Composable Quantum Fault-Tolerance

Zhiyang He (Sunny)[1], Quynh Nguyen[2], Christopher Pattison[3,4]

[1] MIT, [2] Harvard, [3] Caltech and [4] UC Berkeley

Prelude: A Summer Day in Benasque

Fault-Tolerant Quantum Technologies (FTQT) Workshop in Benasque, Spain, 2024

Fault-Tolerant Quantum Technologies Workshop

Chris Pattison

Kenneth Brown

In breakout sessions, we are assigned into small groups and asked to write down questions we'd like to understand better through this workshop.

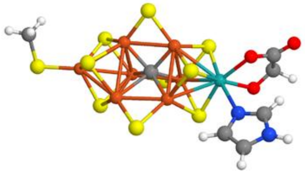
Ken asked the first question in our group:
What is Fault Tolerance?

The Promise of Fault-Tolerance

We want to run large quantum algorithms, but how large?

- Circuit volume = Width (space) × Depth (time), V = WD.
- Fault-tolerant execution → gate error rate at $\Theta(1/V)$.
- Factoring 2048-bit: around 10^{15} qubit-steps.

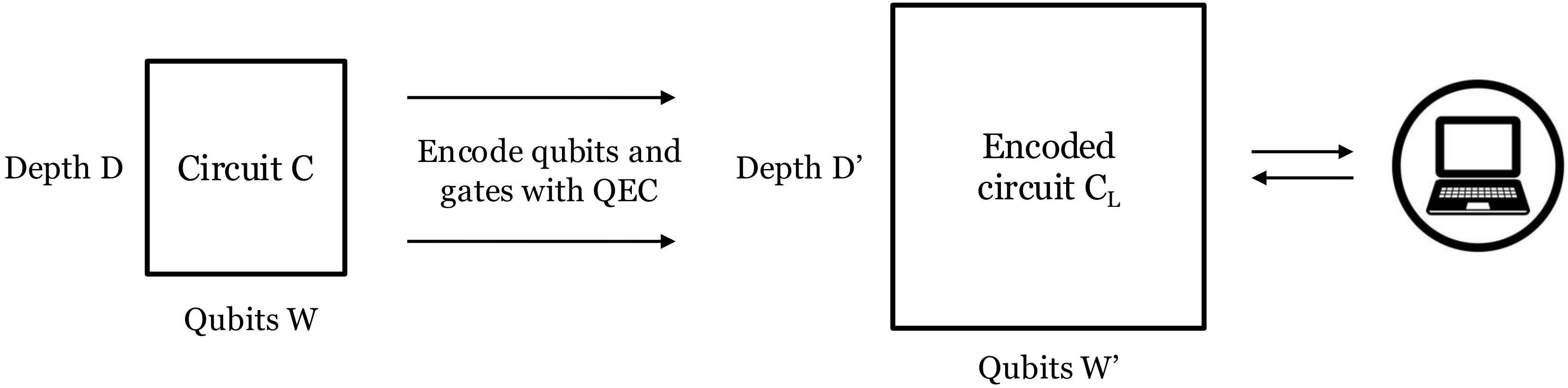
$n = p \cdot q$



Encoded Computation

Encode raw qubits/gates/error rates into error-corrected qubits/gates/error rates.

- Space overhead: W'/W, Time overhead: D'/D.
- Often measured in terms of physical circuit volume V, specifically log(V).



What is Quantum Fault-Tolerance?

Gold Standard: Existence of a constant noise threshold ϵ, such that if the physical noise strength is below ϵ, we can realize arbitrarily large computation fault-tolerantly (with certain overheads).

Threshold theorem for quantum computation: A quantum circuit containing $p(n)$ gates may be simulated with probability of error at most ϵ using

$$O(\text{poly}(\log p(n)/\epsilon)p(n)) \qquad (10.116)$$

gates on hardware whose components fail with probability at most p, provided p is below some constant *threshold*, $p < p_{\text{th}}$, and given reasonable assumptions about the noise in the underlying hardware.

Threshold Theorem from Nielsen & Chuang

Formally, this means that there is a family of encoding schemes (we will call them FT schemes) that achieve arbitrarily low encoded error rate at growing costs.

Great! So what's unclear about this? Looks like a reasonable theorem?

What is Quantum Fault-Tolerance?

Threshold theorem for quantum computation: A quantum circuit containing $p(n)$ gates may be simulated with probability of error at most ϵ using

$$O(\text{poly}(\log p(n)/\epsilon)p(n)) \tag{10.116}$$

gates on hardware whose components fail with probability at most p, provided p is below some constant *threshold*, $p < p_{\text{th}}$, and given reasonable assumptions about the noise in the underlying hardware.

So what's unclear about this? Looks like a reasonable theorem?

- What's the noise model? Locally stochastic? Independent Pauli error? Adversarial? Coherent Noise?
- What are the computation assumptions? Free & noiseless classical compute, or more thorough analysis?
- Most significant issue: Really hard to prove rigorously! Most papers propose new techniques, but do not prove threshold theorem. Instead, they justify FT in other ways.

FT or not FT, That is The Question

E.g.: Logical computation gadgets on LDPC codes. Lots of recent papers proposing many interesting ways of performing encoded computation on high-rate quantum LDPC codes.

Performing logical computation on QLDPC memory has been a long standing challenge in theory and in practice, with extensive research proposing many schemes [BB24, BCG+24, QWV23, ES24, ZSP+23, SPW24, BDET24, HKZ24, Lin24, GL24, MGF+25, BMD09, VB22, BVC+17, LB18, JO19, KP21, CKBB22, SKW+24, CB24, Cow24, CHRY24, WY24, SJOY24, IGND24, ZL24, CHWY25, HJOY23, XZZ+24, BGH+25, HCWY25, YSR+25].

They all justify fault-tolerance, but with different definitions, assumptions, and noise models. As a result, they cannot be easily combined in a rigorous fashion. This is a lack of composability.

What other notions of FT have we been using?

Quantum Fault-Tolerance Alignment Chart

	Lawful (Rigorously proved)	Neutral	Chaotic (Less based on proofs)
Good (Standard, easy to convince others)	Threshold under stochastic noise	Circuit has large spacetime distance	Experimental data
Neutral	Input/Output behavior of circuit under noise	Code has large distance	Numerical simulations
Evil (Uncommon, hard to derive and/or convince)	Threshold under coherent noise	Fault-Tolerance in Amortization	“This looks FT ☺”

Goal: Rigorous, Composable Quantum Fault-Tolerance

Our work: a mathematical framework for proving threshold theorems that

- Enables composition of complex, drastically distinct gadgets
- Re-establishes several well-known threshold results in the same language
- Separates the probabilistic analysis of noise model from the combinatorial analysis of error propagation.

By making fault-tolerance composable, future works can continue to prove new gadgets, and compose with existing library to derive rigorous threshold theorems for novel FTQC schemes.

We bring standard results closer to the lawful side

	Lawful (Rigorously proved)	Neutral	Chaotic (Less based on proofs)
Good (Standard, easy to convince others)	Threshold under stochastic noise	Circuit has large spacetime distance	Experimental data
Neutral	Input/Output Behavior of circuit under noise (e.g. Nielsen & Chuang)	Code has large distance	Numerical simulations
Evil (Uncommon, hard to derive and convince)	Threshold under coherent noise	Fault-Tolerance in Amortization	"This looks FT ☺"

And unifies many results in the lawful camp.

Goal: Rigorous, Composable Quantum Fault-Tolerance

Building on the groundwork established by the formalism, we prove (or reprove):

- Constant overhead FTQC via LDPC codes
- First written proof (?) of a threshold theorem using surface code + distillation
- A threshold theorem for surface code FTQC under coherent noise

Many possible future works, which we will discuss at the end.

We bring standard results closer to the lawful side

	Lawful (Rigorously proved)	Neutral	Chaotic (Less based on proofs)
Good (Standard, easy to convince others)	Threshold under stochastic noise	Circuit has large spacetime distance	Experimental data
Neutral	Input/Output Behavior of circuit under noise (e.g. Nielsen & Chuang)	Code has large distance	Numerical simulations
Evil (Uncommon, hard to derive and convince)	Threshold under coherent noise	Fault-Tolerance in Amortization	"This looks FT ☺"

And unifies many results in the lawful camp.

I. Faults and Bad Sets

II. Fault-Tolerant Gadgets

III. Weight Enumerators and Gadget Composition

IV. Assembly: Building Fault-Tolerant Schemes

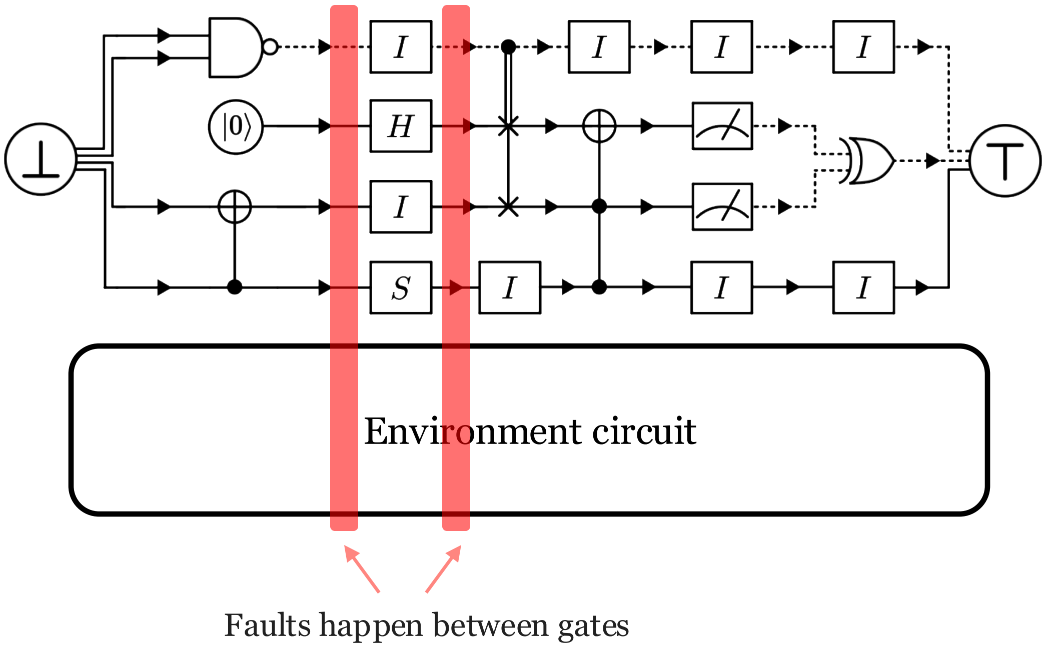
V. Discussions and Outlook

Circuits, Environment and Faults

To capture different noise and computation models, we consider a general model of circuit and fault:

- Circuit has quantum and classical registers, inputs, gates, and outputs.
- Faults in the circuit interact with an 'environment circuit'.
- Faults are inserted before and after gates.

While the faults are unitary in the global space, their actions when restricted to the logical workspace can be non-physical!



Circuits, Environment and Faults

What is a fault mathematically? We start with an error channel:

$$\mathscr{E}(\rho) = (1-p)\,\rho + p_1 K_1 \rho K_1^\dagger + p_2 K_2 \rho K_2^\dagger + \ldots$$

And insert errors between gates:

$$\cdots\; G_4 \circ \mathscr{E} \circ G_3 \circ \mathscr{E} \circ G_2 \circ \mathscr{E} \circ G_1(|x\rangle\langle x|)$$

$$= \sum_{\mu_1,\mu_2,\ldots} \left(p_{\mu_1} p_{\mu_2} \cdots\right) \left(\cdots U_4\, K_{\mu_3}\, U_3\, K_{\mu_2}\, U_2\, K_{\mu_1}\, U_1\, |x\rangle\langle x| \cdots\right)$$

These K are superoperators (interacting with external environment), they are the faults. Our final circuit looks like:

$$\widetilde{C}(|x\rangle\langle x|) = \sum_{\text{faults } \mathbf{f}} \Pr(\mathbf{f})\, C[\mathbf{f}](|x\rangle\langle x|)$$

We can then analyze $C[\mathbf{f}]$ as a fixed circuit, where the faults are inserted deterministically.

Capturing Faults by Sets

Given a fixed faulty circuit $C[\mathbf{f}]$, how does $\mathbf{f}$ affect the execution? Conceptually, the locations which $\mathbf{f}$ corrupt are critical: some failure locations can be corrected, while other cannot. We capture this intuition as follows:

Definition: Avoiding Sets

Given a set of locations/qubits [n], a family of bad sets $\mathcal{F}$ is a subset of the powerset of [n].

$$\mathcal{F} \subseteq P(\{1, \ldots, n\})$$

For a set $E \subseteq [n]$, we say that E is $\mathcal{F}$-avoiding if

$$\forall S \in \mathcal{F}, S \nsubseteq E.$$

Capturing Faults by Sets

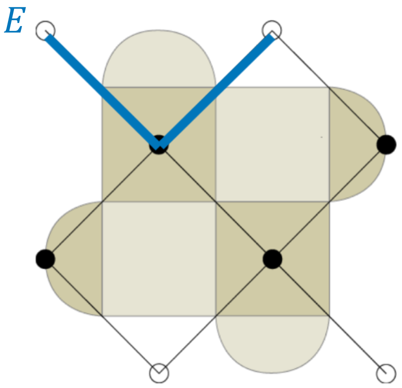
Definition: Avoiding Sets

Given a set of locations/qubits [n], a family of bad sets $\mathcal{F}$ is a subset of the powerset of [n].

$$\mathcal{F} \subseteq P(\{1, \dots, n\})$$

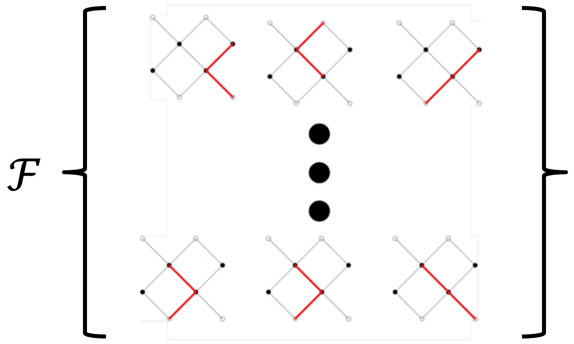
For a set $E \subseteq [n]$, we say that E is $\mathcal{F}$-avoiding if

$$\forall S \in \mathcal{F}, S \nsubseteq E.$$



E is $\mathcal{F}$-avoiding

Example: on a [n,k,d] error correcting code, define $\mathcal{F}$ to be all sets from which a minimum weight decoder cannot recover. Then all errors E that is $\mathcal{F}$-avoiding are recoverable, and can be considered ‘good errors’. We can picture this on a d=2 rotated surface code.



Capturing Faults by Sets

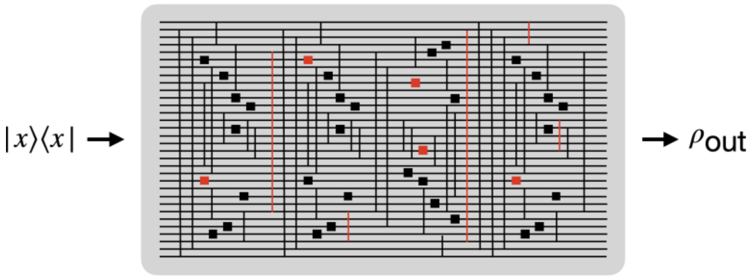
Definition: Avoiding Sets

Given a set of locations/qubits [n], a family of bad sets $\mathcal{F}$ is a subset of the powerset of [n].

$$\mathcal{F} \subseteq P(\{1, \dots, n\})$$

For a set $E \subseteq [n]$, we say that E is $\mathcal{F}$-avoiding if

$$\forall S \in \mathcal{F}, S \nsubseteq E.$$



$C_{\mathrm{FT}}[\mathbf{f}]$

Example: for a circuit C_{FT} that is encoding an ideal circuit C, [n] would be the spacetime locations where fault could occur. The bad sets $\mathcal{F}$ would be ‘witnesses of failed circuit execution’.

If $\mathbf{f}$ is $\mathcal{F}$-avoiding, then the circuit was ‘successful’ and correct:

$$C_{FT}[\mathbf{f}]\left(|x\rangle\langle x|\right) = (\text{const.}) \cdot C\left(|x\rangle\langle x|\right)$$

It’s hard to argue about the correctness of one big circuit. Let’s capture correctness modularly.

I. Faults and Bad Sets

II. Fault-Tolerant Gadgets

III. Weight Enumerators and Gadget Composition

IV. Assembly: Threshold Theorems

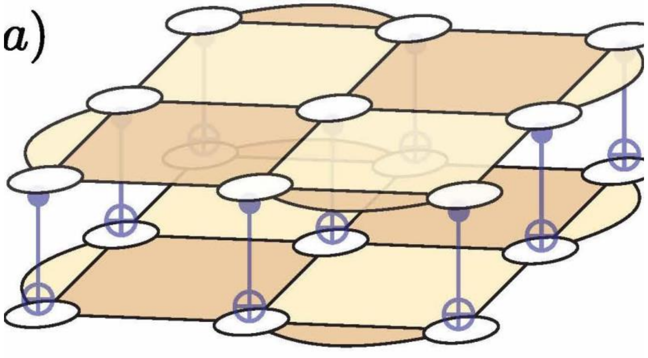
V. Discussions and Outlook

What's a Gadget?

Recall that there is a circuit C we wish to implement with low error rate, and we need to encode C into a bigger circuit, C_{FT}.

Every qubit of C will be encoded in error-correcting codes in C_{FT}. Similarly, every gate of C will be encoded into a gadget, which is a sub-circuit of C_{FT}.

- Example: a CNOT gate in C may be encoded into a transversal CNOT in C_{FT}.

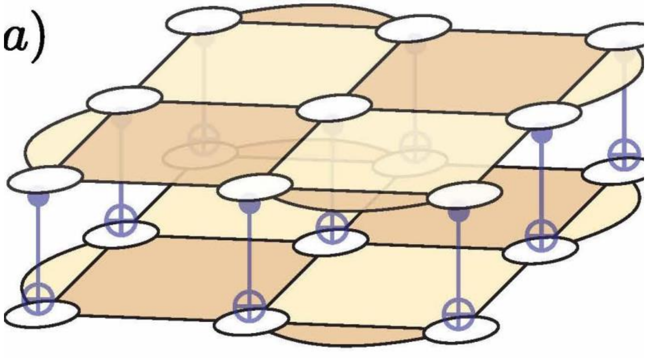


Transversal CNOT. Fig. credit 2408.01393

What's a Gadget?

Concretely:
- ➢ In C, a gate g acts on some set of qubits Q.
- ➢ In C_{FT}, the corresponding gadget C_g acts on error-correcting code blocks which encode Q.

When there is no error/fault, C_g correctly performs g on encoded qubits.
What if there are errors? How can we say C_g is FT?



Transversal CNOT. Fig. credit 2408.01393

Errors on Input and Output Codes

Let's start by looking at the input/output of a gadget. They are states in quantum codes, impacted by errors on qubits.

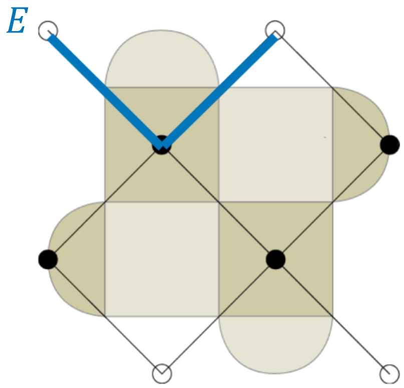
Definition: Code Type

A code is specified by two pieces of data:

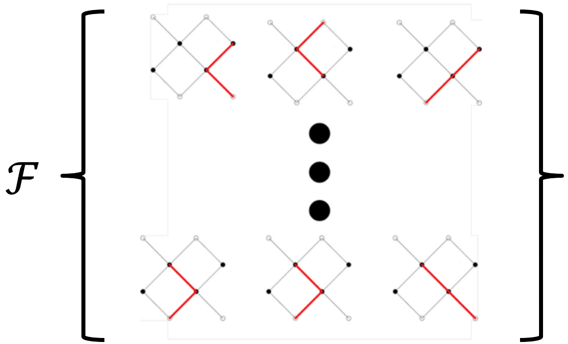
- ➢ An encoding unitary **enc**, and
- ➢ A family of bad error supports $\mathcal{B}$, such that if an error E is $\mathcal{B}$-avoiding, then E can be recovered using the encoding unitary **enc**.

We say that (**enc**, $\mathcal{B}$) is a code type.

For a gadget, a 'good input' would be one where the error E is $\mathcal{B}$-avoiding. Otherwise it's a 'bad input'.



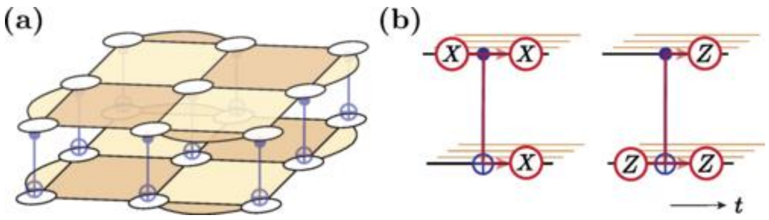
E is $\mathcal{F}$-avoiding



Fault-Tolerant Gadgets

Besides input errors, faults can corrupt the physical gates in the gadgets, introducing more noise into the system.

We say that a gadget is FT if, under bounded noise, it maps good input to good output.



Transversal CNOT and error propagation.
Fig. credit 2408.01393

Definition: FT Gadget (Part 1)

A gadget for a gate g is specified by a circuit C_g with locations L and a family of bad fault paths $\mathcal{F} \subseteq P(L)$.

Given a 'good fault' **f** that is $\mathcal{F}$-avoiding:

1. On an input state enc(ρ) with a 'good error', $C_g[\mathbf{f}]$ outputs the state enc(g(ρ)) with a (different) 'good error'.

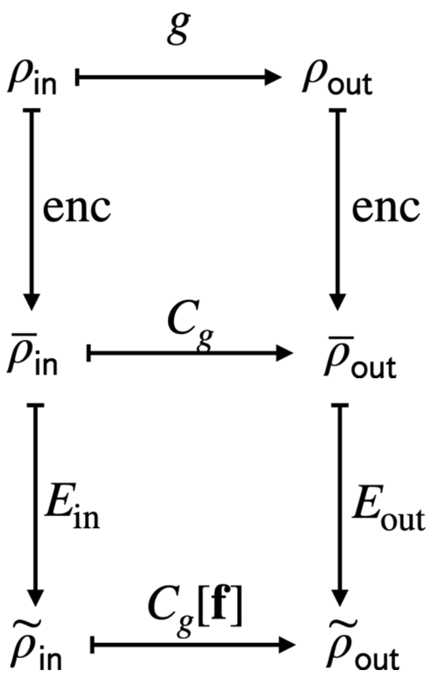
Fault-Tolerant Gadgets

Definition: FT Gadget (Part 1)

A gadget for a gate g is specified by a circuit C_g with locations L and a family of bad fault paths $\mathcal{F} \subseteq P(L)$.

Given a 'good fault' **f** that is $\mathcal{F}$-avoiding:

1. On an input state enc(ρ) with a 'good error', C_g[**f**] outputs the state enc(g(ρ)) with a (different) 'good error'.



Remarks:

- This is a completely combinatorial definition! Everything is bounded in terms of sets.
- Different gadgets with compatible input/output codes can be easily composed! More on that later.

Fault-Tolerant Gadgets

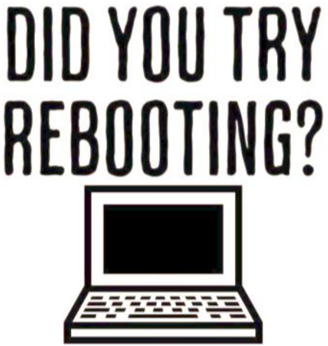
What if the input state is bad?

Definition: FT Gadget (Part 2)

Given a 'good fault' **f** that is $\mathcal{F}$-avoiding:

1. On an input state enc(ρ) with a 'good error', C_g[**f**] outputs the state enc(g(ρ)) with a (different) 'good error'.

2. On an input state enc(ρ) with a 'good error', C_g[**f**] outputs enc(σ) with a 'good error', where σ can be arbitrary.



We call this second condition 'friendliness'. This is a reset mechanism that catches irrecoverable errors. Not all gadgets need to be friendly – this is an optional property.

Gadgets on Fire

What if the fault **f** is bad?

- The gadget failed, and there is no guarantee. The output state is automatically treated as ‘bad’.

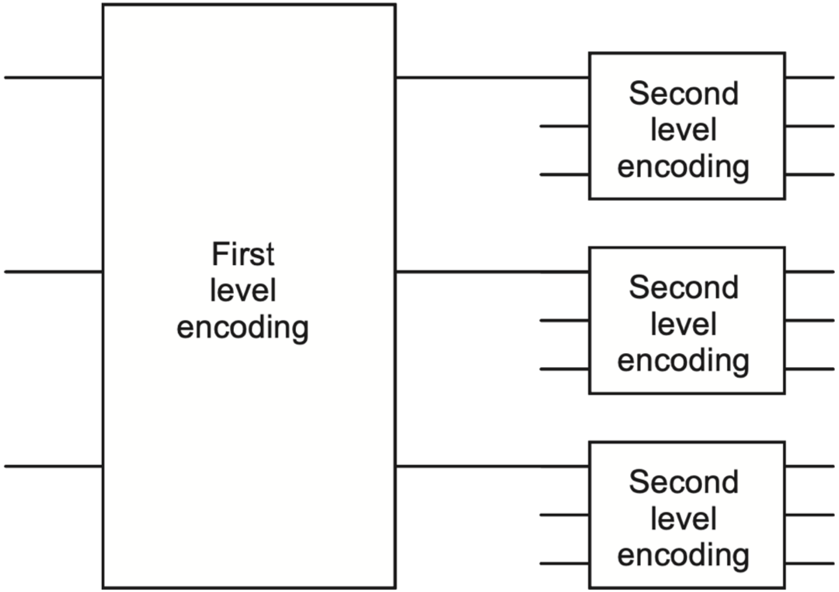
However, that doesn’t always mean failure for the global computation – a local gadget failure may be corrected by other successful gadgets, especially when we do recursive simulation!

Recursive Simulation

Concatenation: Given circuit C, we first encode into C_{FT}^1, then further encode C_{FT}^1 into C_{FT}^2, so on and so forth.

- Time-tested idea in classical and quantum error correction.
- Failure of gadgets in C_{FT}^2 correspond to failure of gates in C_{FT}^1, which is correctable in C_{FT}^1.

This recursive simulation method, sometimes called level reduction, can be derived using our definitions (see Theorem 4.11).



Concatenated encoding scheme, Nielsen & Chuang

Fault Tolerance as a Combinatorial Property

Challenge: $\mathcal{F}$ can be very complicated for a big circuit, this definition is useless if we can't properly describe $\mathcal{F}$!

- How can we build these bad fault paths and bound their probabilities?
- What happen to these bad fault paths when we compose gadgets, or do recursive simulation?

Definition: FT Gadget

A gadget for a gate g is specified by a circuit C_g with locations L and a family of bad fault paths $\mathcal{F} \subseteq P(L)$.

Given a 'good fault' **f** that is $\mathcal{F}$-avoiding:

1. On an input state enc(ρ) with a 'good error', C_g[**f**] outputs the state enc(g(ρ)) with a (different) 'good error'.

2. On an input state enc(ρ) with a 'good error', C_g[**f**] outputs enc(σ) with a 'good error', where σ can be arbitrary.

I. Faults and Bad Sets

II. Fault-Tolerant Gadgets

III. **Weight Enumerators and Gadget Composition**

IV. Assembly: Threshold Theorems

V. Discussions and Outlook

Weight Enumerators

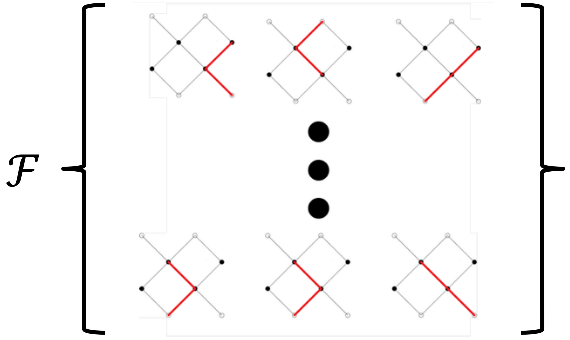
Definition: Weight enumerator

Consider a family of bad sets $\mathcal{F} \subseteq P(\{1, \dots, n\})$. Let

$$A_w^{(\mathcal{F})} = (\#\text{ of elements of weight } w \text{ in } \mathcal{F})$$

The weight enumerator polynomial of $\mathcal{F}$ is

$$W(\mathcal{F}; x) = \sum_{w=0}^{\infty} A_w^{(\mathcal{F})} x^w$$



$W(\mathcal{F}; x) = 18x^2$

Upper Bounding Bad Errors Probability

Weight enumerators and bad sets make the interface that separates the combinatorial FT analysis from the probabilistic distribution of faults.

Consider $E \subseteq [n]$ which has a locally stochastic distribution. That means for any $S \in [n]$, we have

$$\Pr(S \subseteq E) \le \epsilon^{|S|}$$

Then

$$\Pr(E \text{ is not } \mathcal{F}\text{−avoiding}) \le \sum_{S \in \mathcal{F}} \epsilon^{|S|} = W(\mathcal{F}; \epsilon)$$

In other words, the weight enumerator polynomial upper bounds the probability of ‘bad error’ occurring. This bounding works for most noise models, such as adversarial, locally stochastic, and even coherent noise.

Definition: Weight enumerator

Consider a family of bad sets $\mathcal{F} \subseteq P(\{1, \dots, n\})$. Let

$$A_w^{(\mathcal{F})} = (\#\text{ of elements of weight } w \text{ in } \mathcal{F})$$

The weight enumerator polynomial of $\mathcal{F}$ is

$$W(\mathcal{F}; x) = \sum_{w=0}^{\infty} A_w^{(\mathcal{F})} x^w$$

Composing Gadgets and Polynomials

Consider two gadgets with bad faults $\mathcal{F}$ and $\mathcal{G}$. Define

$$\mathcal{F} \boxplus \mathcal{G} \coloneqq \text{Sets containing a subset from } \mathcal{F} \text{ or } \mathcal{G},$$
$$\mathcal{F} \circledast \mathcal{G} \coloneqq \text{Sets containing a subset from } \mathcal{F} \text{ and } \mathcal{G}.$$

It's easy to show that

$$W(\mathcal{F} \boxplus \mathcal{G};\ x) = W(\mathcal{F};x) + W(\mathcal{G};x),$$
$$W(\mathcal{F} \circledast \mathcal{G};\ x) = W(\mathcal{F};x) \times W(\mathcal{G};x).$$

For an error $E \subseteq [n]$,

$$\Pr\begin{pmatrix} E \text{ is not } \mathcal{F}\text{−avoiding} & \\ E \text{ is not } \mathcal{G}\text{−avoiding} & \textbf{\textit{or}} \end{pmatrix} \leq W(\mathcal{F};\epsilon) + W(\mathcal{G};x),$$

$$\Pr\begin{pmatrix} E \text{ is not } \mathcal{F}\text{−avoiding} & \\ E \text{ is not } \mathcal{G}\text{−avoiding} & \textbf{\textit{and}} \end{pmatrix} \leq W(\mathcal{F};\epsilon) \times W(\mathcal{G};x),$$

Definition: Weight enumerator

Consider a family of bad sets $\mathcal{F} \subseteq P(\{1, \ldots, n\})$. Let

$$A_w^{(\mathcal{F})} = (\#\text{ of elements of weight } w \text{ in } \mathcal{F})$$

The weight enumerator polynomial of $\mathcal{F}$ is

$$W(\mathcal{F};x) = \sum_{w=0}^{\infty} A_w^{(\mathcal{F})}\, x^w$$

Composing Gadgets and Polynomials

Consider two gadgets with bad faults $\mathcal{F}$ and $\mathcal{G}$.

$$\Pr\begin{pmatrix} E \text{ is not } \mathcal{F}\text{–avoiding} & \boldsymbol{or} \\ E \text{ is not } \mathcal{G}\text{–avoiding} & \end{pmatrix} \leq W(\mathcal{F};\epsilon) + W(\mathcal{G};x),$$

$$\Pr\begin{pmatrix} E \text{ is not } \mathcal{F}\text{–avoiding} & \boldsymbol{and} \\ E \text{ is not } \mathcal{G}\text{–avoiding} & \end{pmatrix} \leq W(\mathcal{F};\epsilon) \times W(\mathcal{G};x).$$

So long as the gadgets are independent in the combinatorial sense (not overlapping), the above equations hold.

- The noise distribution can be arbitrary! Correlated, different for each gadget, train running near optical devices…

This is the key property of our definitions: composable gadgets with no assumptions except for disjointness.

Definition: Weight enumerator

Consider a family of bad sets $\mathcal{F} \subseteq P(\{1,\ldots,n\})$. Let

$$A_w^{(\mathcal{F})} = (\#\text{ of elements of weight } w \text{ in } \mathcal{F})$$

The weight enumerator polynomial of $\mathcal{F}$ is

$$W(\mathcal{F};x) = \sum_{w=0}^{\infty} A_w^{(\mathcal{F})}\, x^w$$

Concatenate Gadgets and Polynomials

We also have an operation defined for concatenated gadgets

$$\mathcal{F} \bullet \{\mathcal{S}_i\}_i := \boxplus_{f \in \mathcal{F}} \left(\mathbb{I}_{n \in f}[\mathcal{S}_n] \circledast \mathbb{I}_{n-1 \in f}[\mathcal{S}_{n-1}] \circledast \cdots \circledast \mathbb{I}_{1 \in f}[\mathcal{S}_1]\right).$$

The weight enumerator is bounded as

$$W(\mathcal{F} \cdot \{S_i\}_i;\ x) = W\big(\mathcal{F};\ W(\{S_i\}_i; x)\big)$$

More details can be found in the paper.

I. Faults and Bad Sets

II. Fault-Tolerant Gadgets

III. Weight Enumerators and Gadget Composition

IV. Assembly: Threshold Theorems

V. Discussions and Outlook

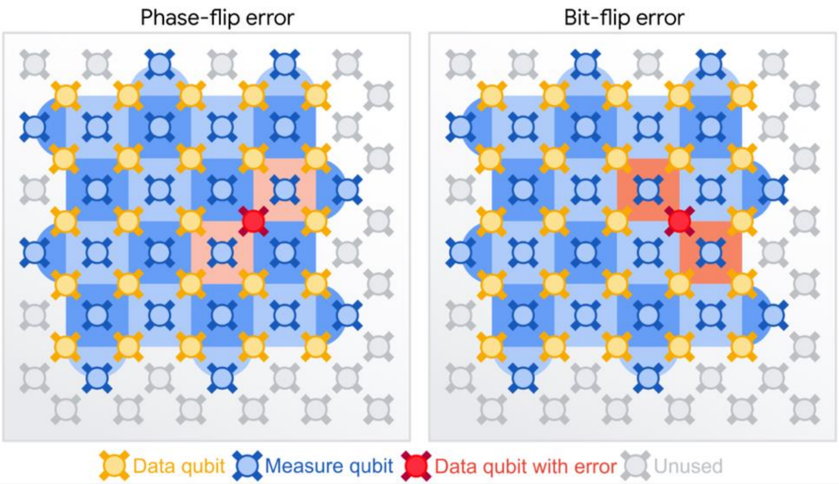
Gadget Example: Error Correction

Let's consider gadgets on surface codes.

- ➢ SC^d denote surface code of distance d
- ➢ $U_c^d \subseteq [d^2]$, for a fraction $c \in [0,1]$, is defined as all sets of qubits which contain more than a c fraction of a connected component of size $\geq d$.
- ➢ If above definition is unclear, just know that

  bigger $c \Rightarrow$ more errors.

The standard error correction gadget, where we measure stabilizers for d rounds and correct for errors, can be written down as usual.



Surface code with error on data qubit, Fig. by Google

Gadget Example: Error Correction

Let's consider gadgets on surface codes.

- SC^d denote surface code of distance d
- $U_c^d \subseteq [d^2]$, for a fraction $c \in [0,1]$, is defined as all sets of qubits which contain more than a c fraction of a connected component of size $\geq d$.
- If above definition is unclear, just know that

  bigger c $\Rightarrow$ more errors.

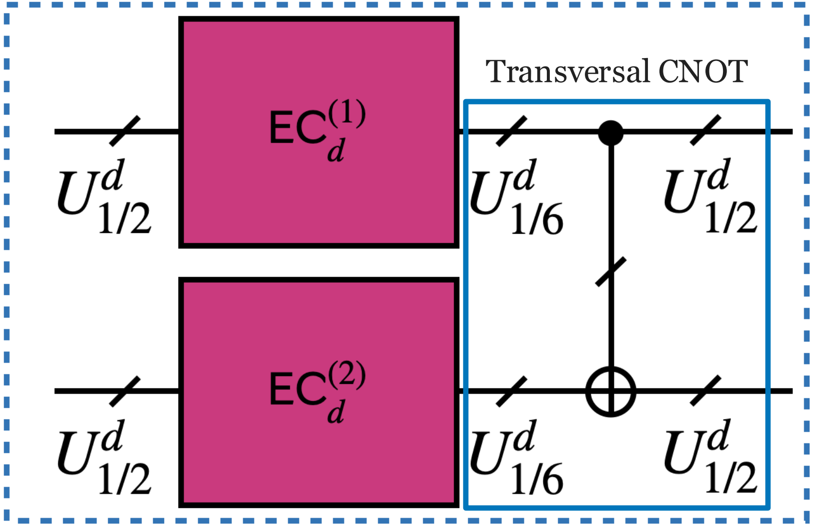
The standard error correction gadget, where we measure stabilizers for d rounds and correct for errors, can be written down as usual.

- Input error is suppressed by the EC gadget.
- Gadget has bad faults $\mathcal{F}_{\mathrm{EC}}$, which has weight enumerator bounded by

$$W(\mathcal{F}_{\mathrm{EC}};\, x) \leq poly(d) \cdot e^{-d}$$

when x is below a constant threshold.

Gadget Example: CNOT



Gadget for CNOT, built from smaller gadgets

Let's build the bad faults of this gadget by composition.

- $\mathcal{F}_{\mathrm{EC}}(1), (2)$ is bad faults for EC gadget (1), (2);
- Let $\mathcal{F}_{\mathrm{CNOT}}$ be bad faults for transversal CNOT;

The bad faults for this whole gadget is simply

$$\mathcal{F} \coloneqq \mathcal{F}_{\mathrm{EC}}(1) \boxplus \mathcal{F}_{\mathrm{EC}}(2) \boxplus \mathcal{F}_{\mathrm{CNOT}}$$

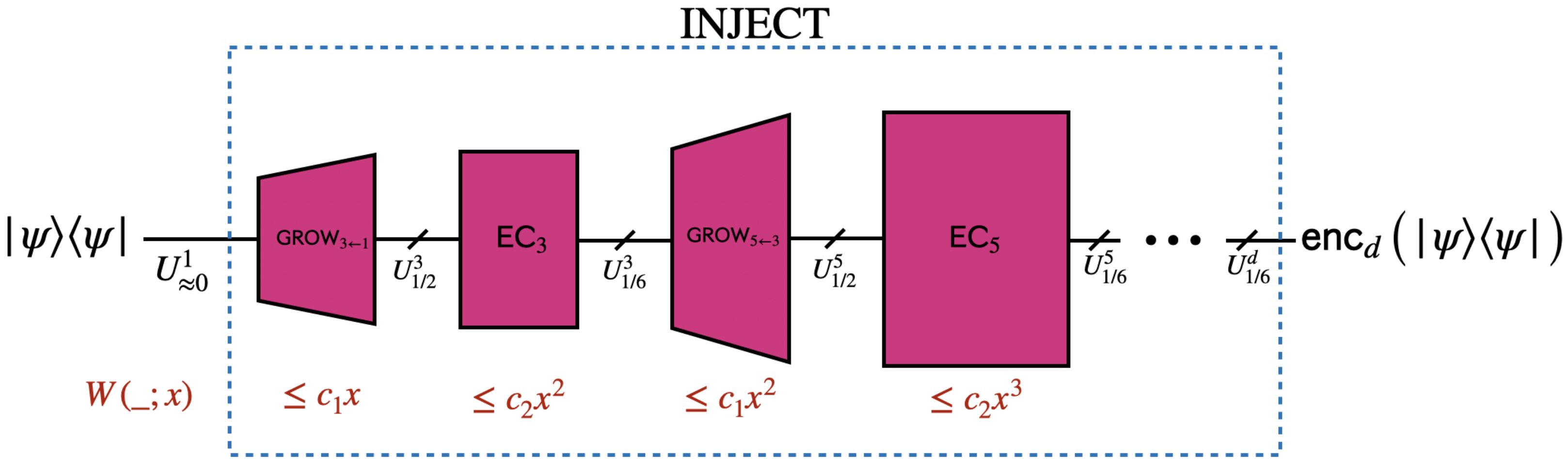
And its weight enumerator is

$$W(\mathcal{F}; x) = 2 \times W(\mathcal{F}_{\mathrm{EC}};\ x) + W(\mathcal{F}_{\mathrm{CNOT}}; x).$$

Since each of the component polynomials have constant threshold, the whole gadget has constant threshold.

Gadget Example: State Injection

State injection: prepare a noisy logical state in the code. For surface code, we can start with the physical qubit, and iterate the following procedure: grow the lattice length by 2, and perform error correction.



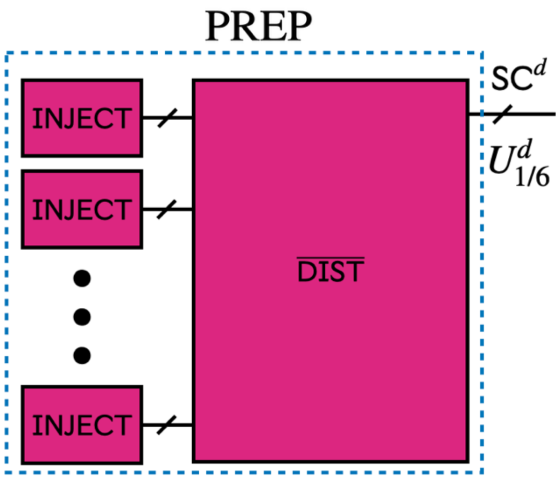
The weight enumerator can be calculated from those of the smaller components, and we have

$$W(\mathcal{F}_{\text{INJECT}}; x) \le O(x).$$

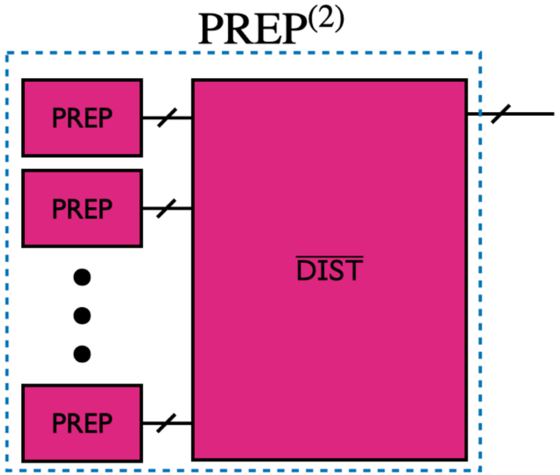
Gadget Example: Magic State Distillation

Given state injection, we can inject magic states and perform magic state distillation (MSD).

- This procedure can be applied iteratively for multiple rounds;
- Weight enumerators can be calculated using ⊞ and ⊛, more details in paper.



One round MSD



Two-round MSD

Threshold Theorems: Surface Code FTQC

Theorem 7.11 (Threshold theorem for surface code quantum computation)**.** There exists a constant $\epsilon_* \in (0,1)$ such that, for any Clifford+T circuit C of width W and depth D and any $\epsilon \in (0,1)$, there exists a circuit $\overline{C}$ that is a fault-tolerant gadget for C with bad fault paths $\mathcal{F}$.

Let $V = \frac{WD}{\epsilon}$. Then, $\overline{C}$ has width $\overline{W}$ and depth $\overline{D}$ satisfying the bounds

$$\overline{W} = O(W \log^{5.91}(V) \text{polyloglog}(V)) \tag{7.42}$$

$$\overline{D} = O(D \log^{2}(V) \text{polyloglog}(V)) \ . \tag{7.43}$$

On $x \in [0, \epsilon_*]$, the weight enumerator of $\mathcal{F}$ satisfies the bound

$$\mathcal{W}(\mathcal{F};\ x) \leq \epsilon \ . \tag{7.44}$$

- Constructed using transversal Cliffords and MSD. First written proof of surface code FTQC threshold (?)
- Constant threshold against locally stochastic noise, inverse-log threshold against coherent noise (new!).
- Building the formalism took us 60 pages; this Theorem took 10; handling coherent noise took 3;

Threshold Theorems: Constant Space Overhead FTQC

Theorem 7.21 (Constant space overhead threshold theorem)**.** There exists a constant $\epsilon_* \in (0,1)$ such that, for any Clifford+T circuit C with classical input and classical output of width W, depth D, and $\epsilon \in (0,1)$ [redacted] there exists a circuit $\overline{C}$ that is a fault-tolerant gadget for C with trivial input and output and bad fault paths $\mathcal{F}$. Let $V = \frac{WD}{\epsilon}$. Then, $\overline{C}$ has width $\overline{W}$ and depth $\overline{D}$ satisfying the bounds

$$\overline{W} = O(W) \tag{7.83}$$

$$\overline{D} = O(D \log^{8.91}(V) \mathrm{polyloglog}(V)) \ . \tag{7.84}$$

On $x \in [0, \epsilon_*]$, the weight enumerator of $\mathcal{F}$ satisfies the bound

$$\mathcal{W}(\mathcal{F};\ x) \leq \epsilon \ . \tag{7.85}$$

- Reproduced construction from [Gottesman 2014] and [Tamiya-Koashi-Yamasaki 2024]
- This constructions utilizes previous surface code theorem; took us 5 pages to prove. Average FT threshold paper takes 40~100 pages!

Threshold Theorems: Almost-Log Overhead FTQC

Theorem 3.23 (Main result). *There exists a function* $f(x)$ [redacted] *and a value* $\epsilon_* \in (0,1)$ *such that for any* $\epsilon_L \in (0,1)$ *and (Clifford+*CCZ*) classical input / classical output quantum circuit* C *with width* W *and depth* D [redacted]
There is a corresponding efficiently constructable classical input / classical output quantum circuit $\overline{C}$ *with width* $\overline{W}$ *and depth* $\overline{D}$ *satisfying*

$$\frac{\overline{W}}{W} = O_{W\to\infty}(1) \tag{35}$$

$$\frac{\overline{D}}{D} = O_{W\to\infty}\left(\left(\log \frac{WD}{\epsilon_L}\right)^{1+o(1)}\right) \tag{36}$$

and using auxiliary $O(1)$*-time classical computation per quantum time step, such that the following guarantees hold. For a random physical fault* $\mathbf{f}$ *distributed according to* ϵ*-locally stochastic faults model with* $\epsilon \in [0, \epsilon_*]$*, the output distribution of* $\overline{C}$ *subject to* $\mathbf{f}$ *is* ϵ_L*-close in TVD to the output distribution of* C*.*

- Result from [Nguyen-Pattison 2024], convoluted assembly of many components. This is one of the most asymptotically optimal FTQC scheme as of now.

I. Faults and Bad Sets

II. Fault-Tolerant Gadgets

III. Weight Enumerators and Gadget Composition

IV. Assembly: Threshold Theorems

V. Discussions and Outlook

Future Directions

❖ So what is fault tolerance?

Fault tolerance is a combinatorial bound on the fault propagation behavior of a circuit.

Through avoiding sets and weight enumerators, the Composable FT formalism separates the combinatorial analysis of fault propagation from the probabilistic analysis of noise distribution.

We hope our definitions could be groundwork that inspires future proofs of fault-tolerance. Lots of future works that can be done:

- Threshold proofs for LDPC code surgery operations;
- Constant threshold against coherent noise;
- Two-dimensional FTQC via concatenating QLDPC codes;
- And many more, talk to us if you're interested in proving FT!

Composable Quantum Fault-Tolerance

Zhiyang He (Sunny)[1], Quynh Nguyen[2], Christopher Pattison[3,4]

[1] MIT, [2] Harvard, [3] Caltech and [4] UC Berkeley