Practical Fault-Tolerance in Near-Term Devices

Reem Mandil, Amita Gnanapandithan, Calvin (Hang Yu) Xu

December 6 2019

Abstract

In this paper, we introduce the theory of fault-tolerant quantum computation and present results from recent literature that demonstrate the feasibility and advantage of performing fault-tolerant quantum error correction on near-term devices.

Contents

1 Introduction to fault-tolerant quantum computation

In recent years, there has been significant progress on the engineering of reliable qubits [1]. However, a qubit is always susceptible to interactions with its environment, which can be described as the application of unintended unitary operations during the preparation, free evolution, quantum gate-processing, or measurement of a qubit. Thus, it is implausible to expect that a long quantum computation can be performed error-free. To build an error-resistant quantum computer, we must be able to perform quantum error correction with noisy gates. Protocols that succeed in achieving this task are said to be fault-tolerant (FT). A more precise definition of fault-tolerance will be presented in Sec. 1.2.

We know that FT quantum computation is indeed possible thanks to the threshold theorem [2], which states that a logical circuit with n qubits and T gates can be replaced by a FT circuit using $O(n \cdot polylog(nT))$ qubits, as long as the physical error rate per gate and per time step is below some constant threshold value. In the asymptotic regime, the ratio of physical qubits to logical qubits (i.e. the 'overhead') scales favourably. For the purposes of near-term devices, however, where the number of qubits that can be operated reliably is 50-100, the constant factors hidden by the big-O notation are significant and the vast majority of available qubits would be needed for correcting errors alone. Thus, FT protocols with low overhead are necessary if we wish to test their performance on noisy intermediate-scale quantum (NISQ) devices and learn better approaches.

This paper is organized as follows: Sec. 1.1 and Sec. 1.2 will present the necessary formalism for analyzing FT protocols. Sec. 2 will present a low-overhead protocol for FT quantum error correction, recently revised and improved by Chamberland and Beverland, called 'flag FT error correction' [3]. Sec. 3 will present low-overhead methods for fault-tolerant computation by Chao and Reichardt [4]. Sec. 4 will report on the practical feasibility of error-detection and fault-tolerance based on an experiment by Vuillot [5]. Finally, we make summarizing remarks on FT quantum computation and its future in Sec. 5.

1.1 Properties of quantum error-correcting codes

An example of a simple quantum error-correcting code (QECC) is Shor's nine-qubit code, which corrects for a bit flip (X), a phase flip (Z), and combinations of the two ($Y = iXZ$). The group generated by the 2^{2n} tensor products of the four operators I, X, Y, Z is called the n-qubit Pauli group, $\mathcal{P}_n$, which spans the space of $2^n \times 2^n$ matrices. That is, each $2^n \times 2^n$ operator can be expressed as a tensor-product "string" of length n, where each operator in the string is either I, X, Y, or Z. Note that the Pauli group is non-Abelian, as any pair of its elements do not necessarily commute – the noncommuting elements of this group anticommute.

Consider a single qubit that interacts with its environment in an arbitrary manner. The evolution of the qubit and its environment can be described by a unitary transformation that is a linear combination of the Pauli operators. The action of an arbitrary unitary operator on n qubits plus their environment can be expanded as [6]

$$|\psi\rangle \, |0\rangle_E \rightarrow \sum_{i=1}^{2^{2n}} E_a \, |\psi\rangle \, |e_a\rangle_E \, , \tag{1}$$

where $|\psi\rangle$ represents our n-qubit state, $|0\rangle_E$ represents (without loss of generality) the initial state of the environment, and the $\{E_a\} \equiv \mathcal{P}_n$ act on $|\psi\rangle$ and create some error. The $\{|e_i\rangle_E\}$ are the corresponding evolved states of the environment, which are not assumed to be normalized or orthogonal here. Eq. 1 lays the foundation for devising a QECC. We define a subset, $\mathcal{E}$, of $\{E_a\}$; this subset consists of all the errors we wish to be able to correct. Now let us define the weight of an operator $Q \in \mathcal{E}$ to be the number of tensor factors which are not the identity, denoted by an integer t where $0 \leq t \leq n$. For example, $XYIZI$ has weight 3. Then, if we devise a QECC that can correct all weight t operators, that QECC can correct t errors [7]. Note that we are implicitly assuming weak or zero correlation between errors on different qubits.

A QECC requires a mapping of k 'logical' qubits into a 'code block' of n qubits, for integers $n > k$ (k qubits are encoded into n qubits). In other words, it is a mapping from a Hilbert space of $dim(2^k)$ to $dim(2^n)$. The resultant quantum state from this mapping is called a 'codeword' and the error correction protocol is applied on this codeword. The complete set of codewords for a quantum code form a $dim(2^k)$ linear subspace of the $dim(2^n)$ Hilbert space, called the 'coding space'. Another important parameter that characterizes a QECC is its distance, d. A code that corrects t errors is said to have distance $2t + 1$, since it takes $2t + 1$ single-qubit changes to get from one codeword to another. A quantum code with block size n, k encoded qubits, and distance d is written as an [[n, k, d]] quantum code.

1.1.1 Stabilizer codes

We can also characterize a QECC in the following way. Let S denote an Abelian subgroup of $\mathcal{P}_n$. Then all elements of S acting on the $dim(2^n)$ Hilbert space ($\mathcal{H}_{2^n}$) can be simultaneously diagonalized, as a simple consequence of the commutative property of Abelian groups. We associate with S a 'stabilizer code' $\mathcal{H}_S \subseteq \mathcal{H}_{2^n}$, which is the simultaneous eigenspace with eigenvalue $+1$ of all elements of S. This means

$$|\psi\rangle \in \mathcal{H}_S \text{ iff } M|\psi\rangle = |\psi\rangle \;\; \forall \; M \in S.$$

The group S preserves all of the codewords and is thus called the 'stabilizer' of the code. We can characterize the group S by its generators, which are the $\{M_i\}$ that are independent and 'complete' (i.e. no element of $\{M_i\}$ can be expressed as a product of the others and each element of S can be expressed as a product of elements in $\{M_i\}$). If S has $n-k$ generators, it can be shown that the coding space $\mathcal{H}_S$ has $dim(2^k)$, or in other words, that there are k encoded qubits.

Most QECCs that have been developed are stabilizer codes. The stabilizer language is useful because it provides a simple way to characterize the errors that the code can detect and correct. A particular element of the Pauli group, E_a, either commutes or anticommutes with a particular stabilizer generator M. If they commute, then for $|\psi\rangle \in \mathcal{H}_S$,

$$ME_a|\psi\rangle = E_aM|\psi\rangle = E_a|\psi\rangle\,,$$

so the error preserves the value $M = 1$. If they anticommute, then

$$ME_a|\psi\rangle = -E_aM|\psi\rangle = -E_a|\psi\rangle\,,$$

so the error flips the value of M. Thus, we can detect the error by measuring M. In general, we can write

$$M_iE_a = (-1)^{s_{ia}}E_aM_i,$$

where $i = 1, ..., n-k$ and s_{ia} represents a 'syndrome' for the error E_a. When we have an [[n, k, d]] stabilizer code, to do error correction, we wish to measure the eigenvalue of each of the $n-k$ generators. Each eigenvalue tells us one bit of the error syndrome, representing whether the error E commutes or anticommutes with each of the generators. For the example of the nine-qubit code, which is a [[9, 1, 3]] quantum code, the stabilizer consists of the group of the 8 measured operators and the syndromes were the parity checks in the bit flip correction. Every non-trivial Pauli operator in $\mathcal{E}$ has some error syndrome due to the non-Abelian nature of $\mathcal{P}_n$.

The necessary and sufficient condition for error recovery to be possible is summarized by the following theorem [8], which is given here without proof:

Theorem 1.1. *Suppose $\mathcal{E}$ is a linear space of errors acting on $\mathcal{H}_{2^n}$ and $E_a, E_b \in \mathcal{E}$. Then a subspace C of $\mathcal{H}_{2^n}$ forms a QECC correcting the errors $\mathcal{E}$ iff*

$$\langle\psi| E_a^\dagger E_b |\psi\rangle = C_{ab} \tag{2}$$

for all $E \in \mathcal{E}$ and $|\psi\rangle$ in the code space, where C_{ab} is independent of $|\psi\rangle$.

We can extend this theorem to find a condition to be satisfied by the stabilizer of a code [6].

Theorem 1.2. *The condition in Eq. 2 is satisfied if, for each $E_a, E_b \in \mathcal{E}$, one of the following holds:*

1. *$E_a^\dagger E_b \in S$,*
2. *There is an $M \in S$ that anticommutes with $E_a^\dagger E_b$.*

Proof. In case (1), $\langle\psi| E_a^\dagger E_b |\psi\rangle = \langle\psi|\psi\rangle = 1$ for $|\psi\rangle \in \mathcal{H}_S$. In case (2), suppose $M \in S$ and $ME_a^\dagger E_b = -E_a^\dagger E_b M$. Then, $\langle\psi| E_a^\dagger E_b |\psi\rangle = \langle\psi| E_a^\dagger E_b M |\psi\rangle = -\langle\psi| M E_a^\dagger E_b |\psi\rangle = -\langle\psi| E_a^\dagger E_b |\psi\rangle$, and therefore $\langle\psi| E_a^\dagger E_b |\psi\rangle = 0$. □

1.2 Defining fault-tolerance

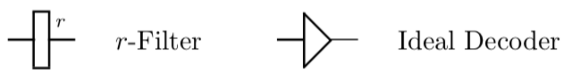
Performing a QECC on its own is insufficient for making a quantum computer that is resistant to errors. FT quantum protocols must be able to perform operations on encoded qubits and produce the 'correct answer' even when any individual component of the quantum circuit may fail. The four basic components of a universal quantum computer are:

1. Preparation: operations which prepare, for example, a $|0\rangle$ state,
2. Quantum Gates: a universal set of quantum gates such as $\{H, R_{\frac{\pi}{8}}, CNOT\}$,
3. Measurement: measuring individual qubits in, for example, the standard basis, and
4. Wait: no action performed on a qubit.

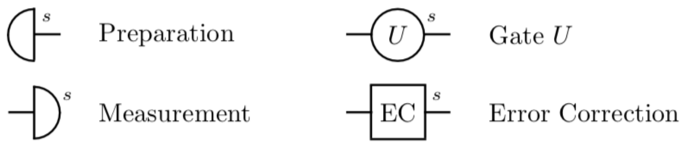
Each instantiation of one of these components is called a location; that is, the maximum number of locations in a circuit is the total number of qubits times the total number of time steps. The construction of a FT version of any location is called a 'gadget' for that particular operation. Clearly, we need gadgets representing components (1)-(3) above as well as an error correction gadget so that we prevent accumulation of errors throughout the computation as well as avoid creating errors with our QECC. Now we use the treatment by Gottesman [7] to present several definitions that precisely describe what is meant by fault-tolerance. For simplicity, these definitions focus on fault-tolerance in an [[n, 1, 2t + 1]] code. That is, there is one encoded qubit per block and the code can correct t errors. For the treatment of multiple qubits encoded per block, see Ref. [9].

Definition 1.1. (r-filter and ideal decoder). An r-filter is a projector onto the subspace spanned by all states $Q|\psi\rangle$, where $Q \in \mathcal{E}$ is a Pauli error of weight $\leq r$ and $|\psi\rangle$ is an arbitrary codeword. An ideal decoder is a map that takes the input (encoded) state, corrects any errors, performs a decoding operation, and returns an unencoded state with no faulty locations.

The ideal decoder allows us to talk about the *logical* state of the quantum computer at any point during the computation, and the r-filter makes precise the notion of a state having "at most r errors" since only the states which could possibly be created from a valid codeword with at most r single-qubit errors can pass through the r-filter. The graphical representation of these objects is

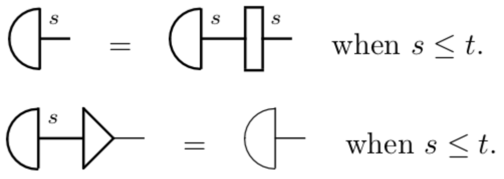


where the thick horizontal lines represent a single block of a QECC (i.e. n qubits), except for the output of the ideal decoder, which is a single unencoded qubit. The graphical representation for our gadgets is



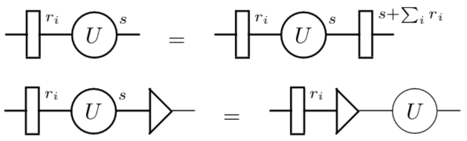
where s is the maximum number of faulty locations that may be present in the gadget. A similar diagram but with thin lines and no indication of the number of errors represents an idealized, unencoded version of the same operation.

Definition 1.2. (FT preparation). A preparation gadget is fault-tolerant if it satisfies the following two properties:



That is, a FT preparation step should output a state that is within s errors of a properly encoded state. Also, the prepared state should decode to the correct state under an ideal decoder.

Definition 1.3. (FT quantum gate). A gate gadget is fault-tolerant if it satisfies the following two properties:

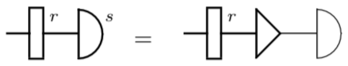


when $s + \sum_i r_i \leq t$.

when $s + \sum_i r_i \leq t$.

That is, the final number of errors in each output block should not exceed the total number of errors on the incoming states, plus the number of errors that occurred during the gate gadget. Also, a FT gate gadget should perform the correct encoded gate if there are not too many errors in the incoming blocks and gadget combined.

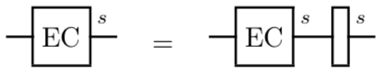
Definition 1.4. (FT measurement). A measurement gadget is fault-tolerant if it satisfies the following property:



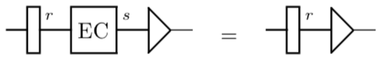
when $r + s \leq t$.

That is, if the sum total of errors in the incoming state and measurement gadget is at most t, then we should get the same result out of the real gadget as if we had performed ideal decoding on the incoming state and measured the decoded qubit.

Definition 1.5. (FT error correction). An error correction (EC) gadget is fault-tolerant if it satisfies the following two properties:



when $s \leq t$.



when $r + s \leq t$.

That is, a FT EC step should output a state that is at most s errors away from some encoded state. Also, if the sum total of errors in the incoming state and EC gadget is at most t, the state has been corrected, in the sense that the logical state after the EC step is the same as the logical state before it.

Definition 1.6. (t-FT error correction). For $t = [(d-1)/2]$, an error correction protocol using a distance-d stabilizer code, C, is t-fault-tolerant if the following two conditions are satisfied:

1. For an input codeword with error of weight s_1, if s_2 faults occur during the protocol with $s_1 + s_2 \leq t$, ideally decoding the output state gives the same codeword as ideally decoding the input state,
2. For s faults during the protocol with $s \leq t$, the output state differs from a codeword by an error of at most weight s no matter how many errors are present in the input state.

1.2.1 Threshold theorem

Suppose we wish to simulate a quantum circuit, C, and achieve a final accuracy of ϵ. There is a family of codes called 'concatenated codes' that give rise to the threshold theorem for quantum computation. Briefly, the idea is to create a FT circuit, C', simulating C, then create a FT circuit C'' simulating C', and so on *ad infinitum* [10]. Analysis of these codes tells us how large a FT simulating circuit must be to achieve this level of accuracy.

Theorem 1.3. *For any ideal circuit C of n qubits and $T = poly(n)$ gates, there exists a fault-tolerant circuit C' that produces an output with statistical distance at most ϵ from the output of C, provided that each location of C' fails independently with probability $p < p_{th}$, where p_{th} is some constant threshold. C' is at most of size $O(n \cdot polylog(nT/\epsilon))$.*

A proof of the threshold theorem may be found in p. 42 of Ref. [7]. This result states that noise is not a fundamental barrier to scalable quantum computing and thus gives grounds for the pursuit of engineering quantum computers.

2 Flag fault-tolerant error correction

2.1 Motivation

As previously mentioned, schemes that tolerate errors incur a large space overhead. This presents a big obstacle for scalable quantum computing and, perhaps more importantly for the near term, makes it difficult to run fault-tolerance experiments. By definition, flag FT error-correcting circuits "use extra ancilla qubits to signal when errors resulting from v faults in the circuit have weight $> v$" [3]. This type of error correction requires the fewest ancilla qubits known to date, making it a contender for low-overhead FT error correction. Another low-overhead competitor worth mentioning is a "surface code with a minimum weight matching error correction scheme." This scheme has one of the highest p_{th} values, meaning it can be implemented using relatively imperfect resources, but it demands an impractical overhead.

In addition to requiring the least overhead, flag FT error correction is much more general in that it can be used with a large class of stabilizer codes, as long as they satisfy some conditions. In the paper, *Flag fault-tolerant error correction with arbitrary distance codes* by Chamberland and Beverland [3], the scheme is discussed in detail and is shown to be usable with stabilizer codes of arbitrary distance. Previous works were unable to take flag FT error correction beyond stabilizer codes of distance three. Here, the details of this protocol will not be explained; rather, the conditions required by the protocol and a comparison with other schemes will be discussed.

2.2 Eligible stabilizer codes

Stabilizer codes to be used with flag FT error correction must meet certain criteria, as described by the theorem [3]:

Theorem 2.1. *For a stabilizer code (group) $S = \langle g_1, g_2, ..., g_\tau \rangle$ to be compatible with flag fault-tolerant error correction with t-flag circuits $C(g_1), C(g_2), ..., C(g_\tau)$, we require the following. For any set of m stabilizer generators $\{g_{i_1}, ..., g_{i_m}\}$ such that $1 \leq m \leq t$, every pair of elements $E, E' \in \bigcup_{j=0}^{t-m} \mathcal{E}_{t-j}(g_{i_1}, ..., g_{i_m}) \times \mathcal{E}_j$ either satisfy $s(E) \neq s(E')$ or $E \sim E'$, where s is the syndrome of the error.*

However, it can be seen that these conditions are also dependent on $C(g_1), C(g_2), ..., C(g_\tau)$ (t-flag circuit choices). The authors show that stabilizer codes which also satisfy the conditions below can be used for flag FT error correction using an arbitrary selection of t-flag circuits. A proof can be found in Ref. [3].

Theorem 2.2. *For a stabilizer code (group) $S = \langle g_1, g_2, ..., g_\tau \rangle$ and distance $d > 1$ to be compatible with flag fault-tolerant error correction with arbitrary t-flag circuits $C(g_1), C(g_2), ..., C(g_\tau)$, we require the following. For all $v = 0, 1, ..., t$, all choices $\mathcal{Q}_{t-v}$ of 2$(t-v)$ qubits, and all subsets of v stabilizer generators $\{g_{i_1}, ..., g_{i_m}\} \subset \{g_1, ..., g_\tau\}$, there is no logical operator $l \in \mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$ such that $supp(l) \subset supp(g_{i_1}) \cup ... \cup supp(g_{i_v}) \cup \mathcal{Q}_{t-v}$, where $\mathcal{N}(\mathcal{S})$ is the normalizer of the stabilizer group.*

Examples of stabilizer codes which satisfy this condition include the "rotated surface code" family of stabilizer codes, represented as $[[d^2, 1, d]]$ where $d = 2t + 1$.

2.3 Comparison with other schemes

Chamberland and Beverland compared their flag EC with other FT error-correcting schemes by assessing their performances under various circuit noise conditions. Their simulation results can be seen in Figure 1.

The parameter $\tilde{p}$ represents the failure probability of idle qubits in the circuit, while p represents the failure probability of all other qubits. The ratio of these parameters is the noise level of the circuit. The expression $p_L = \lim_{N\to\infty} \frac{N_{fail}}{N}$ is the logical failure rate for the circuit, where N_{fail} is the number of logical X or Z errors that occur in N rounds of running the circuit. p_L is a function of p. Figures 1a, 1c, and 1d compare the failure probability with respect to p for a certain set of circuits ([[5, 1, 3]] flag EC, [[7, 1, 3]] flag EC, [[7, 1, 3]] Steanne EC, and surface code with $d = 3$) for $\tilde{p} = p$, $\tilde{p} = p/10$, and $\tilde{p} = p/100$ respectively. Figures 1b, 1d, and 1f do the same for another set of circuits ([[19, 1, 5]] flag EC, [[19, 1, 5]] Steanne EC, and surface code with $d = 5$).

There are some notable observations that can be made from these simulation results. Firstly, for lower n ($7 \geq n \geq 17$), flag EC has logical failure rates that are several orders of magnitude higher than the other schemes, especially when $\tilde{p}/p$ is small (see Figures 1a, 1c, and 1d). For larger n, however, (see Figure 1b, 1d, and 1f), flag EC demonstrates a lower logical failure rate than other schemes for low enough p. Conclusively, flag FT error correction shows an improvement over other schemes under specific conditions.

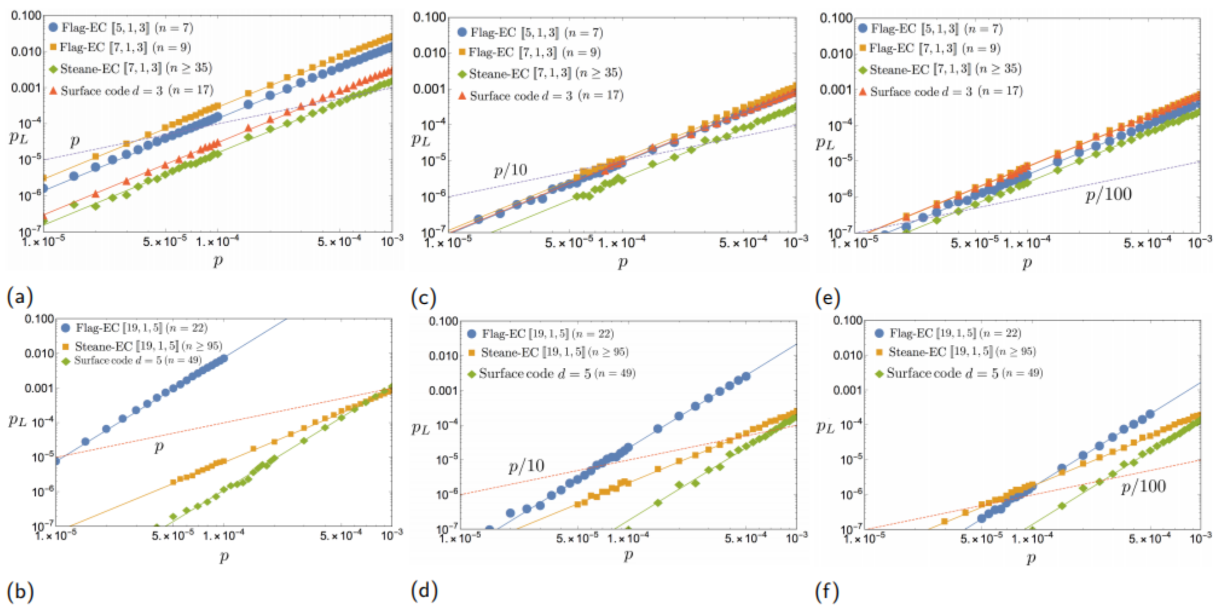


Figure 1: Simulated results of logical failure rates with respect to number of non-idle qubits for various error-correcting circuits and noise levels.

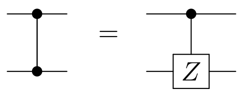
3 Low-overhead methods for fault-tolerant quantum computation

In addition to low-overhead error correction, a scalable universal quantum computer will also need low-overhead quantum gates. Chao and Reichardt give FT procedures for applying a universal gate set in a [[n, n-2, 2]] quantum code (for even n), using only two extra qubits [4]. Note that this code has multiple encoded qubits per block, making it more efficient but less tolerant to noise as single-gate failures can cause correlated errors.

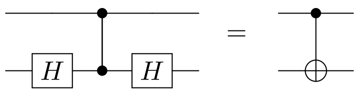
We restrict our attention to how they fault-tolerantly apply encoded CNOT and Hadamard gates on arbitrary logical qubits in a code block. For the purposes of a distance-two code, if any single fault within an operation leads to a detectable error or no error, then the operation can be deemed FT. Two-ancilla-qubit FT procedures for state preparation, EC, and measurement have also been introduced by the same authors in Ref. [11].

3.1 Fault-tolerant CZ gate

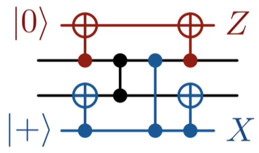
The controlled-Z (CZ) gate is denoted by



and, along with two Hadamard gates, can be used to implement the CNOT gate:

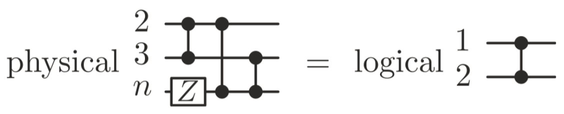


So we focus on implementing a FT CZ gate. When a noisy CZ gate fails, it applies the ideal CZ gate followed by one of the 15 nontrivial two-qubit Pauli operators. A CZ gate gadget that detects all $s = 1$ two-qubit correlated faults looks like



where the red and blue qubits are the ancilla. If there are no faults, the gadget implements a CZ gate and the ancilla qubits measure 0 and + in the standard and X basis, respectively. If there is one fault and the measurements still return 0 and +, then it is not hard to work out that neither XX, XY, YX, YY, nor ZZ errors could have occurred on the output of the CZ gate. The remaining two-qubit errors ($IX, IY, IZ, XI, YI, ZI, XZ, YZ, ZX, ZY$) go undetected.

Now consider the following implementation of a logical $CZ_{1,2}$ gate as a $Z_n CZ_{2,3} CZ_{2,n} CZ_{3,n}$ circuit:



The CZ gate faults that are detectable in this circuit are precisely the ones that are undetectable in the CZ gadget above, and vice versa. For example, if the $CZ_{2,3}$ gate fails as XI, it will result in the error $X_2 Z_n$, which is detectable. Thus, if we replace each physical CZ gate in this circuit with the CZ gadget, we achieve a FT implementation of a logical $CZ_{1,2}$ gate. Note that the circuit uses at most two ancilla qubits at a time.

3.2 Fault-tolerant Hadamard gate

By similar means, the authors also construct a FT circuit for the logical Hadamard gate. They use the CZ gadget in various locations of a larger, physical circuit for a FT implementation of H_1 using at most two ancilla at a time.

One might observe that these FT circuits send us in a loop: a CNOT gate is simulated with Hadamard and CZ gates, which are in turn simulated with CNOT, CZ, and Hadamard gates. The idea however, is that after a sufficient number of recursive steps, the operation provides the 'correct answer' with high probability. The practical feasibility of these procedures has not been tested, but these results show that substantial FT quantum calculations can, in theory, be implemented with fewer than 20 qubits for the majority of codes.

4 Experimental demonstration of fault-tolerance on IBM 5Q chip

In 2017, the first experimental demonstration of error detection and fault-tolerance improving a quantum computation was shown by Vuillot [5]. Inspired by a recent paper by Gottesman [12], Vuillot provides validation for FT schemes using the IBM 5Q chip, a 5 qubit quantum chip released by IBM in 2016 and accessible on the cloud. Using this chip, Vuillot demonstrates that the performance of simple sampling tasks can be improved by implementing error detection combined with post-selection. The performance of the circuit is evaluated by comparing the probability distribution obtained by sampling from the final state with the expected distribution. The circuits under consideration are small enough to be classically simulated, and so the expected distribution can be computed on a classical computer.

4.1 Overview of the fault-tolerant scheme

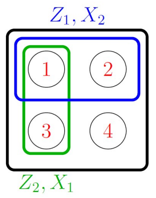
Vuillot attempts to prepare a set of FT states, S_{FT}, and gates, G_{FT}, given by

$$S_{FT} = \left\{ |00\rangle_L , |0+\rangle_L , \frac{|00\rangle_L + |11\rangle_L}{\sqrt{2}} \right\}$$
$$G_{FT} = \{X_1, X_2, Z_1, Z_2, H \otimes H \cdot \text{SWAP}, CZ\} .$$

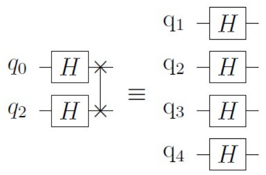
where $|0+\rangle_L = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$. The FT scheme encodes the logical states in S_{FT} in the following way:

$$|00\rangle_L \to \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle) \qquad |01\rangle_L \to \frac{1}{\sqrt{2}}(|1100\rangle + |0011\rangle)$$
$$|10\rangle_L \to \frac{1}{\sqrt{2}}(|1010\rangle + |0101\rangle) \qquad |11\rangle_L \to \frac{1}{\sqrt{2}}(|1001\rangle + |0110\rangle),$$

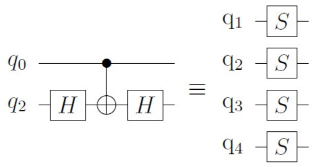
so each 2-qubit bare implementation is mapped to a 4-qubit encoded implementation, with the 5th qubit being used as an ancilla. The logical Pauli gates $X_i, Z_j \in G_{FT}$ operate on the qubits as shown in Figure 2a, and the $H \otimes H \cdot \text{SWAP}$ and CZ operations are implemented as shown in Figure 2b and 2c respectively.



(a) Operation of the Pauli gates on the encoded qubits.



(b) Bare and encoded implementation of the $H \otimes H \cdot \text{SWAP}$ operation.



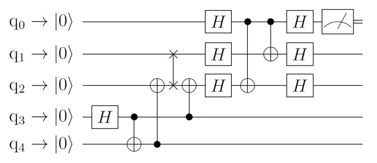
(c) Bare and encoded implementation of the CZ gate, where S is the phase gate $\text{diag}(1, i)$.

Figure 2: Overview of the gates, G_{FT}, used by Vuillot.

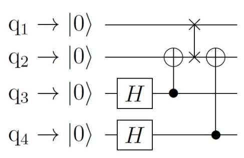
Due to limited connectivity between qubits on the IBM 5Q chip, however, it is difficult to prepare the $|00\rangle_L$ and $|0+\rangle_L$ encoded states in the fully FT way proposed by Gottesman [12]. To work around this, Vuillot uses a SWAP gate to commute the qubits such that all the required operations can be performed on the IBM 5Q chip. The preparation of each state is shown in Figure 3. This trick, however, introduces undetectable errors. For the circuits in Figures 3a and 3b, the failure of the SWAP gate leads to an undetected $X_1 \otimes X_2$ error. Any other single fault will lead to the error being detected by the ancilla; therefore, the ancilla acts as a flag for post-selection: if the ancilla $q_0 = |1\rangle$ then an error has been detected and the measurement is discarded.

4.2 Experimental results

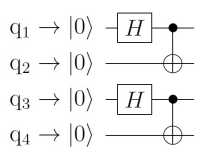
Using the IBM 5Q chip, Vuillot compares the performance of the bare and encoded implementations of the 20 different circuits described in Table 1, in Appendix A. A jupyter notebook to reproduce the experiment is accessible at Ref. [13]. The experiment was performed 36 times. During each experiment, the circuit was quickly run 8192 times, so that the chip is considered to be in the same condition within each experiment, and each experiment is considered independent of the others.



(a) Preparation of the state $|00\rangle_L$. We can verify that if the SWAP gate fails it will not be detected by the ancilla q_0.



(b) Preparation of the state $|0+\rangle_L$. Similar to 3a, if the SWAP gate fails it will not be detected.



(c) Preparation of the state $\frac{1}{\sqrt{2}}(|00\rangle_L + |11\rangle_L)$.

Figure 3: Overview of the preparation of the initial states S_{FT}.

The observed outcome distribution is compared with the expected distribution for each circuit. The statistical distance

$$D = \frac{1}{2}\sum_{i,j} |p_{ij} - \tilde{p}_{ij}|$$

is used as a metric to determine the performance of each circuit, where p_{ij} is the expected theoretical probability of outcome ij. In this case there are 4 possible measurement results so $p_{ij} \in \{p_{00}, p_{01}, p_{10}, p_{11}\}$. The observed probabilities $\tilde{p}_{ij}$ were calculated by computing n_{ij}/n_{valid}, where n_{ij} was the number of observations with outcome ij and n_{valid} was the total number of observations kept after post-selection. The post-selection ratio varied from 32% to 85%, with a mean of 56% over all circuits and all runs. Note that the choice of statistical distance as the metric was not unique and any other reasonable metric could have been used.

The experimental results are shown in Figure 4. We can see that there is a crossover point at approximately 3 gates in the bare implementation. Below this point the bare implementation performs better, while above this point the encoded implementation performs better. This is evidence that fault-tolerance can improve the performance for quantum circuits. Given this trend, we would expect that as the number of gates increases the encoded implementation would outperform the bare implementation even more.

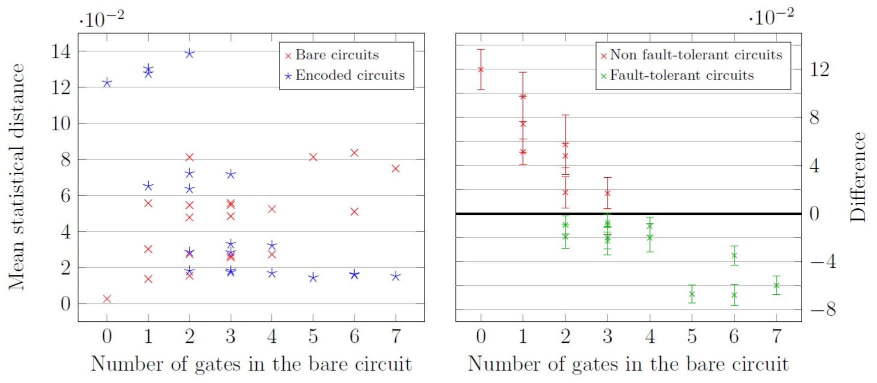


Figure 4: Performance of bare and encoded circuits on the IBM 5Q chip. (Left) There are two data points per circuit: red crosses are the bare implementations and blue stars are the encoded implementations. Both data points are placed on the same vertical axis, corresponding to the number of gates in the bare circuit, and show the mean statistical distance to the ideal distribution. (Right) The difference between the encoded and bare implementations' performances is shown for each of the 20 circuits. The circuits with bare implementation advantage are in red, while the circuits with encoded implementation advantage are in green. The error bars show the 99% confidence interval.

Now we take a closer look at the four simple circuits used for state preparation. These are the circuits with ID #1-4 in Table 1, and they have 0, 1, 1, and 2 gates in their bare implementations, respectively. The observed probability distribution is shown in Figure 5. The expected probability distribution in this case is

1 for the specific state being prepared and 0 for the other states. As mentioned in the previous section, the most prominent error in the encoded implementation is the $X_1 \otimes X_2$ bit flip.

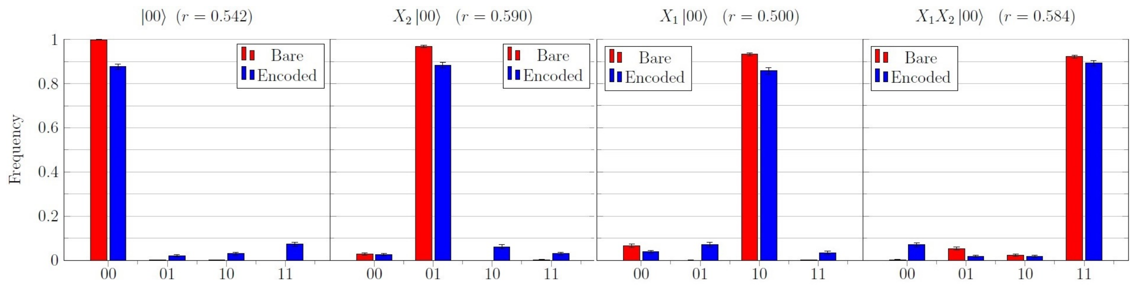


Figure 5: Histogram of outcomes for the state preparation circuits. The given r is the rate of post-selection, and the error bars show 99% confidence intervals.

We also take a closer look at two complex circuits, #13 and #20. The observed probability distributions are shown in Figure 6. In this case, the expected probability distribution is uniform for all states, and we can observe graphically that the encoded implementation has lower statistical distance compared to the bare implementation.

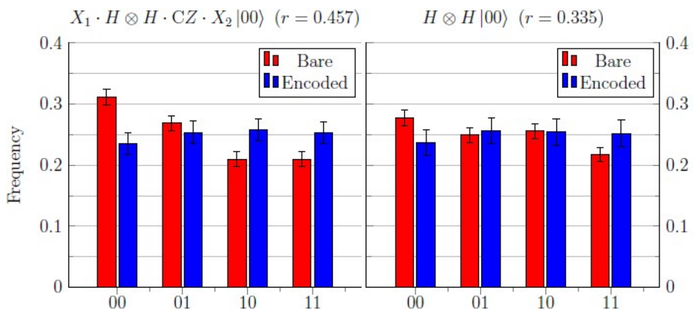


Figure 6: Histogram of outcomes for circuit #13 (left) and #20 (right). The ideal probability distribution is uniform with frequency 25%. As we can see, the encoded implementation is closer to the ideal distribution than the bare implementation. The given r is the rate of post-selection, and the error bars show 99% confidence intervals.

5 Conclusions and outlook

In this paper, we've explored flag fault-tolerant error correction as a candidate for low-overhead FT error correction. Unfortunately this error correction scheme, despite its low overhead, is still unable to compete with higher overhead alternatives in certain regimes, such as when the number of qubits is $7 \geq n \geq 17$. Regardless, this protocol is in its early stages and has potential for growth. We've also seen low-overhead procedures for implementing FT quantum gates. This work shows potential for FT universal quantum computation using fewer than 20 qubits. Lastly, we saw an experimental achievement of FT quantum computation on the IBM 5Q chip, which demonstrated an advantage over non-FT circuits.

Only a few months ago, claims of quantum supremacy were made by Google [14]. Despite the contentious nature of their results, one thing is clear: researchers are pushing the boundaries of what we can accomplish with NISQ devices. Alongside this advancement must come measures for fault-tolerance. Fortunately, significant process is being made in this area, with error rates in qubits starting to approach the threshold needed for fault-tolerance. One development that is worth mentioning is that of topological qubits for computing. Despite being very much in the theoretical stages, with researchers questioning whether their creation is even possible, these qubits would be inherently FT. Given the inherently delicate nature of quantum systems, fault-tolerance is necessary for taking us from the NISQ era to scalable quantum computation.

References

[1] W. Huang, C. H. Yang, K. W. Chan, T. Tanttu, B. Hensen, R. C. C. Leon, M. A. Fogarty, J. C. C. Hwang, F. E. Hudson, K. M. Itoh, A. Morello, A. Laucht, and A. S. Dzurak, "Fidelity benchmarks for two-qubit gates in silicon," *Nature*, vol. 569, no. 7757, pp. 532–536, may 2019.

[2] D. Aharonov and M. Ben-Or, "Fault-tolerant Quantum Computation with Constant Error," in *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, ser. STOC '97. New York, NY, USA: ACM, 1997, pp. 176–188.

[3] C. Chamberland and M. E. Beverland, "Flag fault-tolerant error correction with arbitrary distance codes," *Quantum*, vol. 2, p. 53, Feb. 2018.

[4] R. Chao and B. W. Reichardt, "Fault-tolerant quantum computation with few qubits," *npj Quantum Information*, vol. 4, no. 1, p. 42, 2018.

[5] C. Vuillot, "Error detection is already helpful on the IBM 5Q chip," may 2017. [Online]. Available: http://arxiv.org/abs/1705.08957v1.pdf

[6] J. Preskill, "Quantum error correction," 2016. [Online]. Available: http://www.theory.caltech.edu/people/preskill/ph229/notes/chap7.pdf

[7] D. Gottesman, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," 05 2009.

[8] E. Knill and R. Laflamme, "Theory of quantum error-correcting codes," *Phys. Rev. A*, vol. 55, pp. 900–911, Feb 1997.

[9] D. Gottesman, "A theory of fault-tolerant quantum computation," *Physical Review A*, vol. 57, 02 1997.

[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. New York, NY, USA: Cambridge University Press, 2011.

[11] R. Chao and B. W. Reichardt, "Quantum error correction with only two extra qubits," *Phys. Rev. Lett.*, vol. 121, p. 050502, Aug 2018.

[12] D. Gottesman, "Quantum fault tolerance in small experiments," Oct 2016. [Online]. Available: http://arxiv.org/abs/1610.03507

[13] C. Vuillot, "Code for the fault tolerance experiment," https://github.com/ChristopheVuillot/Fault-tolerant-demo-IBM5Q.

[14] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

Appendix A Circuits tested on the IBM 5Q chip

A list of the circuits tested by Vuillot on the IBM 5Q chip is given in Table 1.

Characteristics	Initial state	Unitary	Final state	ID#
• Non fault-tolerant • worse performance when encoded	$\|00\rangle$	$\mathbb{1}\otimes\mathbb{1}$	$\|00\rangle$	1
	$\|00\rangle$	$\mathbb{1}\otimes X$	$\|01\rangle$	2
	$\|00\rangle$	$X\otimes\mathbb{1}$	$\|10\rangle$	3
	$\|00\rangle$	$X\otimes X$	$\|11\rangle$	4
	$\|0+\rangle$	$\mathbb{1}\otimes\mathbb{1}$	$\frac{\|00\rangle+\|01\rangle}{\sqrt{2}}$	5
	$\|0+\rangle$	$\mathbb{1}\otimes Z$	$\frac{\|00\rangle-\|01\rangle}{\sqrt{2}}$	6
	$\|0+\rangle$	$X\otimes\mathbb{1}$	$\frac{\|10\rangle+\|11\rangle}{\sqrt{2}}$	7
	$\|0+\rangle$	$X\otimes Z$	$\frac{\|10\rangle-\|11\rangle}{\sqrt{2}}$	8
• Fault-tolerant • better performance when encoded	$\frac{\|00\rangle+\|11\rangle}{\sqrt{2}}$	$\mathbb{1}\otimes\mathbb{1}$	$\frac{\|00\rangle+\|11\rangle}{\sqrt{2}}$	9
	$\frac{\|00\rangle+\|11\rangle}{\sqrt{2}}$	$\mathbb{1}\otimes Z$	$\frac{\|00\rangle-\|11\rangle}{\sqrt{2}}$	10
	$\frac{\|00\rangle+\|11\rangle}{\sqrt{2}}$	$X\otimes\mathbb{1}$	$\frac{\|10\rangle+\|01\rangle}{\sqrt{2}}$	11
	$\frac{\|00\rangle+\|11\rangle}{\sqrt{2}}$	$\mathbb{1}\otimes ZX$	$\frac{\|10\rangle-\|01\rangle}{\sqrt{2}}$	12
	$\|00\rangle$	$H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle+\|01\rangle+\|10\rangle+\|11\rangle}{2}$	13
	$\|00\rangle$	$\mathbb{1}\otimes Z\cdot H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle-\|01\rangle+\|10\rangle-\|11\rangle}{2}$	14
	$\|00\rangle$	$Z\otimes\mathbb{1}\cdot H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle+\|01\rangle-\|10\rangle-\|11\rangle}{2}$	15
	$\|00\rangle$	$Z\otimes Z\cdot H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle-\|01\rangle-\|10\rangle+\|11\rangle}{2}$	16
	$\|00\rangle$	$\mathrm{CZ}\cdot H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle+\|01\rangle+\|10\rangle-\|11\rangle}{2}$	17
	$\|00\rangle$	$\mathrm{CZ}\cdot\mathbb{1}\otimes Z\cdot H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle-\|01\rangle+\|10\rangle+\|11\rangle}{2}$	18
	$\|00\rangle$	$\mathrm{CZ}\cdot Z\otimes\mathbb{1}\cdot H\otimes H\cdot\mathrm{SWAP}$	$\frac{\|00\rangle+\|01\rangle-\|10\rangle+\|11\rangle}{2}$	19
	$\|00\rangle$	$\mathbb{1}\otimes X\cdot\mathrm{CZ}\cdot H\otimes H\cdot\mathrm{SWAP}\cdot X\otimes\mathbb{1}$	$\frac{\|00\rangle-\|01\rangle-\|10\rangle-\|11\rangle}{2}$	20

Table 1: List of circuits tested on the IBM 5Q chip by Vuillot [5]. The red (green) circuits are the non-FT (FT) circuits shown also in red (green) on the right side of Figure 4. The ‘Unitary’ column lists the gates used in the bare implementation of the circuit.