Multi-path Summation for Decoding 2D Topological Codes

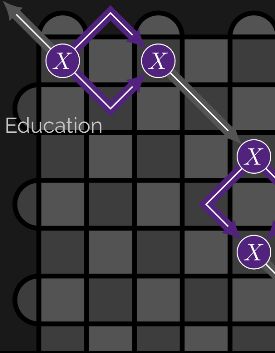
Quantum 2, 102 (2018) $\vee$ arXiv:1709.02154v5

Ben Criger[1,2] Imran Ashraf[1]

[1]Qutech, TU Delft

[2]Institute for Globally Distributed Open Research and Education (`igdore.org`)

November 30, 2018



Outline

- Quantum Computing ∩ Social & Economic Justice

Outline

- Quantum Computing $\cap$ Social & Economic Justice
- The Surface Code & Decoding

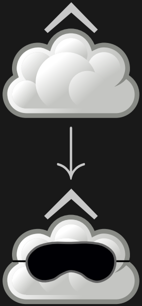
Outline

- Quantum Computing ∩ Social & Economic Justice
- The Surface Code & Decoding
- Counting and Summing Paths

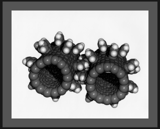
Outline

- Quantum Computing ∩ Social & Economic Justice
- The Surface Code & Decoding
- Counting and Summing Paths
- Results

Quantum Computing: The Good …



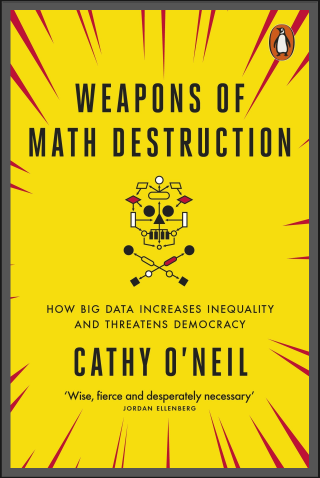
- quantum networks allow private communication
- 'blind' quantum computation allows private processing



- simulating quantum physics allows 'quantum CAD'

The Bad ...

- people are impressed by algorithms
- they trust algorithms to be objective and correct
- important decisions are being made with unverified machine learning
- quantum computers: more impressive, but less verifiable?



and the NISQ

and the NISQ

- analog quantum simulation

and the NISQ

- analog quantum simulation
  - easy to make logical errors

and the NISQ

- analog quantum simulation
    - easy to make logical errors

- variational quantum eigensolving

and the NISQ

- analog quantum simulation
    - easy to make logical errors
- variational quantum eigensolving
    - inherently heuristic method

and the NISQ

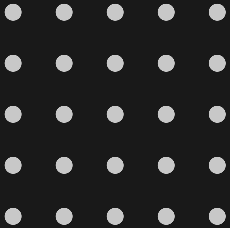
- analog quantum simulation
  - easy to make logical errors
- variational quantum eigensolving
  - inherently heuristic method
- low-depth circuits for producing random numbers

and the NISQ

- analog quantum simulation
  - easy to make logical errors
- variational quantum eigensolving
  - inherently heuristic method
- low-depth circuits for producing random numbers
  - single niche application = “supremacy”?

The Surface Code

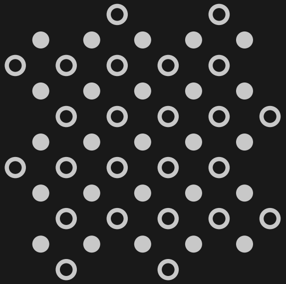
For now, large-scale quantum computation needs surface codes:



- lay data qubits out in a square grid

The Surface Code

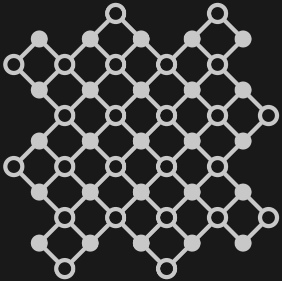
For now, large-scale quantum computation needs surface codes:



- lay data qubits out in a square grid
- add local ancilla qubits

The Surface Code

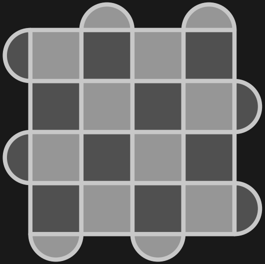
For now, large-scale quantum computation needs surface codes:



- lay data qubits out in a square grid
- add local ancilla qubits
- connect each ancilla to a small number of data qubits using constant-length planar links

The Surface Code

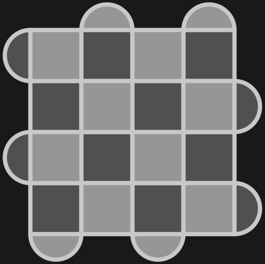
For now, large-scale quantum computation needs surface codes:



- lay data qubits out in a square grid
- add local ancilla qubits
- connect each ancilla to a small number of data qubits using constant-length planar links
- measure *stabilisers* indirectly using H/CNOT gates

The Surface Code

For now, large-scale quantum computation needs surface codes:



- : weight-4 X check
- : weight-4 Z check
- Weight-2 boundary checks: ,
- logical Z operator runs from left to right, logical X from top to bottom
- $[\![d^2, 1, d]\!]$ code

Disclaimer

Disclaimer

Reality: Stabilisers must be measured repeatedly, and results must be compared to each other to determine where measurement errors have occurred.

Disclaimer

Reality: Stabilisers must be measured repeatedly, and results must be compared to each other to determine where measurement errors have occurred.

This Talk: Stabiliser measurements will always return the right answer, and not cause errors themselves.

Disclaimer

Reality: Stabilisers must be measured repeatedly, and results must be compared to each other to determine where measurement errors have occurred.

This Talk: Stabiliser measurements will always return the right answer, and not cause errors themselves.

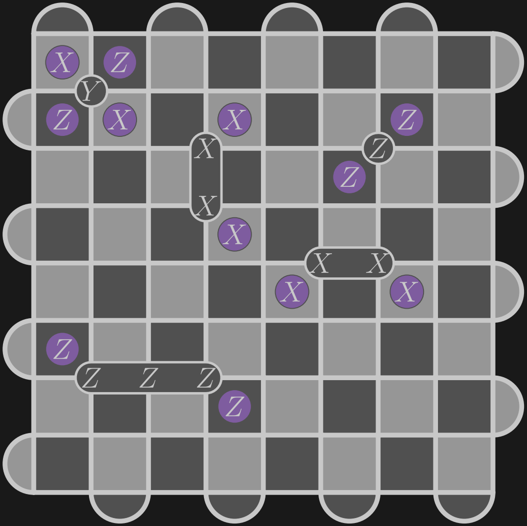
Justification: We'll still be able to observe the effects of different decoding algorithms in this simpler scenario.

Decoding Algorithms

stabiliser measurements don't tell you exactly which error has occurred:

Decoding Algorithms

stabiliser measurements don't tell you exactly which error has occurred:



Decoding Algorithms

assuming:

Decoding Algorithms

assuming:

- errors occur with the same probability on each qubit

Decoding Algorithms

assuming:

- errors occur with the same probability on each qubit
- X and Z errors are i.i.d. with probability p, $\therefore Y$ errors occur with probability p^2

Decoding Algorithms

assuming:

- errors occur with the same probability on each qubit
- X and Z errors are i.i.d. with probability p, $\therefore Y$ errors occur with probability p^2
- if two syndromes are connected by *any* chain of errors, they are connected by *a prescribed minimum-length* chain of errors

Decoding Algorithms

assuming:

- errors occur with the same probability on each qubit
- X and Z errors are i.i.d. with probability p, $\therefore Y$ errors occur with probability p^2
- if two syndromes are connected by *any* chain of errors, they are connected by *a prescribed minimum-length* chain of errors

"we" (Dennis/Kitaev/Landahl/Preskill '01) can do a derivation …

Decoding Algorithms

Decoding Algorithms

$$p(E) = p^{w(E)}(1-p)^{n-w(E)} = (1-p)^n \left(\frac{p}{1-p}\right)^{w(E)}$$

Decoding Algorithms

$$p(E) = p^{w(E)}(1-p)^{n-w(E)} = (1-p)^n \left(\frac{p}{1-p}\right)^{w(E)}$$

$$p(E) = (1-p)^n \prod_{e \in \mathrm{edges}(E)} \left(\frac{p}{1-p}\right)^{|e|}$$

Decoding Algorithms

$$p(E) = p^{w(E)}(1-p)^{n-w(E)} = (1-p)^n \left(\frac{p}{1-p}\right)^{w(E)}$$

$$p(E) = (1-p)^n \prod_{e \in \text{edges}(E)} \left(\frac{p}{1-p}\right)^{|e|}$$

$$p(E) \propto \prod_{e \in \text{edges}(E)} \left(\frac{p}{1-p}\right)^{|e|} \sim \sum_{e \in \text{edges}(E)} |e| \ln\left(\frac{p}{1-p}\right)$$

Decoding Algorithms

$$p(E) = p^{w(E)}(1-p)^{n-w(E)} = (1-p)^n \left(\frac{p}{1-p}\right)^{w(E)}$$

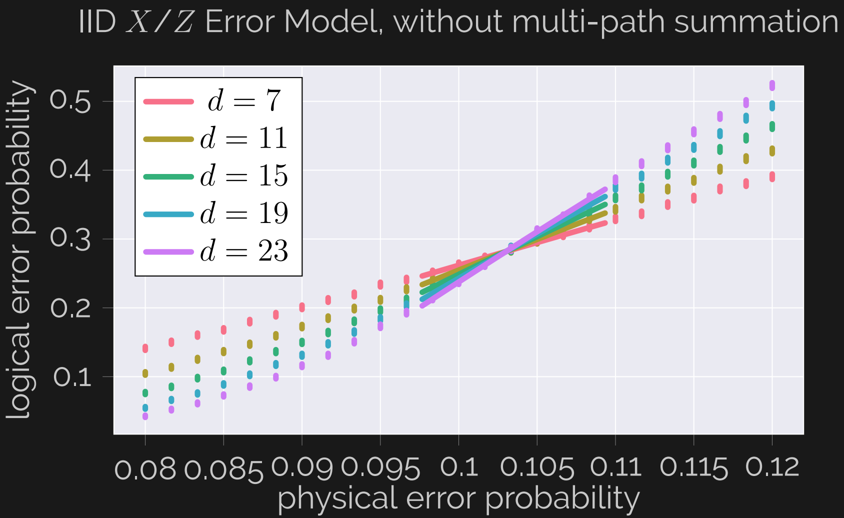
$$p(E) = (1-p)^n \prod_{e \in \mathrm{edges}(E)} \left(\frac{p}{1-p}\right)^{|e|}$$

$$p(E) \propto \prod_{e \in \mathrm{edges}(E)} \left(\frac{p}{1-p}\right)^{|e|} \sim \sum_{e \in \mathrm{edges}(E)} |e| \ln\left(\frac{p}{1-p}\right)$$

$$\propto -\sum_{e \in \mathrm{edges}(E)} |e|$$

Decoding Algorithms

this results in a threshold error rate of 10.3%:

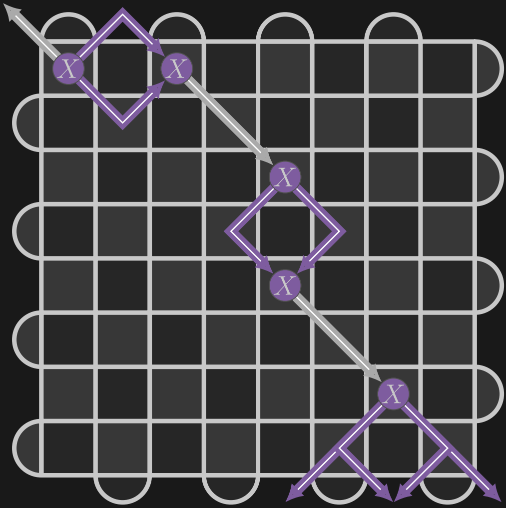


Performance

optimal threshold against this error model near 10.9%

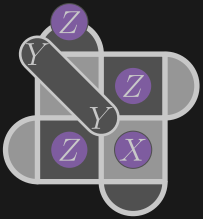
Performance

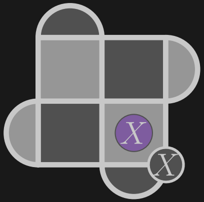
optimal threshold against this error model near 10.9%

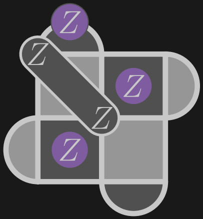


Performance

optimal threshold against this error model near 10.9%

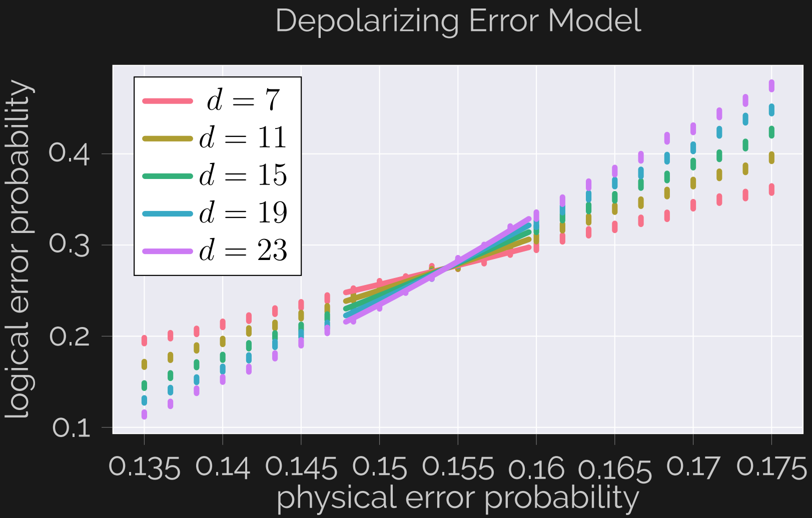






Decoding Algorithms

threshold vs. depolarising noise: 15.4% (optimal: 18.9%)



Multi-Path Summation

- imagine that we have different probabilities of error on each qubit

Multi-Path Summation

- imagine that we have different probabilities of error on each qubit

- we only consider minimum-length paths

Multi-Path Summation

- imagine that we have different probabilities of error on each qubit
- we only consider minimum-length paths
- we consider two possibilities, either the vertices being considered are joined by a single path, or none of the relevant qubits are in error

Multi-Path Summation

maximum-likelihood derivation from earlier is now uglier, but still tractable

Multi-Path Summation

maximum-likelihood derivation from earlier is now uglier, but still tractable

set $o_q \equiv \frac{p_q}{1-p_q}$

Multi-Path Summation

maximum-likelihood derivation from earlier is now uglier, but still tractable

set $o_q \equiv \frac{p_q}{1-p_q}$

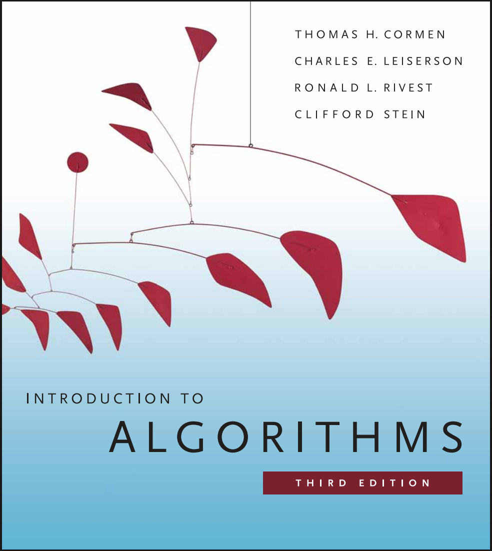
$$p(E) = \prod_q (1 - p_q) \times \prod_{e \in \text{edges}(E)} \sum_{p \in \text{paths}(e)} \prod_{q \in p} o_q$$
$$\sim \sum_{e \in \text{edges}(E)} \log \left(\sum_{p \in \text{paths}(e)} \prod_{q \in p} o_q \right)$$

Multi-Path Summation

let's 'borrow' a technique from CS:

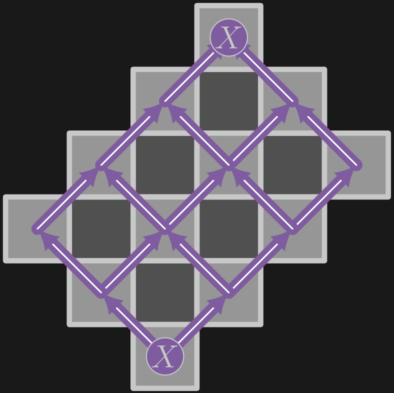
Multi-Path Summation

let's 'borrow' a technique from CS:



Multi-Path Summation

let's 'borrow' a technique from CS:



Belief Propagation

approximate marginal probabilities (Poulin/Chung '08)

Belief Propagation

approximate marginal probabilities (Poulin/Chung '08)

processors associated with ancillas calculate messages to be passed to neighbouring processors associated with data qubits

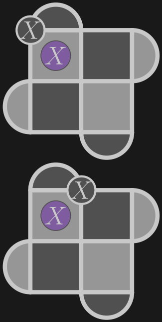
$$m_{c\to q}(E_q) \propto \sum_{E_{q'},\, q' \in \mathrm{supp}(c) \setminus q} \left(\delta_{\mathrm{synd}_c, S_c \cdot E_c} \prod_{q' \in \mathrm{supp}(c) \setminus q} m_{q' \to c}(E_{q'}) \right)$$

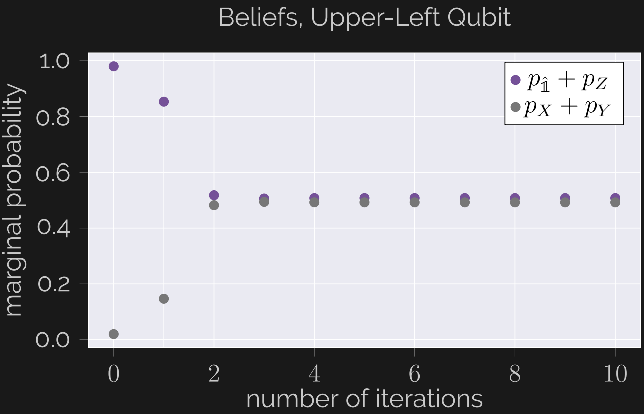
qubit processors calculate new messages to be passed back:

$$m_{q\to c}(E_q) \propto p(E_q) \prod_{c' \in \mathrm{supp}(q) \setminus c} m_{c' \to q}(E_q),$$

Belief Propagation

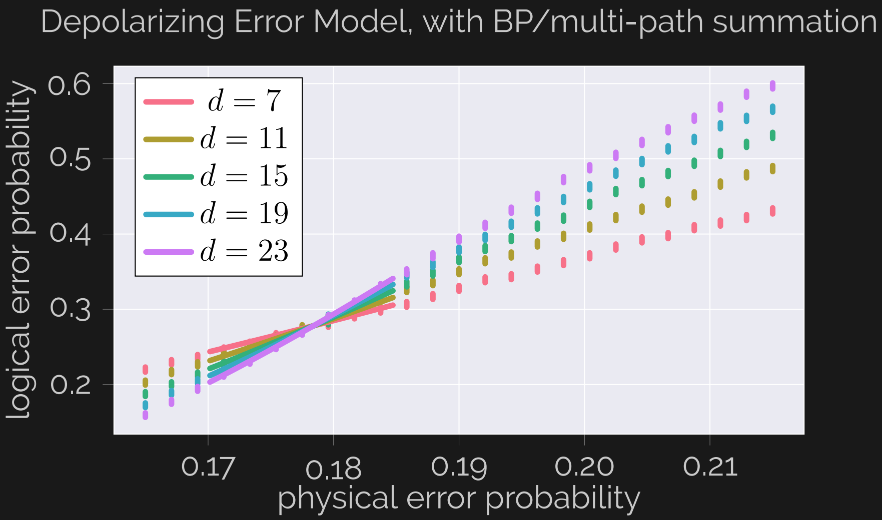
approximate marginal probabilities (Poulin/Chung '08)





Results

threshold goes about $^2/_3$ of the way toward optimal



Summary/Future Work

Summary/Future Work

- by calculating edge weights carefully (see also Stace/Barrett/Doherty ’09), we can increase the threshold

Summary/Future Work

- by calculating edge weights carefully (see also Stace/Barrett/Doherty '09), we can increase the threshold
- uses parallel algorithms with time cost proportional to d

Summary/Future Work

- by calculating edge weights carefully (see also Stace/Barrett/Doherty '09), we can increase the threshold
- uses parallel algorithms with time cost proportional to d
- **question:** can probability of logical error be quickly calculated as a path sum? (See Bravyi/Suchara/Vargo '14)

Summary/Future Work

- by calculating edge weights carefully (see also Stace/Barrett/Doherty '09), we can increase the threshold
- uses parallel algorithms with time cost proportional to d
- **question:** can probability of logical error be quickly calculated as a path sum? (See Bravyi/Suchara/Vargo '14)
- **question:** does the threshold increase appreciably when we extend this to the realistic case?

To Learn More

- QuTech: `qutech.nl` & `qutechacademy.nl`
- IGDORE: `igdore.org`
- Quantum Error Correction & Fault Tolerance: `pirsa.org/C17045`
- My other slides: `https://github.com/bcriger/slide_decks`
- My research: `scirate.com/ben-criger/papers`
- My Twitter: `@BenCriger`