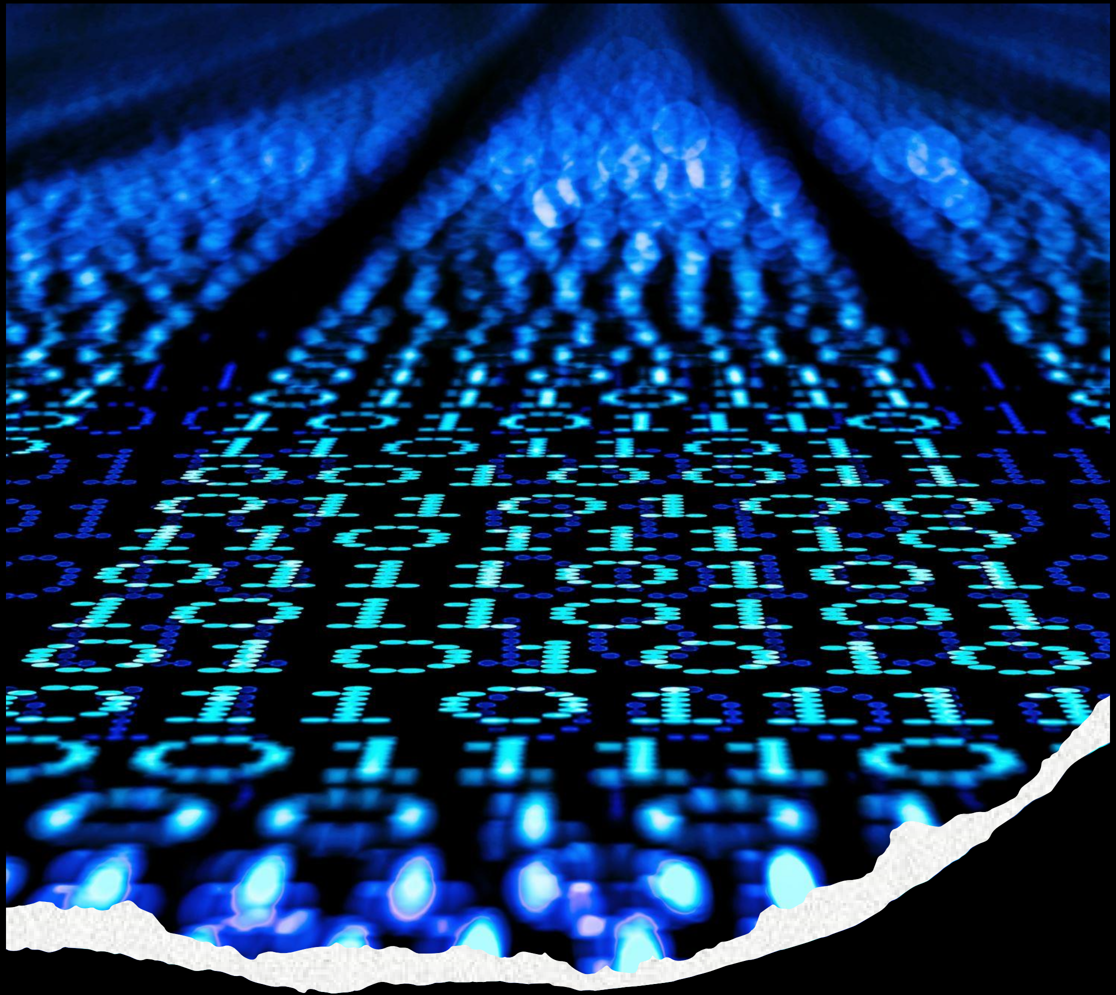
Floquet code as code deformation

Xiaozhen Fu

Joint work with Daniel Gottesman (UMD)

arXiv:2403.04163

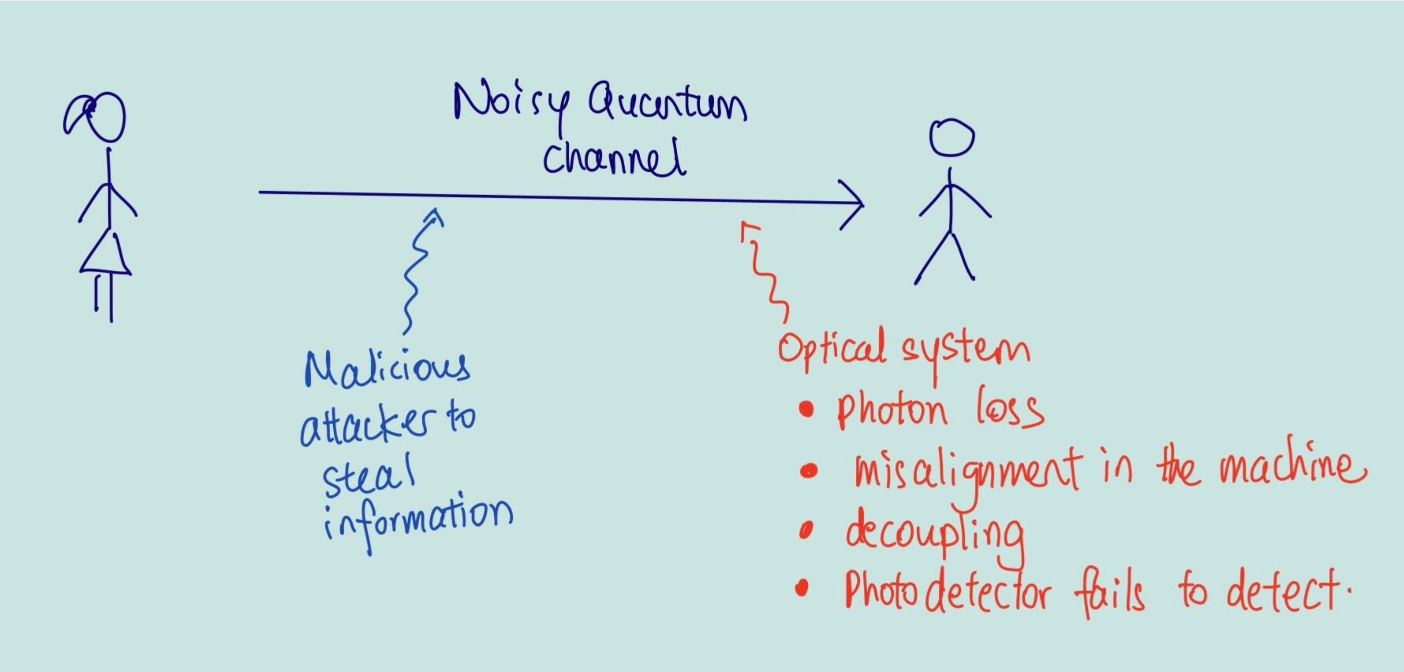


Outline

- **Quantum error correction**
- Motivation
- Our work:
    - Distance of a dynamical code
    - Distance algorithm
    - Applications of distance algorithm

Quantum Error Correction

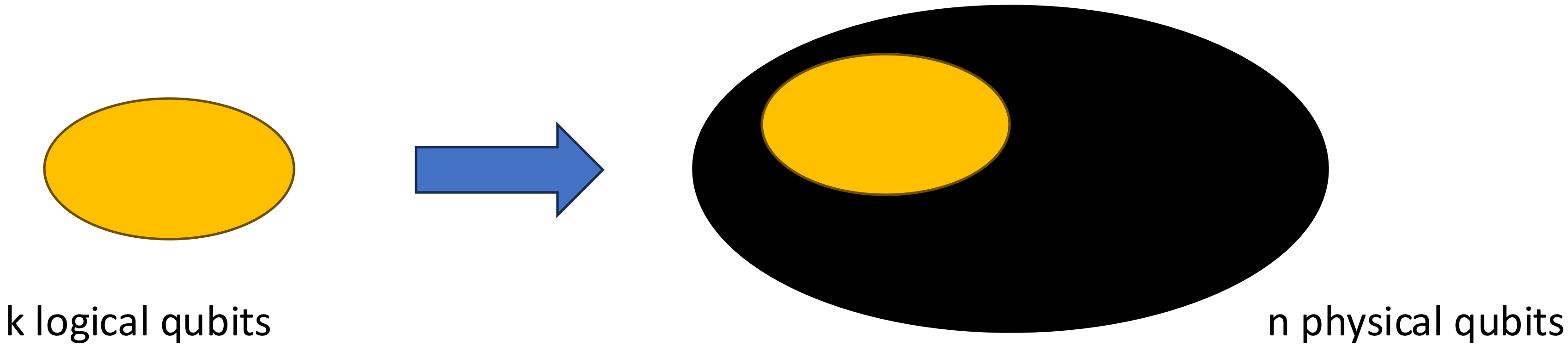
- Imagine Alice wants to send some quantum information to Bob



Gottesman, “Stabilizer Codes and Quantum Error Correction,” arXiv:quant-ph/9705052

Quantum Error Correction

- Encode a logical qubit in a subspace of a bigger Hilbert space.



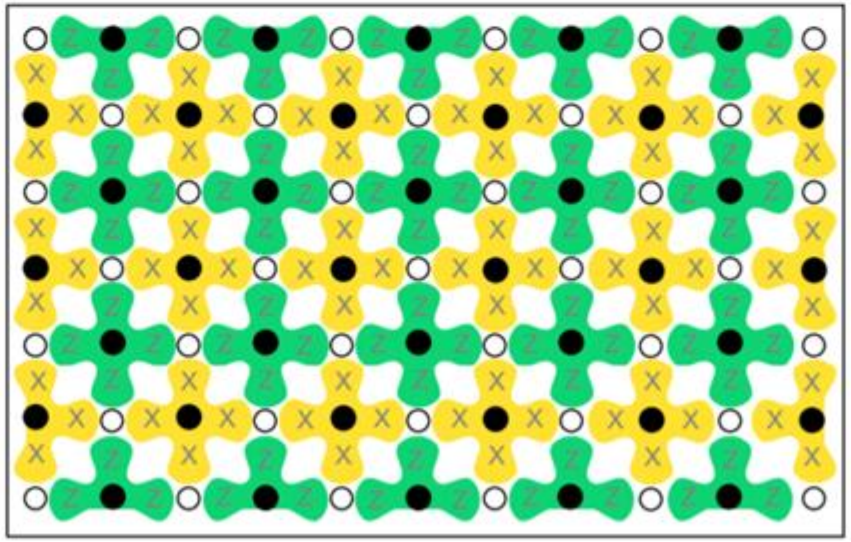
Quantum Error Correction

- Encode a logical qubit in a subspace of a bigger Hilbert space.

Stabilizer Code Formalism
Choose an **Abelian subgroup** of the Pauli group. This will be the stabilizer S of the QECC.
The codewords are the +1 eigenstates of S: $$\{\vert\psi\rangle \text{ s.t. } s \in S, s\vert\psi\rangle = \vert\psi\rangle \ \forall \text{ s } \in S\}$$
If S has r generators on n qubits, number of logical qubits is given by $$k = n - r$$

Error correction procedure

Example: Surface code



Legend

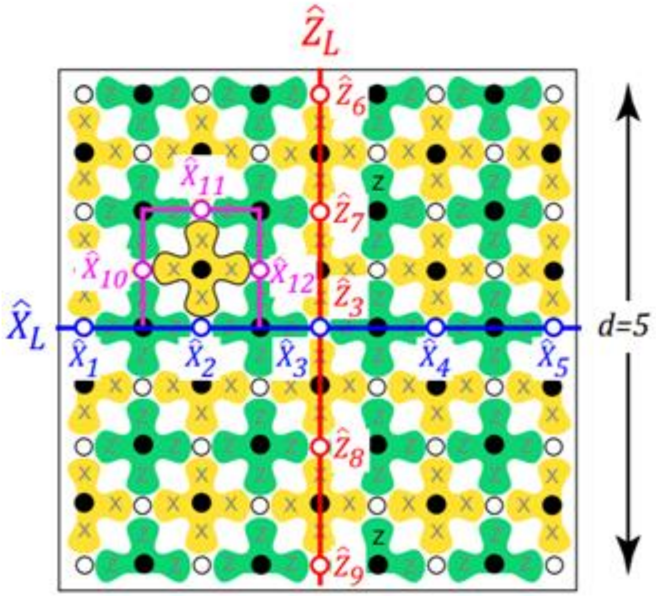
Stabilizer with X Paulis

Stabilizer with Z Paulis

1. Measure the stabilizers to obtain the *syndrome bits*
2. Pass the bits into a decoder
3. The decoder outputs the most likely error that has occurred.
4. Correct the errors on the code

Error correction procedure

Example: Surface code



Legend

Stabilizer with X Paulis

Stabilizer with Z Paulis

1. Measure the stabilizers to obtain the *syndrome bits*
2. Pass the bits into a decoder
3. The decoder outputs the most likely error that has occurred.
4. Correct the errors on the code
5. Deduce the *logical information*

Review of quantum subsystem codes

- Subsystem code can be regarded as stabilizer code but with only a subset of the logical qubits used to store information
- $L = L_{logical} \otimes L_{gauges}$
- Defined by the gauge group G.
- Stabilizers given by $\mathrm{S} = Z(G)$.
- The syndrome information can be obtained through products of low weight gauge operators.
- Distance is given by $N(S)\backslash \mathrm{G}$.

Dynamical code

Typical procedure

1. Pick a stabilizer code
2. Measure the stabilizers of the code to obtain the syndromes

Dynamical code

~~Typical procedure~~

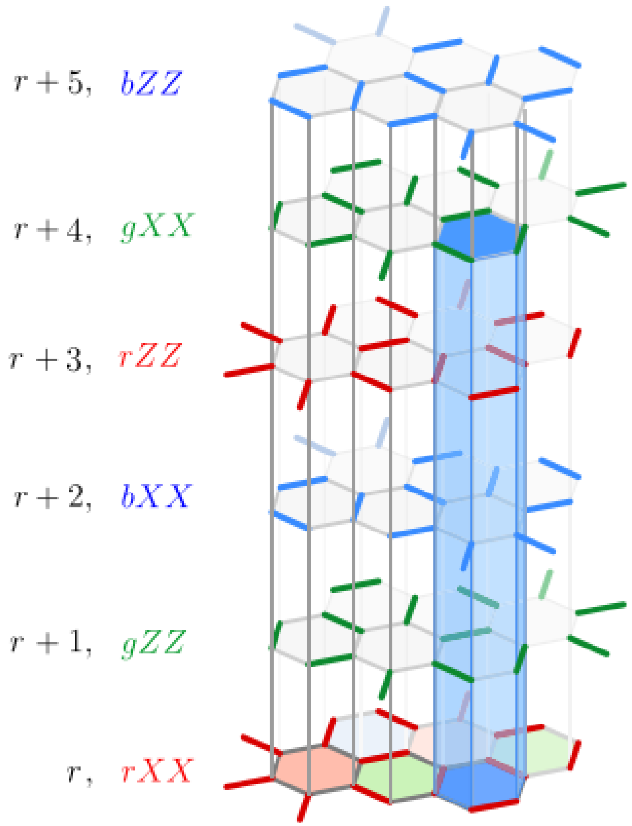
1. Pick a stabilizer code
2. ~~Measure the stabilizers of the code to obtain the syndromes~~
2. Pick a sequence of measurements

Outline

- Quantum error correction
- **Motivation**
- Our work:
  - Distance of a dynamical code
  - Distance algorithm
  - Applications of distance algorithm

Motivation 1: Honeycomb code

- Each round of measurements anti-commute with the previous round.
- The outcome of the measurements are random but the product of 6 checks give the syndromes of one plaquette stabilizer.
- Stabilizer groups and the logical operators evolve with time.



Hastings and Haah, “Dynamically Generated Logical Qubits,” Quantum 5, 564 (2021), arXiv:2107.02194
M. Davydova, N. Tantivasadakarn, and S. Balasubramanian, PRX Quantum 4, 020341 (2023)

Motivation 2: Circuit – Code equivalence

- **Construction of sparse codes**

  Bacon, Flammia, Harrow, Shi (2014)
  arXiv:1411.3334

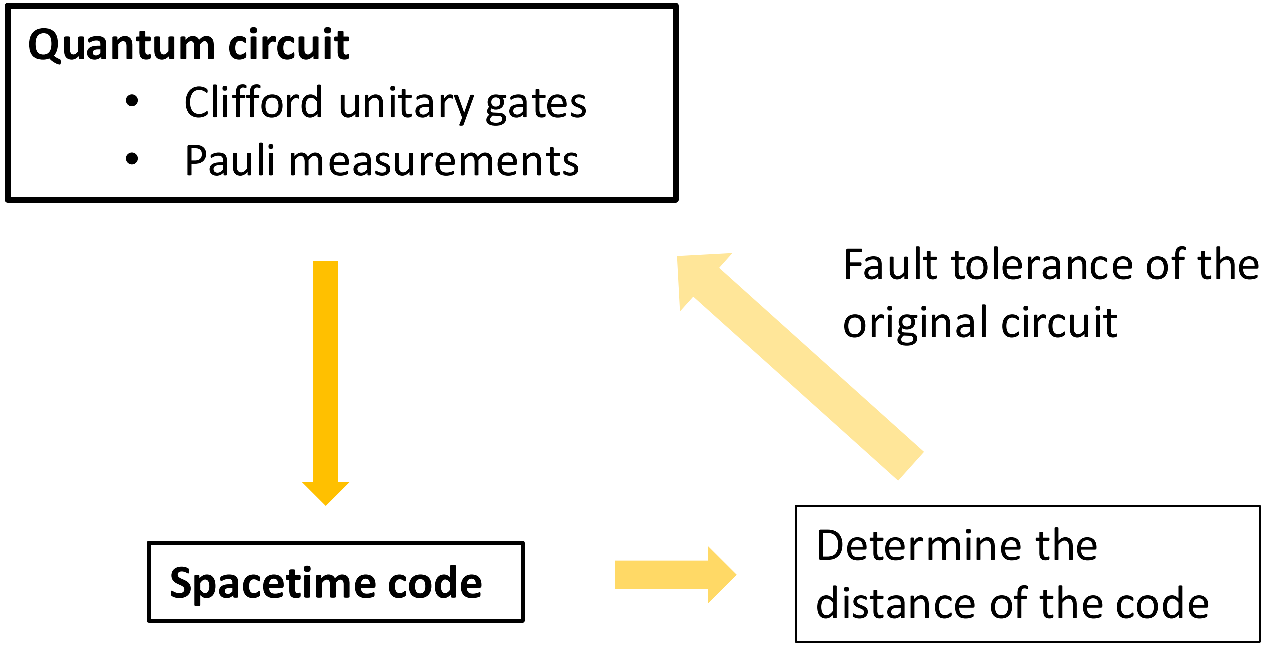
- **Spacetime code from circuits**

  Delfosse, Paetznick (2023) arXiv:2304.05943

  Gottesman (2022) arXiv:2210.15844

- **Classical code and circuit equivalence**

  Li (2024) arXiv:2403.10268



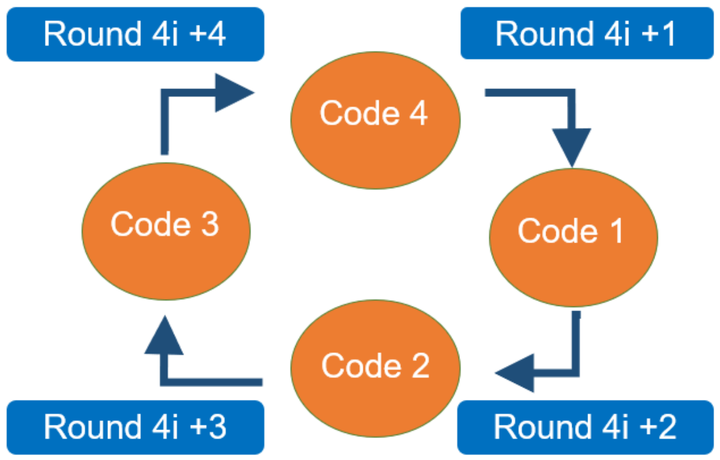
-Can we build fault tolerant circuits from spacetime codes?
-Can we create more general dynamical codes?
-Can we get better code parameters with dynamical codes?

Outline

- Quantum error correction
- Motivation
- Our work:
    - **Distance of a dynamical code**
    - Distance algorithm
    - Applications of distance algorithm

Set-up: Dynamical Codes

- Codes that are defined by a sequence of measurements.
- Start with a stabilizer code, with stabilizer group S_1
- Perform a series of measurements, with each round of measurements consisting of commuting Pauli operators
- Codespace evolves with measurements:
  - After the i^{th} round:
    $\mathcal{C}_i = \{|\psi\rangle = \pm s \cdot |\psi\rangle, \forall s \in S_i\}$
- Example: Floquet codes
- **Q: Error correction property?**



Distance

- Distance d_{ISG} is given by the minimum of the distances of all ISGs.
- Distance $d_{\mathrm{subsystem}}$ is given by the minimum of the distances of all subsystem codes formed by neighboring ISGs.
  - $d_{\mathrm{subsystem}} = \min \mathrm{wt}\{L_{\mathrm{dressed}} \backslash \mathrm{G}\}$
- This work: Distance d_{u} is given by the minimum of the unmasked distances of all ISGs.

$$d_{\mathrm{u}} \leq d_{\mathrm{subsystem}} \leq d_{\mathrm{ISG}}.$$

Classification of Stabilizers in S_1

Example: $S_1 = \{x_1x_2x_3, z_2z_4, x_1x_3x_5\}$

Measurement sequence $\{x_1x_2, x_3, x_1x_4\}$

Def: An **unmasked stabilizer** is a stabilizer whose outcome can be obtained through measurements in subsequent rounds.

- $x_1x_2x_3$ is an unmasked stabilizer.

Def: A **permanently masked stabilizer** is a stabilizer whose outcome can never be obtained through the sequence of measurements or future measurements.

- z_2z_4 is permanently masked.

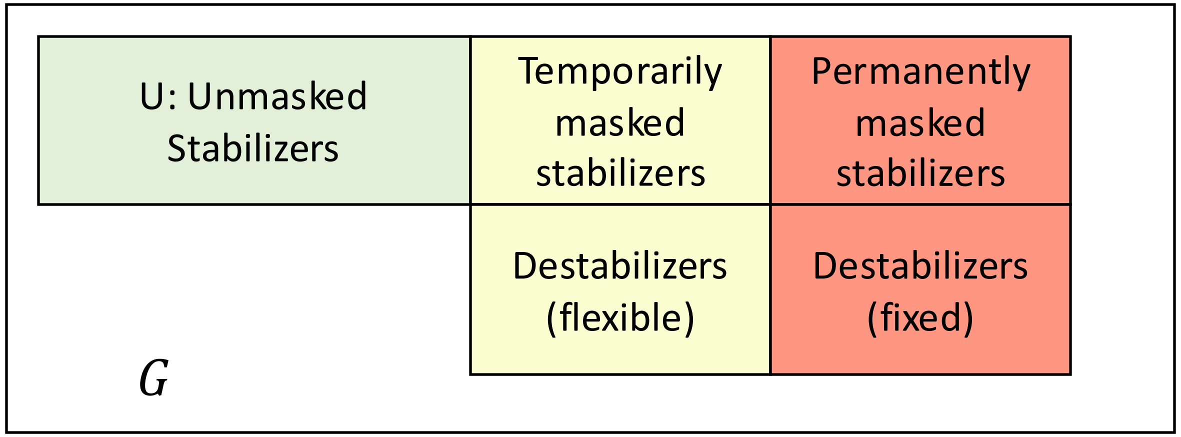
Def: A temporarily masked stabilizer is a stabilizer whose syndrome cannot be obtained given a sequence of measurements, but can potentially be obtained by creating new measurements.

- $x_1x_3x_5$ is temporarily masked.

Theorem: Distance of a Dynamical code

Theorem: For a masked stabilizer code, its unmasked distance is given by:

$$d_u = \min \mathrm{wt}\{N(U) \backslash G\}$$



*Where G is the gauge group that depends partly on the freedom of choice of the destabilizers for the temporarily masked stabilizers, and partly on the measurement sequence that fixes the destabilizers of the permanently masked stabilizers.

Intuition of theorem

When there is masking, the set of correctable errors shrinks.

Errors that have the same unmasked syndrome, but different masked syndromes are indistinguishable.

Example: [[9,1,3]] Shor code.

Logical operators:

$$Z_L = x_1 x_2 x_3, X_L = z_1 z_4 z_7$$

Stabilizers: $z_1 z_2, z_2 z_3, z_4 z_5, z_5 z_6, z_7 z_8, z_8 z_9; x_1 x_2 x_3 x_4 x_5 x_6, x_4 x_5 x_6 x_7 x_8 x_9$

If we mask $z_2 z_3$, then error x_3 and trivial error are now indistinguishable.

However, we can choose specific *logical representatives* to avoid undetectable errors.

Example of masking

Example: [[9,1,3]] shor code.

Logical operators:

$$Z_L = x_1 x_2 x_3, X_L = z_1 z_4 z_7$$

Stabilizers: $z_1 z_2, z_2 z_3, z_4 z_5, z_5 z_6, z_7 z_8, z_8 z_9; x_1 x_2 x_3 x_4 x_5 x_6, x_4 x_5 x_6 x_7 x_8 x_9$

$Z_L = x_1 x_2 x_3$
$X_L = z_1 z_4 z_7$
$\mathcal{G} = \mathcal{S} \cup \{x_3\}$
Distance = 2

$Z_L = x_1 x_2 x_3$
$X_L = z_3 z_4 z_7$
$\mathcal{G} = \mathcal{S} \cup \{x_1 x_2\}$
Distance = 1

Choice of destabilizers fixes the distance of the code.

Outline

- Quantum error correction
- Motivation
- Our work:
  - Distance of a dynamical code
  - **Distance algorithm**
  - Applications of distance algorithm

Finding the unmasked and masked stabilizers.

Can be tricky. Why?

Examples:

High weight stabilizer $\{x_1 x_2 x_3 x_4 x_5 x_6\}$,

- Measure $\{x_1 x_2, x_3 x_4, x_5 x_6\}$ in the order they appear.
  - Yes
- If we measure $\{x_1 x_2, z_2 z_3, x_3 x_4, x_5\ x_6\}$.
  - No
- we change the sequence order to this: $\{x_1 x_2, x_3 x_4, z_2 z_3, x_5 x_6\}$?
  - Yes
- What about $\{x_5 x_6, z_6 z_7, x_1 x_2, x_3 x_4\}$?
  - Yes

Main result

- Distance Algorithm
  - Classify the stabilizers according to their masking properties.
  - Keep track of the relevant syndromes and measurement outcomes
  - Needed to calculate the unmasked distance of the code.
- Optimized
  - All syndromes can be obtained through the algorithm.
  - Classically efficiently, $O(n^3)$.

Input:
- A stabilizer code
- Measurement sequence

Output:
- Unmasked, temporarily masked and permanently masked stabilizers
- Destabilizers
- Syndrome information

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

Initialization: $\mathbf{C} = \boldsymbol{S_1}, \mathbf{V} = \{\}$

Measurement	C			V
	$XYZII$	$IXXII$	$IIIXX$	

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

Case 1: m commutes with C and V
Update: Add m to V

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

Case 2: m commutes with only C
Update V only.

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
	↓	↓	↓	↓
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

Case 3: m commutes with only V
Update both C and V.

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
	↓	↓	↓	↓
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$
	↓	↓	↓	↓
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$YZZII$

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

PMS:
$IIIXX$

Case 3: m commutes with only V
Update both C and V.

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$
$IXXZI$	$XYZII$	$IXXII$	$\cancel{IIIXX}$	$YZZII, IXXZI$

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

PMS:
$IIIXX$

Case 4: m commutes with neither
Update both C and V

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$
$IXXZI$	$XYZII$	$IXXII$	$\cancel{IIIXX}$	$YZZII, IXXZI$
$ZIIXX$	$XYZII$	$IXXII$		$YZZII, IXXZI$

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

PMS:
$IIIXX$

Case 4: m commutes with neither
Update both C and V

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$
$IXXZI$	$XYZII$	$IXXII$	$\cancel{IIIXX}$	$YZZII, IXXZI$
$ZIIXX$	$XYZII \rightarrow ZXIII$	$IXXII$		$\cancel{YZZII}, IXXZI \rightarrow YYYZI, ZIIXX$

Illustration of the algorithm

Input: $S_1 = \{XYZII, IXXII, IIIXX\}$

Measurements: $\{XXXXX, YZZII, IXXZI, ZIIXX, IXIXX\}$

PMS:
$IIIXX$

Case 1: m commutes both
Update V

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$
$IXXZI$	$XYZII$	$IXXII$	$\cancel{IIIXX}$	$YZZII, IXXZI$
$ZIIXX$	$XYZII \rightarrow ZXIII$	$IXXII$		$\cancel{YZZII}, IXXZI \rightarrow YYYZI, ZIIXX$
$IXIXX$	$ZXIII$	$IXXII$		$YYYZI, ZIIXX, IXIXX$

Illustration of the algorithm

PMS: $IIIXX$

Measurement	C			V
$XXXXX$	$XYZII$	$IXXII$	$IIIXX$	$XXXXX$
$YZZII$	$XYZII$	$IXXII$	$IIIXX$	$\cancel{XXXXX}, YZZII$
$IXXZI$	$XYZII$	$IXXII$	$\cancel{IIIXX}$	$YZZII, IXXZI$
$ZIIXX$	$XYZII \rightarrow ZXIII$	$IXXII$		$\cancel{YZZII}, IXXZI \rightarrow YYYZI, ZIIXX$
$IXIXX$	$ZXIII$	$IXXII$		$YYYZI, ZIIXX, IXIXX$

Compute $\langle C \rangle \cap \langle V \rangle$ to find unmasked stabilizers:

$ZIIXX \cdot IXIXX = ZXIII \in C$

$\Rightarrow XYZII$ *is unmasked*

$\Rightarrow IXXII$ *is temporarily masked*

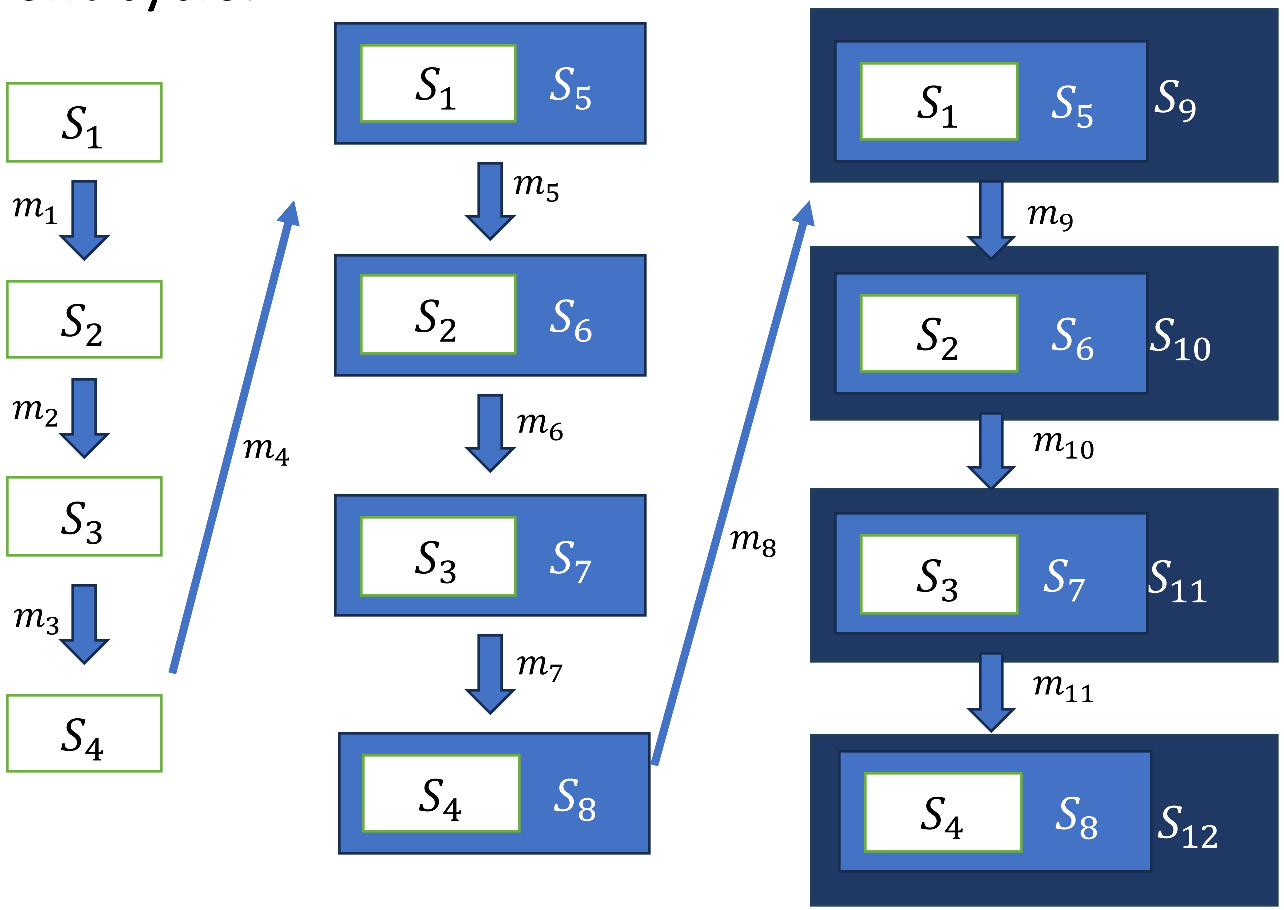
And $IIIXX$ *is permanently masked*

Outline

- Quantum error correction
- Motivation
- Our work:
  - Distance of a dynamical code
  - Distance algorithm
  - **Applications of distance algorithm**

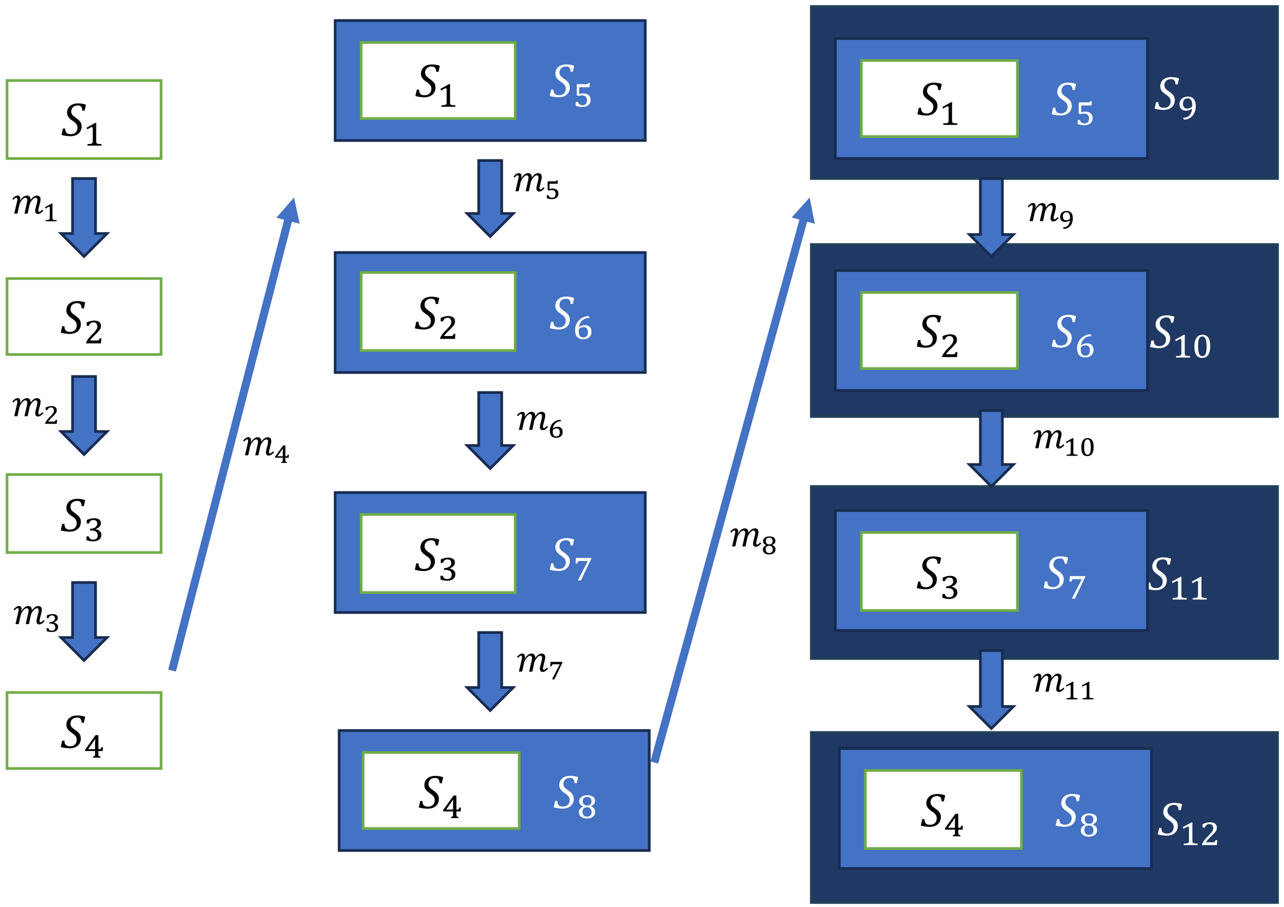
Structure of Floquet code

The instantaneous stabilizer generators after measuring m_i in the previous cycle is a subset of the instantaneous stabilizer generators after measuring m_i in the current cycle.



Structure of Floquet code

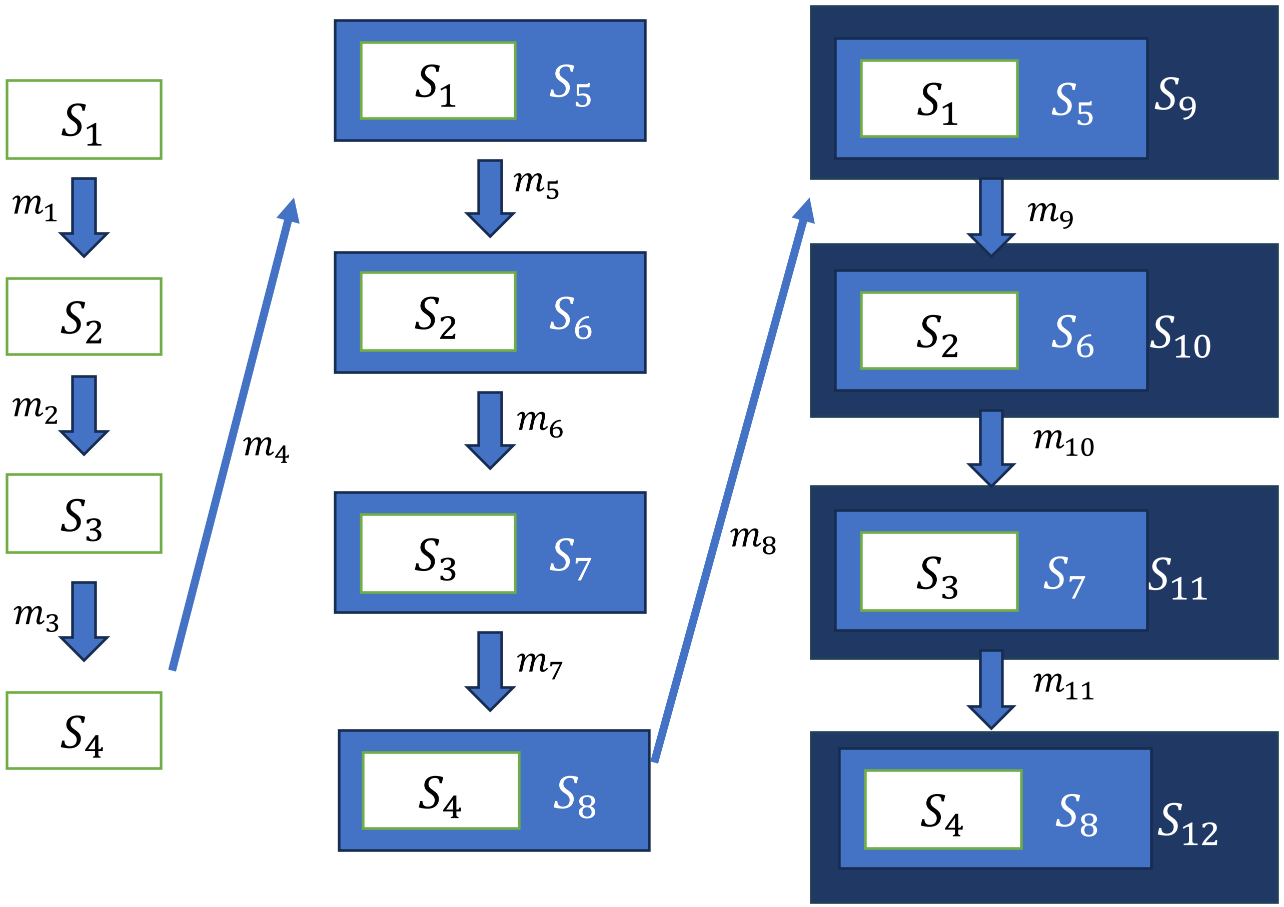
If there are k new generators added from S_j to S_{j+L}, then there can be at most k more new generators added in the next cycle from S_{j+L} to S_{j+2L}.



There exists a measurement sequence for a Floquet code that takes $n - 1$ cycles to fully initialize for a code with n stabilizer generators.

Structure of Floquet code

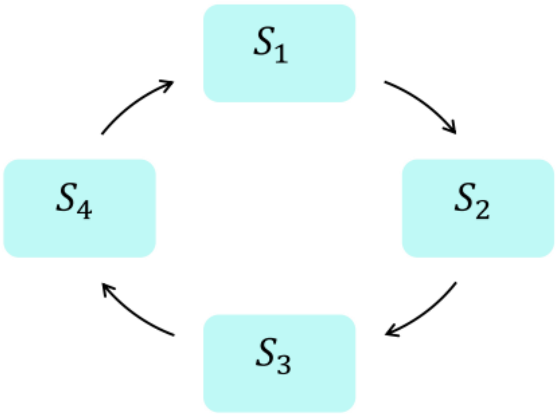
It takes at most k cycles to unmask all the stabilizers from an ISG, with k given by the number of cycles required to initialize a Floquet code.



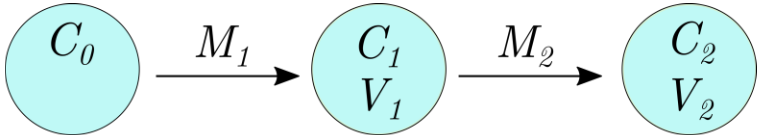
For a Floquet code with $k = O(1)$ cycles to initialize and $m = O(1)$ measurements per cycle, then to fully determine the unmasked stabilizers for an ISG, it takes at most $m \cdot k = O(1)$ measurements.

Summary and outlook

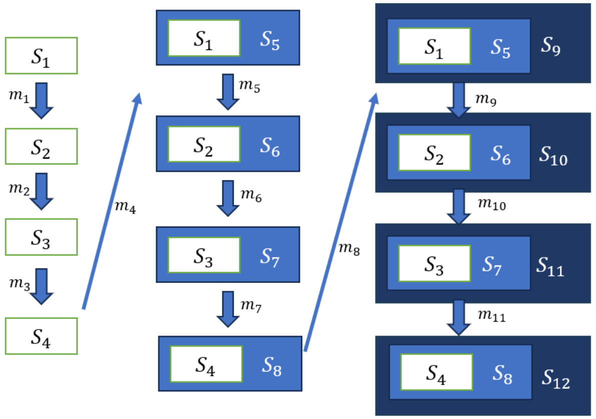
arXiv:2403.04163



Unmasked distance of dynamical codes



Distance algorithm for classifying stabilizers



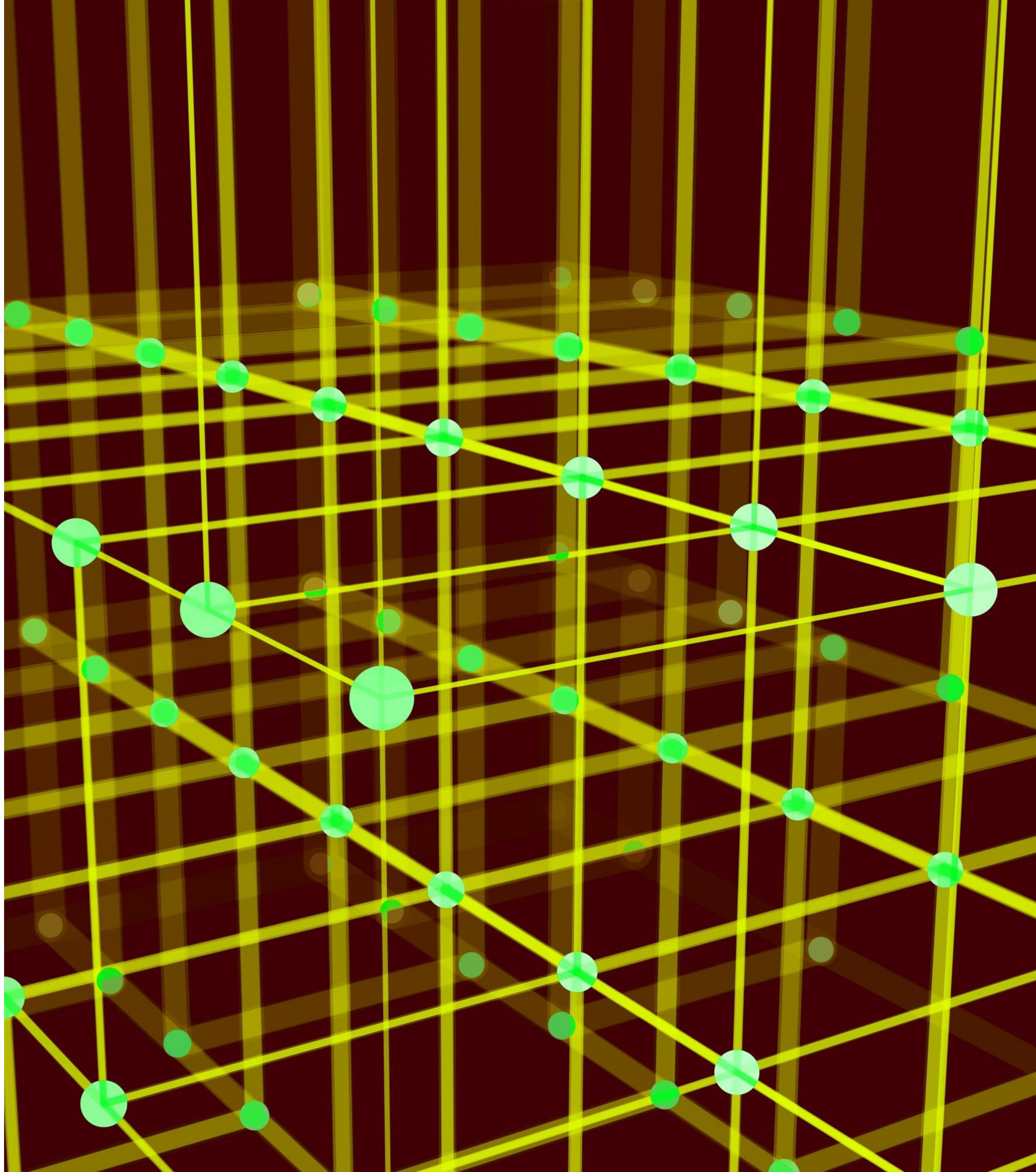
Interesting Floquet code structures

Many future directions:

1. Understanding circuit-code equivalence
2. Decoding for space-time code and Floquet code
3. No-go theorems specific to dynamical codes
4. Inventing interesting Floquet codes

Thank you!

DETECTORS



Detectors

- A detector is a parity constraint on the measurement outcomes. In the absence of errors, the sum of the measurement outcomes should be 0.
- It is also required that there exist a set of generators of detectors such that for each parity constraint, we have that each term is a direct measurement (as opposed a product of different measurement outcomes).

Detectors

In the previous example, O(ZXIII) is not directly measured but is found as the outcome of other measurements. The following set of relations is not a valid set of generators of detectors.

$$O(XYZII) = O(ZXIII) + O(m1)$$

$$O(ZXIII) = O(m3) + O(m4)$$

- The valid detector cell in the example above is:

$$O(XYZII) = O(m1) + O(m3) + O(m4)$$

Relation between detectors and unmasked stabilizers

- Each detector cell consists of several layers of unmasked stabilizers at different time steps.
- O(XYZII) = O(m1) + O(m3) + O(m4)

t = 0	$XYZII$
t = 1	$XYZII \cdot m1$
t =2	$XYZII \cdot m1$
t =3	$XYZII \cdot m1 \cdot m3 = m4$
t = 4	

- Syndrome of an error E is given by the commutation relation between the detector cell and the spacetime error E.

Syndromes of a Dynamical code

- The syndromes for a dynamical code is obtained from a set of linearly independent detector cells. Given a fault configuration in spacetime, we have a corresponding syndrome.

  The syndrome for a measurement error can be computed by checking which detector is the measurement involved in. These detectors will acquire a sign flip from the measurement error, giving non-trivial syndromes for that measurement error.

Logical error

- A logical operator is dynamically updated by the sequence of measurements. A logical representative is updated by measurements, and its logical outcome is also updated by the measurement outcomes.
- A logical error on the logical representative can occur if there is a measurement error but the measurement is used to update the logical. The logical outcome will be updated wrongly.
  - For example, a code is initialized in the +1 eigenstate of the logical operator $X_1X_2X_3X_4$.
  - Suppose we measure Z_2Z_3 but obtained a measurement error of -1 instead of +1.
  - In the next time step, when we measure Y_3, the logical representative is updated to $X_1Y_2Y_3X_4$ which should have an outcome of +1, but we will think that it should have the opposite sign.
- It is important to be able to identify the spacetime location of the measurement error in order to prevent a single error from creating a logical error in a dynamical code.