Subsystem codes with high thresholds by gauge fixing and reduced qubit overhead

Oscar Higgott
University College London

Joint work with Nikolas P. Breuckmann
arXiv: 2010.09626

Background and Motivation

- Quantum error correcting codes (QECs) will be required to protect large-scale quantum computers from noise
- QECs correct errors by introducing *redundancy*
- Goal: use as little redundancy as possible to correct as many errors as possible

The Surface Code

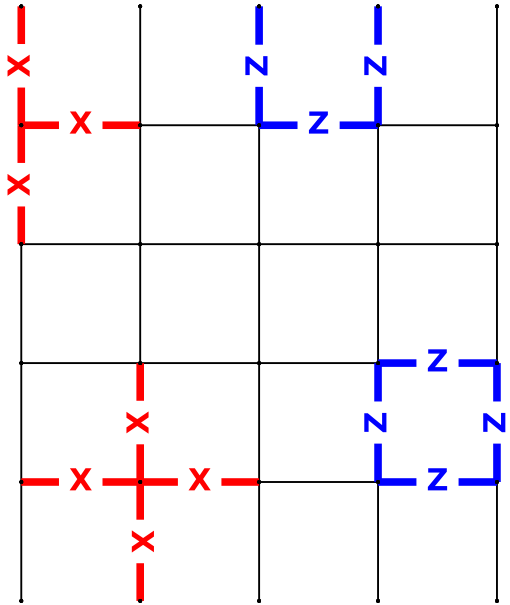
The surface code is the focus of efforts to build a fault-tolerant quantum computer

Advantages:

- Four-qubit measurements
- Good threshold (~1%)

Challenges:

- Huge overhead (~1000x)
- Even weight-four checks can be too large (crosstalk)
- Hardware currently above threshold



Dennis, et al. "Topological quantum memory." *Journal of Mathematical Physics* 43.9 (2002): 4452-4505.
Chamberland, et al. "Topological and subsystem codes on low-degree graphs with flag qubits." *Physical Review X* 10.1 (2020): 011022.

Prior work reducing qubit overhead

- Promising Low-Density Parity Check (LDPC) code families:
    - 2D and 4D Hyperbolic codes
    - Hypergraph (and homological) product codes
    - Fibre-bundle, lifted product and balanced product codes
- But…
    - All have high weight (>4) checks, and not clear how to schedule them efficiently (gate errors can eliminate the advantage)
    - As a result, *no code has previously been shown to outperform the surface code under circuit-level depolarising noise*

This work

1. Reduce check weight *and* qubit overhead relative to the surface code:
    - Finite-rate quantum LDPC codes with three-qubit check operators
    - Outperform the surface code under circuit-level depolarising noise
2. Improve thresholds, especially under biased noise:
    - New technique for decoding subsystem codes
    - Use gauge fixing *in software* to obtain more useful information

Stabilizer Codes

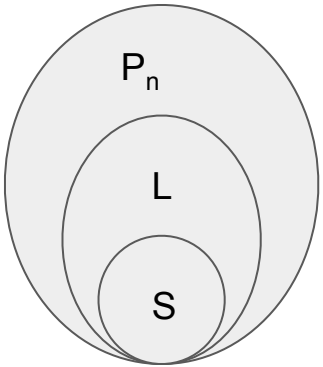
$$\mathcal{S} \subset \mathcal{P}_n$$

Code space is +1-eigenspace of elements of the *stabiliser group*:

$$\mathcal{C} = \{|\psi\rangle \mid \forall s \in \mathcal{S} : s|\psi\rangle = |\psi\rangle\}$$

Logical group is the *centraliser* of the stabiliser group:

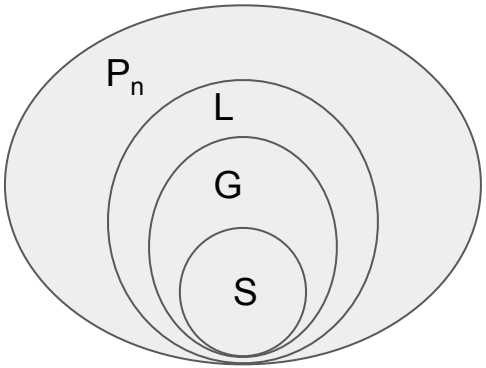
$$\mathcal{L} = C(\mathcal{S})$$



Subsystem Codes

Subsystem codes defined by *gauge group*: $\mathcal{G}$

Stabiliser group is the *center* of the gauge group: $\mathcal{S} = Z(\mathcal{G}) = C(\mathcal{G}) \cap \mathcal{G}$



Essentially a stabiliser code where we choose not to store quantum information in a subset of the logical qubits (the gauge qubits)

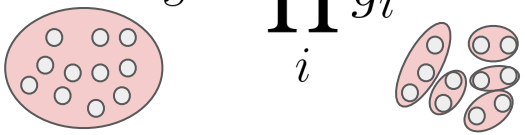
Poulin, “Stabilizer formalism for operator quantum error correction,” PRL 95, 230504 (2005)

Subsystem Codes: why?

Gauge qubits can help with syndrome measurement

(Some) Subsystem Codes:
Stabilizers are products of
low weight gauge operators

$$s = \prod_i g_i$$



high weight

low weight

Measure g_i *individually* and multiply the results to obtain measurement of s

Important facts:

1. Gauge operators generally do not commute: *their outcome is randomized*

$$+1 = (+1)(-1)(+1)(-1)$$
$$+1 = (-1)(+1)(+1)(-1)$$

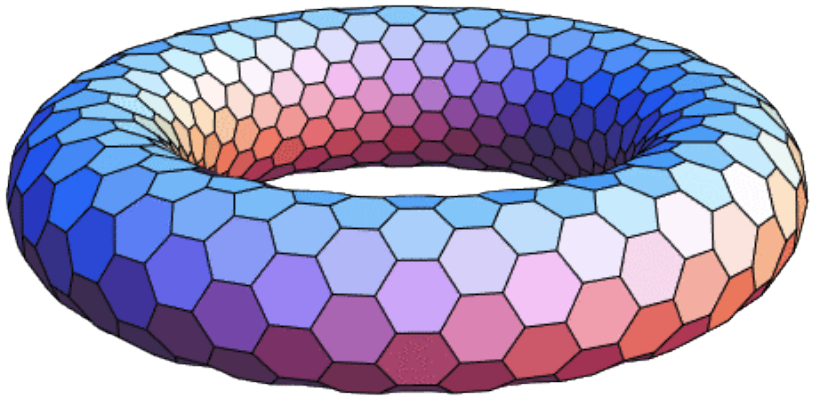
2. *High-weight stabilizers provide poor performance*:
High-weight: less information
& more measurement errors

$$s = g_1\ g_2\ g_3\ g_4$$

Subsystem Code Construction

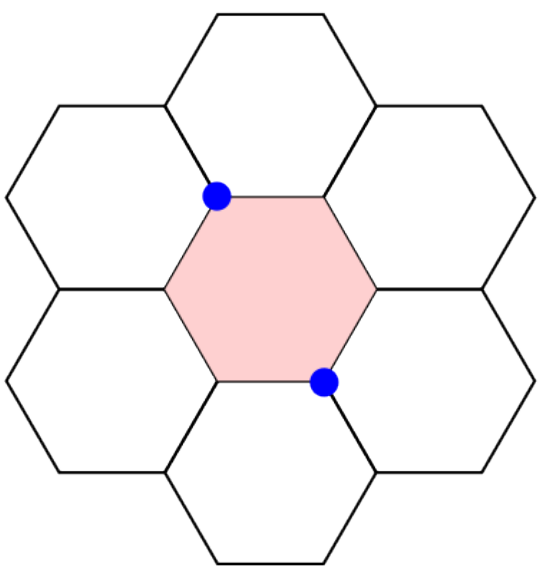
LDPC Subsystem Construction

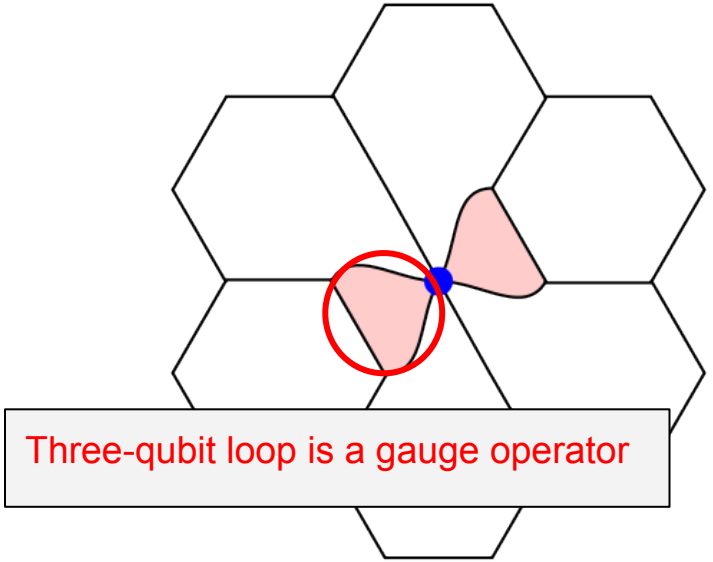
Example: hexagonal toric code



Stabilizer check weights are 3 and 6

LDPC Subsystem Construction: **Merging**



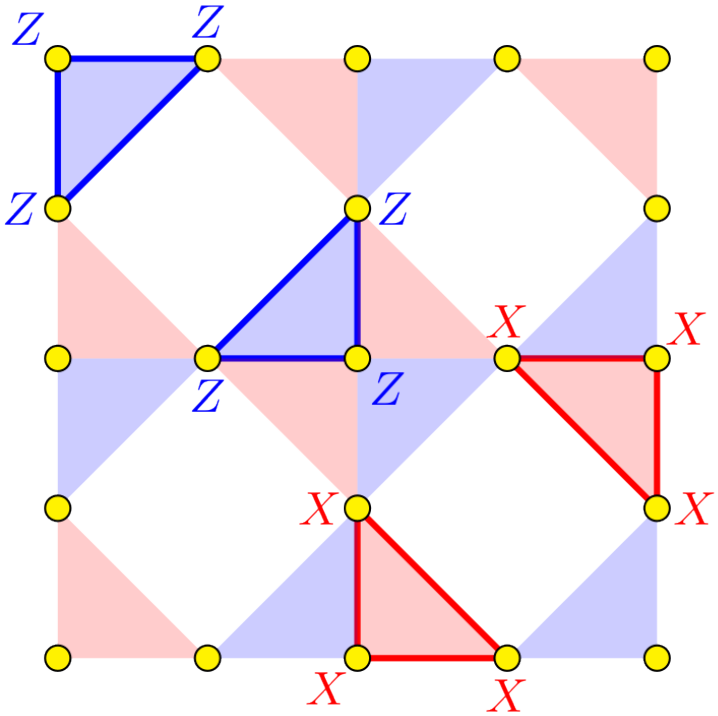


All stabilizers still commute!

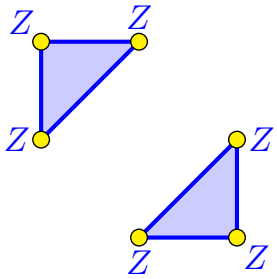
encoded qubits = # physical qubits - # independent checks

Subsystem Surface Code

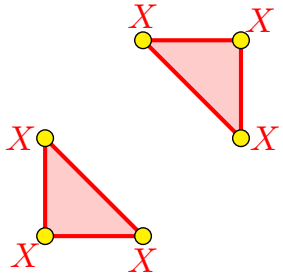
Qubits on vertices



Z stabilisers

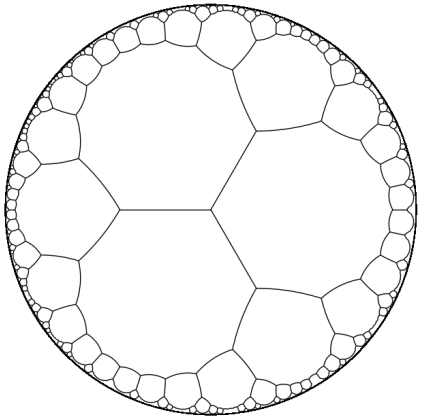


X stabilisers



Bravyi, Duclos-Cianci, Poulin, Suchara, “Subsystem surface codes with three-qubit check operators,” QIC 13, 963–985 (2013)

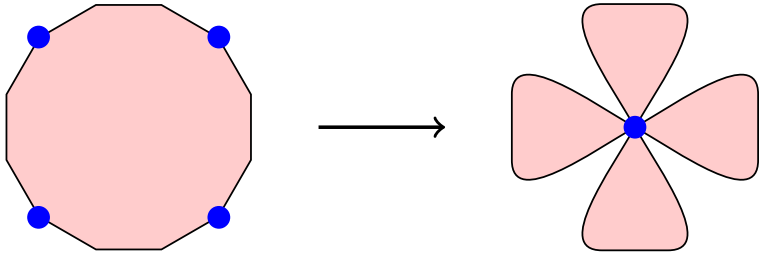
LDPC Subsystem Construction



Close cousin of toric code:
Hyperbolic Codes

$$k = \left(1 - \frac{2}{r} - \frac{2}{s}\right) n + 2$$

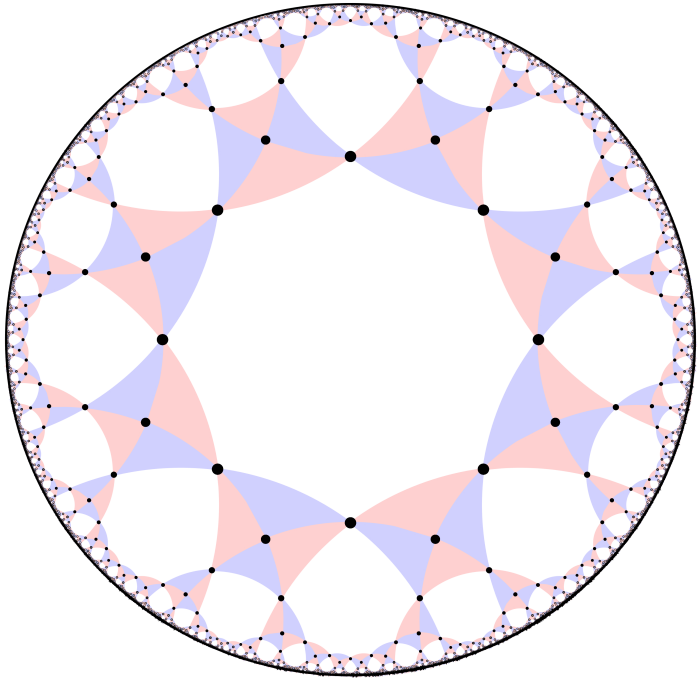
Merge four vertices per face

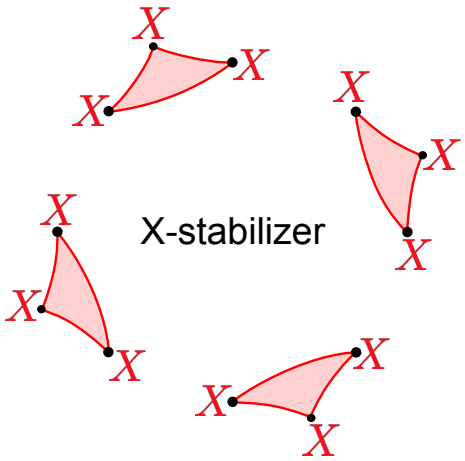


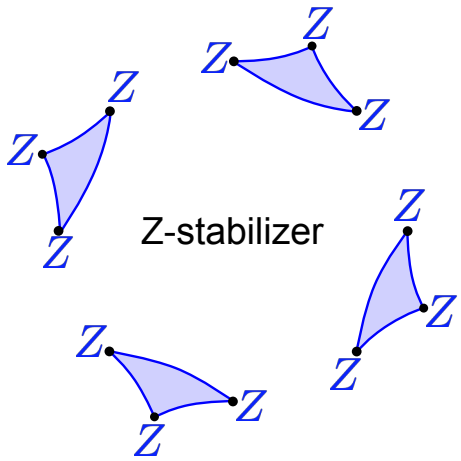
Obtain code with k = n/6 and weight-3 checks

LDPC Subsystem Construction

- stabilizer-weight 12 & gauge-weight 3
- all check operators are weight 3
- improved encoding rate

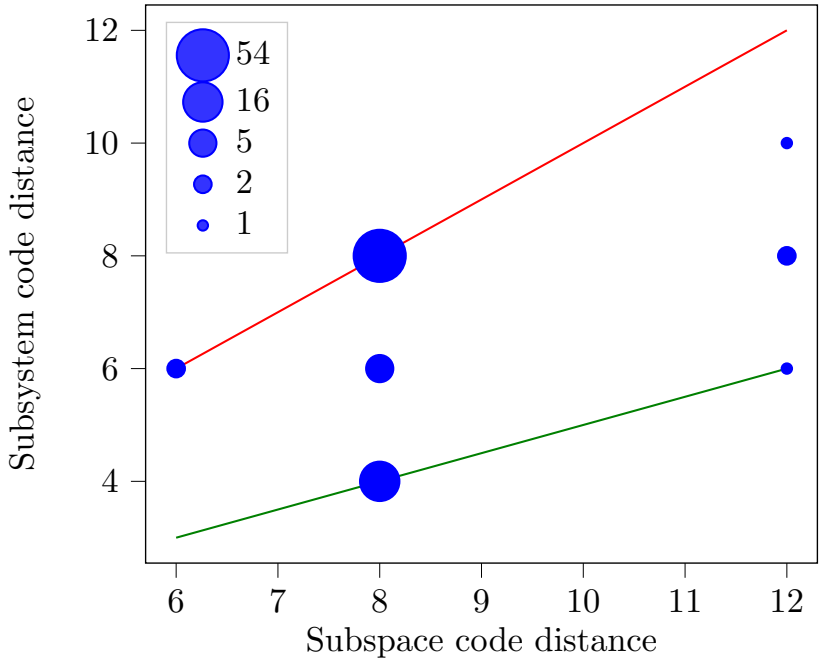






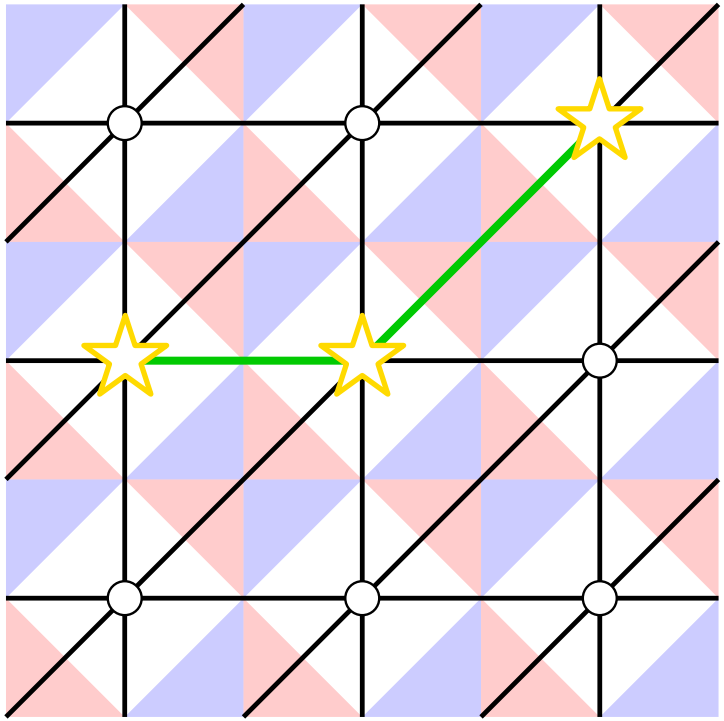
Distance of subsystem hyperbolic codes

- Multiplying logicals by gauge operators can decrease distance
- Dressed distance of subsystem hyperbolic code at worst half that of stabiliser code derived from the same tessellation



Decoding subsystem toric and hyperbolic codes

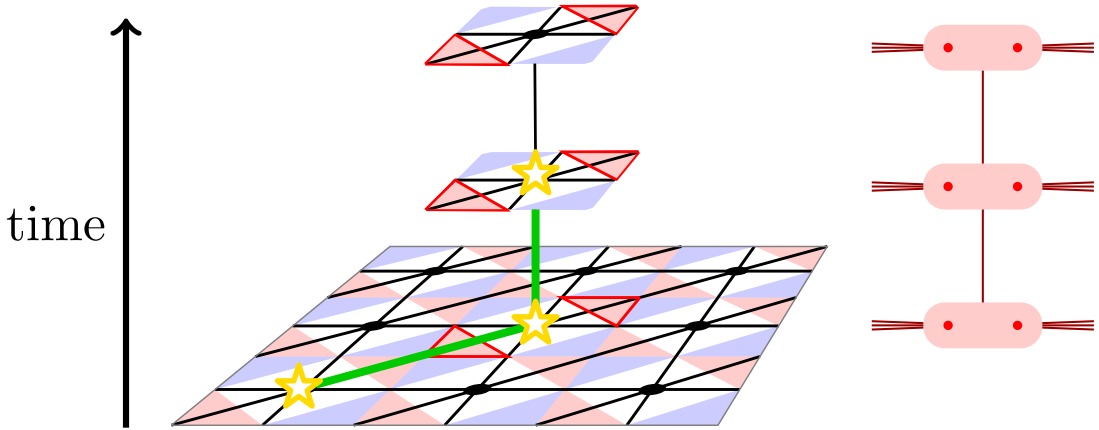
- Each qubit is incident to two X-type stabilisers
- Can construct a matching graph *V* with:
  - A vertex for each stabiliser
  - An edge for each qubit
- Given a syndrome (set of defects), we decode by finding the minimum-weight perfect matching in *V*



Bravyi, Duclos-Cianci, Poulin, Suchara, “Subsystem surface codes with three-qubit check operators,” QIC 13, 963–985 (2013)

Decoding with noisy syndrome measurements

- Repeat measurements O(L) times
- Use the *difference syndrome (*parity of consecutive measurements)
- Measurement errors are now time-like edges
- Find the minimum-weight matching in the 3D matching graph
- Code at: https://github.com/oscarhiggott/PyMatching



Summary

- Low check weight makes measurements much simpler to facilitate
- Hyperbolic tessellations allow for a better encoding rate

But: Gauge measurements have to be combined to a higher-weight stabilizer

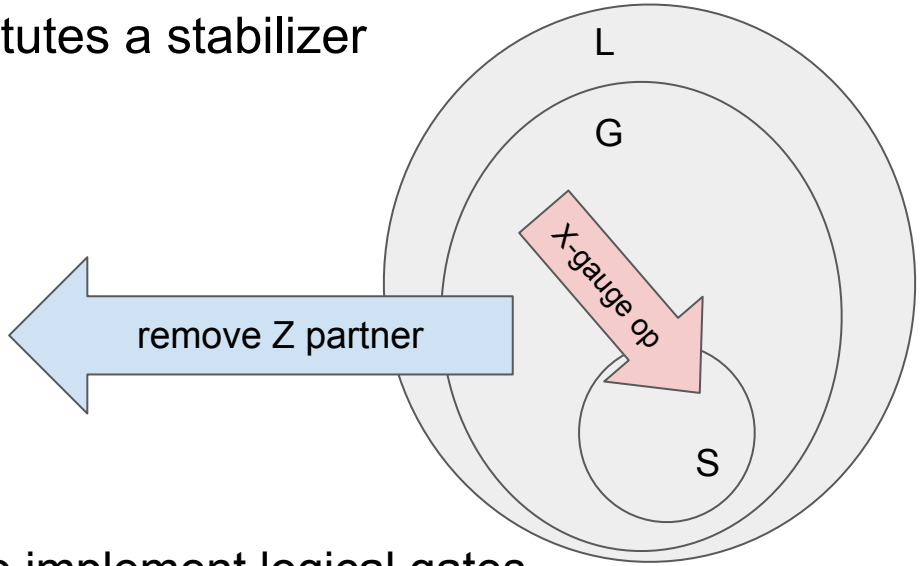
$\Rightarrow$ Do not expect improvement in threshold

Schedule-Induced Gauge Fixing

Gauge-Fixing

Add commutative subset of gauge operators to stabilizer group

‘Change our mind’ about what constitutes a stabilizer

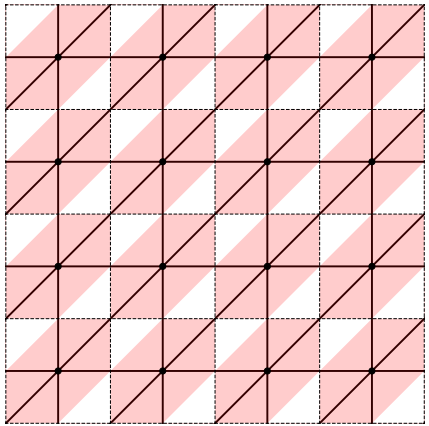


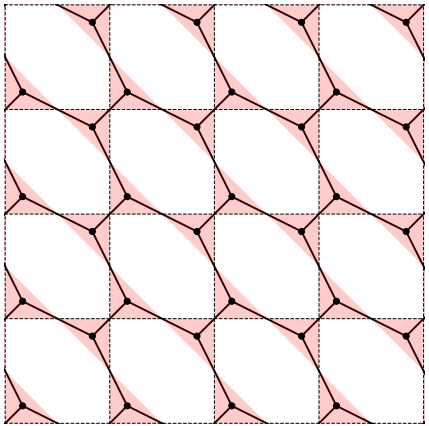
Introduced by Paetznick & Reichardt to implement logical gates

This work: *improve error correction performance*

Gauge fixes of the subsystem toric code

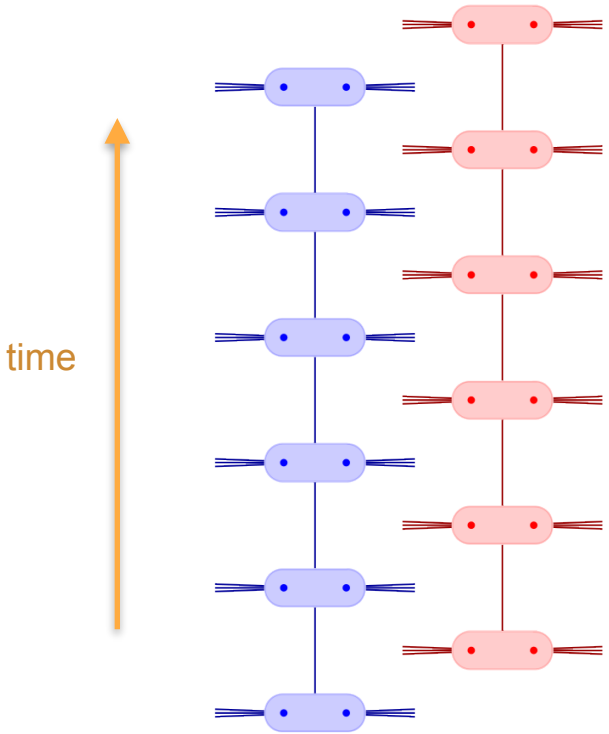
- Without gauge fixing, the X-type and Z-type matching graphs are both a triangular lattice, which has a threshold (using MWPM) of **6.5%** with perfect syndrome measurements
- By fixing X-type gauge operators as stabilisers (and removing Z-type), we obtain a hexagonal surface code. This has a Z threshold of **15.9%** and an X threshold of **6.5%**
- This compares to the surface code threshold of **10.3%** under perfect measurements





Fujii and Tokunaga "Error and loss tolerances of surface codes with general lattice structures." *Physical Review A* 86.2 (2012): 020303.

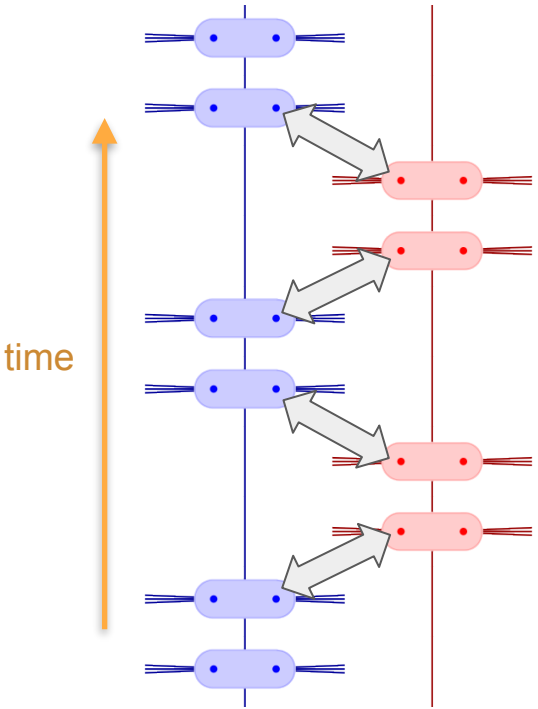
Schedule-Induced Gauge-Fixing



Example: Subsystem Surface Code

- Syndrome measurement
  - Dots are gauge checks
  - Ovals are stabilizer checks
  - Errors occur on edges
- X-checks and Z-checks alternate
- Values of gauge operators randomised

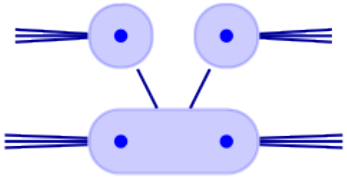
Schedule-Induced Gauge-Fixing



Consider schedule with repeated X/Z-checks

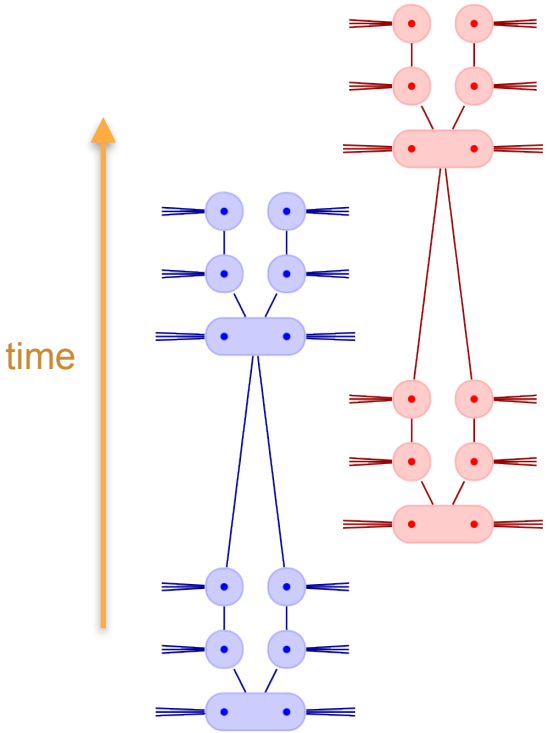
Gauge operators of different types
generally do not commute

Operators following same type (trivially) commute



Can use gauge checks as if they were stabilizers

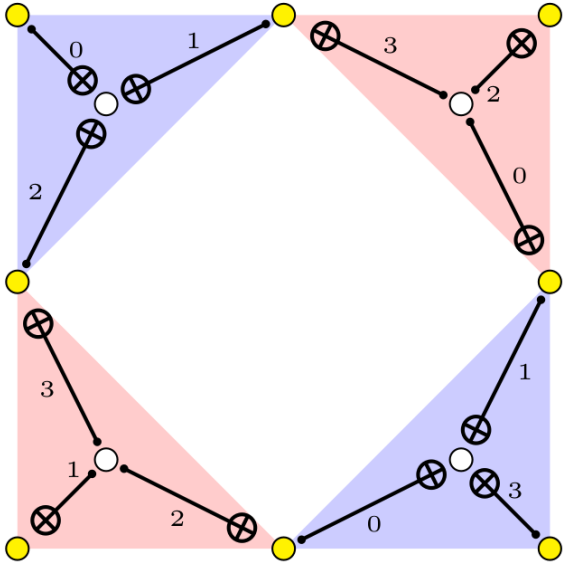
Schedule-Induced Gauge-Fixing

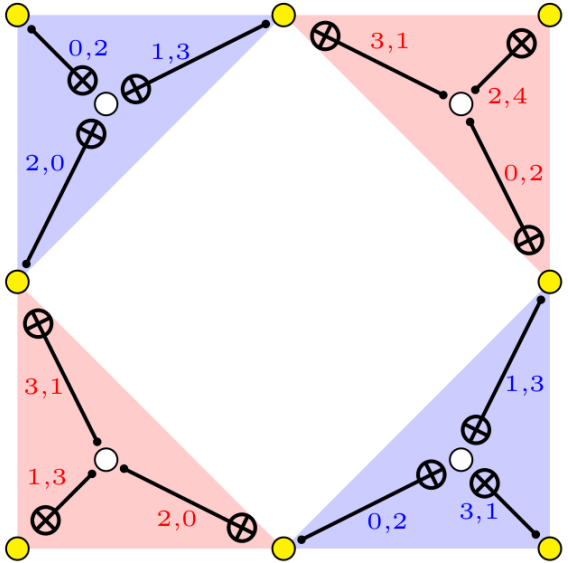


Different Schedules

- repeating same Pauli-type several times
- increase number of time steps where gauge measurements can be used as stabilizers
- average stabilizer weight decreases
- average node degree decreases

Simulation: Scheduling



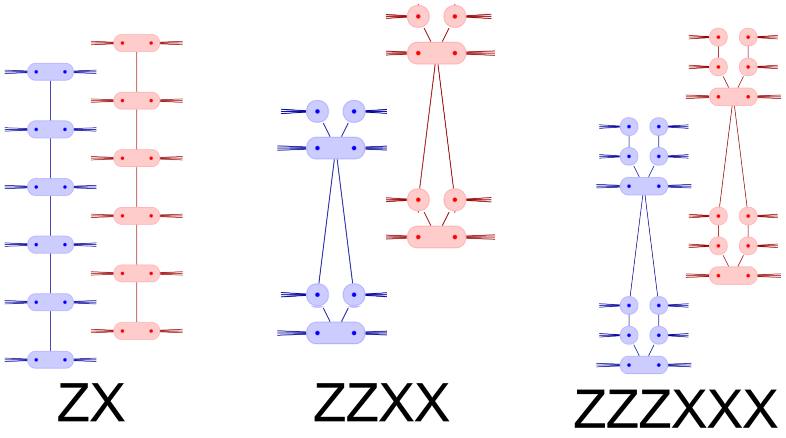


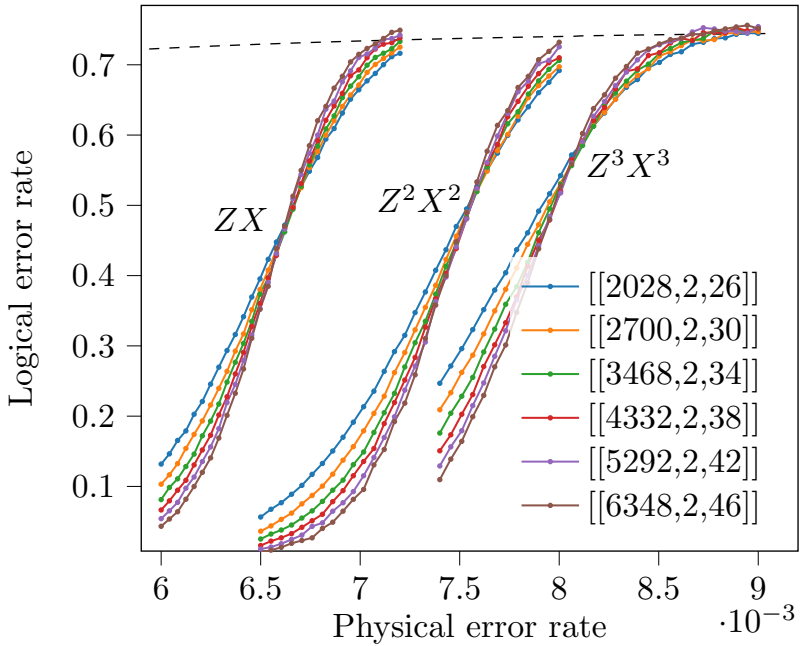
$(ZX)^r$

Z^r or X^r

weight-3 & gauge: no hook errors

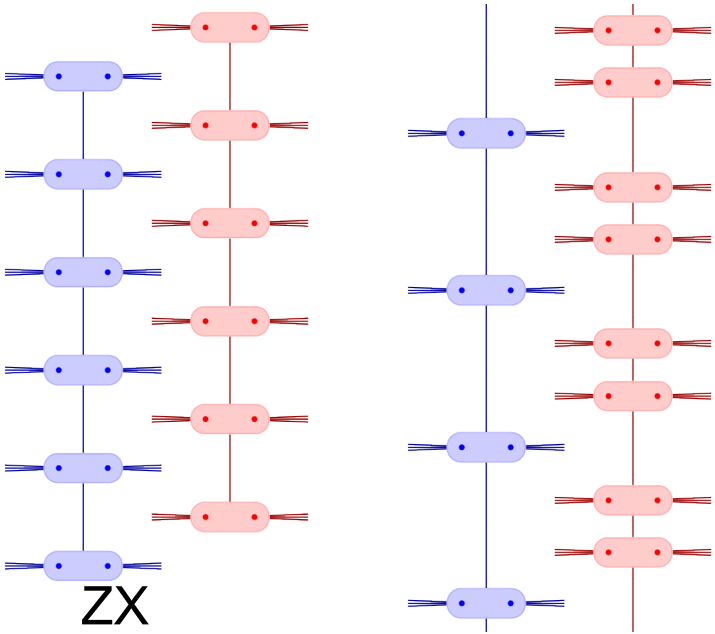
Simulation: Circuit-Level Depolarising Noise





- Changing schedule already improves threshold
- Two competing effects:
  - longer waits between “bursts”
  - average check weight per round is lower
  - vertex degree reduced
- Beyond 4 repetitions threshold decreases

Gauge-Fix Schedules

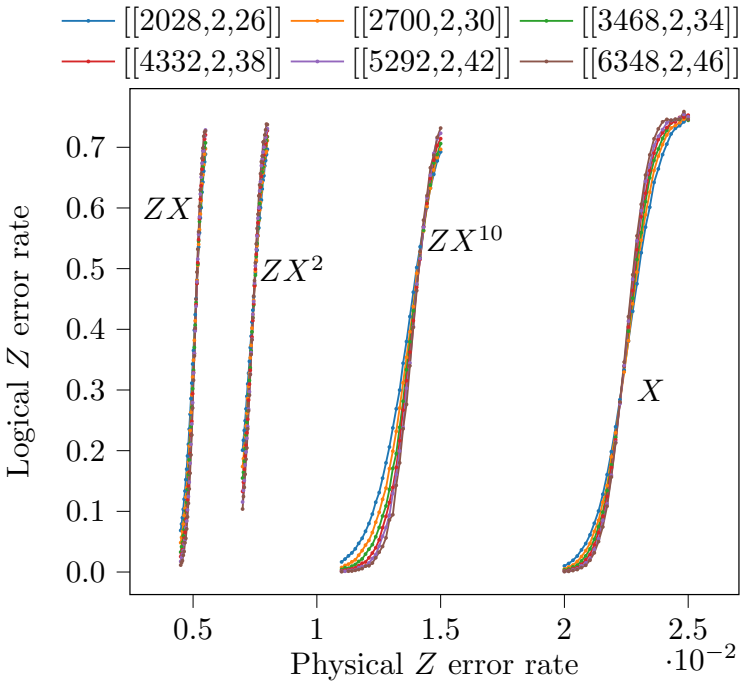


Balanced vs Imbalanced:

- *balanced*: repeat each Pauli-type the same number of times X^aZ^a
- *imbalanced*: schedule of the form X^aZ^b with $a \neq b$
- gather more syndrome for one Pauli-type: biased noise [1]

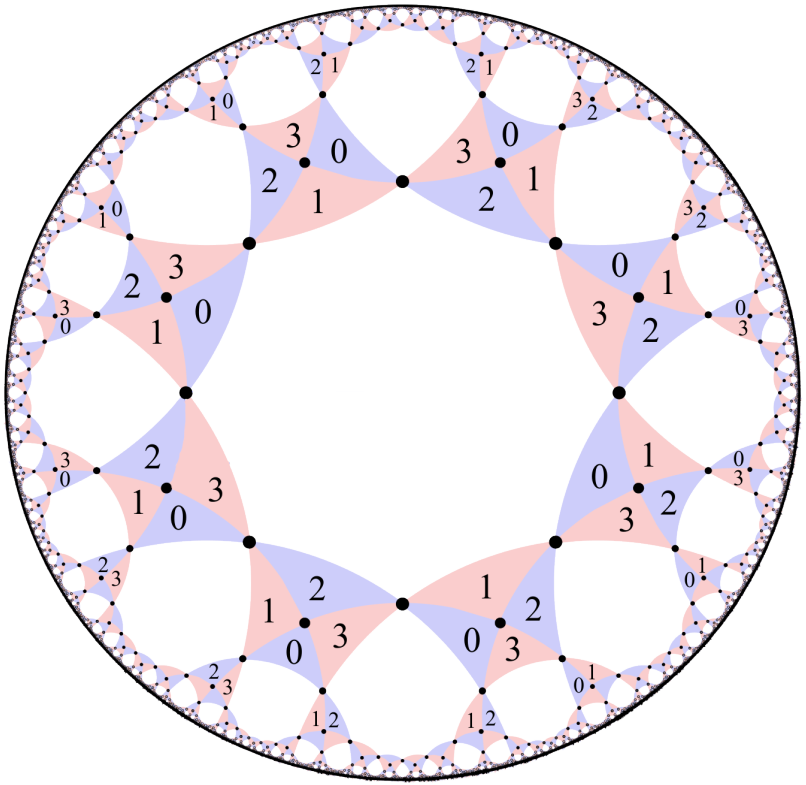
[1] Xu, Zhao, Yuan, Benjamin (2019). High-Threshold Code for Modular Hardware With Asymmetric Noise. *Physical Review Applied*, *12*(6)

Simulation: Circuit-Level Biased Noise



- Repeating X-measurements increases performance significantly
- longer sequences where X-gauge operators are fixed to stabilizers
- *upper bound*: only measure X-gauge operators (2.2%)

Scheduling Subsystem Hyperbolic Codes

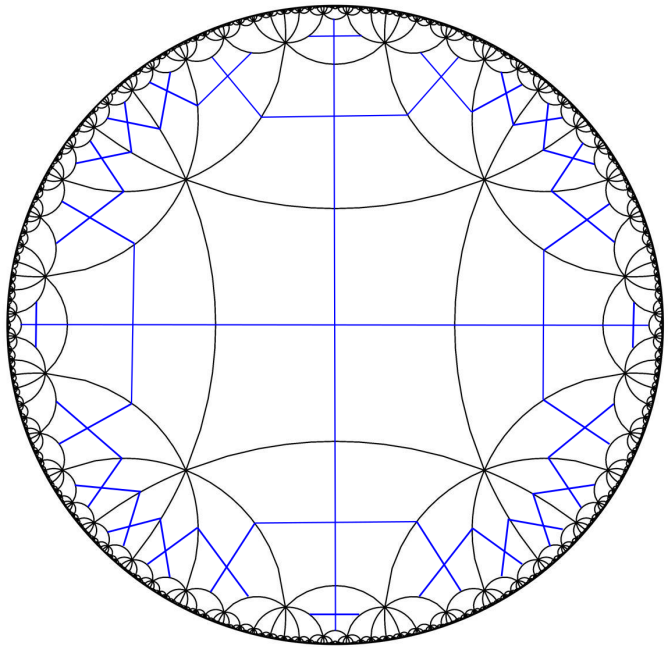


Scheduling

- Schedule constructed from symmetry group
- First finite-rate code (k/n→1/6) with efficient schedule (4 time steps)
- Each data qubit adjacent to 4 gauge operators → best we can hope for
- No hook errors
- Works for {4k,4} subsystem hyperbolic and semi-hyperbolic codes

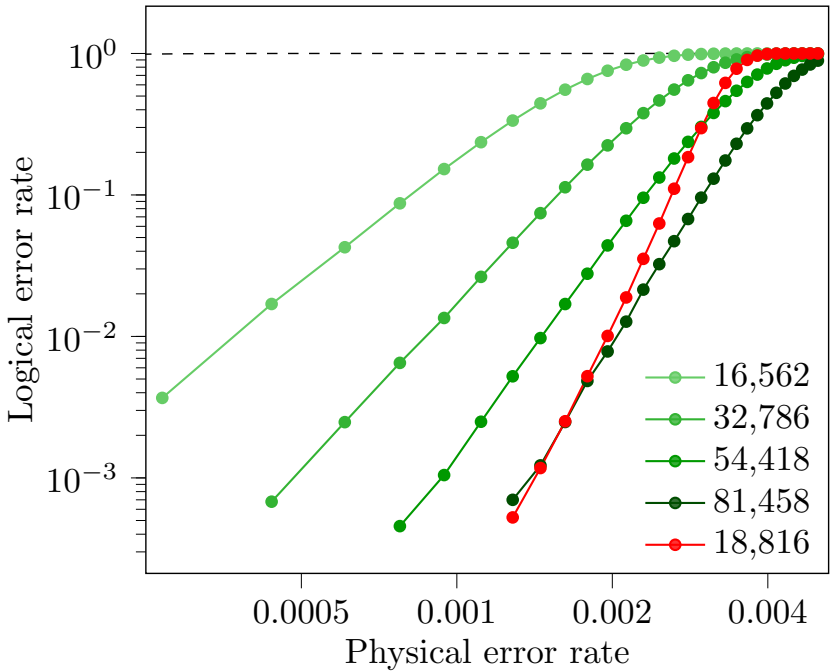
Semi-Hyperbolic Lattices

- Tile each face with a square lattice
- Distance scales as $\sqrt{n}$
- Subsystem semi-hyperbolic code derived from the dual of this lattice



[1] Breuckmann et al. "Hyperbolic and semi-hyperbolic surface codes for quantum storage." *Quantum Science and Technology* 2.3 (2017): 035007

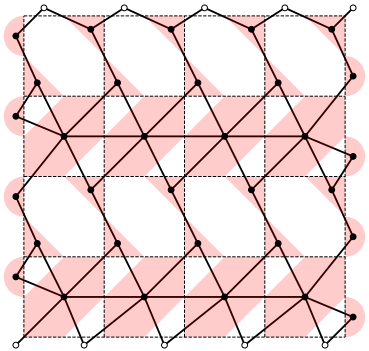
Subsystem Semi-Hyperbolic vs Rotated Surface Code



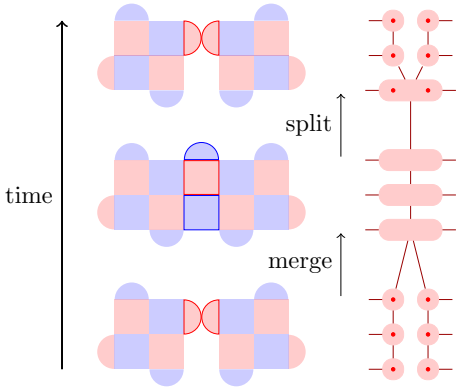
Subsystem {8,4} semi-hyperbolic (red) vs rotated surface code (green)

- Outperforms similar-rate (distance 6) rotated surface code below at least p=0.43%
- 4.3x reduction in qubit overhead at p=0.1%-0.2%
- Circuit-level depolarising noise
- ZX schedule

Other applications of our techniques



- **Inhomogeneous schedules**: changing the schedule in space can increase the Z distance



- **Lattice surgery**: schedule-induced gauge fixing can improve decoding for measurement-based logical CNOT gates in the surface code [1]

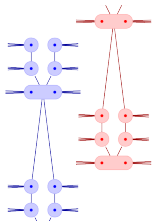
[1] Vuillot, et al. "Code deformation and lattice surgery are gauge fixing." *New Journal of Physics* 21.3 (2019): 033028.

Future Work

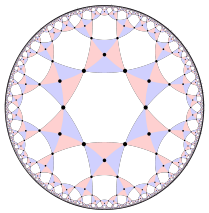
- Decoders for a larger variety of LDPC codes
  - Non-CSS
  - different decoders from MWPM
- Generalise scheduling of measurement circuits
  - Use fewer ancillas to reduce overhead further
  - Schedule wider classes of LDPC codes
- More constructions
  - Subsystem codes can be used as an ansatz to construct stabiliser codes via permanently gauge fixing [1]

[1] Li, et al. "2d compass codes." *Physical Review X* 9.2 (2019): 021041.

Conclusion



- **Schedule-Induced Gauge Fixing**
  - all in software with same hardware weight-3 checks
  - increase threshold of subsystem toric code for isotropic noise (0.67%→0.81%)
  - biased noise: increase up up to 2.2% for $p_z >> p_x$
  - idea is general: will immediately apply to many other codes & decoders



- **Finite Rate Subsystem Code**
  - rate k/n = 1/6 with weight-3 checks
  - first example of a code outperforming the surface code under circuit-level depolarising noise, reducing the overhead by 4.3x at ~0.2%