Low-distance surface codes under realistic quantum noise

Yu Tomita[1] and Krysta M. Svore[2]

[1]*School of Computational Science and Engineering, Georgia Institute of Technology, North Avenue NW, Atlanta, Georgia 30332, USA*

[2]*Quantum Architectures and Computation Group, Microsoft Research, 1 Microsoft Way, Redmond, Washington 98052, USA*

(Received 13 June 2014; published 11 December 2014)

Experimental implementation of the surface code will be a significant milestone for quantum computing. We develop a circuit and a decoder targeted for near-term implementation of a distance-3 surface code. We simulate the code under amplitude and phase damping and compare the threshold to a Pauli-twirl approximation. We find that the approximation yields a pessimistic threshold estimate. From numerical Monte Carlo simulations, we identify the gate and measurement speeds required to achieve reliable error correction. For superconductor devices, a qubit encoded in a 17-qubit surface code demonstrates a lower error rate than an unencoded qubit assuming gate times of 5–40 ns and T_1 times of at least 1–2 μs. If $T_1 \geqslant 10$ ns, the difference is significant and can be experimentally measured, allowing near-term implementation and verification of a small surface code. For ion trap devices, gates times of 1 μs and $T_1 \geqslant 40$ ms admit measurable differences in error rate.

DOI: 10.1103/PhysRevA.90.062320 PACS number(s): 03.67.Lx, 03.67.Pp

I. INTRODUCTION

Topological quantum error-correcting codes are a leading approach to scalable fault-tolerant quantum computation [1,2]. The most practical topological code to date is the surface code, which calls for a two-dimensional (2D) planar qubit layout with only nearest-neighbor interactions [3–7]. It has been shown to allow error rates up to a threshold of approximately 1% [1,8–10]. Several quantum architectures, including superconducting devices [11,12] and ion traps [13–16], are suitable for realizing the surface code. Recent experiments on superconducting qubits have even demonstrated error rates in the required range [17].

Until recently, the threshold for the surface code has been primarily calculated for the depolarizing channel [1,8–10]. Simulation of the surface code and the depolarizing channel requires only Clifford operations and Pauli measurements on stabilizer states, allowing efficient simulation on a classical computer under the Gottesman-Knill theorem [18,19].

It has been shown that realistic quantum noise such as decoherence can be sufficiently approximated by a depolarizing noise model parametrized by a method such as Pauli twirling [20,21], enabling efficient simulation. Simulations of the surface code with noise based on Pauli-twirl approximations have been performed for several superconductor architectures [22]. Other studies have achieved efficient classical simulation of realistic noise models by using Clifford gates to approximate arbitrary gates [23,24] and amplitude damping [25].

More recently, it has been shown that the surface code threshold is significantly degraded in the presence of qubit leakage in conjunction with depolarizing noise [26]. It has also been shown to achieve arbitrary reliability given modest additional qubit resources under local many-qubit errors and nonlocal two-qubit errors [27]. A recent study has determined a threshold for the surface code considering correlated errors and the coupling between qubits and the environment by formulating the problem as an Ising model [28].

In all cases, the threshold has been calculated for a standard surface code layout. Variations of the surface code layout have been proposed [29,30] that reduce the qubit and gate resources necessary for implementation. To the best of our knowledge, the pseudothresholds for modified surface code layouts have not been analyzed. In addition, studies of the threshold and pseudothresholds under realistic (non-Clifford) noise models have been limited due to the exponential cost of simulation. With device error rates rapidly approaching the surface code threshold, it is timely to investigate the performance and requirements of low-distance surface code layouts for near-term experimental implementation.

In this paper, we study low-distance surface code layouts optimized for near-term experimental implementation. We develop a fast decoder for these layouts which can run in limited-memory, low-temperature environments. We simulate the layouts under depolarizing noise, amplitude, and phase damping and a Paul-twirl approximation [22]. While depolarizing noise and the approximate model can be simulated efficiently, simulation of amplitude and phase damping is exponential. In response, we perform simulations using the LIQ$Ui|\rangle$ software architecture [31], which is highly optimized to allow simulation of up to 31 qubits in 32-GB RAM. Assuming realistic noise, we determine the gate and measurement speeds for superconductor and ion trap architectures that are required for reliable error correction with a small surface code. Moreover, we find that the error rate of a qubit encoded in a 17-qubit surface code is significantly better than an unencoded qubit, and the difference can be experimentally detected. Our work provides a framework for near-term demonstration of a reliable small-distance surface code acting as a single-qubit quantum memory.

Our paper is organized as follows. Section II briefly reviews the surface code and three layouts for the distance-3 code. We introduce our decoding method, based on a small lookup table, in Sec. III. Section IV describes the realistic noise models and their approximations. Our experimental methodology is introduced in Sec. V. In Sec. VI, we present our surface code simulation results. Finally, we conclude in Sec. VII.

II. LOW-DISTANCE SURFACE CODES

The surface code is a stabilizer code arranged on a 2D lattice with nearest-neighbor interactions [3]. It encodes a single

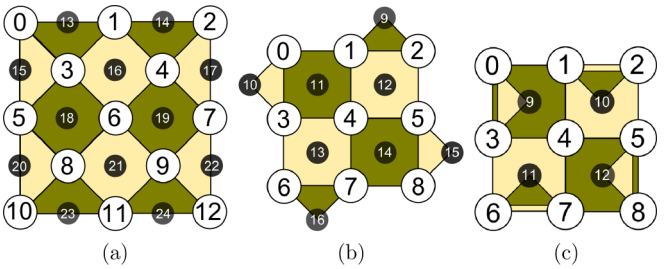


FIG. 1. (Color online) Distance-3 surface code layouts with (a) 25, (b) 17, and (c) 13 qubits. White circles represent data qubits; black circles, syndrome qubits. Dark square and triangular patches represent X stabilizers; light patches represent Z stabilizers. The layered patches on surface-13 indicate use of the syndrome qubit first to measure a four-qubit stabilizer and then to measure a two-qubit stabilizer.

logical qubit in a number of physical qubits that is determined by the code distance d and desired layout (described below). Through repeated measurement of its stabilizer generators, the surface code in conjunction with a classical decoding algorithm can detect errors and subsequently correct up to $\lfloor (d-1)/2 \rfloor$ physical errors. The distance d dictates the length of the shortest undetectable error chain and in turn is also the length of the shortest logical operator. For an excellent review of the surface code, we refer the reader to [1].

A. Twenty-five-qubit layout

We study three distance $d = 3$ layouts, shown in Fig. 1. We begin by discussing the standard layout, referred to as surface-25, shown in Fig. 1(a). It uses a $(2d-1) \times (2d-1)$ square grid of qubits with a smooth and rough boundary [5]. For $d = 3$, the grid contains 25 qubits, of which 13 *data* qubits (large white circles) are used to encode the logical qubit and 12 *syndrome* qubits (small black circles) are used to extract the error syndromes by way of stabilizer measurements.

Surface-25 is simultaneously stabilized by the group of stabilizer generators listed in Table II. In Fig. 1, the Z stabilizers are represented by light (yellow) patches and the X stabilizers are represented by dark (green) patches, where each patch represents a tensor product of Z (or X) operators on the data qubits surrounding the patch.

A logical X operator X_L is defined as a chain of physical X operations between two data qubits on opposite smooth boundaries (top and bottom edges). The chain is allowed to cross any Z stabilizer patch and follow any edge of an X stabilizer patch. A logical Z operator Z_L is defined analogously as a chain of physical Z operations between two data qubits on opposite rough boundaries (left and right edges). Table II lists one possible logical X and Z operator. There are 2^G equivalent logical operators for each logical Pauli operator (X and Z), where G is the number of stabilizer generators for the given surface code. Since X_L and Z_L commute with all of the stabilizers and cannot be written as a product of them, logical errors, which come in the form of logical operators, cannot be detected by the code.

The surface code detects errors through the eigenvalues of the stabilizers. A bit-flip (phase-flip) on a data qubit will change the eigenvalue of adjacent Z (X) stabilizers. To extract an eigenvalue, also referred to as an *error syndrome*, a given stabilizer is measured. Figure 2 shows the standard quantum circuit for measuring the stabilizers [1,10], where data qubit b corresponds to the top (north) qubit and c corresponds to the bottom (south) qubit of each diamond-shaped patch in Fig. 1(a).

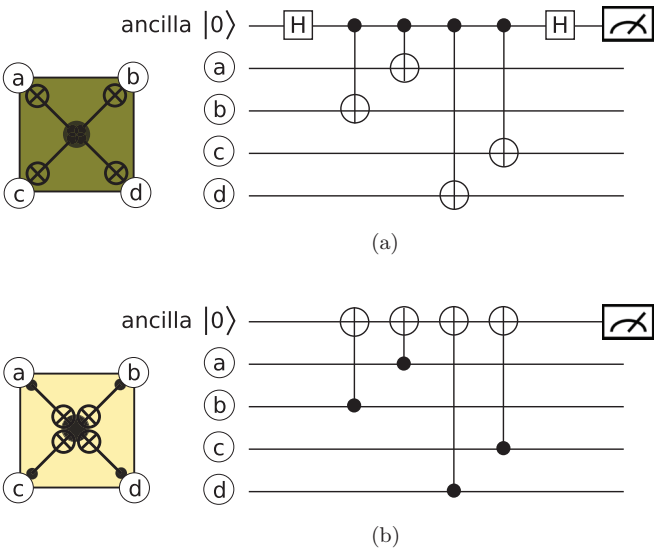


FIG. 2. (Color online) Standard quantum circuits to measure stabilizers (a) $X_a X_b X_c X_d$ and (b) $Z_a Z_b Z_c Z_d$.

The circuit begins with CNOT gates that propagate error information from the data qubits a, b, c, and d to the syndrome qubit (black circle). CNOT gates are performed in the order top (b), left (a), right (d), bottom (c). Cyclic orders, such as a clockwise or counterclockwise, i.e., $bdca$, fail to maintain commutation of nearby stabilizers, which in turn can cause random measurement outcomes [1]. Thus the order of CNOT gates is required to follow an S or Z shape.

The syndrome qubit is then measured to extract the eigenvalue of the stabilizer. These error syndromes are input to a classical decoding algorithm to determine an appropriate correction operator. Details of our decoding algorithm are given in Sec. III. The total number of operations in a given round of stabilizer measurements for the surface code is listed in Table I.

B. Thirteen- and 17-qubit layouts

The number of qubits in surface-25 can be reduced while maintaining the same code distance by rotating it clockwise by 45° and removing the four corner data qubits [29,30], shown

TABLE I. Number of operations in one round of the surface code for distance-3 layouts.

Code	CNOT	I	H	Measure	Prepare	Depth
Surface-13	24	99	8	8	8	14
Surface-17	24	56	8	8	8	8
Surface-25	40	72	12	12	12	8

TABLE II. List of X and Z stabilizers and logical X_L and Z_L operators for surface-13, -17, and -25.

Surface-25		Surface-13, surface-17	
X stabilizers	Z stabilizers	X stabilizers	Z stabilizers
$X_0X_1X_3$	$Z_0Z_3Z_5$	$X_0X_1X_3X_4$	Z_0Z_3
$X_1X_2X_4$	$Z_1Z_3Z_4Z_6$	X_1X_2	$Z_1Z_2Z_4Z_5$
$X_3X_5X_6X_8$	$Z_2Z_4Z_7$	$X_4X_5X_7X_8$	$Z_3Z_4Z_6Z_7$
$X_4X_6X_7X_9$	$Z_5Z_8Z_{10}$	X_6X_7	Z_5Z_8
$X_8X_{10}X_{11}$	$Z_6Z_8Z_9Z_{11}$		
$X_9X_{11}X_{12}$	$Z_7Z_9Z_{12}$		
Logical X	Logical Z	Logical X	Logical Z
$X_0X_5X_{10}$	$Z_0Z_1Z_2$	$X_2X_4X_6$	$Z_0Z_4Z_8$

in Fig. 1(b). The number of data qubits is reduced from 13 to 9 and the number of syndrome qubits is reduced to 8 for a total of 17 qubits. We call this layout surface-17. The stabilizer generators contain weight-4 and weight-2 stabilizers (Table II). Figure 3(a) shows the circuit for a simultaneous weight-4 X and weight-2 Z stabilizer measurement.

A further reduction in qubits can be obtained by reusing the syndrome qubits [30]. Surface-13 uses only four syndrome qubits as shown in Fig. 1(c). Each syndrome qubit is used twice, once for X stabilizer measurement and once for Z stabilizer measurement. Figure 3(b) contains the corresponding circuit for measuring a weight-4 X stabilizer followed by a weight-2 Z stabilizer. Surface-13 reduces the number of qubits but increases the depth of a round by four time steps. The depth and number of operations required for one round of the surface code for surface-17 and -13 are listed in Table I. The stabilizers and logical operations for these two layouts are listed in Table II.

Despite having fewer stabilizers, surface-17 and surface-13 still remain distance-3 surface codes [29,30]. Due to their reduction in resources by 32%–48%, these layouts are promising candidates for early experimental implementation. In Sec. VI, we determine which layout is most promising based on its pseudothreshold and resource costs.

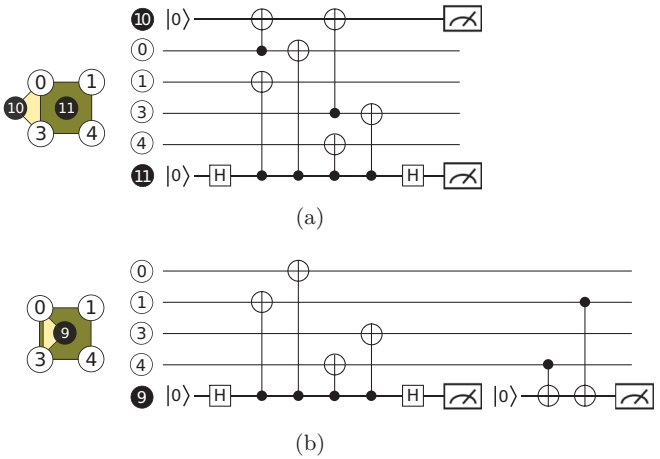


FIG. 3. (Color online) Quantum circuits for measuring $X_0X_1X_3X_4$ and Z_0Z_3 in (a) surface-17 and (b) surface-13.

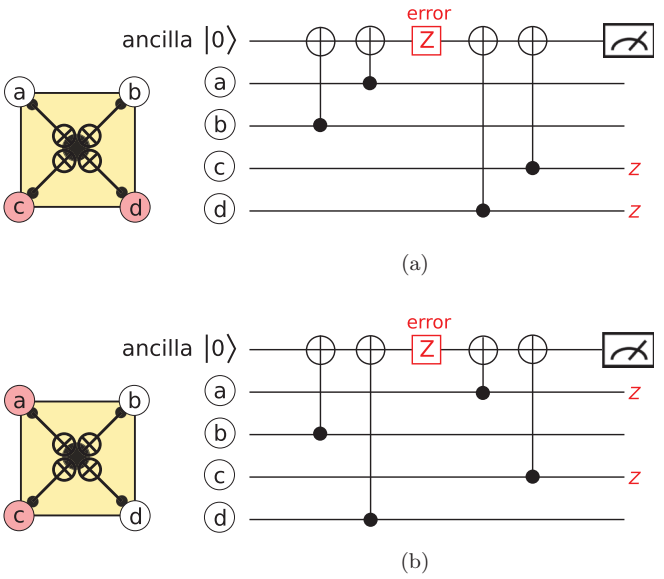


FIG. 4. (Color online) (a) Stabilizer measurement circuit showing the propagation of a Z error to two Z errors. (b) Reordered stabilizer measurement circuit.

C. Improved stabilizer measurement circuits

In our simulations of surface-13 and -17 under noise, we find that using the same CNOT ordering for both X- and Z-type stabilizer measurements can result in a single error on a syndrome qubit, leading to a logical X or Z error (details on noise are given in Sec. IV). Figure 4(a) shows an example. A Z error on a Z-stabilizer syndrome qubit after the first two CNOT gates propagates onto two horizontally aligned data qubits. Since our surface codes require only three data qubits to complete a logical error chain, the next round of syndrome measurements will incorrectly diagnose a Z error on the third qubit, leading to a logical Z error chain.

To prevent the creation of a logical error, we propose to measure X- and Z-type stabilizers in different orders. The sequence for X stabilizers is the same as in Fig. 2. We modify the order of CNOTs in Z stabilizers as top right (b), bottom right (d), top left (a), bottom left (c) [Fig. 4(b)]. This order maintains the alignment of qubits a and c such that they are perpendicular to the direction of the corresponding logical chain. It also preserves the commutation relations as well as the circuit depth and size. Figure 4 shows an example where two Z errors map to a single Z error with the improved order, versus a logical error with the old order. We use this improved order for all simulations in this paper.

III. DECODING METHOD

A standard method for mapping error syndromes to the most probable error chain is the minimum-weight perfect matching algorithm [8,32,33]. It requires time $O(n)$ for n detection events if executed serially and time $O(1)$ if executed in parallel [34]. The algorithm independently corrects X and Z errors by identifying the most likely error chain for each type such that the total chain weight is minimal. The algorithm has recently been extended to handle correlations between X and Z errors, in which case the chains are not constructed

independently [35]. Corrections are applied along the chain(s). If, after correction, a chain of errors connecting two smooth (rough) boundaries remains, then a logical error has occurred. If errors are assumed to be independent, then long chains will be exponentially unlikely.

Lookup table decoder

In this work we target first-generation implementations of a single qubit protected by a small surface code. While the classical time and space requirements of the minimum-weight perfect matching algorithm are modest, we further reduce the classical computational overhead by designing a lookup table based on the algorithm that can be implemented on a small classical device. Our lookup table is designed to find the most probable low-weight error chain from a short history of error syndromes.

Consider the set of error syndromes that indicate an error after one full (noisy) round of the surface code, that is, those indicating a -1 eigenvalue. Based on the error syndrome locations, the decoder determines the probable data-qubit error locations. For example, consider a Z error on qubit 4 in surface-17 [Fig. 1(b)]. Given that no other errors occur, after one round of the surface code syndrome qubits 11 and 14 will indicate an error. The decoder will determine that the shortest error chain connecting these two syndromes includes data qubit 4. To correct the error chain, Z_4 will be applied.

As another example, consider an X error on qubit 6. It will cause syndrome 13 to indicate an error. Since syndromes 10 and 12 do not indicate errors, the decoder will infer an error on either data qubit 6 or data qubit 7. In this case, the decoder can correct either X_6 or X_7 since X_6X_7 is a stabilizer.

An error syndrome may also occur due to a measurement error. However, the decoder may interpret it as a data-qubit error. For example, consider a measurement error on qubit 11. The decoder will apply either Z_0 or Z_3 to "correct" the error, thereby adding an error to a clean data qubit.

To improve identification of actual data-qubit errors, inference is performed based on several rounds of stabilizer measurements [8]. Consider performing r rounds of the surface code consecutively. Instead of storing the syndromes for each round, we store the locations in time and space of the syndromes whose values change, or "flip," between the current and the previous round.

For r rounds, this requires storing a 3D space-time array containing at most $s \times r$ values, where s is the maximum number of syndrome changes in a round. We refer to this 3D array as the *syndrome volume*, where dimension r represents time. The goal is to determine a correction operator (a product of X and/or Z operators) based on the syndrome volume such that the number of errors remaining after correction is minimized, in turn reducing the chance of forming a logical error chain.

Our lookup table is based on the fact that short error chains are more likely than long chains. Assuming that a syndrome volume contains r rounds, we construct a lookup table based on the following rules (Fig. 5 shows the rules visually):

(1) If the same syndrome flips twice in two consecutive rounds, the pair (in time) of syndromes is ignored since it most likely indicates a measurement error.

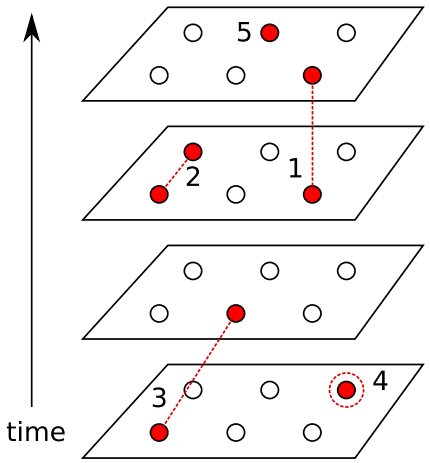


FIG. 5. (Color online) Lookup table decoding rules. Each circle represents a syndrome measurement. Filled (red) circles indicate "flips." Five types of flips are shown: (1) measurement error, (2, 3) paired flips indicating a single-qubit error on the data qubit, (4) a single flip indicating one data-qubit error, and (5) an undetermined flip. These numbers correspond to the rules in Sec. III.

(2) If a pair (in space) of neighboring syndromes flips in the same round, a correction on the data qubit between the pair is applied.

(3) If a syndrome flips in round $r-1$ and its neighboring syndrome flips in round r, a correction on the data qubit between the pair (in time) is applied.

(4) If a syndrome flips only once and in a round other than the last, a correction is applied to a data qubit on the boundary such that the data qubit is not between two stabilizers that did not indicate a syndrome.

(5) If a single syndrome flips only once and in the last round, the information is kept until the next round of error correction. No correction based on this syndrome is applied. In this case the location of the error, if any, is inconclusive without another round of syndrome measurements.

We decode by checking the above rules in order and determining the set of data-qubit error locations. We then switch the order of rules 2 and 3 and determine another set of possible error locations. We correct based on the set with fewer error locations, since fewer errors are more likely. Here we assume that $r = 3$.

These rules are equivalent to the minimum-weight perfect matching algorithm applied to only neighboring-syndrome pairs, with a uniform weight for the same distance. Since our surface codes are small, performance of the code does not improve when decoding considers more distant pairs.

We encode these rules into a lookup table. The lookup table maps the syndrome volume of measurement flips to a set of probable errors on the data qubits. The table requires constant time and $2n$ space, where n is the number of data qubits. The lookup table minimizes the time and space requirements of decoding, enabling implementation, for example, in limited cryogenic classical logic.

IV. NOISE MODELS

In this section, we present the noise models considered in our surface code simulations. We review two noise models that can be simulated efficiently on a classical computer (depolarizing and Pauli-twirl approximation) and one noise model that requires exponential memory to simulate (amplitude and phase damping).

A. Symmetric and asymmetric depolarizing channels

The depolarizing channel (D) is a standard quantum noise model in which a qubit becomes depolarized with a given probability p. This channel transforms a density matrix of a single qubit as

$$\rho \to \epsilon_D(\rho) = p_I\rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z, \quad (1)$$

where $p_I = (1 - p_X - p_Y - p_Z)$. In this model, a qubit suffers from discrete Pauli bit-flip (X), phase-flip (Z), or bit-and-phase flip (Y) errors with probabilities p_X, p_Z, and p_Y, respectively. When $p_X = p_Y = p_Z$, this channel is called a *symmetric* depolarizing channel. When the probabilities are independent, the model is called an *asymmetric* depolarizing channel.

B. Amplitude and phase damping channel

The amplitude damping channel (AD) characterizes the behavior of energy dissipation of the quantum system, including spontaneous emission of a photon from a qubit. This channel transforms the density matrix of a single qubit as

$$\rho \to \epsilon_{\mathrm{AD}}(\rho) = E_1^{\mathrm{AD}}\rho E_1^{\mathrm{AD}\dagger} + E_2^{\mathrm{AD}}\rho E_2^{\mathrm{AD}\dagger}, \quad (2)$$

where

$$E_1^{\mathrm{AD}} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_{\mathrm{AD}}} \end{pmatrix}, \quad E_2^{\mathrm{AD}} = \begin{pmatrix} 0 & \sqrt{p_{\mathrm{AD}}} \\ 0 & 0 \end{pmatrix}, \quad (3)$$

and p_{AD} is the probability of a qubit emitting a single photon.

Figure 6 expresses the amplitude damping of a single qubit in the form of a quantum circuit where an ancilla qubit is used to represent the environment and $\sin^2(\theta/2) = p_{\mathrm{AD}}$ [21]. The input is an arbitrary single-qubit state $|\psi_{\mathrm{in}}\rangle = a|0\rangle + b|1\rangle$ and the output state is given by

$$|\psi_{\mathrm{out}}\rangle = \begin{cases} Na|0\rangle + Nb\sin(\theta/2)|1\rangle & \text{if measure 0,} \\ |0\rangle & \text{if measure 1,} \end{cases}$$

where N is a normalization constant. The probabilities of measuring 0 and 1 are $1 - b^2 p_{\mathrm{AD}}$ and $b^2 p_{\mathrm{AD}}$, respectively.

During simulation, we do not use an extra ancilla as shown in the circuit in Fig. 6. Instead, we calculate the probability of measuring 0 and 1 given input state $|\psi_{\mathrm{in}}\rangle$ and simulate the measurement outcome with a random number. When the simulated measurement is 0, we apply the rotation $R_y(\theta)$ on $|\psi_{\mathrm{in}}\rangle$. When it is 1, we apply damping and the state becomes $|0\rangle$.

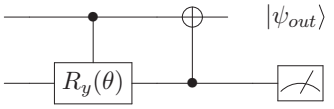


FIG. 6. Circuit representation of amplitude damping [21].

The phase damping channel (PD) is described similarly as

$$\rho \to \epsilon_{\mathrm{PD}}(\rho) = E_1^{\mathrm{PD}}\rho E_1^{\mathrm{PD}\dagger} + E_2^{\mathrm{PD}}\rho E_2^{\mathrm{PD}\dagger}, \quad (4)$$

where

$$E_1^{\mathrm{PD}} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_{\mathrm{PD}}} \end{pmatrix}, \quad E_2^{\mathrm{PD}} = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{p_{\mathrm{PD}}} \end{pmatrix}. \quad (5)$$

Phase damping noise, also called pure dephasing, is equivalent to the phase-flip channel. By unitary freedom of operator-sum representation, we can derive a new set of operation elements to express the channel in terms of the probability of a phase-flip (Z) error,

$$E_1'^{\mathrm{PD}} = \sqrt{1-p_Z}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad E_2'^{\mathrm{PD}} = \sqrt{p_Z}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (6)$$

where $p_Z = \frac{1-\sqrt{1-p_{\mathrm{PD}}}}{2}$.

We assume that amplitude and phase damping (APD) are the main sources of decoherence. Using these two channels together, decoherence on a single qubit transforms the density matrix as

$$\rho \to \epsilon_{\mathrm{APD}}(\rho) = \begin{pmatrix} 1-\rho_{11}e^{-t/T_1} & \rho_{01}e^{-t/T_2} \\ \rho_{01}^* e^{-t/T_2} & \rho_{11}e^{-t/T_1} \end{pmatrix}, \quad (7)$$

where t is the execution time of the gate including identity, T_1 and T_2 are the single-qubit relaxation and dephasing times, respectively, and $e^{-t/T1} = 1 - p_{\mathrm{AD}}$ and $e^{-t/T2} = \sqrt{(1-p_{\mathrm{AD}})(1-p_{\mathrm{PD}})}$ [21].

C. Approximate amplitude and phase damping channel

Using a technique called Pauli twirling (PT) [20], a Pauli channel ϵ_T can be used to approximate the decoherence channel given in Eq. (7) [22,36], where

$$\epsilon_{\mathrm{PT}}(\rho) = \frac{1}{4}\sum_{A\in 1-X,Y,Z} A^\dagger\epsilon(A\rho A^\dagger)A. \quad (8)$$

Twirling results in removal of the off-diagonal terms and in turn allows expression of the channel as an asymmetric depolarizing noise channel [given in Eq. (1)], with the probabilities given by

$$p_X = p_Y = \frac{1-e^{-t/T_1}}{4}, \quad (9)$$

$$p_Z = \frac{1-e^{-t/T_2}}{2} - \frac{1-e^{-t/T_1}}{4}, \quad (10)$$

where the probabilities of failure are expressed in terms of the execution time t of a gate, the qubit relaxation time T_1, and the qubit dephasing time T_2 [22].

Assuming that errors are independent, the probabilities of two-qubit errors, for example, when a CNOT gate fails, are

approximated as in [22] as

$$
\begin{aligned}
p_{I(X\text{ or }Y)} &= p_{(X\text{ or }Y)I} = p_X(1 - p_X - p_Y - p_Z),\\
p_{(X\text{ or }Y)(X\text{ or }Y)} &= p_X p_X,\\
p_{Z(X\text{ or }Y)} &= p_{(X\text{ or }Y)Z} = p_X p_Z,\\
p_{IZ} &= p_{ZI} = p_Z(1 - p_X - p_Y - p_Z),\\
p_{ZZ} &= p_Z p_Z.
\end{aligned}
$$

V. SIMULATION PROCEDURE

We use the LIQ$Ui|\rangle$ software architecture [31] to perform simulations of the surface code under noise. LIQ$Ui|\rangle$ (Language-Integrated Quantum Operations) contains an embedded, domain-specific language for programming quantum circuits as well as two circuit simulation environments. The first environment allows efficient simulation of Clifford circuits, based on the Gottesman-Knill theorem, and is called *stabilizer* simulation [18,19]. The second environment, called *universal* simulation, allows full simulation of arbitrary quantum circuits.

While some of our noise models allow stabilizer simulation, we have chosen, for consistency, to perform all simulations within the universal simulation environment. LIQ$Ui|\rangle$ allows universal simulation of a number of qubits that is limited by the main memory of the machine. We ran simulations on a large HPC cluster containing several hundred nodes with 32 GB of RAM each, allowing simulation of up to roughly 30 qubits on each node. Our simulations required thousands of hours of computing time.

We restrict the operations in our circuits to the five types listed in Table I, which we refer to as *location types*: I, H, CNOT, Prepare a $|0\rangle$ state, and Measure in the Z basis. When no location type is specified on a qubit, the identity gate I is applied to that qubit, where the duration of the identity is set by the location type occurring on other qubits in the time step. When a qubit is idle for a duration of t time steps (while gates are being applied on other qubits), we apply t identity gates to it to simplify the simulations. Figure 7 shows the circuit in Fig. 3(a) with identity gates inserted. Further circuit optimization can be performed, for example, by delaying qubit preparation and measuring a qubit as soon as gate operations complete. Such optimization will result in improved pseudothresholds. For simplicity, we choose to maintain gate alignment between stabilizers.

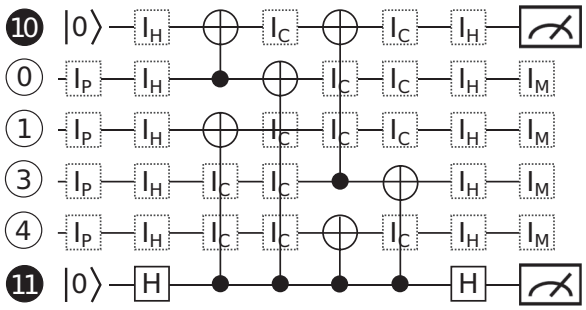


FIG. 7. Insertion of identity gates during simulation of the circuit in Fig. 3(a). Four identity gates are used, based on the other location type in the given time step: preparation (P), single-qubit gate (H), two-qubit gate (C), and measurement (M).

A. Monte Carlo simulation

We perform Monte Carlo simulation of the surface code layouts to compute the logical error rates. Monte Carlo simulation consists of repeated random sampling to estimate the logical error rate based on execution of the circuit with probabilistic errors on the locations. At each time step in the circuit, each qubit undergoes a location type followed by the given noise model. For depolarizing noise and approximate damping noise, we replace each location type except measurement with the location type followed by an X, Y, or Z gate ("error") with probability p_X, p_Y, and p_Z, respectively. In the case of measurements, X, Y, or Z errors are placed before the measurement location.

For amplitude and phase damping and the Pauli-twirl approximation, we apply the noise model after every location given the duration of the current time step t. The duration values we consider are listed in Table III.

B. Logical error rate calculation

We calculate the logical error rate of a given layout by simulating it under the various noise models. At the start of each simulation, we initialize all data qubits to $|0\rangle$ (if preparing $|0_L\rangle$) or $|+\rangle$ (if preparing $|1_L\rangle$) and run a noise-free cycle of syndrome measurements to project into an initial stabilizer state of the code. We refer to this state as the *quiescent* state [1]. Note that for a code with s stabilizers, there are 2^s possible quiescent states, since each stabilizer measurement can randomly project to a ± 1 eigenstate. In the absence of

TABLE III. Qubit relaxation, dephasing, and gate times assumed for different architectures. DiVincenzo and Helmer parameters are taken from [22]. SC, superconductor; IT, ion trap architecture.

Parameter	Location type	SC_S (slow)	SC_F (fast)	SC_D (DiVincenzo)	SC_H (Helmer)	IT_S (slow)	IT_F (fast)
T_1	Qubit relaxation time	T_1	T_1	T_1	T_1	T_1	T_1
T_2	Qubit dephasing time	T_1	T_1	$2\,T_1$	T_1	$0.1\,T_1$	$0.1\,T_1$
t_{prep}	State preparation	5 μs	1 μs	40 ns	40 ns	100 μs	30 μs
t_1	Single-qubit rotation	100 ns	10 ns	5 ns	5 ns	1 μs	1 μs
t_{meas}	Measurement	5 μs	1 μs	35 ns	35 ns	100 μs	30 μs
t_{CNOT}	CNOT	1 μs	100 ns	80 ns	20 ns	100 μs	10 μs
$t_{r,13}$	One round (S-13)	28.2 μs	4.82 μs	800 ns	320 ns	1202 μs	202 μs
$t_{r,17\&25}$	One round (S-17, S-25)	14.2 μs	2.42 μs	405 ns	165 ns	602 μs	102 μs

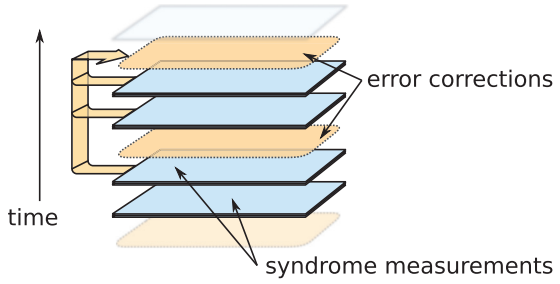


FIG. 8. (Color online) Illustration of the syndrome volume. Each window consists of three rounds of the surface code. Corrections are applied to the state after the final round in the window. The state is then checked for a logical error. An example window consists of the top three blue layers, where one layer overlaps with the previous window.

noise, the quiescent state will be maintained during subsequent rounds of the surface code.

After initialization of the quiescent state, the simulation proceeds as follows:

(1) Execute two rounds of the surface code with noise (execute three if this is the first execution of the loop). Record the list of syndrome flips between contiguous rounds in the syndrome volume. For the first round, compare to the quiescent state.

(2) Apply the decoder (Sec. III) to the three-layer syndrome volume to determine the most probable set of error locations.

(3) Apply noise-free corrections to the state. In practice, corrections can be tracked directly in software.

(4) Check for a logical error by calculating the distance of the state to the possible logical states. If the closest logical state is incorrect, count a logical error.

(5) Repeat from Step 1 until m logical errors are detected.

After each logical error check (step 4) the syndrome volume contains a list of unpaired syndrome flips due to the last two rules of our decoder. Each syndrome volume, as shown in Fig. 8, thus contains three layers: the final layer from the previous volume and two layers from two additional rounds of the surface code. We refer to the number of rounds in the volume as the *window size*. We experimented with various window sizes and found that three was optimal for distance-3 layouts. In our simulations, m varies between 10 and 200 depending on the size of the physical error rates.

We calculate the logical error rate per window since, in an experiment, the logical qubit will be measured after completion of a window to ensure optimal decoding and correction. For a window containing r rounds, the logical error rate P_r is given by

$$P_r = m/R, \tag{11}$$

where R represents the number of windows executed to observe m logical errors. When $r = 1$, Eq. (11) represents the logical error rate per round of the surface code.

Since we only calculate P_r for distance $d = 3$, we estimate the pseudothreshold [37,38], denoted P_r^{th}, as opposed to the asymptotic threshold as $d \to \infty$. The pseudothreshold can be defined by the crossing point between the line $x = y$ and the plot p vs P_r. If the error rate p of each physical location type falls below the pseudothreshold P_r^{th}, then the code is guaranteed to lower the logical error rate below p.

The logical error rate per window P_r and the logical error rate per round P_1 are related by

$$P_r \approx r P_1 (P_1)^{r-1} + (r-2) P_1^3 (1-P_1)^{r-3}. \tag{12}$$

For depolarizing noise, we calculate P_1 (to compare with previous work) and P_3. For amplitude and phase damping and the Pauli-twirl approximation, we calculate P_3.

C. Architecture settings

For amplitude and phase damping and the Pauli-twirl approximation, we consider several parameter settings derived from superconductor and ion trap architectures. These architectures are well suited to 2D, nearest-neighbor operations required for the surface code. Table III lists the different parameter settings considered for each architecture. The time per round $t_{r,\{13,17,25\}}$ indicates the time required to complete one round of the surface code given the other parameters. These six architecture settings represent a range of round times between 165 ns and 602×10^3 ns for surface-17 and surface-25. Note that the surface-13 layout requires roughly twice the amount of time as for surface-17.

Two-dimensional superconducting architectures have demonstrated fast single- and two-qubit gate execution times in recent years [17,39–41]. Current gate times are in the range of 10–20 and 30–80 ns for single-qubit and two-qubit gates, respectively, with experimental T_1 times as long as 20–40 μs [17,41]. The DiVincenzo (SC_D) [12] and Helmer (SC_H) [11] superconductor parameters are derived from [22]. SC_D requires longer CNOT gate times than SC_H. SC_S and SC_F represent parameters for slow and fast gate times, respectively, based on recent experiments [39,40]. In particular, they account for microsecond preparation and measurement times, while SC_D and SC_H assume nanosecond times.

Ion traps are another promising architecture with demonstrable quantum gates [13–16,42]. While trapped ion devices tend to have longer gate execution times than superconductor devices, they have been shown to have much longer relaxation and dephasing times, in the range of 780–1800 ms [43,44]. Recently, a T_2^* time of 50 s has been reported [42]. IT_S accounts for gate times observed in the current experiments and longer preparation and measurement times [43–45]. IT_F accounts for gate, preparation, and measurement times of a proposed scalable ion trap quantum computer model [14]. It assumes that all gate operations are within one elementary logic unit (ELU) with 10–100 qubits arranged linearly. ELUs are connected to each other using photonic quantum channels to achieve modular scalability.

VI. RESULTS AND ANALYSIS

In this section we analyze numerical Monte Carlo simulations of the distance-3 surface code layouts under the multiparameter noise models. We first determine the distance-3 layout that admits the highest pseudothreshold under depolarizing noise. We then study the performance of the preferred layout under several realistic noise models. In particular, for the

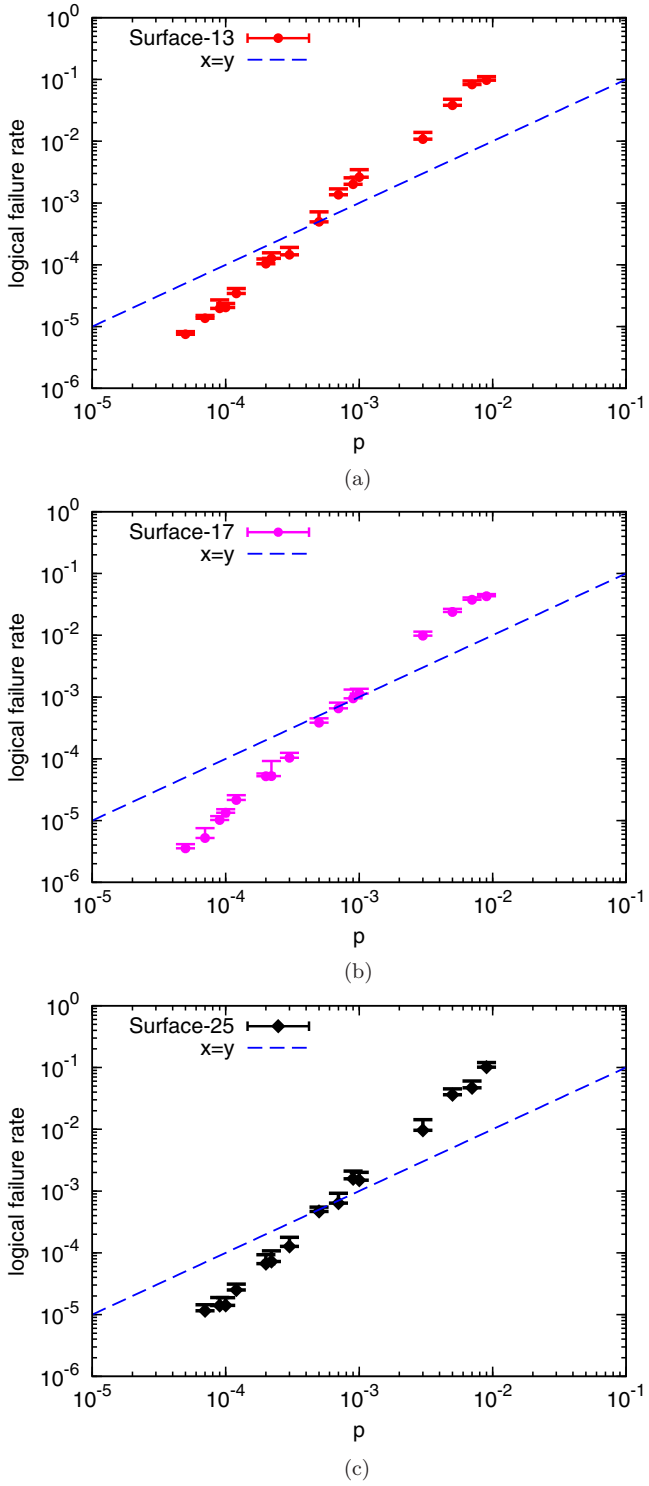


FIG. 9. (Color online) Location error rate p versus logical error rate $P_{1,X}$ for a logical $|1_L\rangle$ state encoded in (a) surface-13, (b) surface-17, and (c) surface-25 under symmetric depolarizing noise.

six architecture settings we compare the accuracy of the approximate amplitude and phase damping channel, which can be efficiently simulated, to the amplitude and phase damping channel, which requires universal simulation. In each plot, error bars indicate the upper bound statistical significance using the standard deviation.

TABLE IV. Comparison of thresholds and pseudothresholds for the surface code under symmetric depolarizing noise.

Code	Threshold	$P^{\text{th}}_{1,X}$	$P^{\text{th}}_{3,X}$
Surface-13	–	3.0×10^{-4}	1.2×10^{-4}
Surface-17	–	8.0×10^{-4}	2.0×10^{-4}
Surface-25	–	5.0×10^{-4}	1.4×10^{-4}
Wang [9]	1×10^{-2}	–	–
Fowler [8,46]	9×10^{-3}	$\sim2\times10^{-3}$	–

A. Depolarizing noise

We begin by calculating the symmetric depolarizing noise pseudothreshold for each distance-3 layout. In this model, each location fails with probability p. For single-qubit locations, $P_I = 1-p$ and $P_X = P_Y = P_Z = p/3$. For two-qubit locations, $P_{I,I} = 1-p$ and $P_{\{I,X,Y,Z\},\{I,X,Y,Z\}} = p/15$. Since the circuits and round times differ, we expect the pseudothreshold to vary for each layout.

Figure 9 plots the location error rate p versus the logical X error rate per round $P_{1,X}$ for surface-13, -17, and -25, where each layout encodes a logical $|1_L\rangle$ state and we check for a logical bit-flip X_L. Each point represents between 10 and 200 independent simulation runs.

The corresponding pseudothresholds calculated per round ($P^{\text{th}}_{1,X}$) and per window ($P^{\text{th}}_{3,X}$) are listed in Table IV. We find that surface-13 exhibits slightly lower pseudothresholds due to its higher circuit depth. Similarly, surface-25 requires more data qubits and syndrome measurements, thus exhibiting a small decrease in its pseudothreshold compared to surface-17.

Table IV also contains the pseudothreshold and threshold calculated by Fowler *et al.* for surface-25 [8,46]. Our surface-25 per-round pseudothreshold is slightly lower. Our simulations use a constant window size (Sec. V) to set the volume history, while Fowler *et al.* use a volume including a history of rounds limited only by the data available. They perform minimum-weight matching continuously, round by round, based on a large volume, while we perform correction based on our lookup table and three rounds of history in the volume. We use a static, small window in order to mimic future experimental implementations, which are likely to be limited to a small number of rounds and to restricted cold classical processing.

We also calculate the logical Z error rate $P_{L,Z}$ for each layout by encoding a logical $|+_L\rangle$ state and checking for a logical phase-flip Z_L. Figure 10 plots the location error rate p versus $P_{1,\{X,Z\}}$ for surface-17. It is apparent from the plot that the pseudothresholds $P^{\text{th}}_{1,X}$ and $P^{\text{th}}_{1,Z}$ are comparable. We find similar results for surface-13 and surface-25.

Based on these results, we conclude that surface-17 is the preferable layout. It requires roughly half the depth of surface-13 and significantly fewer qubits and gates than surface-25. In addition, surface-17 exhibits slightly higher pseudothresholds than the other layouts. For the remaining experiments, we thus perform all simulations based on the surface-17 layout.

B. Amplitude and phase damping vs Pauli twirling

In this section, we compare the accuracy of the approximate amplitude and phase damping channel using Pauli twirling to the amplitude and phase damping channel. We first verify

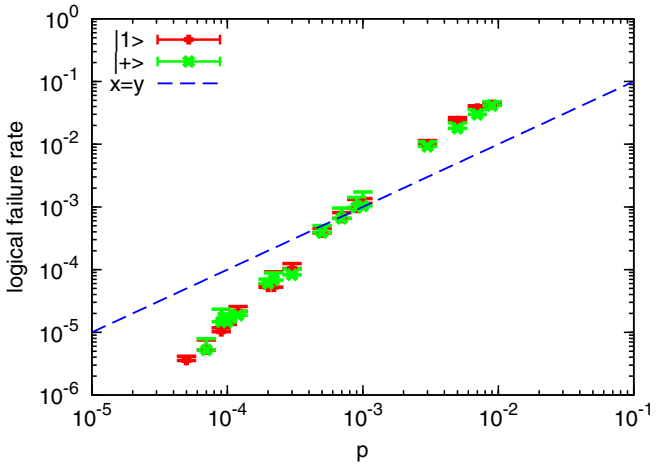


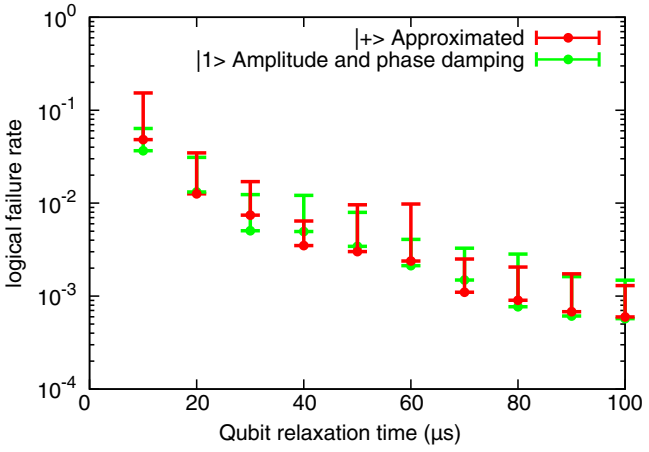
FIG. 10. (Color online) Surface-17 logical X error rate $P_{1,X}$ (red; qubit encoded in $|1_L\rangle$) and logical Z error rate $P_{1,Z}$ (green; qubit encoded $|+_L\rangle$) under depolarizing noise.

that our logical Z and X error rates per round for the Pauli-twirl approximation of surface-17 align with those reported in [22]. For $T_1 = 10$ μs, we find $P_{Z,1} = 4.27\times10^{-3}$ and $P_{X,1} = 4.41\times10^{-3}$. These results are very similar to those in [22]; small differences are expected since surface-25 is used in [22].

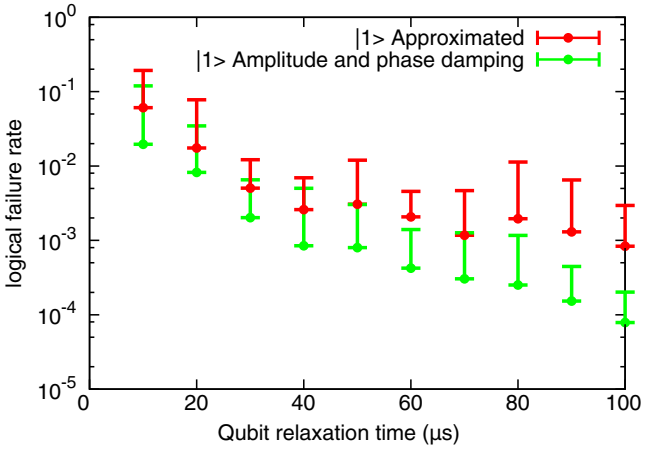
We then calculate the logical Z error rate per window, $P_{3,Z}$, for a qubit in the encoded $|+_L\rangle$ state in surface-17 for both channels for SC_H. Figure 11(a) plots T_1 versus $P_{3,Z}$ for approximate (red) and amplitude and phase damping (green). We see that the approximate channel using Pauli twirling results in a logical Z error rate that closely matches that of the actual channel.

We also calculate the logical X error rate per window, $P_{3,Z}$, for a qubit in the encoded $|1_L\rangle$ state in surface-17 for both channels under SC_H, plotted in Fig. 11(b). We find that the approximation channel results in much higher logical X error rates, in particular, as the qubit relaxation time T_1 increases. Pauli twirling results in a pessimistic estimate of the error rate, indicating that the threshold under decoherence may be significantly better than previously calculated with this technique.

Since the Pauli-twirl approximation aligns well for phase-flip errors, we further compare its performance on bit-flip errors. Figure 12 plots T_1 time (μs) versus memory duration (μs) versus the logical X failure rate $P_{3,X}$ of a qubit encoded in $|1_L\rangle$ in surface-17 for the SC_H setting under the Pauli-twirl approximation [Fig. 12(a)] and amplitude and phase damping [Fig. 12(b)]. On the left, the blue surface represents the amplitude damping probability of an unencoded qubit in $|1\rangle$ for a given T_1 time and memory duration. Since the qubit is in $|1\rangle$, phase damping does not apply. The yellow surface represents the logical error rate $P_{3,X}$ of an encoded qubit for a given T_1 time and surface code round time (see Table III). The orange surface indicates the upper error bar of $P_{3,X}$. For the yellow and orange surfaces, the encoded qubit undergoes the surface code three-round window time. For the blue surface, the unencoded qubit undergoes the given memory duration.



(a)



(b)

FIG. 11. (Color online) Comparison of the logical error rate of a qubit encoded in surface-17 under amplitude and phase damping (green) and the Pauli-twirl approximation (red) for the SC_H setting. (a) Logical Z error rate $P_{3,Z}$ (in the logical $|+_L\rangle$ state); (b) logical X error rate $P_{3,X}$ (in the logical $|1_L\rangle$ state).

The region where the blue surface lies above the orange and yellow surfaces represents the regime where surface-17 encoding improves the logical error rate of the qubit (similar to being below pseudothreshold). The region is larger in Fig. 12(b) than Fig. 12(a), indicating that Pauli twirling results in a pessimistic estimate of the logical error rate.

The 2D plots on the right are a view from the $+z$ axis. The blue and red regions indicate T_1 times (x axis) for which encoding a qubit in $|1_L\rangle$ in surface-17 reduces or increases, respectively, the logical error rate compared to an unencoded $|1\rangle$ qubit in memory for a given duration (y axis). The purple region indicates the upper error bar of $P_{3,X}$ where the orange and blue surfaces cross in the 3D plots. Surface-17 again demonstrates superior performance under amplitude and phase damping compared to Pauli twirling. For example, for $T_1 = 1$ μs, memory durations above 150 ns result in lower logical error rates for an encoded qubit than an unencoded qubit, while the Pauli-twirl approximation lowers error rates only for memory durations longer than 350 ns. At $T_1 = 50$ μs, memory durations of 20 ns result in lower logical error rates,

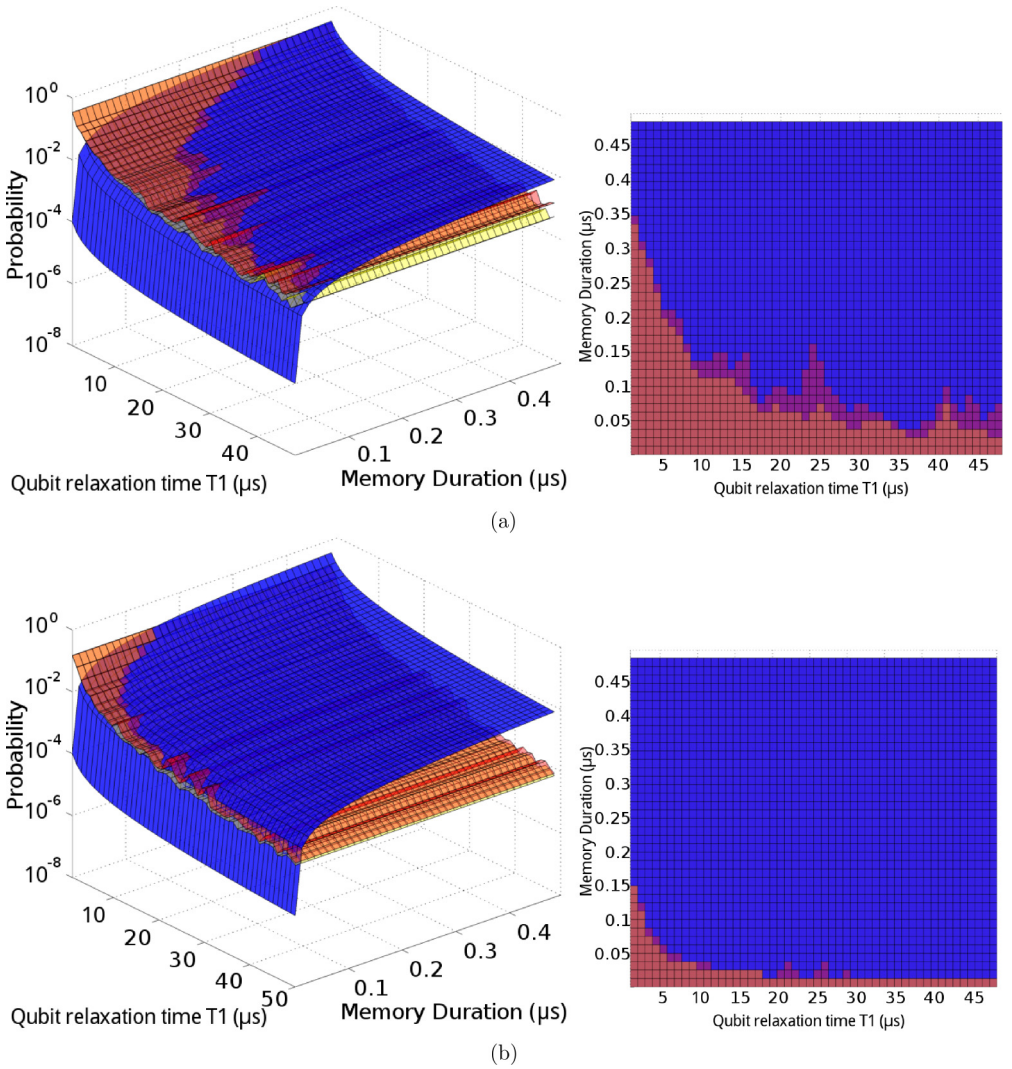


FIG. 12. (Color online) Left: 3D plots of T_1 time (μs) vs memory duration (ns) vs $P_{3,X}$ for the SC_H setting under (a) the Pauli-twirl approximation and (b) amplitude and phase damping. The blue surface represents the amplitude damping probability at a given T_1 and memory duration (unencoded qubit). The yellow surface is the simulated logical error rate given T_1 for a qubit encoded in the $|1_L\rangle$ state in surface-17. The orange surface indicates the upper error bar on $P_{3,X}$. Right: 2D plots from the $+z$ axis. The blue and red regions indicate a range of T_1 times (x axis) for which encoding a qubit in the $|1_L\rangle$ state in surface-17 reduces or increases, respectively, the logical error rate compared to an unencoded $|1\rangle$ qubit in memory for a range of durations (y axis).

while Pauli twirling indicates lower rates at memory durations longer than 30–70 ns.

C. Amplitude and phase damping

Figure 13 shows the same 2D plots for surface-17 for all six architecture settings under amplitude and phase damping. For each architecture, the y axis ranges from 0 μs to the time per surface code window. In all plots, we see that as T_1 increases, encoding improves the logical error rate for a larger range of memory durations. This behavior is expected since the amplitude damping probability monotonically increases with memory duration.

In Fig. 13(a), at $T_1 = 1$ μs we observe that for the SC_S parameters, encoding does not improve the logical error rate for any plotted memory duration. However, with 10 times faster gates (SC_F), we see performance improvement, as shown in Fig. 13(b). At $T_1 = 1$ μs, encoding provides a better logical error rate than an unencoded qubit in memory for at least 8 μs. At $T_1 = 10$ μs, SC_S shows no improvement with encoding, while SC_F exhibits improvements for memories of 2 μs or longer.

The SC_H setting accounts for 100 times faster preparation and measurement than SC_F and roughly 10 times faster gates. The faster times lead to significantly better performance under encoding. For example, at $T_1 = 1$ μs and 10 μs in Fig. 13(d), the logical error rate decreases due to encoding for memory durations longer than 150 and 20 ns, respectively.

Comparing Figs. 13(c) and 13(d), we find that CNOT time strongly influences performance. A CNOT gate is four times longer in SC_D than SC_H. The longer two-qubit gate time is reflected in the poorer performance of surface-17 under SC_D parameters. At $T_1 = 1$ μs, SC_D only indicates logical error rate reduction due to encoding at memory

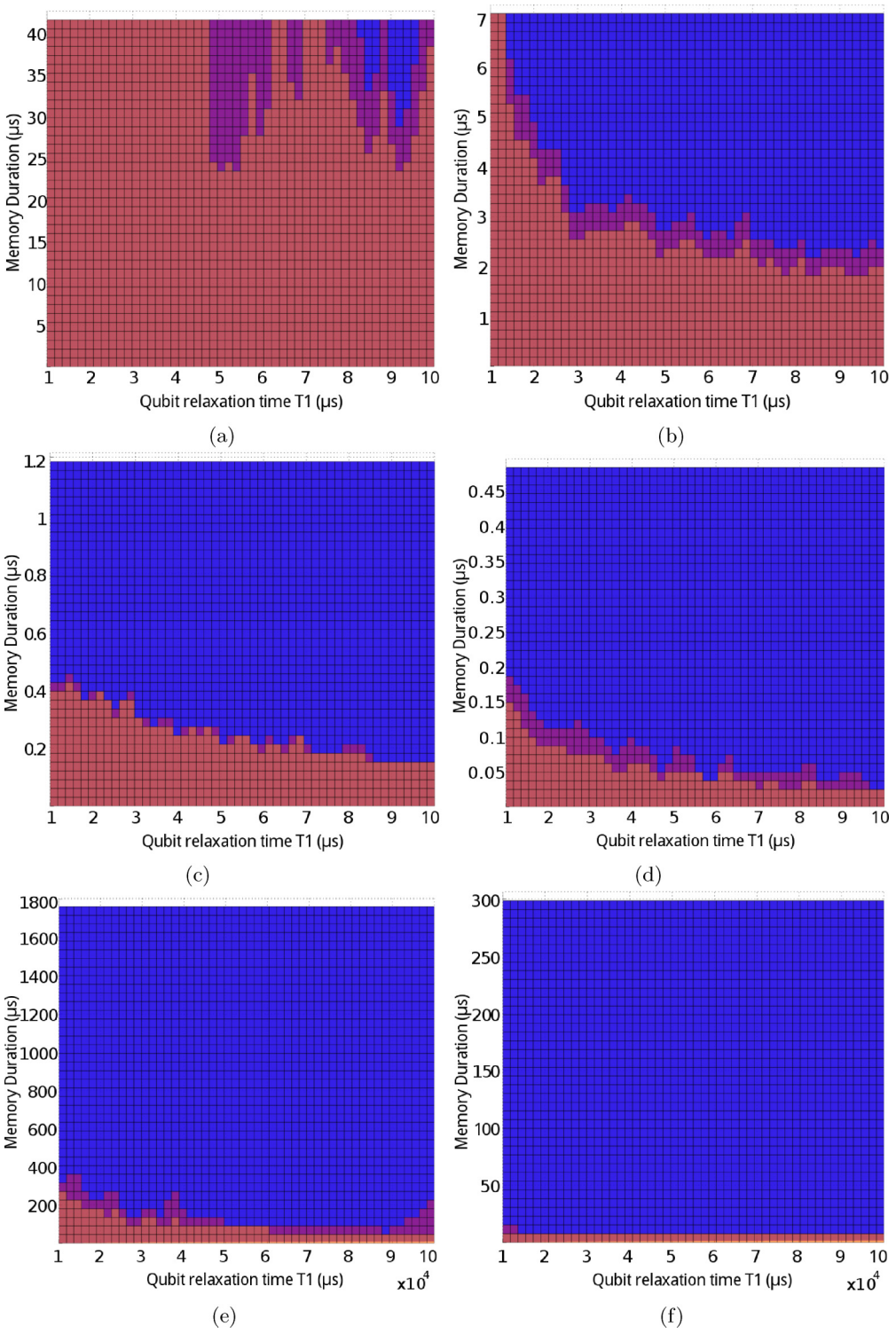


FIG. 13. (Color online) Plots of T_1 time (μs) versus memory duration (μs) for six architecture settings under amplitude and phase damping. The blue and red regions indicate a range of T_1 times (x axis) for which encoding a qubit in $|1_L\rangle$ in surface-17 reduces or increases, respectively, the logical error rate compared to an unencoded $|1\rangle$ qubit in memory for a range of durations (y axis). (c) SC_D; (d) SC_H.

durations roughly three times longer than those required for SC_H.

Figures 13(e) and 13(f) show similar results for the ion trap settings. While IT_F assumes 10 times faster CNOT gates, both IT_S and IT_F yield lower logical error rates upon encoding for a range of T_1 times. IT_S results in improvements for memory durations longer than 300–400 μs, while IT_F results in improvements for memory durations above around 15 μs.

In Fig. 14, we plot qubit relaxation time T_1 (μs) versus logical error rate $P_{3,X}$ (red) or amplitude damping probability (blue) for the six architecture settings [analogous to Fig. 11(b)]. All plots assume a three-round memory duration. From the

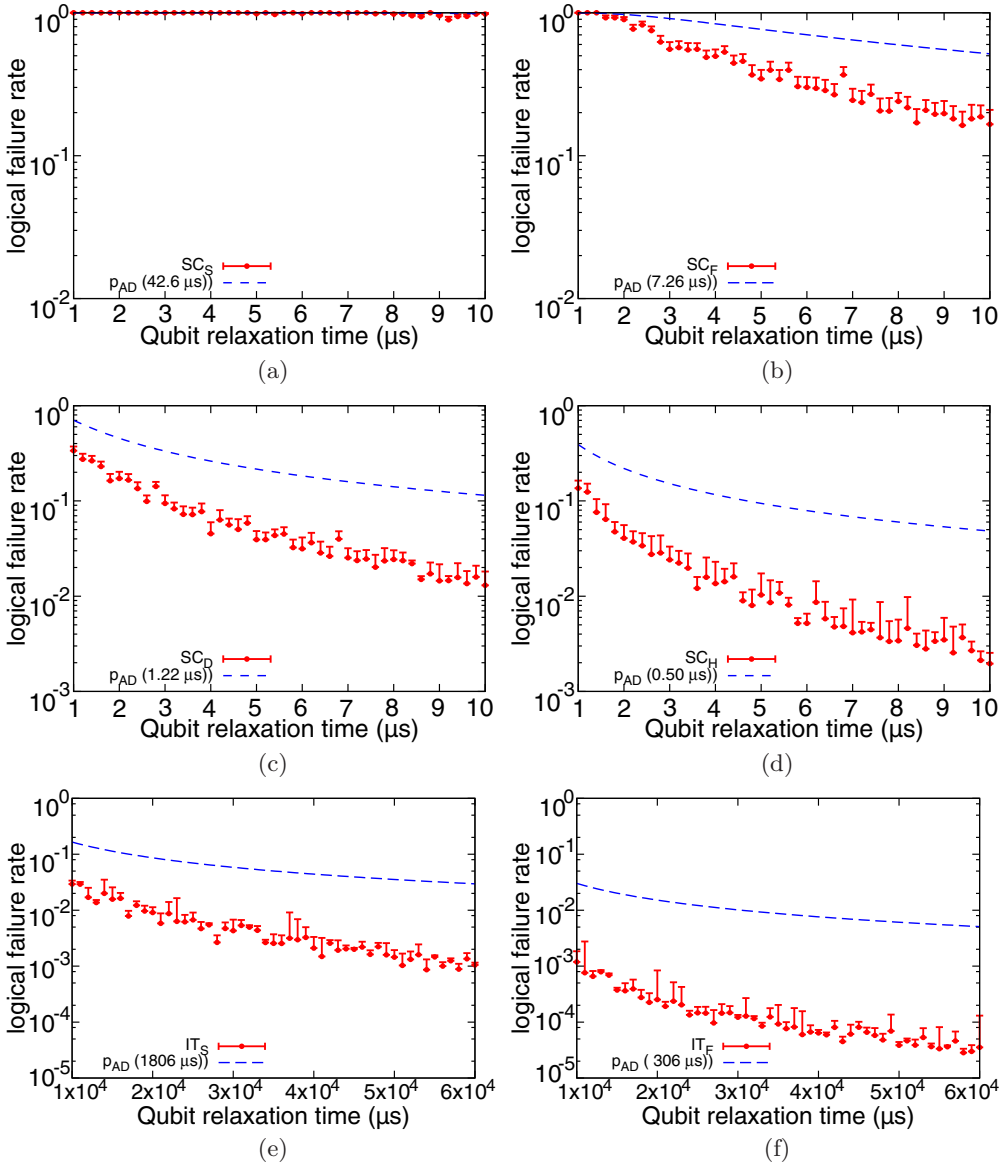


FIG. 14. (Color online) Plots of T_1 time (μs) versus the logical X error rate $P_{3,X}$ for six architecture settings for a qubit encoded in $|1_L\rangle$ in surface-17 subject to amplitude and phase damping [(red) points] and an unencoded $|1\rangle$ qubit subject to amplitude damping [dashed (blue) line] for the duration of three rounds of the surface code.

plots, the logical error rate for each architecture for a given T_1 time can be extracted. As gate times and T_1 times improve, the logical error rate decreases. An order of magnitude improvement in logical error rate can be obtained, for example, in improving gates time from those of SC_F to those of SC_H.

The plots also indicate that near-term experiments may be able to detect improved logical error rates due to encoding, providing experimental evidence of surface code error correction. For example, for $T_1 = 1$ μs, no difference in the logical error rate can be detected between an encoded and an unencoded qubit given settings SC_S and SC_F. However, SC_F exhibits differences of the order of one magnitude at T_1 times longer than 30 μs. Both SC_D and SC_H settings indicate significant differences in the logical error rate of an encoded versus an unencoded qubit. In the case of both ion trap settings, a difference in logical error rate can be detected starting at $T_1 = 10$ μs.

Figure 15 contains plots for T_1 times up to 50 μs for the four superconducting architectures. The left column contains 3D plots of T_1 time (μs) versus memory duration (μs) versus logical failure rate. The middle column contains 2D plots of T_1 time (μs) versus memory duration (μs). The right column contains 2D plots of T_1 time (μs) versus logical failure rate.

We conclude that gate durations in the SC_S setting are too slow for surface-17 to significantly decrease the logical error rate given realistic T_1 times [Fig. 15(a)]. However, given gate durations between the SC_F and the SC_H settings and current T_1 times of 20–40 μs, encoding a qubit in surface-17 results in significantly improved error rates over an unencoded qubit [Figs. 15(b)–15(d)]. For both superconductor and ion trap architectures, near-term experimental implementations could

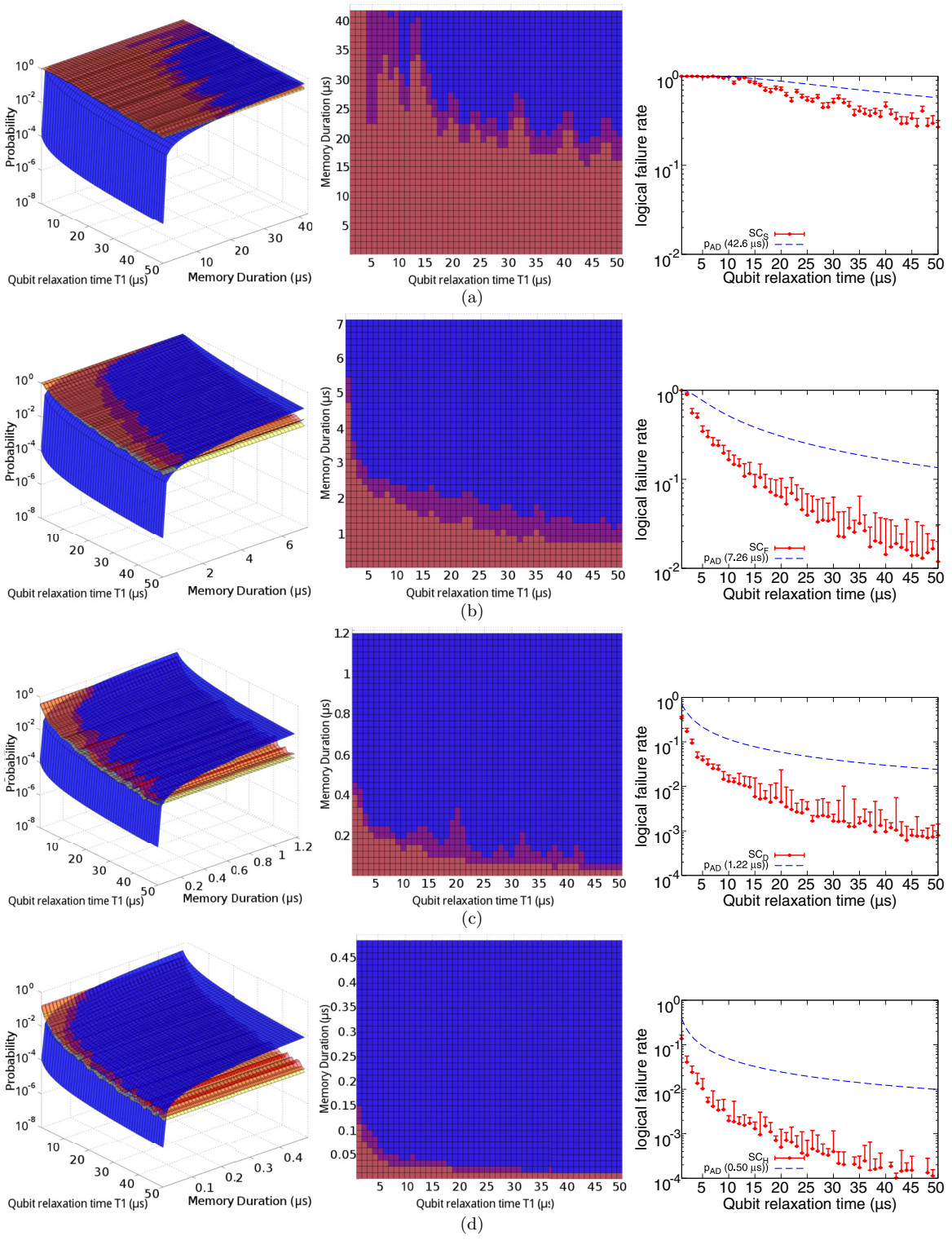


FIG. 15. (Color online) Plots of four superconductor architecture settings under amplitude and phase damping for T_1 times up to 50 μs. Left: Plots of T_1 time (μs) versus memory duration (μs) versus logical failure rate. Middle: Plots of T_1 time (μs) versus memory duration (μs). Right: Plots of T_1 time (μs) versus logical failure rate. The blue and red regions indicate a range of T_1 times (x axis) for which encoding a qubit in $|1_L\rangle$ in surface-17 reduces or increases, respectively, the logical error rate compared to an unencoded $|1\rangle$ qubit in memory for a range of durations (y axis).

demonstrate surface code error correction of a single logical qubit and measure signficiant improvements in the logical error rate. We find that the previous estimates of 2.6–2.8 μs T_1 times [22] to achieve improved logical error rates are too high, and in fact at a T_1 time of only 1 μs, the logical error rate can be improved using surface-17.

VII. CONCLUSIONS AND FUTURE WORK

We have presented circuits and a decoder for surface code error correction targeted at near-term experimental settings under realistic noise models. Surface-17 demonstrates the highest pseudothreshold under depolarizing noise. Its pseudothreshold under amplitude and phase damping is higher than under a Pauli-twirl approximation, indicating that approximation can be used to efficiently lower bound higher-distance surface code pseudothresholds.

We have outlined the experimental settings necessary to demonstrate a working surface code assuming amplitude and phase damping noise. Overall, we find that with near-term experimental settings for both superconductor and ion trap architectures, a qubit encoded in surface-17 will exhibit *significantly lower logical error rates* compared to an unencoded qubit. The result indicates that, surprisingly, even with a *distance-3* code, significant improvement, of 1–2 orders of magnitude, in the qubit lifetime can be detected. Experimental implementations could demonstrate surface code error correction of a single logical qubit and measure the difference in error rate. This would result in the first experimental demonstration and verification of a working surface code.

Methods of approximating decoherence using Clifford gates have recently been shown to be more accurate than Pauli twirling [23,25]. However, studies have only been conducted at the gate operation level, as opposed to the circuit level of a given code. A direction for future work is to simulate these noise models in surface-17 to compare to amplitude and phase damping. Another direction is to determine the performance of surface-17 under leakage and correlated qubit errors. Finally, development and simulation of realistic noise models for specific architectures will be important for guiding experimental surface code implementations.

ACKNOWLEDGMENTS

We are especially grateful to Dave Wecker for helping with the surface code and noise model implementations in LIQ$Ui|\rangle$. We thank Ken Brown, Leo DiCarlo, and Andrew Landahl for valuable discussions. We also thank Leo Kouwenhoven, Lieven Vandersypen, Austin Fowler, and Blake Johnson for useful feedback.

[1] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[2] D. A. Lidar and T. A. Brun, *Quantum Error Correction* (Cambridge University Press, Cambridge, 2013).
[3] S. B. Bravyi and A. Y. Kitaev, Quantum Comp. Comput. **2**, 43 (2001).
[4] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[5] A. Y. Kitaev, Ann. Phys. **303**, 2 (2003).
[6] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).
[7] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).
[8] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Phys. Rev. Lett. **108**, 180501 (2012).
[9] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Phys. Rev. A **83**, 020302 (2011).
[10] A. M. Stephens, Phys. Rev. A **89**, 022321 (2014).
[11] F. Helmer, M. Mariantoni, A. G. Fowler, J. von Delft, E. Solano, and F. Marquardt, Europhys. Lett. **85**, 50007 (2009).
[12] D. P. DiVincenzo, Phys. Scripta **T137**, 014020 (2009).
[13] D. Kielpinski, C. R. Monroe, and D. J. Wineland, Nature **417**, 709 (2002).
[14] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, Phys. Rev. A **89**, 022317 (2014).
[15] J. T. Merrill, C. Volin, D. Landgren, J. M. Amini, K. Wright, S. C. Doret, C.-S. Pai, H. Hayden, T. Killian, D. Faircloth, K. R. Brown, A. W. Harter, and R. E. Slusher, New J. Phys. **13**, 103005 (2011).
[16] K. Wright, J. M. Amini, D. L. Faircloth, C. Volin, S. C. Doret, H. Hayden, C.-S. Pai, D. W. Landgren, D. Denison, T. Killian, R. E. Slusher, and A. W. Harter, New J. Phys. **15**, 033004 (2013).
[17] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, Nature **508**, 500 (2014).
[18] D. Gottesman, in *Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics (GROUP22)*, edited by S. P. Corney, R. Delbourgo, and P. D. Jarvis (International Press, New York, 1999), pp. 32–43.
[19] S. Aaronson and D. Gottesman, Phys. Rev. A **70**, 052328 (2004).
[20] M. Silva, E. Magesan, D. W. Kribs, and J. Emerson, Phys. Rev. A **78**, 012347 (2008).
[21] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
[22] J. Ghosh, A. G. Fowler, and M. R. Geller, Phys. Rev. A **86**, 062318 (2012).
[23] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, Phys. Rev. A **87**, 012324 (2013).
[24] D. Puzzuoli, C. Granade, H. Haas, B. Criger, E. Magesan, and D. G. Cory, Phys. Rev. A **89**, 022306 (2014).
[25] M. Gutiérrez, L. Svec, A. Vargo, and K. R. Brown, Phys. Rev. A **87**, 030302 (2013).
[26] A. G. Fowler, Phys. Rev. A **88**, 042308 (2013).
[27] A. G. Fowler and J. M. Martinis, Phys. Rev. A **89**, 032316 (2014).
[28] P. Jouzdani, E. Novais, I. S. Tupitsyn, and E. R. Mucciolo, Phys. Rev. A **90**, 042315 (2014).
[29] H. Bombin and M. A. Martin-Delgado, Phys. Rev. A **76**, 012305 (2007).
[30] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, New J. Phys. **14**, 123011 (2012).
[31] D. Wecker and K. M. Svore, arXiv:1402.4467 [quant-ph].
[32] J. Edmonds, Canad. J. Math. **17**, 449 (1965).
[33] J. Edmonds, J. Res. Nat. Bur. Stand. Sect. B **69B**, 67 (1965).
[34] A. G. Fowler, Quantum Inf. Comput. **15**, 0145-0158 (2015).

[35] A. G. Fowler, arXiv:1310.0863v1 [quant-ph].
[36] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, Proc. R. Soc. A **465**, 1645 (2009).
[37] K. M. Svore, A. W. Cross, I. L. Chuang, and A. V. Aho, Quantum Inf. Comp. **6**, 193 (2006).
[38] K. M. Svore, B. M. Terhal, and D. P. DiVincenzo, Phys. Rev. A **72**, 022317 (2005).
[39] L. DiCarlo, M. D. Reed, L. Sun, B. R. Johnson, J. Chow, J. M. Gambetta, L. Frunzio, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf, Nature **467**, 574 (2010).
[40] J. M. Chow, L. DiCarlo, J. M. Gambetta, F. Motzoi, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. A **82**, 040305(R) (2010).
[41] J. M. Chow, J. M. Gambetta, E. Magesan, D. W. Abraham, A. W. Cross, B. R. Johnson, N. A. Masluk, C. A. Ryan, J. A. Smolin, S. J. Srinivasan, and M. Steffen, Nat. Commun. **5**, 4015 (2014).
[42] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, Phys. Rev. Lett. **113**, 220501 (2014).
[43] H. Häffner, C. Roos, and R. Blatt, Phys. Rep. **469**, 155 (2008).
[44] R. Noek, G. Vrijsen, D. Gaultney, E. Mount, T. Kim, P. Maunz, and J. Kim, Opt. Lett. **38**, 4735 (2013).
[45] K. R. Brown, private communication (2013).
[46] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Phys. Rev. A **86**, 042313 (2012).