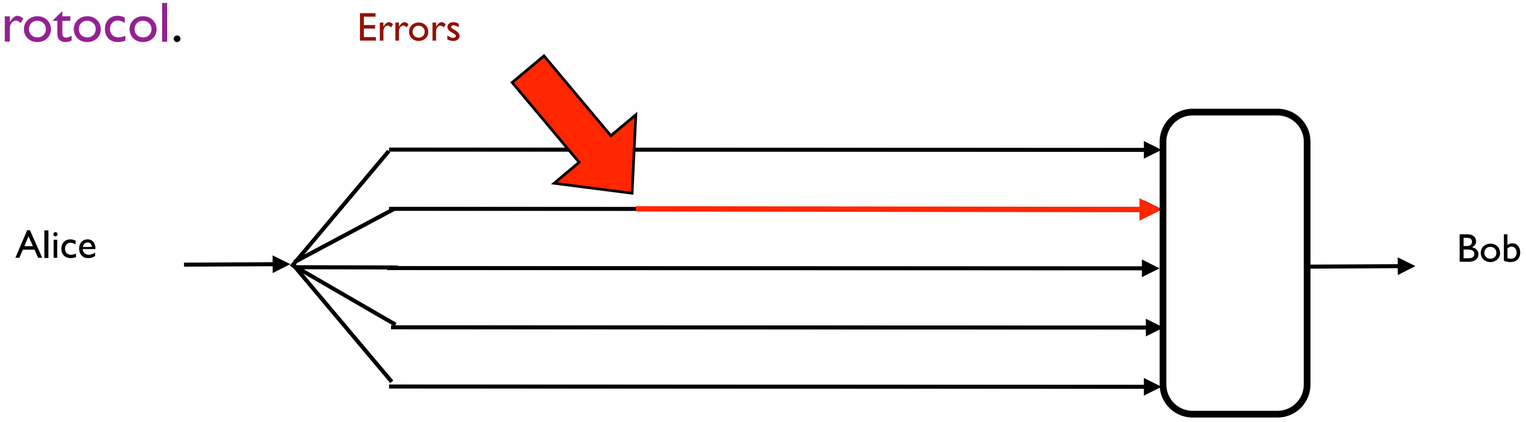
Fault Tolerance with Dynamical Codes

Daniel Gottesman
University of Maryland

Fault Tolerance

In order to build a large quantum computer, we expect that we will need a fault-tolerant protocol.

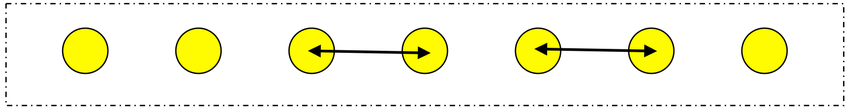


The first component of a fault-tolerant protocol is a quantum error-correcting code, which encodes some qubits using a larger number of qubits in such a way that errors can be identified and corrected.

We also need a way of performing gates (and state preparation and measurement and error correction) on states while they are encoded.

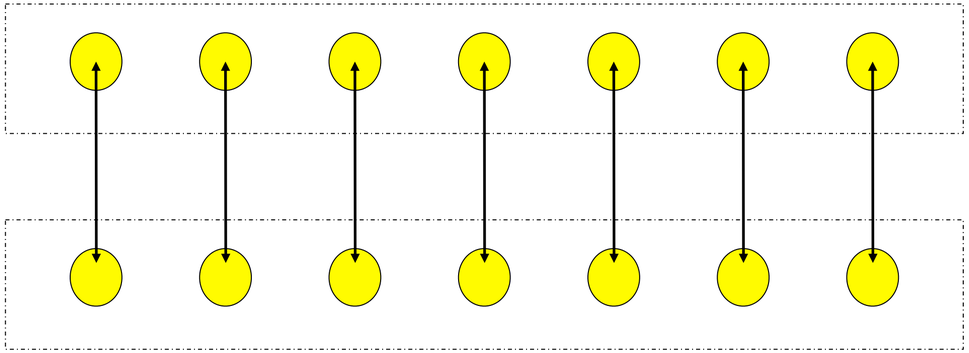
Fault-Tolerant Gadgets

To achieve fault tolerance, we need that faults in single physical gates don't destroy the encoded data. We want to avoid bad error propagation, where one gate error can cause multiple qubit errors in the same code block.



Bad! One gate error can ruin two qubits.

Good! A single gate error can cause two qubits to have errors, but they are in different blocks of the code, so can be corrected.



Transversal gates are always fault-tolerant. For instance, applying H to every qubit of the 7-qubit code does logical H and applying CNOT transversally to two blocks of the 7-qubit code does the logical CNOT.

[S96]

Limits of Fault Tolerance

There is a threshold for fault-tolerant quantum computation: Below a certain error rate per gate or time step, we can correct errors faster than they occur and can use fault tolerance to make arbitrarily large quantum computers. Above the threshold, errors accumulate faster than they can be corrected by a fault-tolerant protocol.

Lower bound on the threshold: Around 3%

Upper bound on the threshold: 45%

Existing fault-tolerant protocols need many extra qubits.

Upper bound on the overhead: # Physical qubits/# logical qubits > 20

Lower bound on the overhead: # Physical qubits/# logical qubits > 1

[K05, BCL+06, BCG+24, XBP+23]

New Approaches to Fault Tolerance

We still have plenty of room for improvement of fault tolerant protocols. New techniques and new ways of thinking about fault tolerance would be helpful.

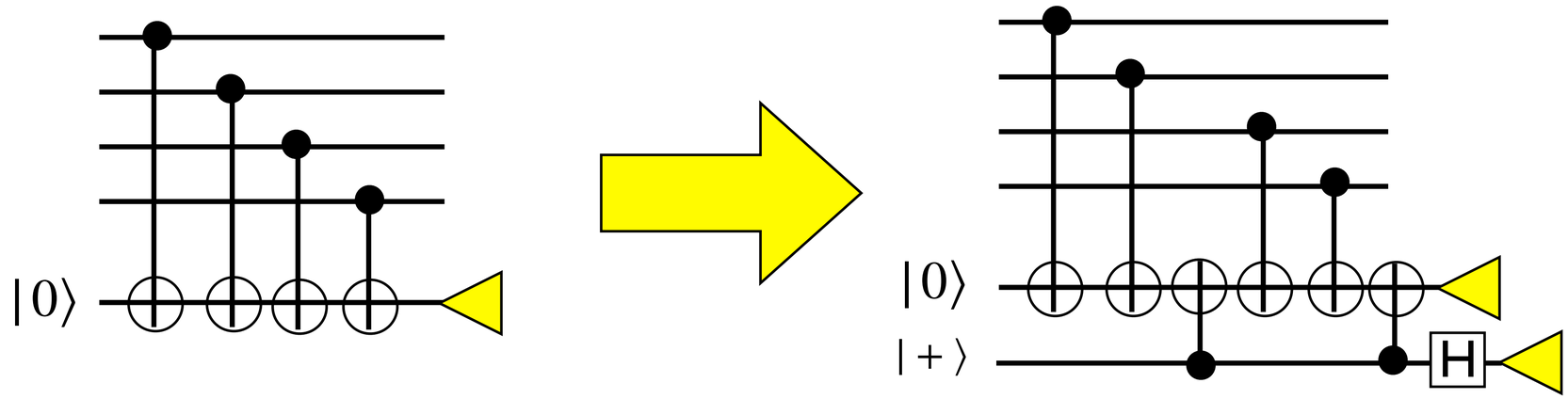
One such new approach would be to think about fault-tolerant protocols in a picture treating space and time on a more equal footing. Consider the following fault-tolerant tools:

- Flag fault tolerance
- Code deformation
- Floquet codes
- Nickerson-Bombin states

Together they point the way to this idea.

Flag Qubits

Traditional approaches to fault tolerance insist that we should arrange our circuits so one faulty gate can only cause one error in a block of the code.



Flag fault tolerance relaxes that constraint, instead allowing gates which cause error propagation within a code block but adding extra checks so that we can identify the location which originally caused the error. If we know where the error began, we know what kind of propagation it has undergone and we can correct it even though it is on multiple qubits.

Lesson: Error propagation is OK if we know the faulty gate.

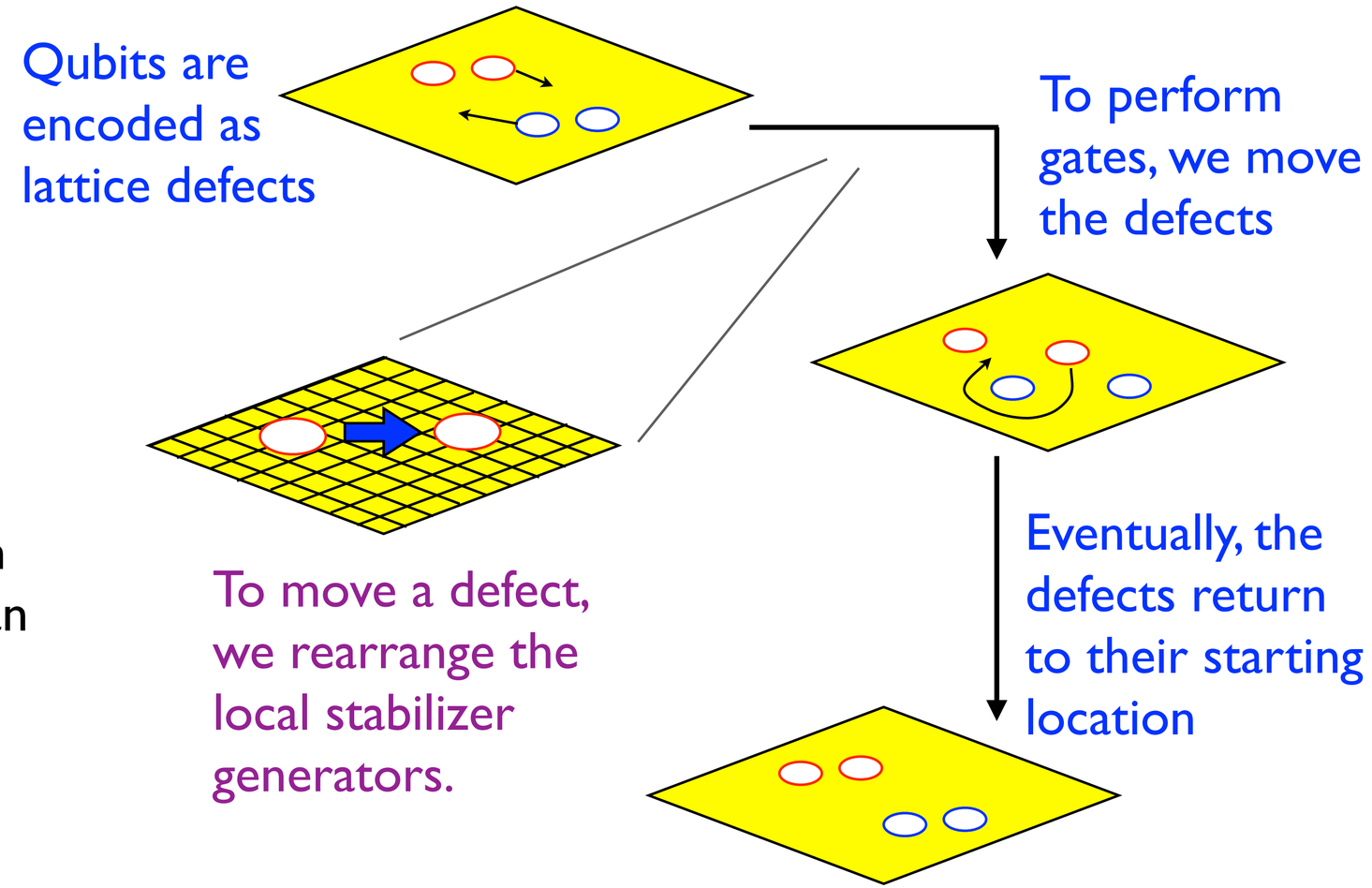
[CR17, SYK+22]

Code Deformation

What does it mean to do gates via code deformation?

We deform through a series of codes, eventually returning to the original code. By doing a topologically non-trivial path, we can do an encoded gate.

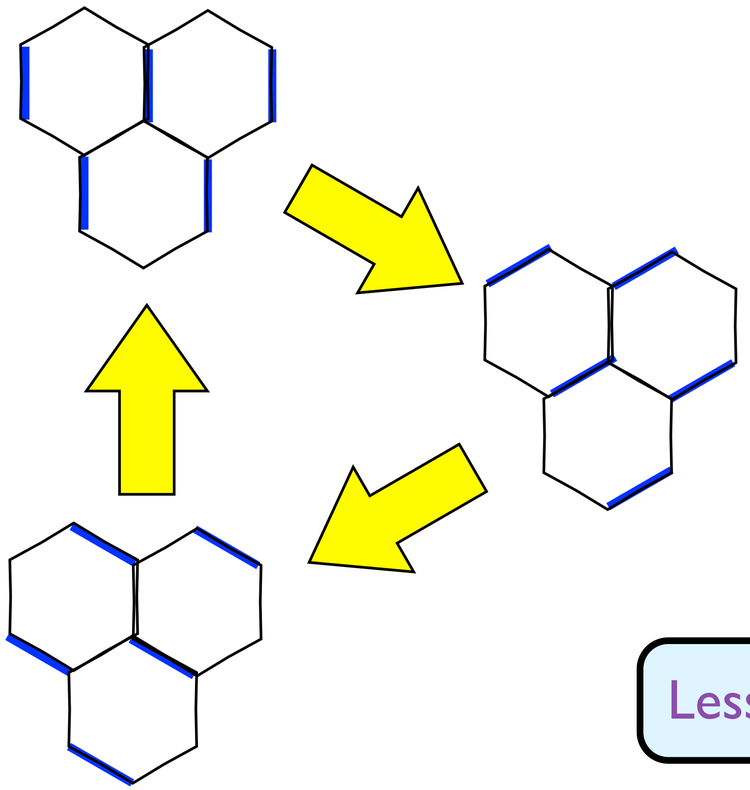
This example shows how to do it for surface codes, but in fact, any fault tolerant gate can be thought of as a code deformation.



[RH06, GZ13]

Floquet Codes

If we are doing gates by shifting between a sequence of codes, why do we insist that one of them is the "right" code and the others are just temporary stopping points?



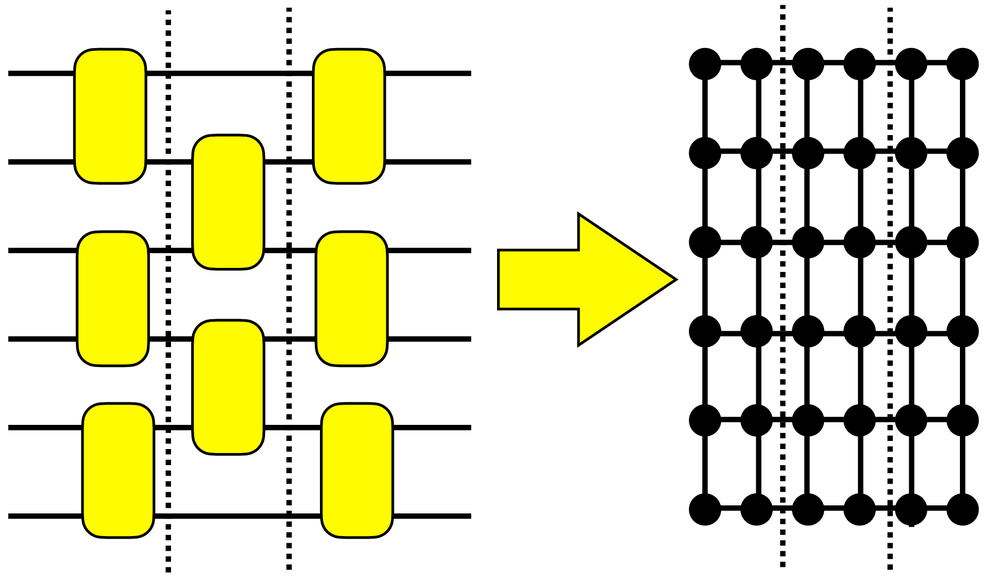
Floquet codes dispense with the idea of a single home code and instead cycle through a sequence of codes.

For instance, the honeycomb code implements something equivalent to a surface code by measuring sequences of two-qubit operators to learn error syndromes and shift between codes.

Lesson: The code can change with time.

[HH21]

Nickerson-Bombin States

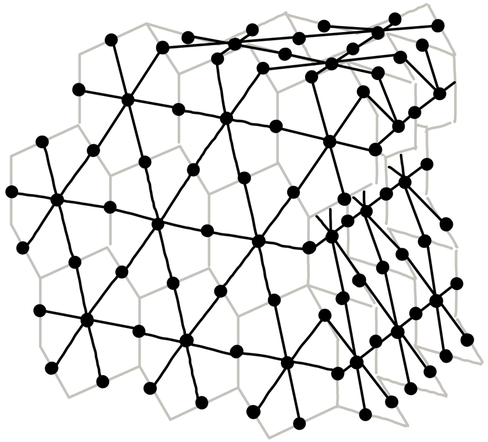


In measurement-based computation, we start with a large entangled state (a cluster state) and make a sequence of single-qubit measurements. We can convert any quantum circuit into an appropriate measurement pattern, with the measurements arranged into layers, one for each time step in the circuit.

But in the measurement-based model, there is no requirement to have such layers, and Nickerson and Bombin found states and measurement patterns with improved fault tolerance that do not have a natural breakdown into time steps.

Lesson: Look at space and time together.

[NB18]

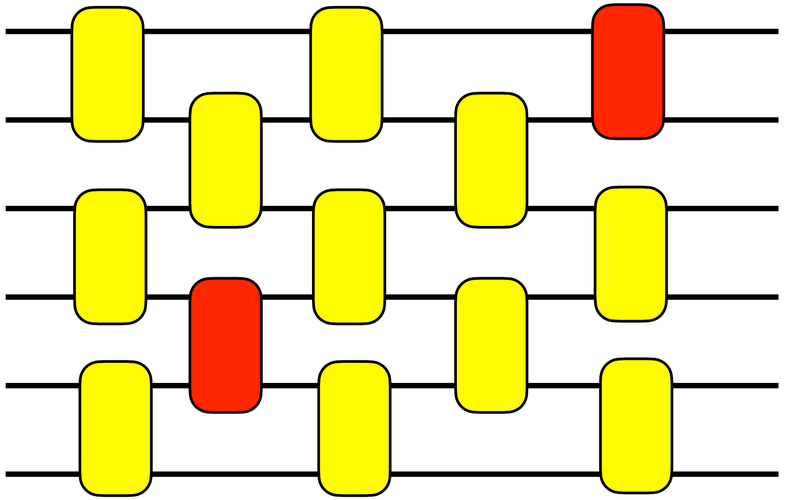


(from arXiv:1810.09621 [quant-ph])

Lessons Reviewed

Let's put this together:

- Error propagation is OK if we know the faulty gate.
- The code can change with time.
- Look at space and time together.



So, we'd like to:

- Look at the fault-tolerant circuit as a whole.
- Not care about the instantaneous code per se.
- Identify the location *and* time of each error.

Corollaries:

- The distinction between code qubits, ancilla qubits, and flag qubits is gone. There are just qubits, and really just spacetime locations, in the circuit.
- Gates are important for how they help us gather information, not for how they control error propagation.

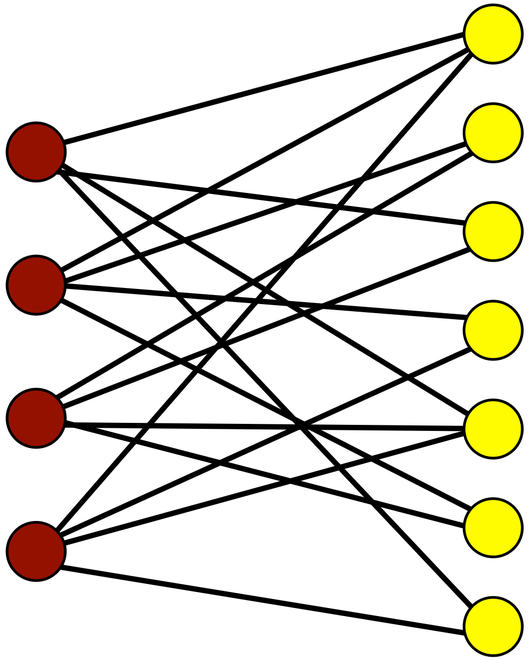
[arXiv:2403.04163 [quant-ph]]

Benefit of Dynamical Fault Tolerance

Why do we want to do this?

For one thing, it gives us more freedom to design fault-tolerant protocols. But there is a particular reason to want that freedom.

The goal in error correction is to collect information about the errors. Ideally, each bit of syndrome information is *independent* of the others, so it gives us a maximum amount of *new* information.

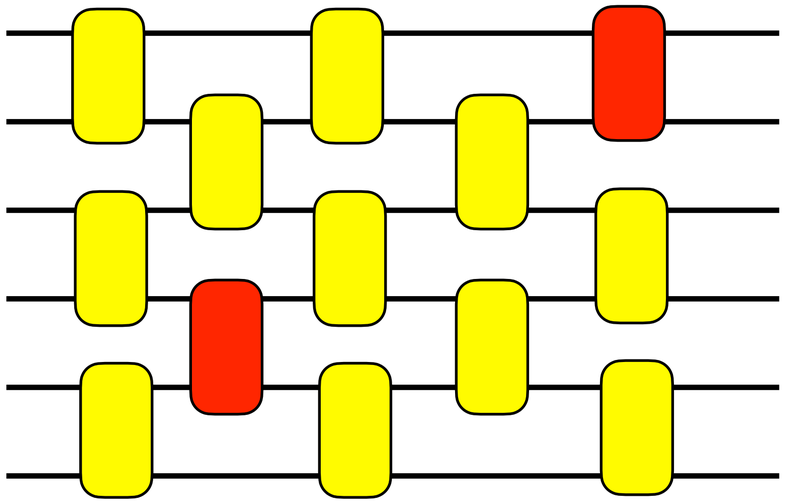


We'd like to achieve the same thing in fault tolerance. Each measurement should give us independent information about the spacetime location of faults.

Independence requires that separate checks collect information in different ways. This is difficult or impossible to achieve using the same code repeatedly or collecting all information at a fixed time. If we can remove these restrictions, possibly we can substantially improve the efficiency of fault tolerance.

Counting the Information Available

Is it possible to collect enough information to identify the spacetime location of errors?



Let's look at one error correction "cycle", during which we introduce m new qubits and measure m qubits.

Suppose we have n+m total qubits. The cycle contains T gates, each of which could have a possible types of errors.

We have up to m bits of information about the errors.

If the error rate is p, we expect pT faulty gates. To find the exact error, we need

$T[h(p) + p \log_2 a]$ bits of information, where $h(p) = -p \log_2 p - (1-p)\log_2(1-p)$

But if we have a circuit of depth d, we have T = d(n+m), so, information theoretically, it is possible if d is constant as a function of n and m is linear in n.

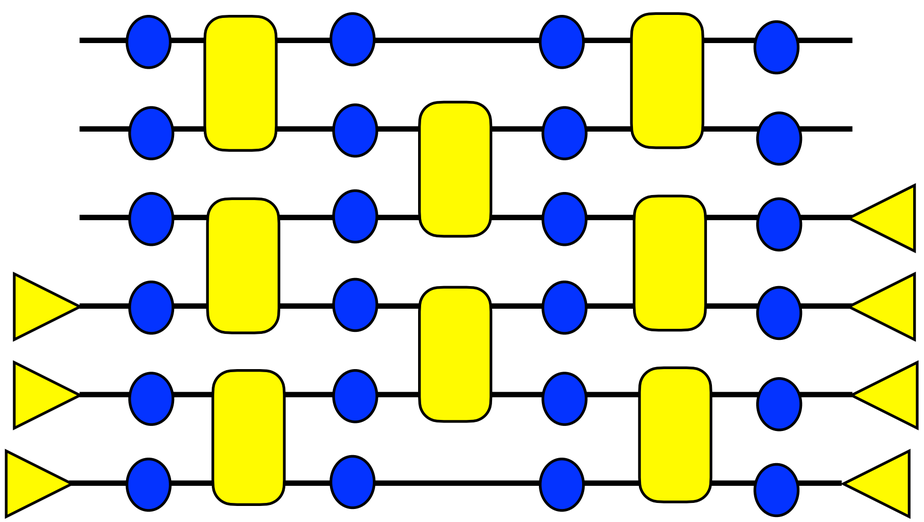
Q&A on Spacetime Codes

We would like to think of a fault-tolerant protocol as a space-time code: Its goal is to find the locations in space and time that have faults.

- But how is fault tolerance possible with non-constant depth circuits? The design of most fault-tolerant circuits ensures that many different faults will have the same effect on the data. They are degenerate space-time codes.
- What is the quantum code? If we stop at any time slice, the data must be encoded in a quantum error-correcting code (or we wouldn't be able to correct faults occurring then). But the code changes with time.
- How do we do logical gates if the code keeps changing? As we continually update the code, we also update what we consider to be the logical basis states. The relative relationship between the current logical state and the logical basis states may change over time, implying logical gates have been performed.
- How do we design fault-tolerant protocols in this picture? Beats me.

A Framework for Spacetime Codes

One option for how to analyze such a protocol is to use spacetime codes in the vein of [BHFS17].



The idea is, given a fault-tolerant protocol, to write down a code with a qubit for each spacetime location.

We define a subsystem stabilizer code. Each gate in the FT circuit gives a few gauge operators representing possible error propagation through the gate. Stabilizer elements represent the propagation of stabilizer constraints from the FT protocol through the circuit.

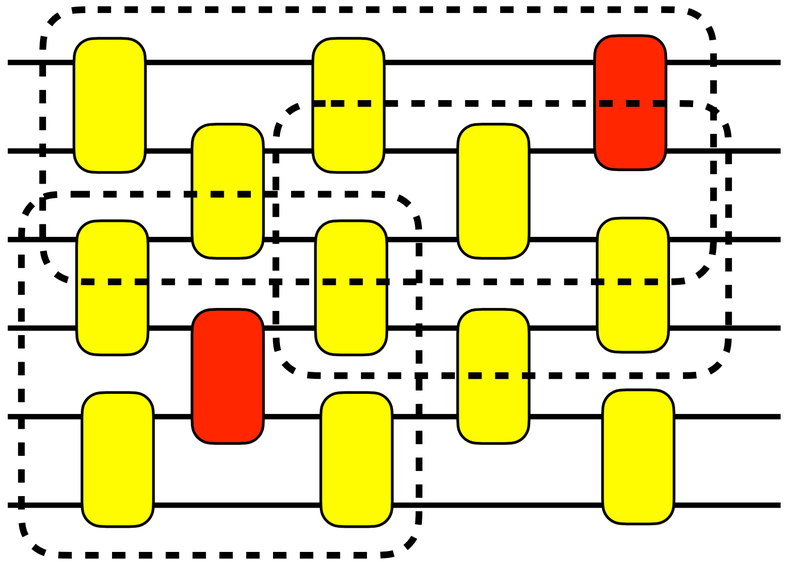
The ability of the FT protocol to distinguish faults in different spacetime locations then corresponds to some error correction properties of the spacetime code. A full analysis of the fault tolerance of the protocol remains complicated, but also see [DP23].

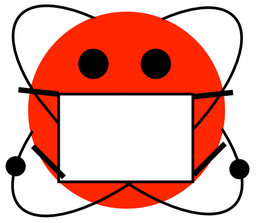
Dynamical Error Syndromes

One feature of dynamical codes is that information about the errors is learned gradually over time.

One way to organize this information is through *detectors*, which are collections of measurements in a circuit with a fixed relationship of their outcomes in the absence of errors. If that relationship is not satisfied, it indicates an error in the spacetime region covered by the detector.

[MBG23]





Error syndrome information is *masked* until we make the right sequence of measurements to unmask it. Information can be *temporarily* or *permanently* masked. We have developed an algorithm (arXiv:2403.04163 [quant-ph]) to determine what is temporarily masked and when it is unmasked vs. what is permanent masked.

Summary

- Dynamical codes are a promising approach to finding new fault-tolerant protocols.
- We are still learning the properties of dynamical codes, but they can potentially behave very differently from static quantum error-correcting codes.
- Using dynamical codes, we should consider fault-tolerant protocols which are very different from existing protocols.
- We should consider a spacetime picture where we view the evolution of the code and errors over time as part of the same structure. We could then try to track the errors back to the time and place of their origin rather than restricting error propagation.
- There are many open questions, including how to design such spacetime codes and to what extent it really is possible to identify the full spacetime origin of an error.