Distance-four quantum codes with combined postselection and error correction

Prithviraj Prabhu and Ben W. Reichardt
Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, California 90089, USA

(Received 26 May 2023; accepted 20 March 2024; published 8 July 2024)

When storing encoded qubits, if single faults can be corrected and double faults can be postselected against, logical errors occur due to at least three faults. At current noise rates, having to restart when two errors are detected prevents very long-term storage, but that should not be an issue for low-depth computations. We consider distance-four efficient encodings of multiple qubits into a modified planar patch of the 16-qubit surface code. We simulate postselected error correction for up to 12 000 rounds of parallel stabilizer measurements and subsequently estimate the cumulative probability of logical error for up to 12 encoded qubits. Our results demonstrate a combination of low logical error rate and low physical overhead. For example, the distance-four surface code, using postselection, accumulates 25 times less error than its distance-five counterpart. For *six* encoded qubits, a distance-four code using 25 qubits protects as well as the distance-five surface code using 246 qubits. Hence, distance-four codes, using postselection in a planar geometry, are qubit-efficient candidates for fault-tolerant, moderate-depth computations.

DOI: 10.1103/PhysRevA.110.012419

I. INTRODUCTION

Error correction and postselection. Noisy intermediate-scale quantum (NISQ) algorithms for eigensolvers [1,2] and machine learning [3] are growing popular as applications for state-of-the-art few-qubit quantum systems. Unfortunately, these devices are still prone to large amounts of noise [4–6]. Although error correction can decrease error rates [7–9], current experiments encode only one logical qubit that is still fairly noisy [10–13]. In this paper we simulate storing multiple logical qubits in a lattice as a first step toward modeling few-qubit computations. We repeatedly correct and remove single-qubit errors. On detecting a more dangerous—and less common—two-qubit error, we reject and restart. This "postselection" technique allows distance-four codes to achieve logical error rates similar to those of distance-five codes. For example, as shown in Table I, a distance-four code can correct errors on six logical qubits with a failure rate similar to that of the distance-five surface code using only 10% as many physical qubits. Table I also shows that acceptance rates are fairly high, so occasional restarts should not be a major issue for low-depth NISQ algorithms [14,15].

Postselection is a versatile tool in the quantum tool kit. In experiments, it has been used to decode the $[\![4,2,2]\!]$ error-detecting code [16,17] and the $[\![4,1,2]\!]$ surface code [11,18]. In theoretical research, it has been used to reduce the logical error probability of state preparation [19,20] and magic-state distillation [21,22]. Recently, postselection was used to improve quantum key distribution [23,24] and learning quantum states [25,26].

Knill previously combined postselection with error correction on concatenated distance-two codes to show an impressive 3% fault-tolerance threshold [27]. We also combine postselection and error correction, but with distance-four codes. As Fig. 1 indicates, distance-two codes can detect single errors, and distance-three codes can correct them, meaning logical errors are due to second-order faults. Distance-three codes may alternatively be used to detect one or two errors, but then they lose the ability to correct, and computations are very short-lived. We choose to use distance-four codes since they can simultaneously correct an error and detect two errors. Correcting some errors ensures restarts are less frequent, so longer computations can be run. Since logical errors are caused only by third-order faults, logical error rates are very low.

Physical layout. In practice, it is difficult to build a quantum computer with native (fast, reliable) two-qubit gates between *every* pair of qubits. Instead, qubits are placed on a one- or two-dimensional lattice, and two-qubit gates are mediated by local interactions, as in superconducting architectures and solid-state systems. Current ion-trap systems use long-range gates [28] and transport mechanisms [10] to connect all the qubits, but some degree of locality is required for larger systems. In light of these connectivity constraints, it may be wiser to choose quantum codes that can be laid out on a lattice such that error correction requires the fewest number of local native gates. The popular surface code has the attractive feature that it requires only nearest-neighbor interactions on a two-dimensional (2D) square lattice [29,30]. Similarly, error correction for topological codes has been investigated on sparser degree-three lattices [31–33]. But there is insufficient research on the performance gains of more densely connected layouts.

We suggest using 16-qubit codes on the 25-qubit rotated square lattice in Fig. 2, where ancilla qubits additionally interact with neighboring ancillas. This allows the use of flag qubits for fault tolerance [34–37], in turn allowing measurement of large stabilizers. As shown in Fig. 3, we choose distance-four codes whose stabilizer generators are fairly local, with short Shor-style stabilizer measurement sequences that do not require any SWAP gates. We consider two block codes and a color code that encode multiple qubits [38] and

TABLE I. Postselected error correction for six logical qubits using [[16, k, 4]] codes on the 25-qubit planar layout in Fig. 2. The probability of logical error, acceptance, and expected time to complete are shown for 300 time steps, with noise rate $p = 5 \times 10^{-4}$. The $k = 6$ code achieves a logical error rate close to the distance-five surface code using only 10% of the qubits. In comparison, for six physical qubits at a memory error rate of $p/10$, the probability of error is about $6 \times 300 \times p/10 = 0.09$. The most favorable numbers for each metric are indicated in bold. Colors for the multi-qubit codes reflect those in the graphs of Fig. 11.

	Code	Qubits	P(Logical error)	Postselection	
				Acceptance	$\mathbb{E}[time]$
	$k = 2$	$25 \times 3 =$ **75**	0.032	11.5%	1067
	$k = 4$	**50**	0.029	20.5%	725
	$k = 6$	**25**	**0.017**	35.5%	530
surface	$d = 4$	150	**0.009**	3.5%	2820
surface	$d = 3$	78	0.143	100%	300
surface	$d = 5$	246	**0.015**	100%	300

the rotated surface code [39] as a benchmark for postselection. In contrast, before the advent of topological codes, block codes were used for the simulation of 2D local error correction [40–42]. These proposals performed Steane error correction on small distance-two and -three codes and required many swaps.

Results. We compare our 16-qubit codes with the 25-qubit, distance-five surface code. We show below in Fig. 10 that, with rejection, the normalized logical error rate of the proposed codes is less than that of the distance-five surface code by as much as 1 order of magnitude. The distance-four surface code actually achieves a separation of 2 orders of magnitude.

However, the logical error rate per time step does not capture the drawback of restarts. Instead, a better metric is the cumulative probability of logical error. Figure 4 compares this metric between the different codes for short computations that do not restart too often (more information is given in Fig. 11 below). For one logical qubit, the distance-four surface code vastly outperforms its distance-five counterpart, and the $k = 2$ and $k = 4$ codes achieve a good balance of low qubit overhead and a low logical error rate. We also show that just 50–75 physical qubits are sufficient for good protection of 12 logical qubits. Overall, we obtain lower logical error rates with higher encoding rates using postselection and multiqubit codes.

In Appendix B, we compare the storage error rate of unencoded qubits with the encodings in Fig. 3. As expected, at error rates up to 10^{-3}, fault-tolerant error correction is more robust than leaving qubits idle.

Distance	Error weight 1	Error weight 2	Error weight 3
2	Detect, restart	Logical error	
3	Correctable	Logical error	
4	Correctable	Detect, restart	Logical error
5	Correctable	Correctable	Logical error

FIG. 1. Distance-four codes with postselection lead to $O(p^3)$ logical errors, much like distance-five codes. Even-distance codes require restarts, however, unlike odd-distance codes.

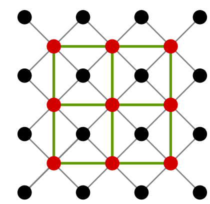


FIG. 2. Planar layout of 16 data and 9 ancilla qubits, shown in black and red respectively. CNOT gates are allowed along the edges. Gray edges are required for the surface code, and green edges between ancillas are required for the codes in this paper.

Future work. In order to verify these results on current quantum systems, some work is required. Dense qubit connectivity in ion-trap systems may allow for simple measurement of high-weight stabilizers, but superconducting devices generally prefer a low qubit degree due to high cross-talk errors. It may be possible to modify the circuits in this work to allow a maximum qubit degree of at most five or six, such as in the IBM Tokyo device [43]. Consequently, in Fig. 12 below, we show that error correction of the $k = 2$ code is possible with degree-four connectivity but requires many extra qubits.

We show only how to do fault-tolerant error correction, but the ultimate goal is to perform quantum computation. Selective logical measurements could induce computation within a patch, and transversal gates between vertically stacked code patches could facilitate non-Clifford gates. If these operations introduce a low amount of error, it may be possible to execute relatively high-depth circuits. These tools can then be used to execute short NISQ and magic-state distillation algorithms. As an example, our results show that just 50 physical qubits may be sufficient to demonstrate 10 − to − 2 Meier-Eastin-Knill (MEK) distillation experimentally with $O(p^3)$ logical errors [22].

Organization. In Sec. II we provide more details about distance-four codes and the examples we choose in this paper. Section III details the methods used for fault tolerance. In particular, stabilizer measurement circuits are dealt with in Sec. III A, and sequences of stabilizer measurements are handled in Sec. III B. The noise model and results of simulations are contained in Sec. IV. Section V concludes with a discussion of future work and open questions.

II. CODES

We compare the error-correction performance of six $[[n, k, d]]$ stabilizer quantum codes, where n is the number of physical data qubits and k is the number of logical qubits. A distance-d quantum code should correct all errors of weight $j \leqslant t = \lfloor \frac{d-1}{2} \rfloor$, occurring at rate $O(p^j)$, for error rate p. At low error rates, an unlikely error of weight $(d - j)$ may be misidentified as the more likely weight-j error, inducing a logical flip on recovery. In even-distance codes, errors of weight $d/2$ can be detected, but applying a correction may induce a logical flip. In this paper, we stop the computation instead of attempting to correct, ensuring logical flips occur only at rate $O(p^{t+2})$ and not at $O(p^{t+1})$ as before.

For the distance-four codes shown in Fig. 3, we show that the logical error rate scales as $O(p^3)$ like a distance-five code.

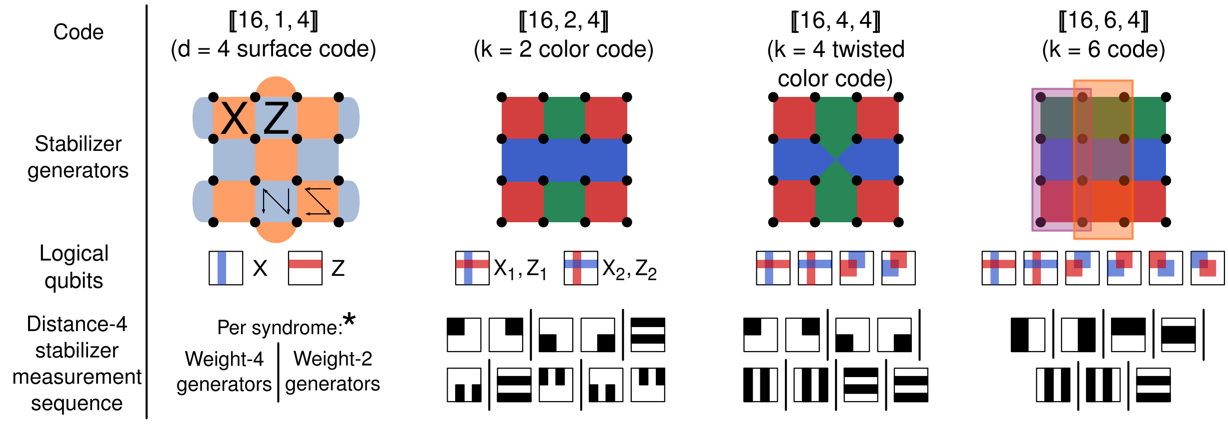


FIG. 3. Codes considered in this paper, with associated distance-four fault-tolerant Z or X stabilizer measurement sequences. (The last three codes are self-dual CSS.) Time steps of parallel measurements are separated by a vertical line (|). Corresponding to the asterisk (∗), for the surface code, fault-tolerant X and Z error correction is carried out using a rolling window of four syndromes, each measured in two time steps.

As a benchmark we first consider the rotated distance-four surface code [44] in the layout in Fig. 2. For a fair comparison of both the resource requirements and logical error rate, we consider additional benchmarks: the distance-three and -five surface codes. As in Ref. [44], each distance-d surface code uses d^2 data qubits and $(d-1)^2$ ancilla qubits.

The next three codes are the central focus of this work. These self-dual Calderbank-Shor-Steane (CSS) codes were first considered in Ref. [38] to show examples of codes that can be constructed to have single-shot sequences of stabilizer measurements. By fixing some of the logical operators, the $k=6$ code can be transformed into the $k=4$ and $k=2$ codes. Alternatively, puncturing the $k=6$ code yields the well-known ⟦15, 7, 3⟧ Hamming code.

Improvements. Although these codes encode more logical qubits, they suffer from the difficult task of having to measure weight-eight stabilizers. It is possible to construct a ⟦16, 2, 4⟧ subsystem code with only weight-four stabilizers and gauge operators. Using the layout of Fig. 2, we compared this code with the $k=2$ subspace code in this paper but found no significant improvements. This code is still useful, however, as we show in Sec. V.

Many other codes can also be constructed with 16 qubits. For a biased-noise system, a CSS code with two logical qubits can be constructed with Z distance six and X distance four. For more logical qubits, a non-CSS ⟦16, 7, 4⟧ code can be used [45]. Although its stabilizer generators are larger, flag-based measurement may still offer a low-overhead route to fault tolerance.

III. FAULT-TOLERANT ERROR CORRECTION

A stabilizer measurement *circuit* is made fault tolerant to quantum errors by using extra physical qubits. These ancillas are used to catch faults that may spread to high-weight errors. In contrast, the bad faults in a syndrome extraction *sequence* flip syndrome bits. Additional stabilizers are measured, essentially encoding the syndrome into a classical code.

A circuit is fault tolerant to distance d if $j \leqslant t = \lfloor \frac{d-1}{2} \rfloor$ midcircuit faults cause an output error of weight of at most j. Additionally, for even-distance fault tolerance, sets of $d/2$ faults spreading to weight $>d/2$ errors should be detected, so the computation can be restarted. When these faults yield an error of weight $d/2$, the computation is restarted if the faults can be detected; otherwise, it is rejected in the next round of error correction.

A. Stabilizer measurement circuits

Quantum error correction involves the measurement of a set of operators called stabilizers to diagnose the location of errors. For fault-tolerant error correction, these stabilizers may be measured individually, as in Shor's scheme [46], or together, using Steane- or Knill-type syndrome extraction [47,48].

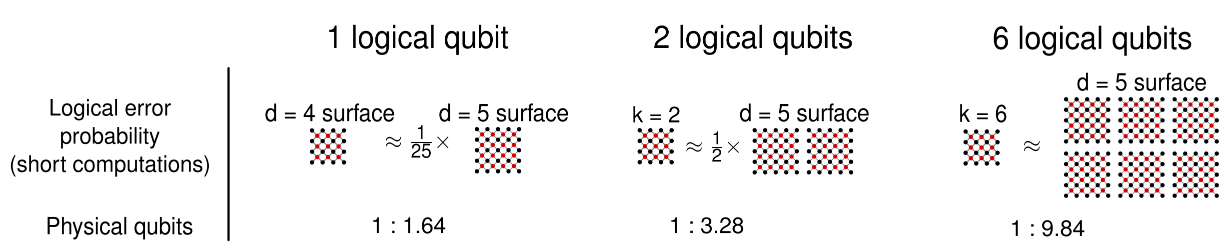
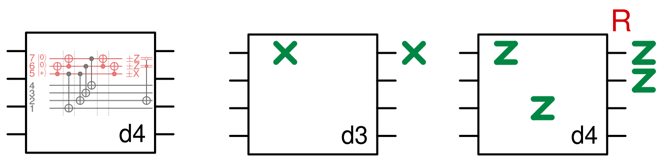
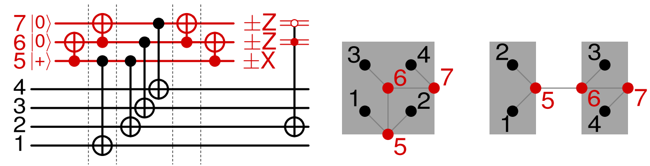


FIG. 4. Summary of the results. For short computations, the probability of a logical error in the distance-four rejection-based surface code is approximately 25 times lower than that of the distance-five variant. Further, for six logical qubits, the $k=6$ code on one patch of 25 qubits can match six patches of the distance-five surface code.



(a) (b)

FIG. 5. (a) A distance-four stabilizer measurement circuit contains ancilla preparation, CNOT gates, measurement, and a recovery. (b) Rules for fault tolerance. One fault should be corrected to an error of X/Z weight of at most 1—this is sufficient for distance three. Two faults should either be rejected (R) or result in an error of weight 2.



(a) (b)

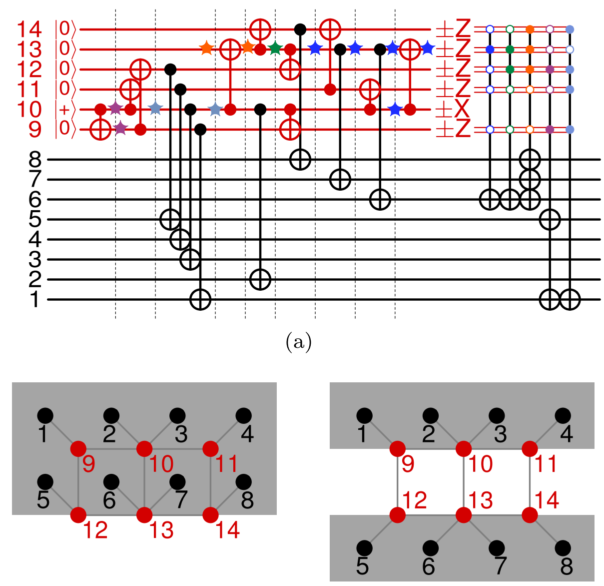
FIG. 6. (a) Circuit to measure a weight-four X stabilizer fault tolerantly to distance four, satisfying the locality constraints in (b). The $\pm Z$ measurements are used to flag midcircuit faults. Gates bunched together can be performed in parallel. (b) Two layouts for measuring stabilizers in the sequences of Fig. 3.

The flag method is a popular spin-off of Shor's scheme [34–37]. By connecting multiple data qubits to each flag qubit, large stabilizers can be measured with relatively low overhead. In addition, flag circuits can be made fault tolerant only up to a desired degree. For example, Shor-style measurement of a weight-w stabilizer needs $w+1$ ancillas and is fault tolerant to distance w, but we show a weight-eight stabilizer measurement circuit with six ancillas that is fault tolerant to distance four.

For distance-three fault tolerance, we show in Fig. 5 that one fault in the circuit should result in an error of X and Z weight of at most 1. For distance four, if two faults occur and can be detected, the computation must be rejected and restarted. If this detection is not possible, the circuit must be designed to ensure errors cannot spread to weight greater than 2. Note that a fault may alter the value of the measured syndrome bit; syndrome bit errors are dealt with in Sec. III B.

We develop flag-based stabilizer measurement circuits. For this, we use a randomized search algorithm constrained by the above fault-tolerance rules and the geometric locality of Fig. 2. We used a stabilizer circuit simulator [49] to determine the flag configurations caused by different faults. A circuit is described as valid if, for every flag configuration, corrections and rejections can be applied while satisfying all the fault-tolerance conditions. With all six codes in this work, the stabilizers that are measured are of weights two, four, and eight. At the circuit level, the measurement of a weight-two stabilizer is automatically fault tolerant (one fault causes an error of weight of at most 1). A weight-four stabilizer measured fault tolerantly to distance three (i.e., one fault results in an error of weight of at most 1) is automatically fault tolerant to distance four, as two faults occurring in the circuit cannot create data errors with X and Z weight greater than 2. In Fig. 6(a), the weight-four stabilizer measurement circuit applies a correction only for the 01 ancilla measurement. Figure 7(a) shows a circuit to measure weight-eight stabilizers fault tolerantly to distance four. This circuit uses different patterns of flag-qubit measurements to either correct an error (for an $O(p)$ fault event), or reject (upon detecting an $O(p^2)$ fault event). The flag patterns associated with corrections and with rejection are tabulated in Appendix A. Figures 6(b) and 7(b) show different ways of arranging the qubits to measure weight-four and weight-eight operators.

Improvements. The benefit of measuring stabilizers individually is that error decoding is relatively simple. When stabilizers with overlapping support are measured in parallel, as in the surface code, more complicated decoding algorithms like minimum-weight perfect matching are required. However, we can still make small improvements for additional parallelism. In the $k=2$ and $k=4$ codes, only two of the corner weight-four stabilizers can be measured simultaneously, as each stabilizer requires three ancilla qubits. We conjecture that by sharing one ancilla qubit among all the corner stabilizers, it may be possible to fault-tolerantly measure all four of them using just nine ancilla qubits, as in Ref. [50]. Alternatively, in Steane-style syndrome extraction, subsets of stabilizers are measured in parallel using n-qubit resource states. Nine ancilla qubits are not sufficient for Steane's method, but Ref. [51] showed that any subset of stabilizers can be jointly measured with specific resource states. If the fault-tolerant preparation of those resource states is possible on the nine-qubit ancilla sublattice in Fig. 2, it will be possible



(b)

FIG. 7. (a) Circuit to measure a weight-eight X stabilizer fault tolerantly to distance four, satisfying the locality constraints in (b). One fault is corrected to at most a weight-one error, but two or more faults may be either corrected or detected, resulting in rejection. We show the flag-based corrections in different colors. The stars represent locations where an X fault results in a correction of that color. Double faults causing rejection are detailed in Appendix A. (b) Two layouts for measuring stabilizers in the sequences of Fig. 3.

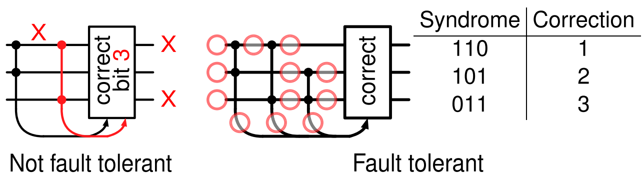


FIG. 8. Fault-tolerant error correction with the three-bit repetition code {000, 111} (adapted from Fig. 1 of Ref. [38]). It is not fault tolerant to correct errors based on the two parity measurements $1 \oplus 2$ and $1 \oplus 3$. An internal fault on bit 1 can be mistaken for an input error on bit 3, as they yield the same syndrome. Errors can be corrected fault tolerantly by adding another parity check, $2 \oplus 3$. Now for up to one fault at any of the circled locations, an input error is corrected, and an internal fault leaves an output error of weight 0 or 1.

to develop faster and more efficient stabilizer measurement circuits.

The circuit of Fig. 7(a) uses six ancilla qubits for distance-four fault tolerance. We found a distance-three fault-tolerant circuit using only four ancillas [qubits 9, 10, 11, and 13 in Fig. 7(b)], which also requires fewer rounds of parallel gates. In the layout in Fig. 2, this may free up enough ancillas to measure two weight-eight stabilizers in each time step. The result is that more stabilizers can be measured faster and data qubits in an error-correction block experience less idle noise. Since these circuits are fault tolerant only to distance three, a future avenue of research could use techniques in Ref. [35] to look at their performance in adaptive distance-four error correction.

B. Stabilizer measurement sequences

Introduction. The correction of errors in a quantum code requires a syndrome built from the measurement results of a sequence of stabilizers. Since syndrome extraction is noisy, it is generally not sufficient to measure just a set of stabilizer generators, as shown in Fig. 8. Even one erroneous collected syndrome bit can result in an incorrect recovery, pushing the code into a state of logical error. Instead, more stabilizers are redundantly measured to protect from quantum faults that cause syndrome bit flips. The distance-d surface code does this by measuring the stabilizer generators sequentially d times in a syndrome repetition code. In our case, for the distance-three surface code, we use three rounds of syndromes consisting of two new ones and one old one from the previous round of decoding. We may port this technique to the $k = 2$, 4, and 6 codes, but recent research has shown that these codes have very small stabilizer measurement sequences [38].

The depth of a quantum circuit is generally calculated as the number of rounds of parallel two-qubit gates since single-qubit gates are trivially short. However, in current systems, the time needed for measurement dominates over the length of a CNOT [10–12,52]. Hence, the focus shifts from minimizing CNOT depth to reducing the rounds of measurements needed for error correction. We therefore denote by "time step" the time needed to measure a set of stabilizers in parallel, as shown in Fig. 9(a). In addition to finding short fault-tolerant sequences of stabilizers, we carefully parallelize their measurement circuits to further speed up error correction.

Fault tolerance rules. We follow the "exRec" formalism of Ref. [53] to determine rules for fault-tolerant error correction, as shown in Fig. 9(b). For distance-three fault tolerance, only two rules are needed. If the input to an error-correction block has a weight-one error and there are no internal faults, the syndrome must be sufficient to correct back to the code space. This is actually the basic rule for an ideal error-correction block. If there are no input errors and one internal fault occurs, the weight of the output error after recovery should be at most 1.

For distance-four fault tolerance, we must consider the effect of up to two input errors or internal faults. If the input error has weight two and there are no internal faults, then the stabilizer measurement sequence must detect the error and restart the computation. If there is a weight-one input error and an internal fault, either the computation is restarted, or the output of the error-correction block must have error of X and Z weight of at most 1. Finally, if two internal faults occur with no input error, either the computation is restarted, or the output error must have a weight of at most 2.

(In the last four rules, the resulting syndrome must never be equivalent to a weight-one input error on a different qubit. This ensures that every weight-one input error can be reliably corrected.)

Solutions. To perform distance-four fault-tolerant error correction on the layout of Fig. 2, we consider measuring stabilizers only of the form given in Sec. III A. The goal is then to devise short parallel stabilizer measurement sequences occupying the fewest time steps while satisfying the fault-tolerance rules above. For each of the proposed codes, the sequences in

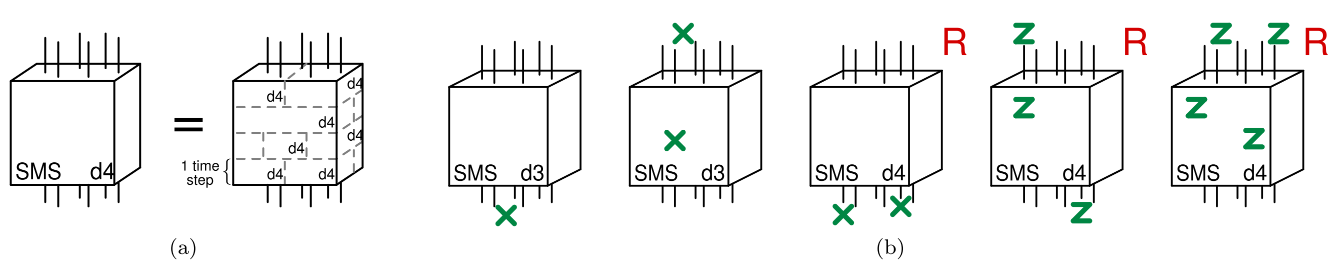


FIG. 9. (a) A stabilizer measurement sequence (SMS) consists of multiple time steps of parallel stabilizer measurement circuits, where the end of a time step denotes the simultaneous measurement of all the ancilla qubits. (b) Rules for distance-four fault tolerance—the first two are sufficient for distance three: (1) An input one-qubit error must be corrected. (2) One internal fault must be corrected to an error of weight of at most 1. (3) A two-qubit input error is rejected. (4) One input error and one internal fault should be corrected to an error of weight of at most 1 or rejected. (5) Two internal faults must be rejected or propagate to an error of weight of at most 2.

Fig. 3 were found using the techniques of Ref. [54]. First, we tested long, randomly generated sequences. By observing patterns in the valid sequences and correlations between stabilizers, we improved the sequences to the concise ones in Fig. 3.

The $k = 2$ code measures 10 X (or Z) stabilizers over five time steps; hence, recovery occurs every 10 time steps. On the other hand, the $k = 6$ code contains no stabilizers of weight less than 8, so parallelism is difficult. Here, seven X (or Z) stabilizers are measured over seven time steps for a total of 14 time steps between recoveries. The distance-four surface code measures all its stabilizer generators in two time steps. The first is used to measure the nine weight-four stabilizer generators using the nine ancilla qubits, and the second time step is used to measure the boundary weight-two stabilizers. Recovery occurs after two fresh syndrome layers are measured, at a frequency of four time steps.

Rejection decoding with the surface code. Many algorithms exist to decode errors on the surface code. With the hope of testing larger surface codes with rejection decoders, we implemented a version of the union-find decoding algorithm [55]. To apply postselection rules for distance-four fault tolerance, we start at the end of union-find, with a set of edges representing X or Z faults. If there are three or more edges, we immediately reject and restart. If there are two or fewer edges, some choices can be made. First, if the two faults are well separated in time, the older fault can be reliably corrected. We experimented with different separation lengths and settled on a configuration that rejects the least often but has the highest $O(p^3)$ logical errors. Second, measurement faults do not contribute to data errors. They are exempted from postselection by projecting edges onto the 2D plane. Evidently, there is a lot of scope for further improvements to the postselection rules. Research also needs to be done on how to generalize these rules to higher-distance surface codes.

IV. RESULTS

Noise model. For simulation, we consider independent circuit-level noise as follows:

(1) With probability p, the preparation of $|0\rangle$ is replaced by $|1\rangle$ and vice versa, and similarly, $|\rangle+$ is replaced by $|-\rangle$.

(2) With probability p, a $\pm X$ or $\pm Z$ measurement on any qubit has its outcome flipped.

(3) With probability p, a one-qubit gate is followed by a random Pauli error drawn uniformly from $\{X, Y, Z\}$.

(4) With probability p, the two-qubit CNOT gate is followed by a random two-qubit Pauli error drawn uniformly from $\{I, X, Y, Z\}^{\otimes 2} \setminus \{I \otimes I\}$.

(5) After each time step, with probability $p(1 + m/10)$, each data qubit is acted upon by a random one-qubit Pauli error drawn uniformly from $\{X, Y, Z\}$. (A time step denotes one round of parallel stabilizer measurements of maximum CNOT depth m, as in Sec. III B.)

The rest error rate models the observed performance of present-day quantum systems, where the time taken to measure an ancilla qubit is long compared to the CNOT gate time. We model the rest error rate during measurement as p and during CNOT gates as $p/10$. Even with dynamical decoupling [56], the error incurred by the idle data qubits can be quite high.

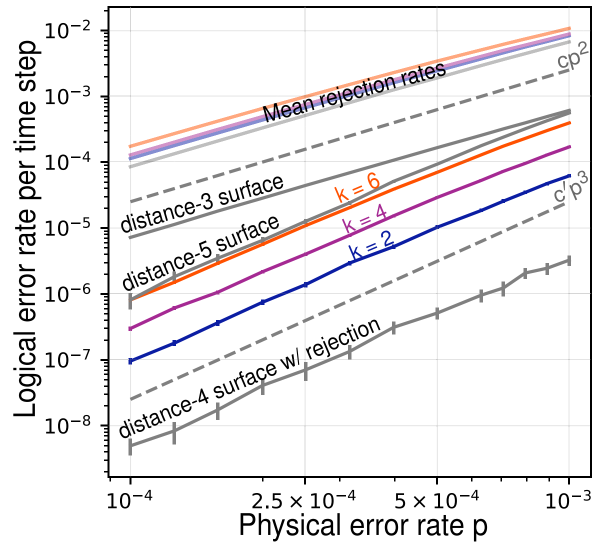


FIG. 10. $O(p^3)$ scaling of the X logical error rate and $O(p^2)$ scaling of the rejection rate, with error bars, for the distance-four codes. The distance-three and distance-five surface codes are shown for comparison. Our codes have a logical error rate per time step as low as 1/10 the distance-five surface code. The distance-four surface code is as low as 1/100.

Normalized logical error rate. The logical error rate of fault-tolerant storage can be estimated by checking for a logical error after each block of error correction. However, different codes correct errors at different frequencies—once every 4 time steps for the surface code but 14 for the $k = 6$ code. To compare the codes on a similar timescale, we normalize the logical error rates with respect to the time step.

We plot the logical error rate per time step in Fig. 10, where we show that a distance-four surface code has a storage error rate of $O(10^{-9})$ for a CNOT gate error rate of just 10^{-4}. Even with the infidelity of present-day CNOT gates, $\sim 10^{-3}$, we show logical error rates approaching 10^{-6}. These results demonstrate the benefits of postselection.

Cumulative logical error probability. The mean rejection rates in Fig. 10 provide a good comparison of how often the different codes reject but do not accurately describe behavior for bounded-length computation. Here, a more useful metric is the probability of acceptance $P_a(t)$, which is how often a t-time-step computation completes. This quantity can be estimated empirically by simulating the application of noisy error correction to an initial state for bounded time, which we denote as a simulation "run." If R is the total number of executed runs and $R_a(t)$ is the number of runs that have not been rejected until time step t,

$$P_a(t) = \frac{R_a(t)}{R}. \tag{1}$$

Similar to the rejection rate, the logical error rate per time step is indicative of the frequency of logical errors but does not help us to understand the drawbacks of postselection. We again refer to a cumulative metric, the probability of a logical error after t time steps of error correction, empirically given by

$$P_L(t) = \frac{R_L(t)}{R}, \tag{2}$$

where $R_L(t)$ is the number of runs in a state of logical error at time step t. For even-distance codes, one must instead look at the probability of logical error conditioned on acceptance, which is calculated as

$$P_{L|a}(t) = \frac{P_L(t)}{P_a(t)} = \frac{R_L(t)}{R_a(t)} = \frac{R_L(t)}{R\,P_a(t)}. \tag{3}$$

For even-distance codes with postselection, $P_a(t) < 1$, and so $P_{L|a}(t) > P_L(t)$. For odd-distance codes where we do not perform rejection and instead only apply corrections, $P_{L|a}(t) = P_L(t)$.

The above formula holds only for a single code patch. The probability of logical error while using multiple patches can be upper bounded from the data for a single patch as

$$P_{L|a}(t, c) \leqslant \frac{c\,R_L(t)}{R\,P_a^c(t)}, \tag{4}$$

where c is the number of code patches used. Note that the number of logically incorrect runs grows linearly with the number of patches, but the probability of acceptance of multiple patches is the probability that every patch has been accepted.

Discussion. We simulated fault-tolerant error correction of the codes in Fig. 3 for up to 12 000 time steps at error rate $p \in \{0.001, 0.0005, 0.00025, 0.0001\}$. Using the empirical formulas above, we plot in Fig. 11 the probability of X logical error conditioned on acceptance and the probability of acceptance for 1, 2, 6, or 12 logical qubits. Note that some plots look discontinuous. This is because an error-correction block spans multiple time steps, but logical errors and rejection syndromes are only checked for at the end of every error-correction block. Above the graphs of Fig. 11, we compare the number of physical qubits required for each code.

There is much to learn from Fig. 11. To start, the first column of graphs shows how a single patch of each code fares against the others for different error rates. The $d = 4$ surface code with rejection boasts the lowest logical error probability overall and has the highest acceptance rates among all the even-distance codes. The logical error probability of the $k = 2$ code actually matches the distance-five surface code, even though it encodes twice as much information. This is also apparent from Table II, where we show the probability of acceptance and logical error for one logical qubit at $p = 10^{-3}$.

For two logical qubits (second column of graphs), the surface codes need two patches of qubits; hence, the probability of logical error doubles, and the acceptance is squared. The distance-four surface code now has the lowest acceptance probability among the distance-four codes. We keep the range of time steps consistent between the first and second columns to show that the curves for the multiqubit codes are unchanged. As shown in Table III, for two logical qubits, the $k = 2$ code halves the logical error probability of the $d = 5$ surface code, using fewer than one third as many physical qubits.

Going further, we analyzed error correction for 6 and 12 encoded qubits, as shown in the last two columns. With only one tenth of the physical overhead, a single $k = 6$ code patch rivals the performance of six patches of the $d = 5$ surface code. The single patch of the $k = 6$ code even outperforms the $k = 2$ and $k = 4$ codes, but this is precisely because only one code patch is used. When multiple code patches are used for many logical qubits, the acceptance rate of the distance-four codes drops exponentially. This is also observed in Tables IV and V, as the acceptance probability of the distance-four surface code quickly approaches zero.

In the last column of graphs, we compare statistics for 12 logical qubits. Although current NISQ systems protect only one logical qubit, our results show that just 50–75 physical qubits are sufficient for 12 logical qubits. In this regime, the $k = 4$ code achieves lower logical error probability than the $k = 6$ code with only 50% more overhead. Unfortunately, at longer timescales, postselection sharply increases the logical error probability, rendering the distance-four codes much less useful.

All simulations in this paper, developed in PYTHON, were executed on the University of Southern California Center for Advanced Research Computing (CARC) high-performance

TABLE II. Error correction for one logical qubit at $p = 0.001$. The probabilities of logical error and acceptance are shown for 80 and 200 time steps. Each code uses one patch of qubits. The distance-four surface code has the lowest logical error probability for short computations. Colors for the multi-qubit codes reflect those in the graphs of Fig. 11.

			$t = 80$		$t = 200$	
	Code	Qubits	$P_{L\|a}$	P_a	$P_{L\|a}$	P_a
	$k = 2$	② 25	② 0.0038	53.9%	② 0.0099	19.6%
	$k = 4$	25	0.0114	50.5%	0.0283	17.2%
	$k = 6$	25	0.0275	44%	0.0687	11.1%
surface	$d = 4$	25	① 0.0001	59.5%	① 0.0004	26.8%
	$d = 3$	① 13	0.0239	100%	0.0587	100%
	$d = 5$	③ 41	③ 0.005	100%	③ 0.0121	100%

TABLE III. Error correction for two logical qubits at $p = 0.0005$. The probabilities of logical error and acceptance are shown for 300 and 750 time steps. The surface codes require more than one patch of physical qubits. Among our codes, the $k = 2$ color code has few large stabilizers and a fast sequence. These advantages help it achieve the lowest logical error probability at the highest acceptance rates. Colors for the multi-qubit codes reflect those in the graphs of Fig. 11.

			$t = 300$		$t = 750$	
	Code	Qubits	$P_{L\|a}$	P_a	$P_{L\|a}$	P_a
	$k = 2$	① 25	② 0.0025	48.5%	② 0.006	16%
	$k = 4$	25	0.0066	45.4%	0.0178	13.1%
	$k = 6$	25	0.0172	35.7%	0.0441	7.6%
surface	$d = 4$	③ 50	① 0.0003	31.9%	① 0.0021	5.6%
	$d = 3$	② 26	0.0477	100%	0.1145	100%
	$d = 5$	82	③ 0.005	100%	③ 0.0124	100%

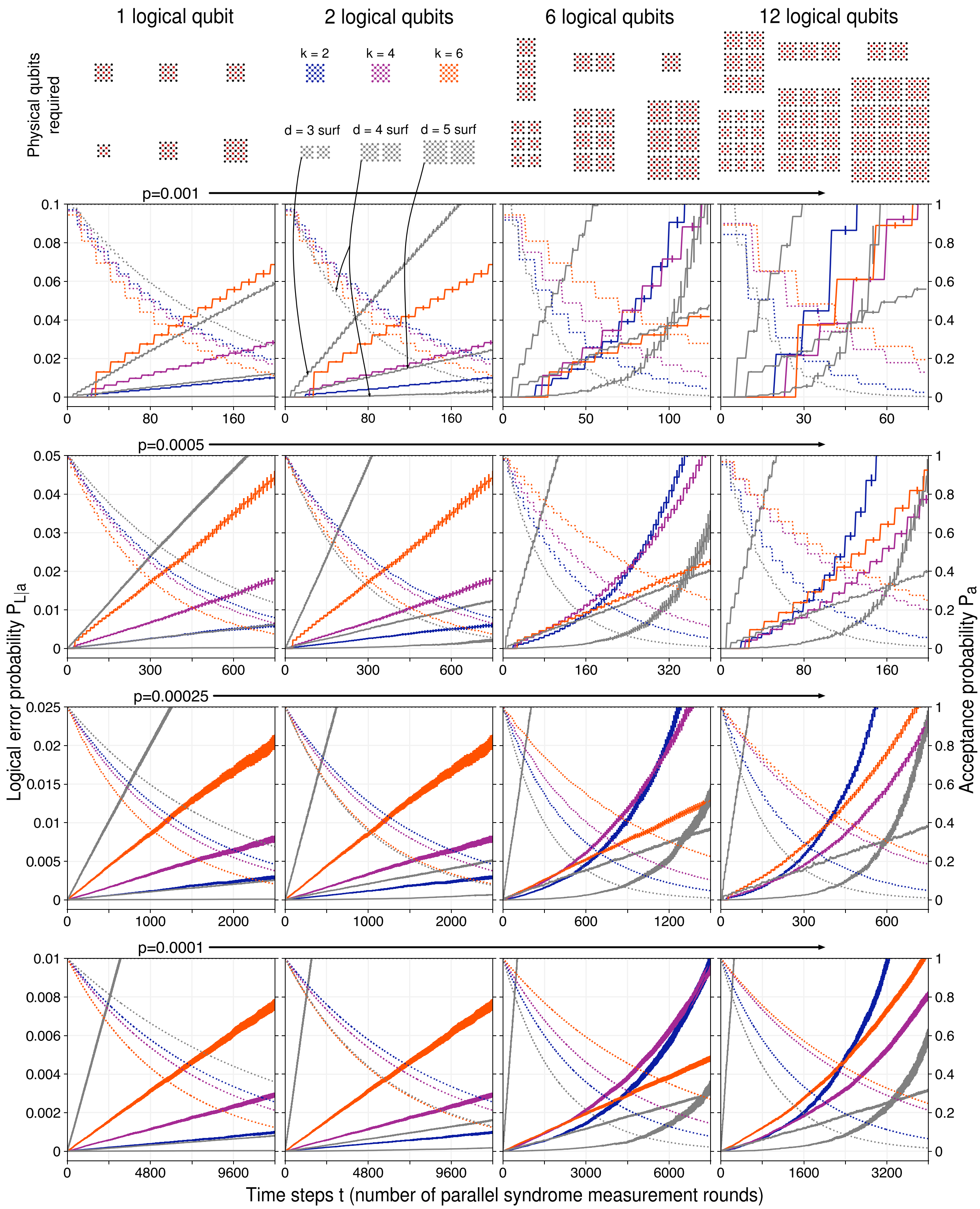


FIG. 11. Probability of X logical error (solid lines) and acceptance (dotted lines) for t time steps of error correction on six codes as a function of physical error rate (row) and desired logical qubits (column). The three colored curves correspond to the $k = 2$ (blue), $k = 4$ (purple), and $k = 6$ (orange) codes, and the three gray curves are the surface codes. The graphs (especially for few time steps) look like a step function because the code patches are checked for logical errors only after blocks of error correction, not time steps. The top row compares the number of physical qubits required to achieve the desired number of logical qubits.

TABLE IV. Error correction for six logical qubits at $p = 0.00025$. The probabilities of logical error and acceptance are shown for 700 and 1500 time steps. The $k = 6$ code requires one tenth the physical qubits as the distance-five surface code while nearly matching the logical error probability. Colors for the multi-qubit codes reflect those in the graphs of Fig. 11.

			$t = 700$		$t = 1500$	
	Code	Qubits	$P_{L\|a}$	P_a	$P_{L\|a}$	P_a
	$k = 2$	③ 75	0.0061	24.5%	0.0412	4.8%
	$k = 4$	② 50	0.0078	34.7%	0.0306	10.4%
	$k = 6$	① 25	③ 0.0060	50.1%	② 0.0129	22.3%
surface	$d = 4$	150	① 0.0009	11.5%	③ 0.0137	1%
surface	$d = 3$	78	0.0847	100%	0.1796	100%
surface	$d = 5$	246	② 0.0044	100%	① 0.0092	100%

computing cluster. The simulations used over 1 million minutes of CPU core time on Intel Xeon processors operating at 2.4 GHz.

Take-home message. Postselection can play a crucial role in reducing logical error rates. However, when logical information is stored for too long, it is likely to be wiped and reset. This is okay for some algorithms: applications with low depth, like variational algorithms [14,15], or those that are designed with rejection, like magic-state distillation [21]. If only one or two qubits are required, the distance-four surface code and the $k = 2$ code offer a very low probability of logical error. For more qubits, we advise using the $k = 4$ or $k = 6$ codes, as they use far fewer physical resources to achieve competitively low logical error. We show that 50–75 good physical qubits are sufficient to correct errors on 12 logical qubits. Even at a CNOT error rate as high as 5×10^{-4}, error correction up to 100 time steps can be run with error probability as low as 1%.

V. FUTURE WORK

In this paper, we showed how to perform fault-tolerant storage with 16-qubit codes. There are two immediate roadblocks to universal fault-tolerant quantum computation. Currently, no devices exist with the layout in Fig. 2, so until they are fabricated, we turn to other layout improvements. The middle

TABLE V. Error correction for 12 logical qubits at $p = 0.0001$. The probabilities of logical error and acceptance are shown for 1800 and 4500 time steps. The $k = 4$ code is well balanced, achieving competitive logical error rates with low qubit overhead. Colors for the multi-qubit codes reflect those in the graphs of Fig. 11.

			$t = 1800$		$t = 4500$	
	Code	Qubits	$P_{L\|a}$	P_a	$P_{L\|a}$	P_a
	$k = 2$	③ 150	0.0025	29.3%	0.0285	4.6%
	$k = 4$	② 75	③ 0.0022	50%	③ 0.0101	17.5%
	$k = 6$	① 50	0.0032	53.4%	0.0127	20.8%
surface	$d = 4$	300	① 0.0003	15.8%	② 0.0094	1%
surface	$d = 3$	156	0.0708	100%	0.1761	100%
surface	$d = 5$	492	② 0.0013	100%	① 0.0036	100%

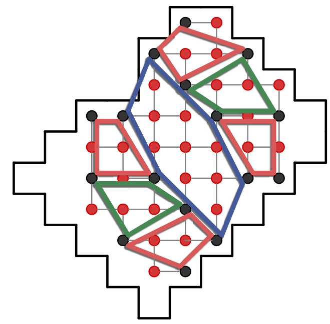


FIG. 12. A degree-four layout for flag-fault-tolerant error correction of the $k = 2$ code, using 43 of the 53 qubits on the Google Sycamore lattice. The stabilizer generators of the code are overlaid. Note that qubits have a degree of 4 only in the ancillas measuring the weight-eight stabilizer, but elsewhere the maximum qubit degree is 3. It may be possible for the $k = 2$ subsystem code with only weight-four stabilizers and gauges to fit on a layout with a maximum degree of 3.

ancilla qubit in Fig. 2 is connected to eight neighboring qubits. However, careful analysis and modification of the stabilizer measurement routines may yield solutions that only require maximum qubit degree five or six. This may not be interesting for densely connected ion-trap quantum computers but is necessary in superconducting architectures to maintain low cross talk. Alternatively, we showed that if we are allowed extra ancilla qubits, maximum degree four is possible, as in the Google Sycamore lattice in Fig. 12. The stabilizer measurement circuits are all fault tolerant to distance four, but since all the stabilizer generators are measured simultaneously, error decoding will require new strategies. The weight-four stabilizers can be measured using the circuit in Fig. 6(a), but the weight-eight stabilizer requires a new circuit, as we detail in Appendix A. In Fig. 12, the only qubits with degree-four connectivity are the ancillas used for measuring the weight-eight stabilizer. For systems with high cross talk, qubits of degree three may be sufficient to correct errors on the $k = 2$ subsystem code since errors can be corrected by measuring only weight-four operators.

On the theoretical front, we must develop encoding circuits and a universal logical gate set. States may be prepared either by using flags for fault tolerance or by combining patches of distance-two code states into a distance-four state. For fault-tolerant universal computation, one possible route is teleportation and logical measurements with distilled magic states. In fact, logical measurements can be performed along with error correction [38]. Another route to universality is to use transversal multiqubit gates between vertically stacked code patches. It may be possible for gates like the CCZ to induce magic [57], as the required $[[15, 7, 3]]$ code can be obtained by puncturing the $[[16, 6, 4]]$ code. If the error introduced by logical operations is kept low, many logical gates can be applied every time step, allowing high-depth logical circuits.

Near the threshold of the odd-distance surface codes, postselection on the distance-two and -four variants shows reduced logical error rates. With higher distance, the compounding effect is larger, meaning distance-eight or -ten surface codes may be sufficient for very precise computations. At higher distance, rejections also become exceedingly rare, increasing the possible duration of computations. For larger patches on lattices like that in Fig. 2, more qubits can be encoded at high distance.

The biggest difficulty will then be in performing operations on or between different logical qubits in the same patch. Another avenue to pursue is concatenation. This technique can combine the low logical error rates of the surface code with the high encoding rates of block codes.

VI. CONCLUSION

We simulated postselected fault-tolerant error correction for distance-four 16-qubit codes. Qubits were positioned on a 2D grid, and the only interactions allowed were local CNOT gates. Flag qubits were used to measure high-weight stabilizers, allowing for small error correction circuits for block codes encoding up to six logical qubits. With a phenomenological noise model, it is not possible to correct errors reliably using a syndrome of just the stabilizer generators. Instead, we proposed longer sequences of stabilizers that use redundant measurements to detect syndrome bit flips. These techniques are still in early development, and we suggested multiple improvements for each of them along the way.

We showed a variety of results interpolating between low logical error rates and low physical overhead. The downside of using postselection is that the logical qubits cannot be stored for too long, but low-depth NISQ algorithms are definitely possible. For these shorter timescales, we compared the logical error probability to a distance-five surface code: (1) The distance-four surface code has at least a 25-fold decrease in logical error rate. (2) A $k = 2$ color code with weight-eight stabilizers halves the probability of a logical error with only 30% of the qubits. (3) A $k = 6$ block code matches performance with just 25 qubits as opposed to 246. Using just 25–75 physical qubits, we demonstrated that it is possible to protect well 6 to 12 logical qubits.

ACKNOWLEDGEMENTS

The authors thank T. Brun, A. Lanka, D. Debroy, N. Delfosse and Z. Jiang for useful conversations. Research supported by Google and by MURI Grant No. FA9550-18-1-0161. This material is based on work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. The authors acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources that have contributed to the research results reported within this publication [58].

TABLE VI. For the circuit in Fig. 7, we list the appropriate corrections for flag outcomes caused by single faults and the list of flag outcomes that result from two faults.

Raised flags	Correction	Raised flags	Correction
{13}	{6}	{12,13}	{6}
{9,12}	{1,5}	{12,13,14}	{6,7,8}
{9,11,12,14}	{1,4,5}		

Rejections
{9,11}, {9,13}, {9,14}, {11,12}, {12,14}, {13,14}, {9,11,12}, {9,11,14}, {9,12,13}, {9,12,14}, {9,13,14}, {11,12,13}, {11,12,14}, {11,13,14}, {9,11,12,13}, {9,11,13,14}, {9,12,13,14}, {11,12,13,14}, {9,11,12,13,14}

APPENDIX A: CORRECTIONS AND REJECTIONS FOR WEIGHT-EIGHT STABILIZER MEASUREMENTS

The corrections and rejections for the fault-tolerant weight-eight stabilizer measurement circuit in Fig. 7(a) are shown in Table VI.

For the layout described in Fig. 12, the weight-eight stabilizer is measured fault tolerantly with the circuit in Fig. 13, with associated corrections and rejections tabulated in Table VII.

APPENDIX B: COMPARING MEMORY AGAINST UNENCODED QUBITS

In this section, we determine whether information protected by fault-tolerant error correction is more reliable than being stored in an unprotected qubit. For the postselection codes, Fig. 14 plots the CNOT depth at which qubits have accumulated 1% probability of logical error. The unprotected qubit is modeled to accumulate errors only through rest noise, whereas logical errors in the encoded qubits are due to circuits for fault-tolerant error correction. For the error rates we consider ($\leqslant 10^{-3}$), it is clear that the encoded qubits are better preserved for much longer than an unprotected qubit.

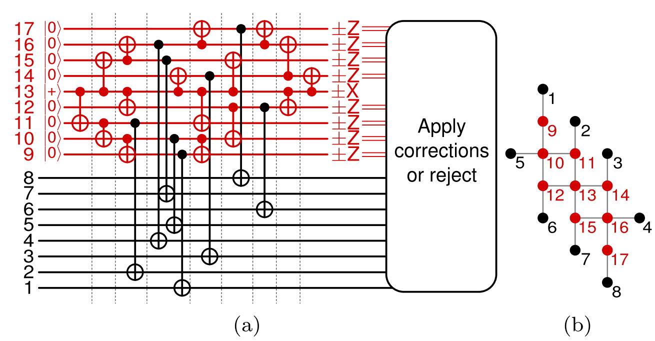


FIG. 13. (a) Distance-four fault-tolerant circuit for measuring a weight-eight stabilizer on a square lattice layout, as arranged in (b).

TABLE VII. For the circuit in Fig. 13, we list the appropriate corrections for flag outcomes caused by single faults and the list of flag outcomes that result from two faults.

Raised flags	Correction	Raised flags	Correction
{10}	{1}	{11}	{2}
{12}	{6}	{14}	{8}
{15}	{7}	{16}	{8}
{10,11}	{1,2}	{10,12}	{1,6}
{15,16}	{7,8}	{10,11,15,16}	{3}

Rejections

{9, 11}, {9, 12}, {9, 15}, {9, 16}, {9, 17}, {10, 14}, {10, 15}, {10, 16}, {10, 17}, {11, 12}, {11, 15}, {11, 16}, {11, 17}, {12, 15}, {12, 16}, {12, 17}, {14, 15}, {15, 17}, {9, 10, 11}, {9, 10, 12}, {9, 10, 15}, {9, 10, 16}, {9, 10, 17}, {9, 14, 16}, {9, 15, 16}, {10, 11, 12}, {10, 11, 14}, {10, 11, 15}, {10, 11, 16}, {10, 11, 17}, {10, 12, 14}, {10, 12, 15}, {10, 12, 16}, {10, 12, 17}, {10, 14, 16}, {10, 15, 16}, {10, 16, 17}, {11, 12, 15}, {11, 12, 16}, {11, 14, 16}, {11, 15, 16}, {12, 14, 16}, {12, 15, 16}, {14, 15, 16}, {14, 16, 17}, {15, 16, 17}, {9, 10, 14, 16}, {9, 10, 15, 16}, {9, 11, 15, 16}, {10, 11, 12, 14}, {10, 11, 14, 15}, {10, 11, 14, 16}, {10, 11, 15, 17}, {10, 11, 16, 17}, {10, 12, 14, 15}, {10, 12, 14, 16}, {10, 12, 15, 16}, {11, 12, 14, 16}, {12, 14, 15, 16}, {9, 10, 11, 15, 16}, {9, 11, 12, 15, 16}, {10, 11, 12, 14, 15}, {10, 11, 12, 14, 16}, {10, 11, 12, 15, 16}, {10, 11, 14, 15, 16}, {10, 11, 15, 16, 17}, {11, 12, 14, 15, 16}, {11, 12, 15, 16, 17}, {9, 10, 11, 12, 15, 16}, {10, 11, 12, 14, 15, 16}

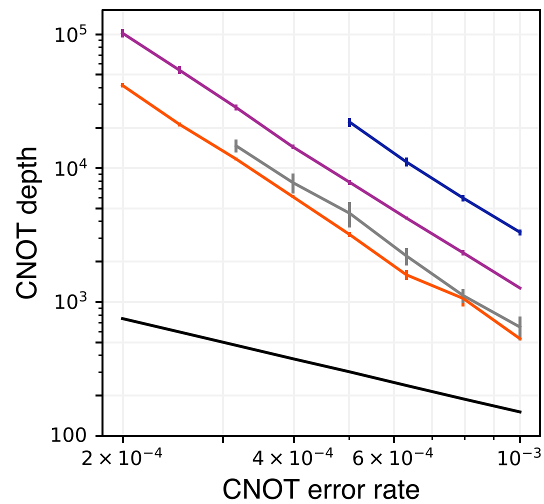


FIG. 14. CNOT depth at which each code has accumulated 1% probability of the X logical error. In black, the depth is plotted for one unencoded qubit at a rest error rate 1/10 the CNOT error rate. Plots shown for the $k = 2$ (blue), $k = 4$ (purple), and $k = 6$ (orange) and $d = 4$ surface (gray) codes assume the depth of ancilla qubit measurement is 10 times the depth of a CNOT gate. The CNOT depth shown for the surface code is for 0.01% probability of the X logical error. All data points shown for the postselection-equipped codes have acceptance >5%.

[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[2] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[4] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, Whole-device entanglement in a 65-qubit superconducting quantum computer, Adv. Quantum Technol. **4**, 2100061 (2021).

[5] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[6] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Hofer, C. Wachter, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz, Compact ion-trap quantum computing demonstrator, PRX Quantum **2**, 020343 (2021).

[7] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Phys. Rev. A **52**, R2493(R) (1995).

[8] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology, 1997.

[9] B. M. Terhal, Quantum error correction for quantum memories, Rev. Mod. Phys. **87**, 307 (2015).

[10] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz, Realization of real-time fault-tolerant quantum error correction, Phys. Rev. X **11**, 041058 (2021).

[11] Z. Chen, M. Plasencia, Z. Li, S. Mukherjee, D. Patra, C.-L. Chen, T. Klose, X.-Q. Yao, A. A. Kossiakoff, L. Chang *et al.*, Exponential suppression of bit or phase errors with cyclic error correction, Nature (London) **595**, 600 (2021).

[12] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina *et al.*, Fault-tolerant control of an error-corrected qubit, Nature (London) **598**, 281 (2021).

[13] J. R. Wootton, Benchmarking near-term devices with quantum error correction, Quantum Sci. Technol. **5**, 044004 (2020).

[14] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[15] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann,

T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[16] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, Fault-tolerant quantum error detection, Sci. Adv. **3**, e1701074 (2017).

[17] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, Experimental demonstration of fault-tolerant state preparation with superconducting qubits, Phys. Rev. Lett. **119**, 180501 (2017).

[18] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, Repeated quantum error detection in a surface code, Nat. Phys. **16**, 875 (2020).

[19] A. Paetznick and B. M. Reichardt, Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit Golay code, Quantum Inf. Comput. **12**, 1034 (2012).

[20] Y.-C. Zheng, C.-Y. Lai, and T. A. Brun, Efficient preparation of large-block-code ancilla states for fault-tolerant quantum computation, Phys. Rev. A **97**, 032331 (2018).

[21] S. Bravyi and A. Y. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[22] A. M. Meier, B. Eastin, and E. Knill, Magic-state distillation with the four-qubit code, Quantum Inf. Comput. **13**, 195 (2013).

[23] Y. Jing, D. Alsina, and M. Razavi, Quantum key distribution over quantum repeaters with encoding: Using error detection as an effective postselection tool, Phys. Rev. Appl. **14**, 064037 (2020).

[24] C. Sekga and M. Mafu, Security of quantum-key-distribution protocol by using the post-selection technique, Phys. Open **7**, 100075 (2021).

[25] S. Aaronson, Shadow tomography of quantum states, in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, edited by I. Diakonikolas, D. Kempe, and M. Henzinger (Association for Computing Machinery, New York, 2018), pp. 325–338.

[26] J. R. McClean, Z. Jiang, N. C. Rubin, R. Babbush, and H. Neven, Decoding quantum errors with subspace expansions, Nat. Commun. **11**, 636 (2020).

[27] E. Knill, Quantum computing with realistically noisy devices, Nature (London) **434**, 39 (2005).

[28] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt, Quantum computations on a topologically encoded qubit, Science **345**, 302 (2014).

[29] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[30] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, Nature (London) **549**, 172 (2017).

[31] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and subsystem codes on low-degree graphs with flag qubits, Phys. Rev. X **10**, 011022 (2020).

[32] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, Triangular color codes on trivalent graphs with flag qubits, New J. Phys. **22**, 023019 (2020).

[33] C. Gidney, M. Newman, A. Fowler, and M. Broughton, A fault-tolerant honeycomb memory, Quantum **5**, 605 (2021).

[34] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).

[35] R. Chao and B. W. Reichardt, Quantum error correction with only two extra qubits, Phys. Rev. Lett. **121**, 050502 (2018).

[36] R. Chao and B. W. Reichardt, Flag fault-tolerant error correction for any stabilizer code, PRX Quantum **1**, 010302 (2020).

[37] P. Prabhu and B. W. Reichardt, Fault-tolerant syndrome extraction and cat state preparation with fewer qubits, in *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*, edited by M.-H. Hsieh (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021), pp. 51–513.

[38] N. Delfosse and B. W. Reichardt, Short Shor-style syndrome sequences, arXiv:2008.05051.

[39] H. Bombin and M. A. Martin-Delgado, Optimal resources for topological two-dimensional stabilizer codes: Comparative study, Phys. Rev. A **76**, 012305 (2007).

[40] K. M. Svore, D. P. DiVincenzo, and B. M. Terhal, Noise threshold for a fault-tolerant two-dimensional lattice architecture, Quantum Inf. Comput. **7**, 297 (2007).

[41] F. M. Spedalieri and V. P. Roychowdhury, Latency in local, two-dimensional, fault-tolerant quantum computing, Quantum Inf. Comput. **9**, 666 (2009).

[42] C.-Y. Lai, G. Paz, M. Suchara, and T. A. Brun, Performance and error analysis of Knill's postselection scheme in a two-dimensional architecture, Quantum Inf. Comput. **14**, 807 (2014).

[43] B. Tan and J. Cong, Optimality study of existing quantum computing layout synthesis tools, IEEE Trans. Comput. **70**, 1363 (2021).

[44] Y. Tomita and K. M. Svore, Low-distance surface codes under realistic quantum noise, Phys. Rev. A **90**, 062320 (2014).

[45] M. Grassl, Bounds on the minimum distance of linear codes and quantum codes, 2007, http://www.codetables.de.

[46] P. W. Shor, Fault-tolerant quantum computation, in *FOCS '56: Proceedings of the 37th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, USA, 1996), p. 56.

[47] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, Phys. Rev. Lett. **78**, 2252 (1997).

[48] E. Knill, Scalable quantum computing in the presence of large detected-error rates, Phys. Rev. A **71**, 042322 (2005).

[49] D. Gottesman, The Heisenberg representation of quantum computers, in *Group Theoretical Methods in Physics*, edited by S. P. Corney, R. Delbourgo, and P. D. Jarvis (International Press, Hobart, Australia, 1998), p. 32.

[50] B. W. Reichardt, Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits, Quantum Sci. Technol. **6**, 015007 (2021).

[51] S. Huang and K. R. Brown, Between Shor and Steane: A unifying construction for measuring error syndromes, Phys. Rev. Lett. **127**, 090505 (2021).

[52] D. Ristè, L. C. G. Govia, B. Donovan, S. D. Fallek, W. D. Kalfus, M. Brink, N. T. Bronn, and T. A. Ohki, Real-time processing of stabilizer measurements in a bit-flip code, npj Quantum Inf. **6**, 71 (2020).

[53] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2006).

[54] N. Delfosse, B. W. Reichardt, and K. M. Svore, Beyond single-shot fault-tolerant quantum error correction, IEEE Trans. Inf. Theory **68**, 287 (2022).

[55] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, Quantum **5**, 595 (2021).

[56] L. Viola, E. Knill, and S. Lloyd, Dynamical decoupling of open quantum systems, Phys. Rev. Lett. **82**, 2417 (1999).

[57] A. Paetznick and B. W. Reichardt, Universal fault-tolerant quantum computation with only transversal gates and error correction, Phys. Rev. Lett. **111**, 090505 (2013).

[58] https://carc.usc.edu.