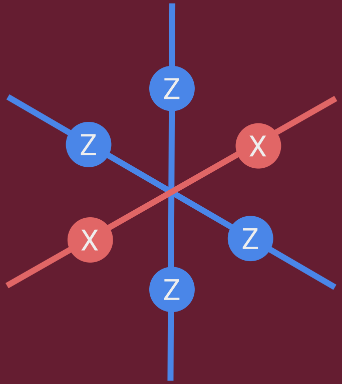
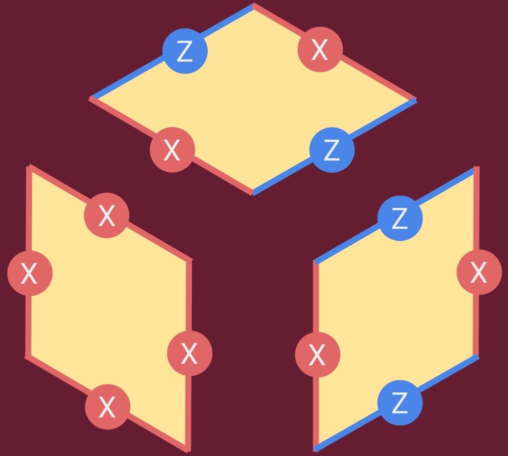
Tailoring 3D topological codes for biased noise

arXiv: 2211.02116



Arthur Pesah
University College London



Joint work with:
Eric Huang (University of Maryland)
Christopher Chubb (ETH Zürich)
Michael Vasmer (Perimeter Institute)
Arpit Dua (Caltech)

Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

Biased noise: Z errors more likely than X and Y errors

Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

Biased noise: Z errors more likely than X and Y errors

Experimentally demonstrated for several types of quantum systems (e.g. cat qubits)

Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

Biased noise: Z errors more likely than X and Y errors

Experimentally demonstrated for several types of quantum systems (e.g. cat qubits)

Typical bias level: $\eta = 100$ (e.g. at AWS), i.e. Z errors 100x more likely than X and Y

Overview

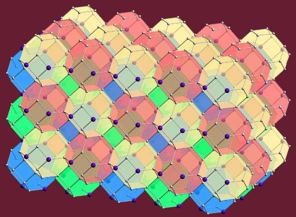
Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

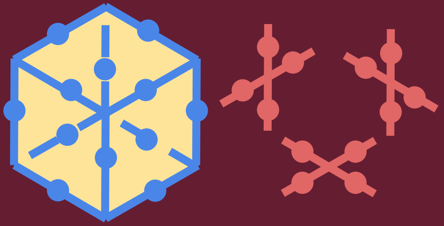
② 3D topological code

Three main code families considered in this work:

3D toric codes



Color code



Fracton codes

Overview

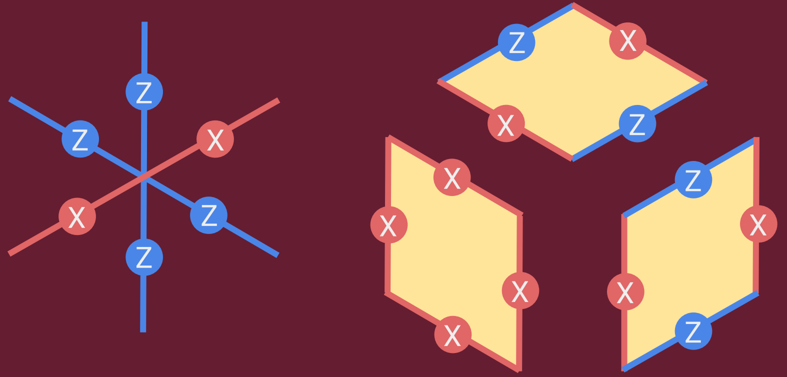
Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

② 3D topological code

③ Small changes

Clifford-deformation: we apply a Clifford gate (typically a Hadamard) on one axis



Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

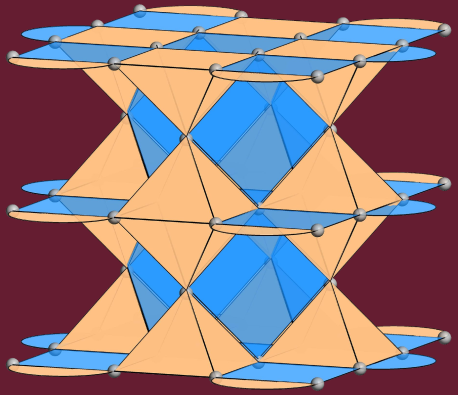
① Biased noise

② 3D topological code

③ Small changes

Clifford-deformation: we apply a Clifford gate (typically a Hadamard) on one axis

Dimension and layout: rotated 3D toric code



Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

② 3D topological code

③ Small changes

④ Big improvements

Code threshold of 50% at infinite bias using for all our codes

Overview

Under biased noise, small changes to 3D topological codes can result in big improvements of their performance

① Biased noise

② 3D topological code

③ Small changes

④ Big improvements

Code threshold of 50% at infinite bias using for all our codes

Subthreshold error rate of the 3D rotated toric code with some specific dimensions scales as

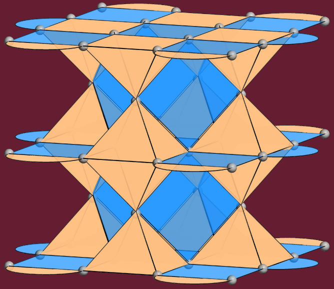
$$\bar{p} \propto e^{-\alpha d^3}$$

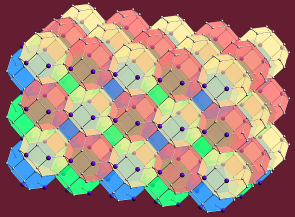
with the distance d of the code

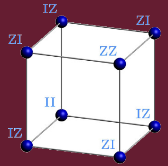
What you will learn in this talk

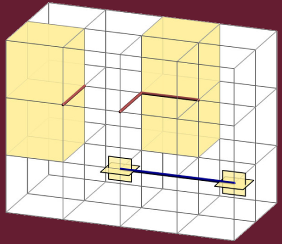
What are 3D codes and why are they interesting?

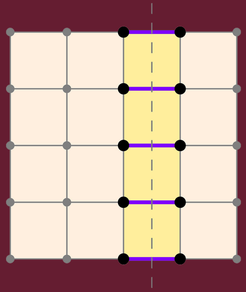
How to prove that a code has a 50% threshold?

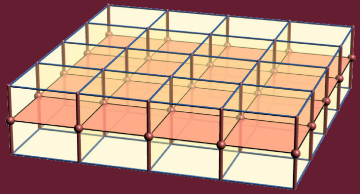


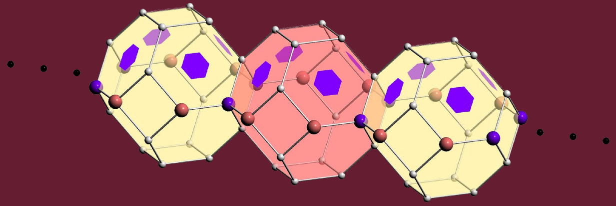












Single-shot

Partial self-correction

Transversal T

New phases of matter

Materialized symmetries

Weight-reduction

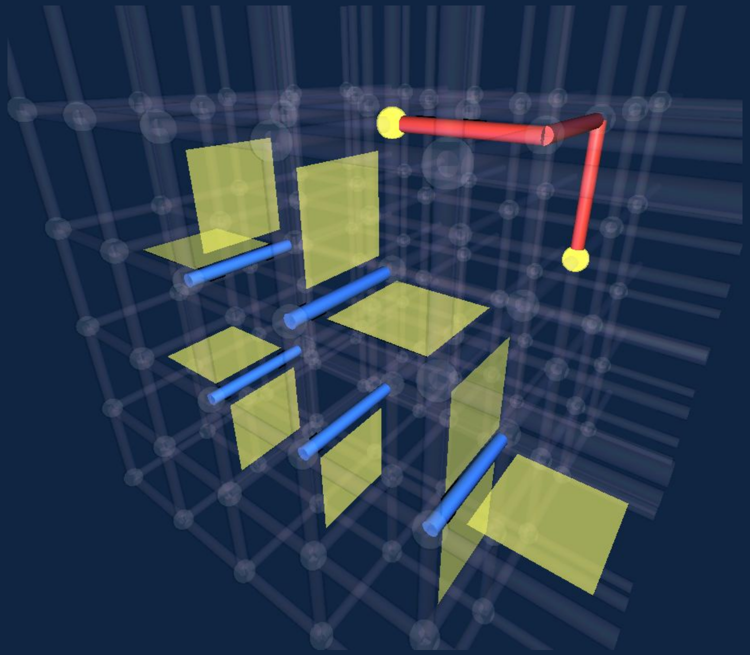
Repetition codes

Outline

① A tour of 3D topological codes

② Clifford deformations of quantum codes

③ Code boundaries and subthreshold scaling



A TOUR OF 3D TOPOLOGICAL CODES

Why are 3D codes interesting?

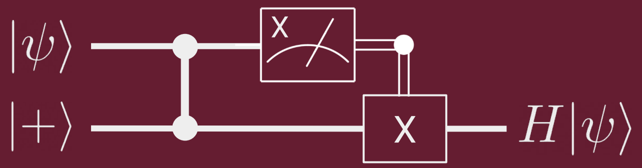
1. They can implement transversal non-Clifford gates

Bravyi-König theorem: transversal gates of a D-dimensional code are restricted to the Dth level of the Clifford hierarchy

⇒ 3D codes can (in principle) implement a T gate transversally, while 2D cannot (costly methods like magic state distillation are required)

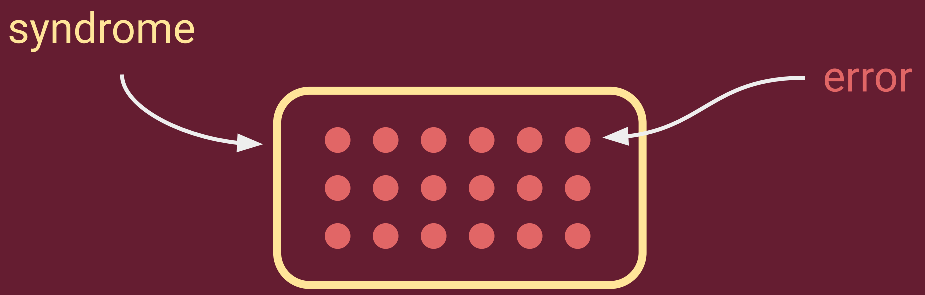
Eastin-Knill theorem: no code has a universal set of transversal gate

⇒ 3D codes often have a non-Clifford gate that cannot be implemented transversally (e.g. Hadamard), but state injection is possible for them without distillation.



Why are 3D codes interesting?

1. They can implement transversal non-Clifford gates
2. They can have single-shot error correction



Why are 3D codes interesting?

1. They can implement transversal non-Clifford gates
2. They can have single-shot error correction

Why are 3D codes interesting?

1. They can implement transversal non-Clifford gates
2. They can have single-shot error correction

Examples:

- 3D toric/color code for Z errors
- Subsystem 3D toric/color code for all errors

Why are 3D codes interesting?

1. They can implement transversal non-Clifford gates

2. They can have single-shot error correction

3. They can have partial self-correction

   Self-correction: when putting the code in a thermal bath, the coherence time of the logical qubits is exponential in the lattice size (no decoding needed)

   Partial self-correction: the coherence time is exponential up to a given lattice size, then decreases

   Fractons such as the Haah code have partial self-correction

Why are 3D codes interesting?

1. They can implement transversal non-Clifford gates
2. They can have single-shot error correction
3. They can have partial self-correction
4. They correspond to interesting new phases of matter

   2D translation-invariant stabilizer codes have been fully classified (for prime dimensional qudits), and they are all copies of the 2D toric codes up to local unitaries *[Haah, 2018]*

   On the other hand, 3D codes are much more diverse (e.g. with fractons).
   Classifying all 3D phases is still an open problem.

What is the catch?

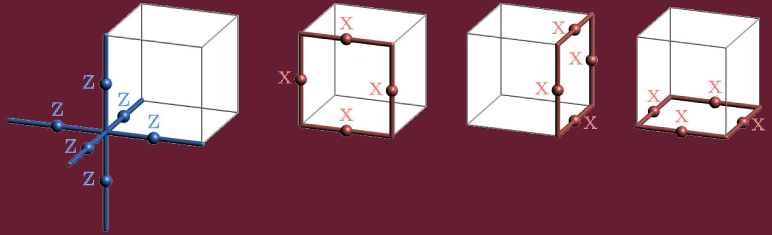
1. They require a higher connectivity
2. They often require more qubits to achieve a given distance
3. This added overhead can make their non-Clifford gates more costly than magic state distillation [Kubica et al., 2021]

However, several reasons to be optimistic:

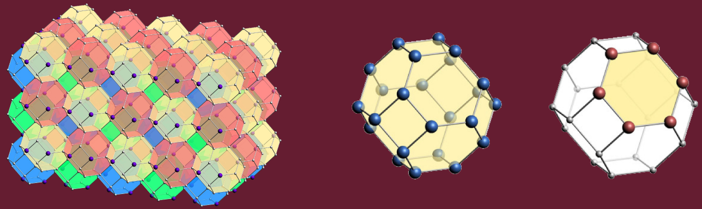
1. Recent work on single-shot decoding of the 3D subsystem toric code has shown a considerably improved threshold [Kubica & Vasmer, 2022]
2. Fractal 3D codes could improve the qubit count of those codes [Zhu et al, 2021]
3. This work: biased noise can also improve the threshold

Main 3D code families

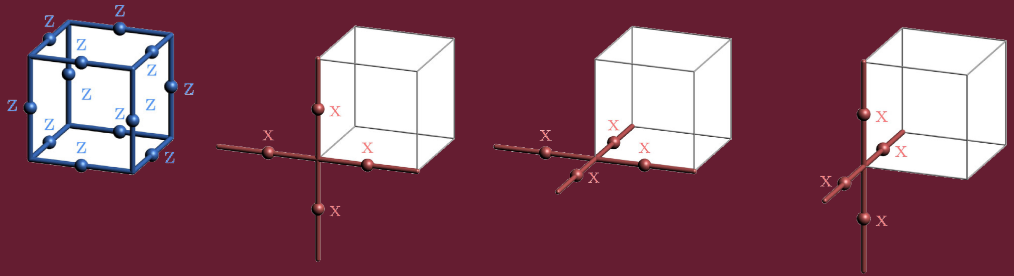
3D toric codes

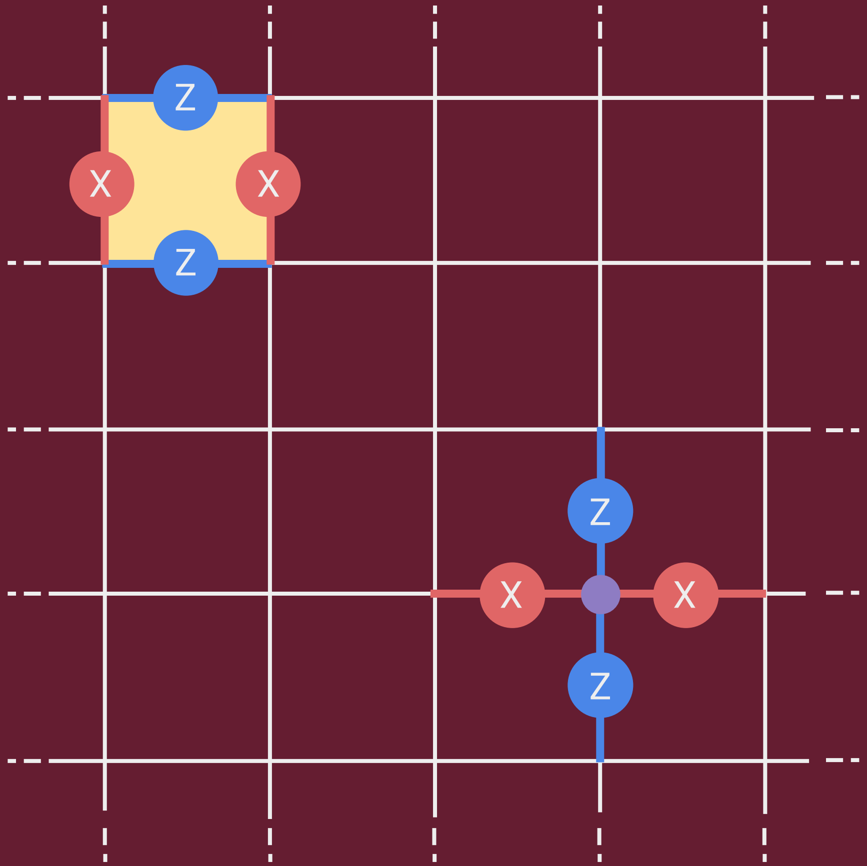


3D color codes



Fracton codes
(e.g. X-cube model)





CLIFFORD-DEFORMATION OF QUANTUM CODES

XZZX surface code

Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code

Motivation:

1) Classical codes usually have a 50% threshold (e.g. rep. code)

Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code

Motivation:

1) Classical codes usually have a 50% threshold (e.g. rep. code)

2) If we have infinite bias noise (e.g. pure Z noise), we could use a classical code and obtain a 50% threshold

Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code

Motivation:

1) Classical codes usually have a 50% threshold (e.g. rep. code)

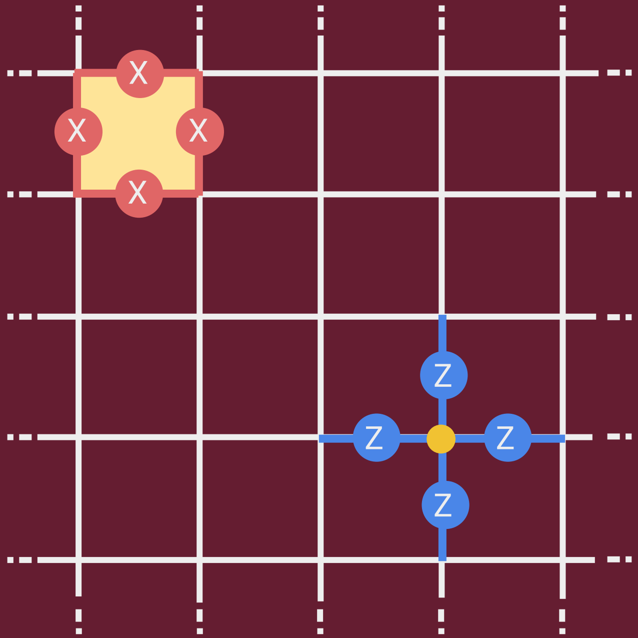
2) If we have infinite bias noise (e.g. pure Z noise), we could use a classical code and obtain a 50% threshold

3) However, the surface code (and many other codes) don't have a 50% threshold at infinite bias (e.g. the surface code has 10%)

Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code

Goal: find stabilizers that work better under biased noise

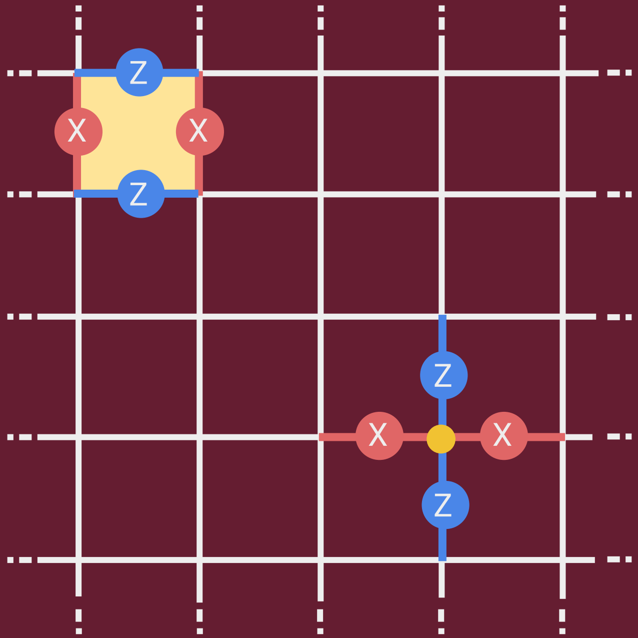


Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XXZX surface code

Goal: find stabilizers that work better under biased noise

Idea: apply a Hadamard operator on the horizontal axis



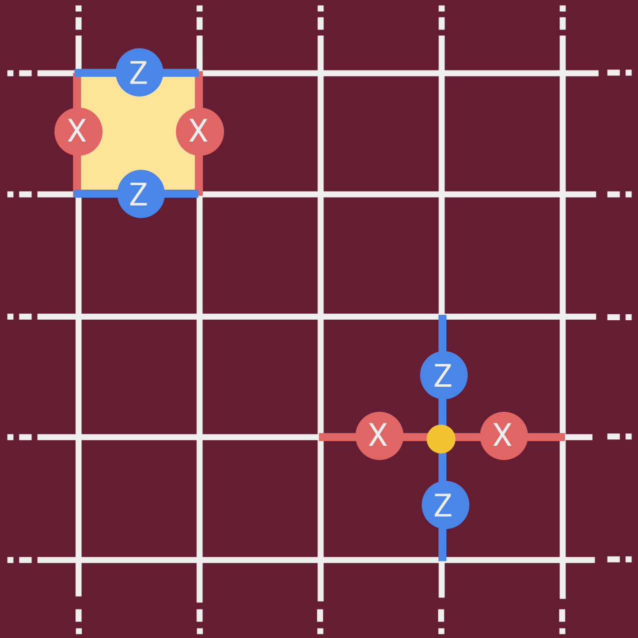
Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code

Goal: find stabilizers that work better under biased noise

Idea: apply a Hadamard operator on the horizontal axis

Infinite Z bias: the Z part of the stabilizers becomes useless



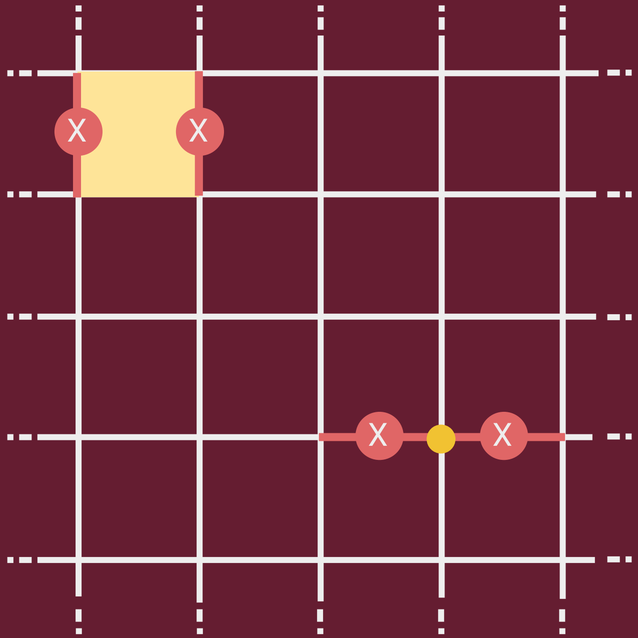
Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XXZX surface code

Goal: find stabilizers that work better under biased noise

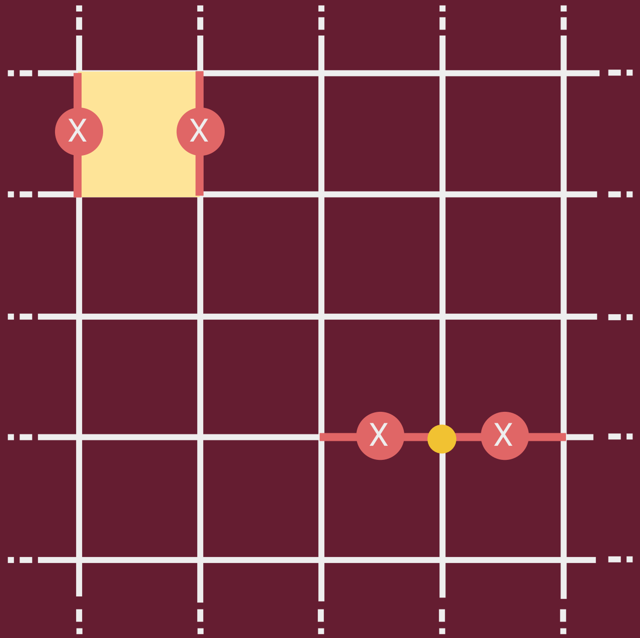
Idea: apply a Hadamard operator on the horizontal axis

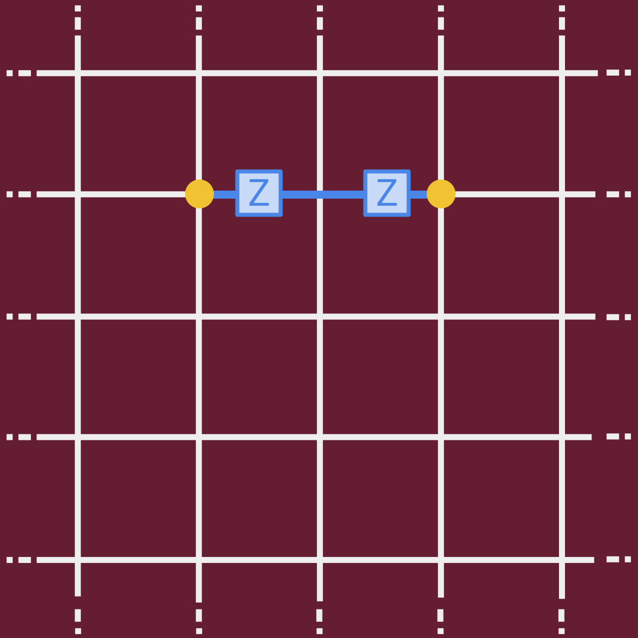
Infinite Z bias: the Z part of the stabilizers becomes useless



Bonilla Ataides et al., *The XZZX Surface Code*, 2020

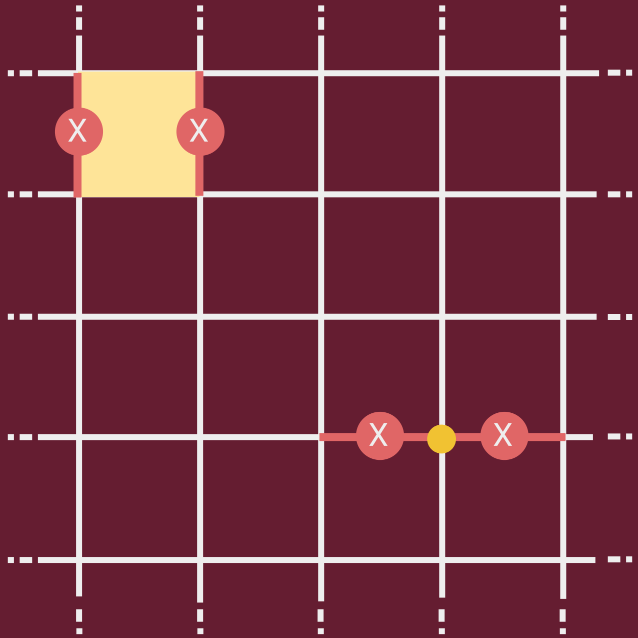
XZZX surface code

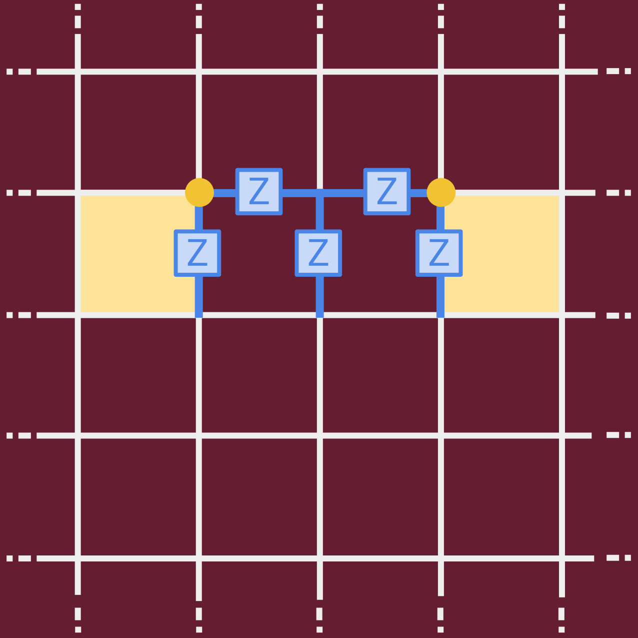




Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code





Bonilla Ataides et al., *The XZZX Surface Code*, 2020

XZZX surface code

Extremely biased noise

Only Z errors

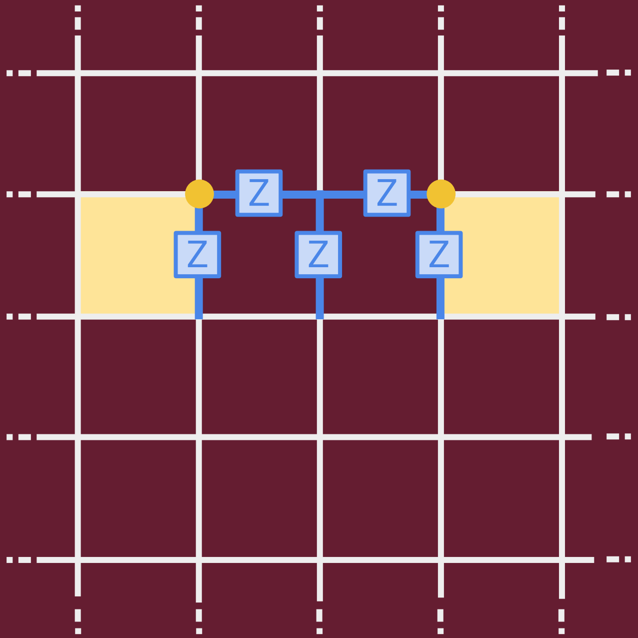
Decoding problem

Tackle each row of the lattice independently

Threshold?

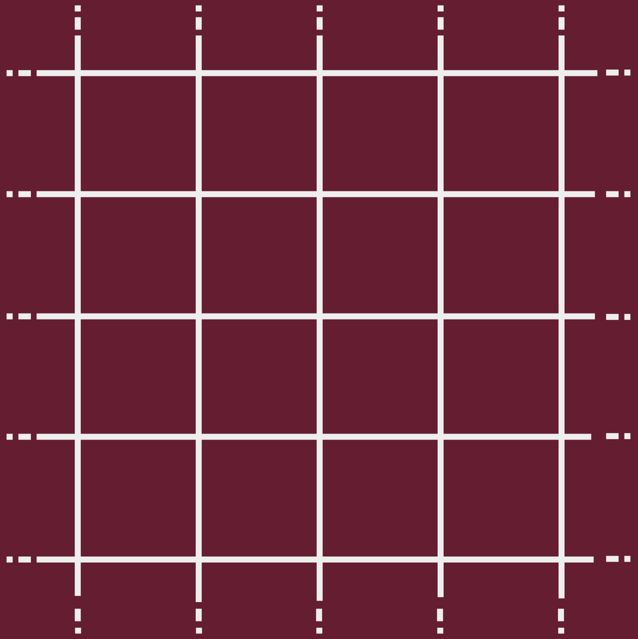
50%

(same as the repetition code)



XZZX surface code

The symmetry perspective

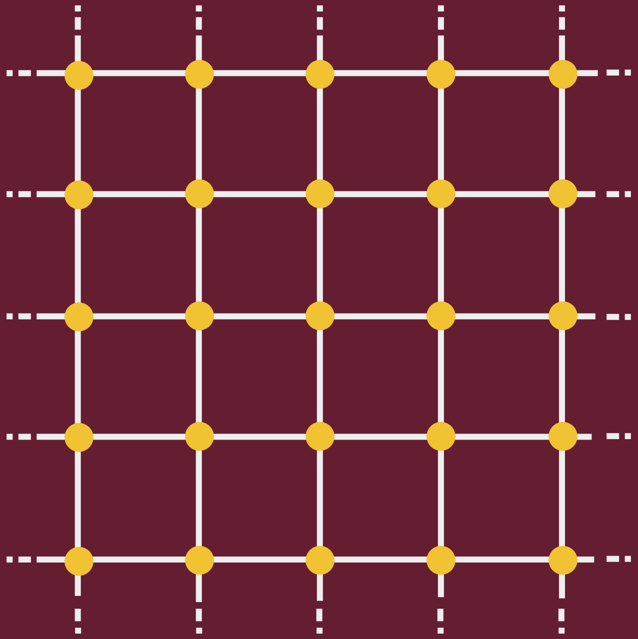


XZZX surface code

The symmetry perspective

In the normal surface code, we have:

$$\prod_{f\in\text{lattice}} S_f = I \qquad \prod_{v\in\text{lattice}} S_v = I$$



XXZX surface code

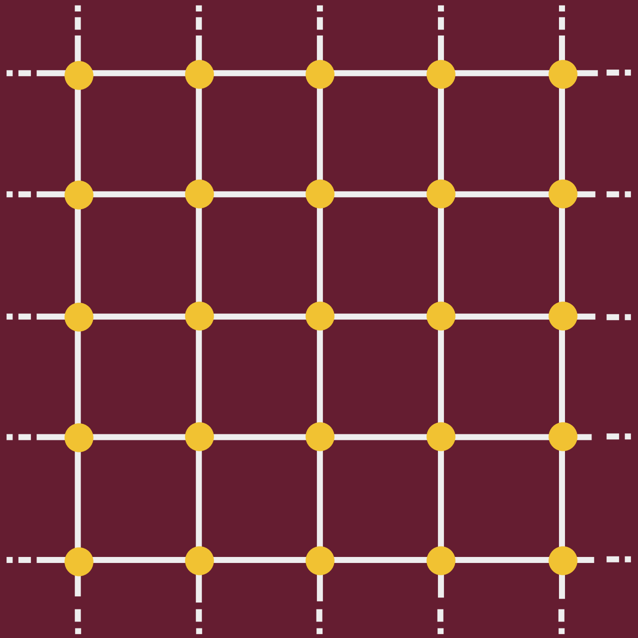
The symmetry perspective

In the normal surface code, we have:

$$\prod_{f \in \text{lattice}} S_f = I \qquad \prod_{v \in \text{lattice}} S_v = I$$

That's what we call a materialized symmetry & it leads to a conservation law for the syndrome:

$$\prod_{v \in \text{lattice}} s_v = 1 \qquad \prod_{f \in \text{lattice}} s_f = 1$$



XZZX surface code

The symmetry perspective

In the normal surface code, we have:

$$\prod_{f \in \text{lattice}} S_f = I \qquad \prod_{v \in \text{lattice}} S_v = I$$

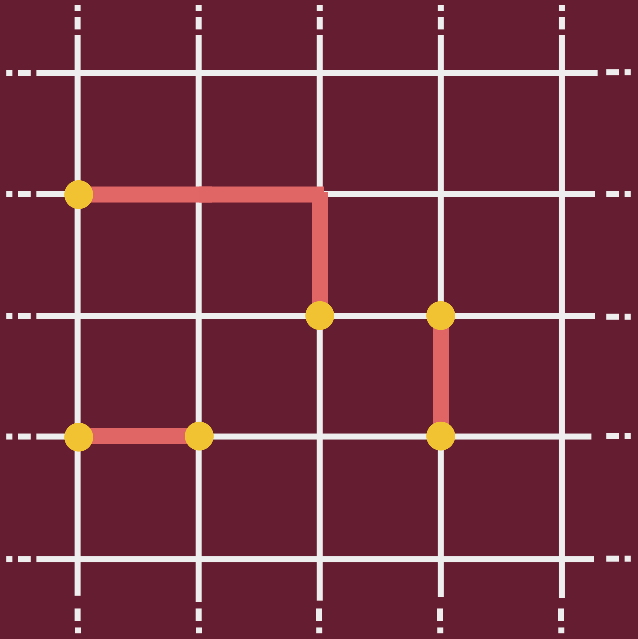
That's what we call a materialized symmetry & it leads to a conservation law for the syndrome:

$$\prod_{v \in \text{lattice}} s_v = 1 \qquad \prod_{f \in \text{lattice}} s_f = 1$$

⇒ even number of -1 in the syndrome

⇒ even number of face and vertex excitations

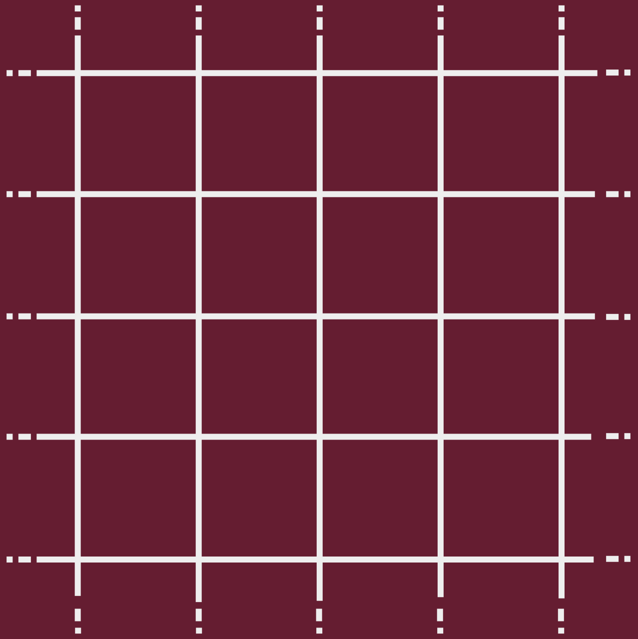
⇒ matching!



XZZX surface code

The symmetry perspective

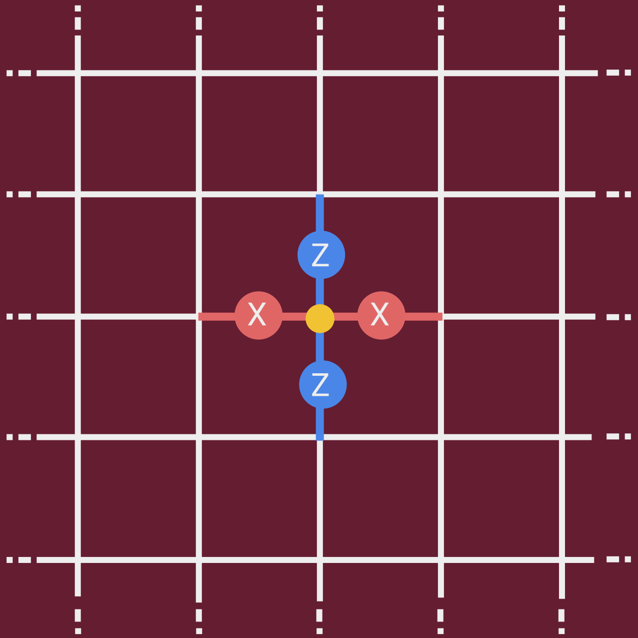
In the XZZX surface code, we have effective linear symmetries under pure Z noise:



XZZX surface code

The symmetry perspective

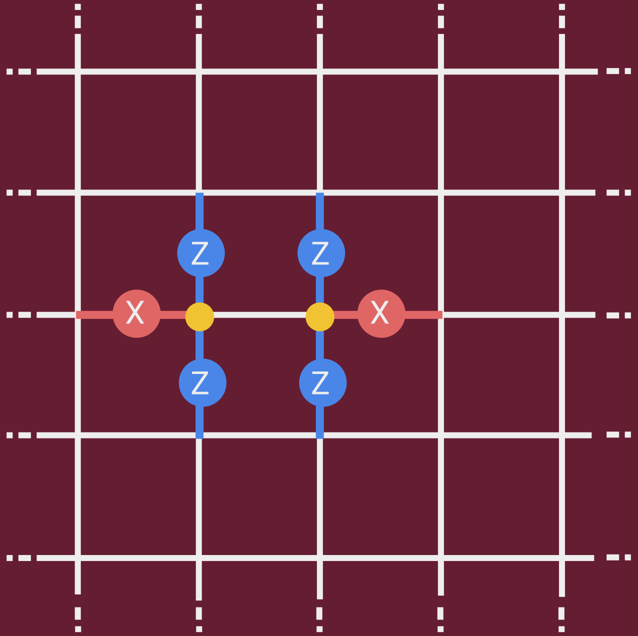
In the XZZX surface code, we have effective linear symmetries under pure Z noise:



XZZX surface code

The symmetry perspective

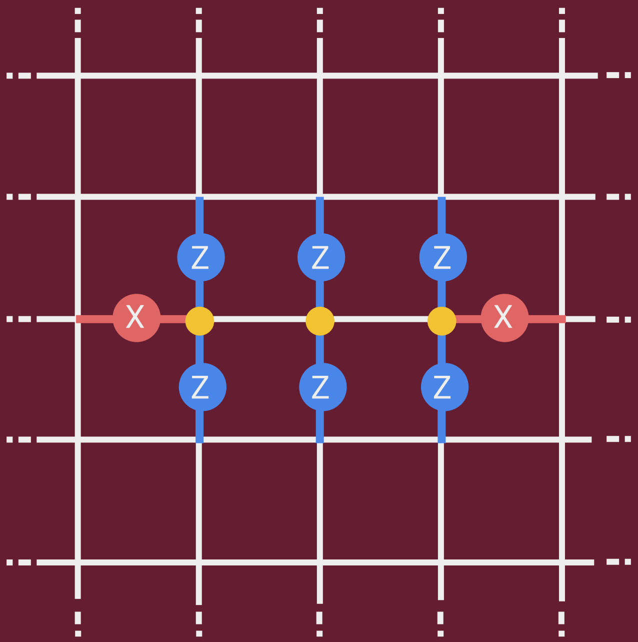
In the XZZX surface code, we have effective linear symmetries under pure Z noise:



XZZX surface code

The symmetry perspective

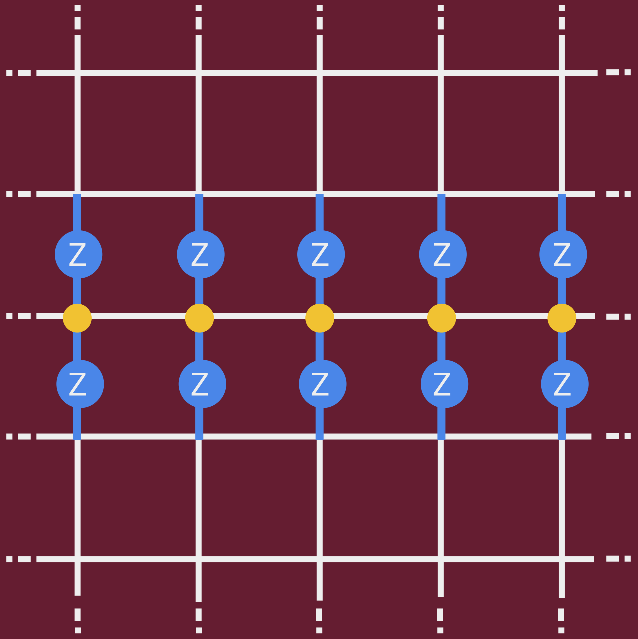
In the XZZX surface code, we have effective linear symmetries under pure Z noise:



XZZX surface code

The symmetry perspective

In the XZZX surface code, we have effective linear symmetries under pure Z noise:



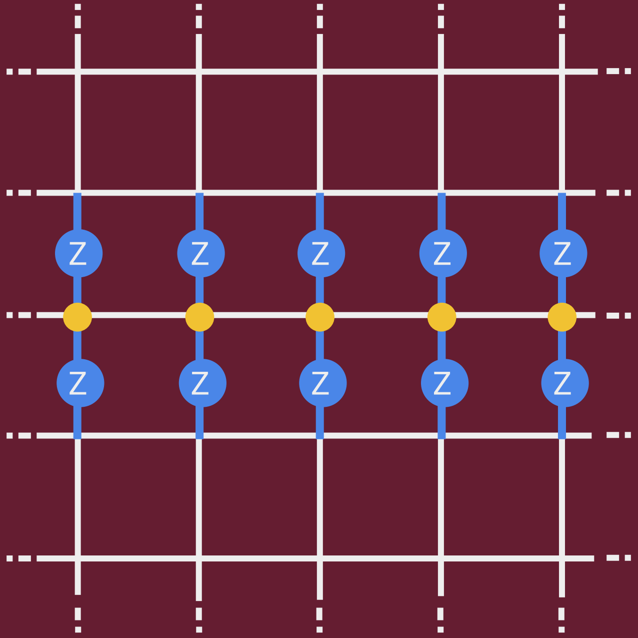
XZZX surface code

The symmetry perspective

In the XZZX surface code, we have effective linear symmetries under pure Z noise:

$$\prod_{f \in \mathrm{row}} S_f = I \qquad \prod_{v \in \mathrm{row}} S_v = I$$

as the Z part of stabilizers is irrelevant under pure Z noise



XXZZX surface code

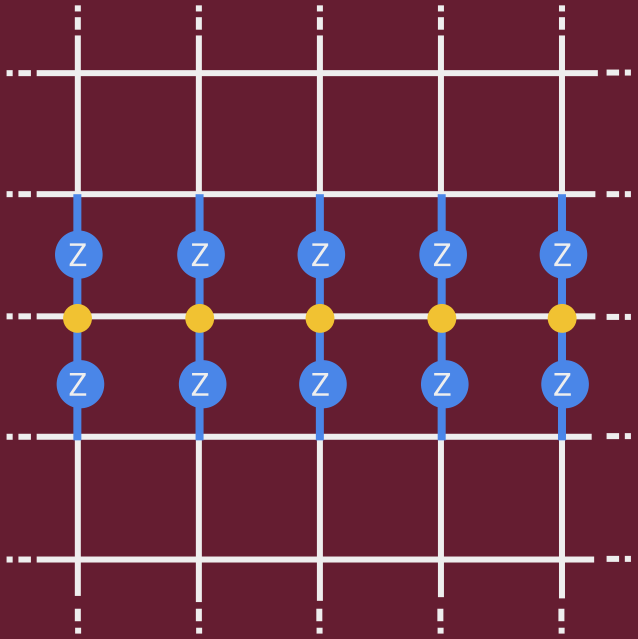
The symmetry perspective

In the XZZX surface code, we have effective linear symmetries under pure Z noise:

$$\prod_{f \in \mathrm{row}} S_f = I \qquad \prod_{v \in \mathrm{row}} S_v = I$$

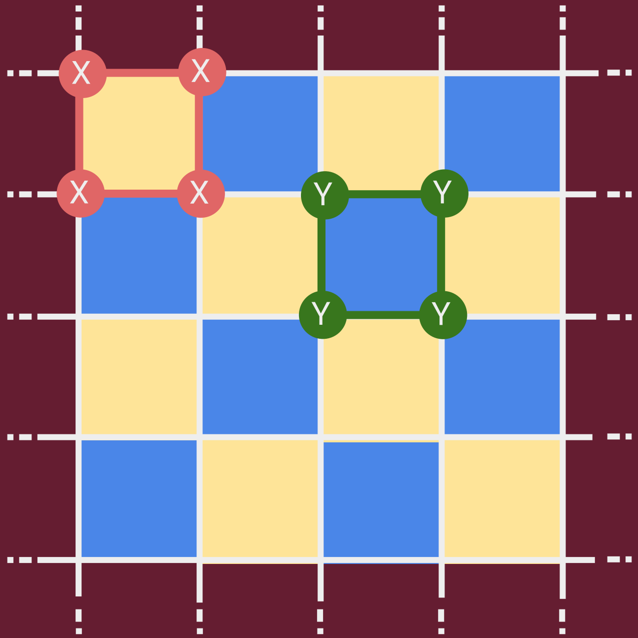
as the Z part of stabilizers is irrelevant under pure Z noise

⇒ even number of excitation along each line
⇒ matching along each line!



XY surface code

Clifford-deformation:
Hadamard + S gate on all qubits

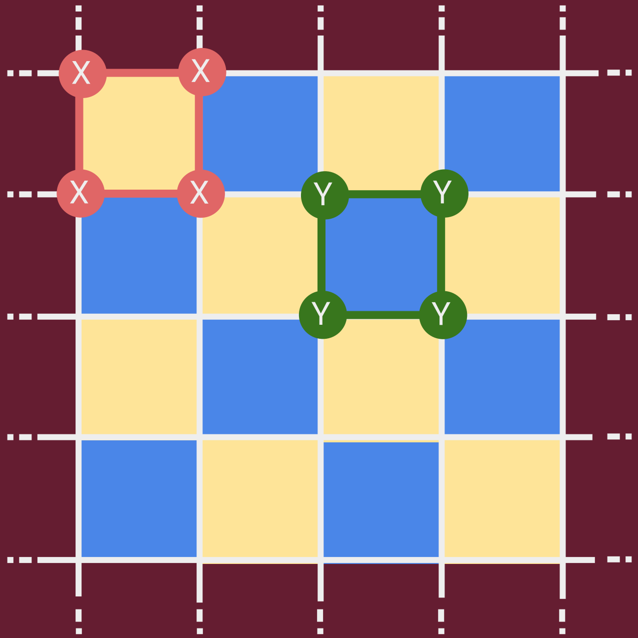


Tuckett et al., *Tailoring surface codes for highly biased noise*, 2018

XY surface code

Clifford-deformation: Hadamard + S gate on all qubits

Infinite Z-bias: X and Y acts similarly

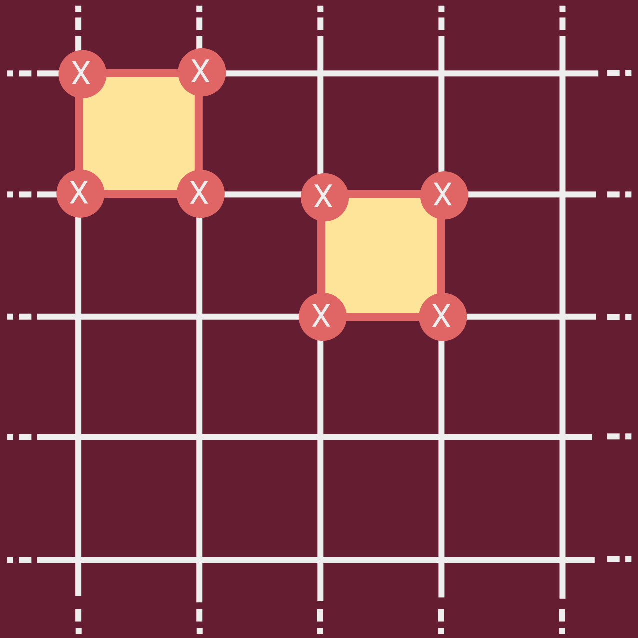


Tuckett et al., *Tailoring surface codes for highly biased noise*, 2018

XY surface code

Clifford-deformation:
Hadamard + S gate on all qubits

Infinite Z-bias: X and Y acts similarly



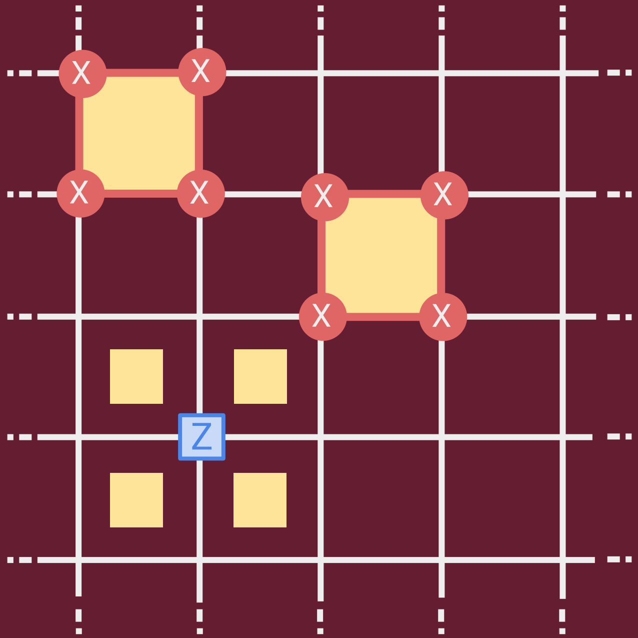
Tuckett et al., *Tailoring surface codes for highly biased noise*, 2018

XY surface code

Clifford-deformation: Hadamard + S gate on all qubits

Infinite Z-bias: X and Y acts similarly

Z errors activate the 4 neighboring plaquettes



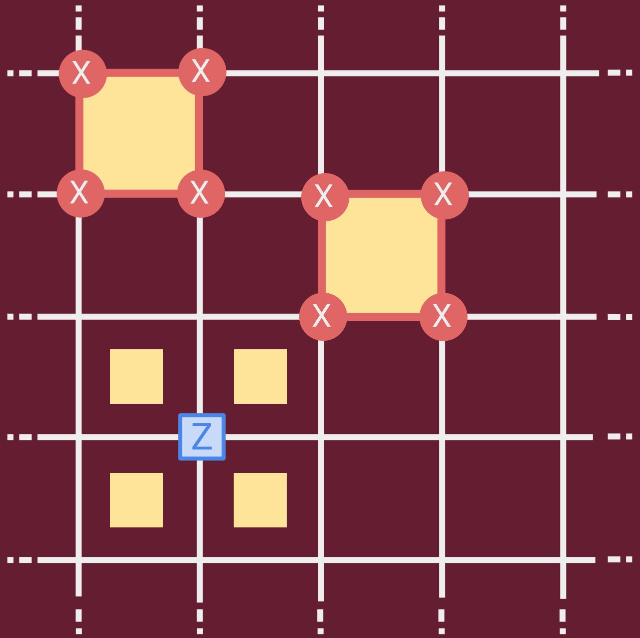
Tuckett et al., *Tailoring surface codes for highly biased noise*, 2018

XY surface code

Clifford-deformation: Hadamard + S gate on all qubits

Infinite Z-bias: X and Y acts similarly

Z errors activate the 4 neighboring plaquettes

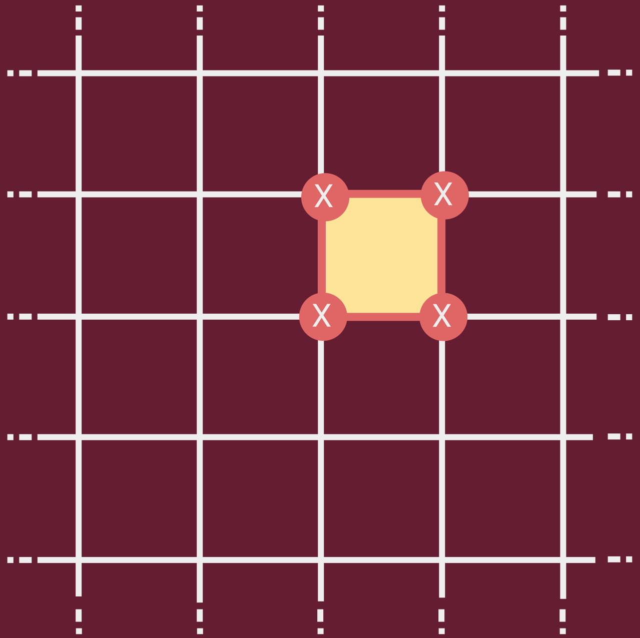


Question: why does this code has a 50% threshold?

Tuckett et al., *Tailoring surface codes for highly biased noise*, 2018

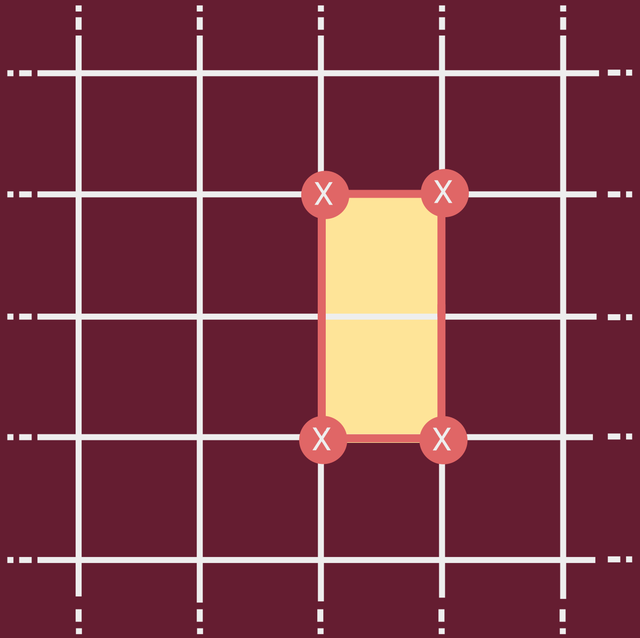
XY surface code

Materialized symmetry:
along every row & column



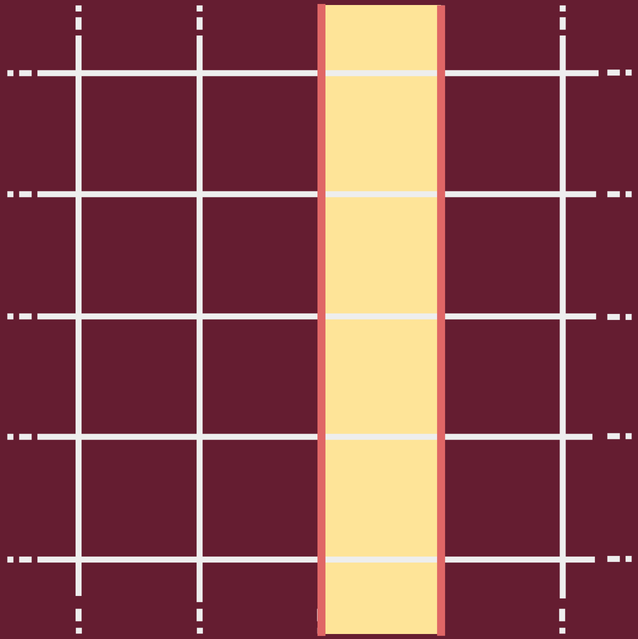
XY surface code

Materialized symmetry:
along every row & column



XY surface code

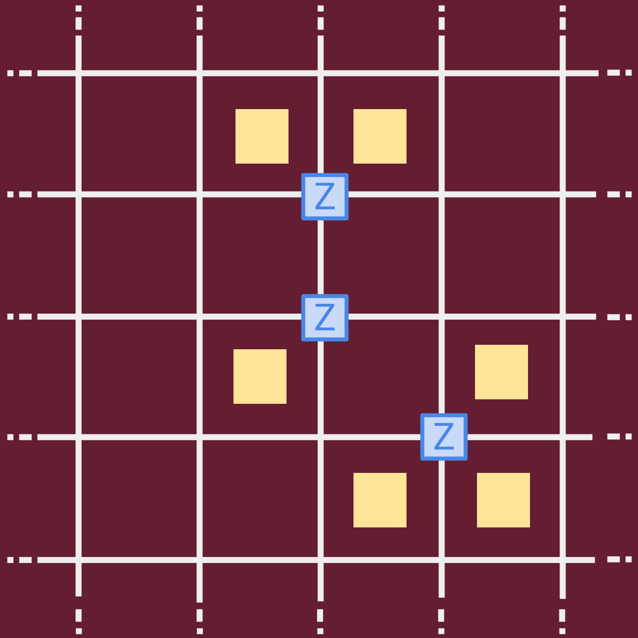
Materialized symmetry:
along every row & column



XY surface code

Materialized symmetry:
along every row & column

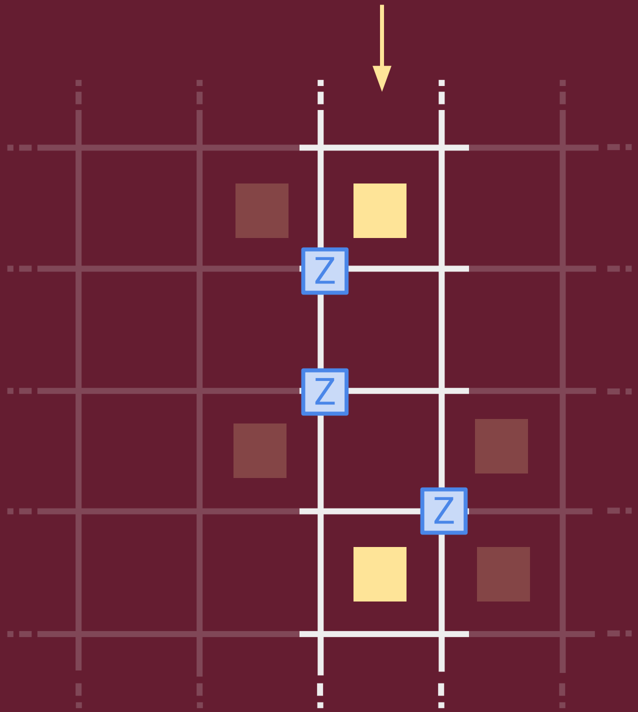
Conservation law:
Each column has an even number of excitations



XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

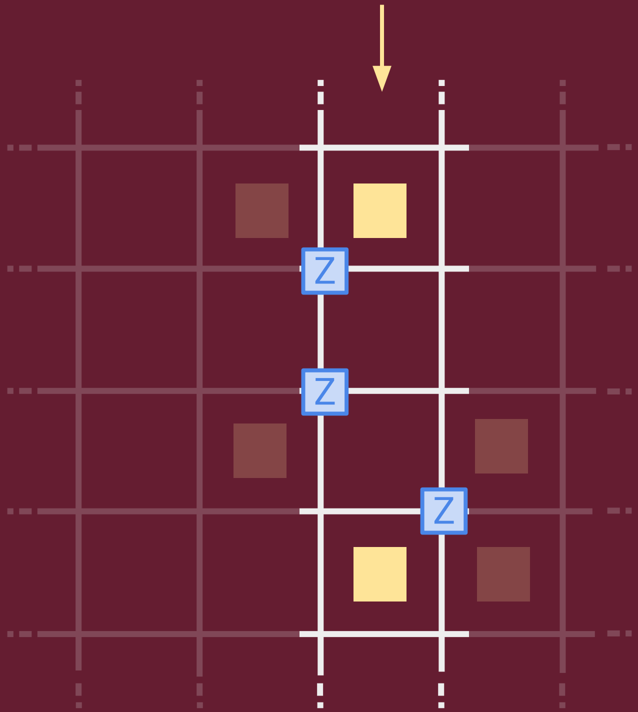


XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

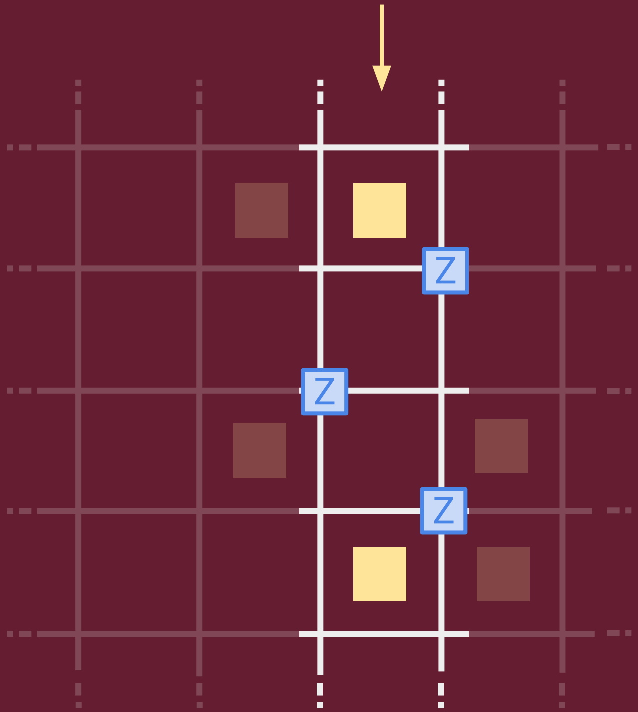


XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

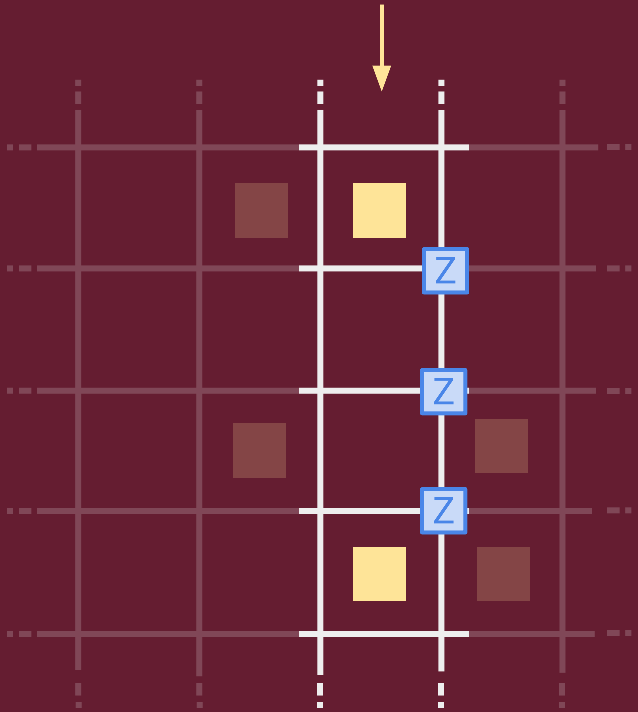


XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

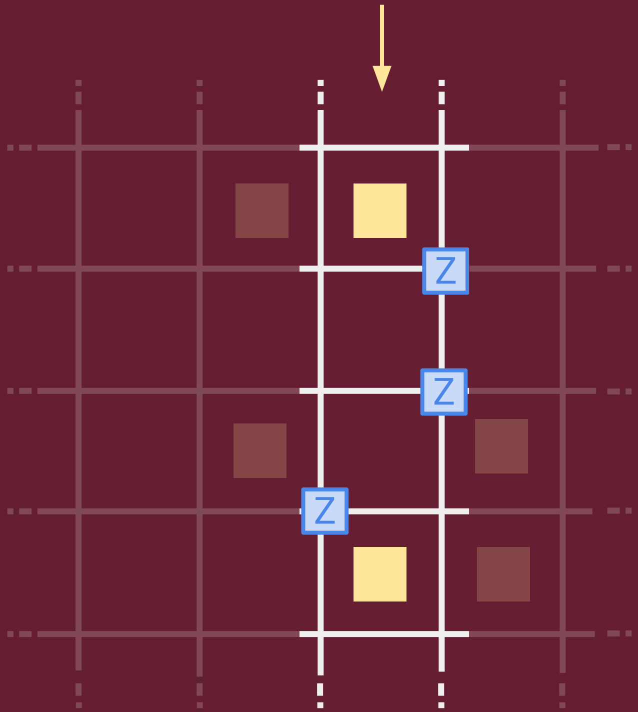


XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

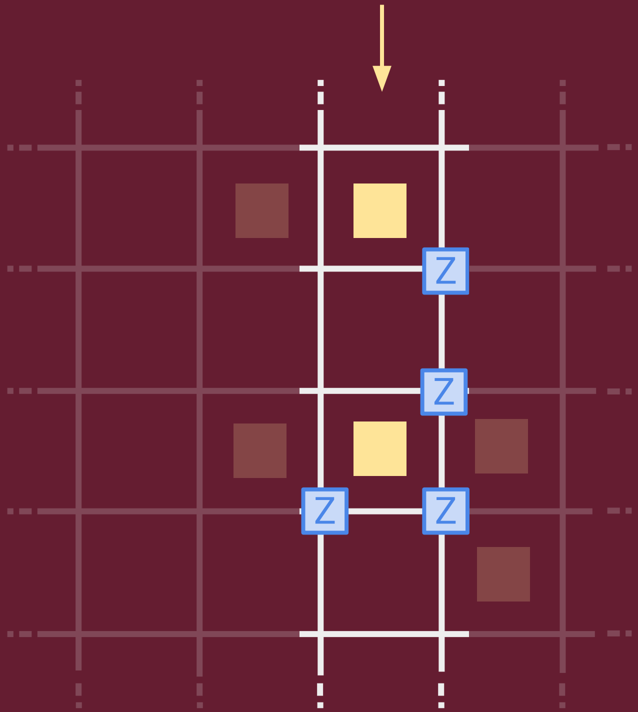


XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column



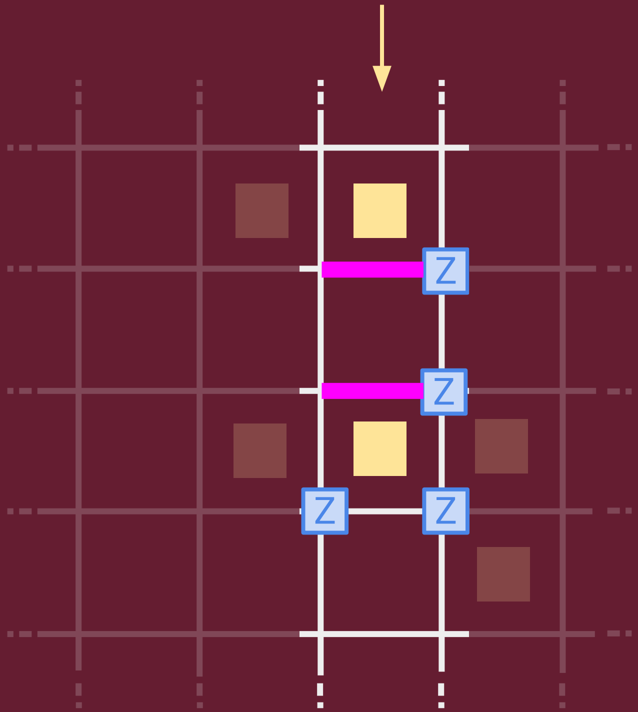
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



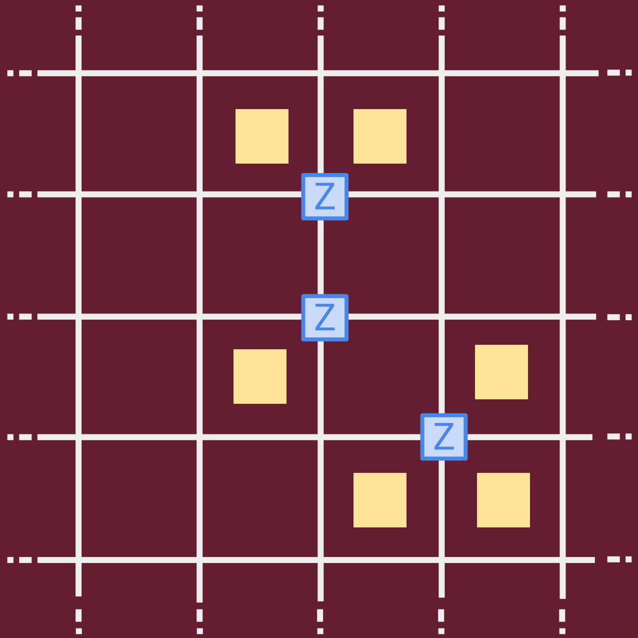
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

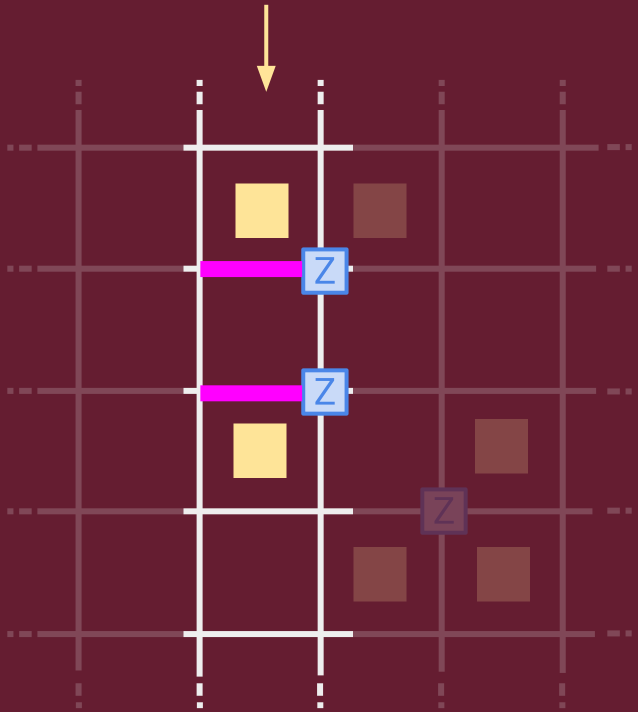
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

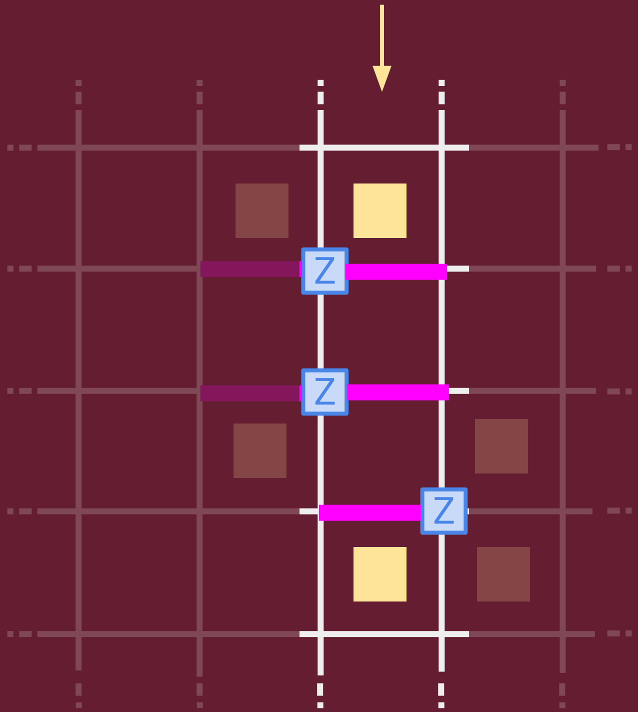
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

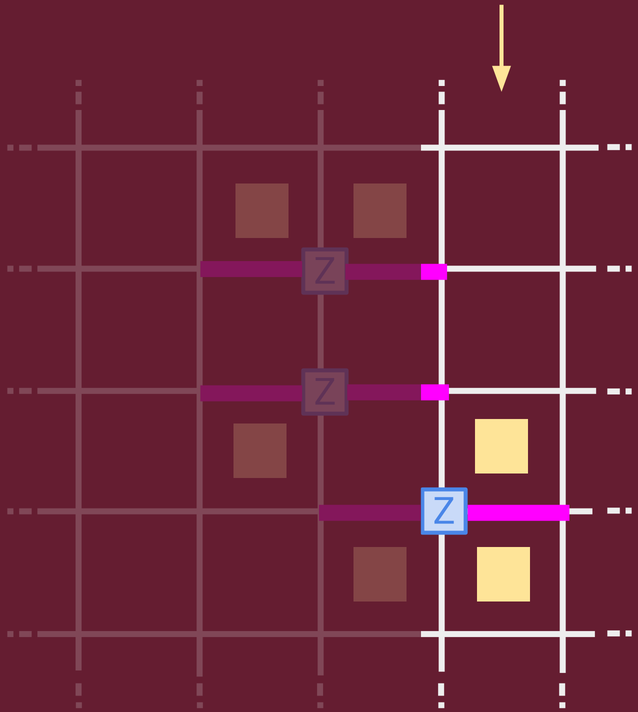
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

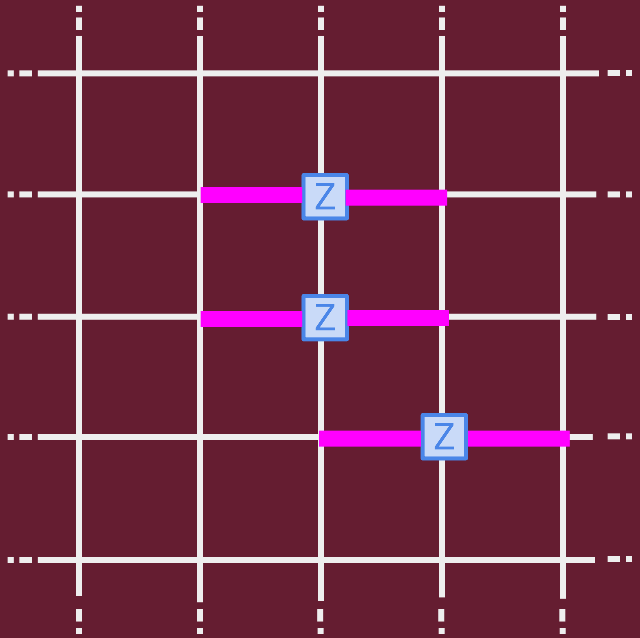
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

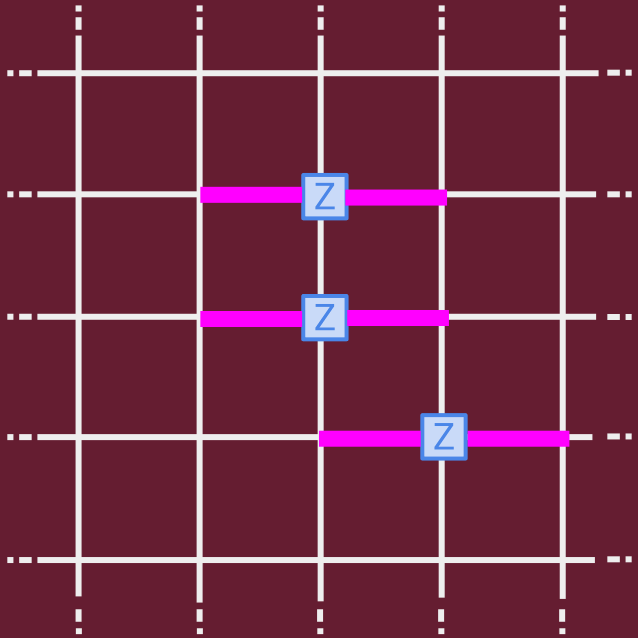
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

Step 2: match along each row

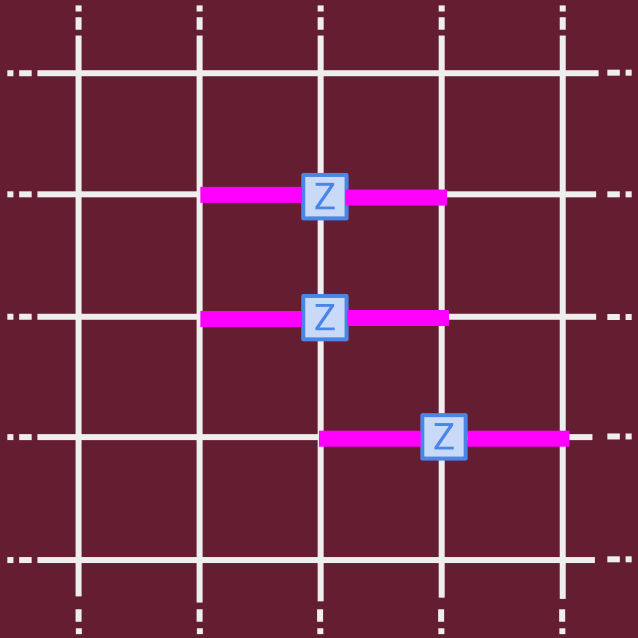
XY surface code

Materialized symmetry:
along every row & column

Conservation law:
Each column has an even number of excitations

High degeneracy on a given column

The parity at each horizontal edge is what matters!



Decoding strategy

Step 1: match along each column and predict the parity of each horizontal edge

Step 2: match along each row

Why 50% threshold?

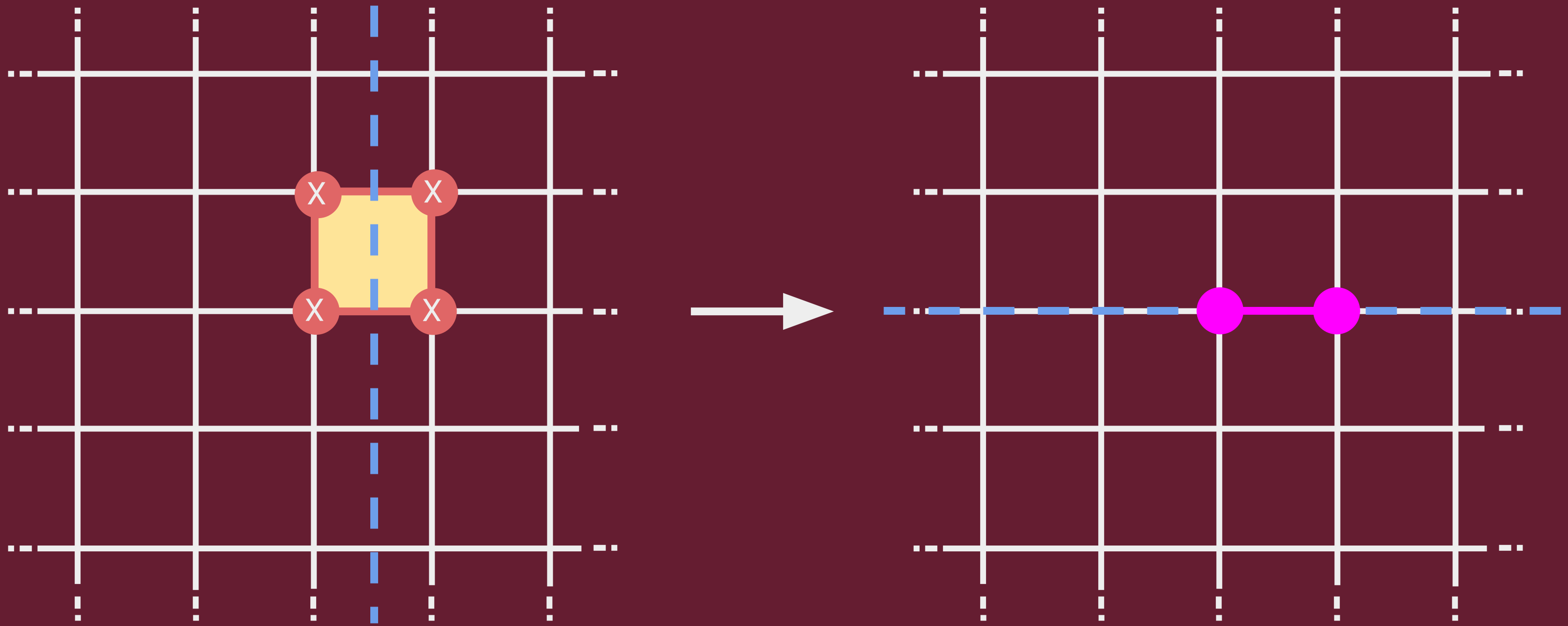
(Show on blackboard)

XY surface code

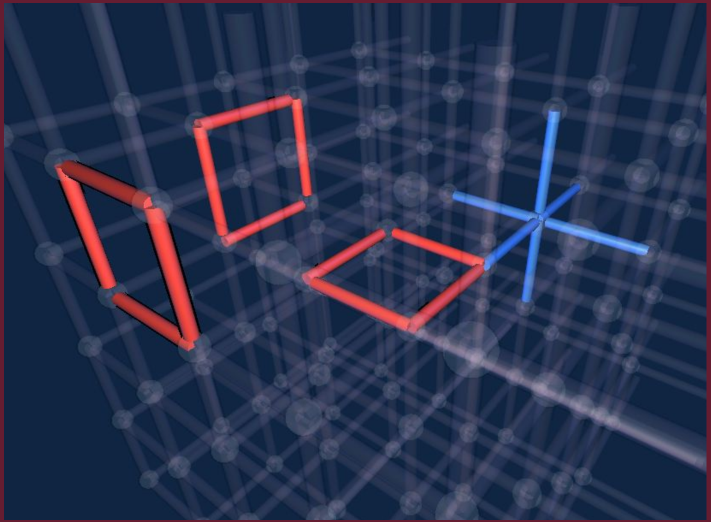
Weight-reduction technique

Step 1: exploit vertical symmetry

Step 2: exploit horizontal symmetry

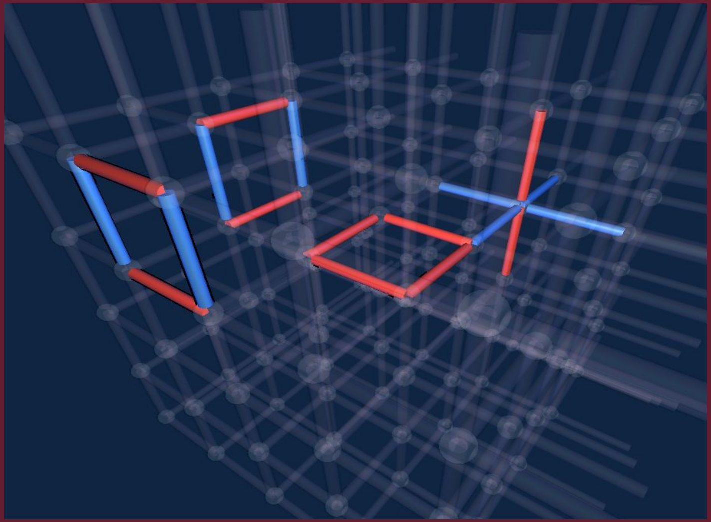


3D toric code



Stabilizers of the CSS toric code

Hadamard on
the vertical axis

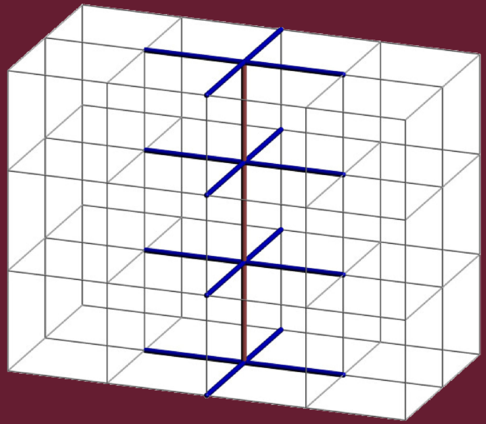


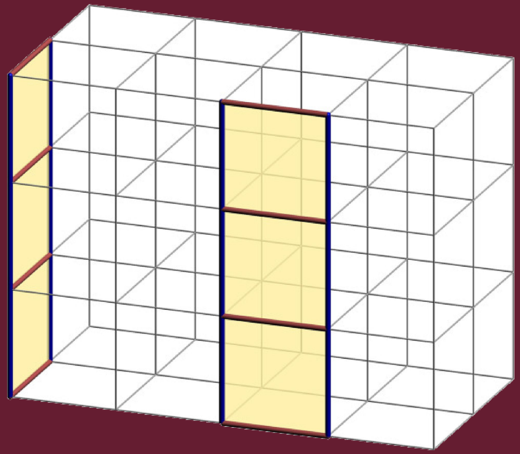
Stabilizers of the deformed toric code

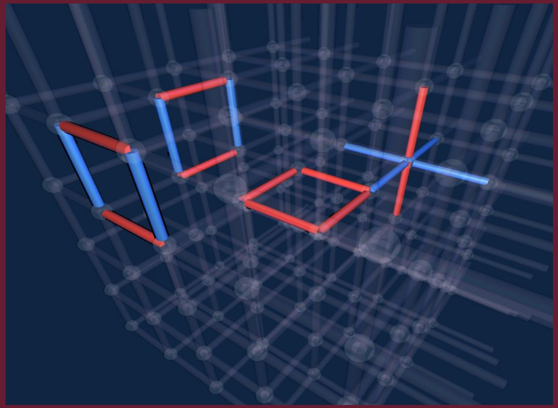
3D toric code

What happens at infinite Z bias?

Linear symmetries for vertex stabilizers and vertical plaquettes





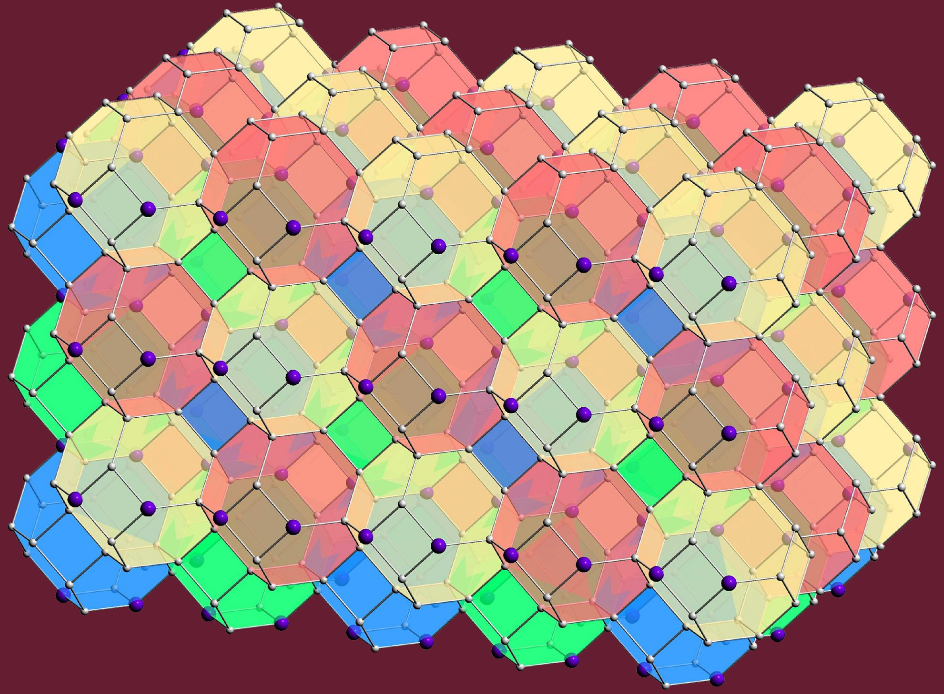


Stabilizers of the deformed toric code

We can decode all the qubits by solving repetition codes along those symmetries

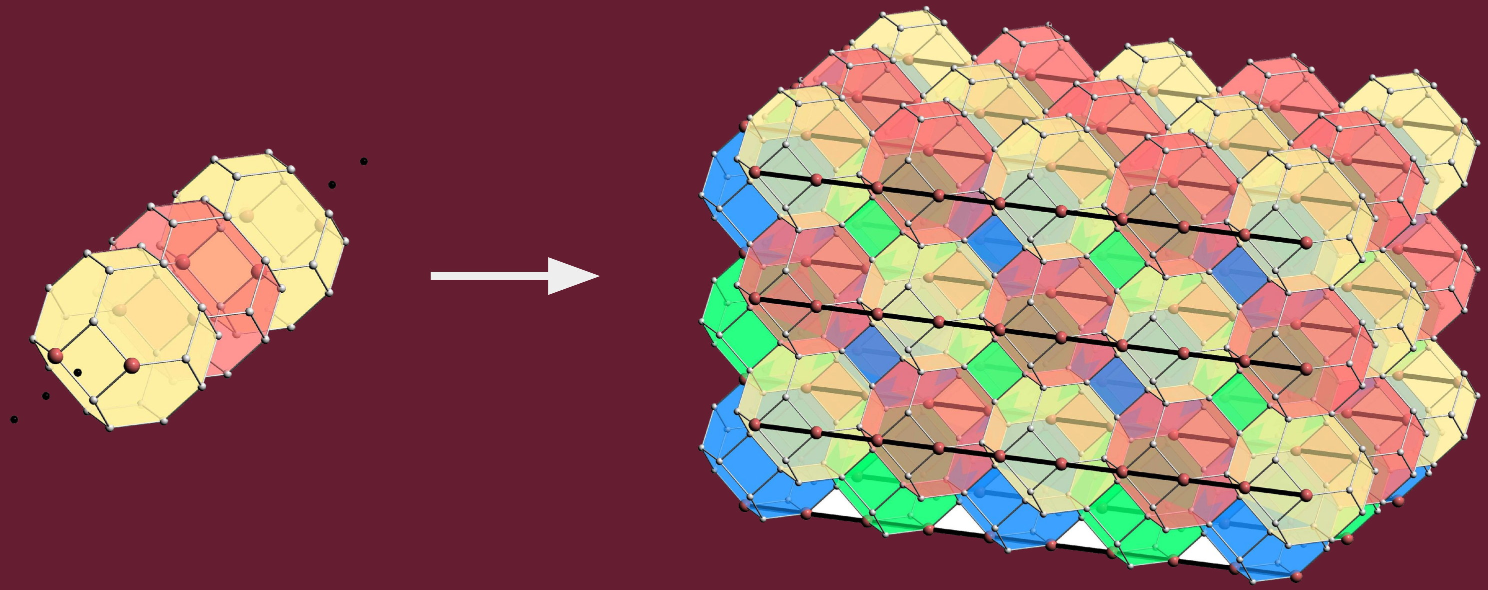
3D color code

Clifford-deformation: Hadamard on each purple vertex

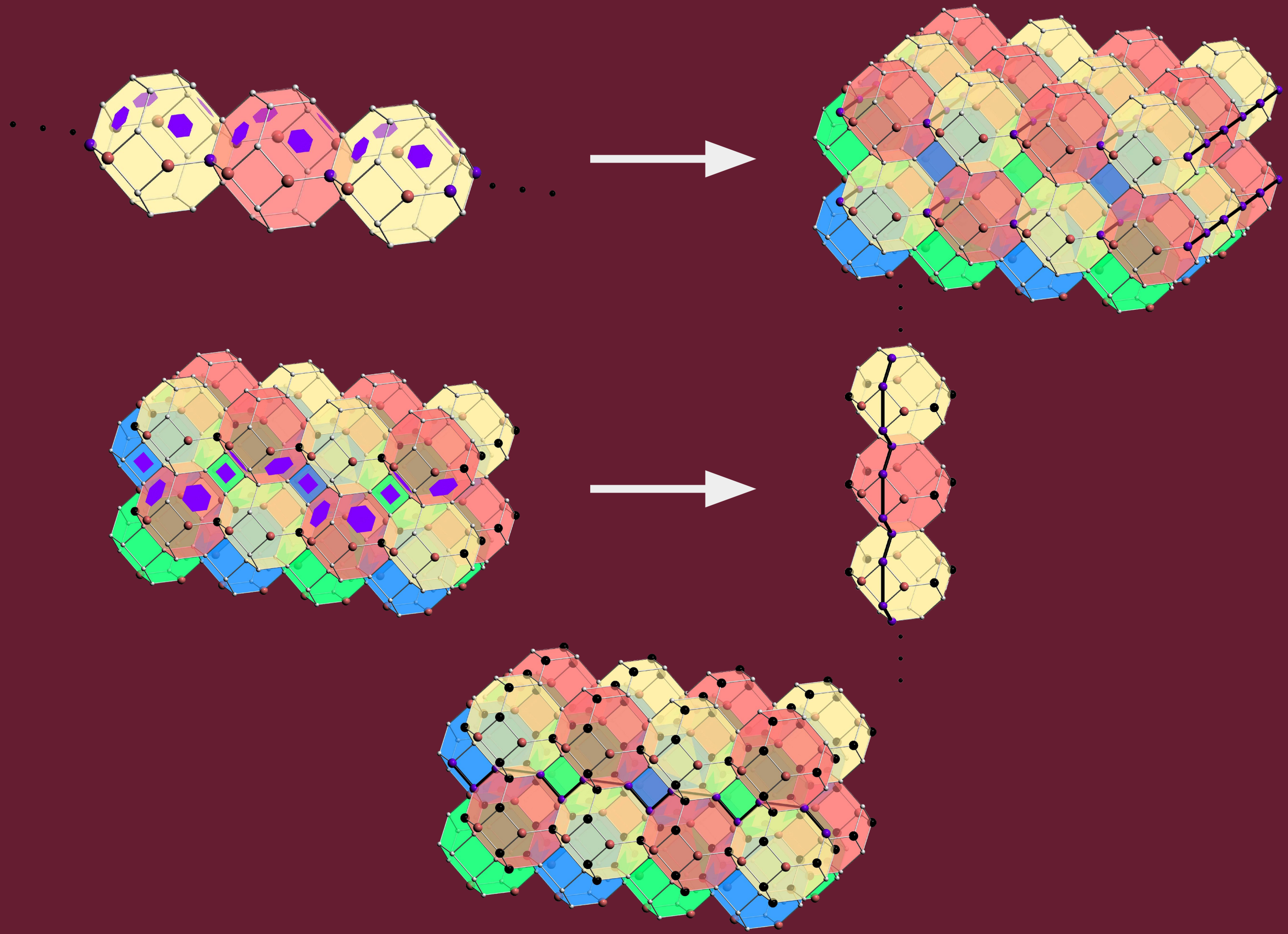


3D color code

Cell decoding: 2-step weight-reduction



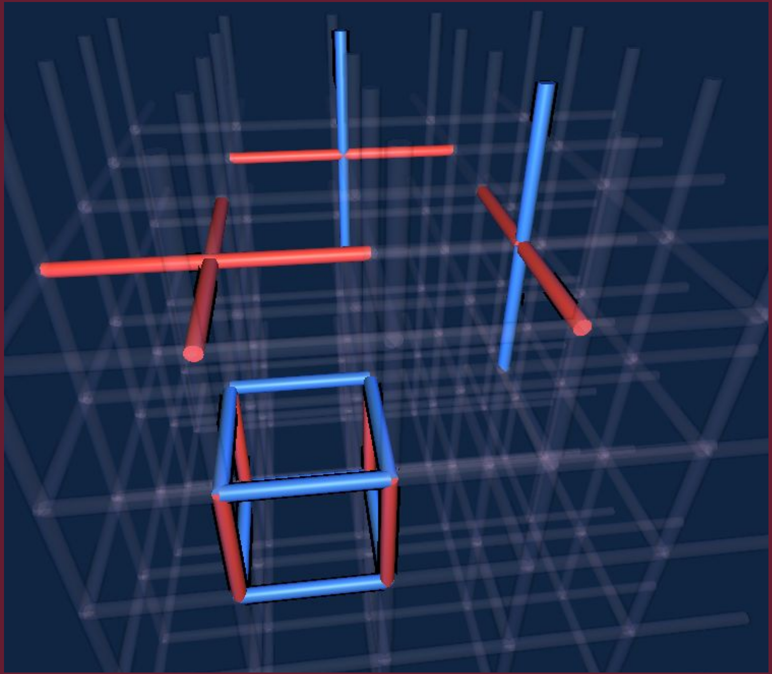
3D color code



Plaquette decoding:
weight-reduction on several subsets of qubits

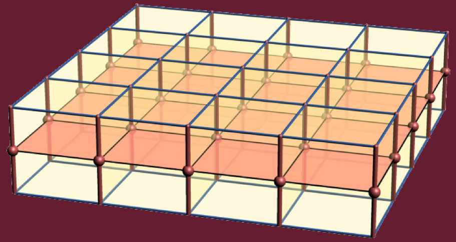
X-cube model

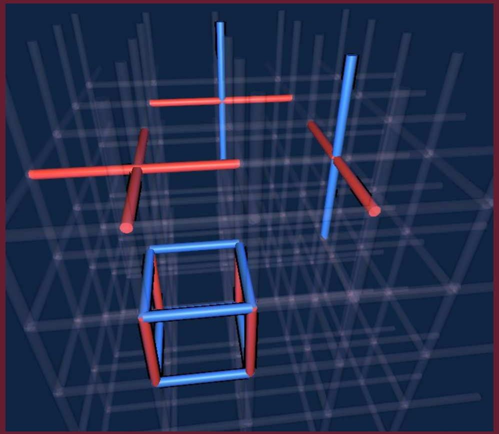
Clifford-deformation: Hadamard on horizontal qubits



X-cube model

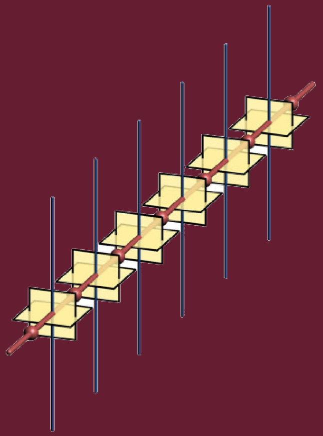
Cube decoding: reduces to an XY surface code on each layer

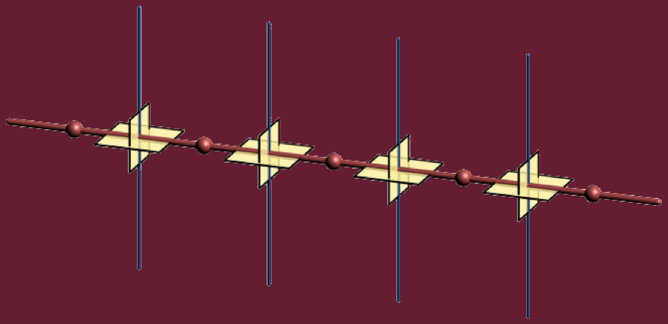




Stabilizers of the deformed X-cube

Vertex decoding: exploit simple linear materialized symmetries

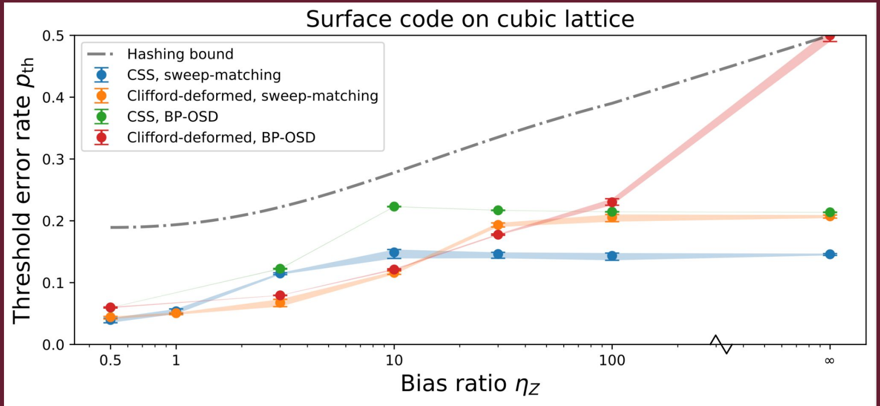


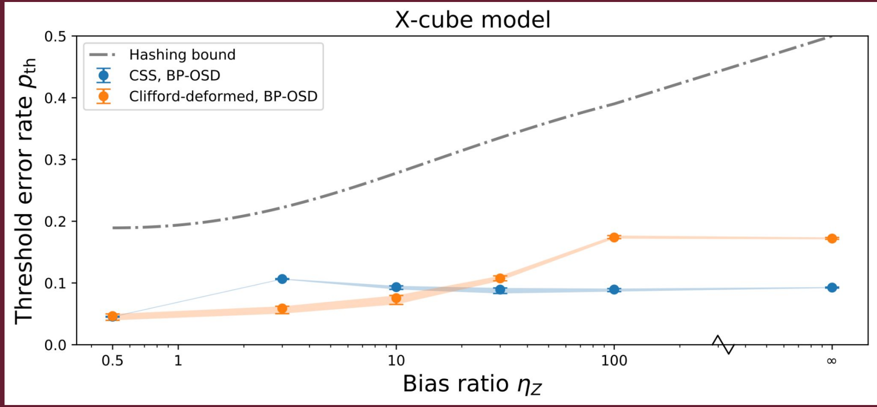


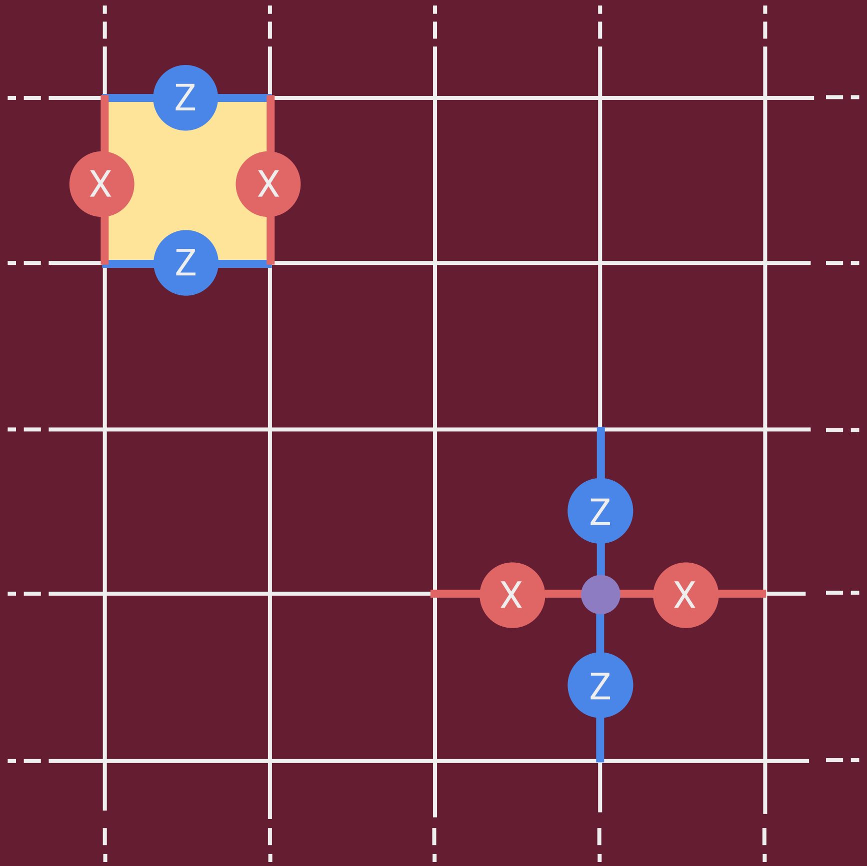
Finite-bias analysis

3D toric code, decoded with Sweep-Matching and BP-OSD (courtesy Joschka for the ldpc library)

X-cube model, decoded with BP-OSD







CODE BOUNDARIES & SUBTHRESHOLD SCALING

Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

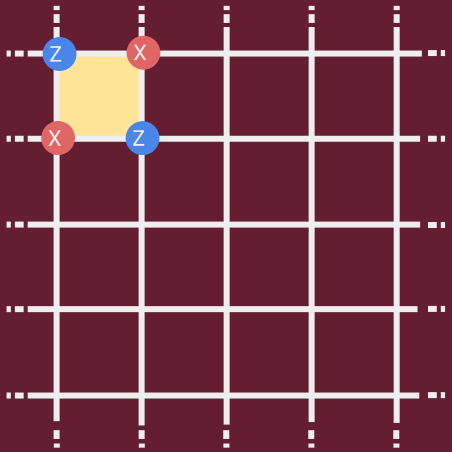
Example: coprime-XZZX

Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

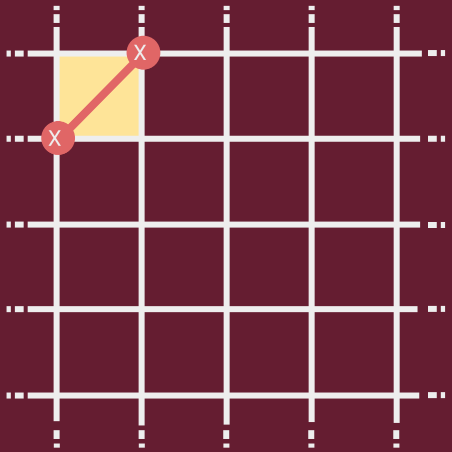


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

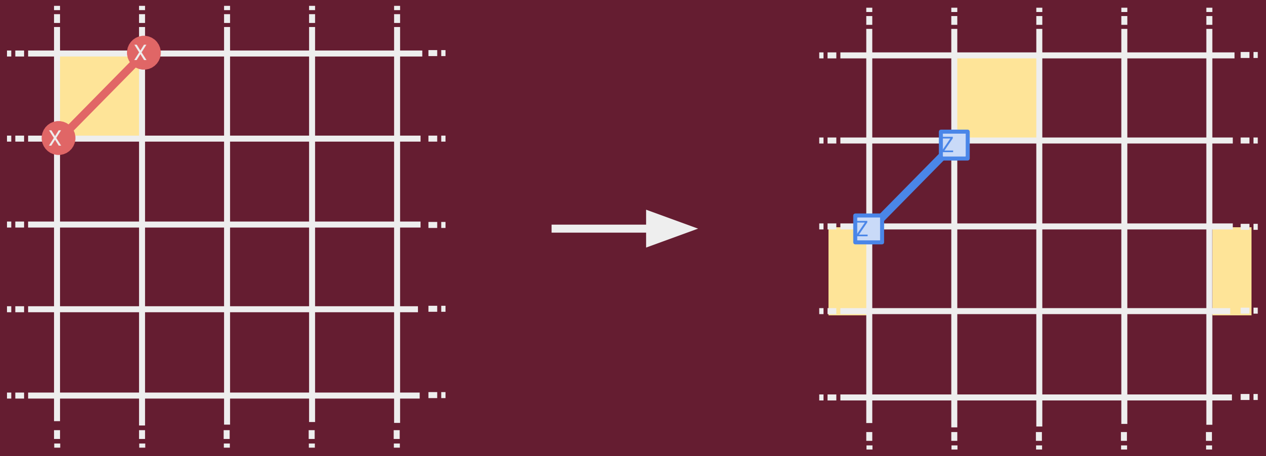


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

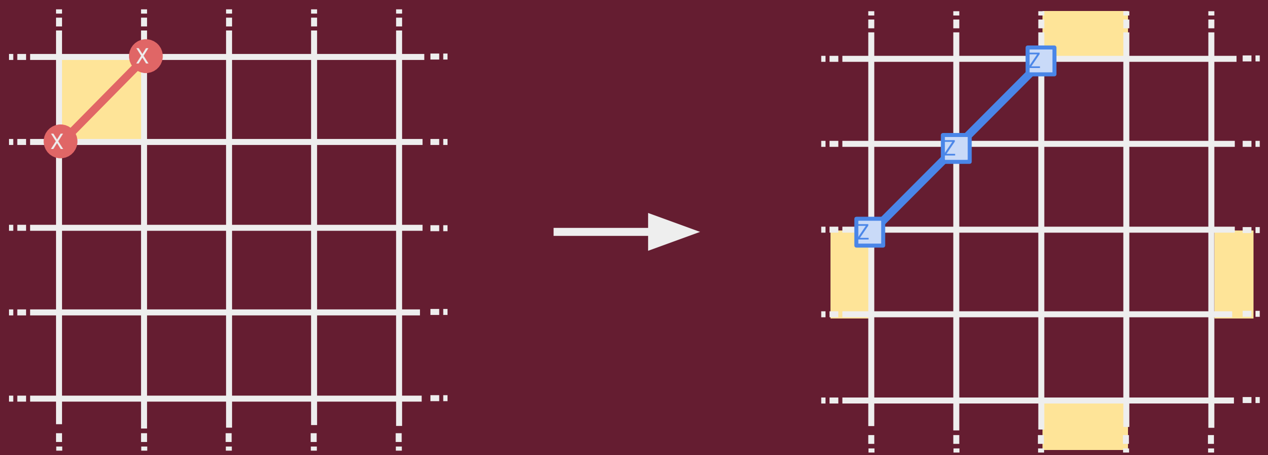


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

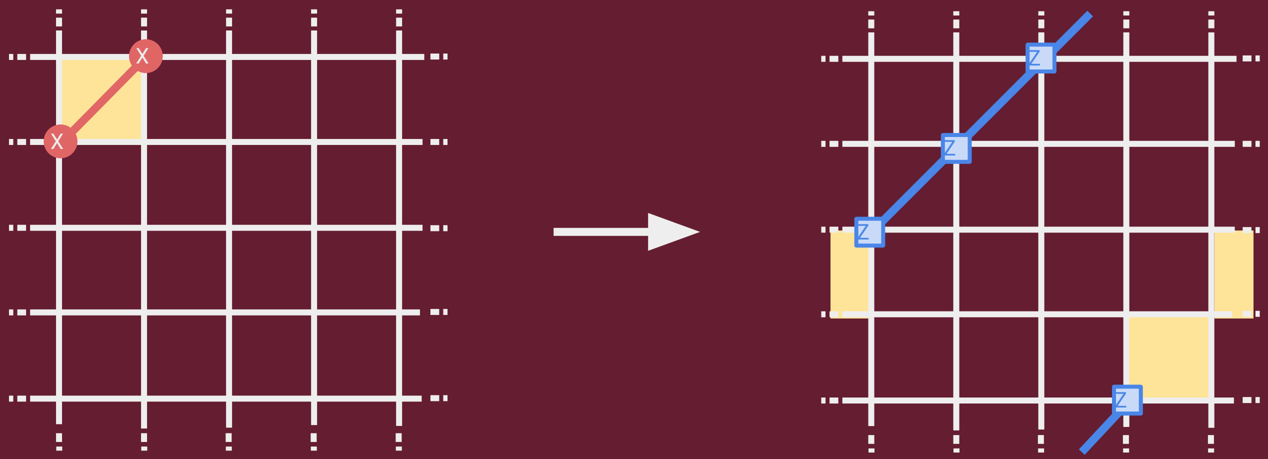


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

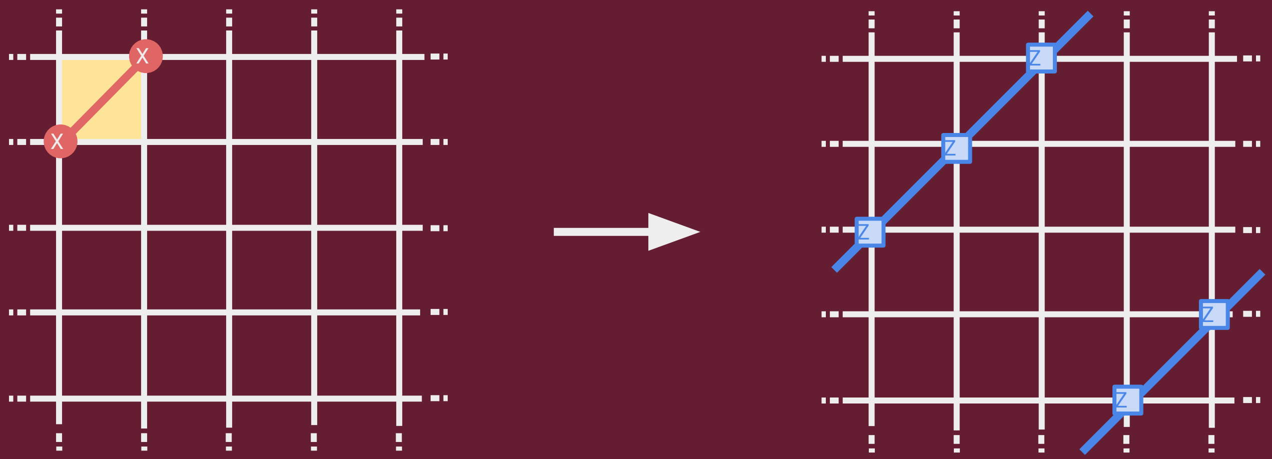


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

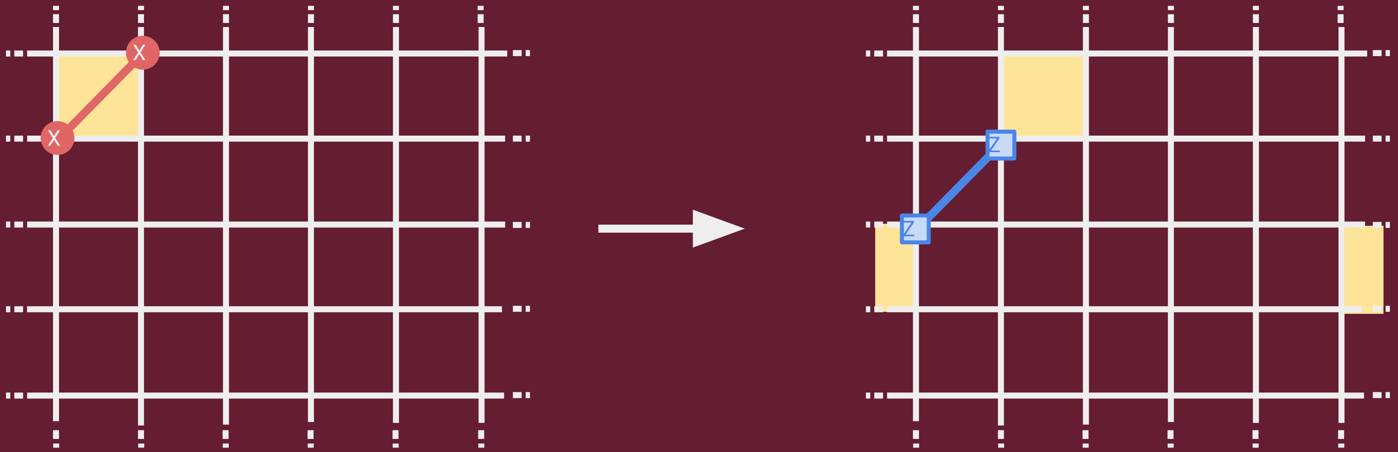


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

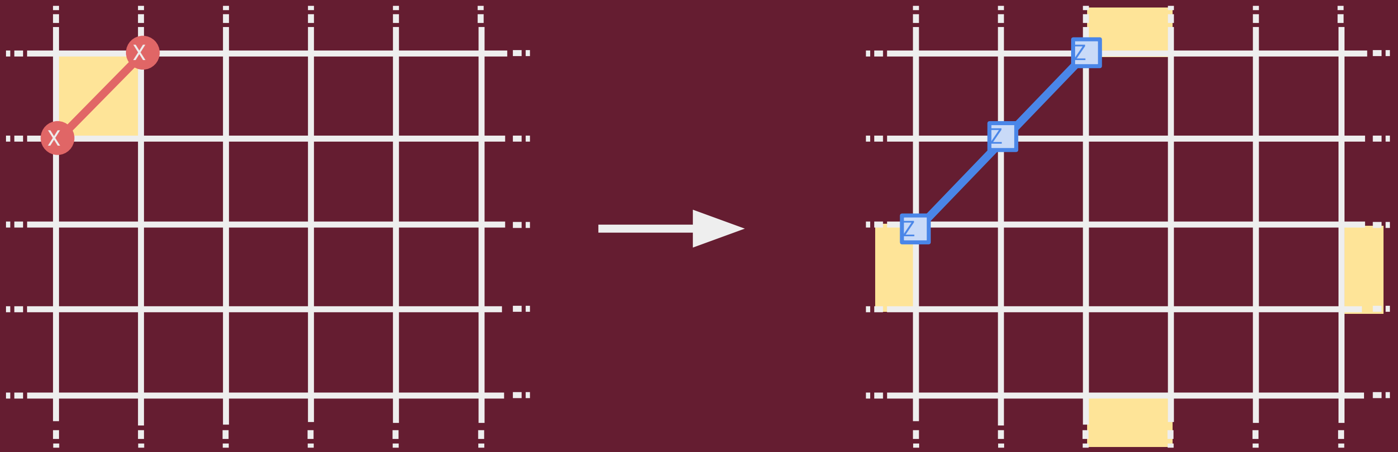


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

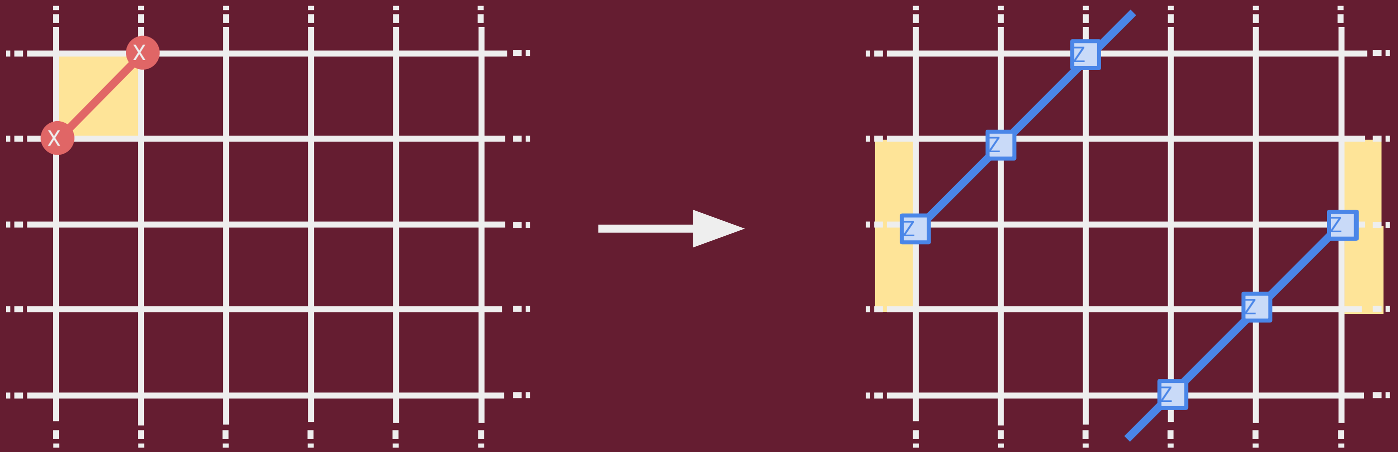


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

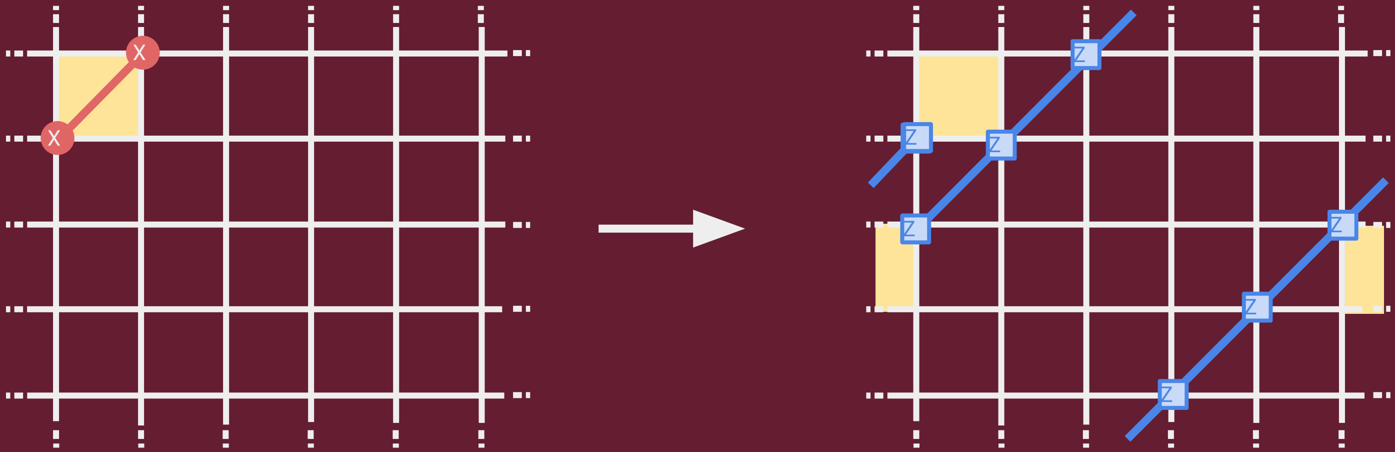


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

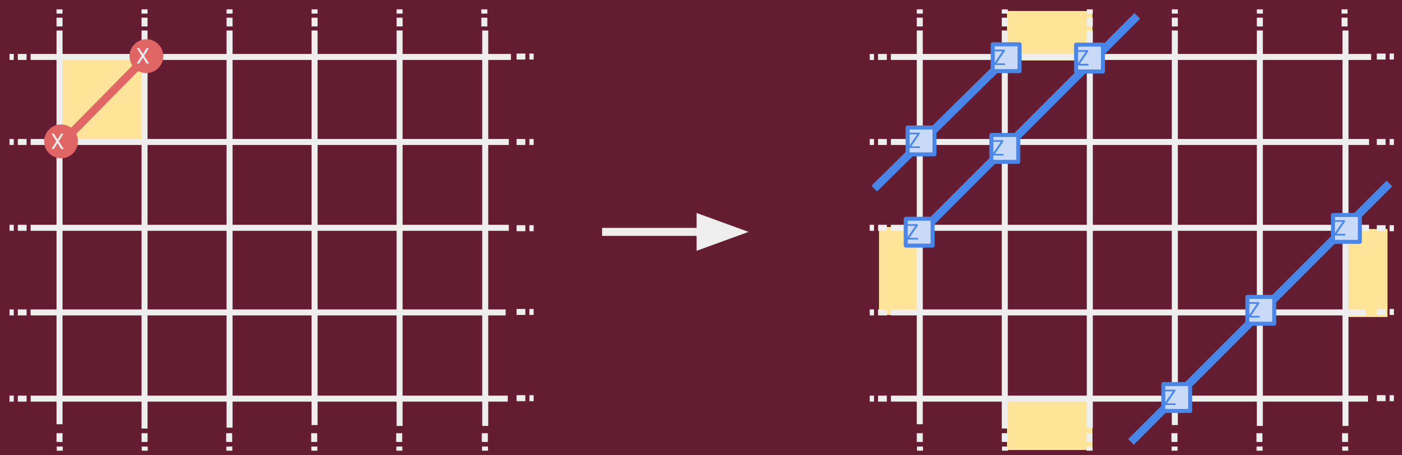


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

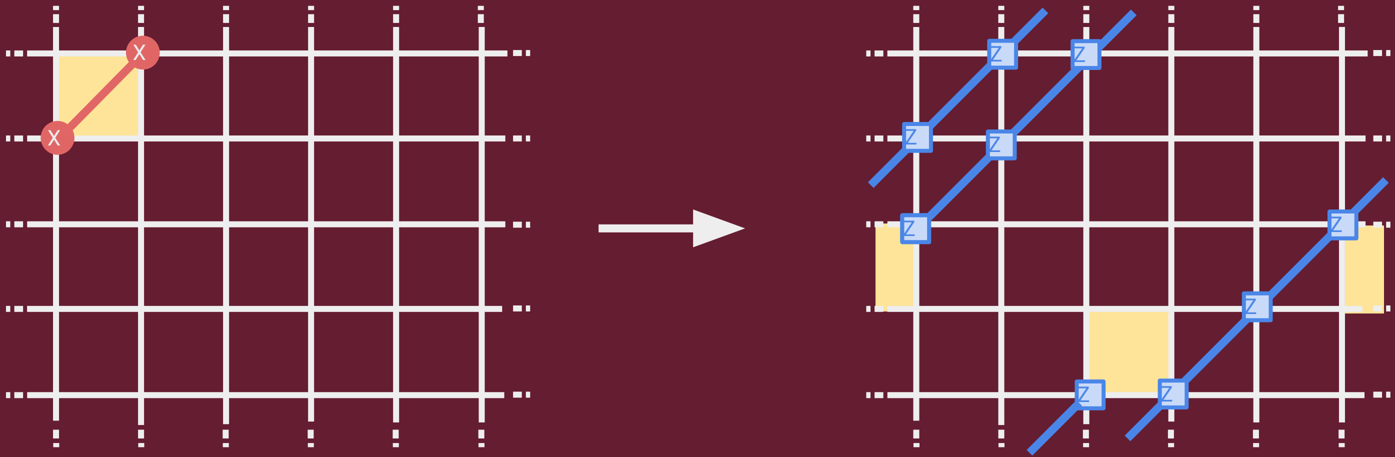


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

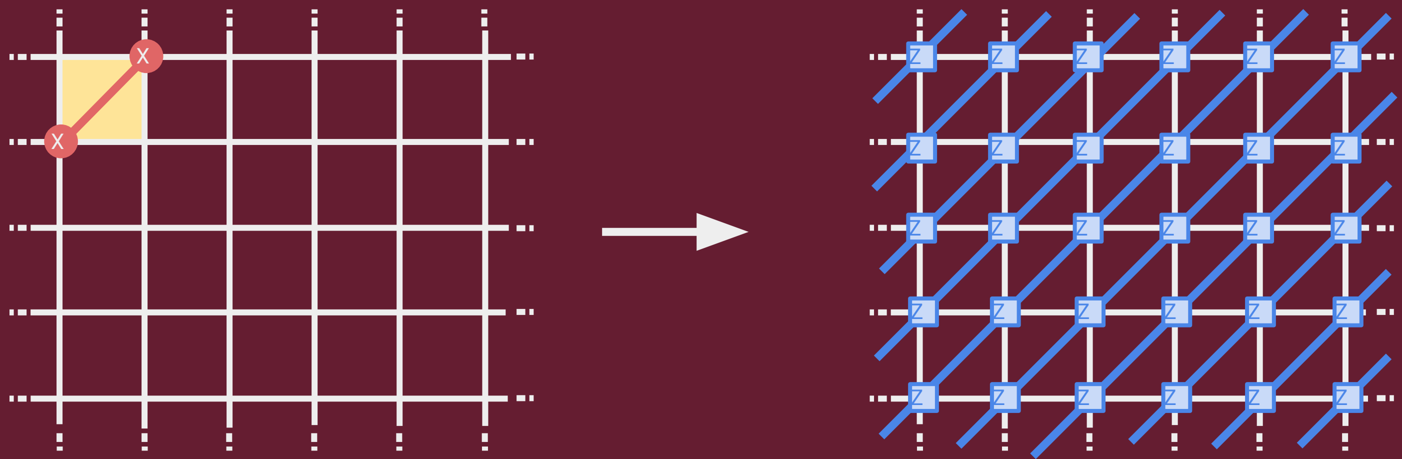


Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX



Pure Z logicals

Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

Consequences: improved subthreshold scaling at infinite bias

$$\bar{p} \propto e^{-\alpha N}$$

Pure Z logicals

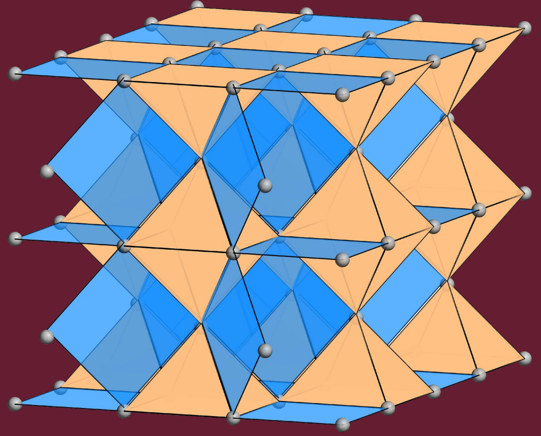
Question: what is the infinite-bias distance of those Clifford-deformed codes?

Answer: it depends on the boundary conditions & lattice dimensions

Example: coprime-XZZX

Consequences: improved subthreshold scaling at infinite bias

$$\overline{p} \propto e^{-\alpha N}$$



Our work: coprime rotated 3D toric code

Pure Z logical supported on O(N) qubits if the lattice has dimensions

$(4n+1) \times (4n+2) \times L_z$ or $(4n+2) \times (4n+3) \times L_z$

Discussion

In conclusion:

1. 3D codes have many useful properties, such as single-shot QEC, transversal T and partial self-correction, but a setting where they are better than 2D codes is yet to be found

2. They naturally improve under biased noise, but for very large bias, we found Clifford-deformation that can push their performance even further

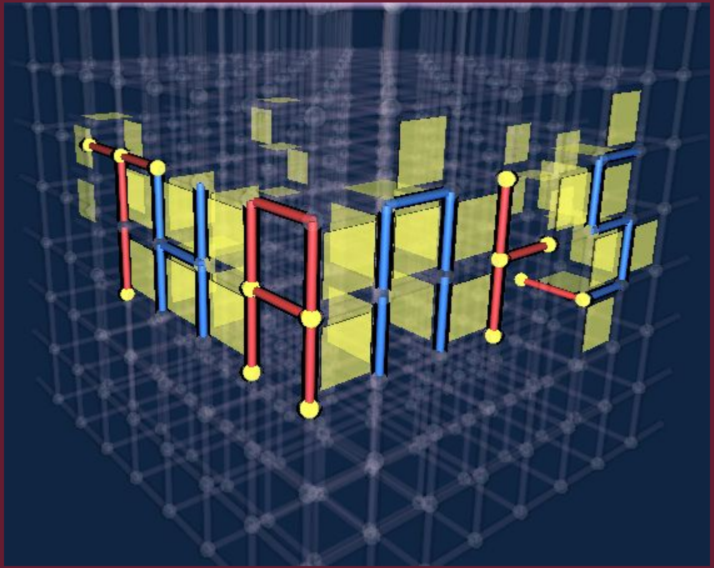
3. Symmetries and weight-reduction can be used to show that Clifford-deformed codes have a 50% threshold

Open questions:

1. All costs taken into account (circuit-level noise, gates, etc.), can 3D codes have an advantage compared to 2D codes under biased noise?

2. Do all stabilizer codes have a deformed version with 50% threshold?

THANKS



3D code visualizer available at: https://gui.quantumcodes.io

PanQEC

BETA

QEC made deliciously easy

github.com/panqec

Interactive tool: gui.quantumcodes.io

PanQEC BETA

Interactive visualization of codes and decoders

- Interactively insert & decode errors on 2D & 3D codes
- Helpful to debug code & decoders
- Useful to test research ideas
- Educational tool to learn QEC

Simple & performant simulator

- Thresholds computable with only a few lines of code
- Tools to submit and track jobs on the cluster
- Analysis and plotting toolbox

Large collection of codes

- Many variants of the 2D and 3D surface & color codes
- Fractons codes
- More codes to come soon… (fermionic codes, hypergraph product codes, etc.)

Interactive tool: `gui.quantumcodes.io`