TOWARDS HIGH-THRESHOLD DECODING OF THE GAUGE COLOR CODE

Will Zeng – Rigetti Computing

Benjamin Brown – Niels Bohr Institute, University of Copenhagen
Fault-tolerant quantum computing with low overhead

Ex. Surface Code (2D):

> 3.3% Threshold (optimal phenomenological noise) [1]
> Non-universal Encoded Gates
> w/ Magic State distillation for T gates

> 4,000 logical qubits for Shor’s factoring algorithm
1 billion physical qubits [2]

94% are for magic state distillation

Fault-tolerant quantum computing with low overhead

Ex. Surface Code (2D):

- 3.3% Threshold (optimal phenomenological noise) [1]
- Non-universal Encoded Gates
  - w/ Magic State distillation for T gates
- 4,000 logical qubits for Shor’s factoring algorithm
- 1 billion physical qubits [2]
- 94% are for magic state distillation

Gauge Color Code (3D):

- Universal Encoded Gates via gauge fixing [3]
- 0.31% Threshold (phenomenological noise) [4]
- Optimal Threshold ???

Fault-tolerant quantum computing with low overhead

Ex. Surface Code (2D):

- > 3.3% Threshold (optimal phenomenological noise) \[1\]
- Non-universal Encoded Gates
  - w/ Magic State distillation for T gates
  - 4,000 logical qubits for Shor's factoring algorithm
    - 94% are for magic state distillation
    - 1 billion physical qubits \[2\]

Present Goal: Push the gauge color code threshold higher with:

(i) A different lattice
(ii) A higher threshold decoder
  (efficient but computationally challenging)

Gauge Color Code (3D):

- Universal Encoded Gates via gauge fixing \[3\]
- > 0.31% Threshold (phenomenological noise) \[4\]
  - Optimal Threshold ???

# GAUGE COLOR CODES


> Four valent, four colorable lattice
> Can be implemented with only weight 4 & 6 check operators

<table>
<thead>
<tr>
<th>Simplex</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-simplex (vertex)</td>
<td>Qubit</td>
</tr>
<tr>
<td>1-simplex (edge)</td>
<td>Qubit coupling</td>
</tr>
<tr>
<td>2-simplex (face)</td>
<td>Gauge operator</td>
</tr>
<tr>
<td>3-simplex (cell)</td>
<td>Stabilizer</td>
</tr>
</tbody>
</table>

[Diagram of bulk lattice and distance 3 (primal)]
ERRORS ON GAUGE COLOR CODES

Distance 3 (primal + dual)
ERRORS ON GAUGE COLOR CODES

Distance 3 (primal + dual)
ERRORS ON GAUGE COLOR CODES

Distance 3 (primal + dual)
ERRORS ON GAUGE COLOR CODES

Distance 3 (primal + dual)

Bulk (primal)
ERRORS ON GAUGE COLOR CODES

Distance 3 (primal + dual)

Bulk (primal)
ERRORS ON GAUGE COLOR CODES

Correction operators are strings

... like in the toric code –

Distance 3 (primal + dual)

Bulk (primal)
Anyons are removed by either:

- Matching to the boundary of their color
- Matching r+b+y+g to the same cell

> 0.31% threshold for phenomenological noise (just Pauli X errors)
> Used an adapted clustering decoder

Q: How to improve on this threshold?  A: (i) a different lattice & (ii) a MCMC decoder
GAUGE COLOR CODE LATTICES

2D Color Codes

3D Color Codes

BNB

Weight 8 & 32 stabilizers
6 & 18 gauges per stabilizer
Weight 4 & 6 gauge operators


Bombin

Weight 24 stabilizers
14 gauges per stabilizer
Weight 4 & 6 gauge operators


GAUGE COLOR CODE LATTICES

2D Color Codes

3D Color Codes

Weight 8 & 32 stabilizers
6 & 18 gauges per stabilizer
Weight 4 & 6 gauge operators


Weight 24 stabilizers
14 gauges per stabilizer
Weight 4 & 6 gauge operators


We choose this lattice
MARKOV CHAIN MONTE CARLO DECODING

> MCMC decoders for the surface code:

  • Their MCMC decoder achieves surface code threshold of 18.5%
    (upper bound is 18.9%; Masayuki. Phys. Rev. A 85.6 (2012): 060301.)
Apply Errors → Measure Stabilizer Syndrome → Use virtual anyons to complete color parity

Match to center: $c_0$

Apply to $c_i$ a random gauge operator: $c'$

$c_{i+1} = c'$

If length $c' < c_i$

If $T(c_i, c')$

$c_{i+1} = c_i$

> Run for chainlength = $\alpha \times d^6$ steps, where $d$ is the distance of the code and $\alpha$ is a constant.
Apply Errors → Measure Stabilizer Syndrome → Use virtual anyons to complete color parity

Match to center: $c_0$

Apply to $c_i$ a random gauge operator: $c'$

If $c_{i+1} = c'$

If length $c' < c_i$

If $T(c_i, c')$
Apply Errors → Measure Stabilizer Syndrome → Use virtual anyons to complete color parity

Match to center: $c_0$

Apply to $c_i$ a random gauge operator: $c'$

$c_{i+1} = c'$

If length $c' < c_i$

$c_{i+1} = c_i$

If $T(c_i, c')$

Run for chainlength $= \alpha \times d^6$ steps, where $d$ is the distance of the code and $\alpha$ is a constant.
MARKOV CHAIN MONTE CARLO DECODING

Apply Errors → Measure Stabilizer Syndrome → Use virtual anyons to complete color parity

1. Match to center: \( c_0 \)
2. Apply to \( c_i \) a random gauge operator: \( c' \)
3. \( c_{i+1} = c' \) if \( c_{i+1} = c_i \)
4. \( c_{i+1} = c_i \) if length \( c' < c_i \)
5. \( c_{i+1} = c_i \) if \( T(c_i, c') \)

> Run for chainlength = \( \alpha \times d^6 \) steps, where \( d \) is the distance of the code and \( \alpha \) is a constant.
MARKOV CHAIN MONTE CARLO DECODING

- Apply Errors
- Measure Stabilizer Syndrome
- Use virtual anyons to complete color parity

1. Match to center: $c_0$
2. Apply to $c_i$ a random gauge operator: $c'$
3. If length $c' < c_i$:
   - If $T(c_i, c')$ is True, $c_{i+1} = c'$
   - $c_{i+1} = c_i$

> Run for chainlength = $\alpha \times d^6$ steps, where $d$ is the distance of the code and $\alpha$ is a constant.
MARKOV CHAIN MONTE CARLO DECODING

1. Apply Errors
2. Measure Stabilizer Syndrome
3. Use virtual anyons to complete color parity

- Match to center: $c_0$
- Apply to $c_i$ a random gauge operator: $c'$

- If $c_{i+1} = c'$, go to step 3
- If $c_{i+1} = c_i$, go to step 3
- If length $c' < c_i$, go to step 3
- If $T(c_i, c')$, go to step 3

> Run for chainlength $= \alpha \times d^6$ steps, where $d$ is the distance of the code and $\alpha$ is a constant.
MARKOV CHAIN MONTE CARLO DECODING

Apply Errors → Measure Stabilizer Syndrome → Use virtual anyons to complete color parity

- Match to center: \( c_0 \)
- Apply to \( c_i \) a random gauge operator: \( c' \)

- If length \( c' < c_i \):
  - \( c_{i+1} = c' \)
- If \( T(c_i, c') \):
  - \( c_{i+1} = c_i \)
- If \( c_{i+1} = c' \):
  - \( \text{True} \)

> Run for chainlength = \( \alpha \times d^6 \) steps, where \( d \) is the distance of the code and \( \alpha \) is a constant.
MARKOV CHAIN MONTE CARLO DECODING

> Run for chainlength = $\alpha \times d^6$ steps, where $d$ is the distance of the code and $\alpha$ is a constant.
MARKOV CHAIN MONTE CARLO DECODING

Apply Errors → Measure Stabilizer Syndrome → Use virtual anyons to complete color parity

- Match to center: $c_0$
  - Apply to $c_i$ a random gauge operator: $c'$
    - If $c' < c_i$
      - If $T(c_i, c')$
        - $c_{i+1} = c'$
      - $c_{i+1} = c_i$
    - $c_{i+1} = c'$

> Run for chainlength = $\alpha \times d^6$ steps, where $d$ is the distance of the code and $\alpha$ is a constant.
LOOKING FOR A THRESHOLD WITH MCMC DECODING

> Perfect Measurements; X errors
> Compare to 0.45% from Brown et al. arXiv:1503.08217 (2015)
> $L(c) =$ weight of correction $c$

\[
\text{ChainLength} = 10 \times d^6
\]

\[
T(c_i, c') = 100 \left(1 - \frac{p}{3} \right)^{L(c_i) - L(c')}
\]

20k samples per point (except $d=11$ at 2k)
LOOKING FOR A THRESHOLD WITH MCMC DECODING

- Perfect Measurements; X errors
- $L(c) =$ weight of correction $c$

\[
\text{ChainLength} = 10 \times d^6
\]

\[
T(c_i, c') = 100 \left( \frac{1 - \frac{p}{3}}{p} \right)^{L(c_i) - L(c')}
\]

\[
\text{ChainLength} = 100 \times d^6
\]

\[
T(c_i, c') = \frac{1}{p} \left( \frac{1 - \frac{p}{3}}{p} \right)^{L(c_i) - L(c')}
\]

![Graph showing the relationship between physical error rate and threshold with different decoding distances](image)

20k samples per point (except $d=11$ at 2k)
Evidence towards a GCC threshold > 1.2%*

* perfect measurements

CONCLUSION

FUTURE WORK

1) More evidence:
   - Markov chain parallelization: $O(L^4) \rightarrow O(L^2)$ in 2D case
   - Larger lattices ($d \approx 41$)
   - Is this really efficient?
   - What are the optimal parameters?

2) New error models:
   - Single-shot decoding makes GCC able to easily detect measurement errors
APPENDIX
LOOKING FOR A THRESHOLD WITH MCMC DECODING

\[ \text{ChainLength} = \alpha \times d^6 \]

\[ T(c_i, c') = 100 \left( \frac{1 - \frac{p}{3}}{p} \right)^{L(c_i) - L(c')} \]

Logical accuracy vs. chainlength at different physical error rates

\[ \text{ChainLength} = 30 \times d^6 \]

\[ T(c_i, c') = \gamma \left( \frac{1 - \frac{p}{3}}{p} \right)^{L(c_i) - L(c')} \]

Logical accuracy vs. pseudo-temperature at different physical error rates
Gauge Color Codes


- A topological quantum error correcting code (3D)
- Four valent, four colorable lattice
- Admits universal transversal encoded gates via gauge fixing
- Can be implemented with only weight 4 & 6 check operators

Distance 3 (primal + dual)

Distance 15 (dual):
- 671 qubits
LARGER BOMBIN LATTICES

Distance 5 (primal)

Distance 5 (dual)

Distance 7 (dual)

Distance 15 (dual)

65 qubits

175 qubits

671 qubits