote the permanent  $Z$ -type stabilizers along plaquette rows  $AB$  and  $A'B'$  respectively. Then, this error is equivalent to  $\bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} \bar{X}_A \bar{X}_{A'}$ , up to stabilizer transforma-

tions of the ISG at round 0 mod 4, where the product  $\bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}}$  gives a single qubit  $X$  operator that lies on the left-most column on this row. At round 1 mod 4, this error will flip the value of the permanent  $Z$ -type stabilizer  $\bar{Z}_{s_{A'B'}}$  as well as the transient stabilizer  $\bar{Z}_{A'}$ . Note that the permanent  $Z$ -type stabilizer  $\bar{Z}_{s_{AB}}$  does not get measured in this round. Suppose, without loss of generality, that in the absence of such an error, both  $\bar{Z}_{s_{A'B'}}$  and  $\bar{Z}_{A'}$  would have measured out to be +1, so that this error flips the recorded values to be -1 instead.

A correction operation of  $\bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}}$  suffices after round 1 mod 4, as the subsequent measurements of round 2 mod 4 then restore the correct logical state, as can be

seen through the following sequence of transformations

---


$$\begin{aligned}
|\psi^{(0)}\rangle &\xrightarrow{\text{error}} \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\
&\xrightarrow{\text{detection}} [\dots (\mathbb{I} - \bar{Z}_{s_{A'B'}})(\mathbb{I} - \bar{Z}_{A'})] \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\
&\xrightarrow{\text{correction}} \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} [\dots (\mathbb{I} - \bar{Z}_{s_{A'B'}})(\mathbb{I} - \bar{Z}_{A'})] \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\
&\xrightarrow{\text{measurements}} [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} [\dots (\mathbb{I} - \bar{Z}_{s_{A'B'}})(\mathbb{I} - \bar{Z}_{A'})] \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\
&= \pm [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] \bar{X}_A \bar{X}_{A'} \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} [\dots (\mathbb{I} - \bar{Z}_{s_{A'B'}})(\mathbb{I} - \bar{Z}_{A'})] \bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}} \bar{X}_A \bar{X}_{A'} |\psi^{(0)}\rangle \\
&= \pm [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] [\dots (\mathbb{I} + \bar{Z}_{s_{A'B'}})(\mathbb{I} + \bar{Z}_{A'})] |\psi^{(0)}\rangle \\
&= \pm [\dots (\mathbb{I} \pm \bar{X}_A \bar{X}_{A'})] |\psi^{(1)}\rangle \\
&= \pm |\psi^{(2)}\rangle
\end{aligned}
\tag{B2}$$

Note that the same correction operation would also work if the factor of  $\bar{X}_A \bar{X}_{A'}$  was absent from the error, and it had instead occurred on the left side of the blue domain

---

wall on that row, which would have been equivalent to  $\bar{X}_{s_{AB}} \bar{X}_{s_{A'B'}}$  upto transformations of the ISG at round 0 mod 4.