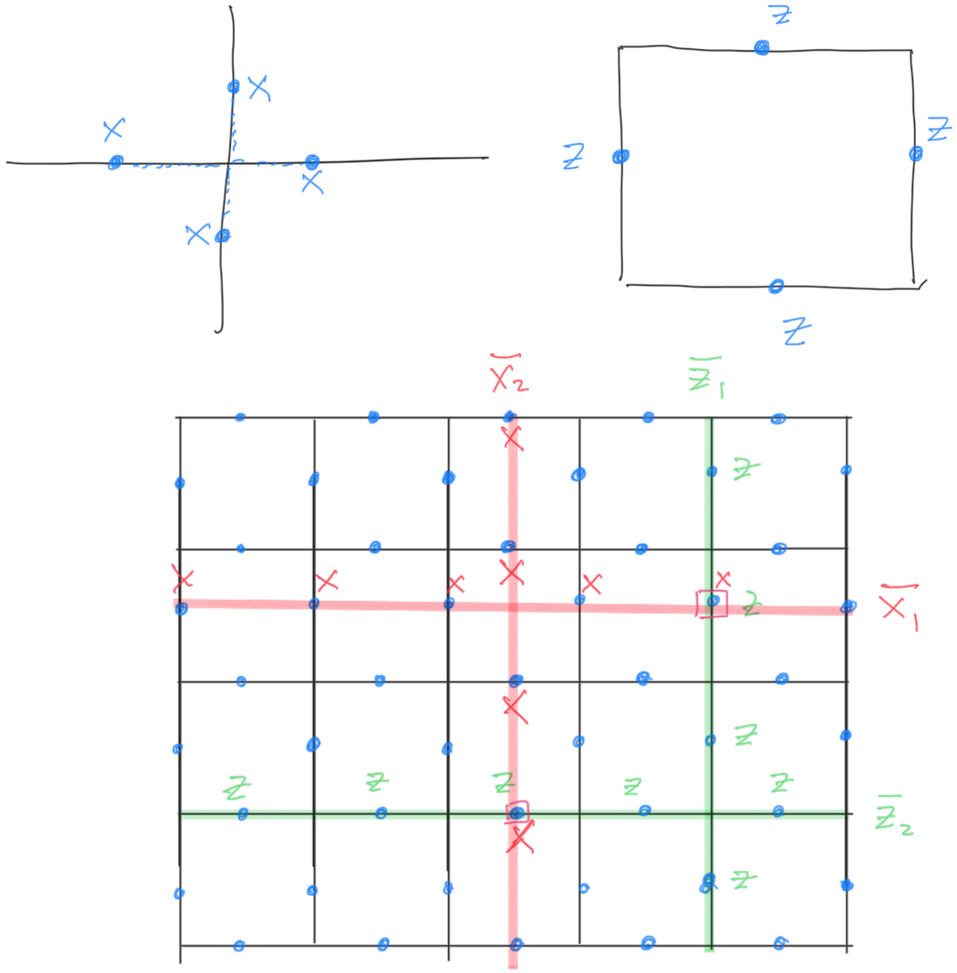
Floquet Codes

With a few relations to Fracton Phases

Edward Kim

Topological Quantum Codes

- The Toric Code and its variants have been shown to be promising candidates for fault-tolerant universal Quantum computation.
- Local stabilizer interactions undergird the detection and correction of Pauli errors.
- The Surface code is one such variant with boundary which has been studied and extended in sophisticated ways.
    1. Techniques such as Lattice Surgery, Twists, Fold-transversal Logical Gates have been shown to facilitate fault-tolerant QC
    2. Many approaches to error decoding have been found, such as Minimum-weight Perfect Matching, Union-Find decoding, etc.



Honeycomb Code

A Departure from Traditional Topological Codes

- Kiteav's honeycomb model was studied in earlier works as an error-correcting code.
- The gauge checks were taken to be weight-two terms of the Hamiltonian of the honeycomb model.
- It was discovered that *as a subsystem code* the honeycomb model cannot encode any logical qubits.
- In order to encode logical qubits, a new approach is required beyond the usual stabilizer/subsystem formalism.

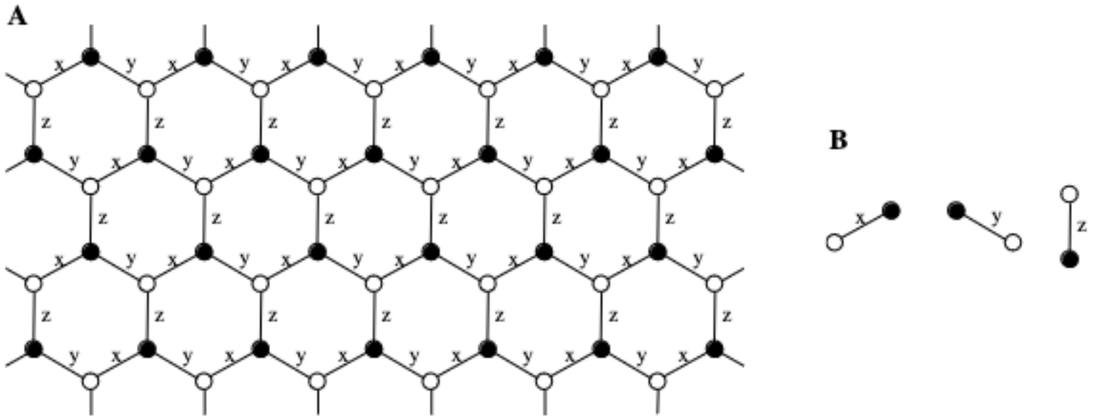


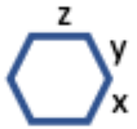
Fig. 3. Three types of links in the honeycomb lattice.

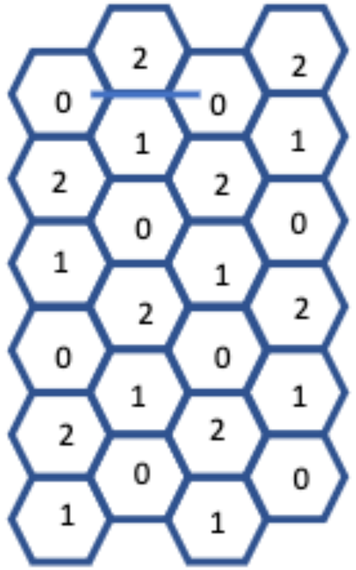
$$H = -J_x \sum_{x\text{-links}} \sigma_j^x \sigma_k^x - J_y \sum_{y\text{-links}} \sigma_j^y \sigma_k^y - J_z \sum_{z\text{-links}} \sigma_j^z \sigma_k^z,$$

Source: A. Y. Kitaev, Anyons in an exactly solved model and beyond.

Solution: Periodic Measurement Schedule

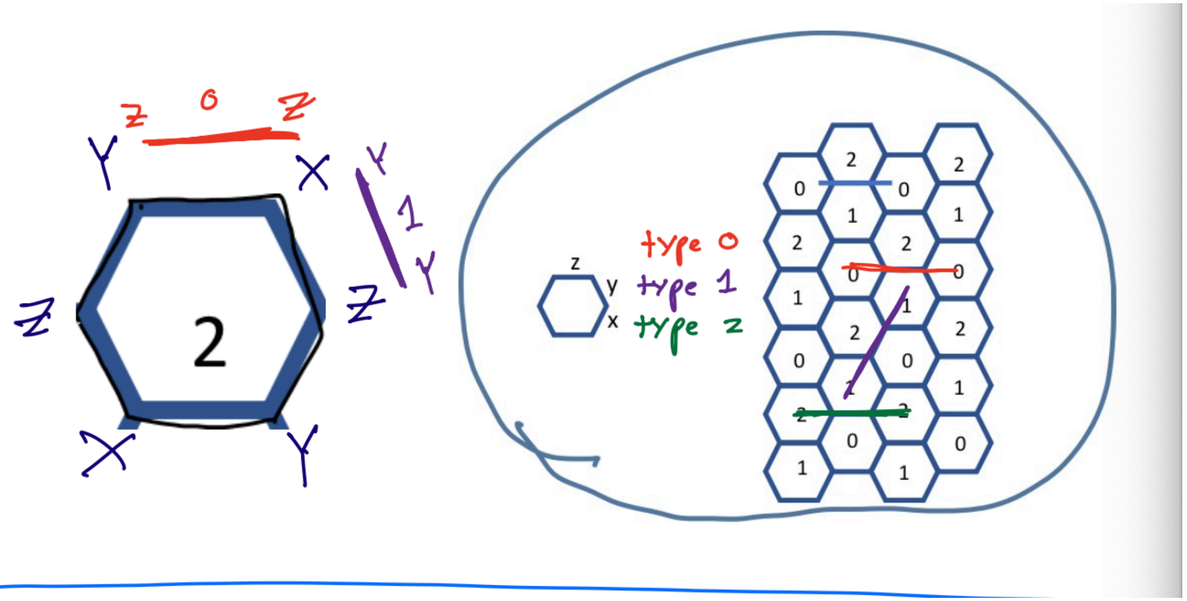
- Although it cannot encode logical qubits directly, the honeycomb model does support logical qubits!
- Instead of generating the stabilizer group through the gauge checks all at once, measure the checks **sequentially through rounds** and in a **particular order.**
- This generates an *Instantaneous Stabilizer Group (ISG),* a stabilizer group which manifests according to the type of checks measured at each round.
- The ISG *evolves* as the measurement schedule proceeds.





Honeycomb Code Definition

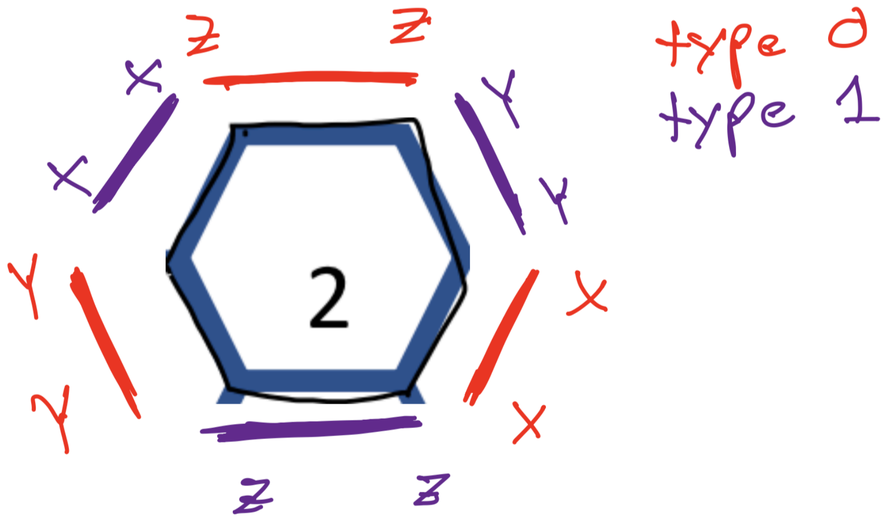
- Qubits are identified with the vertices.
- Each qubit is incident to three edges.
- Each edge is labeled with a Pauli operator and a type.
- The type of the edges are determined by the hexagon plaquettes it connects.
- Boundaries are periodic.
- Types of hexagons are such that adjacent hexagons cannot share the same label.
- The plaquette stabilizers are defined as the product of the gauge checks around the boundary of hexagon plaquettes.



Types and Pauli flavor of edges are labeled explicitly. The long, colored lines indicate edge type. Observe that the plaquette stabilizers commute with all gauge checks on the lattice.

Measurement Schedule

- The measurement schedule proceeds by measuring checks of a single type at each round.
- The schedule measures **type 0, type 1, and type 2 checks sequentially**. This schedule is continuously repeated.
- The checks of the next round anti-commute with the checks of the current round.



The checks of the next round anti-commute with the current round's checks

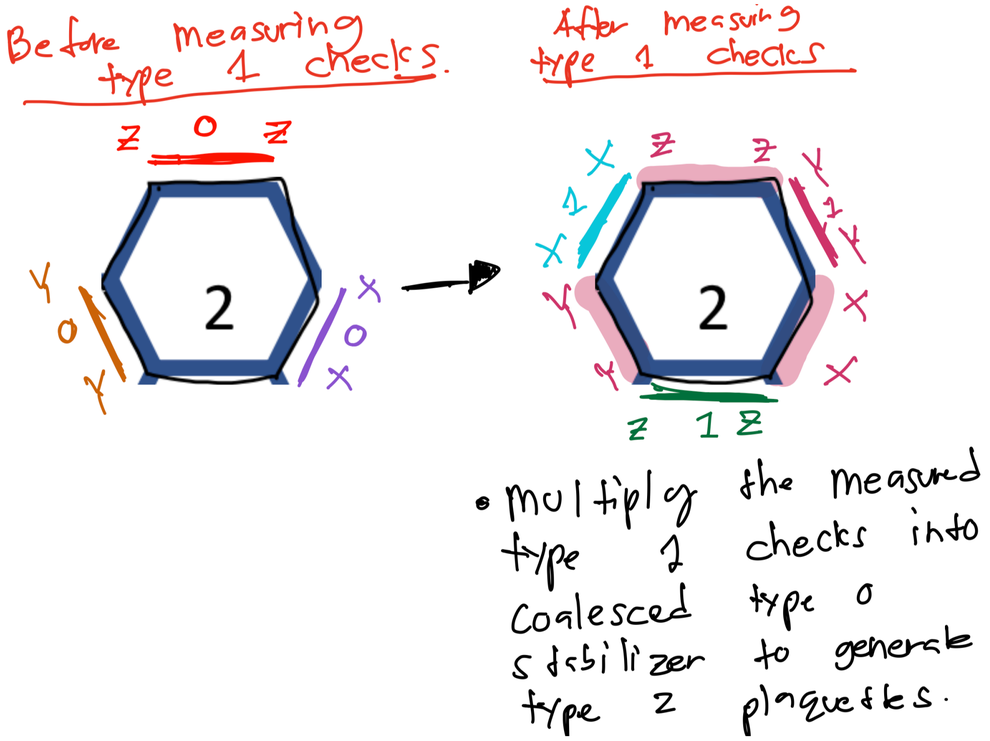
Generating “Static” Stabilizers

- For example, suppose we have just concluded round 0.

  A. During the next round (round 1), the type 1 checks will be measured.

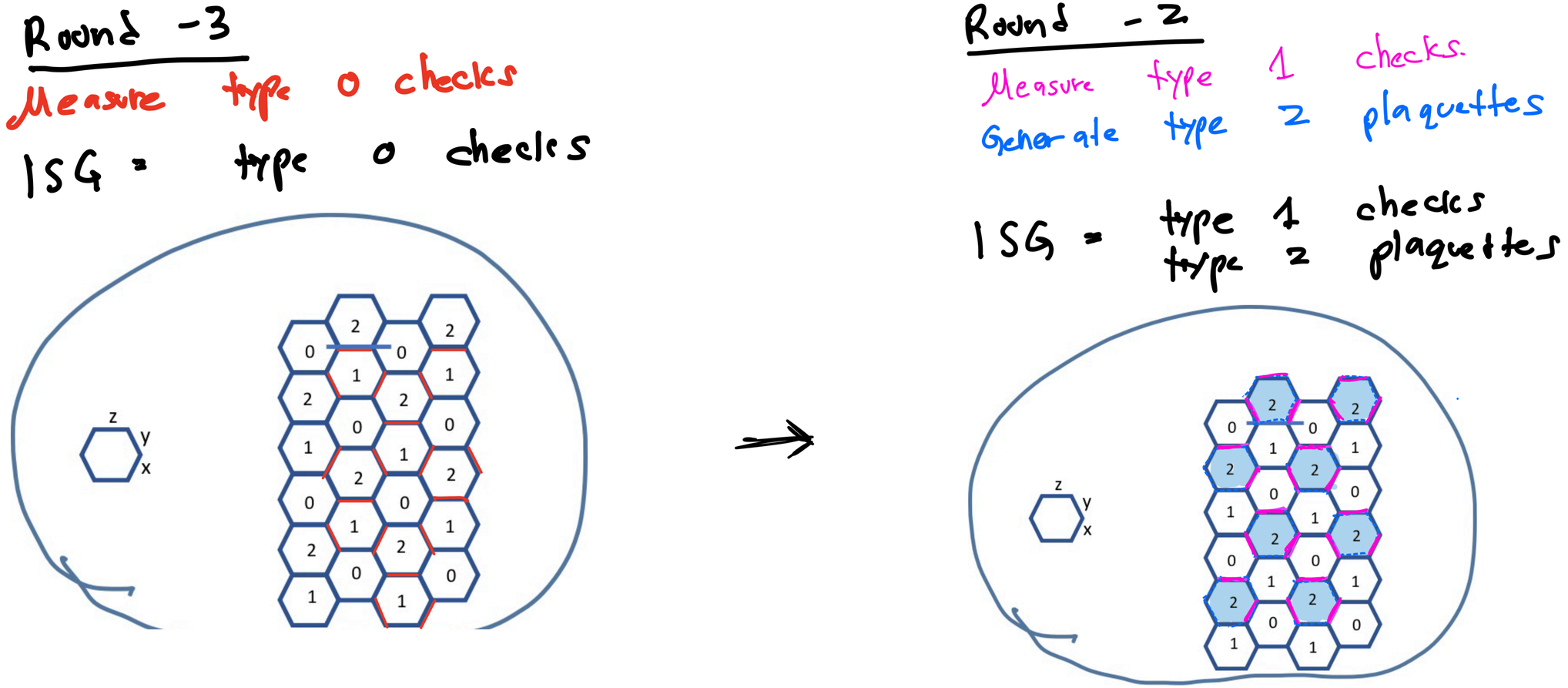
  B. The type 1 checks and the previous round’s type 0 checks will ***generate the type 2 plaquettes*.**

- Since the plaquettes commute with all gauge checks, the plaquette stabilizers of all types will ***remain as stabilizers at every round*** after initialization!



- Different colors represent different stabilizers in the ISG.
- The number associated with each check is the type.
- After the measurement of the type 1 checks, the type 0 checks coalesce into *one* stabilizer.

Code Initialization



A trace of initializing the honeycomb code. Notice how the ISG adds plaquette stabilizers of all types by the end of initialization.

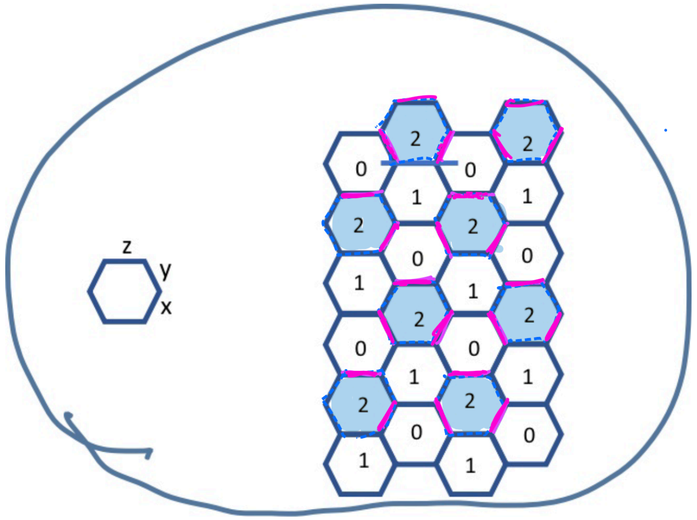
Code Initialization

Round - 2

Measure type 1 checks.

Generate type 2 plaquettes

ISG = type 1 checks
type 2 plaquettes

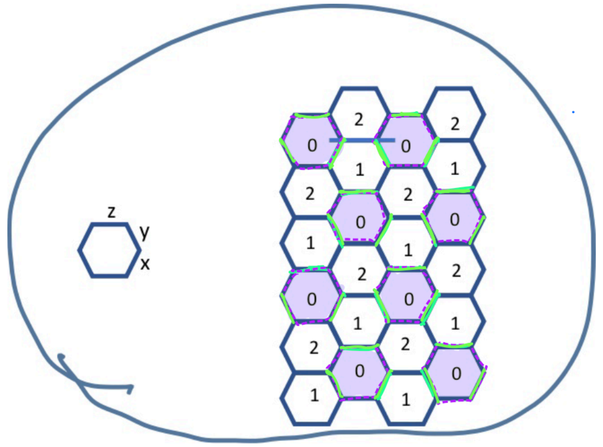


Round - 1

Measure type 2 checks

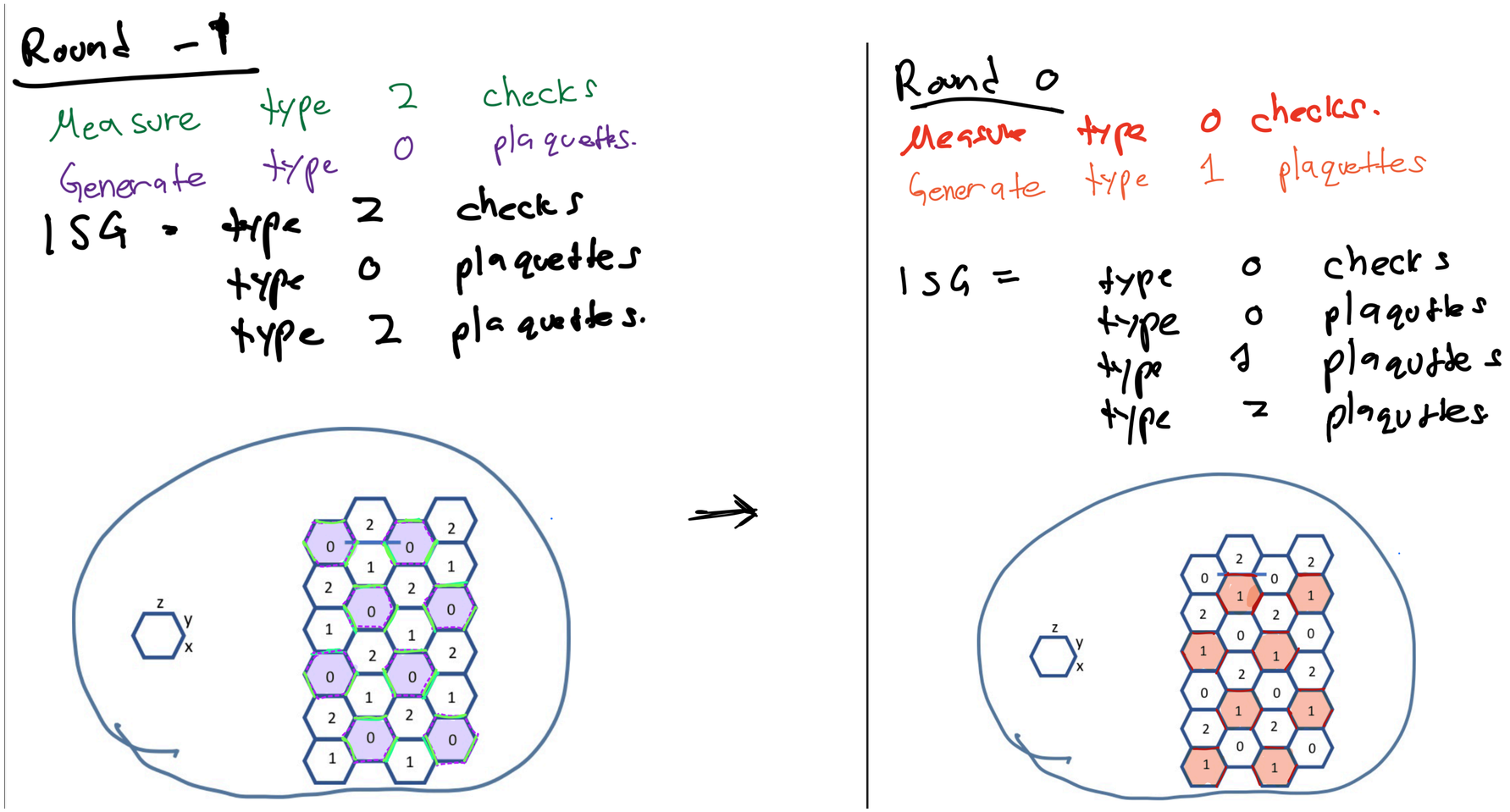
Generate type 0 plaquettes.

ISG = type 2 checks
type 0 plaquettes
type 2 plaquettes.



A trace of initializing the honeycomb code. Notice how the ISG adds plaquette stabilizers of all types by the end of initialization.

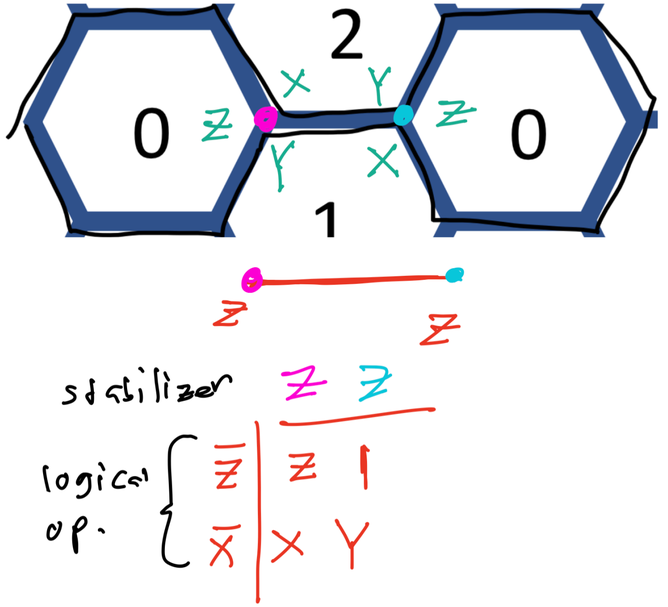
Code Initialization



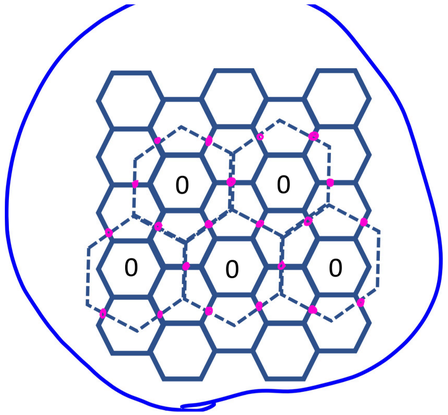
A trace of initializing the honeycomb code. Notice how the ISG adds plaquette stabilizers of all types by the end of initialization.

Embedded Toric Codes

- After each round r, each measured gauge check of type r projects the qubits within its support down to the space of an effective qubit (subspace of dimension 2).
- The plaquettes of type r act as *logical plaquette stabilizers.*
- The plaquettes of the complementary types (r+1, r+2 mod 3) act as *logical vertex stabilizers.*
- All together, an embedded toric “super code” can be identified as a result.



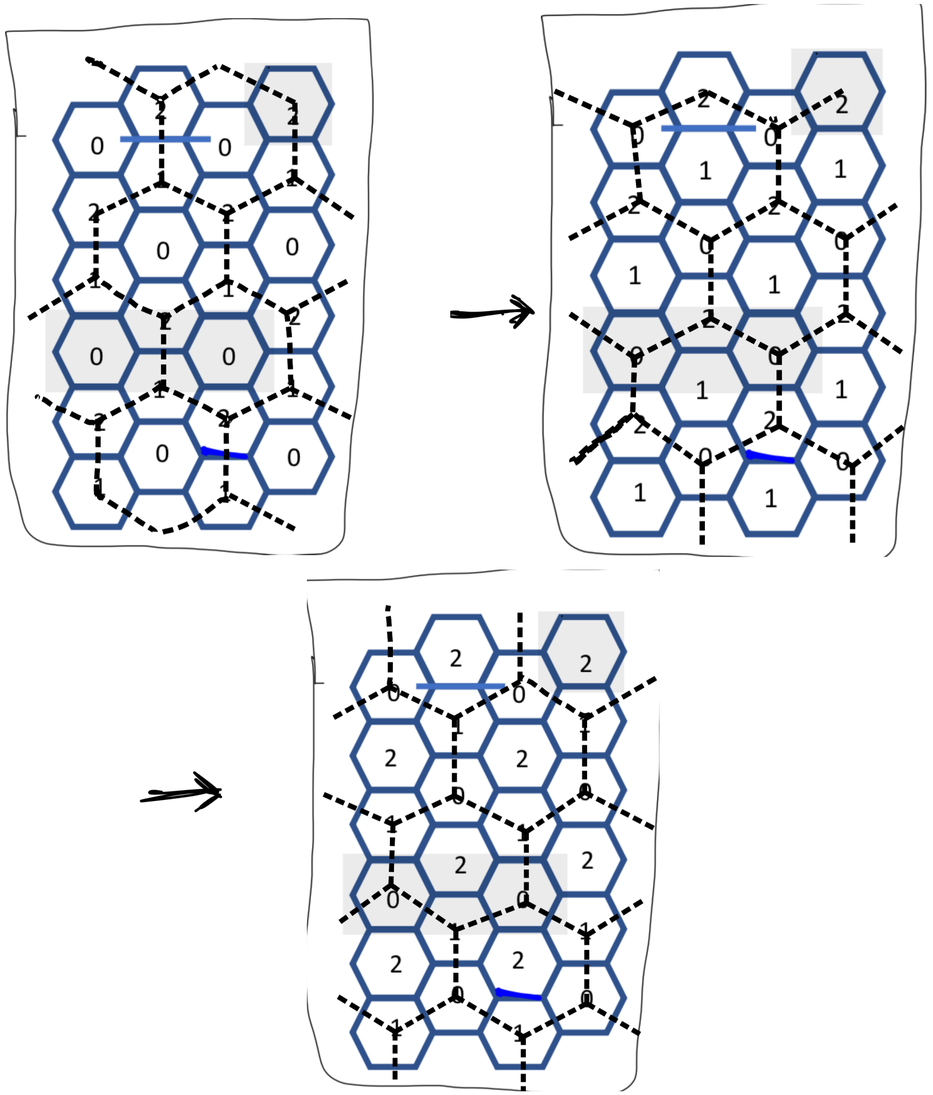
The code after round 0 is finished. Note that the two qubits within the support of a type 0 check project down to that of a single qubit.



The superhexagonal lattice identified with the embedded toric code. The effective qubits are labeled with a pink dot.

Evolution of Embedded Toric Code.

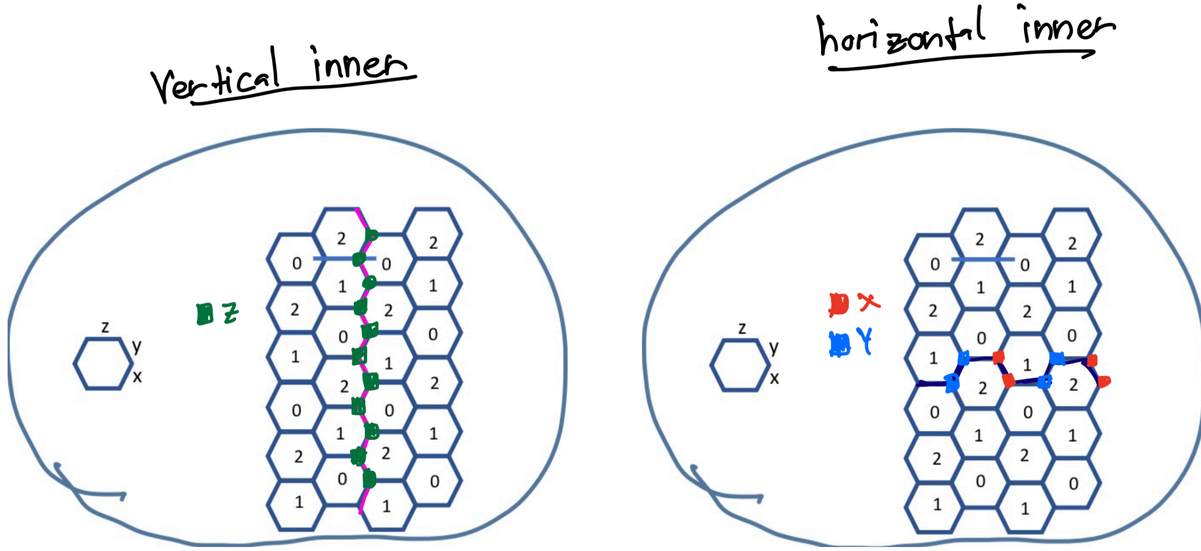
- After initialization, every round has such a copy of the embedded toric code.
- After measuring the type r checks during round r, the embedded toric code will have the type r plaquettes acting as **“super-plaquettes”** and the type $r+1$ and type $r+2$ plaquettes acting as **“super-vertices.”**



Successive shifts from round 0 to round 2.

Inner/Outer Logical Operators

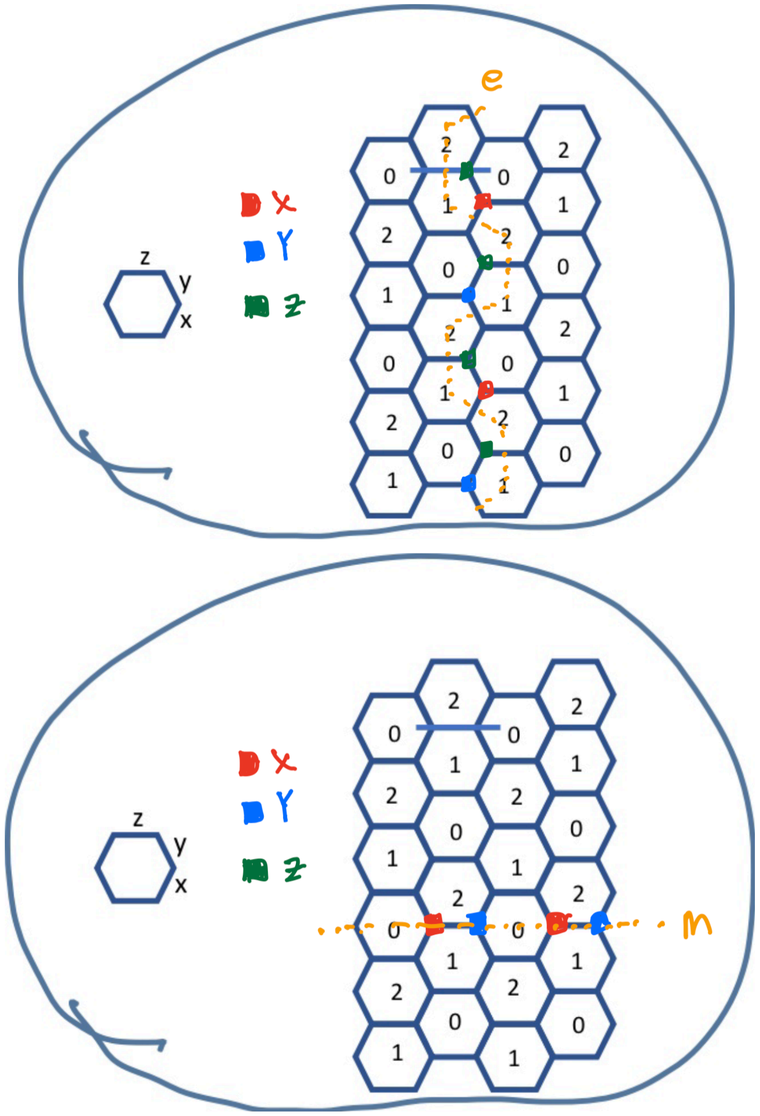
- Counting the number of independent stabilizers at every round shows that the code encodes *two logical qubits every round*.
- Two types of logical operators can be identified within the honeycomb code.
- Inner logical operators **commute** with every gauge check. Hence, they belong the gauge subgroup of the code, and the gauge checks do not destroy their encoded state.
- Hence, inner logical operators stay static throughout evolution of the ISG.



Representatives of the vertical and horizontal inner logical operators.

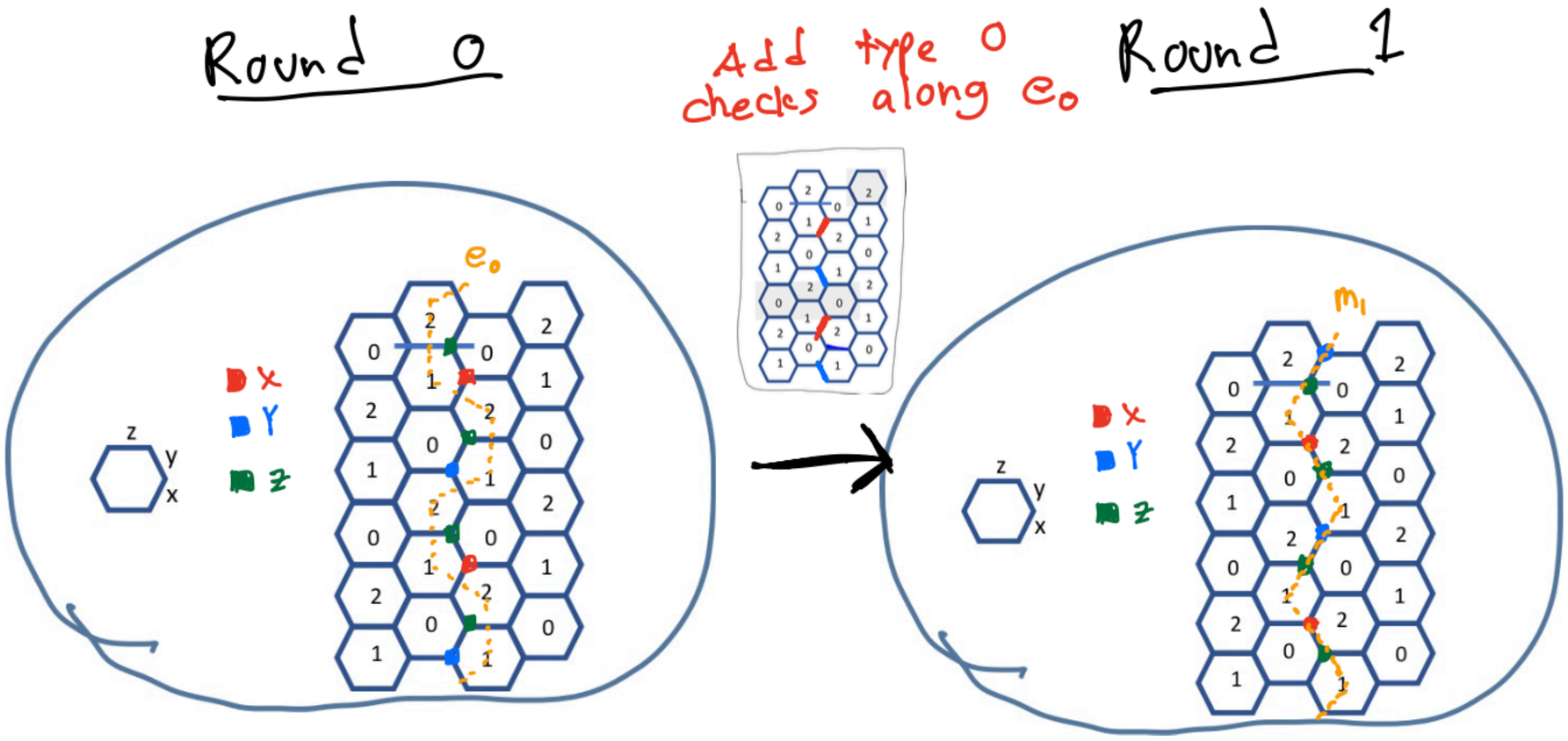
Inner/Outer Logical Operators

- The outer logical operators correspond to direct/ dual logical string operators along the embedded toric code.
- Outer logical operators ***anti-commute*** with some gauge check.
- Logical operators corresponding to direct strings are called *electric operators (e).*
- Those corresponding to dual strings are called *magnetic operators (m).*
- Since the copy of the embedded toric code evolves, so do the outer logical operators.



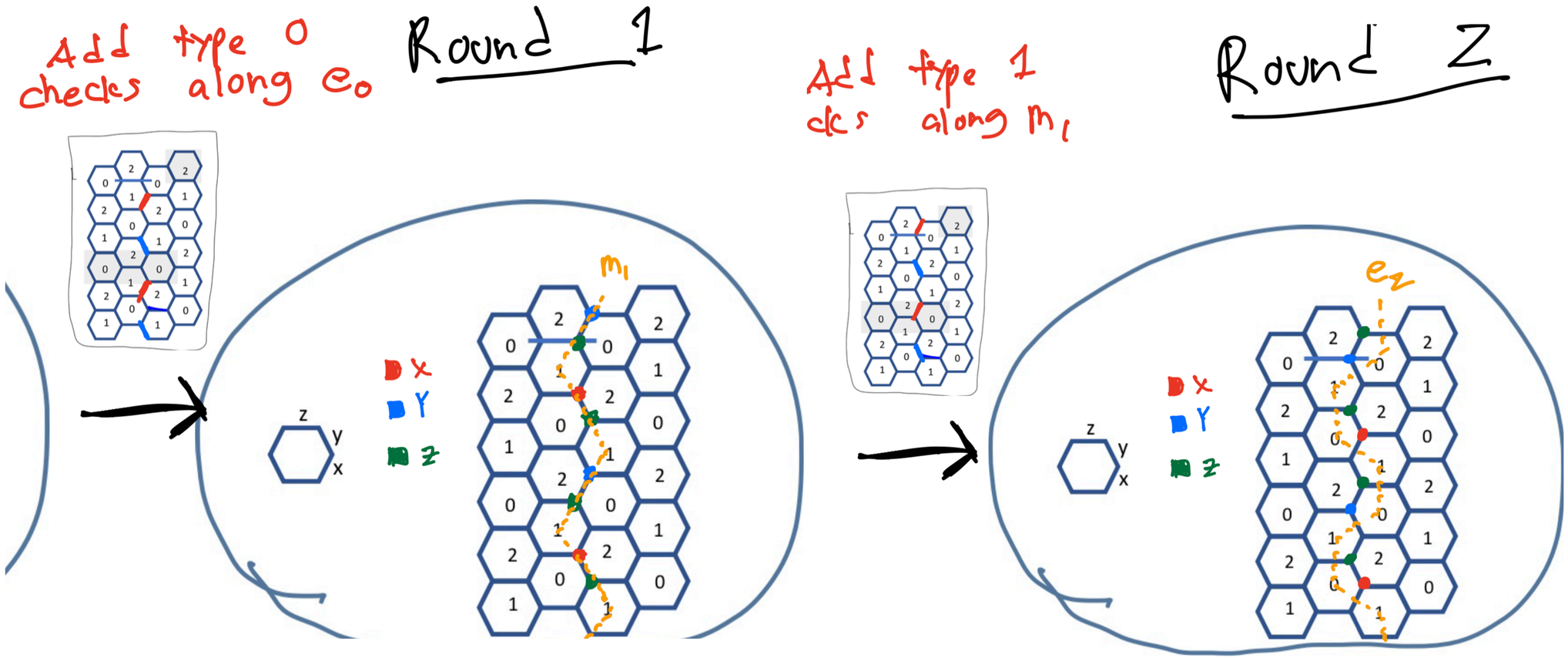
The electric and magnetic operators in respect to round 0 mod 3.

Evolution of Outer Logical Operator.



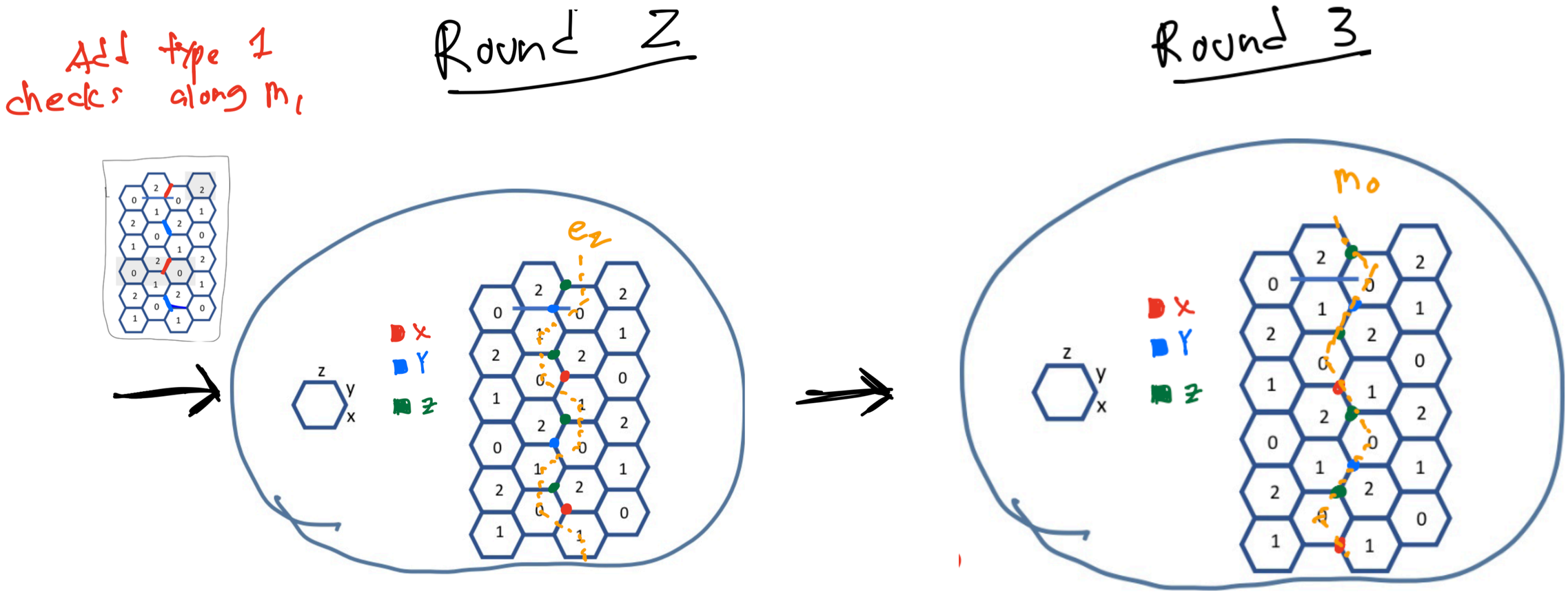
Evolution of logical operator from round 0 to round 3 after code initialization.

Evolution of Outer Logical Operator.



Evolution of logical operator from round 0 to round 3 after code initialization.

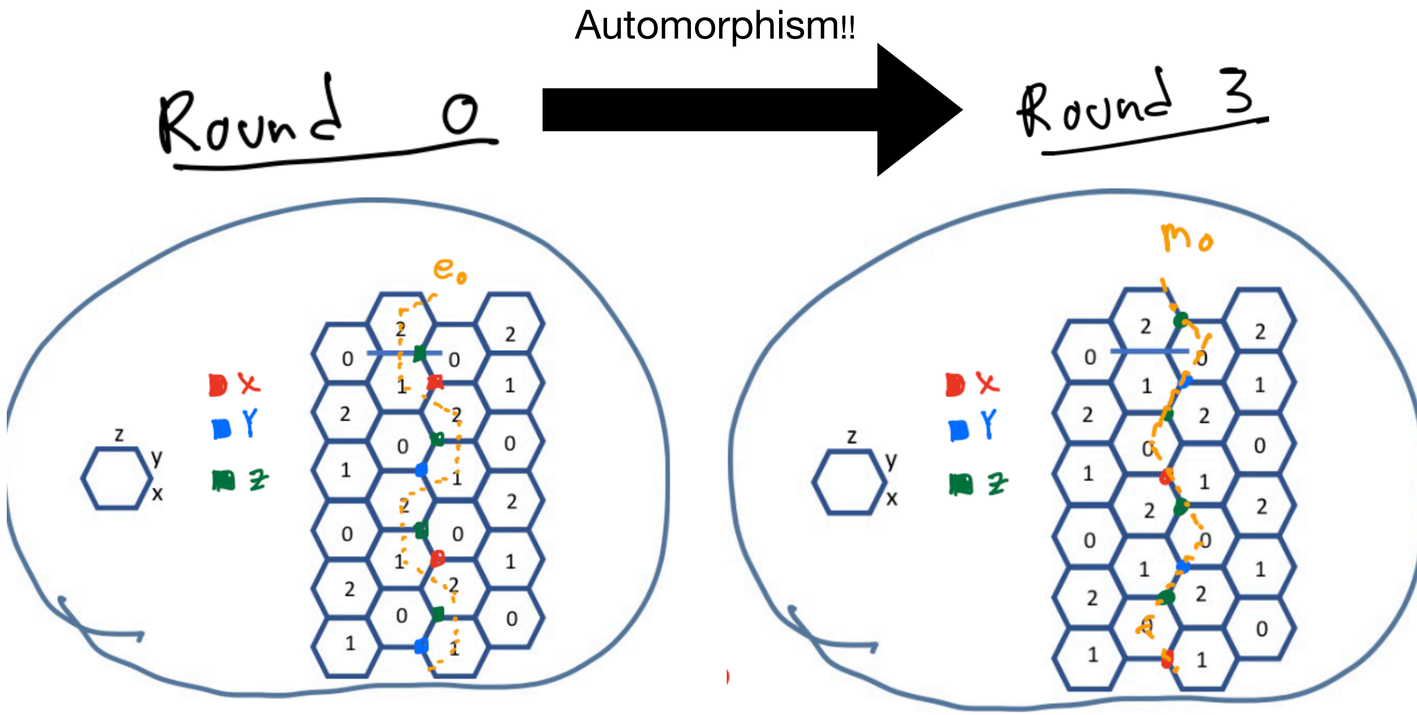
Evolution of Outer Logical Operator.



Evolution of logical operator from round 0 to round 3 after code initialization.

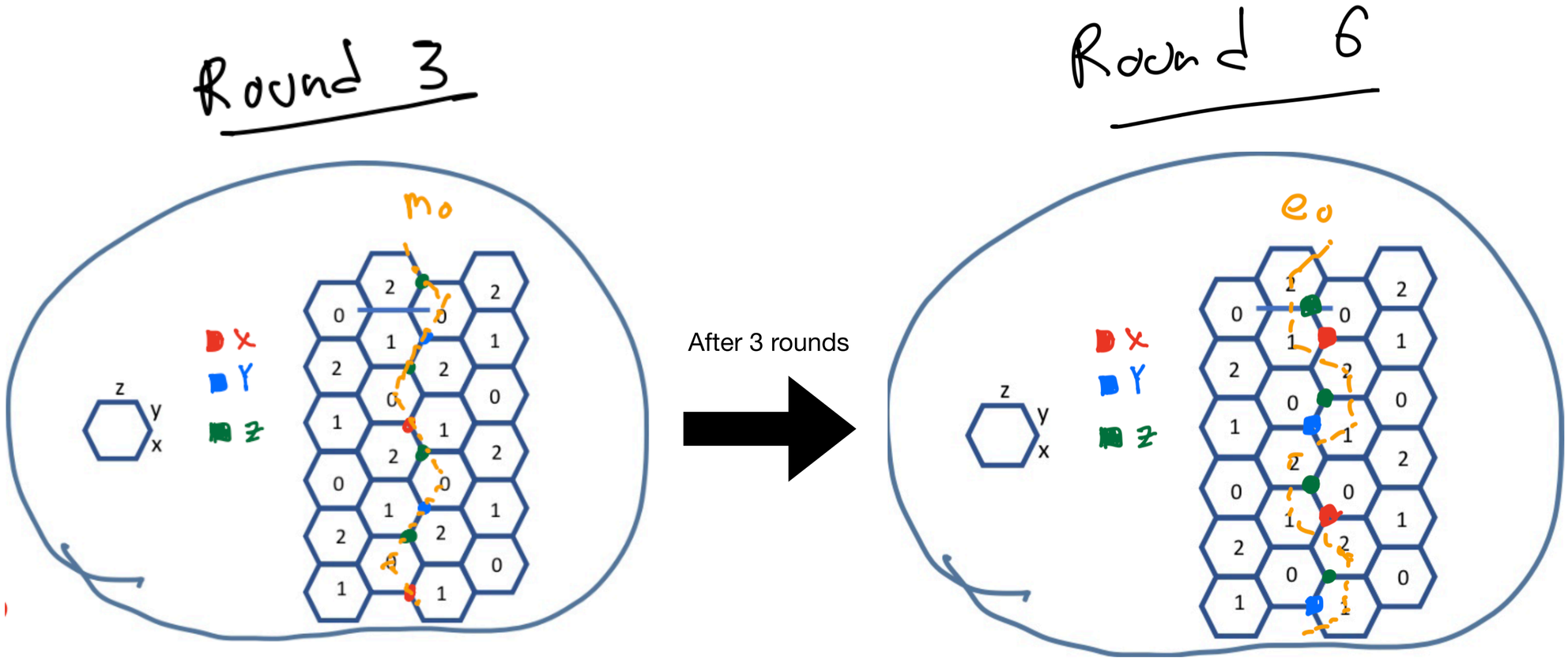
E/M Automorphisms

- Due to the effects of the periodic measurement schedule, electric operators of the current round evolve into magnetic operators of the next round and vice-versa.
- As the period (3 rounds) is odd, **electric operators** from round r evolve into **magnetic operators** by the end of round r+3.
- Since the ISGs of rounds r and r+3 coincide, one period of measurements induces **an automorphism** of the toric code.



The evolution of an electric operator to a magnetic operator in one period. We start at round 0 mod 3.

E/M Automorphisms



- Evolution of the electric operator at round 3 to the magnetic operator at round 6.
- Note that the code returns to itself in six rounds, not three.

Two Caveats

- The measurement schedule here is particular as a badly constructed schedule will lead to the unintended measurement of a logical operator.

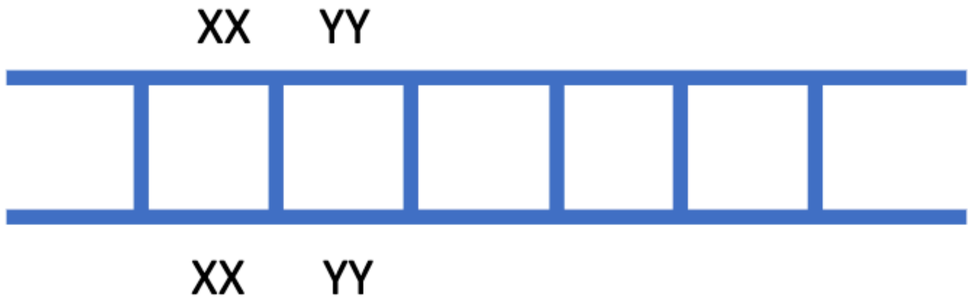


Figure 5: Ladder code. There is one qubit per vertex. Checks on vertical legs are ZZ on the two qubits. Checks on horizontal legs are alternately XX or YY; some of the horizontal checks are shown.

- Correlated errors could break fault-tolerance in certain instances.

Fracton Phases.

What do I know about the Physics? Not much so far!

- Fracton phases are characterized by immobile fractional excitations deemed as ***fractons.*** Fracton excitations are characterized by their inability to move without spawning additional fracton excitations.

- Two broad classes of fracton phases have been discovered:

    1. **Type 1 Phases:** These have fracton excitations appearing at the corners of *membrane operators*. Any mobile excitation must live in a lower dimensional subsystem

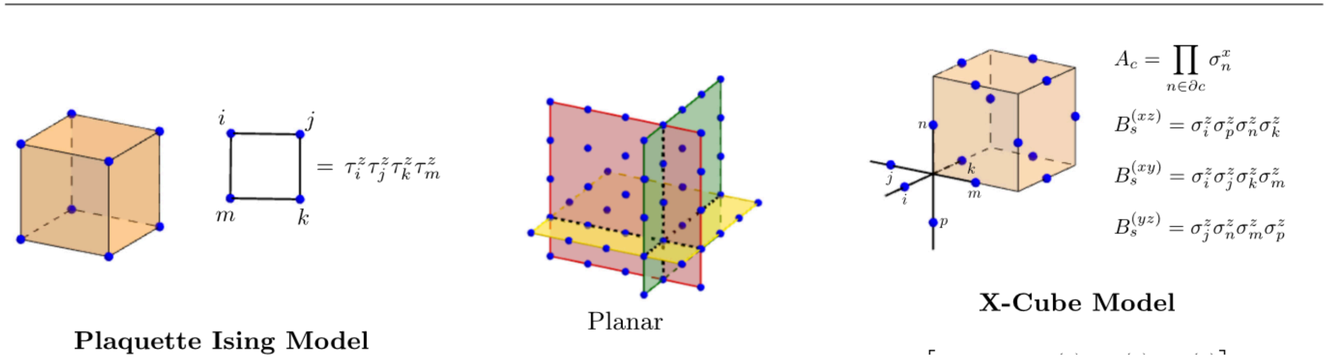
        A. **Examples**: X-Cube Model, CBLT Model.

    2. **Type 2 Phases:** The fracton excitations appear at the corners of *fractal operators.* No mobile excitations can exist.

        B. **Examples**: Haah's Code

X-Cube Model

- Type I Fracton model arising from a plaquette Ising model enriched with planar subsystem symmetry.
- The stabilizers will be “star” stabilizers oriented along xy, yz, and xz planes, and “cubic” stabilizers supported along the qubits incident to an edge of a cube.
- Notice how every qubit is incident to four cubic stabilizers.
- Hence, an application of σ^z will excite *four* cubic stabilizers. Repeated applications will push the fracton excitations to the corners of the lattice.



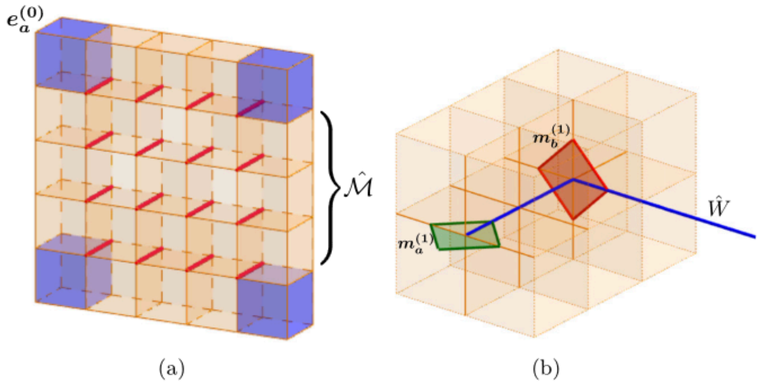


FIG. 1. The fundamental excitations of the X-cube model are shown in (a) and (b). Acting on the ground state of the X-cube model with a product of σ^z operators along the colored red links that lie within a flat, rectangular region $\mathcal{M}$ generates four fracton cube excitations $e_a^{(0)}$ at the corners of the region. A straight Wilson line of σ^x operators acting on the blue links in (b) isolates a pair of quasiparticles ($m_a^{(1)}$ or $m_b^{(1)}$) at the ends that are only free to move along the line. Attempting to move these quasiparticles in any other direction by introducing a corner in the Wilson line creates a topological excitation at the corner, as shown in (b).

Fracton Codes

- Vertex defects are caused by X-errors. Note that **four** vertex stabilizers are triggered here.
- Cell defects are caused by Z-errors.
- The logical operators correspond to certain classes of excitations traveling along the lattice
  - Lineons, planeons, fractons excitations
- “Remarkably, our example shows that we can decode this model by matching pairs defects of the syndrome on two-dimensional planes of the three-dimensional model in parallel, thus significantly speeding up the process.

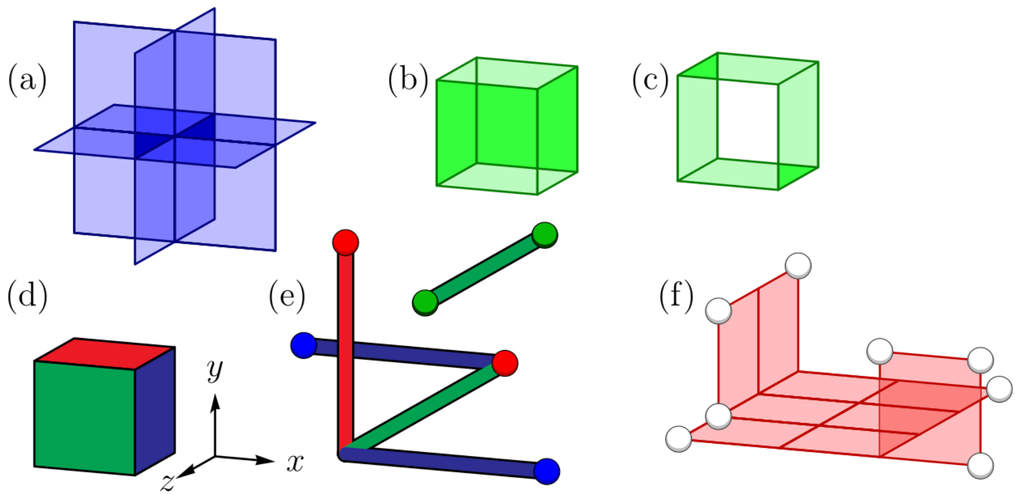


FIG. 1. The support of stabilizer generators A_v, $B_c^{\mathbf{r}}$, and $B_c^{\mathbf{g}}$ are shown in (a), (b), and (c), respectively. The faces of the lattice can be three-colored as in (d). (e) Lineon excitations which are created by applying Pauli-X errors over dual lines. The color of the excitation is determined by the orientation of the line operator that creates the excitation. (f) Fracton excitations of the vertex stabilizers caused by Pauli-Z operators supported on the red faces.

Parallelized quantum error correction with fracton topological codes

Benjamin J. Brown[1] and Dominic J. Williamson[2]
[1]*Centre for Engineered Quantum Systems, School of Physics, University of Sydney, Sydney, New South Wales 2006, Australia*
[2]*Department of Physics, Yale University, New Haven, Connecticut 06520-8120, USA*

Decoding Fracton Codes

- "Our strategy to find neutralizable sets is to pairwise group nearby defects within specifically chosen subregions of the lattice that reflect the materialized symmetries of the model."
- Well-established matching algorithms for the surface code can be adapted to exploit the materialized planar symmetries.

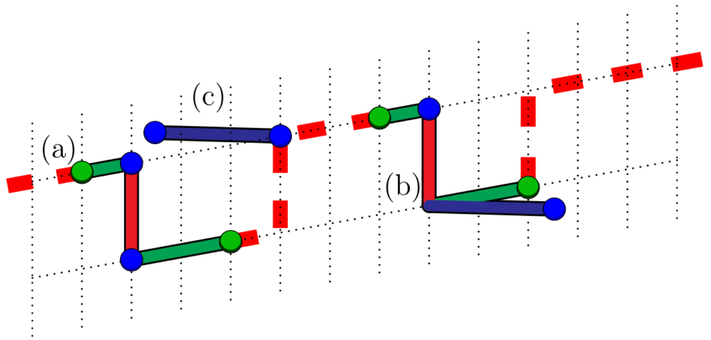


FIG. 2. A Pauli-X error. On the plane marked by the dotted grid we match all the green and red excitations. We use the blue excitations to help find the correct matching. Panel (a) shows an error where two green defects are caused by a stringlike error with blue defects at its corners. Nearby errors can lead to confusion. At (b) an error moves a blue defect away from the plane, and at (c) an additional error introduces a blue defect that could also mislead the decoder into finding an incorrect matching along the red dashed line.

3D Fracton Floquet Codes

- 3D CSS Floquet Code which embeds both an X-Cube model and a Checkerboard model.
- The measurement schedule induces an Floquet code alternating between these two models.
- Not all logical operators of the X-Cube model are permuted correctly to induce an automorphism of the X-Cube model.
- “From the quantum matter perspective, the honeycomb code not only switches between different realizations of Z2 topological order but also exhibits a kind of time crystalline behavior – while the period of the cycling is 3, the period of the code is 6 because after 3 rounds an e/m automorphism occurs.” (Source: Davydova et al.)

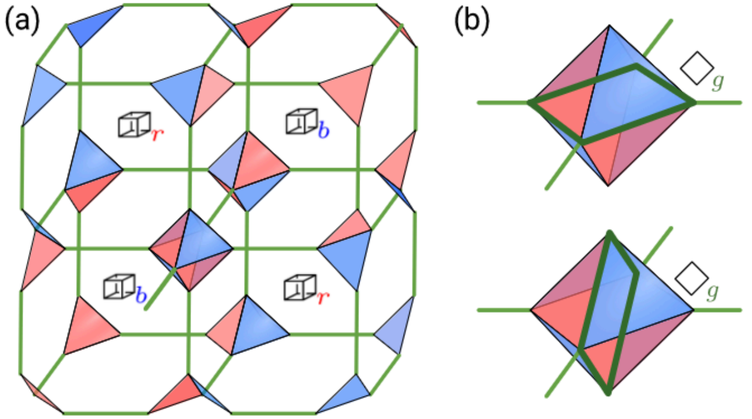


FIG. 3. (a) Decorated cubic lattice with two qubits per edge (located at the vertices of the resulting lattice). The cubes of the two types correspond to $\square_r$ and $\square_b$, respectively, and the triangular plaquettes between the octahedra located at each vertex and the cube of type $b(r)$ are shaded red(blue), respectively, i.e. the complementary color. The two square plaquettes $\diamond_g$ that produce two independent stabilizers are shown in (b).

r	ISG				Syndrome	string			Code
	Measure	m_1	e	m_2		m_1	e	m_2	
-3	$rXXX$								
-2	gZZ	$\square_b(X)$							
-1	$bXXX$	$\square_b(Z)$	$\square_b(X)$						
0	$bZZZ$	$\diamond_g(X)$	$\square_b(Z)$	$\square_b(X)$		gZZ	$rXXX$	$rZZZ$	CB(b)
1	gXX	$\square_r(Z)$	$\diamond_g(X)$	$\square_b(Z)$	$\square_b(X)$	$bXXX$	gZZ	$rXXX$	$\overline{\text{XC}(g)}$
2	$rZZZ$	$\square_r(X)$	$\square_r(Z)$	$\diamond_g(X)$	$\square_b(Z)$	$bZZZ$	$bXXX$	gZZ	CB(r)
3	$rXXX$	$\diamond_g(Z)$	$\square_r(X)$	$\square_r(Z)$	$\diamond_g(X)$	gXX	$bZZZ$	$bXXX$	$\overline{\text{CB}(r)}$
4	gZZ	$\square_b(X)$	$\diamond_g(Z)$	$\square_r(X)$	$\square_r(Z)$	$rZZZ$	gXX	$bZZZ$	XC(g)
5	$bXXX$	$\square_b(Z)$	$\square_b(X)$	$\diamond_g(Z)$	$\square_r(X)$	$rXXX$	$rZZZ$	gXX	$\overline{\text{CB}(b)}$
6	$bZZZ$	$\diamond_g(X)$	$\square_b(Z)$	$\square_b(X)$	$\diamond_g(Z)$	gZZ	$rXXX$	$rZZZ$	CB(b)
7									
8	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Floquet-enriched Topological Order

Measurement-induced Floquet enriched topological order

DinhDuy Vu,[1,2] Ali Lavasani,[2] Jong Yeon Lee,[2] and Matthew P. A. Fisher[3]

[1] *Condensed Matter Theory Center and Joint Quantum Institute, Department of Physics, University of Maryland, College Park, MD 20742, USA*

[2] *Kavli Institute for Theoretical Physics, University of California, Santa Barbara, CA 93106, USA*

[3] *Department of Physics, University of California, Santa Barbara, CA 93106, USA*

The Floquet code utilizes a periodic sequence of two-qubit measurements to realize the topological order. After each measurement round, the instantaneous stabilizer group can be mapped to a honeycomb toric code, thus explaining the topological feature. The code also possesses a time-crystal order distinct from the stationary counterpart – the $e - m$ transmutation after every cycle. In this work, we construct a continuous path interpolating between the Floquet and toric codes, focusing on the transition between the time-crystal and non-time-crystal phases. We show that this transition is characterized by a diverging length scale. We also add single qubit perturbations to the model and obtain a richer two-dimensional parametric phase diagram of the Floquet code, showing the stability of the Floquet topological order.