Decoding problem for topological quantum codes

David Poulin

Département de Physique
Université de Sherbrooke

Joint work with: G. Duclos-Cianci and H. Bombin

Topological Quantum Computing
Simons Center for Geometry and Physics, NY, September 2011
A different mindset...

- Topological order can be used as a software for quantum error correction.
  - There are no anyons in sight, no topologically ordered system, etc.
  - We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to “simulate” a topologically ordered system, inheriting its intrinsic robustness.
- Diffusion of thermal defects can destroy topological order.
- Errors in our simulation will cause such defects.
- We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)
- Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
A different mindset...

- Topological order can be used as a software for quantum error correction.
- There are no anyons in sight, no topologically ordered system, etc.
  - We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to "simulate" a topologically ordered system, inheriting its intrinsic robustness.
  - Diffusion of thermal defects can destroy topological order.
  - Errors in our simulation will cause such defects.
  - We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)
- Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
A different mindset...

- Topological order can be used as a software for quantum error correction.
- There are no anyons in sight, no topologically ordered system, etc.
- We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to “simulate” a topologically ordered system, inheriting its intrinsic robustness.
- Diffusion of thermal defects can destroy topological order.
- Errors in our simulation will cause such defects.
- We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)
- Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
Topological order can be used as a software for quantum error correction.
There are no anyons in sight, no topologically ordered system, etc.
We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to “simulate” a topologically ordered system, inheriting its intrinsic robustness.
Diffusion of thermal defects can destroy topological order.
Errors in our simulation will cause such defects.
We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)
Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
A different mindset...

- Topological order can be used as a software for quantum error correction.
- There are no anyons in sight, no topologically ordered system, etc.
- We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to “simulate” a topologically ordered system, inheriting its intrinsic robustness.
- Diffusion of thermal defects can destroy topological order.
- Errors in our simulation will cause such defects.
- We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)
- Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
Topological order can be used as a software for quantum error correction.

There are no anyons in sight, no topologically ordered system, etc.

We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to “simulate” a topologically ordered system, inheriting its intrinsic robustness.

Diffusion of thermal defects can destroy topological order.

Errors in our simulation will cause such defects.

We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)

Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
A different mindset...

- Topological order can be used as a software for quantum error correction.
- There are no anyons in sight, no topologically ordered system, etc.
- We use a garden-variety noisy quantum computer (e.g. 2D superconducting circuit) to “simulate” a topologically ordered system, inheriting its intrinsic robustness.
- Diffusion of thermal defects can destroy topological order.
- Errors in our simulation will cause such defects.
- We imagine monitoring the presence of such defects, and eliminating them. (Decoding problem)
- Quantum computing can be perform reliably provided that the creation rate of defects is low on our monitoring timescale.
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
**Definition**

- **Kitaev’s code**

- \[ H = - \left( \sum_s A_s + \sum_p B_p \right) \]

- The \( A_s \) et \( B_p \) are trivial cycles

- Trivial action on ground space
  - \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)

- \( A_s \) \( B_p \) generate all trivial loops.

- Non-trivial cycles are symmetries of the Hamiltonian.

- Represent encoded information.
Kitaev’s code

Definition

- \( H = - (\sum_s A_s + \sum_p B_p) \)
- The \( A_s \) et \( B_p \) are trivial cycles
  - Trivial action on ground space
    - \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
  - \( A_s \) \( B_p \) generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Definition

- \( H = -\left( \sum_s A_s + \sum_p B_p \right) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
- \( A_s \) \( B_p \) generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Kitaev’s code

Definition

- \( H = - (\sum_s A_s + \sum_p B_p) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
- \( A_s \) \( B_p \) generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Definition

- \( H = - (\sum_s A_s + \sum_p B_p) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
- \( A_s \) \( B_p \) generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Definition

- \( H = -\left( \sum_s A_s + \sum_p B_p \right) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
- \( A_s \) \( B_p \) generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
**Definition**

- $H = - (\sum_s A_s + \sum_p B_p)$
- The $A_s$ et $B_p$ are trivial cycles
- Trivial action on ground space $A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle$
- $A_s B_p$ generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Definition

- $H = -\left( \sum_s A_s + \sum_p B_p \right)$
- The $A_s$ and $B_p$ are trivial cycles.
- Trivial action on ground space $A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle$
- $A_s$ and $B_p$ generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Kitaev's code

**Definition**

- \( H = - (\sum_s A_s + \sum_p B_p) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  \[ A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \]
- \( A_s \) \( B_p \) generate all trivial loops.
- Non-trivial cycles are symmetries of the Hamiltonian.
- Represent encoded information.
Consider error $E = \sigma^i_x$.

- $\sigma^i_x$ anti-commutes with adjacent plaquettes.
- $\sigma^i_x |\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.
- Since $H = -(\sum_s A_s + \sum_p B_p)$, $\sigma^i_x$ costs 2 energy units.
- This error has created a pair of magnetic particles.
Consider error $E = \sigma^i_x$.

$\sigma^i_x$ anti-commutes with adjacent plaquettes.

$\sigma^i_x |\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.

Since $H = - (\sum_s A_s + \sum_p B_p)$, $\sigma^i_x$ costs 2 energy units.

This error has created a pair of magnetic particles.
Consider error \( E = \sigma^i_x \).
\( \sigma^i_x \) anti-commutes with adjacent plaquettes.
\( \sigma^i_x |\psi\rangle \) is a -1 eigenstate of \( B_p \) and \( B_{p'} \).
Since \( H = - (\sum_s A_s + \sum_p B_p) \), \( \sigma^i_x \) costs 2 energy units.
This error has created a pair of magnetic particles.
Consider error \( E = \sigma_x^i \).

\( \sigma_x^i \) anti-commutes with adjacent plaquettes.

\( \sigma_x^i |\psi\rangle \) is a -1 eigenstate of \( B_p \) and \( B_{p'} \).

Since \( H = - (\sum_s A_s + \sum_p B_p) \), \( \sigma_x^i \) costs 2 energy units.

This error has created a pair of magnetic particles.
Consider error $E = \sigma_x^i$.

- $\sigma_x^i$ anti-commutes with adjacent plaquettes.
- $\sigma_x^i |\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.
- Since $H = - (\sum_s A_s + \sum_p B_p)$, $\sigma_x^i$ costs 2 energy units.
- This error has created a pair of magnetic particles.
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
Error chains are attached to particles, each with given energy.

- Particles can move around at no energy cost.
- Error chains can be stretched freely.
Error chains are attached to particles, each with given energy.

Particles can move around at no energy cost.

Error chains can be stretched freely.
Error chains are attached to particles, each with given energy.

Particles can move around at no energy cost.

Error chains can be stretched freely.
An error can annihilate two particles.

The particle’s worldline is left behind after fusion.

Particle fusion can leave behind a worldline corresponding to a logical operation.

Memory corruption
**Particle annihilation**

- An error can annihilate two particles.
- The particle’s worldline is left behind after fusion.
- Particle fusion can leave behind a worldline corresponding to a logical operation.

**Memory corruption**
• An error can annihilate two particles.
• The particle’s worldline is left behind after fusion.
• Particle fusion can leave behind a worldline corresponding to a logical operation.
An error can annihilate two particles.
The particle’s worldline is left behind after fusion.
Particle fusion can leave behind a worldline corresponding to a logical operation.

Memory corruption
The same story holds for $\sigma_z$ errors.

These will create electrical particles located at the lattice’s vertices (plaquette of dual lattice).
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
An error produces defects (error syndrome)

- Measure particle position, but not worldline.
- Many worldlines consistent with defects.
- Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding

Infer worldline homology from particle location.

15 % Noise rate
An error produces defects (error syndrome)

Measure particle position, but not worldline.

Many worldlines consistent with defects.

Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding

Infer worldline homology from particle location.
Decoding problem

Error syndrome & decoding

- An error produces defects (error syndrome)
- Measure particle position, but not worldline.
- Many worldlines consistent with defects.
- Worldline with different homologies have different effect on ground space: MUST be distinguished.

15% Noise rate

Decoding

Infer worldline homology from particle location.
Error syndrome & decoding

- An error produces defects (error syndrome)
- Measure particle position, but not worldline.
- Many worldlines consistent with defects.
- Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding
Infer worldline homology from particle location.

15% Noise rate
Decoding problem

Error syndrome & decoding

- An error produces defects (error syndrome)
- Measure particle position, but not worldline.
- Many worldlines consistent with defects.
- Worldline with different homologies have different effect on ground space: MUST be distinguished.

15% Noise rate

Decoding

Infer worldline homology from particle location.
An error produces defects (error syndrome)

Measure particle position, but not worldline.

Many worldlines consistent with defects.

Worldline with different homologies have different effect on ground space: MUST be distinguished.

15 % Noise rate

Decoding
Infer worldline homology from particle location.
Decoding problem

Existing methods

Energy Minimization

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $\mathcal{O}(\ell^6)$

- It is very slow, $\mathcal{O}(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
### Existing methods

#### Energy Minimization
- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

#### Depolarization error model
- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
**Existing methods**

**Energy Minimization**

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

**Depolarization error model**

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Decoding problem

Existing methods

Energy Minimization

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error X, Y, or Z with probability $p/3$. 
Decoding problem

Existing methods

Energy Minimization

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Existing methods

Energy Minimization

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Decoding problem

Existing methods

Energy Minimization

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Two possible pairings with different homologies

- First one has lower weight (Energy).
- Second one is highly degenerate (Entropy).

Optimal decoding
Homology class with lowest free energy $F = E - TS$.

- Nishimori $T^{-1} = \ln \frac{3(1-p)}{p}$.
- Sum over all equivalent errors.
Two possible pairings with different homologies

- First one has lower weight (Energy).
- Second one is highly degenerate (Entropy).

Optimal decoding

Homology class with lowest free energy $F = E - TS$.

- Nishimori $T^{-1} = \ln \frac{3(1-p)}{p}$.
- Sum over all equivalent errors.
Two possible pairings with different homologies
- First one has lower weight (Energy).
- Second one is highly degenerate (Entropy).

Optimal decoding
Homology class with lowest free energy $F = E - TS$.
- Nishimori $T^{-1} = \ln \frac{3(1-p)}{p}$.
- Sum over all equivalent errors.
Two possible pairings with different homologies
- First one has lower weight (Energy).
- Second one is highly degenerate (Entropy).

Optimal decoding
Homology class with lowest free energy $F = E - TS$.
- Nishimori $T^{-1} = \ln \frac{3(1-p)}{p}$.
- Sum over all equivalent errors.
Decoding problem

Energy vs Free Energy

- Two possible pairings with different homologies
  - First one has lower weight (Energy).
  - Second one is highly degenerate (Entropy).

Optimal decoding

Homology class with lowest free energy $F = E - TS$.

- Nishimori $T^{-1} = \ln \frac{3(1-p)}{p}$.
- Sum over all equivalent errors.
Two possible pairings with different homologies

- First one has lower weight (Energy).
- Second one is highly degenerate (Entropy).

**Optimal decoding**

Homology class with lowest free energy $F = E - TS$.

- Nishimori $T^{-1} = \ln \frac{3(1-p)}{p}$.
- Sum over all equivalent errors.
Two possible pairings with different homologies

- Both seemingly have same weight
- A Y error has same weight as X and Z: we were overcounting.
- Electric Z and Magnetic X errors are not independent.
Two possible pairings with different homologies

- Both seemingly have same weight
- A Y error has same weight as X and Z: we were overcounting.
- Electric Z and Magnetic X errors are not independent.
Two possible pairings with different homologies

- Both seemingly have same weight
- A Y error has same weight as X and Z: we were overcounting.
- Electric Z and Magnetic X errors are not independent.
Two possible pairings with different homologies

Both seemingly have same weight

A $Y$ error has same weight as $X$ and $Z$: we were overcounting.

Electric $Z$ and Magnetic $X$ errors are not independent.
Two possible pairings with different homologies
Both seemingly have same weight
A $Y$ error has same weight as $X$ and $Z$: we were overcounting.

Electric $Z$ and Magnetic $X$ errors are not independent.
Two possible pairings with different homologies
Both seemingly have same weight
A $Y$ error has same weight as $X$ and $Z$: we were overcounting.
Electric $Z$ and Magnetic $X$ errors are not independent.
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
Renormalization Group Decoder

Scale invariance

- **Original $B_p$ checks**
  - Basis change (row operations on $C$)
  - Obtain scale invariant generators
  - Structure similar to a concatenated code.
- Soft-decode each small block.
- Pass information to next encoding level.
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
  - Obtain scale invariant generators
  - Structure similar to a concatenated code.
- Soft-decode each small block.
- Pass information to next encoding level.
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
  - Obtain scale invariant generators
  - Structure similar to a concatenated code.
- Soft-decode each small block.
- Pass information to next encoding level.
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

- Soft-decode each small block.
- Pass information to next encoding level.
Original $B_p$ checks
Basis change (row operations on $C$)
Obtain scale invariant generators
Structure similar to a concatenated code.

- Soft-decode each small block.
- Pass information to next encoding level.
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

- Soft-decode each small block.
- Pass information to next encoding level.
Renormalization Group Decoder

Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

- Soft-decode each small block.
- Pass information to next encoding level.
**Scale invariance**

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

- Soft-decode each small block.
- Pass information to next encoding level.
Concatenated code

- Encoding one qubit in 3 qubits.

Decoding

\[ P(L) = \sum'_E P(E) \]

- Sum over \( E \) equivalent to \( L \) and with right syndrome.
- \( P(E) \) given by error model.

- Encode each of these qubits...

Decoding

- Compute error probability for each encoded qubit.
- Pass that probability to the next level up.
Concatenated code

- Encoding one qubit in 3 qubits.

**Decoding**

\[ P(L) = \sum'_E P(E) \]

- Sum over \( E \) equivalent to \( L \) and with right syndrome.
- \( P(E) \) given by error model.

- Encode each of these qubits...

**Decoding**

- Compute error probability for each encoded qubit.
- Pass that probability to the next level up.
Concatenated code

- Encoding one qubit in 3 qubits.

Decoding

\[ P(L) = \sum'_E P(E) \]

- Sum over \( E \) equivalent to \( L \) and with right syndrome.
- \( P(E) \) given by error model.

- Encode each of these qubits...

Decoding

- Compute error probability for each encoded qubit.
- Pass that probability to the next level up.
Concatenated code

- Encoding one qubit in 3 qubits.

Decoding

\[ P(L) = \sum_{E} P(E) \]

- Sum over \( E \) equivalent to \( L \) and with right syndrome.
- \( P(E) \) given by error model.

- Encode each of these qubits...

Decoding

- Compute error probability for each encoded qubit.
- Pass that probability to the next level up.
Concatenated code

- Encoding one qubit in 3 qubits.

Decoding

\[ P(L) = \sum_E' P(E) \]

- Sum over \( E \) equivalent to \( L \) and with right syndrome.
- \( P(E) \) given by error model.

- Encode each of these qubits...

Decoding

- Compute error probability for each encoded qubit.
- Pass that probability to the next level up.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.
- Given the particle configuration in a unit cell, compute the prob.
  - A magnetic particle has crossed the cell from top to bottom.
  - A electric particle has crossed the cell from top to bottom.
  - A magnetic particle has crossed the cell from left to right.
  - etc.
- There are 16 possibilities corresponding to \( \{I, X, Y, Z\}^2 \), 2 qubits.
- This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to \( \{I, X, Y, Z\}^2 \), 2 qubits.
- This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to \( \{I, X, Y, Z\}^2 \), 2 qubits.
- This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to $\{I, X, Y, Z\}^2$, 2 qubits.
- This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to \( \{I, X, Y, Z\}^2 \), 2 qubits.

This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to $\{I, X, Y, Z\}^2$, 2 qubits.

This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to \( \{I, X, Y, Z\}^2 \), 2 qubits.

This is done by brute force: sum over all worldline configurations.
Think of the Kitaev’s code as a concatenated code:
- It is made up of a bunch of small (open boundary) topological codes, joined into larger topological codes, etc.

Given the particle configuration in a unit cell, compute the prob.
- A magnetic particle has crossed the cell from top to bottom.
- A electric particle has crossed the cell from top to bottom.
- A magnetic particle has crossed the cell from left to right.
- etc.

There are 16 possibilities corresponding to \( \{I, X, Y, Z\}^2 \), 2 qubits.

This is done by brute force: sum over all worldline configurations.
Topological codes are NOT concatenated codes.

- Cannot break lattices into constant-size cells in such a way that each stabilizer overlaps with a single region.
- Use overlapping cells instead.
Topological codes are NOT concatenated codes.

- Cannot break lattices into constant-size cells in such a way that each stabilizer overlaps with a single region.
- Use overlapping cells instead.
Topological codes are NOT concatenated codes.

- Cannot break lattices into constant-size cells in such a way that each stabilizer overlaps with a single region.
- Use overlapping cells instead.
Topological codes are NOT concatenated codes.

- Cannot break lattices into constant-size cells in such a way that each stabilizer overlaps with a single region.
- Use overlapping cells instead.
Topological codes are NOT concatenated codes.

- Cannot break lattices into constant-size cells in such a way that each stabilizer overlaps with a single region.
- Use overlapping cells instead.
Self-consistence

- Boundary qubits treated as independent variables on neighboring unit cells.
- Probabilities assigned by different cells to a given qubit differ.
- Impose mean-field consistencies conditions on marginal probabilities.
- Solve by belief propagation.
- Complexity $O(\ell^2)$ parallelizable to constant time.
Renormalization Group Decoder

Self-consistence

- Boundary qubits treated as independent variables on neighboring unit cells.
- Probabilities assigned by different cells to a given qubit differ.
- Impose mean-field consistencies conditions on marginal probabilities.
- Solve by belief propagation.
- Complexity $O(\ell^2)$ parallelizable to constant time.

David Poulin (Sherbrooke)
Self-consistence

- Boundary qubits treated as independent variables on neighboring unit cells.
- Probabilities assigned by different cells to a given qubit differ.
- Impose mean-field consistencies conditions on marginal probabilities.
- Solve by belief propagation.
- Complexity $O(\ell^2)$ parallelizable to constant time.
Boundary qubits treated as independent variables on neighboring unit cells.

Probabilities assigned by different cells to a given qubit differ.

Impose mean-field consistencies conditions on marginal probabilities.

Solve by belief propagation.

Complexity $O(\ell^2)$ parallelizable to constant time.
Self-consistence

- Boundary qubits treated as independent variables on neighboring unit cells.
- Probabilities assigned by different cells to a given qubit differ.
- Impose mean-field consistencies conditions on marginal probabilities.
- Solve by belief propagation.
- Complexity $O(\ell^2)$ parallelizable to constant time.
1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
Threshold \( \approx \) 15\%, compared to 15.5\% for PMA.
- Exponentially faster with marginal performance loss.
Bit-flip threshold $\approx 8.2\%$, compared to 10.3\% for PMA.

Much faster even without parallelization (10$^6$ sites).

Illustrates flexibility.
- Use of additional belief propagation.
- Threshold \( \approx 16.5\% \) (not to date), compared to 15.5\% for PMA.
- Illustrates flexibility.
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
Topological color code

- Qubits located on vertices
  - Plaquette stabilizers $S_p = \bigotimes_{j \in \partial p} \sigma_j$, for $\sigma = \sigma^x$ and $\sigma^z$.
  - 16 topological charges
    - 10 bosons
    - 6 fermions
  - Same particle type and statistics as 2 copies of Kitaev’s code
  - Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

- Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Topological color code

- Qubits located on vertices
- Plaquette stabilizers $S_p = \bigotimes_{j \in \partial p} \sigma_j$, for $\sigma = \sigma^x$ and $\sigma^z$.
  - 16 topological charges
    - 10 bosons
    - 6 fermions
  - Same particle type and statistics as 2 copies of Kitaev’s code
- Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Topological color code

- Qubits located on vertices
- Plaquette stabilizers \( S_p = \bigotimes_{j \in \partial p} \sigma_j \), for \( \sigma = \sigma^x \) and \( \sigma^z \).
- 16 topological charges
  - 10 bosons
  - 6 fermions
- Same particle type and statistics as 2 copies of Kitaev’s code
- Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

- Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Topological color code

- Qubits located on vertices
- Plaquette stabilizers $S_p = \bigotimes_{j \in \partial p} \sigma_j$, for $\sigma = \sigma^x$ and $\sigma^z$.
- 16 topological charges
  - 10 bosons
  - 6 fermions
- Same particle type and statistics as 2 copies of Kitaev’s code
- Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

- Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Topological color code

- Qubits located on vertices
- Plaquette stabilizers $S_p = \bigotimes_{j \in \partial p} \sigma_j$, for $\sigma = \sigma^x$ and $\sigma^z$.
- 16 topological charges
  - 10 bosons
  - 6 fermions
- Same particle type and statistics as 2 copies of Kitaev’s code
- Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

- Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Topological color code

- Qubits located on vertices
- Plaquette stabilizers $S_p = \bigotimes_{j \in \partial p} \sigma_j$, for $\sigma = \sigma^x$ and $\sigma^z$.
- 16 topological charges
  - 10 bosons
  - 6 fermions
- Same particle type and statistics as 2 copies of Kitaev’s code
- Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

- Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Topological color code

- Qubits located on vertices
- Plaquette stabilizers $S_p = \bigotimes_{j \in \partial p} \sigma_j$, for $\sigma = \sigma^x$ and $\sigma^z$.
- 16 topological charges
  - 10 bosons
  - 6 fermions
- Same particle type and statistics as 2 copies of Kitaev’s code
- Efficient decoding algorithm for this code?

Every 2D, translationally invariant, non-chiral stabilizer code with local generators and macroscopic minimal distance is locally equivalent to a finite number of copies of Kitaev’s code.

- Map this code to two copies of Kitaev’s code and operate decoding on those instead.
Results

Decoding error probability vs. Bit-Flip channel strength for different block lengths $l$:
- $l=16$
- $l=32$
- $l=64$
- $l=128$
- $l=256$

The graph shows the decoding error probability as the Bit-Flip channel strength $p\%$ increases for various block lengths.
Extension to other codes

Results for topological subsystem color code

Decoding error probability vs. Depolarizing channel strength, $p\%$

- $l=8$
- $l=16$
- $l=32$
- $l=64$
- $l=128$

David Poulin (Sherbrooke)

Decoding Problem

IQI Caltech’11 29 / 38
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
So far, we have assumed that we can perfectly monitor the presence of defects.

- Measurements will themselves be noisy.
- Performing error correction with noisy instruments can kill the computation.
- Can model erroneous measurements by ghost defects appearing with probability $p$.
- Can overcome measurement errors by repeating the measurements periodically.
- Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

- Performing error correction with noisy instruments can kill the computation.
- Can model erroneous measurements by ghost defects appearing with probability $p$.
- Can overcome measurement errors by repeating the measurements periodically.
- Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

Can model erroneous measurements by ghost defects appearing with probability $p$.

Can overcome measurement errors by repeating the measurements periodically.

Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

Can model erroneous measurements by ghost defects appearing with probability $p$.

Can overcome measurement errors by repeating the measurements periodically.

Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

Can model erroneous measurements by ghost defects appearing with probability $p$.

Can overcome measurement errors by repeating the measurements periodically.

Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

Can model erroneous measurements by ghost defects appearing with probability $p$.

Can overcome measurement errors by repeating the measurements periodically.

Model becomes 2+1 dimensional.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.

- Some defects diffuse.
- Some charges can fuse.
- Some charges can nucleate.
- Some defects are missing, and assumed to be there.
- Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Fault tolerant threshold of roughly 1.8 %

Comparable to the 2.9% reported by Harrington et al. using slow decoders
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Results for Kitaev’s code
5. Extension to other codes
6. Fault tolerance
7. 2D Fault-Tolerant Quantum Cellular Automaton
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.

Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.
- Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.
Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.
Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.
Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.
Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

If particles were attracting each other, they would be confined and none of this would be necessary.
Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

- If particles were attracting each other, they would be confined and none of this would be necessary.
- Errors would be thermally suppressed (keep system cool).

Simulated confinement

Control unit at each site location to

1. Perform syndrome measurements
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Control unit

- Control unit holds value of local potential $V$ and $\nabla V$
- Measures presence of defect (syndrome)
- Updates potential $\nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho$
- Move particles according to force $F = -\nabla V$
Control unit

- Control unit holds value of local potential $V$ and $\nabla V$
- Measures presence of defect (syndrome)
- Updates potential
  \[ \nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho \]
- Move particles according to force $F = -\nabla V$
Control unit holds value of local potential $V$ and $\nabla V$

Measures presence of defect (syndrome)

Updates potential

$$\nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho$$

Move particles according to force $F = -\nabla V$
Control unit holds value of local potential $V$ and $\nabla V$

Measures presence of defect (syndrome)

Updates potential

$$\nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho$$

Move particles according to force $F = -\nabla V$
**Problem with EM over \( Z_2 \)**

**Problem with proper lattice scaling**
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

**RG decoding algorithm**
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

**Local equivalence between codes**
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance ?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
  - Universality of decoding algorithms
  - Enhanced fault tolerance ?
  - All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
  - True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

**RG decoding algorithm**
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

**Local equivalence between codes**
- Defines topological phases
- Universality of decoding algorithms
  - Enhanced fault tolerance ?
  - All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
  - True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance ?
  - All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
  - True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

**RG decoding algorithm**
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

**Local equivalence between codes**
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton
Decoding problem: infer defect worldline homology from "snapshots" of their configuration.

RG decoding algorithm
- Exponentially faster.
- Versatile (other codes, time/performance tradeoff).
- Higher threshold.
- Heuristic (Bravyi has proved a threshold... $10^{-22}$)
- Extends beyond 2D (Fault tolerance)

Local equivalence between codes
- Defines topological phases
- Universality of decoding algorithms
- Enhanced fault tolerance ?
- All 2D stabilizer codes topologically equivalent to Kitaev. (Chiral?)
- True for some subsystem codes as well.

Possible fault-tolerant 2D quantum cellular automaton