Flag fault-tolerant error correction with arbitrary distance codes

Christopher Chamberland[1] and Michael E. Beverland[2]

[1] Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

[2] Station Q Quantum Architectures and Computation Group, Microsoft Research Redmond, WA 98052, USA

February 2, 2018

arXiv:1708.02246v3 [quant-ph] 31 Jan 2018

In this paper we introduce a general fault-tolerant quantum error correction protocol using flag circuits for measuring stabilizers of arbitrary distance codes. In addition to extending flag error correction beyond distance-three codes for the first time, our protocol also applies to a broader class of distance-three codes than was previously known. Flag circuits use extra ancilla qubits to signal when errors resulting from v faults in the circuit have weight greater than v. The flag error correction protocol is applicable to stabilizer codes of arbitrary distance which satisfy a set of conditions and uses fewer qubits than other schemes such as Shor, Steane and Knill error correction. We give examples of infinite code families which satisfy these conditions and analyze the behaviour of distance-three and -five examples numerically. Requiring fewer resources than Shor error correction, flag error correction could potentially be used in low-overhead fault-tolerant error correction protocols using low density parity check quantum codes of large code length.

1 Introduction and formalism

Scalable quantum computers are expected to require some form of error correction (EC) to function reliably. Unfortunately, no practical model for a self-correcting quantum memory has been proposed to date, despite considerable effort [1]. The models that come closest to this goal involve topological protection in the presence of physically imposed symmetries [2, 3], but even these are not expected to reduce error rates sufficiently for large computations. Therefore active protocols that require measuring the check operators of an error correcting code are probably necessary to realize scalable quantum computing.

Christopher Chamberland: c6chambe@uwaterloo.ca
Michael E. Beverland: mibeverl@microsoft.com

There are three general approaches of fault-tolerant error correction (FTEC) applicable to a wide range of stabilizer codes due to Shor [4], Steane [5], and Knill [6]. There are also a number of promising code-specific FTEC schemes, most notably the surface code with a minimum weight matching error correction scheme [7–9]. This approach gives the best fault-tolerant thresholds to date and only requires geometrically local measurements. A high threshold [4, 10–12] implies that relatively imperfect hardware could be used to reliably implement long quantum computations. Despite this, the hardware and overhead requirements for the surface code are sufficiently demanding that it remains extremely challenging to implement in the lab.

Fortunately, there are reasons to believe that there could be better alternatives to the surface code. For example, dramatically improved thresholds could be possible using concatenated codes if they enjoyed the same level of optimization as the surface code has in recent years [13, 14]. Another enticing alternative is to find and use efficiently-decodable low density parity check (LDPC) codes with high rate [15–17] in a low-overhead FTEC protocol [18]. For these and other reasons, it is important to have general FTEC schemes applicable to a wide range of codes and to develop new schemes.

Shor EC can be applied to any stabilizer code, but typically requires more syndrome measurement repetitions than Steane and Knill EC. Furthermore, all weight-w stabilizer generators are measured sequentially using w-qubit verified cat states. On the other hand, Steane EC has higher thresholds than Shor EC and has the advantage that all Clifford gates are applied transversally during the protocol. However, Steane EC is only applicable to CSS [5, 19] codes and uses a verified logical $|+\rangle$ state encoded in the same code to simultaneously obtain all X-type syndromes, using transversal measurement (similarly for Z). Knill EC can also be applied to any stabilizer code but requires two additional ancilla code blocks (encoded in the same code that protects the data) prepared in a logical Bell state. The Bell state teleports the encoded information to one of

the ancilla code blocks and the extra information from the transversal Bell measurement gives the error syndrome. Knill EC typically achieves higher thresholds than Shor and Steane EC but often uses more qubits [20, 21]. It is noteworthy that for large LDPC codes, in which low weight generators are required be fault-tolerantly measured, Shor EC is much more favourable than Steane or Knill EC. Many improvements in these schemes have been made. For examples, in [22], ancilla decoding was introduced to correct errors arising during state preparation in Shor and Steane EC rather than simply rejecting all states which fail the verification procedure.

In this work, we build on a number of recent papers [23–25] that demonstrate flag error correction for particular distance-three and error detecting codes and present a general protocol for arbitrary distance codes. Flag error correction uses extra ancilla qubits to detect potentially problematic high weight errors that arise during the measurement of a stabilizer. We provide a set of requirements for a stabilizer code (along with the circuits used to measure the stabilizers) which, if satisfied, can be used for flag error correction. We are primarily concerned with extending the lifetime of encoded information using fault-tolerant error correction and defer the study of implementing gates fault-tolerantly to future work. Our approach can be applied to a broad class of codes (including but not limited to surface codes, color codes and quantum Reed-Muller codes). Of the three general schemes described above, flag EC has most in common with Shor EC. Further, flag EC does not require verified state preparation, and for all codes considered to date, requires fewer ancilla qubits. Lastly, we note that in order to satisfy the fault-tolerant error correction definition presented in Section 1.1, our protocol applied to distance-three codes differs from [23].

We foresee a number of potential applications of these results. Firstly we believe it is advantageous to have new EC schemes with different properties that can be used in various settings. Secondly, flag EC involves small qubit overhead, hence possibly the schemes presented here and in other flag approaches [23–25] will find applications in early qubit-limited experiments. Thirdly, we expect the flag EC protocol presented here could potentially be useful for LDPC codes as described in [18].

In Sections 2.1 and 2.2 we provide important definitions and introduce flag FTEC for distance-three and -five codes. In Section 2.3 we apply the protocol to two examples: the $[\![19, 1, 5]\!]$ and $[\![17, 1, 5]\!]$ color codes, which importantly have a variety of different weight stabilizers. The general flag FTEC protocol for arbitrary distance codes is given in Section 3.1. A proof that the general protocol satisfies the fault-tolerance criteria is given in Appendix A. In Section 3.2 we provide examples of codes that satisfy the conditions that we required for flag FTEC. Flag circuit constructions for measuring stabilizers of the codes in Section 3.2 are given Section 3.3. We also provide a candidate circuit construction for measuring arbitrary weight stabilizers in Appendix C. In Section 4, we analyze numerically a number of flag EC schemes and compare with other FTEC schemes under various types of circuit level noise. We find that flag EC schemes, which have large numbers of idle qubit locations, behave best in error models in which idle qubit errors occur with a lower probability than CNOT errors. The remainder of this section is devoted to FTEC and noise model/simulation methods.

1.1 Fault-tolerant error correction

Throughout this paper, we assume a simple depolarizing noise model in which idle qubits fail with probability $\tilde{p}$ and all other circuit operations (gates, preparations and measurements) fail with probability p, which recovers standard circuit noise when $\tilde{p} = p$. A detailed description is given in Section 1.2.

The weight of a Pauli operator E ($\mathrm{wt}(E)$) is the number of qubits on which it has non-trivial support. We first make some definitions,

Definition 1. *Weight-t Pauli operators*

$$\mathcal{E}_t = \{E \in \mathcal{P}_n | wt(E) \leq t\}, \tag{1}$$

where $\mathcal{P}_n$ is the n-qubit Pauli group.

Definition 2. *Stabilizer error correction*

Given a stabilizer group $\mathcal{S} = \langle g_1, \cdots, g_m \rangle$, we define the syndrome $s(E)$ to be a bit string, with i'th bit equal to zero if g_i and E commute, and one otherwise. Let $E_{min}(s)$ be a minimal weight correction E where $s(E) = s$. We say operators E and E' are logically equivalent, written as $E \sim E'$, iff $E' \propto gE$ for $g \in \mathcal{S}$.

An error correction protocol typically consists of a sequence of basic operations to infer syndrome measurements of a stabilizer code C, followed by the application of a Pauli operator (either directly or through Pauli frame tracking [22, 26, 27]) intended to correct errors in the system. Roughly speaking, a given protocol is fault-tolerant if for sufficiently weak noise, the effective noise on the logical qubits is even weaker. More precisely, we say that an error correction protocol is a t-FTEC if the following is satisfied:

Definition 3. *Fault-tolerant error correction*

For $t = \lfloor (d-1)/2 \rfloor$, 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 $s \leq t$, no matter how many errors are present in the input state, the output state differs from a codeword by an error of at most weight s.*

Here ideally decoding is equivalent to performing fault-free error correction. By codeword, we mean any state $|\bar{\psi}\rangle \in C$ such that $g|\bar{\psi}\rangle = |\bar{\psi}\rangle \, \forall \, g \in \mathcal{S}$ where $\mathcal{S}$ is the stabilizer group for the code C. Note that for the second criteria in Definition 3, the output and input codeword can differ by a logical operator.

The first criteria in Definition 3 ensures that correctable errors don't spread to uncorrectable errors during the error correction protocol. Note however that the first condition alone isn't sufficient. For instance, the trivial protocol where no correction is ever applied at the end of the EC round also satisfies the first condition, but clearly is not fault-tolerant.

The second condition is not always checked for protocols in the literature, but it is important as it ensures that errors do not accumulate uncontrollably in consecutive rounds of error correction (see [28] for a rigorous proof and [29] for an analysis of the role of input errors in an extended rectangle). To give further motivation as to why the second condition is important, consider a scenario with s faults introduced during each round of error correction, and assume that $t/n < s < (2t+1)/3$ for some integer n (see Fig. 1). Consider an error correction protocol in which r input errors and s faults in an EC block leads to an output state with at most $r+s$ errors[1]. Clearly condition 1 is satisfied.

With the above considerations, an input state $E_1|\bar{\psi}\rangle$ with $\mathrm{wt}(E_1) \leq s$ is taken to $E_2|\bar{\psi}\rangle$, with $\mathrm{wt}(E_2) \leq 2s$ by one error correction round with s faults. After the jth round, the state will be $E_j|\bar{\psi}\rangle$ with the first condition implying $\mathrm{wt}(E_j) \leq j \cdot s$ provided that $j \leq n$. However, when $j > n$, the requirement of the first condition is no longer satisfied so we cannot use it to upper bound $\mathrm{wt}(E_j)$. Now consider the same scenario but assuming both conditions hold. The second condition implies that after the first round, the input state $E_1|\bar{\psi}\rangle$ becomes $E_2'|\bar{\phi}\rangle = E_2|\bar{\psi}\rangle$, with $\mathrm{wt}(E_2') \leq s$, and where $|\bar{\phi}\rangle$ is a codeword. Therefore the codewords are related by: $|\bar{\phi}\rangle = (E_2'^{\dagger} E_2)|\bar{\psi}\rangle$, with logical operator $(E_2'^{\dagger} E_2)$ having weight at most $3s$, since $\mathrm{wt}(E_2) + \mathrm{wt}(E_2') \leq 3s$. However, the minimum non-trivial logical operator of the code has weight $(2t+1) > 3s$, implying that $|\bar{\psi}\rangle = |\bar{\phi}\rangle$, and therefore that $\mathrm{wt}(E_2) = \mathrm{wt}(E_2') \leq s$. Hence, for the jth round, $\mathrm{wt}(E_j) \leq s$ for all j, i.e. the distance from the codeword is not increased by consecutive error correction rounds with s faults, provided $s < (2t+1)/3$.

[1] This is the case for Shor, Steane and Knill EC with appropriately verified ancilla states. However the surface code does not satisfy this due to hook errors but nonetheless still satisfies condition 1 of Definition 3.

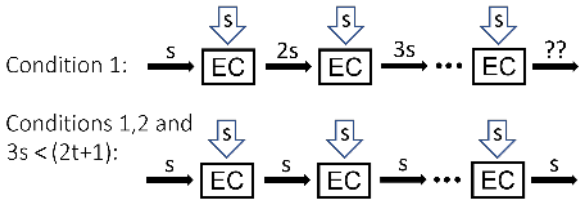

Figure 1: An example showing the first fault tolerance condition alone in Definition 3 is not sufficient to imply a long lifetime. We represent s faults occurring during a round of error correction with a vertical arrow, and a state a distance r from the desired codeword with a horizontal arrow with r above. The first condition alone allows errors to build up over time as in the top figure, which would quickly lead to a failure. However provided $s < (2t+1)/3$, both conditions together ensure that errors in consecutive error correction rounds do not build up, provided each error correction round introduces no more than s faults, which could remain true for a long time.

1.2 Noise model and pseudo-threshold calculations

In Section 4, we perform a full circuit level noise analysis of various error correction protocols. Unless otherwise stated, we use the following depolarizing noise model:

1. With probability p, each two-qubit gate is followed by a two-qubit Pauli error drawn uniformly and independently from $\{I, X, Y, Z\}^{\otimes 2} \setminus \{I \otimes I\}$.

2. With probability $\frac{2p}{3}$, the preparation of the $|0\rangle$ state is replaced by $|1\rangle = X|0\rangle$. Similarly, with probability $\frac{2p}{3}$, the preparation of the $|+\rangle$ state is replaced by $|-\rangle = Z|+\rangle$.

3. With probability $\frac{2p}{3}$, any single qubit measurement has its outcome flipped.

4. Lastly, with probability $\tilde{p}$, each resting qubit location is followed by a Pauli error drawn uniformly and independently from $\{X, Y, Z\}$.

Some error correction schemes that we analyze contain a significant number of idle qubit locations. Consequently, most schemes will be analyzed using three ratios ($\tilde{p} = p$, $\tilde{p} = p/10$ and $\tilde{p} = p/100$) to highlight the impact of idle qubit locations on the logical failure rate.

The two-qubit gates we consider are: CNOT, XNOT= $H_1(\text{CNOT})H_1$, and CZ= $H_2(\text{CNOT})H_2$.

Logical failure rates are estimated using an N-run Monte Carlo simulation. During a particular run, errors are added at each location following the noise model described above. Once the error locations are fixed, the errors are propagated through a fault-tolerant error correction circuit and a recovery operation is applied. After performing a correction, the output is ideally decoded to verify if a logical fault occurred. For an error correction protocol implemented using a stabilizer code C and a fixed value of p, we define the logical failure rate

$$p_{\mathrm{L}}^{(C)}(p) = \lim_{N\to\infty} \frac{N_{\mathrm{fail}}^{(C)}(p)}{N}, \tag{2}$$

where $N_{\mathrm{fail}}^{(C)}(p)$ is the number of times a logical X *or* logical Z error occurred during the N rounds. In practice we take N sufficiently large to estimate $p_{\mathrm{L}}^{(C)}(p)$, and provide error bars [30, 31].

In this paper we are concerned with evaluating the performance of FTEC protocols (i.e. we do not consider performing logical gates fault-tolerantly). We define the pseudo-threshold of an error correction protocol to be the value of p such that

$$\tilde{p}(p) = p_L^{(C)}(p). \tag{3}$$

Note that it is important to have $\tilde{p}$ on the left of Eq. (3) instead of p since we want an encoded qubit to have a lower logical failure rate than an unencoded idle qubit. From the above noise model, a resting qubit will fail with probability $\tilde{p}$.

2 Flag error correction for small distance codes

In this and the next section, we present a t-fault-tolerant flag error correction protocol with distance-$(2t+1)$ codes satisfying a certain condition. Our approach extends that introduced by Chao and Reichardt [23] for distance three codes, which we first review using our terminology in Section 2.1. In Section 2.2 we present the protocol for distance five CSS codes which contains most of the main ideas of the general case (which is provided in Section 3). Lastly, in section Section 2.3 we provide examples of how the protocol is applied to the $[\![19, 1, 5]\!]$ and $[\![17, 1, 5]\!]$ color codes.

2.1 Definitions and Flag 1-FTEC with distance-3 codes

In what follows, we use the term location to refer to a gate, state preparation, measurement or idle qubit where a fault may occur. Note also that a two-qubit Pauli error $P_1 \otimes P_2$ arising at a two-qubit gate location counts as a single fault.

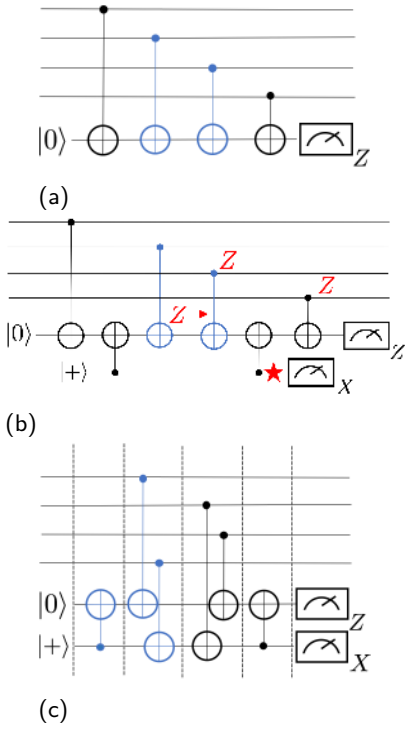

Figure 2: Circuits for measuring the operator $ZZZZ$ (can be converted to any Pauli by single qubit Cliffords). (a) Non-fault-tolerant circuit. A single fault IZ occurring on the third CNOT (from the left) results in the error $IIZZ$ on the data block. (b) Flag version of Fig. 2a. An ancilla (flag) qubit prepared in $|+\rangle$ and two extra CNOT gates signals when a weight two data error is caused by a single fault. Subsequent rounds of error correction may identify which error occurred. Consider an IZ error on the second CNOT, in the non-flag circuit this would result in a weight two error, but in this case, this fault causes the circuit to flag. (c) An alternative flag circuit with lower depth than (b). All bad locations are illustrated in blue.

It is well known that with only a single measurement ancilla, a single fault in a blue CNOT of the stabilizer measurement circuit shown in Fig. 2a can result in a multi-weight error on the data block. This could cause a distance-3 code to fail, or more generally could cause a distance-d code to fail due to fewer than $(d-1)/2$ total faults. We therefore say the blue CNOTs are *bad* according to the following definition:

Definition 4. *Bad locations*

A circuit location in which a single fault can result in a Pauli error E on the data block with $\mathrm{wt}(E) \geq 2$ *will be referred to as a bad location.*

As shown in Fig. 2b, the circuit can be modified by including an additional ancilla (flag) qubit, and two extra

CNOT gates. This modification leaves the bad locations and the fault-free action of the circuit unchanged. However, any single fault leading to an error E with $\text{wt}(E) \geq 2$ will also cause the measurement outcome of the flag qubit to flip [23]. The following definitions will be useful:

Definition 5. *Flags and measurements*

Consider a circuit for measuring a stabilizer generator that includes at least one flag ancilla. The ancilla used to infer the stabilizer outcome is referred to as the measurement qubit. We say the circuit has flagged if the eigenvalue of a flag qubit is measured as -1. If the eigenvalue of a measurement qubit is measured as -1, we will say that the measurement qubit flipped.

The purpose of flag qubits is to signal when high weight data qubit errors result from few fault locations during a stabilizer measurement. Two key definitions are:

Definition 6. *t-flag circuit*

A circuit[2] *$C(P)$ which, when fault-free, implements a projective measurement of a weight-w Pauli P without flagging is a t-flag circuit if the following holds: For any set of v faults at up to t locations in $C(P)$ resulting in an error E with $min(wt(E), wt(EP)) > v$, the circuit flags.*

Note that a t-flag circuit for measuring a weight-t stabilizer P is also a k-flag circuit for any $k > t$. In Section 3.3 we give constructions for some t-flag circuits.

Definition 7. *Flag error set*

Let $\mathcal{E}(g_i)$ be the set of all errors caused by one fault which caused the circuit $C(g_i)$ to flag.

Note that the flag error set can contain the identity as well as weight one errors.

Suppose all errors in a flag error set $\mathcal{E}(g)$ for a 1-flag circuit $C(g)$ have distinct syndromes. As $C(g)$ is a 1-flag circuit, a single fault that leads to an error of weight greater than one will cause the circuit $C(g)$ to flag. Moreover, when a flag has occurred due to at most one fault, a complete set of fault-free stabilizer measurements will infer the resulting element of the flag error set which has been applied to the data qubits. In fact, one would only require distinct syndromes for errors in the flag error set that are logically inequivalent, as defined in Definition 2.

As an example, consider the 1-flag circuit in Fig. 2b. A single fault at any of the blue CNOT gates can lead to an error E_b with $\text{wt}(E_b) \leq 2$ on the data. The set $\mathcal{E}(Z^{\otimes 4})$ contains all errors E_b which resulted from a fault at a blue CNOT gate causing the circuit $C(Z^{\otimes 4})$ of Fig. 2b to flag, i.e., $\mathcal{E}(g) = \{I, Z_{q_3}Z_{q_4}, X_{q_2}Z_{q_3}Z_{q_4}, Z_{q_1}X_{q_2}, Z_{q_4}, X_{q_3}Z_{q_4}, Y_{q_3}Z_{q_4}\}$ with qubits q_1 to q_4.

With the above definitions, we can construct a fault-tolerant flag error correction protocol for $d = 3$ stabilizer codes satisfying the following condition.

Definition 8. ***Flag* 1-*FTEC condition:***

Consider a stabilizer code $\mathcal{S} = \langle g_1, g_2, \cdots, g_r \rangle$ and 1-flag circuits $\{C(g_1), C(g_2), \cdots, C(g_r)\}$. For every generator g_i, all pairs of elements $E, E' \in \mathcal{E}(g_i)$ satisfy $s(E) \neq s(E')$ or $E \sim E'$.

In other words, we require that any two errors that arise when a circuit $C(g_i)$ flags due to a single fault must be either distinguishable or logically equivalent. For the following protocol to satisfy the FTEC conditions in Definition 3, one can assume there is at most 1 fault. If the Flag 1-FTEC condition is satisfied, the protocol is implemented as follows:

Flag 1-FTEC Protocol:
Repeat the syndrome measurement using flag circuits until one of the following is satisfied:

1. If the syndrome s is repeated twice in a row and there were no flags, apply the correction $E_{\min}(s)$.
2. If there were no flags and the syndromes s_1 and s_2 from two consecutive rounds differ, repeat the syndrome measurement using non-flag circuits yielding syndrome s. Apply the correction $E_{\min}(s)$.
3. If a circuit $C(g_i)$ flags, stop and repeat the syndrome measurement using non-flag circuits yielding syndrome s. If there is an element $E \in \mathcal{E}(g_i)$ which satisfies $s(E) = s$, then apply E, otherwise apply $E_{\min}(s)$.

A tree diagram for the flag 1-FTEC Protocol is illustrated in Fig. 3. We now outline the proof that the flag 1-FTEC protocol satisfies the fault-tolerance criteria of Definition 3 (a more rigorous proof of the general case is presented in Appendix A). To show that Flag 1-FTEC Protocol satisfies the criteria of Definition 3, we can assume there is at most one fault during the protocol. If a single fault occurs in either the first or second round leading to a flag, repeating the syndrome measurement will correctly diagnose the error. If there are no flags and a fault occurs which causes the syndromes in the first two rounds to change, then the syndrome during the third round will correctly diagnose the error. There could also be a fault during either the first or second

[2] To avoid confusing the notation of $C(P)$ that represents a circuit and C that represents a code space, we always include the measured Pauli in parenthesis unless clear from context.

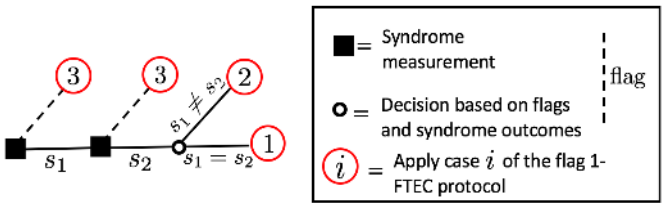

Figure 3: Tree diagram illustrating the possible paths of the Flag 1-FTEC Protocol. Numbers enclosed in red circles at the end of the edges indicate which step to implement in the Flag 1-FTEC Protocol. A dashed line is followed when any of the 1-flag circuits $C(g_i)$ flags. Solid squares indicate a syndrome measurement using 1-flag circuits whereas rings indicate a decision based on syndrome outcomes. Note that the syndrome measurement is repeated at most three times.

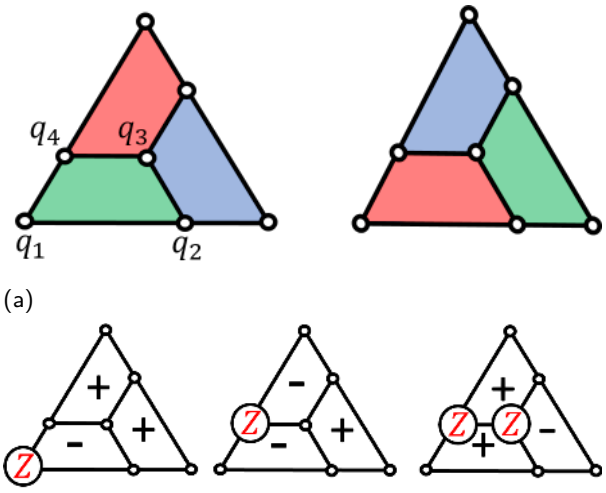

Figure 4: (a) A representation of the Steane code where each circle is a qubit, and there is an X- and a Z-type stabilizer generator for each face. Stabilizer cicuits are specified from that in Fig. 2(a) after rotating the lattice such that the relevant face is on the bottom left. (b) For $g = Z_{q_1}Z_{q_2}Z_{q_3}Z_{q_4}$, the flag error set is $\mathcal{E}(g) = \{I, Z_{q_3}Z_{q_4}, X_{q_2}Z_{q_3}Z_{q_4}, Z_{q_1}X_{q_2}, Z_{q_4}, X_{q_3}Z_{q_4}, X_{q_3}Z_{q_3}Z_{q_4}\}$ which contains all errors arising from a single fault that causes the stabilizer measurement circuit $C(g)$ to flag. Since the Steane code is a CSS code, the X component of an error will be corrected independently allowing us to consider the Z-part of the flag error set $\mathcal{E}_Z(g) = \{I, Z_{q_1}, Z_{q_4}, Z_{q_3}Z_{q_4}\}$. As required, the elements of $\mathcal{E}_Z(g)$ all have distinct syndromes (with satisfied stabilizers represented by a plus).

round that goes undetected. But since there were no flags it cannot spread to an error of weight-2. In this case applying a minimum weight correction based on the measured syndrome of the second round will guarantee that the output codeword differs from a valid codeword by an error of weight at most one. Note that the above argument applies irrespective of any errors on the input state, hence the second criteria of Definition 3 is satisfied. It is worth pointing out that up to three repetitions are required in order to guarantee that the second criteria of Definition 3 is satisfied (unless the code has the property that all states are at most a weight-one error away from a valid codeword, as in [23]).

The Steane code is an example which satisfies the Flag 1-FTEC condition with a simple choice of circuits. To verify this, the representation of the Steane code given in Fig. 4b is useful. There is an X- and a Z-type stabilizer generator supported on the four qubits of each of the three faces. First let us specify all six stabilizer measurement circuits. The circuit that measures $Z_{q_1}Z_{q_2}Z_{q_3}Z_{q_4}$ is specified by taking qubits q_1, q_2, q_3, and q_4 to be the four data qubits in descending order in the 1-flag circuit in Fig. 2b. The other two Z-stabilizer measurement circuits are obtained by first rotating Fig. 4b by 120° and 240° and then using Fig. 2b. The X-stabilizer circuit for each face is the same as the Z-stabilizer circuit for that face, replacing CNOT gates acting on data qubits by XNOT gates. The Z component of the flag error set of the circuit in Fig. 2b is $\mathcal{E}_Z(Z_{q_1}Z_{q_2}Z_{q_3}Z_{q_4}) = \{I, Z_{q_1}, Z_{q_4}, Z_{q_3}Z_{q_4}\}$. As can be seen from Fig. 4b, each of these has a distinct syndrome, thus the measurement circuit for $Z_{q_1}Z_{q_2}Z_{q_3}Z_{q_4}$ satisfies the flag 1-FTEC condition, as do the remaining five measurement circuits by symmetry.

2.2 Flag 2-FTEC with distance-5 codes

Before explicitly describing the conditions and protocol, we discuss some of the complications that arise for codes with $d > 3$.

For distance-5 codes, we must ensure that if two faults occur during the error correction protocol, the output state will differ from a codeword by an error of at most weight-two. For instance, if two faults occur in a circuit for measuring a stabilizer of weight greater than four, the resulting error E on the data should satisfy $\mathrm{wt}(E) \leq 2$ unless there is a flag. In other words, all stabilizer generators should be measured using 2-flag circuits.

In another case, two faults could occur during the measurement of *different* stabilizer generators g_i and g_j. If two bad locations fail and are both flagged, and assuming there are no more faults, the measured syndrome will correspond to the product of the error caused in each circuit (which could have weight greater than

two). Consequently, one should modify Definition 7 of the flag error set to include these types of errors. One then decodes based on the pair of errors that resulted in the measured syndrome, provided logically inequivalent errors have distinct syndromes.

Before stating the protocol, we extend some definitions from Section 2.1.

Consider a stabilizer code $\mathcal{S} = \langle g_1, g_2, \cdots, g_r \rangle$ and t-flag circuits $C(g_i)$ for measuring the generator g_i.

Definition 9. *Flag error set*

Let $\mathcal{E}_m(g_{i_1}, \cdots, g_{i_k})$ be the set of all errors caused by precisely m faults spread amongst the circuits $C(g_{i_1}), C(g_{i_2}), \cdots, C(g_{i_k})$ which all flagged.

Note that there could be more than one fault in a single circuit $C(g_{i_k})$. Examples of flag error sets are given in Table 1 where only contributions from Z errors are included (since the considered code is a CSS code). We also define a general t-fault correction set:

$$\tilde{E}_t^m(g_{i_1}, \cdots, g_{i_k}, s) = \begin{cases} \{E \in \mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}) \times \mathcal{E}_{t-m} \\ \text{ such that } s(E) = s\} \\ \{E_{\min}(s)\} \text{ if above set empty.} \end{cases} \tag{4}$$

By $E \in \mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}) \times \mathcal{E}_{t-m}$, we are considering the set consisting of products between errors caused by k flags and any error of weight $t - m$.

As will be seen below, the correction set will form a critical part of the protocol by specifying the correction applied based on the measured syndrome and flag outcomes over multiple syndrome measurement rounds. In the case where k t-flag circuits flagged caused by $k \leq m \leq t$ faults, the correction applied to the data block will correspond to an element of $\mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}) \times \mathcal{E}_{t-m}$ if the measured syndrome corresponds to an element in this set (there could also be $t - m$ faults which did not give rise to a flag). However in practice, there could be more than t faults and so the measured syndrome may not be consistent with any element of the set $\mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}) \times \mathcal{E}_{t-m}$. In this case, and for the error correction protocol to satisfy the second criteria of Definition 3, the correction will correspond to $E_{\min}(s)$. These features are all included in the set $\tilde{E}_t^m(g_{i_1}, \cdots, g_{i_k}, s)$.

Definition 10. ***Flag 2-FTEC condition:***

Consider a stabilizer code $\mathcal{S} = \langle g_1, g_2, \cdots, g_r \rangle$ and 2-flag circuits $\{C(g_1), C(g_2), \cdots, C(g_r)\}$. For any choice of generators $\{g_i, g_j\}$:

1. $E, E' \in \mathcal{E}_2(g_i, g_j) \Rightarrow s(E) \neq s(E')$ *or* $E \sim E'$,
2. $E, E' \in \mathcal{E}_2(g_i) \cup (\mathcal{E}_1(g_i) \times \mathcal{E}_1) \Rightarrow s(E) \neq s(E')$ *or* $E \sim E'$.

In order to state the protocol, we define an update rule given a sequence of syndrome measurements using t-flag circuits for the counters[3] n_{diff} and n_{same} as follows:

Flag 2-FTEC protocol – update rules:

Given a sequence of consecutive syndrome measurement outcomes s_k and s_{k+1}:

1. If n_{diff} didn't increase in the previous round, and $s_k \neq s_{k+1}$, increase n_{diff} by one.
2. If a flag occurs, reset n_{same} to zero.
3. If $s_k = s_{k+1}$, increase n_{same} by one.

For the following protocol to satisfy Definition 3, one can assume there are at most 2 faults. If the Flag 2-FTEC condition is satisfied, the protocol is implemented as follows:

Flag 2-FTEC protocol – corrections:

Set $n_{\text{diff}} = 0$ and $n_{\text{same}} = 0$.
Repeat the syndrome measurement using flag circuits until one of the following is satisfied:

1. The same syndrome s is repeated $3 - n_{\text{diff}}$ times in a row and there were no flags, apply the correction $E_{\min}(s)$.
2. There were no flags and $n_{\text{diff}} = 2$. Repeat the syndrome measurement using non-flag circuits yielding syndrome s. Apply the correction $E_{\min}(s)$.
3. Some set of two circuits $C(g_i)$ and $C(g_j)$ have flagged. Repeat the syndrome measurement using non-flag circuits yielding syndrome s. Apply any correction from the set $\tilde{E}_2^2(g_i, g_j, s)$.
4. Any circuit $C(g_i)$ has flagged and $n_{\text{diff}} = 1$. Repeat the syndrome measurement using non-flag circuits yielding syndrome s. Apply any correction from the set $\tilde{E}_2^1(g_i, s)$.
5. Any circuit $C(g_i)$ has flagged and $n_{\text{diff}} = 0$ and $n_{\text{same}} = 1$. Use the measured syndrome s from the last round. Apply any correction from the set $\tilde{E}_2^1(g_i, s) \cup \tilde{E}_2^2(g_i, s)$.

Note that when computing the update rules, if a

[3] n_{diff} tracks the minimum number of faults that could have caused the observed syndrome outcomes. For example, if the sequence s_1, s_2, s_1 was measured, n_{diff} would increase by one since a single measurement fault could give rise to the given sequence (for example, this could be caused by a single CNOT failure which resulted in a data qubit and measurement error). However for the sequence s_1, s_2, s_1, s_2, n_{diff} would increase by two.

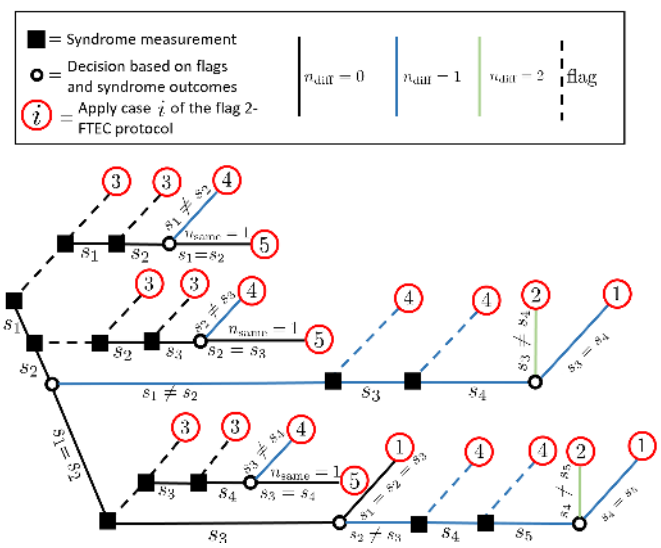

Figure 5: Tree diagram for the Flag 2-FTEC protocol. Numbers encircled in red at the end of the edges indicate which step to implement in the Flag 2-FTEC Protocol. A dashed line is followed when any of the 2-flag circuits $C(g_i)$ flags. Solid squares indicate a syndrome measurement using 2-flag circuits whereas rings indicate a decision based on syndrome outcomes. Edges with different colors indicate the current value of n_{diff} in the protocol. Note that the protocol is repeated at most 6 times.

flag occurs during the j'th round of syndrome measurements, the syndrome is not recorded for that round since all stabilizers must be measured. Thus when computing n_{diff} and n_{same} using consecutive syndromes s_k and s_{k+1}, we are assuming that no flags occurred during rounds k and $k+1$.

In each case of the protocol, the correction sets correspond to those data errors which could arise from up to two faults which are consistent with the conditions of the case. As the elements are logically equivalent (by Eq. (4) and Definition 10), which element is applied is unimportant.

The general protocol for codes of arbitrary distance is given in Section 3.

2.3 Examples of flag 2-FTEC applied to $d = 5$ codes

In this section we give examples of the flag 2-FTEC protocol applied to the 2-dimensional $[\![19, 1, 5]\!]$ and $[\![17, 1, 5]\!]$ color codes, (see Figs. 6a and 6b). We first find 2-flag circuits for all generators (weight-4 and -6 for the 19-qubit code and weight-4 and -8 for the 17-qubit code). We also show that the flag 2-FTEC condition is satisfied for both codes.

For a 2-flag circuit, two faults leading to an error of

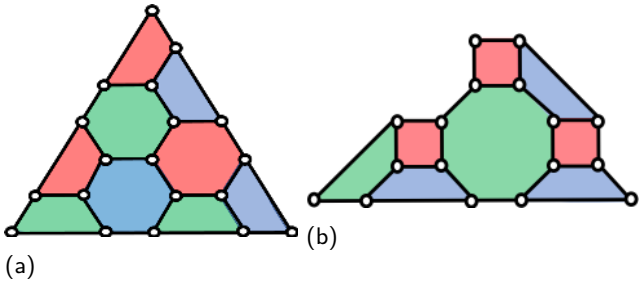

Figure 6: Graphical representation of (a) the 19-qubit 2D color code and (b) the 17-qubit 2D color code. The X and Z stabilizers of the code are symmetric, given by the vertices of each plaquette. Both codes have distance-5.

Weight-4 measurement		Weight-6 measurement	
1-fault	2-faults	1-fault	2-faults
I,Z_1	I,Z_1,Z_2	I,Z_1,Z_6	I,Z_1,Z_2
Z_4	Z_3,Z_4	Z_1Z_2	Z_3, Z_4,Z_5,Z_6
Z_3Z_4	Z_1Z_2	Z_5Z_6	Z_1Z_2,Z_1Z_3
	Z_1Z_4	$Z_4Z_5Z_6$	Z_1Z_4,Z_1Z_5
	Z_2Z_4		Z_1Z_6,Z_2Z_3
			Z_2Z_6,Z_3Z_4
			Z_3Z_6,Z_4Z_5
			Z_4Z_6,Z_5Z_6
			$Z_1Z_2Z_3$,$Z_1Z_5Z_6$
			$Z_2Z_5Z_6$,$Z_3Z_4Z_5$
			$Z_3Z_4Z_6$,$Z_3Z_5Z_6$
			$Z_4Z_5Z_6$

Table 1: Z part of the flag error set of Definition 9 for flag circuits used to measure the stabilizers $g_1 = Z_1Z_2Z_3Z_4$ and $g_3 = Z_1Z_2Z_3Z_4Z_5Z_6$ (we removed errors equivalent up to the stabilizer being measured).

weight greater or equal to 3 (up to multiplication by the stabilizer) must always cause at least one of the flag qubits to flag. As shown in Section 3.3, a 2-flag circuit satisfying these properties can always be constructed using at most four flag qubits. We show 2-flag circuits for measuring weight six and eight generators in Fig. 7.

In Section 3.2, it will be shown that the family of color codes with a hexagonal lattice satisfy a sufficient condition which guarantees that the flag 2-FTEC condition is satisfied. However, there are codes that do not satisfy the sufficient condition but which nonetheless satisfy the 2-Flag FTEC condition. For the 19-qubit and 17-qubit color codes, we verified that the flag 2-FTEC condition was satisfied by enumerating all errors as one would have to for a generic code. In particular, in the case where the 2-flag circuits $C(g_i)$ and $C(g_j)$ flag, the resulting errors belonging to the set $\mathcal{E}_2(g_i, g_j)$ must be logically equivalent or have distinct syndromes (which

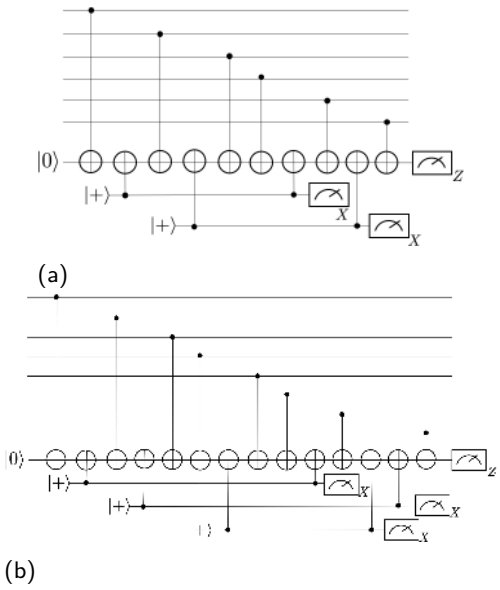

Figure 7: Illustration of 2-flag circuits for measuring (a) $Z^{\otimes 6}$ requiring only two flag qubits and (b) $Z^{\otimes 8}$ requiring only three flag qubits. Flag qubits are prepared in the $|+\rangle$ state, and measurement qubits in the $|0\rangle$ state.

we verified to be true). If a single circuit $C(g_i)$ flags, there could either have been two faults in the circuit or a single fault along with another error that did not cause a flag. If the same syndrome is measured twice in a row after a flag, then errors in the set $\mathcal{E}_2(g_i) \cup (\mathcal{E}_1(g_i) \times \mathcal{E}_1)$ must be logically equivalent or have distinct syndromes (which we verified). If there is a flag but two different syndromes are measured in a row, errors belonging to the set $\mathcal{E}_1(g_i) \times \mathcal{E}_1$ must be logically equivalent or have distinct syndromes (as was already checked). The flag error sets (see Definition 9) for the 19-qubit code can be obtained using the Pauli's shown in Table 1.

Given that the flag 2-FTEC condition is satisfied, the flag 2-FTEC protocol can be implemented following the steps of Section 2.2 and the tree diagram illustrated in Fig. 5.

3 Flag error correction protocol for arbitrary distance codes

In this section we first provide the general flag t-FTEC protocol in Section 3.1. In Section 3.2 we give a sufficient condition for stabilizer codes that allow us to easily prove that flag FTEC can be applied to a number of infinite code families. We show that the families of surface codes, hexagonal lattice color codes and quantum Reed-Muller codes satisfy the sufficient condition. Lastly, in Section 3.3, we give general t-flag circuit constructions which are applicable to the code families described in Section 3.2.

We assume the reader is familiar with all previous definitions. However, to make this section reasonably self contained, we repeat some key definitions below.

Definition 6. *t-flag ciruit* *A circuit $C(P)$ which, when fault-free, implements a projective measurement of a weight-w Pauli P without flagging is a t-flag circuit if the following holds: For any set of v faults at up to t locations in $C(P)$ resulting in an error E with $min(wt(E), wt(EP)) > v$, the circuit flags.*

Definition 9. *Flag error set*
Let $\mathcal{E}_m(g_{i_1}, \cdots, g_{i_k})$ be the set of all errors caused by precisely m faults spread amongst the circuits $C(g_{i_1}), C(g_{i_2}), \cdots, C(g_{i_k})$ which all flagged.

We also remind the reader of the correction set

$$\tilde{E}_t^m(g_{i_1}, \cdots, g_{i_k}, s) = \begin{cases} \{E \in \mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}) \times \mathcal{E}_{t-m} \\ \text{ such that } s(E) = s\} \\ \{E_{\min}(s)\} \text{ if above set empty.} \end{cases} \quad (5)$$

3.1 Conditions and protocol

In what follows we generalize the fault-tolerant error correction protocol presented in Section 2.2 to stabilizer codes of arbitrary distance.

Definition 11. ***Flag t-FTEC condition:***
Consider a stabilizer code $\mathcal{S} = \langle g_1, g_2, \cdots, g_r \rangle$ and t-flag circuits $\{C(g_1), C(g_2), \cdots, C(g_r)\}$. For any set of m stabilizer generators $\{g_{i_1}, \cdots, g_{i_m}\}$ such that $1 \leq m \leq t$, every pair of elements $E, E' \in \bigcup_{j=0}^{t-m} \mathcal{E}_{t-j}(g_{i_1}, \cdots, g_{i_m}) \times \mathcal{E}_j$ either satisfy $s(E) \neq s(E')$ or $E \sim E'$.

The above conditions ensure that if there are at most $t = \lfloor (d-1)/2 \rfloor$ faults, the protocol described below will satisfy the fault-tolerance conditions of Definition 3.

In order to state the protocol, we define an update rule given a sequence of syndrome measurements using t-flag circuits for the counters n_{diff} and n_{same} as follows (see also Section 2.2 and the associated footnote):

Flag t-FTEC protocol – update rules:
Given a sequence of consecutive syndrome measurement outcomes s_k and s_{k+1}:

1. If n_{diff} didn't increase in the previous round, and $s_k \neq s_{k+1}$, increase n_{diff} by one.
2. If a flag occurs, reset n_{same} to zero.
3. If $s_k = s_{k+1}$, increase n_{same} by one.

Flag t-FTEC protocol – corrections:
Set $n_{\text{diff}} = 0$ and $n_{\text{same}} = 0$.
Repeat the syndrome measurement using flag circuits until one of the following is satisfied:

1. The same syndrome s is repeated $t - n_{\text{diff}} + 1$ times in a row and there are no flags, apply the correction $E_{\min}(s)$.

2. There were no flags and $n_{\text{diff}} = t$. Repeat the syndrome measurement using non-flag circuits yielding the syndrome s. Apply the correction $E_{\min}(s)$.

3. Some set of t circuits $\{C(g_{i_1}), \cdots, C(g_{i_t})\}$ have flagged. Repeat the syndrome measurement using non-flag circuits yielding syndrome s. Apply any correction from the set $\tilde{E}_t^t(g_{i_1}, \cdots, g_{i_t}, s)$.

4. Some set of m circuits $\{C(g_{i_1}), \cdots, C(g_{i_m})\}$ have flagged with $1 \leq m < t$ and $n_{\text{diff}} = t - m$. Repeat the syndrome measurement using non-flag circuits yielding syndrome s. Apply any correction from the set $\tilde{E}_t^m(g_{i_1}, \cdots, g_{i_m}, s)$.

5. Some set of m circuits $\{C(g_{i_1}), \cdots, C(g_{i_m})\}$ have flagged with $1 \leq m < t$; $n_{\text{diff}} < t - m$ and $n_{\text{same}} = t - m - n_{\text{diff}} + 1$. Use the syndrome s obtained during the last round and apply any correction from the set $\bigcup_{j=0}^{t-m-n_{\text{diff}}} \tilde{E}_t^{t-j-n_{\text{diff}}}(g_{i_1}, \cdots, g_{i_m}, s)$.

In each case of the protocol, the correction sets correspond to those data errors which could arise from up to t faults which are consistent with the conditions of the case. As the elements are logically equivalent (by Eq. (5) and Definition 11), which element is applied is unimportant.

For the protocol to satisfy the fault-tolerance criteria, the syndrome measurement needs to be repeated a minimum of $t+1$ times. In the scenario where the most syndrome measurement rounds are performed, t identical syndromes are obtained before a fault causes the $t+1$'th syndrome to change (in which case n_{diff} would increase by one). Afterwords, one measures the same syndrome $t-1$ times in a row until another fault causes the syndrome to change. This continues until all of the t possible faults have been exhausted. At this stage, $n_{\text{diff}} = t$ so an extra syndrome measurement round will be performed using non-flag circuits. Thus the maximum number of syndrome measurement rounds $n_{\max}$ is given by

$$n_{\max} = \sum_{j=0}^{t-1}(t-j) + t + 1 = \frac{1}{2}(t^2 + 3t + 2). \quad (6)$$

Note that a similar approach by repeating syndrome measurements is used for Shor error correction [28, 32]. However, our scheme requires fewer syndrome measurement repetitions than is often described for Shor error correction and does not require the preparation and verification of a w-qubit cat state when measuring a stabilizer of weight-w. [4]

For codes that satisfy the flag t-FTEC condition, we also show in Appendix B how to fault-tolerantly prepare and measure logical states using the flag t-FTEC protocol.

3.2 Sufficient condition and satisfying code families

The general flag t-FTEC condition can be difficult to verify for a given code since it depends on precisely which t-flag circuits are used. A sufficient (but not necessary) condition that implies the flag t-FTEC condition is as follows:

Sufficient flag t-FTEC condition:
Given a stabilizer code with distance $d > 1$, and $\mathcal{S} = \langle g_1, g_2, \cdots, g_r \rangle$, we require that for all $v = 0, 1, \ldots t$, all choices Q_{t-v} of $2(t-v)$ qubits, and all subsets of v stabilizer generators $\{g_{i_1}, \ldots, g_{i_v}\} \subset \{g_1, \cdots, g_r\}$, there is no logical operator $l \in N(\mathcal{S}) \setminus \mathcal{S}$ such that

$$\text{supp}(l) \subset \text{supp}(g_{i_1}) \cup \cdots \cup \text{supp}(g_{i_v}) \cup Q_{t-v}, \quad (7)$$

where $N(\mathcal{S})$ is the normalizer of the stabilizer group.

If this condition holds, then the flag t-FTEC condition is implied for any choice of t-flag circuits $\{C(g_1), C(g_2), \cdots, C(g_r)\}$.

To prove this, we must show that it implies that none of the sets appearing in the t-FTEC condition contain elements that differ by a logical operator. Consider the set $\bigcup_{j=0}^{t-m} \mathcal{E}_{t-j}(g_{i_1}, \cdots, g_{i_m}) \times \mathcal{E}_j$ for some set of m stabilizer generators $\{g_{i_1}, \cdots, g_{i_m}\}$ with $1 \leq m \leq t$. An error E from this set will have support in the union of the support of the m stabilizer generators $\{g_{i_1}, \cdots, g_{i_m}\}$, along with up to $t-m$ other single qubits. Another error E' from this set will have support in the union of support of the same m stabilizer generators $\{g_{i_1}, \cdots, g_{i_m}\}$, along with up to $t-m$ other *potentially different* single qubits. If the sufficient condition holds, then $\text{supp}(EE')$ cannot contain a logical operator.

The sufficient flag t-FTEC condition is straightforward to verify for a number of code families with a lot of structure in their stabilizer generators and logical operators. We briefly provide a few examples.

[4] One could also define update rules analogous to those for n_{diff} and n_{same} when implementing Shor-EC which would only require at most $\frac{1}{2}(t^2 + 3t + 2)$ syndrome measurement repetitions as in the flag t-FTEC protocol.

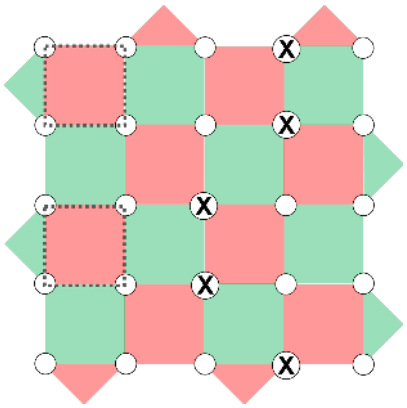

Figure 8: The $d = 5$ rotated surface code. Qubits are represented by white circles, and X and Z stabilizer generators are represented by red and green faces. As in the example, any logical X operator has X operators acting on at least five qubits, with at least one in each row of the lattice, involving an even number in any green face. In this case, no two stabilizer generators can have qubits in five rows, and therefore cannot contain an X type logical operator. The argument is analogous for logical Z operators.

Surface codes flag t-FTEC:

The rotated surface code [7, 33–35] family $[\![d^2, 1, d]\!]$ for all odd $d = 2t + 1$ (see Fig. 8) satisfies the flag t-FTEC condition using any 4-flag circuits.

Firstly, by performing an exhaustive search, we verified that the circuit of Fig. 2b is a 4-flag circuit.

As a CSS code, we can restrict our attention to purely X-type and Z-type logical operators. An X type logical operator must have at least one qubit in each of the $2t+1$ rows of the lattice shown. However, each stabilizer only contains qubits in two different rows. Therefore, with v stabilizer generators, at most $2v$ of the rows could have support. With an additional $2(t - v)$ qubits, at most $2t$ rows can be covered, which is fewer than the number of rows, and therefore no logical X operator is supported on the union of the support of v stabilizers and $2(t-v)$ qubits. An analogous argument holds for Z-type logical operators, therefore the sufficient t-FTEC condition is satisfied.

Color codes flag t-FTEC:

Here we show that any distance $d = (2t+1)$ self-dual CSS code with at most weight-6 stabilizer generators satisfies the flag t-FTEC condition using any 6-flag circuits (see Fig. 10a for an example). Examples include the hexagonal color code [36] family $[\![(3d^2 + 1)/4, 1, d]\!]$ (see Fig. 6a).

As a self-dual CSS code, X and Z type stabilizer generators are identically supported and we can consider a pure X-type logical operator without loss of generality. Consider an X type logical operator l such that

$$\mathrm{supp}(l) \subset \mathrm{supp}(g_{i_1}) \cup \cdots \cup \mathrm{supp}(g_{i_v}) \cup Q_{t-v}, \qquad (8)$$

for some set of v stabilizer generators $\{g_{i_1}, \ldots, g_{i_v}\} \subset \{g_1, \cdots, g_r\}$ along with $2(t-v)$ other qubits Q_{t-v}. Restricted to the support of any of the v stabilizers g_i, $l|_{g_i}$ must have weight 0, 2, 4, or 6 (otherwise it would anti-commute with the corresponding Z type stabilizer). If the restricted weight is 4 or 6, we can produce an equivalent lower weight logical operator $l' = g_i l$, which still satisfies Eq. (8). Repeating this procedure until the weight of the logical operator can no longer be reduced yields a logical operator $l_{\min}$ which has weight either 0 or 2 when restricted to the support of any of the v stabilizer generators. The total weight of $l_{\min}$ is then at most $2v + 2(t - v) = 2t$, which is less than the distance of the code, giving a contradiction which therefore implies that l could not have been a logical operator. An analogous arguments holds for Z-type logical operators, therefore the sufficient t-FTEC condition is satisfied.

This proof can be easily extended to show that any distance $d = (2t + 1)$ self-dual CSS code with at most weight-$2v$ stabilizer generators for some integer v satisfies the flag t'-FTEC condition using any $(v - 1)$-flag circuits, where $t' = t/\lfloor v/2 \rfloor$.

Quantum Reed-Muller codes flag 1-FTEC:

The $[\![n = 2^m - 1, k = 1, d = 3]\!]$ quantum Reed-Muller code family for every integer $m \geq 3$ satisfies the flag 1-FTEC condition using any 1-flag circuits for the standard choice of generators.

We use the following facts about the Quantum Reed-Muller code family (see Appendix D and [37] for proofs of these facts): (1) The code is CSS, allowing us to restrict to pure X type and pure Z type logical operators, (2) all pure X or Z type logical operators have odd support, (3) every X-type stabilizer generator has the same support as some Z-type stabilizer generator, and (4) every Z-type stabilizer generator is contained within the support of an X type generator.

We only need to prove the sufficient condition for $v = 0, 1$ in this case. For $v = 0$, no two qubits can support a logical operator, as any logical operator has weight at least three. For $v = 1$, assume the support of an X-type stabilizer generator contains a logical operator l. That logical operator l cannot be Z type or it would anti-commute with the X-stabilizer due to its odd support. However, by fact (3), there is a Z type stabilizer with the same support as the X type stabilizer, therefore implying l cannot be X type either. Therefore, by contradiction we conclude that no logical operator can be contained in the support of an X stabilizer generator. Since every other stabilizer generator is contained within the support of an X-type stabilizer generator, a logical operator cannot be contained in the

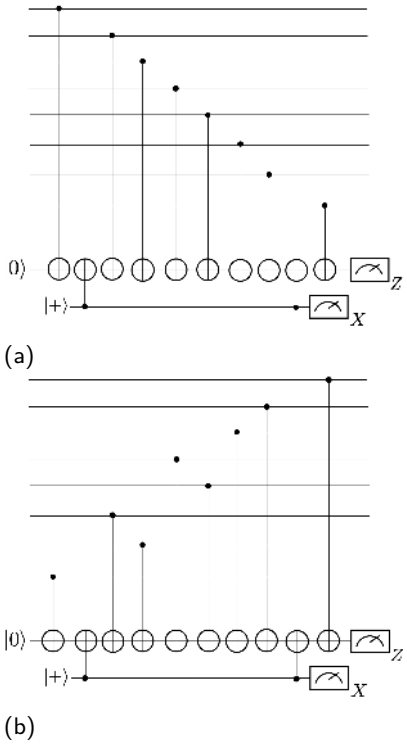

Figure 9: (a) A 1-flag circuit for measuring the stabilizer $Z_8Z_9Z_{10}Z_{11}Z_{12}Z_{13}Z_{14}Z_{15}$ of the $[\![15,7,3]\!]$ Hamming code. However a single fault on the fourth or fifth CNOT can lead to the error $Z_{12}Z_{13}Z_{14}Z_{15}$ on the data which is a logical fault. With the CNOT gates permuted as shown in (b), the $[\![15,7,3]\!]$ satisfies the general flag 1-FTEC condition.

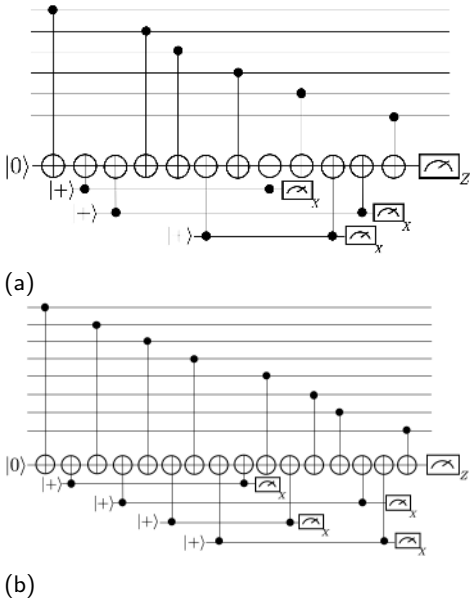

Figure 10: (a) Illustration of a w-flag circuit for measuring the operator $Z^{\otimes w}$ where $w = 6$ using the smallest number of flag qubits. (b) Illustration of a 3-flag circuit for measuring $Z^{\otimes 8}$ using the smallest number of flag qubits.

support of any stabilizer generator.

Note that the Hamming code family has a stabilizer group which is a proper subgroup of that of the quantum Reed-Muller codes described here. The X-type generators of each Hamming code are the same as for a quantum Reed-Muller code, and the Hamming codes are self-dual CSS codes. It is clear that the sufficient condition cannot be applied to the Hamming code since it has even-weight Z-type logical operators (which are stabilizers for the quantum Reed-Muller code) supported within the support of some stabilizer generators.

Codes which satisfy flag t-FTEC condition but not the sufficient flag t-FTEC condition:

Note that there are codes which satisfy the general flag t-FTEC condition but not the sufficient condition presented in this section. An example of such a code is the $[\![5,1,3]\!]$ code (see Table 7 for the codes stabilizer generators and logical operators). Another example includes the Hamming codes as was explained in the discussion on quantum Reed-Muller codes. For instance, consider the $[\![15,7,3]\!]$ Hamming code. Using the 1-flag circuit shown in Fig. 9a, the $[\![15,7,3]\!]$ will not satisfy the general flag 1-FTEC condition since a single fault can lead to a logical error on the data. As was shown in [23], by permuting the CNOT gates resulting in the circuit illustrated in Fig. 9b, the flag 1-FTEC condition is satisfied.

3.3 Circuits

In Section 3.2 we showed that the family of surface codes, color codes with a hexagonal lattice and quantum Reed-Muller codes satisfied a sufficient condition allowing them to be used in the flag t-FTEC protocol. Along with the general 1-flag circuit construction of Fig. 11a, the 6-flag circuit for measuring $Z^{\otimes 6}$ of Fig. 10a can be used as t-flag circuits for all of the codes in Section 3.2. Note that the circuit in Fig. 2b (which is a special case of Fig. 11a when $w = 4$) is a 4-flag circuit which is used for measuring $Z^{\otimes 4}$.

Before describing general 1- and 2-flag circuit constructions, we give the following two definitions which we will frequently use: Any CNOT that couples a data qubit to the measurement qubit will be referred to as CNOT_{dm} and any CNOT coupling a measurement qubit to a flag qubit will be referred to as CNOT_{fm}. In both cases the target qubit will always be the measurement qubit.

1- and 2-flag circuits for weight w stabilizer measurements:

We provide 1- and 2-flag circuit constructions for measuring a weight-w stabilizer. The 1-flag circuit re-

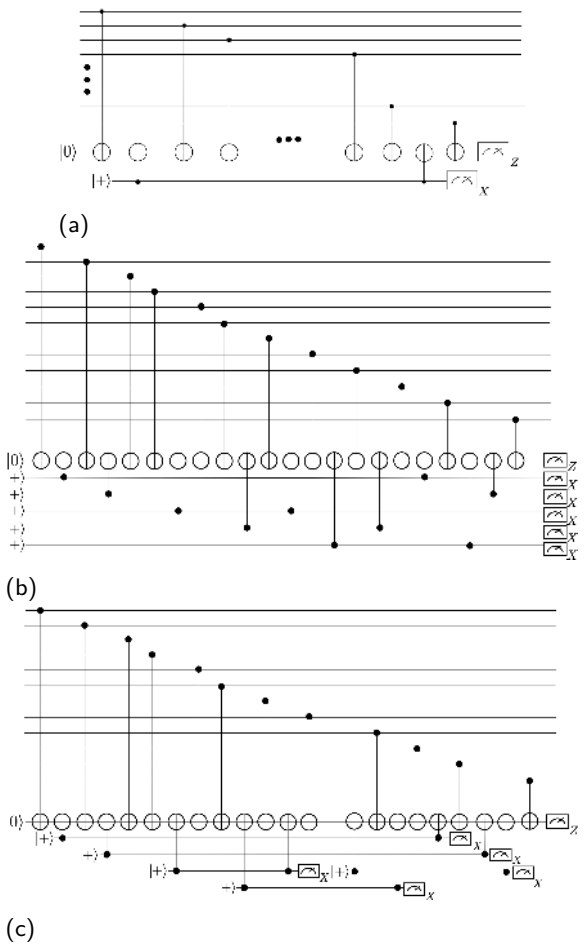

Figure 11: (a) General 1-flag circuit for measuring the stabilizer $Z^{\otimes w}$. (b) Example of a 2-flag circuit for measuring $Z^{\otimes 12}$ using our general 2-flag circuit construction. (c) An equivalent circuit using fewer flag qubits by reusing a measured flag qubit and reinitializing it in the $|+\rangle$ state for use in another pair of CNOT_{fm} gates.

quires a single flag qubit, and the 2-flag circuit requires at most four flag qubits.

Without loss of generality, in proving that the circuit constructions described below are 1- and 2-flag circuits, we can assume that all faults occurred on CNOT gates. This is because any set of v faults (including those at idle, preparation or measurement locations) will have the same output Pauli operator and flag measurement results as some set of at most v faults on CNOT gates (since every qubit is involved in at least one CNOT).

As was shown in Ref. [23], Fig. 11a illustrates a general 1-flag circuit construction for measuring the stabilizer $Z^{\otimes w}$ which requires only two CNOT_{fm} gates. To see that the first construction is a 1-flag circuit, note that an IZ error occurring on any CNOT will give rise to a flag unless it occurs on the first or last CNOT_{dm} gates or the last CNOT_{fm} gate. However, such a fault on any of these three gates can give rise to an error of weight at most one (after multiplying by the stabilizer $Z^{\otimes w}$). One can also verify that if there are no faults, the circuit in Fig. 11a implements a projective measurement of $Z^{\otimes w}$ without flagging. Following the approach in [38], one simply needs to check that the circuit preserves the stabilizer group generated by $Z^{\otimes w}$ and X on each ancilla prepared in the $|+\rangle$ state and Z on each ancilla prepared in the $|0\rangle$ state. By using pairs of CNOT_{fm} gates, this construction satisfies the requirement.

We now give a general 2-flag circuit construction for measuring $Z^{\otimes w}$ for arbitrary w (see Fig. 11b for an example). The circuit consists of pairs of CNOT_{fm} gates each connected to a different flag qubit prepared in the $|+\rangle$ state and measured in the X basis. The general 2-flag circuit construction involves the following placement of $w/2-1$ pairs of CNOT_{fm} gates:

1. Place a CNOT_{fm} pair between the first and second last CNOT_{dm} gates.
2. Place a CNOT_{fm} pair between the second and last CNOT_{dm} gates.
3. After the second CNOT_{fm} gate, place the first CNOT_{fm} gate of the remaining pairs after every two CNOT_{dm} gates. The second CNOT_{fm} gate of a pair is placed after every three CNOT_{dm} gates.

As shown in Fig. 11c, it is possible to reuse some flag qubits to measure multiple pairs of CNOT_{fm} gates at the cost of introducing extra time steps into the circuit. For this reason, at most four flag qubits will be needed, however, if $w \leq 8$, then $w/2-1$ flag qubits are sufficient.

We now show that the above construction satisfies the requirements of a 2-flag circuit. If one CNOT gate fails, by an argument analogous to that used for the 1-flag circuit, there will be a flag or an error of at most weight-one on the data. If the first pair of CNOT_{fm} gates fail causing no flag qubits to flag, after multiplying the data qubits by $Z^{\otimes w}$, the resulting error E_r will have $\text{wt}(E_r) \leq 2$. For any other pair of CNOT_{fm} gates that fail causing an error of weight greater than two on the data, by construction there will always be another CNOT_{fm} gate between the two that fail which will propagate a Z error to a flag qubit causing it to flag. Similarly, if pairs of CNOT_{dm} gates fail resulting in the data error E_r with $\text{wt}(E_r) \geq 2$, by construction there will always be an odd number of Z errors propagating to a flag qubit due to the CNOT_{fm} gates in between the CNOT_{dm} gates that failed causing a flag qubit to flag. The same argument applies if a failure occurs between a CNOT_{dm} and CNOT_{fm} gate.

Lastly, a proposed general w-flag circuit construction for arbitrary w is provided in Appendix C.

Use of flag information:

As seen in Figs. 10a, 10b, 11b and 11c, in general t-flag circuits require more than one flag qubit. Apart from their use in satisfying the t-flag circuit properties, the extra flag qubits could be used to reduce the size of the flag error sets (defined in Definition 9) when verifying the Flag t-FTEC condition of Section 3. To do so, we first define f, where f is a bit string of length u (here u is the number of flag qubits) with $f_i = 1$ if the i'th flag qubit flagged and 0 otherwise. In this case, the correction set of Eq. (5) can be modified to include flag information as follows:

$$\tilde{E}_t^m(g_{i_1}, \cdots, g_{i_k}, s, f_{i_1}, \cdots, f_{i_k}) = \begin{cases} \{E \in \mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}, f_{i_1}, \cdots, f_{i_k}) \times \mathcal{E}_{t-m} \\ \text{ such that } s(E) = s\} \\ \{E_{\min}(s)\} \text{ if above set empty.}, \end{cases} \tag{9}$$

where $\mathcal{E}_m(g_{i_1}, \cdots, g_{i_k}, f_{i_1}, \cdots, f_{i_k})$ is the new flag error set containing only errors caused by precisely m faults spread amongst the circuits $C(g_{i_1}), C(g_{i_2}), \cdots, C(g_{i_k})$ which each gave rise to the flag outcomes $f_{i_1}, \cdots, f_{i_k}$.

Hence only errors which result from the measured flag outcome would be stored in the correction set. With enough flag qubits, this could potentially broaden the family of codes which satisfy the Flag t-FTEC condition.

three-qubit flag EC	pseudo-threshold
$[\![19,1,5]\!]$ and $\tilde{p} = p$	$p_{\text{pseudo}} = (1.14 \pm 0.02) \times 10^{-5}$
$[\![19,1,5]\!]$ and $\tilde{p} = \frac{p}{10}$	$p_{\text{pseudo}} = (6.70 \pm 0.07) \times 10^{-5}$
$[\![19,1,5]\!]$ and $\tilde{p} = \frac{p}{100}$	$p_{\text{pseudo}} = (7.74 \pm 0.16) \times 10^{-5}$

Table 2: Table containing pseudo-threshold values for the flag 2-FTEC protocol applied to the $[\![19,1,5]\!]$ color code for $\tilde{p} = p$, $\tilde{p} = p/10$ and $\tilde{p} = p/100$.

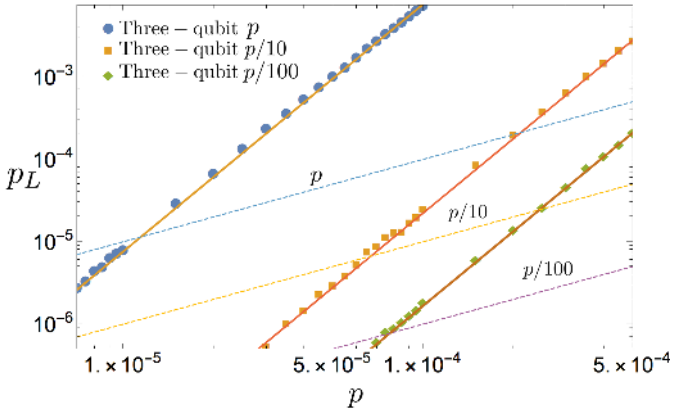

Figure 12: Logical failure rates of the $[\![19,1,5]\!]$ color code after implementing the flag 2-FTEC protocol presented in Section 2.2 for the three noise models described in Section 1.2. The dashed curves represent the lines $\tilde{p} = p$, $\tilde{p} = p/10$ and $\tilde{p} = p/100$. The crossing point between $\tilde{p}$ and the curve corresponding to $p_L^{([19,1,5])}(\tilde{p})$ in Eq. (3) gives the pseudo-threshold.

4 Circuit level noise analysis

The purpose of this section is to demonstrate explicitly the flag 2-FTEC protocol, and to identify parameter regimes in which flag FTEC presented both here and in other works offers advantages over other existing FTEC schemes. In Section 4.1 we analyze the logical failure rates of the $[\![19,1,5]\!]$ color code and compute it's pseudo-threshold for the three choices of $\tilde{p}$. In Section 4.2 we compare logical failure rates of several fault-tolerant error correction schemes applied to distance-three and distance-five stabilizer codes. The stabilizers for all of the studied codes are given in Table 7. Logical failure rates are computed using the full circuit level noise model and simulation methods described in Section 1.2.

4.1 Numerical analysis of the $[\![19,1,5]\!]$ color code

The full circuit-level noise analysis of the flag 2-FTEC protocol applied to the $[\![19,1,5]\!]$ color code was performed using the stabilizer measurement circuits of Figs. 2b and 7a.

In the weight-six stabilizer measurement circuit of Fig. 7a, there are 10 CNOT gates, three measurement and state-preparation locations, and 230 resting qubit locations. When measuring all stabilizer generators using non-flag circuits, there are 42 CNOT and 42 XNOT gates, 18 measurement and state-preparation locations, and 2196 resting qubit locations. Consequently, we expect the error suppression capabilities of the flag EC scheme to depend strongly on the number of idle qubit locations.

Pseudo-thresholds of the $[\![19,1,5]\!]$ code were obtained using the methods of Section 1.2. Recall that for extending the lifetime of a qubit (when idle qubit locations fail with probability $\tilde{p}$), the probability of failure after implementing an FTEC protocol should be smaller than $\tilde{p}$. We calculated the pseudo-threshold using Eq. (3) for the three cases were idle qubits failed with probability $\tilde{p} = p$, $\tilde{p} = p/10$ and $\tilde{p} = p/100$. The results are shown in Table 2.

The logical failure rates for the three noise models are shown in Fig. 12. It can be seen that when the probability of error on a resting qubit decreases from p to $p/10$, the pseudo-threshold improves by nearly a factor of six showing the strong dependence of the scheme on

the probability of failure of idle qubits.

4.2 Comparison of flag 1- and 2-FTEC with other FTEC schemes

The most promising schemes for testing fault-tolerance in near term quantum devices are those which achieve high pseudo-thresholds while maintaining a low qubit overhead. The flag FTEC protocol presented in this paper uses fewer qubits compared to other well known fault-tolerance schemes but typically has increased circuit depth. In this section we apply the flag FTEC protocol of Sections 2.1 and 2.2 to the $[\![5,1,3]\!]$, $[\![7,1,3]\!]$ and $[\![19,1,5]\!]$ codes. We compare logical failure rates for three values of $\tilde{p}$ with Steane error correction applied to the $[\![7,1,3]\!]$ and $[\![19,1,5]\!]$ codes and with the $d=3$ and $d=5$ rotated surface code. More details on Steane error correction and surface codes are provided in Appendices E and F. Note that recent work by Goto has provided optimizations to prepare Steane ancillas [39]. However, our numerical results for Steane-EC were produced using the methods presented in Appendix E.

Results of the logical failure rates for $\tilde{p}=p$, $\tilde{p}=p/10$ and $\tilde{p}=p/100$ are shown in Fig. 13. Various pseudo-thresholds and required time-steps for the considered fault-tolerant error correction methods are given in Tables 3 and 4.

The circuits for measuring the stabilizers of the 5-qubit code were similar to the ones used in Fig. 2b (for an X Pauli replace the CNOT by an XNOT). For flag-FTEC methods, it can be seen that the $[\![5,1,3]\!]$ code always achieves lower logical failure rates compared to the $[\![7,1,3]\!]$ code. However, when $\tilde{p}=p$, both the $d=3$ surface code as well as Steane-EC achieves lower logical failure rates (with Steane-EC achieving the best performance). For $\tilde{p}=p/10$, flag-EC applied to the $[\![5,1,3]\!]$ code achieves nearly identical logical failure rates compared to the $d=3$ surface code. For $\tilde{p}=p/100$, flag 1-FTEC applied to the $[\![5,1,3]\!]$ code achieves lower logical failure rates than the $d=3$ surface code but still has higher logical failure rates compared to Steane-EC.

We also note that the pseudo-threshold increases when $\tilde{p}$ goes from p to $p/10$ for both the $[\![5,1,3]\!]$ and $[\![7,1,3]\!]$ codes when implemented using the flag 1-FTEC protocol. This is primarily due to the large circuit depth in flag-EC protocols since idle qubits locations significantly outnumber other gate locations. For the surface code, the opposite behaviour is observed. As was shown in [9], CNOT gate failures have the largest impact on the pseudo-threshold of the surface code. Thus, when idle qubits have lower failure probability, lower physical error rates will be required in order to achieve better logical failure rates. For instance, if idle qubits never failed, then performing error correction would be guaranteed to *increase* the probability of failure due to the non-zero failure probability of other types of locations (CNOT, measurements and state-preparation). Lastly, the pseudo-threshold for Steane-EC also decreases with lower idle qubit failure rates, but the change in pseudo-threshold is not as large as the surface code. This is primarily due to the fact that all CNOT gates are applied transversally in Steane-EC, so that the pseudo-threshold is not as sensitive to CNOT errors compared to the surface code. Furthermore, most high-weight errors arising during the state-preparation of the logical ancilla's will be detected (see Appendix E). Hence, idle qubit errors play a larger role than in the surface code, but Steane-EC has fewer idle qubit locations compared to flag-EC (see Table 3 for the circuit depths of all schemes).

Although Steane-EC achieves the lowest logical failure rates compared to the other fault-tolerant error correction schemes, it requires a minimum of 35 qubits (more details are provided in Appendix E). In contrast, the $d=3$ surface code requires 17 qubits, and flag 1-FTEC applied to the $[\![5,1,3]\!]$ code requires only 7 qubits. Therefore, if the probability of idle qubit errors is much lower than gate, state preparation and measurement errors, flag-FTEC methods could be good candidates for early fault-tolerant experiments.

It is important to keep in mind that for the flag 1-FTEC protocol applied to the distance-three codes considered in this section, the same ancilla qubits are used to measure all stabilizers. A more parallelized version of flag-FTEC applied to the $[\![7,1,3]\!]$ code using four ancilla qubits instead of two is considered in Appendix G.

In computing the number of time steps required by the flag t-FTEC protocols, a lower bound is given in the case where there are no flags and the same syndrome is repeated $t+1$ times. In Section 3 it was shown that the full syndrome measurement for flag-FTEC is repeated at most $\frac{1}{2}(t^2+3t+2)$ times where $t=\lfloor (d-1)/2 \rfloor$. An upper bound on the total number of required time steps is thus obtained from a worst case scenario where syndrome measurements are repeated $\frac{1}{2}(t^2+3t+2)$ times.

For distance-five codes, the first thing to notice from Fig. 13 is that the slopes of the logical failure rate curves of flag-EC applied to the $[\![19,1,5]\!]$ code and $d=5$ surface code are different from the slopes of Steane-EC applied to the $[\![19,1,5]\!]$ code. In particular, $p_{\mathrm{L}}=cp^3+\mathcal{O}(p^4)$ for flag-EC and the surface code whereas $p_{\mathrm{L}}=c_1p^2+c_2p^3+\mathcal{O}(p^4)$ for Steane-EC (c, c_1 and c_2 are constants that depend on the code and FTEC method). The reason that Steane-EC has non-zero $\mathcal{O}(p^2)$ contributions to the logical failure rates is that there are instances where errors occurring at two different locations can lead to a logical fault. Consequently, the Steane-EC method that was used is not

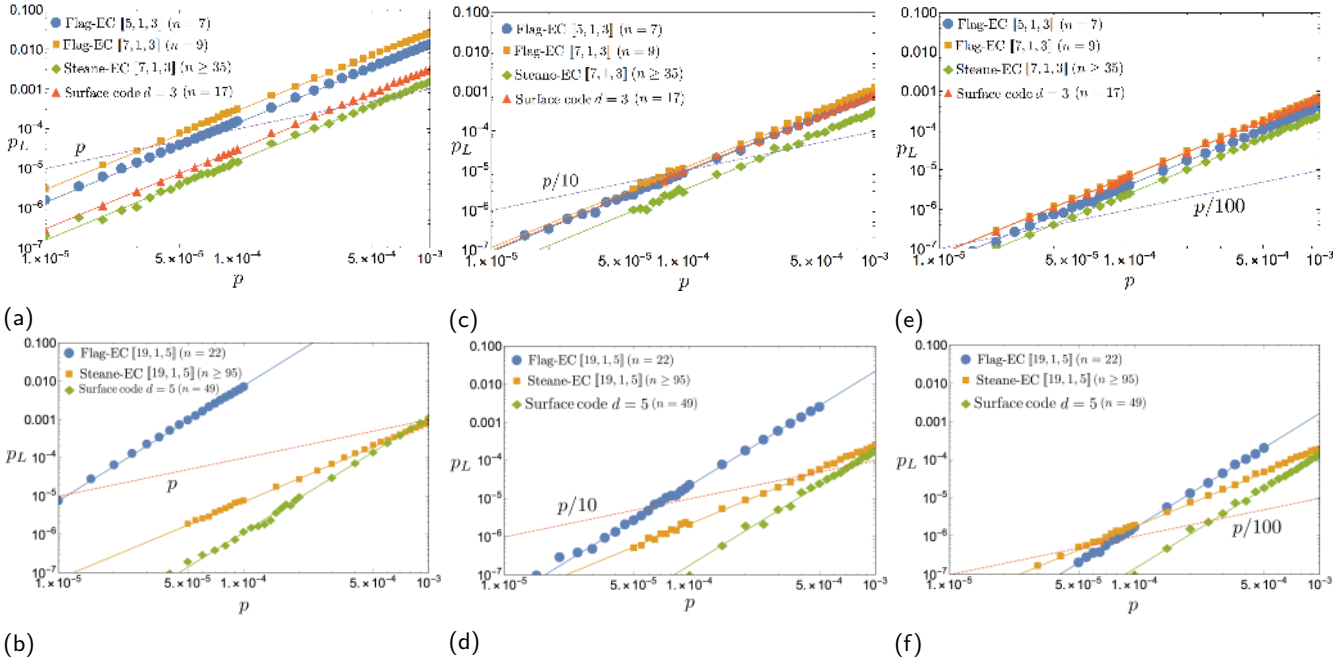

Figure 13: Logical failure rates for various fault-tolerant error correction methods applied to the $[\![5,1,3]\!]$ code, $[\![7,1,3]\!]$ Steane code and the $[\![19,1,5]\!]$ color code. The dashed curves correspond to the lines $\tilde{p}=p$, $\tilde{p}=p/10$ and $\tilde{p}=p/100$. In (a), (c) and (e), the flag 1-FTEC protocol is applied to the $[\![5,1,3]\!]$ and Steane code and the results are compared with the $d=3$ surface code and Steane error correction applied to the Steane code. In (b), (d) and (f), the flag 2-FTEC protocol is applied to the $[\![19,1,5]\!]$ color code, and the results are compared with the $d=5$ surface code and Steane error correction applied to the $[\![19,1,5]\!]$ color code. These numerical results suggest the following fault-tolerant experiments of the schemes we consider for extending the fidelity of a qubit. (1) If $7\leq n\leq 16$, only the 5 and 7 qubit codes with flag 1-FTEC are accessible. However, the performance is much worse than higher qubit alternatives unless $\tilde{p}/p$ is small. (2) For $17\leq n\leq 34$, the $d=3$ surface code seems most promising, unless $\tilde{p}/p$ is small, in which case flag 2-FTEC with the 19-qubit code should be better. (3) For $35\leq n\leq 48$, Steane EC applied to distance-three codes is better than all other approaches studied, except for very low p where flag 2-FTEC should be better due to ability to correct two rather than just one fault. (4) For $n\geq 49$, the d=5 surface code is expected to perform better than the other alternatives below pseudo-threshold.

FTEC scheme	Noise model	Number of qubits	Time steps (T_{time})	Pseudo-threshold
Flag-EC $[\![5,1,3]\!]$	$\tilde{p}=p$	7	$64\leq T_{\text{time}}\leq 88$	$p_{\text{pseudo}}=(7.09\pm0.03)\times10^{-5}$
Flag-EC $[\![7,1,3]\!]$		9	$72\leq T_{\text{time}}\leq 108$	$p_{\text{pseudo}}=(3.39\pm0.10)\times10^{-5}$
$d=3$ Surface code		17	≥ 18	$p_{\text{pseudo}}=(3.29\pm0.16)\times10^{-4}$
Steane-EC $[\![7,1,3]\!]$		≥ 35	15	$p_{\text{pseudo}}=(6.29\pm0.13)\times10^{-4}$
Flag-EC $[\![5,1,3]\!]$	$\tilde{p}=p/10$	7	$64\leq T_{\text{time}}\leq 88$	$p_{\text{pseudo}}=(1.11\pm0.02)\times10^{-4}$
Flag-EC $[\![7,1,3]\!]$		9	$72\leq T_{\text{time}}\leq 108$	$p_{\text{pseudo}}=(8.68\pm0.15)\times10^{-5}$
$d=3$ Surface code		17	≥ 18	$p_{\text{pseudo}}=(1.04\pm0.02)\times10^{-4}$
Steane-EC $[\![7,1,3]\!]$		≥ 35	15	$p_{\text{pseudo}}=(3.08\pm0.01)\times10^{-4}$
Flag-EC $[\![5,1,3]\!]$	$\tilde{p}=p/100$	7	$64\leq T_{\text{time}}\leq 88$	$p_{\text{pseudo}}=(2.32\pm0.03)\times10^{-5}$
Flag-EC $[\![7,1,3]\!]$		9	$72\leq T_{\text{time}}\leq 108$	$p_{\text{pseudo}}=(1.41\pm0.05)\times10^{-5}$
$d=3$ Surface code		17	≥ 18	$p_{\text{pseudo}}=(1.37\pm0.03)\times10^{-5}$
Steane-EC $[\![7,1,3]\!]$		≥ 35	15	$p_{\text{pseudo}}=(3.84\pm0.01)\times10^{-5}$

Table 3: Distance-three pseudo-threshold results for various FTEC protocols and noise models applied to the $[\![5,1,3]\!]$, $[\![7,1,3]\!]$ and $d=3$ rotated surface code. We also include the number of time steps required to implement the protocols.

strictly fault-tolerant according to Definition 3. In Appendix E, more details on the fault tolerant properties

FTEC scheme	Noise model	Number of qubits	Time steps (T_{time})	Pseudo-threshold
Flag-EC $[\![19,1,5]\!]$	$\tilde{p}=p$	22	$504 \leq T_{\text{time}} \leq 960$	$p_{\text{pseudo}} = (1.14 \pm 0.02) \times 10^{-5}$
$d=5$ Surface code		49	≥ 18	$p_{\text{pseudo}} = (9.41 \pm 0.17) \times 10^{-4}$
Steane-EC $[\![19,1,5]\!]$		≥ 95	15	$p_{\text{pseudo}} = (1.18 \pm 0.02) \times 10^{-3}$
Flag-EC $[\![19,1,5]\!]$	$\tilde{p}=p/10$	22	$504 \leq T_{\text{time}} \leq 960$	$p_{\text{pseudo}} = (6.70 \pm 0.07) \times 10^{-5}$
$d=5$ Surface code		49	≥ 18	$p_{\text{pseudo}} = (7.38 \pm 0.22) \times 10^{-4}$
Steane-EC $[\![19,1,5]\!]$		≥ 95	15	$p_{\text{pseudo}} = (4.42 \pm 0.27) \times 10^{-4}$
Flag-EC $[\![19,1,5]\!]$	$\tilde{p}=p/100$	22	$504 \leq T_{\text{time}} \leq 960$	$p_{\text{pseudo}} = (7.74 \pm 0.16) \times 10^{-5}$
$d=5$ Surface code		49	≥ 18	$p_{\text{pseudo}} = (2.63 \pm 0.18) \times 10^{-4}$
Steane-EC $[\![19,1,5]\!]$		≥ 95	15	$p_{\text{pseudo}} = (5.60 \pm 0.43) \times 10^{-5}$

Table 4: Distance-five pseudo-threshold results for various FTEC protocols and noise models applied to the $[\![19,1,5]\!]$ color code and $d=5$ rotated surface code. We also include the number of time steps required to implement the protocols.

of Steane-EC are provided and a fully fault-tolerant implementation of Steane-EC is analyzed (at the cost of using more qubits).

For $d=5$, the surface code achieves significantly lower logical failure rates compared to all other distance 5 schemes but uses 49 qubits instead of 22 for the $[\![19,1,5]\!]$ code. Furthermore, due the differences in the slopes of flag-2 FTEC protocol compared with Steane-EC applied to the $[\![19,1,5]\!]$ code, there is a regime where flag-2 FTEC achieves lower logical failure rates compared to Steane-EC. For $\tilde{p}=p/100$, it can be seen in Fig. 13 that this regime occurs when $p \lesssim 10^{-4}$. We also note that the pseudo-threshold of flag-EC applied to the $[\![19,1,5]\!]$ color code increases for all noise models whereas the pseudo-threshold decreases for the other FTEC schemes. Again, this is due to the fact that flag-EC has a larger circuit depth compared to the other FTEC methods and is thus more sensitive to idle qubit errors.

Comparing the flag 2-FTEC protocol (applied to the $[\![19,1,5]\!]$ color code) to all the $d=3$ schemes that were considered in this section, due to the higher distance of the 19-qubit code, there will always be a parameter regime where the 19-qubit color code acheives lower logical failure rates than both the $d=3$ surface code and Steane-EC applied to the $[\![7,1,3]\!]$ code. In the case where $\tilde{p}=p/100$ and with $p \lesssim 1.5 \times 10^{-4}$, using flag error correction with only 22 qubits outperforms Steane error correction (which uses a minimum of 35 qubits) and the $d=3$ rotated surface code (which uses 17 qubits).

Note the considerable number of time steps involved in a round of flag-EC, particularly in the $d=5$ case (see Table 4). For many applications, this is a major drawback, for example for quantum computation when the time of an error correction round dictates the time of a logical gate. However there are some cases in which having a larger number of time-steps in an EC round while holding the logical error rate fixed is advantageous as it corresponds to a longer physical lifetime of the encoded information. Such schemes could be useful for example in demonstrating that encoded logical quantum information can be stored for longer time scales in the lab using repeated rounds of FTEC.

5 Conclusion

Building on definitions and a new flag FTEC protocol applied to distance-three and -five codes presented in Section 2, in Section 3.1 we presented a general flag FTEC protocol, which we called flag t-FTEC, and which is applicable to stabilizer codes of distance $d=2t+1$ that satisfy the flag t-FTEC condition. The protocol makes use of flag ancilla qubits which signal when v faults lead to errors of weight greater than v on the data when performing stabilizer measurements. In Sections 2.3 and 3.3 we gave explicit circuit constructions, including those needed for distance 3 and 5 codes measuring stabilizers of weight 4, 6 and 8. In Section 3.2 we gave a sufficient condition for codes to satisfy the requirements for flag t-FTEC. Quantum Reed-Muller codes, Surface codes and hexagonal lattice color codes were shown to be families of codes that satisfy the sufficient condition.

The flag t-FTEC protocol could be useful for fault-tolerant experiments performed in near term quantum devices since it tends to use fewer qubits than other FTEC schemes such as Steane, Knill and Shor EC. In Section 4.2 we provided numerical evidence that with only 22 qubits, the flag 2-FTEC protocol applied to the $[\![19,1,5]\!]$ color code can achieve lower logical failure rates than other codes using similar numbers of qubits such as the rotated distance-3 surface code and Steane-EC applied to the Steane code.

A clear direction of future work would be to find optimal general constructions of t-flag circuits for stabiliz-

ers of arbitrary weight that improve upon the general construction given in Appendix C. Of particular interest would be circuits using few flag qubits and CNOT gates while minimizing the probability of false-positives (i.e. when the circuit flags without a high-weight error occurring). Finding other families of stabilizer codes which satisfy the sufficient or more general condition for flag t-FTEC would also be of great interest. One could also envisage hybrid schemes combining flag EC with other FTEC approaches.

Another direction of future research would be to find general circuit constructions for simultaneously measuring multiple stabilizers while minimizing the number of required ancilla qubits. Further, we believe performing a rigorous numerical analysis to understand the impact of more compact circuit constructions on the codes threshold is of great interest.

Lastly, the decoding complexity (i.e. generating the flag error set lookup tables) is limited by the decoding complexity of the code. In some cases, for example concatenated codes, it may be possible to exploit some structure to generate the flag error sets more efficiently. In the case of concatenated code, the decoding complexity would be reduced to the decoding complexity of the codes used at every level. Finding other scalable constructions for efficient decoding schemes using flag error correction remains an open problem.

6 Acknowledgements

The authors would like to thank Krysta Svore, Tomas Jochym-O'Connor, Nicolas Delfosse and Jeongwan Haah for useful discussions. We also thank Steve Weiss for providing the necessary computational tools that allowed us to complete our work in a timely manner. C. C. would like to acknowledge the support of QEII-GSST and thank Microsoft and the QuArC group for its hospitality where all of this work was completed.

References

[1] Benjamin J. Brown, Daniel Loss, Jiannis K. Pachos, Chris N. Self, and James R. Wootton. Quantum memories at finite temperature. *Rev. Mod. Phys.*, 88:045005, Nov 2016. DOI: 10.1103/RevModPhys.88.045005.

[2] A Yu Kitaev. Unpaired majorana fermions in quantum wires. *Physics-Uspekhi*, 44(10S):131, 2001. URL http://stacks.iop.org/1063-7869/44/i=10S/a=S29.

[3] Torsten Karzig, Christina Knapp, Roman M. Lutchyn, Parsa Bonderson, Matthew B. Hastings, Chetan Nayak, Jason Alicea, Karsten Flensberg, Stephan Plugge, Yuval Oreg, Charles M. Marcus, and Michael H. Freedman. Scalable designs for quasiparticle-poisoning-protected topological quantum computation with majorana zero modes. *Phys. Rev. B*, 95:235305, Jun 2017. DOI: 10.1103/PhysRevB.95.235305.

[4] Peter W. Shor. Fault-tolerant quantum computation. *Proceedings., 37th Annual Symposium on Foundations of Computer Science*, pages 56–65, 1996. URL http://dl.acm.org/citation.cfm?id=874062.875509.

[5] A. M. Steane. Active stabilization, quantum computation, and quantum state synthesis. *Phys. Rev. Lett.*, 78:2252–2255, Mar 1997. DOI: 10.1103/PhysRevLett.78.2252.

[6] E. Knill. Scalable quantum computing in the presence of large detected-error rates. *Phys. Rev. A*, 71:042322, Apr 2005. DOI: 10.1103/PhysRevA.71.042322.

[7] Sergey Bravyi and Alexei Kitaev. Quantum codes on a lattice with boundary. *arXiv:quant-ph/9811052*, 1998. URL https://arxiv.org/abs/quant-ph/9811052.

[8] Eric Dennis, Alexei Kitaev, Andrew Landhal, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43:4452–4505, 2002. DOI: 10.1063/1.1499754.

[9] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, Sep 2012. DOI: 10.1103/PhysRevA.86.032324.

[10] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 176–188. ACM, 1997. DOI: 10.1137/S0097539799359385.

[11] John Preskill. Reliable quantum computers. *Proceedings of the Royal Society of London. Se-*

ries A: Mathematical, Physical and Engineering Sciences, 454(1969):385–410, 1998. DOI: 10.1098/rspa.1998.0167.

[12] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Resilient quantum computation. *Science*, 279:342–345, 1998. DOI: 10.1126/science.279.5349.342.

[13] David Poulin. Optimal and efficient decoding of concatenated quantum block codes. *Phys. Rev. A*, 74:052333, Nov 2006. DOI: 10.1103/PhysRevA.74.052333.

[14] David S. Wang, Austin G. Fowler, and Lloyd C. L. Hollenberg. Surface code quantum computing with error rates over 1 *Phys. Rev. A*, 83:020302, Feb 2011. DOI: 10.1103/PhysRevA.83.020302.

[15] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962. DOI: 10.1109/TIT.1962.1057683.

[16] Alexey A. Kovalev and Leonid P. Pryadko. Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Phys. Rev. A*, 87:020304, Feb 2013. DOI: 10.1103/PhysRevA.87.020304.

[17] J. P. Tillich and G. Zmor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2): 1193–1202, Feb 2014. ISSN 0018-9448. DOI: 10.1109/TIT.2013.2292061.

[18] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Info. Comput.*, 14(15-16):1338–1372, November 2014. ISSN 1533-7146. URL http://dl.acm.org/citation.cfm?id=2685179.2685184.

[19] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996. DOI: 10.1103/PhysRevA.54.1098.

[20] Emanuel Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, 2005. DOI: 10.1038/nature03350.

[21] Jesse Fern. An upper bound on quantum fault tolerant thresholds. *arXiv:quant-ph/0801.2608*, 2008. URL https://arxiv.org/abs/0801.2608.

[22] David P. DiVincenzo and Panos Aliferis. Effective fault-tolerant quantum computation with slow measurements. *Phys. Rev. Lett.*, 98:020501, Jan 2007. DOI: 10.1103/PhysRevLett.98.020501.

[23] Rui Chao and Ben W. Reichardt. Quantum error correction with only two extra qubits. *arXiv:quant-ph/1705.02329*, 2017. URL https://arxiv.org/abs/1705.02329.

[24] Rui Chao and Ben W. Reichardt. Fault-tolerant quantum computation with few qubits. *arXiv:quant-ph/1705.05365*, 2017. URL https://arxiv.org/abs/1705.05365.

[25] Theodore J. Yoder and Isaac H. Kim. The surface code with a twist. *Quantum*, 1:2, April 2017. ISSN 2521-327X. DOI: 10.22331/q-2017-04-25-2. URL https://doi.org/10.22331/q-2017-04-25-2.

[26] Barbara M. Terhal. Quantum error correction for quantum memories. *Rev. Mod. Phys.*, 87:307–346, Apr 2015. DOI: 10.1103/RevModPhys.87.307.

[27] Christopher Chamberland, Pavithran Iyer, and David Poulin. Fault-Tolerant Quantum Computing in the Pauli or Clifford Frame with Slow Error Diagnostics. *Quantum*, 2:43, January 2018. ISSN 2521-327X. DOI: 10.22331/q-2018-01-04-43. URL https://doi.org/10.22331/q-2018-01-04-43.

[28] Panos Aliferis, Daniel Gottesman, and John Preskill. Quantum accuracy threshold for concatenated distance-3 codes. *Quantum Info. Comput.*, 6(2):97–165, March 2006. ISSN 1533-7146. URL http://dl.acm.org/citation.cfm?id=2011665.2011666.

[29] Andrew W. Cross, David P. Divincenzo, and Barbara M. Terhal. A comparative code study for quantum fault tolerance. *Quantum Info. Comput.*, 9(7):541–572, July 2009. ISSN 1533-7146. URL http://dl.acm.org/citation.cfm?id=2011814.2011815.

[30] Panos Aliferis and Andrew W. Cross. Subsystem fault tolerance with the bacon-shor code. *Phys. Rev. Lett.*, 98:220502, May 2007. DOI: 10.1103/PhysRevLett.98.220502.

[31] Christopher Chamberland, Tomas Jochym-O'Connor, and Raymond Laflamme. Overhead analysis of universal concatenated quantum codes. *Phys. Rev. A*, 95:022313, Feb 2017. DOI: 10.1103/PhysRevA.95.022313.

[32] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. *Proceedings of Symposia in Applied Mathematics*, 68:13–58, 2010. URL https://arxiv.org/abs/0904.2557.

[33] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2 – 30, 2003. ISSN 0003-4916. DOI: 10.1016/S0003-4916(02)00018-0.

[34] Yu Tomita and Krysta M. Svore. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A*, 90:062320, Dec 2014. DOI: 10.1103/PhysRevA.90.062320.

[35] Xiao-Gang Wen. Quantum orders in an exact soluble model. *Phys. Rev. Lett.*, 90:016803, Jan 2003. DOI: 10.1103/PhysRevLett.90.016803.

[36] H. Bombin and M. A. Martin-Delgado. Topological

quantum distillation. *Phys. Rev. Lett.*, 97:180501, Oct 2006. DOI: 10.1103/PhysRevLett.97.180501.

[37] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. *Phys. Rev. Lett.*, 113:080501, Aug 2014. DOI: 10.1103/PhysRevLett.113.080501.

[38] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. Fault-tolerant quantum computing with color codes. *arXiv:1108.5738*, 2011. URL https://arxiv.org/abs/1108.5738.

[39] Hayato Goto. Minimizing resource overheads for fault-tolerant preparation of encoded states of the steane code. *Scientific Reports*, (6):19578, 2016. DOI: 10.1038/srep19578.

[40] Adam Paetznick and Ben W. Reichardt. Fault-tolerant ancilla preparation and noise threshold lower boudds for the 23-qubit golay code. *Quantum Info. Comput.*, 12(11-12):1034–1080, November 2012. ISSN 1533-7146. URL http://dl.acm.org/citation.cfm?id=2481569.2481579.

[41] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965. DOI: 10.4153/CJM-1965-045-4.

[42] Vladimir Kolmogorov. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1(1):43–67, 2009. DOI: 10.1007/s12532-009-0002-8.

[43] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect quantum error correcting code. *Phys. Rev. Lett.*, 77:198–201, Jul 1996. DOI: 10.1103/PhysRevLett.77.198.

[44] Andrew W. Steane. Multiple-Particle Interference and Quantum Error Correction. *Proc. Roy. Soc. Lond.*, 452:2551–2577, 1996. URL http://www.jstor.org/stable/52827.

A Proof that the flag t-FTEC protocol satisfies the fault-tolerance criteria of Definition 3

Consider the flag t-FTEC protocol described in Section 3.1.

Claim 1. *If the flag t-FTEC condition is satisfied, then both fault-tolerance criteria of Definition 3 will be satisfied.*

Proof. First note that the protocol always terminates. As was shown in the arguments leading to Eq. (6) presented in Section 3.1, the maximum number of syndrome measurement rounds is $\frac{1}{2}(t^2+3t+2)$.

To prove fault-tolerance, in what follows we assume that there are at most t-faults during the protocol. Also, we define a benign fault to be a fault that either leaves all syndrome measurements in the protocol unchanged.

By repeating the syndrome measurement using t-flag circuits, the following cases exhaust all possible errors for the occurrence of at most t faults.

Case 1: The same syndrome is measured $t-n_{diff}+1$ times in a row and there are no flags.

At any time during the protocol, if there are no flags, there can be at most $t-n_{\text{diff}}$ remaining faults that occur (since it is guaranteed that there were at least n_{diff} faults). Therefore, if the same syndrome was measured $t-n_{\text{diff}}+1$ times in a row, at least one round (say r) had to have been fault-free yielding the correct syndrome corresponding to the data qubit errors present at that time. Applying $E_{\min}(s)$ will remove those errors. Furthermore, since all syndrome measurements are identical and there are no flags, there can be at most $t-n_{\text{diff}}$ errors which are introduced on the data blocks from faults during the $t-n_{\text{diff}}+1$ syndrome measurement rounds (excluding round r). Since none of the errors change the syndrome, after applying the correction, the output state can differ from the input codeword by an error of weight at most $t-n_{\text{diff}}$ (if the total number of faults and input errors was t). For input states afflicted by an error of arbitrary weight, the output state will differ from a valid codeword (but not necessarily the input codeword) by an error of weight at most $t-n_{\text{diff}}$. Thus both conditions of Definition 3 are satisfied.

Case 2: There are no flags and $n_{diff}=t$.

The only way that $n_{\text{diff}}=t$ is if there were t-faults that each changed the syndrome measurement outcome. Further since there were no flags, an error E afflicting the data qubits must satisfy $\text{wt}(E)\leq t$. Thus repeating the syndrome measurement using non-flag circuits will correctly identify and remove the error in the case where the number of input errors and faults is t or project the system back to the code space (to a possibly differ codeword) if there were t faults and the input state was afflicted by an error of arbitrary weight .

Case 3: A set of t circuits $\{C(g_{i_1}),\cdots,C(g_{i_t})\}$ flagged.

Since t circuits $\{C(g_{i_1}),\cdots,C(g_{i_t})\}$ flagged, then no other faults can occur during the protocol. Hence, when repeating the syndrome measurement using non-flag circuits, the measured syndrome will correspond to an error $E_r\in\tilde{E}_t^t(g_{i_1},\cdots g_{i_t},s)$. Since from the flag t-FTEC condition all elements of $\tilde{E}_t^t(g_{i_1},\cdots g_{i_t},s)$ are logically equivalent, the product of errors resulting from the flag circuits $\{C(g_{i_1}),\cdots,C(g_{i_t})\}$ will be corrected.

Note that for an input error E_{in} of arbitrary weight and since the final round must be error free, applying a correction a correction from the set $\tilde{E}_t^t(g_{i_1},\cdots g_{i_t},s)$ is guaranteed to return the system to the codespace. Thus both conditions of Definition 3 are satisfied.

Case 4: The m circuits $\{C(g_{i_1}),\cdots,C(g_{i_m})\}$ flagged with $1\leq m<t$, $n_{diff}=t-m$.

Here we can assume that at any point during the protocol and after the j'th flag, the syndrome never repeated more than $t-j-n_{\text{diff}}$ times. Otherwise case 5 of the protocol would already have occurred.

As m circuits $\{C(g_{i_1}),\cdots,C(g_{i_m})\}$ have flagged and $n_{\text{diff}}=t-m$, then there can be no more faults. The final syndrome measurement using non-flag circuits will yield a syndrome corresponding to an error in the set $\tilde{E}_t^m(g_{i_1},\cdots g_{i_m},s)$ (and all elements are logically equivalent from the flag t-FTEC condition). Applying a recovery operator chosen from this set will thus remove the errors afflicting the data. If the input state differs from a valid codeword by an error of arbitrary weight, by definition of $\tilde{E}_t^m(g_{i_1},\cdots g_{i_m},s)$ the output state will be a valid codeword.

Case 5: The m circuits $\{C(g_{i_1}),\cdots,C(g_{i_m})\}$ flagged with $1\leq m<t$, $n_{same}=t-m-n_{diff}+1$.

Given that m circuits $\{C(g_{i_1}),\cdots,C(g_{i_m})\}$ flagged, there are r remaining faults that don't result in a flag with $n_{\text{diff}}\leq r\leq t-m$. In this case, after the m'th flag, the syndrome measurement was repeated using t-flag circuits $t-m-n_{\text{diff}}+1$ times in a row and all syndromes were the same. It is thus guaranteed that at least one of the syndrome measurements s was fault-free and correctly identified the errors arising from the flags and errors causing the syndrome to change giving n_{diff} (along with some error E which did not cause the circuits to flag with $\text{wt}(E)\leq t-m-n_{\text{diff}}$). Consequently, if there are no errors on the input state, the overall error on the data will be EE_r with $E_r\in\bigcup_{j=0}^{t-m-n_{\text{diff}}}\tilde{E}_t^{t-j-n_{\text{diff}}}(g_{i_1},\cdots,g_{i_m},s)$. Since all elements in $\bigcup_{j=0}^{t-m-n_{\text{diff}}}\tilde{E}_t^{t-j-n_{\text{diff}}}(g_{i_1},\cdots,g_{i_m},s)$ are logically equivalent from the flag t-FTEC condition, by choosing a correction from this set, the output state can

differ from the input codeword by an error of at most weight $t - m - n_{\text{diff}}$.

If there is an input error of arbitrary weight, then again one of the $t - m - n_{\text{diff}} + 1$ rounds will have the correct syndrome s. The actual state of the data qubits after the protocol can differ from the state which had the correct syndrome by an error of weight at most $t - m - n_{\text{diff}}$. Therefore applying any correction with syndrome s will return the system to the code space up to an error of weight at most $t - m - n_{\text{diff}}$.

□

B Fault-tolerant state preparation and measurement using flag t-FTEC

In this section we show how to fault-tolerantly prepare a logical $|\overline{0}\rangle$ state and how to perform fault-tolerant measurements for codes that satisfy the flag t-FTEC condition of Section 3. Note that there are several methods that can be used for doing so. Here we follow a procedure similar to that shown in [32] when performing Shor EC. However, compared to Shor EC, the flag t-FTEC protocol requires fewer qubits. Furthermore, postselection is not necessary.

Consider an n-qubit stabilizer code C with stabilizer group $\mathcal{S} = \langle g_1, \cdots, g_{n-k} \rangle$ that can correct up to t errors. Notice that the encoded $|\overline{0}\rangle$ state is a $+1$ eigenstate of the logical $\overline{Z}$ operator and all of the codes stabilizer generators. For k encoded qubits, $|\overline{0}\rangle$ would be $+1$ eigenstate of $\{\overline{Z}_1, \cdots \overline{Z}_k\}$ and all of the codes stabilizers. For notational simplicity, in what follows we assume $k = 1$.

The state $|\overline{0}\rangle$ is a stabilizer state completely specified by the full stabilizer generators of $\mathcal{S}$ and $\overline{Z}$. We can think of $\mathcal{S}' = \langle g_1, \cdots g_{n-1}, \overline{Z} \rangle$ as a stabilizer code with zero encoded qubits and a $2^0 = 1$ dimensional Hilbert space. Thus any state which is a $+1$ eigenstate of all operators in $\mathcal{S}'$ will correspond to the encoded $|\overline{0}\rangle$ state.

Now, suppose we prepare $|\overline{0}\rangle_{\text{in}}$ using a non-fault-tolerant encoding and perform a round of flag t-FTEC using the extended stabilizers $\langle g_1, \cdots g_{n-1}, \overline{Z} \rangle$. Then by the second criteria of Definition 3, the output state $|\overline{0}\rangle_{\text{out}}$ is guaranteed to be a valid codeword with at most t single-qubit errors. But for the extended stabilizers $\langle g_1, \cdots g_{n-1}, \overline{Z} \rangle$ there is only *one* valid codeword which corresponds to the encoded $|\overline{0}\rangle$ state. In fact, by the second criteria of Definition 3, any n-qubit input state prepared using non-fault-tolerant circuits is guaranteed to be an encoded $|\overline{0}\rangle$ state if there are no more than t faults in the EC round.

We point out that the flag t-FTEC condition of Section 3.1 is trivially satisfied for $\mathcal{S}'$ since the codes logical operators are now stabilizers. In other words, if two errors belong to the set $\tilde{E}_t^m(g_{i_1}, \cdots, g_{i_k}, s)$, then their product will always be a stabilizer. Therefore, the flag t-FTEC protocol can always be applied for the code $\mathcal{S}'$.

To summarize, the encoded $|\overline{0}\rangle$ state can be prepared by first preparing any n-qubit state using non-fault-tolerant circuits followed by applying a round of flag t-FTEC using the extended stabilizers $\langle g_1, \cdots g_{n-1}, \overline{Z} \rangle$. This guarantees that the output state will be the encoded $|\overline{0}\rangle$ state with at most t single-qubit errors.

Now suppose we want to measure the eigenvalue of a logical operator $\overline{P}$ where P is a Pauli. If C is a CSS code and the logical operator being measured is X or Z, one could measure the eigenvalue by performing the measurement transversally. So suppose C is not a CSS code. From [32] we require that performing a measurement with s faults on an input state with r errors ($r + s \leq t$) is equivalent to correcting the r errors and performing the measurement perfectly. The protocol for fault-tolerantly measuring the eigenvalue of $\overline{P}$ is described as follows:

1. Perform flag t-FTEC.
2. Use a t-flag circuit to measure the eigenvalue of $\overline{P}$.
3. Repeat steps 1 and 2 $2t + 1$ times and take the majority of the eigenvalue of $\overline{P}$.

Step 1 is used to remove input errors to the measurement procedure. However during error correction, a fault can occur which could cause a new error on the data. Thus by repeating the measurement without performing error correction, the wrong state would be measured each time if there were no more faults. But repeating the syndrome $2t + 1$ times, it is guaranteed that at least $t + 1$ of the syndrome measurements had no faults and that the correct eigenvalue of $\overline{P}$ was measured. Thus taking the majority of the measured eigenvalues will give the correct answer.

Note that during the fault-tolerant measurement procedure, if there is a flag either during the error correction round or during the measurement of $\overline{P}$, when error correction is performed one corrects based on the possible set of errors resulting from the flag.

C Candidate general w-flag circuit construction

In this section we provide a candidate general w-flag circuit construction for measuring the stabilizer $Z^{\otimes w}$. Although we do not provide a rigorous proof that our construction results in a w-flag circuit, we give several arguments as evidence that it satisfies all the criteria of a w-flag circuit. An illustration of the circuit construction (for $w = 12$) is given in Fig. 14 and the description

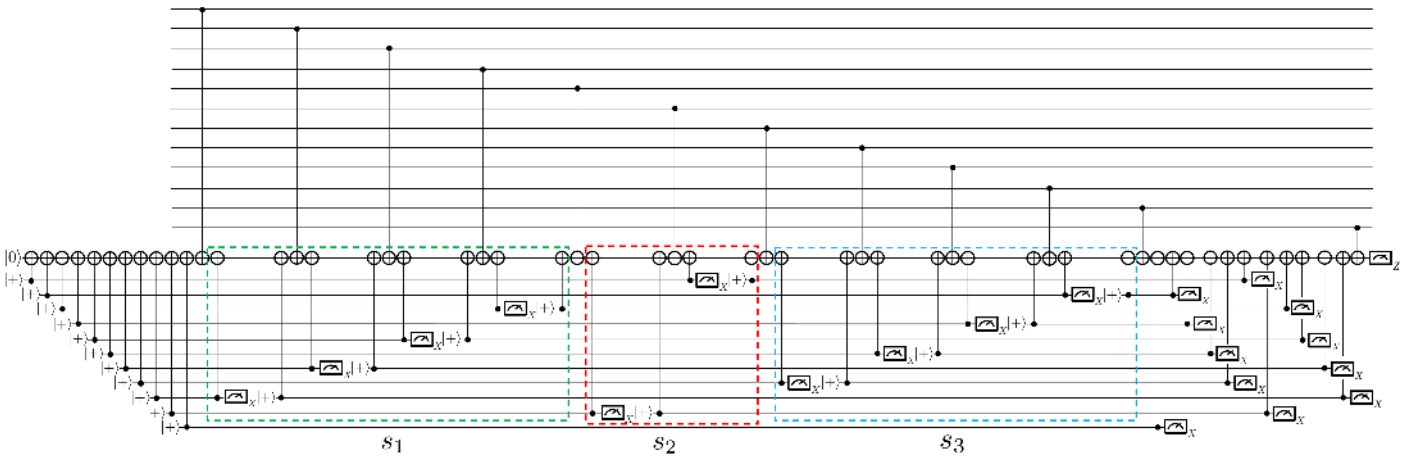

Figure 14: Illustration of the general w-flag circuit construction for $w = 12$. In general, the circuit requires $w - 1$ flag qubits and is implemented using $7w - 8$ time steps. The circuit consists of two families of CNOT_{fm} gates. For the first family, with the first set of CNOT_{fm} gates located before the first CNOT_{dm} gate, the partnering CNOT_{fm} gates are divided into three sets s_1, s_2 and s_3 which are enclosed in the green, red and blue dashed boxes. In general, s_1 and s_3 both contain $(w-4)/2$ CNOT_{fm} gates. In s_1, the j'th control qubit is at position $w + 2(j + 1)$ and in s_3 it is at position $w + 2j + 1$ with $j \in \{1, 2, \cdots, (w-4)/2\}$ In s_2, the control qubits are always located at the $w + 2$'th and $2w - 1$'th qubits. Lastly, note that qubits are reused for implementing the second family of CNOT_{fm} gates. The partnering CNOT_{fm} gates are located in between the $w - 1$ and w'th CNOT_{dm} gates following an identical pattern as in s_1, s_2 and s_3 (in s_1 and s_3 the CNOT's are implemented in reverse order).

for how the circuit is constructed for arbitrary w is provided in the caption.

In what follows, we can restrict our attention to the case in which all v faults occur on CNOT gates in the circuit. The effect on the measurement outcomes and data qubits due to a set of v faults that include faults at idle and measurement locations can always occur due to at most v faults at CNOT locations only (as every qubit is involved in at least one CNOT). Moreover, we can assume that for CNOT_{fm} gates, the faults belong to the set $\{IZ, ZI, ZZ\}$ since X errors would never propagate to the data or affect the measurement outcome of a flag qubit. For CNOT_{dm} gates, we can assume that faults belong to the set $\{XZ, XI\}$. We only consider Z errors on the target qubit of a CNOT_{dm} for the same reason that was given for CNOT_{fm} gates. For the control qubit, an X errors guarantees that the weight of the data qubit error increases even after the application of a satbilizer (since we are measuring $Z^{\otimes w}$).

We will use the following useful terminology: we say that a single-qubit Pauli at a time step in the circuit propagates to a qubit at a particular time-step if it would do so in the fault-free circuit. Given a single-qubit Pauli at a time step in the circuit, we say that another qubit is affected by the Pauli if it propagates to that qubit in any time step.

We now provide arguments for why the circuit is a w-flag circuit. First, note that every CNOT_{fm} gate comes as part of a pair with the measurement qubit being the target qubit. This ensures that when the circuit is fault-free, it implements a projective measurement of $Z^{\otimes w}$ without flagging. Next, notice that apart from the last two CNOT_{dm} gates, each CNOT_{dm} gate is followed by two CNOT_{fm} gates, one with its partnering CNOT_{fm} located before the first CNOT_{dm} and the other partner is located in between the last two CNOT_{dm} gates. Thus if there is a single Z error on the measurement qubit which propagates to any of the data qubits, the circuit will flag.

In all circuits considered in this section, s_0 will correspond to the sequence of CNOT_{fm} gates that come before the first CNOT_{dm} gate. First consider the shorter circuit construction using only the first family of CNOT_{fm} gates from the construction in Fig. 14 (see the example in Fig. 15). We can separate the set of all locations into subsets including two CNOT_{fm} gates and one CNOT_{dm} gate as shown in Fig. 16 (apart from the last CNOT_{dm}). This circuit segment can increase the weight of the data error by at most one. There are four cases with inputs on the measurement qubit before the first CNOT_{fm} and CNOT_{dm} being $\{(I, I), (I, Z), (Z, I), (Z, Z)\}$. Note that if the following property held for each segment, then the circuit would be w-flag: for all inputs to the segment, if the weight of the data error increases and there are no faults in the segment, the segment flags. Unfortunately, for the input (Z, Z), this is not the case. Both input Z must come from at least two faults.

Note that if v faults results in a data qubit error of weight greater than v without causing the circuit in

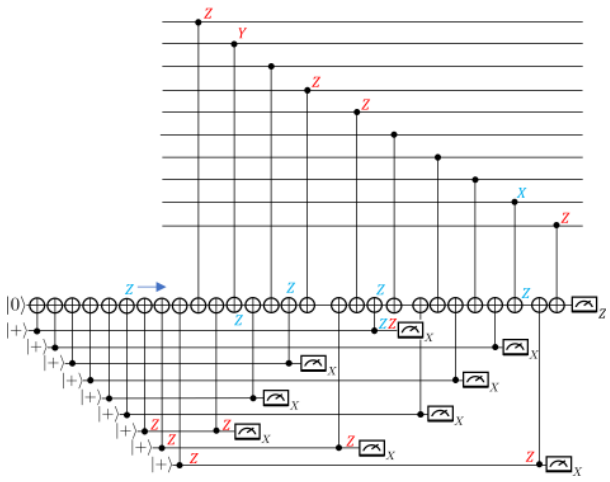

Figure 15: Example of five faults that lead to an error of weight six on the data without causing a flag when only the first family of CNOT_{fm} gates are used in the construction of Fig. 14 (here $w = 10$). Errors arising from faults are shown in blue and the resulting errors after propagating through the CNOT gates are shown in red.

Fig. 15 to flag, there must be either an IZ fault followed by no fault in a consecutive pair of CNOT_{fm} gates belonging to s_0 or a ZZ fault followed by two CNOT_{fm} gates that don't fail in s_0.

Moreover, a poor choice of ordering of the CNOT_{fm} gates in s_1, s_2 and s_3 can result in four faults causing a weight $\frac{w}{2} + 1$ error on the data without causing the circuit to flag. Therefore, the ordering of the CNOT_{fm} gates in the sets s_1, s_2 and s_3 is chosen such that most Z errors in s_0 that first propagate to flag qubits connected to gates in s_1, will then propagate to flag qubits in s_3 and vice-versa. Typically, if a Z error propagates through multiple CNOT_{dm} gates in s_1, then unless CNOT_{fm} gates in s_3 fail, the flag qubits affected by the Z error would flag. Furthermore, the total number of required failures for gates in s_3 to cancel the Z errors will typically be equal to the number of times the Z error propagated to the data.

There are however cases which don't flag in which v faults in the circuit construction presented in Fig. 15 lead to more than v errors on the data qubit, such as the example given in the figure. All such problematic cases that we found had a Z error on the target qubit in one of the last few CNOT_{fm} gates in s_0, followed by a Z error on the target qubit in one of the first few CNOT_{dm} gates in s_1. Then further Z errors occur throughout the remainder of the circuit which propagate to the data while preventing the flag qubits affected by the previous errors from flagging. Further, a Z error on the control qubit of the second CNOT_{fm} in s_2 cancels the Z which propagates to the flag qubit coupled to that CNOT_{fm} gate.

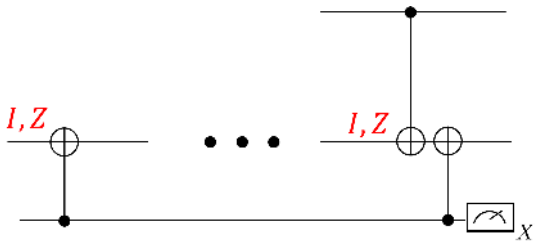

Figure 16: Illustration of a pair of CNOT_{fm} gates as well as a segment of CNOT_{dm} followed by CNOT_{fm} gate. The first CNOT_{fm} gate belongs to the sequence of CNOT_{fm} gates that come before the first CNOT_{dm} gate (see Fig. 14).

This particular problematic fault pattern would lead to flags if it occurred within the full circuit construction of Fig. 14 (if the additional locations of the larger circuit do not fail). As this was the only type of problematic fault pattern that we found, one would hope that all problematic fault patterns are rendered non problematic provided no additional locations fail. Since the additional CNOT_{fm} gates always occur immediately after one of the original CNOT_{fm} gates (or after the last CNOT_{fm} gate), as far as the flag properties of the original circuit are concerned, no new problematic fault patterns are introduced.

We conclude this section by noting that our candidate general w-flag circuit construction requires $w - 1$ flag qubits and is implemented in $7w - 8$ time steps. This is clearly not optimal in general since for example, as shown in Fig. 10a, a w-flag circuit was found (for $w = 6$) which requires only three flag qubits instead of five and the circuit is implemented in 14 time steps instead of 34. It is thus still an open problem to find optimal w-flag circuits for arbitrary w.

D Quantum Reed-Muller codes

In this section we first describe how to construct the family of quantum Reed-Muller codes $\text{QRM}(m)$ with code parameters $[\![2^m - 1, k = 1, d = 3]\!]$ following [37]. We then show that the family of $\text{QRM}(m)$ codes satisfy the sufficient flag 1-FTEC condition of Section 3.2.

Reed-Muller codes of order m ($\text{RM}(1, m)$) are defined recursively from the following generator matrices: First, $\text{RM}(1, 1)$ has generator matrix

$$G_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \tag{10}$$

and $\mathrm{RM}(1, m+1)$ has generator matrix

$$G_{m+1} = \begin{pmatrix} G_m & G_m \\ 0 & 1 \end{pmatrix}, \tag{11}$$

where 0 and 1 are vectors of zeros and ones in Eq. (11). The dual of $\mathrm{RM}(1, m+1)$ is given by the higher order Reed-Muller code $\mathrm{RM}(m-2, m)$. In general, the generator matrices for higher-order Reed-Muller codes $\mathrm{RM}(r, m)$ are given by

$$H_{r,m+1} = \begin{pmatrix} H_{r,m} & H_{r,m} \\ 0 & H_{r-1,m} \end{pmatrix}. \tag{12}$$

with

$$H_{2,1} = H_{1,1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \tag{13}$$

The X stabilizer generators of $\mathrm{QRM}(m)$ are derived from shortened Reed-Muller codes where the first row and column of G_m are deleted. We define the resulting generator matrix as $\overline{G}_m$. The Z stabilizer generators are obtained by deleting the first row and column of $H_{m-2,m}$. Similarly, we define the resulting generator matrix as $\overline{H}_{m-2,m}$.

As was shown in [37], $\mathrm{rows}(\overline{G}_m) \subset \mathrm{rows}(\overline{H}_{m-2,m})$ and each row has weight 2^{m-1}. Therefore, all the X-type stabilizer generators of $\mathrm{QRM}(m)$ have corresponding Z-type stabilizers. By construction, the remaining rows of $\overline{H}_{m-2,m}$ will have weight 2^{m-2}. Furthermore, every weight 2^{m-2} row has support contained within some weight 2^{m-1} row of the generator matrix $\overline{H}_{m-2,m}$. Therefore, every Z-type stabilizer generator has support within the support of an X generator.

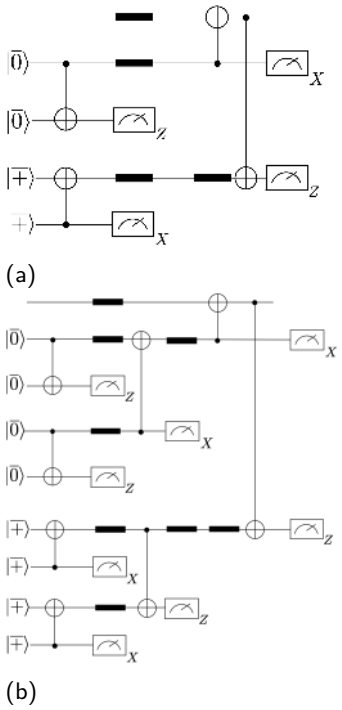

Figure 17: (a) Fault-tolerant Steane error correction circuit for distance-three CSS codes. Each line represents an encoded qubit. The circuit uses only two encoded $|\overline{0}\rangle$ and $|\overline{+}\rangle$ ancilla states (encoded in the same error correcting code which protects the data) to ensure that faults in the preparation circuits of the ancilla's don't spread to the data block. (b) Fault-tolerant Steane error correction circuit which can be used for any distance-three CSS stabilizer code encoding the data. There are a total of eight encoded ancilla qubits instead of four. The dark bold lines represent resting qubits. Note that the circuit in Fig. 17b could in some cases be used for higher distance CSS codes with appropriately chosen circuits for $|\overline{0}\rangle$ and $|\overline{+}\rangle$ ancilla states (see [40]).

E Implementation of Steane error correction

In this section we describe how to implement Steane error correction and discuss its fault-tolerant properties. We also provide a comparison of a version of Steane error correction with flag 2-FTEC protocol described in Section 2.2 applied to the $[\![19, 1, 5]\!]$ code.

Steane error correction is a fault-tolerant scheme that applies to the Calderbank-Shor-Steane (CSS) family of stabilizer codes [5]. In Steane error correction, the idea is to use encoded $|\overline{0}\rangle$ and $|\overline{+}\rangle = (|\overline{0}\rangle + |\overline{1}\rangle)/\sqrt{2}$ ancilla states to perform the syndrome extraction. The ancilla's are encoded in the same error correcting code that is used to protect the data. The X stabilizer generators are measured by preparing the encoded $|\overline{0}\rangle$ state and performing transversal CNOT gates between the ancilla and the data, with the ancilla acting as the control qubits and the data acting as the target qubits. After applying the transversal CNOT gates, the syndrome is obtained by measuring $|\overline{0}\rangle$ transversally in the X-basis. The code construction for CSS codes is what guarantees that the correct syndrome is obtained after applying a transversal measurement (see [32] for more details).

Similarly, the Z-stabilizer generators are measured by preparing the encoded $|\overline{+}\rangle$, applying CNOT gates transversally between the ancilla and the data with the data acting as the control qubits and the ancilla's acting as the target qubits. The syndrome is then obtained by measuring $|\overline{+}\rangle$ transversally in the Z-basis.

The above protocol as stated is not sufficient in order to be fault-tolerant. The reason is that in general the circuits for preparing the encoded $|\overline{0}\rangle$ and $|\overline{+}\rangle$ are not fault-tolerant in the sense that a single error can spread

FTEC scheme	Noise model	Number of qubits	Time steps (T_{time})	Pseudo-threshold
Full Steane-EC	$\tilde{p} = p$	≥ 171	15	$p_{\text{pseudo}} = (3.50 \pm 0.14) \times 10^{-3}$
Full Steane-EC	$\tilde{p} = p/100$	≥ 171	15	$p_{\text{pseudo}} = (1.05 \pm 0.04) \times 10^{-3}$
Flag-EC $[\![19, 1, 5]\!]$	$\tilde{p} = p$	22	$504 \leq T_{\text{time}} \leq 960$	$p_{\text{pseudo}} = (1.14 \pm 0.02) \times 10^{-5}$
Flag-EC $[\![19, 1, 5]\!]$	$\tilde{p} = p/100$	22	$504 \leq T_{\text{time}} \leq 960$	$p_{\text{pseudo}} = (7.74 \pm 0.16) \times 10^{-5}$

Table 5: Pseudo-threshold results for the Full Steane and flag 2-FTEC protocol applied to the $[\![19, 1, 5]\!]$ code. Since the Steane error correction protocol is non-deterministic, the number of qubits will depend on how many times the encoded states are rejected. For low error rates, the states are accepted with high probability so that the average number of qubits is ≈ 171. Our three qubit flag error correction protocol requires at most six rounds of syndrome measurements, with each round using flag circuits requiring 168 time steps and the round using non-flag circuits requiring 120 time steps. However, for low noise rates, the average number of time steps will be close to 504 (since at least three rounds are required for the protocol to be fault-tolerant).

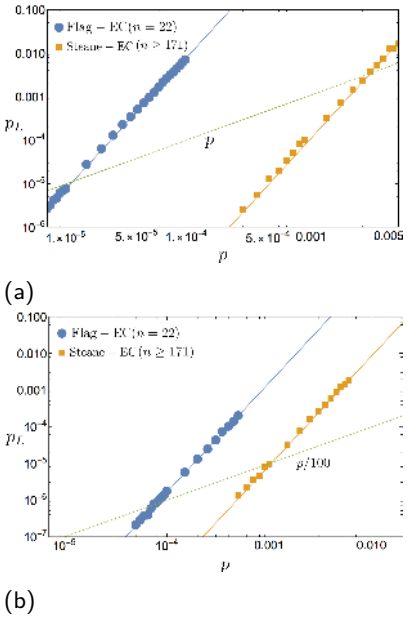

Figure 18: Logical failure rate of the full fault-tolerant Steane error correction approach of Fig. 17b and the flag 2-FTEC protocol of Section 2.2 applied to the $[\![19, 1, 5]\!]$ code. In (a) idle qubits are chosen to fail with a total probability $\tilde{p} = p$ while in (b) idle qubits fail with probability $\tilde{p} = p/100$. The intersection between the dashed curve and solid lines represent the pseudo-threshold of both error correction schemes.

to a multi-weight error which could then spread to the data block when applying the transversal CNOT gates. To make the protocol fault-tolerant, extra $|\overline{0}\rangle$ and $|\overline{+}\rangle$ ancilla states (which we call verifier qubits) are needed to check for multi-weight errors at the output of the ancilla states.

For the $|\overline{0}\rangle$ ancilla, multiple X errors can spread to the data if left unchecked. Therefore, another encoded $|\overline{0}\rangle$ ancilla is prepared and a transversal CNOT gate is applied between the two states with the ancilla acting as the control and the verifier state acting as target. Anytime X errors are detected the state is rejected and the error correction protocol start over. Further, if the verifier qubit measures a -1 eigenvalue of the logical Z operator, the ancilla qubit is also rejected. A similar technique is used for verifying the $|\overline{+}\rangle$ state (see Fig. 17a).

For the $[\![7, 1, 3]\!]$ Steane code, an error $E = Z_i Z_j$ can always be written as $E = \overline{Z} Z_k$ where $\overline{Z}$ is the logical Z operator (this is not true for general CSS codes). But $|\overline{0}\rangle$ is a $+1$ eigenstate of $\overline{Z}$. Therefore, we don't need to worry about Z errors of weight greater than one occurring during the preparation of the $|\overline{0}\rangle$ state.

In [28] it was shown that unlike for the $[\![7, 1, 3]\!]$ code, for general CSS codes, the encoded ancilla states need to be verified for both X and Z errors in order for Steane error correction to satisfy the fault-tolerant properties of Definition 3. We show the general distance-three fault-tolerant scheme in Fig. 17b. Note that the circuit in Fig. 17a will only satisfy the fault-tolerant criteria of Definition 3 for perfect distance-three CSS codes (see [28] for more details).

In Section 4.2 we computed logical failure rates for Steane error correction applied to the $[\![19, 1, 5]\!]$ code using the circuit of figure Fig. 17a in order to minimize the number of physical qubits. However, since the $[\![19, 1, 5]\!]$ code is not a perfect CSS code, only the circuit in Fig. 17b satisfies all the criteria of Definition 3. This explains why the leading order contributions to the logical failure was of the form $p_{\text{L}} = c_1 p^2 + c_2 p^3 + \mathcal{O}(p^4)$ instead of $p_{\text{L}} = cp^3 + \mathcal{O}(p^4)$ (which would be the case for a distance-5 code).

In Fig. 18 we applied Steane error correction using the circuit of Fig. 17b to achieve the full error correcting capabilities of the $[\![19, 1, 5]\!]$ code. We used methods presented in [31, 40] in order to obtain the encoded $|\overline{0}\rangle$ state (since the $[\![19, 1, 5]\!]$ code is self-dual, the $|\overline{+}\rangle$ state is obtain by interchanging all physical $|0\rangle$ and $|+\rangle$ states and reversing the direction of the CNOT gates). Note that not all $|\overline{0}\rangle$ and $|\overline{+}\rangle$ circuits had the same sequence of CNOT gates. This was to ensure that a single fault in two different preparation circuits, i.e. for $|\overline{0}\rangle$ and for

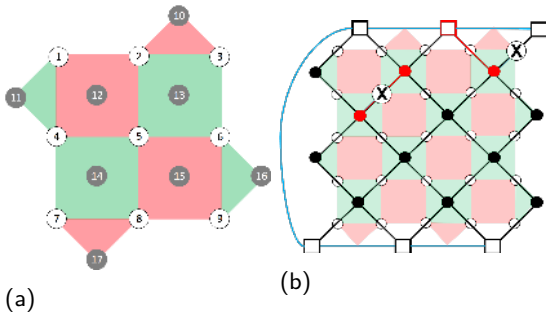

Figure 19: (a) The $d = 3$ surface code, with data qubits represented by white circles. The X (Z) stabilizer generators are measured with measurement ancillas (gray) in red (green) faces (b) For perfectg measurements, the graph G_{2D} used to correct X type errors (here for $d = 5$) consists of a black node for each Z-stabilizer, and a black edge for each data qubit in the surface code. White boundary nodes and blue boundary edges are added. Black and blue edges are given weight one and zero respectively. In this example, a two qubit X error has occurred causing three stabilizers to be violated (red nodes). A boundary node is also highlighted and a minimum weight correction (red edges) which terminates on highlighted nodes is found. The algorithm succeeds as the error plus correction is a stabilizer.

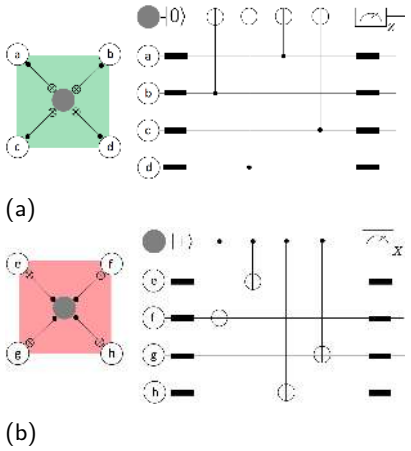

Figure 20: Circuits for measuring (a) Z-type, and (b) X-type generators. Identity gates (black rectangles) are inserted in the Z-type stabilizer measurement circuits to ensure that all measurements are synchronized. Note that unlike in [9], to be consistent with the other schemes in this paper, we assume that we can prepare and measure in both the X and Z basis.

$|\mp\rangle$, would not lead to uncorrectable X or Z errors that would go undetected by the verifier ancillas and at the same time propagate to the data block. The results are compared with the flag 2-FTEC protocol of Section 2.2 applied to the $[\![19, 1, 5]\!]$ for the noise models where idle qubits fail with probability $\tilde{p} = p$ and $\tilde{p} = p/100$. In both cases the logical failure rates have a leading order p^3 contribution (which is determined from finding the best fit curve to the data). The pseudo-threshold results are given in Table 5.

As can be seen, the full Steane-EC protocol using the circuit of Fig. 17b achieves significantly lower logical failure rates compared to Steane-EC using the circuit in Fig. 17a at the cost of using a minimum of 171 qubits compared to a minimum of 95 qubits. In contrast, the flag 2-FTEC scheme of Section 2.2 has a pseudo-threshold that is one to two orders of magnitude lower than than the full Steane-EC scheme but requires only 22 qubits.

F Implementation of Surface code error correction

We consider the rotated surface code [7–9, 33–35] as shown in Fig. 19a, which has $n = d^2$ data qubits for distance d. Although we are concerned with error correction under the circuit level noise model described in Section 1.2, it is useful to build intuition by first considering the idealized noise model in which stabilizer measurements are perfect, and single qubit X errors occur with probability $2p/3$ (Z errors can be treated in the same way). An X type error E occurs with probability $\mathcal{O}(p^{\text{wt}(E)})$, and has syndrome $s(E)$.

The minimum weight X-type correction can be found efficiently for the surface code in terms of the graph G_{2D} shown in Fig. 19b. The graph G_{2D} has a bulk node (black circle) for each Z stabilizer generator, and a bulk edge (black) for each data qubit. A bulk edge coming from a bulk node corresponds to the edge's data qubit being in the support of the node's stabilizer. The graph also contains boundary nodes (white boxes) and boundary edges (blue), which do not correspond to stabilizers or data qubits. Each bulk and boundary edge is assigned weight one and zero respectively. The minimum weight decoder is then implemented as follows. After the error E is applied, the nodes corresponding to unsatisfied stabilizers are highlighted. If an odd number of stabilizers was unsatisfied, one of the boundary nodes is also highlighted. Highlighted nodes are then efficiently paired together by the minimum weight connections in the graph, by Edmonds' algorithm [41, 42]. The correction C is applied to the edges in the connection. Note that any single $\mathcal{O}(p)$ fault in this noise model corresponds to a weight one edge on the graph.

For circuit noise, we introduce a measurement qubit for each stabilizer generator, as represented by gray circles in Fig. 19a, and circuits must be specified to implement the measurements, such as those in Fig. 20. The performance of the code is sensitive to the choice of circuit [34], for example a poor choice could allow a single fault to cause a logical failure for $d = 3$ for any choice

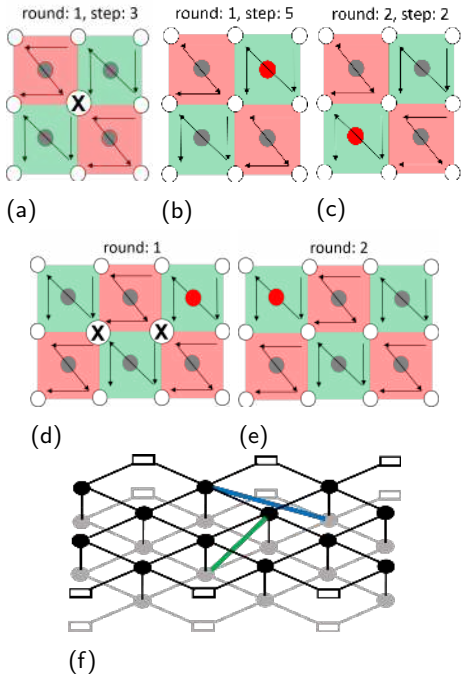

Figure 21: Examples of a single fault leading to diagonal edges in G_{3D}. Dark arrows represent the CNOT sequence. (a) An X error occurs during the third time step in the CNOT gate acting on the central data qubit. (b) During the fifth time step of this round, the X error is detected by the Z type measurement qubit to the top right. (c) The X error is not detected by the bottom left Z type stabilizer until the following round. (d) An XX error occurs on the third CNOT of an X measurement circuit, which is detected by the Z measurement to the right. (e) Detection by the left Z stabilizer does not occur until the next round. (f) The corresponding edges in G_{3D}, green for (a-c), and blue for (d-e). Here we show two rounds of the graph ignoring boundary edges.

of decoder.

To implement the decoder, first construct a new three dimensional graph G_{3D} by stacking d copies of the planar graph G_{2D} that was shown in Fig. 19b, and adding new bulk (boudnary) edges to connect bulk (boudnary) nodes in neighboring layers. We also add additional diagonal edges such that any single $\mathcal{O}(p)$ fault in the measurement circuits corresponds to a weight-one edge in G_{3D} (see Fig. 21). For simplicity, we do not involve further possible optimizations such as setting edge weights based on precise probabilities and including X-Z correlations [14].

All simulations of the surface code are performed using the circuit noise model in Section 1.2, with the graph G_{3D} described above as follows (to correct X errors):

1. Data acquisition: Stabilizer outcomes are stored over d rounds of noisy error correction, followed by one round of perfect error correction. The net error E applied over all d rounds is recorded.

2. Highlight nodes: Nodes in the graph G_{3D} are highlighted if the corresponding Z-type stabilizer outcome changes in two consecutive rounds. [5]

3. Minimum weight matching: Find a minimal edge set forming paths that terminate on highlighted nodes. Highlight the edge set.

4. Vertical collapse: The highlighted edges in G_{3D} are mapped edges in the planar graph G_{2D}, and are then added modulo 2.

5. Correction: The X-type correction C_X is applied to highlighted edges in G_{2D}.

The Z correction C_Z is found analogously. Finally, if the residual Pauli $R = EC_XC_Z$ is a logical operator, we say the protocol succeeded, otherwise we say it failed.

G Compact implementation of flag error correction

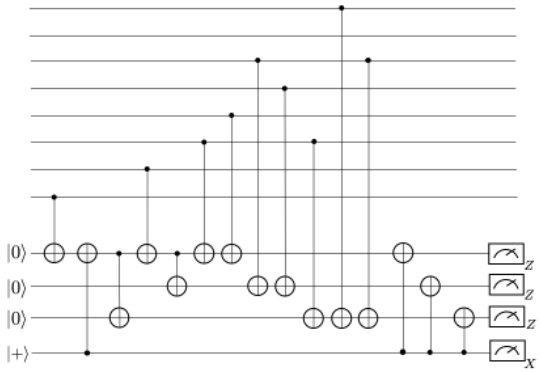

Figure 22: Circuit for measuring the Z stabilizer generators of the $[\![7,1,3]\!]$ code using one flag qubit and three measurement qubits. The circuit is constructed such that any single fault at a bad location leading to an error of weight greater than one will cause the circuit to flag. Moreover, any error that occurs when the circuit flags due to a single fault has a unique syndrome.

In [23], it was shown that by using extra ancilla qubits in the flag-EC protocol, it is possible to measure multiple stabilizer generators during one measurement cycle which could reduce the circuit depth. Note that for the Steane code, measuring the Z stabilizers using Fig. 2b requires only one extra time step. In this section we compare logical failure rates of the $[\![7,1,3]\!]$ code using the flag-EC method of Section 2.1 which requires only

[5] For an odd number of highlighted vertices, highlight the boundary vertex.

FTEC scheme	Noise model	Number of qubits	Time steps (T_{time})	Pseudo-threshold
Flag-EC $[\![7,1,3]\!]$	$\tilde{p}=p$	9	$36 \leq T_{\text{time}} \leq 108$	$p_{\text{pseudo}} = (3.39 \pm 0.10) \times 10^{-5}$
Flag-EC $[\![7,1,3]\!]$		11	$34 \leq T_{\text{time}} \leq 104$	$p_{\text{pseudo}} = (2.97 \pm 0.01) \times 10^{-5}$

Table 6: Pseudo-thresholds and circuit depth for flag-EC protocols using two and four ancilla qubits applied to the $[\![7,1,3]\!]$ code. The results are presented for the noise models where $\tilde{p}=p$ and $\tilde{p}=p/100$.

$[\![5,1,3]\!]$	$[\![7,1,3]\!]$	$[\![19,1,5]\!]$ code	$[\![17,1,5]\!]$ code
$X_1Z_2Z_3X_4$	$Z_4Z_5Z_6Z_7$	$Z_1Z_2Z_3Z_4,\ X_1X_2X_3X_4$	$Z_1Z_2Z_3Z_4,\ X_1X_2X_3X_4$
$X_2Z_3Z_4X_5$	$Z_2Z_3Z_6Z_7$	$Z_1Z_3Z_5Z_7,\ X_1X_3X_5X_7$	$Z_1Z_3Z_5Z_6,\ X_1X_3X_5X_6$
$X_1X_3Z_4Z_5$	$Z_1Z_3Z_5Z_7$	$Z_{12}Z_{13}Z_{14}Z_{15},\ X_{12}X_{13}X_{14}X_{15}$	$Z_5Z_6Z_9Z_{10},\ X_5X_6Z_9Z_{10}$
$Z_1X_2X_4Z_5$	$X_4X_5X_6X_7$	$Z_1Z_2Z_5Z_6Z_8Z_9,\ X_1X_2X_5X_6X_8X_9$	$Z_7Z_8Z_{11}Z_{12},\ X_7X_8X_{11}X_{12}$
	$X_2X_3X_6X_7$	$Z_6Z_9Z_{16}Z_{19},\ X_6X_9X_{16}X_{19}$	$Z_9Z_{10}Z_{13}Z_{14},\ X_9X_{10}X_{13}X_{14}$
	$X_1X_3X_5X_7$	$Z_{16}Z_{17}Z_{18}Z_{19},\ X_{16}X_{17}X_{18}X_{19}$	$Z_{11}Z_{12}Z_{15}Z_{16},\ X_{11}X_{12}X_{15}X_{16}$
		$Z_{10}Z_{11}Z_{12}Z_{15},\ X_{10}X_{11}X_{12}X_{15}$	$Z_8Z_{12}Z_{16}Z_{17},\ X_8X_{12}X_{16}X_{17}$
		$Z_8Z_9Z_{10}Z_{11}Z_{16}Z_{17}$	$Z_3Z_4Z_6Z_7Z_{10}Z_{11}Z_{14}Z_{15}$
		$Z_5Z_7Z_8Z_{11}Z_{12}Z_{13}$	$X_3X_4X_6X_7X_{10}X_{11}X_{14}X_{15}$
		$X_5X_7X_8X_{11}X_{12}X_{13}$	
		$X_8X_9X_{10}X_{11}X_{16}X_{17}$	
$\overline{X}=X^{\otimes 5}, \overline{Z}=Z^{\otimes 5}$	$\overline{X}=X^{\otimes 7}, \overline{Z}=Z^{\otimes 7}$	$\overline{X}=X^{\otimes 19}, \overline{Z}=Z^{\otimes 19}$	$\overline{X}=X^{\otimes 17}, \overline{Z}=Z^{\otimes 17}$

Table 7: Stabilizer generators for the 5-qubit code [43], $d=3$ (Steane code) [44], $d=5$ ($[\![19,1,5]\!]$ code) and $d=5$ ($[\![17,1,5]\!]$ code) family of color codes [36]. The last row illustrates representatives of the codes logical operators.

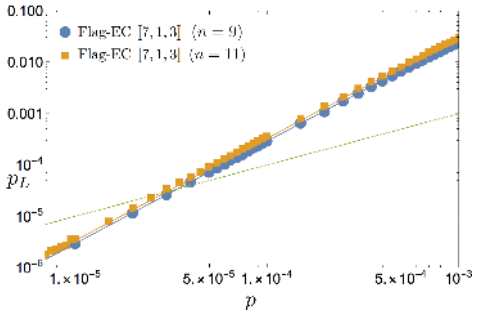

Figure 23: Logical failure rates of the flag 1-FTEC protocols using two and four ancilla qubits applied to the $[\![7,1,3]\!]$ Steane code.

two ancilla qubits and a flag-EC method which uses four ancilla qubits but that can measure all Z stabilizer generators in one cycle (see Fig. 22). All X stabilizers are measured in a separate cycle.

Logical failure rates for $\tilde{p}=p$ are shown in Fig. 23. Pseudo-thresholds and the number of time steps required to implement the protocols are given in Table 6. Note that measuring stabilizers using two ancilla's requires at most two extra time steps. Furthermore, the extra ancilla's for measuring multiple stabilizers result in more idle qubit locations compared to using only two ancilla qubits. With the added locations for errors to be introduced, the flag error correction protocol using only two ancilla's achieves a *higher* pseudo-threshold compared to the protocol using more ancilla's. Thus assuming that reinitializing qubits can be done without introducing many errors into the system, FTEC using fewer qubits could achieve lower logical failure rates compared to certain schemes using more qubits.

H Stabilizer generators of various codes.

In Table 7 we provide stabilizer generators for the $[\![5,1,3]\!]$ code, $[\![7,1,3]\!]$ Steane code, $[\![19,1,5]\!]$ and $[\![17,1,5]\!]$ color codes.