Efficient decoding of random errors for quantum expander codes

Omar Fawzi & **Antoine Grospellier** & Anthony Leverrier

January 17, 2018

Content of the talk

Outline

Main motivation: fault-tolerant quantum computation

Threshold Theorem [Ben-Or & Aharonov, '97]

We can simulate a quantum circuit with T perfect gates and m logical qubits by a fault-tolerant circuit with noisy gates and $\mathcal{O}(m \,\text{polylog}(mT))$ physical qubits.

Main motivation: fault-tolerant quantum computation

Threshold Theorem [Ben-Or & Aharonov, '97]

We can simulate a quantum circuit with T perfect gates and m logical qubits by a fault-tolerant circuit with noisy gates and $\mathcal{O}(m \,\mathrm{polylog}(mT))$ physical qubits.

- Practice: break RSA with 4000 logical qubits, but $10^6 - 10^9$ physical qubits
- [Gottesman, '13] improved this result using constant rate quantum codes instead of concatenated codes

Threshold theorem with constant overhead [Gottesman, '13]

Provided codes with nice properties exist, the ratio physical/logical qubits can be made constant: $\mathcal{O}(m \,\mathrm{polylog}(mT)) \rightsquigarrow \mathcal{O}(m)$

Main motivation: fault-tolerant quantum computation

Threshold Theorem [Ben-Or & Aharonov, '97]

We can simulate a quantum circuit with T perfect gates and m logical qubits by a fault-tolerant circuit with noisy gates and $\mathcal{O}(m \operatorname{polylog}(mT))$ physical qubits.

- Practice: break RSA with 4000 logical qubits, but $10^6 - 10^9$ physical qubits
- [Gottesman, '13] improved this result using constant rate quantum codes instead of concatenated codes

Threshold theorem with constant overhead [Gottesman, '13]

Provided codes with nice properties exist, the ratio physical/logical qubits can be made constant: $\mathcal{O}(m \operatorname{polylog}(mT)) \rightsquigarrow \mathcal{O}(m)$

- Before this work, no existing codes had these "nice properties"
- We proved that quantum expander codes have these "nice properties"

Stabilizer codes

Definition stabilizer codes: given a set $g_1, \ldots, g_{n-k}$ of commuting Pauli operators (product of X and Z Pauli matrices) on n qubits (called generators), we define a quantum code $\mathcal{Q}$ by:

$$\mathcal{Q} = \left\{ |\psi\rangle \in \mathbb{C}^{2^n} : g_1 |\psi\rangle = |\psi\rangle \cdots g_{n-k} |\psi\rangle = |\psi\rangle \right\}$$

Stabilizer codes

Definition stabilizer codes: given a set $g_1, \ldots, g_{n-k}$ of commuting Pauli operators (product of X and Z Pauli matrices) on n qubits (called generators), we define a quantum code $\mathcal{Q}$ by:

$$\mathcal{Q} = \left\{ |\psi\rangle \in \mathbb{C}^{2^n} : g_1 |\psi\rangle = |\psi\rangle \cdots g_{n-k} |\psi\rangle = |\psi\rangle \right\}$$

Parameters of a stabilizer code $[\![n, k, d]\!]$:

- $\mathcal{Q}$ encodes k logical qubits into n physical qubits
  i.e $\mathcal{Q}$ is a 2^k dimensional subspace of $\mathbb{C}^{2^n}$
- A logical error L is a Pauli operator such that $[L, g_i] = 0$ for all i and $L \notin \langle g_1, \ldots, g_{n-k} \rangle$
- The minimal distance d is the minimal weight of a logical error

Stabilizer codes

Definition stabilizer codes: given a set $g_1, \dots, g_{n-k}$ of commuting Pauli operators (product of X and Z Pauli matrices) on n qubits (called generators), we define a quantum code $\mathcal{Q}$ by:

$$\mathcal{Q} = \left\{ |\psi\rangle \in \mathbb{C}^{2^n} : g_1 |\psi\rangle = |\psi\rangle \cdots g_{n-k} |\psi\rangle = |\psi\rangle \right\}$$

Parameters of a stabilizer code $[\![n, k, d]\!]$:

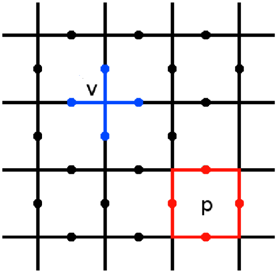
- $\mathcal{Q}$ encodes k logical qubits into n physical qubits
  i.e $\mathcal{Q}$ is a 2^k dimensional subspace of $\mathbb{C}^{2^n}$
- A logical error L is a Pauli operator such that $[L, g_i] = 0$ for all i and $L \notin \langle g_1, \dots, g_{n-k} \rangle$
- The minimal distance d is the minimal weight of a logical error

Decoder for a quantum code:

1. Measurements of the generators $g_1, \dots, g_{n-k}$
   $\rightarrow$ Syndrome $\in \{-1, +1\}^{n-k}$
   Ex: syndrome(code state) $= (+1, +1, +1, \dots)$
2. Syndrome $\rightarrow$ A guess for the error
3. Apply the guessed error to the quantum state

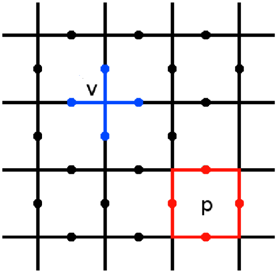
Example: the toric code

- n qubits on edges
- X-type generators associated with vertices
- Z-type generators associated with plaquettes
- $k = \#holes = 2$
- $d = \text{systole} = \sqrt{n/2}$
- Numerical simulations: 10% rate random errors are corrected



Example: the toric code

- n qubits on edges
- X-type generators associated with vertices
- Z-type generators associated with plaquettes
- $k = \#holes = 2$
- $d = \text{systole} = \sqrt{n/2}$
- Numerical simulations: 10% rate random errors are corrected



Adversarial errors VS Random errors:

- "Corrects adversarial errors of size up to $\Theta(\sqrt{n})$": any error of size up to $\Theta(\sqrt{n})$ is corrected
  $\rightarrow$ Link with the minimal distance
- "Corrects random errors of size $\Theta(n)$": an error where qubits are flipped with probability p independently is corrected with high probability
  $\rightarrow$ Framework of our result

"Nice properties" required for [Gottesman, '13]

LDPC

An LDPC code is such that the generators $g_1, \ldots, g_{n-k}$ satisfy:

- The size of the support of each g_i is bounded
- Each qubit is included in the support of a bounded number of g_i

Ex: for the toric code, bounded $= 4$

"Nice properties" required for [Gottesman, '13]

LDPC

An LDPC code is such that the generators $g_1, \ldots, g_{n-k}$ satisfy:

- The size of the support of each g_i is bounded
- Each qubit is included in the support of a bounded number of g_i

Ex: for the toric code, bounded $= 4$

Constant rate

$k = \Theta(n)$
Ex: the toric code does not have a constant rate ($k = 2$)

"Nice properties" required for [Gottesman, '13]

LDPC

An LDPC code is such that the generators $g_1, \ldots, g_{n-k}$ satisfy:

- The size of the support of each g_i is bounded
- Each qubit is included in the support of a bounded number of g_i

Ex: for the toric code, bounded $= 4$

Constant rate

$k = \Theta(n)$

Ex: the toric code does not have a constant rate ($k = 2$)

Efficient decoder

There is a polynomial time decoder which corrects random errors of size $\Theta(n)$ with very high probability

- Very high probability: $\mathbb{P}(\text{correction}) = 1 - o(1/n^c)$ for all $c \in \mathbb{N}$
- $d = \Theta(n^\epsilon)$ is required to get a "very high probability"

Main Theorem

Quantum expander codes are LDPC and have constant rate and have an efficient decoder

Efficient decoder

There is a polynomial time decoder which corrects random errors of size $\Theta(n)$ with very high probability

Main Theorem

Quantum expander codes are LDPC and have constant rate and have an efficient decoder

Efficient decoder

There is a polynomial time decoder which corrects random errors of size $\Theta(n)$ with very high probability

Technical remark:

- The main theorem is true even with syndrome measurements errors (proved after the QIP submission)
- We can apply [Gottesman, '13] with quantum expander codes
- Fault-tolerant quantum computation with constant overhead is possible

Outline

Initial problem:

- The best known minimal distance for a constant rate LDPC code is $\Theta(\sqrt{n}\,\sqrt[4]{\log(n)})$ ([Freedman & Meyer & Luo '02])
- We want to correct random errors of size $\Theta(n)$ with very high probability

Initial problem:

- The best known minimal distance for a constant rate LDPC code is $\Theta(\sqrt{n}\ \sqrt[4]{\log(n)})$ ([Freedman & Meyer & Luo '02])
- We want to correct random errors of size $\Theta(n)$ with very high probability

Solution given by [Dennis & Kitaev & Landahl & Preskill '01], [Kovalev & Pryadko '13]:

- Use of graph percolation theory
- Given a constant rate LDPC code with minimal distance $d = \Omega(n^{\epsilon})$, the maximum likelihood decoder corrects random errors of size $\Theta(n)$ with very high probability

Initial problem:

- The best known minimal distance for a constant rate LDPC code is $\Theta(\sqrt{n}\ \sqrt[4]{\log(n)})$ ([Freedman & Meyer & Luo '02])
- We want to correct random errors of size $\Theta(n)$ with very high probability

Solution given by [Dennis & Kitaev & Landahl & Preskill '01], [Kovalev & Pryadko '13]:

- Use of graph percolation theory
- Given a constant rate LDPC code with minimal distance $d = \Omega(n^{\epsilon})$, the maximum likelihood decoder corrects random errors of size $\Theta(n)$ with very high probability

Remaining problem:

- The maximum likelihood decoder is exponential time in general

Surface codes

- The generators $g_1, \ldots, g_{n-k}$ are given by a tessellations of a surface
- Maximum-likelihood decoding can be done efficiently using Edmond's matching algorithm

Surface codes

- The generators $g_1, \ldots, g_{n-k}$ are given by a tessellations of a surface
- Maximum-likelihood decoding can be done efficiently using Edmond's matching algorithm

	k	Correction up to size	Efficient correction up to size
Toric code [Kit03]	2	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$
Hyperbolic 2D [FML02]	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\log n)$

[Kit03] A Yu Kitaev. "Fault-tolerant quantum computation by anyons". (2003)
[FML02] Michael H Freedman, David A Meyer, and Feng Luo. "Z2-systolic freedom and quantum codes". (2002)

Properties needed to apply [Gottesman, '13]:

- A constant rate quantum code
- $d = \Omega(n^\epsilon)$

Surface codes

- The generators $g_1, \ldots, g_{n-k}$ are given by a tessellations of a surface
- Maximum-likelihood decoding can be done efficiently using Edmond's matching algorithm

	k	Correction up to size	Efficient correction up to size
Toric code [Kit03]	2	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$
Hyperbolic 2D [FML02]	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\log n)$

[Kit03] A Yu Kitaev. "Fault-tolerant quantum computation by anyons". (2003)
[FML02] Michael H Freedman, David A Meyer, and Feng Luo. "Z2-systolic freedom and quantum codes". (2002)

Properties needed to apply [Gottesman, '13]:

- A constant rate quantum code
- $d = \Omega(n^{\epsilon})$

No-go result

We cannot apply [Gottesman '13] using surface codes:

$$kd^2 \leq c(\log k)^2 n \qquad \text{[Delfosse '13]}$$

4 Dimensional hyperbolic codes

- The generators $g_1, \ldots, g_{n-k}$ are given by a tessellation of the 4 Dimensional hyperbolic space
- The bound for surface codes can be beaten by 4D codes
- No efficient maximum-likelihood decoder is known

	k	Correction up to size	Efficient correction up to size
Hyperbolic 4D [GL14], [Has13], [LL17]	$\Theta(n)$	$\Omega(n^{0.2}), \mathcal{O}(n^{0.3})$	$\Theta(\log n)$

[GL14] Larry Guth and Alexander Lubotzky. "Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds". (2014)
[Has13] Matthew B Hastings. "Decoding in Hyperbolic Spaces: LDPC Codes With Linear Rate and Efficient Error Correction". (2013)
[LL17] Vivien Londe and Anthony Leverrier. "Golden codes: 4D hyperbolic regular quantum codes". (2017)

- There might be an efficient decoder to correct any adversarial error of size up to $\Omega(n^{\epsilon})$ but no such algorithm is known
- $\Theta(\log n)$ is not enough to apply [Gottesman '13]

Outline

The CSS construction

Definition [Steane '95], [Calderbank & Shor '95]

We can construct a quantum error correcting code using $\mathcal{C}_X$ and $\mathcal{C}_Z$ two classical error correcting codes such that $\mathcal{C}_X^\perp \subseteq \mathcal{C}_Z$

Each generator $g_1, \ldots, g_{n-k}$ of a CSS-code is either a product of Pauli X matrices or a product of Pauli Z matrices

The CSS construction

Definition [Steane '95], [Calderbank & Shor '95]

We can construct a quantum error correcting code using $\mathcal{C}_X$ and $\mathcal{C}_Z$ two classical error correcting codes such that $\mathcal{C}_X^\perp \subseteq \mathcal{C}_Z$

Each generator $g_1, \ldots, g_{n-k}$ of a CSS-code is either a product of Pauli X matrices or a product of Pauli Z matrices

Remark

The difficulty for constructing CSS code is to find two classical codes which are orthogonal

Hypergraph product codes [Tillich & Zémor '09]

The parity check matrix H of a classical code $\mathcal{C}$ satisfies $\mathcal{C} = \ker H$.
Let H be the parity check matrix of a classical code with constant rate and linear minimal distance.
We define the two classical codes $\mathcal{C}_X$ and $\mathcal{C}_Z$ by their parity check matrices:

$$H_X = (\mathbb{1} \otimes H, H^T \otimes \mathbb{1}) \qquad H_Z = (H \otimes \mathbb{1}, \mathbb{1} \otimes H^T)$$

Then $\mathcal{C}_X^{\perp} \subseteq \mathcal{C}_Z$

Hypergraph product codes [Tillich & Zémor '09]

The parity check matrix H of a classical code $\mathcal{C}$ satisfies $\mathcal{C} = \ker H$.
Let H be the parity check matrix of a classical code with constant rate and linear minimal distance.
We define the two classical codes $\mathcal{C}_X$ and $\mathcal{C}_Z$ by their parity check matrices:

$$H_X = (\mathbb{1} \otimes H, H^T \otimes \mathbb{1}) \qquad H_Z = (H \otimes \mathbb{1}, \mathbb{1} \otimes H^T)$$

Then $\mathcal{C}_X^\perp \subseteq \mathcal{C}_Z$

Definition

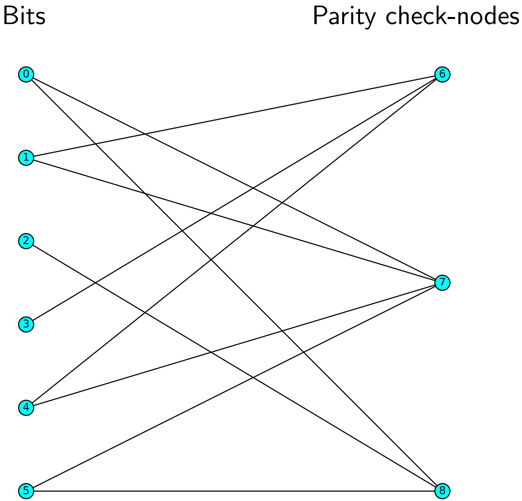
The hypergraph product is defined as $\text{CSS}(\mathcal{C}_X, \mathcal{C}_Z)$.
It's a constant rate code with minimal distance $d = \Theta(\sqrt{n})$

- Freedom to choose H
- [Leverrier & Tillich & Zémor '15] chooses H as the parity check-matrix of a "classical expander code" ([Sipser & Spielman, '96])

Classical expander codes

The parity check matrix H of a classical code $\mathcal{C}$ satisfies $\mathcal{C} = \ker H$
H represented by a factor graph

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$



Classical expander codes

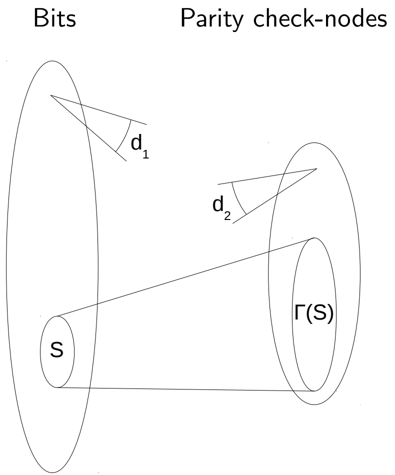
The parity check matrix H of a classical code $\mathcal{C}$ satisfies $\mathcal{C} = \ker H$
H represented by a factor graph

Definition of a (γ, δ) expander graph

For all $S \subseteq \{\text{Bits}\}$, if $|S| \leq \gamma n$ then:

$$|\Gamma(S)| \geq (1-\delta) d_1 |S|$$
$$|\Gamma(S)| \leq d_1 |S|$$

Expander graph
$\rightarrow$ Parity check matrix
$\rightarrow$ Classical expander code
$\rightarrow$ Quantum expander code



- **Classical case (bit-flip algorithm):**
  - As long as it is possible to flip a single bit to decrease the syndrome weight, flip this bit
  - This efficient algorithm corrects any adversarial error of size up to $\Theta(n)$ for classical expander codes [Sipser & Spielman, '96]

Decoder for quantum expander codes

- **Classical case (bit-flip algorithm):**
  - As long as it is possible to flip a single bit to decrease the syndrome weight, flip this bit
  - This efficient algorithm corrects any adversarial error of size up to $\Theta(n)$ for classical expander codes [Sipser & Spielman, '96]
- **Quantum case (small-set-flip algorithm):**
  - The "qubit-flip" algorithm doesn't work
  - Idea: try to flip several qubits at each step
  - As long as it is possible to flip a subset of a generator to decrease the syndrome weight, flip this subset

Decoder for quantum expander codes

- **Classical case (bit-flip algorithm):**
  - As long as it is possible to flip a single bit to decrease the syndrome weight, flip this bit
  - This efficient algorithm corrects any adversarial error of size up to $\Theta(n)$ for classical expander codes [Sipser & Spielman, '96]
- **Quantum case (small-set-flip algorithm):**
  - The "qubit-flip" algorithm doesn't work
  - Idea: try to flip several qubits at each step
  - As long as it is possible to flip a subset of a generator to decrease the syndrome weight, flip this subset

Theorem [Leverrier & Tillich & Zémor '15]

This efficient algorithm corrects any adversarial error of size up to $\Theta(\sqrt{n})$ for quantum expander codes

Outline

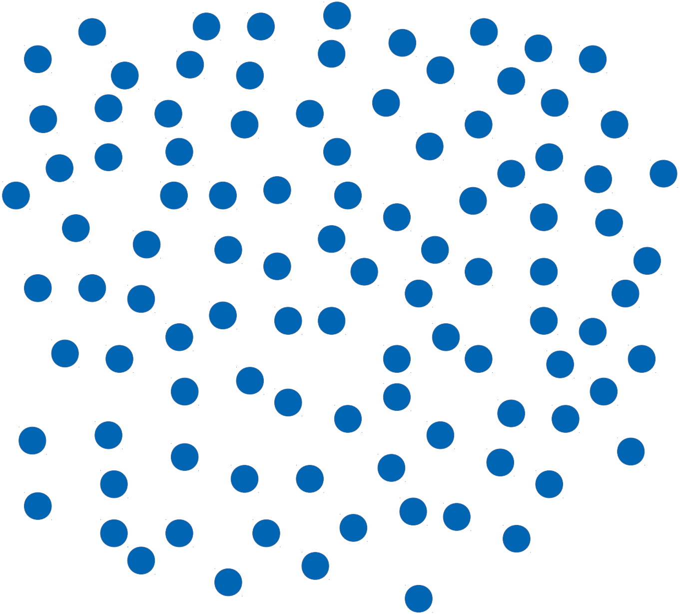
Summary of our contribution

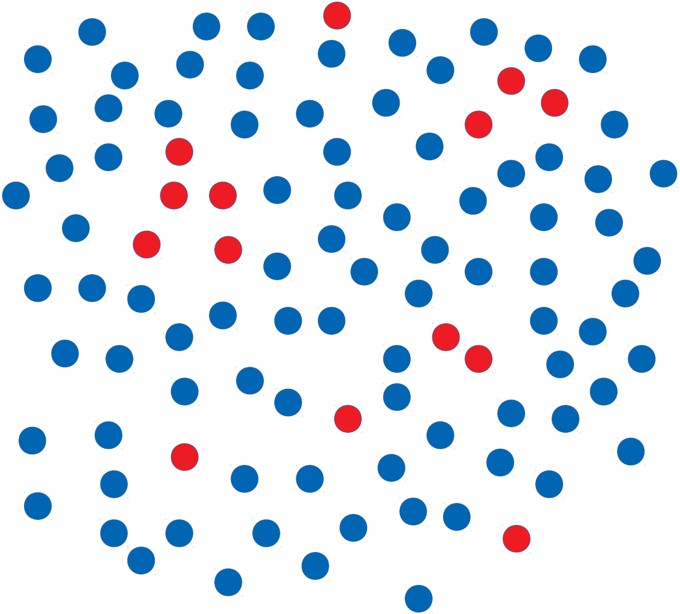
Question: What happens for random errors of size $\Theta(n)$?

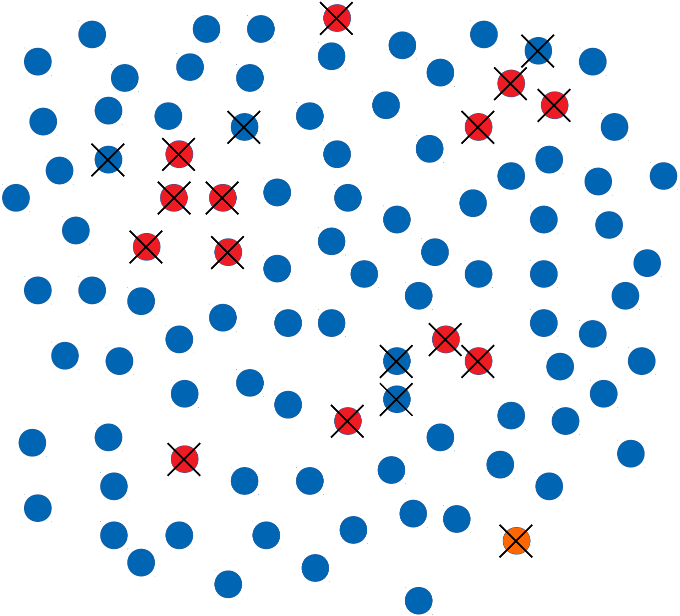
Theorem: what we proved

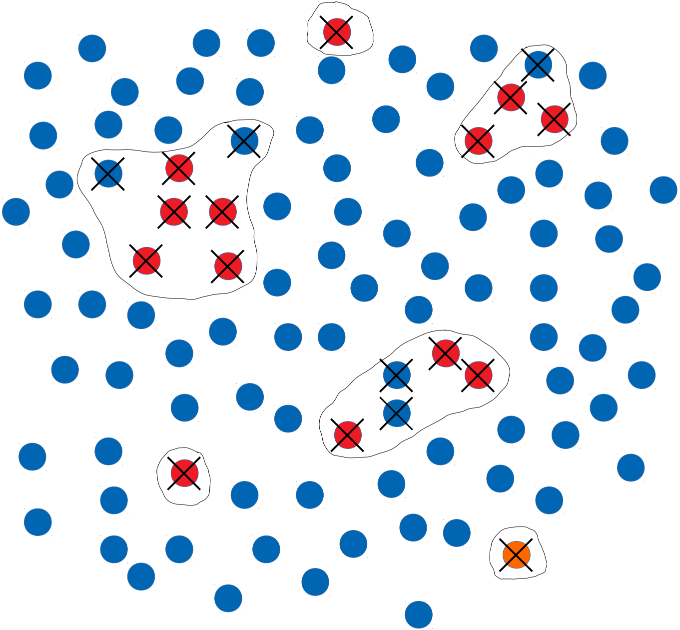
For a probability of error $p < p_{\text{th}}$:
$\mathbb{P}(\text{small-set-flip corrects the error}) = 1 - 1/e^{\Omega(\sqrt{n})}$

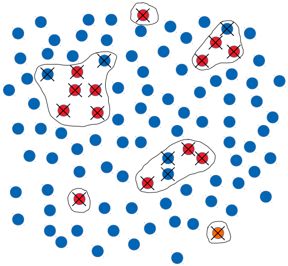
Idea. The algorithm is local with respect to the adjacency graph











The number of flips is linear in the size of the initial error

Definition: α-subset, $\alpha \in (0, 1]$

X is an α-subset of E if $|X \cap E| \geq \alpha |X|$

- Each connected component X is an α-subset of $\{\text{red dots}\} \cap X$

Theorem: what we proved

For a probability of error $p < p_{\text{th}}$:
$\mathbb{P}(\text{small-set-flip corrects the error}) = 1 - 1/e^{\Omega(\sqrt{n})}$

Theorem: what we proved

For a probability of error $p < p_{\text{th}}$:
$\mathbb{P}(\text{small-set-flip corrects the error}) = 1 - 1/e^{\Omega(\sqrt{n})}$

Key lemma: percolation

Let $\alpha \in (0, 1]$ and a probability of error $p < cst(\alpha, d)$.
With probability $1 - 1/e^{\Omega(\sqrt{n})}$:

- If X is a connected α-subset of the error then $|X| < c\sqrt{n}$

Theorem: what we proved

For a probability of error $p < p_{\text{th}}$:
$\mathbb{P}(\text{small-set-flip corrects the error}) = 1 - 1/e^{\Omega(\sqrt{n})}$

Key lemma: percolation

Let $\alpha \in (0,1]$ and a probability of error $p < cst(\alpha, d)$.
With probability $1 - 1/e^{\Omega(\sqrt{n})}$:

- If X is a connected α-subset of the error then $|X| < c\sqrt{n}$

Sketch of the proof of the theorem:
Take a random error and run the small-set-flip algorithm. Let X be a connected component of the marked qubits:

- X is an α-subset of the error
- $|X| < c\sqrt{n}$
- X is corrected

This is true for any $X \to$ the entire error is corrected

Conclusion

Quantum expander codes:

- Are LDPC quantum codes
- Have a constant rate
- Have a good minimal distance: $d = \Theta(\sqrt{n})$

The decoder:

- Corrects any adversarial error of size up to $\Theta(\sqrt{n})$
- For a probability of error $p < p_{\text{th}}$: $\mathbb{P}(\text{correction}) = 1 - 1/e^{\Omega(\sqrt{n})}$

Corollary:

- Fault tolerant quantum computation with constant overhead is possible

Conclusion

Quantum expander codes:

- Are LDPC quantum codes
- Have a constant rate
- Have a good minimal distance: $d = \Theta(\sqrt{n})$

The decoder:

- Corrects any adversarial error of size up to $\Theta(\sqrt{n})$
- For a probability of error $p < p_{\text{th}}$: $\mathbb{P}(\text{correction}) = 1 - 1/e^{\Omega(\sqrt{n})}$

Corollary:

- Fault tolerant quantum computation with constant overhead is possible

Future work ($p_{\text{th}} \sim 10^{-16}$):

- Run simulations
- Improve our numerical value for the threshold

Conclusion

Quantum expander codes:

- Are LDPC quantum codes
- Have a constant rate
- Have a good minimal distance: $d = \Theta(\sqrt{n})$

The decoder:

- Corrects any adversarial error of size up to $\Theta(\sqrt{n})$
- For a probability of error $p < p_{\text{th}}$: $\mathbb{P}(\text{correction}) = 1 - 1/e^{\Omega(\sqrt{n})}$

Corollary:

- Fault tolerant quantum computation with constant overhead is possible

Future work ($p_{\text{th}} \sim 10^{-16}$):

- Run simulations
- Improve our numerical value for the threshold

Thank you for your attention

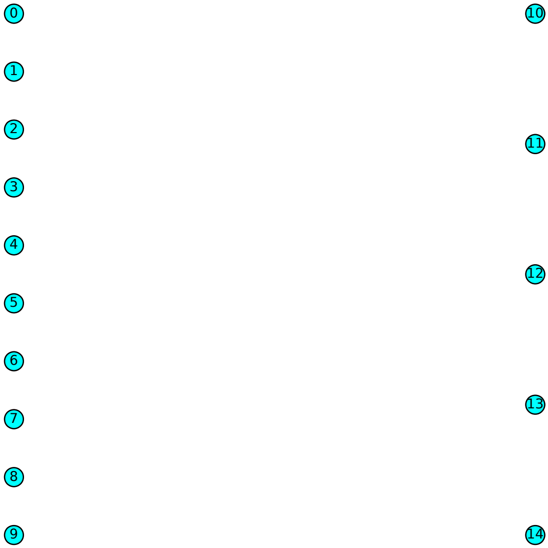
Known constructions of quantum LDPC codes

	k	Correction up to size	Efficient correction up to size
Toric code [Kit03]	2	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$
Hyperbolic 2D [FML02]	$\Theta(n)$	$\Theta(\log n)$	$\Theta(\log n)$
Hyperbolic 4D [GL14], [Has13], [LL17]	$\Theta(n)$	$\Omega(n^{0.2}), \mathcal{O}(n^{0.3})$	$\Theta(\log n)$
Expander codes [TZ14], [LTZ15]	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$

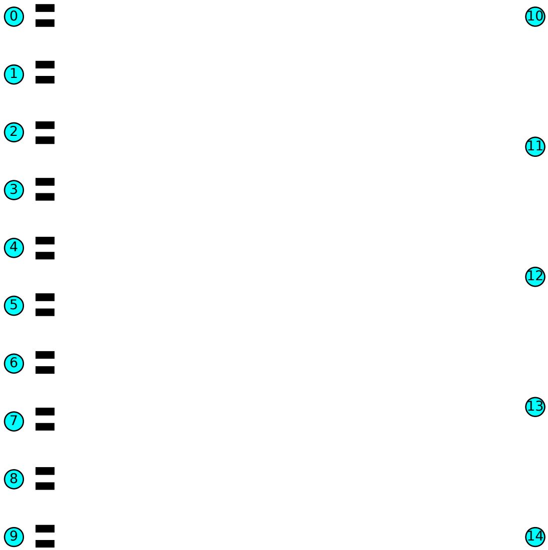
[Kit03] A Yu Kitaev. "Fault-tolerant quantum computation by anyons". (2003)
[FML02] Michael H Freedman, David A Meyer, and Feng Luo. "Z2-systolic freedom and quantum codes". (2002)
[GL14] Larry Guth and Alexander Lubotzky. "Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds". (2014)
[Has13] Matthew B Hastings. "Decoding in Hyperbolic Spaces: LDPC Codes With Linear Rate and Efficient Error Correction". (2013)
[LL17] Vivien Londe and Anthony Leverrier. "Golden codes: 4D hyperbolic regular quantum codes". (2017)
[TZ14] Jean-Pierre Tillich and Gilles Zémor. "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength". (2014)
[LTZ15] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. "Quantum expander codes". (2015)

$n = 10, m = 5, d_1 = 2, d_2 = \frac{n}{m} d_1 = 4$

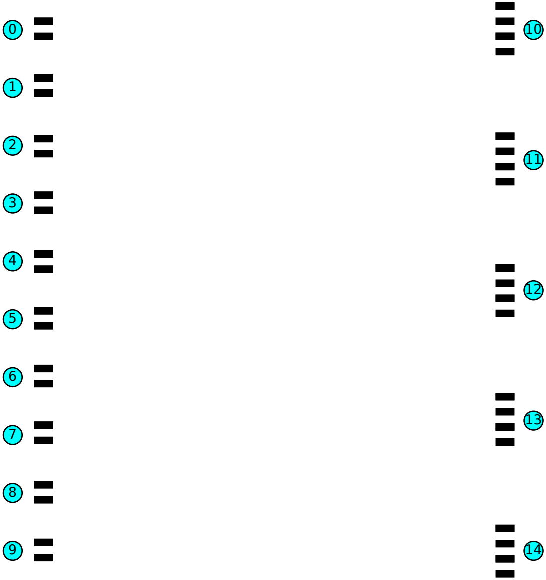
$n = 10, m = 5, d_1 = 2, d_2 = \frac{n}{m} d_1 = 4$



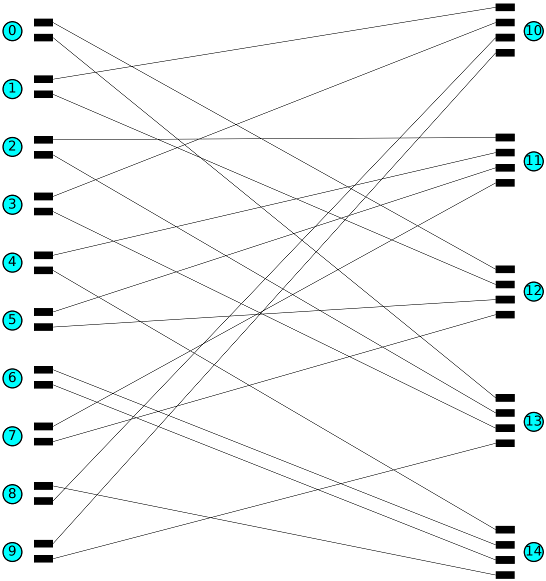
$n = 10, m = 5, d_1 = 2, d_2 = \frac{n}{m} d_1 = 4$



$n = 10, m = 5, d_1 = 2, d_2 = \frac{n}{m} d_1 = 4$



$n = 10, m = 5, d_1 = 2, d_2 = \frac{n}{m} d_1 = 4$



$n = 10, m = 5, d_1 = 2, d_2 = \frac{n}{m} d_1 = 4$