Engineering autonomous error correction in stabilizer codes at finite temperature

C. Daniel Freeman,[1,2,*] C. M. Herdman,[3,4,5] and K. B. Whaley[1,6]

[1]*Berkeley Quantum Information & Computation Center, University of California, Berkeley, California 94720, USA*
[2]*Department of Physics, University of California, Berkeley, California 94720, USA*
[3]*Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[4]*Department of Chemistry, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[5]*Department of Physics & Astronomy, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[6]*Department of Chemistry, University of California, Berkeley, California 94720, USA*

We present an error-correcting protocol that enhances the lifetime of stabilizer code–based qubits which are susceptible to the creation of pairs of localized defects (due to stringlike error operators) at finite temperature, such as the toric code. The primary tool employed is periodic application of a local, unitary operator, which exchanges defects and thereby translates localized excitations. Crucially, the protocol does not require any measurements of stabilizer operators and therefore can be used to enhance the lifetime of a qubit in the absence of such experimental resources.

I. INTRODUCTION

For the past two decades, significant effort has gone into devising schemes for encoding quantum information in reliable and retrievable forms. Stabilizer error–correcting codes are thought to be an effective strategy for performing this encoding, because they allow an efficient means of detecting and correcting errors. Among these, topological stabilizer codes (or topological quantum memories) are particularly promising strategies for storing quantum information due to their intrinsic robustness to errors at zero temperature, their ability to be efficiently implemented via a local Hamiltonian [1], as well as the existence of efficient strategies for performing error detection and correction [2–4], which have been demonstrated in recent experiments [5]. Several exhaustive studies have been performed on calculating error thresholds for these topological codes, such as Kitaev's toric code, both in the presence and absence of error-correcting protocols [2,3,6–8].

However, these topological codes are well known to be poor *passive* quantum memories at finite temperature in less than four spatial dimensions [9–18] (for a thorough review, see Ref. [19]). For physically realistic coupling to an environment, local noise processes drive the creation of localized defects. In the absence of an error-correcting protocol, the propagation of these defects can then lead to decoherence of the memory. For the case of the toric code, these error strings are particularly pathological and cause the maximum lifetime of an encoded qubit to decay exponentially with temperature with a timescale independent of system size [12]. While experimentally intractable, fault tolerant topological quantum memories are nevertheless known to exist in four and six dimensions [20,21].

On the other hand, a variety of active error correction protocols exist for efficient detection and correction of errors. As long as error rates and the temperature are low or, alternatively, as long as detection and correction are fast enough, the lifetime of these codes can in principle be extended indefinitely. But these decoding strategies implicitly

rely on resources that may not always be available or efficiently physically implementable. For example, performing a measurement on a quantum system requires a fresh ancilla qubit for each measurement. Thus, continuously measuring any quantum system requires continuously recycling ancilla qubits for measurement—a procedure that will necessarily be rate limiting for near term quantum architectures [22].

An error-correcting strategy for topological codes *without* the need for stabilizer measurement is desirable. At face value, ignoring the power of the stabilizer group will assuredly provide a suboptimal strategy. But given limited resources and rates of measurement, it is worthwhile to understand the limits of strategies that do not require syndrome measurements and to determine if such strategies can augment known decoding schemes.

We provide here a new protocol for error correction of pairs of localized defects, using a periodic sequence of unitary operators in the presence of thermal dissipation. The required thermal dissipation may be due to an interaction with a physical bath or engineered via a dissipative protocol, but our protocol does not require explicit measurements of stabilizer operators, thus significantly reducing the required physical resources. This pattern of operators is designed to encourage defects in the system to dissipate more quickly. In this work, we explicitly treat the theory for the one-dimensional (1D) Ising model at finite temperature and describe how this approach may be extended to other stabilizer codes, such as the toric code. While dissipative protocols have previously been employed to generate Hamiltonians [23,24], to prepare encoded ground states [25], to mediate long-range interactions [26–28], and to "trap" defects [4,29], a measurement-free protocol that explicitly targets stringlike error processes has not been proposed to date. While it does not completely eliminate errors, the protocol presented here provides a significant enhancement of the lifetime of a finite-size system.

It is known that stabilizer Hamiltonians at finite temperature in dimension less than three have a system-size–independent upper bound to their lifetime [12,14,15,30–34]. These "no-go" theorems necessarily limit the extent to which the method proposed here can be carried out. In fact, a size-independent constant enhancement of a system's lifetime may be the best

———————
*daniel.freeman@berkeley.edu

one can get with a purely local unitary protocol like the one presented here. Thus, this scheme, by itself, will not generate a topologically protected quantum memory at finite temperature for one or two dimensions. It is nonetheless worthwhile to understand how far purely local protocols can be pushed, because a large constant increase in the lifetime of a quantum architecture could mean the difference between a physically realistic architecture that can be fault tolerantly operated versus one that cannot, as discussed in Sec. V C.

The rest of the paper is structured as follows: in Sec. II, we briefly review the theory of stabilizer codes, discuss where our strategy falls on the continuum of different active and passive strategies used in the literature for implementing these codes, and sketch how they can be modeled at finite temperature. In Sec. III, we describe how the 1D Ising model can be treated as a stabilizer code and discuss the low-temperature dynamics of the model. In Sec. IV, we construct our autonomous protocol, built out of local unitary operators, and discuss the scaling behavior of the protocol. We also demonstrate evidence for the enhancement of the lifetime of the 1D Ising model. In Sec. V, we sketch how our protocol generalizes to higher dimensions and to other stabilizer codes, including the toric code.

II. STABILIZER CODES

A. Definitions

In this section, we briefly review the theory of stabilizer error correcting codes [35,36]. Given n qubits, a collection of operators S_i, and 2^k states $|\psi\rangle_i, i = 1,...,2^k$, which span a subspace in which k encoded qubits are defined as

$$S_i|\psi\rangle_i = +1|\psi\rangle_i, \tag{1}$$

$$[S_i, S_j] = 0, \tag{2}$$

for all i, j. Furthermore, suppose there are m *error* operators $E_j, j = 1,...,m$, and that for each of them, there exists some not necessarily unique operator S_j, such that

$$\{E_i, S_j\} = 0. \tag{3}$$

Stabilizer codes are those collections of states $|\psi\rangle_i$ and operators $\{S_j\}$ which satisfy the above conditions for error operators belonging to some subset of the Pauli group—tensor products of Pauli operators with the identity.

For example, given three qubits, let $|\psi\rangle_1 = |\uparrow\uparrow\uparrow\rangle$ and $|\psi\rangle_2 = |\downarrow\downarrow\downarrow\rangle$. Then the set of operators S_i satisfying Eqs. (1) and (2) is $\{\sigma_z\sigma_z I, I\sigma_z\sigma_z\}$. One can easily determine that the set of error operators E_i corresponding to these two *stabilizer* operators is $\{\sigma_x I I, I\sigma_x I, I I\sigma_x, \sigma_x\sigma_x I, \sigma_x I\sigma_x, I\sigma_x\sigma_x\}$.

More transparently, this three-qubit stabilizer code encodes two protected states. If some noise source were to apply any single qubit σ_x operator, or any two-qubit $\sigma_x^i\sigma_x^j$ operator, measurement of the set of stabilizer operators would indicate the presence of an error. Furthermore, the code can actually detect *and* correct single σ_x errors. For example, a measurement result of $-1, +1$ of the stabilizers $\sigma_z\sigma_z I$ and $I\sigma_z\sigma_z$, respectively, indicates either an error on the first qubit or two errors on the latter two qubits. In a sufficiently noisy environment, these two errors would be indistinguishable—i.e., degenerate—but for many noise models, the single error situation is much more likely,

so that a single σ_x operator applied to the first qubit will more often than not return the qubit back into the protected subspace.

B. Active state preservation versus dissipative Hamiltonian engineering

Here we will refine our discussion by broadly classifying error correcting approaches into (1) active state preservation strategies and (2) Hamiltonian engineering strategies.

The target of both strategies is the same: the preservation of an encoded stabilizer state. In state preservation, a stabilizer-encoded state is preserved by the application of a sequence of unitaries. However, ignoring noise sources, the natural Hamiltonian that describes the system is $H = 0$. The target of such a strategy is preservation of the stabilizer state itself. Implicitly, some sort of active error measurement and correction needs to be performed once the state is initially prepared.

In contrast, in Hamiltonian engineering approaches [37], the encoded state is preserved by implementing a Hamiltonian on a set of qubits, which has a ground-state manifold composed of stabilizer-encoded states. The encoded state is then preserved by keeping the quantum system at a sufficiently low temperature to suppress errors.

Combinations of these strategies exist. For example, one could use a Hamiltonian engineering approach to generate a stabilizer encoded state and then immediately turn off the Hamiltonian once the desired encoded subspace was reached, preserving or preparing a particular state at further times with active error correction or other control fields. Alternatively, one could use Hamiltonian engineering to preserve the state in the encoded subspace and then use a combination of dissipation with an additional protocol to detect or correct errors. We shall focus here on this latter strategy. Specifically, we will be concerned with systems being dissipatively driven toward the ground-state subspace of a stabilizer Hamiltonian, and we shall build an autonomous error correction protocol to mitigate the ways in which dissipation alone fails to keep the state in its initial encoded state. Furthermore, we are not explicitly concerned with the process of state creation, and we always assume that our systems are initialized to an encoded state.

C. Error-correcting master equation

To dissipatively generate a stabilizer code, one forms the system Hamiltonian as the negative sum of the stabilizer operators for the code of interest, i.e., $H = -\sum_i S_i$. This guarantees that the ground state of that Hamiltonian will be the encoded subspace. Furthermore, this ensures that configurations of the system with errors present are excited states.

To model dissipation in such a code, we employ here a Lindblad master equation. Without loss of generality, but to simplify analysis, we assume that the bath only operates on the system with purely local errors, and that these local errors correspond to the E_i [see Eq. (3)] of the stabilizer code of interest. For the ground state, the dynamics may be described by the Lindblad equation:

$$\dot{\rho} = \sum_\omega 2c_\omega\rho c_\omega^\dagger - c_\omega^\dagger c_\omega\rho - \rho c_\omega^\dagger c_\omega. \tag{4}$$
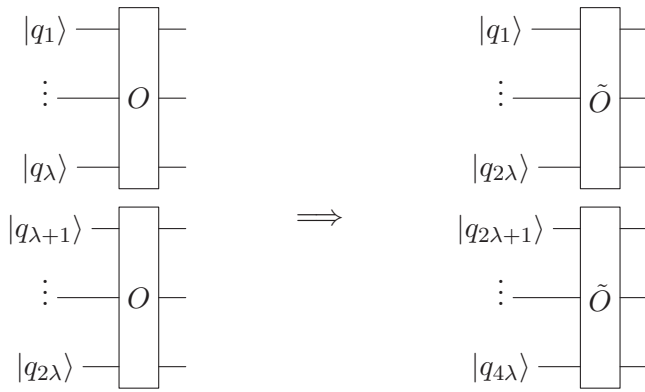
FIG. 1. A snapshot of a linear stabilizer code with detection and correction operators O. The optimal error correction operator has nontrivial system size scaling and in general depends on the particular code being used.

Here, ρ is the system density matrix for some candidate system and $\{c_\omega\} = \{\sqrt{\gamma_\omega} L_\omega\}$ are Lindblad operators arising from interactions with a bath, where the L_ω act on the system with characteristic rates γ_ω. Error processes can then be represented by products of the Lindblad operators: $\{c_\omega^1 c_\omega^2 \cdots c_\omega^n\}$.

A necessary condition for error correction to occur to nth order in the error processes is to apply the inverses of the error processes sufficiently rapidly. If we restrict ourselves to stabilizer codes on lattices, then the recipe for error correction is straightforward: measure the stabilizers of the code and apply correction operations conditioned on the results of the stabilizer measurements.

While it is in principle possible to measure all of the stabilizers of a given system simultaneously because they all commute, it will be convenient to decompose a given correction protocol into groups of terms involving operators only acting within a characteristic length scale λ. This is useful because it provides a natural scale for treating stabilizer codes with fixed resources, and it allows the interpretation of different protocols as the implementation of a certain kind of effective long-range interaction.

D. Error correction thresholds and scaling

Much of the power of stabilizer codes arises from the existence of error thresholds. Specifically, as the stabilizer code is made sufficiently large, the probability of returning to the original code state goes to 1, as long as measurement and correction cycles occur sufficiently fast compared to the threshold error rate. This can give rise to a competition between the resources necessary to perform error correction and detection for stabilizer codes involving many qubits, versus the scaling of the error rate of the code with system size.

For concreteness, consider a stabilizer code, with correction and detection steps idealized by operators O acting over a length scale λ as in Fig. 1. How these operators scale with system size depends crucially on the particular code and the specific error correction scheme. For example, it is known [38] that a purely local scheme with λ independent of system size

can achieve a threshold in the toric code via a cellular automata construction at the expense of a reduction in the error threshold. This scheme requires measurements and classical decoding. In contrast, the naive implementation of minimum weight perfect matching using Edmond's algorithm [39] to error correct the toric code requires measurements as well as global classical processing—that is, the lengthscale λ of the error correcting operator O, which implements Edmond's algorithm grows with system size, but with a larger resulting threshold. In this way, global resources can be converted to local resources at the expense of the magnitude of the fault tolerance threshold.

For protocols like the one presented here, which neither yield a threshold nor use measurements, the tradeoff between resources and system lifetime is less obvious. In anticipation of the following sections, one might expect a larger λ protocol would dominate over a smaller λ protocol, because the larger λ protocol corrects "more" errors. However, this is not generically the case, and the success of a given error-correcting procedure depends nontrivially on λ, L, and the fundamental noise rates of the system.

We spend a large fraction of the remainder of the manuscript elaborating on our protocol's scaling properties. Because we seek a protocol *without* measurements, we first motivate our construction with a summary of the conventional error correction operator for the 1D Ising model in Sec. IV A, discuss the most common finite temperature error processes in Sec. IV B, and then detail our measurement-free construction in Sec. IV C. Next, we explicitly derive how the lifetime of the 1D Ising model is improved by our measurement-free construction and connect the scaling of the enhanced lifetime to the scaling properties of the protocol with λ in Sec. IV D. Finally, we elaborate on how our protocol can be generalized to higher dimensions in Sec. V, including its expected scaling properties.

III. 1D ISING MODEL

A. 1D Ising model as a stabilizer code

The choice of three-qubit stabilizer code introduced in Sec. II A was deliberate, because it can naturally be extended and interpreted as the ground state of a 1D Ising model:

$$H_{\text{1D Ising}} = -\Delta \sum_{i=1}^{L} \sigma_z^i \sigma_z^{i+1}. \qquad (5)$$

The ground state subspace of this model is twofold degenerate and is composed of the states $|\uparrow \cdots \uparrow\rangle$ and $|\downarrow \cdots \downarrow\rangle$. These ground states are exactly the L-qubit analogues of the three-qubit code treated previously. These states are stabilized by the set of all adjacent pairwise σ_z operators $\{II \cdots I \sigma_z^i \sigma_z^{i+1} I \cdots I\}$, where i runs from 1 to L. These are precisely the operators appearing in the Hamiltonian of the 1D Ising model. For the remainder of our analysis, we assume without loss of generality that $\Delta = 1$.

Furthermore, σ_x errors are equivalent to excited states. In the simplest case, errors can be corrected by resorting to a simple majority rule—if most spins point in a particular direction, the correction protocol returns the state to the encoded ground state corresponding to that direction.

B. 1D Ising model at finite temperature

By coupling the Ising model to an external reservoir, one might hope to dissipatively drive the 1D Ising model into one of these encoded states. However, the 1D Ising model has no finite temperature ordered phase, so at all finite temperatures, the system evolves toward the unique thermal state. Furthermore, this timescale over which the system relaxes to a thermal state is known to be independent of the size of the chain, given modest bath assumptions [40]. Thus, dissipation by itself cannot protect the 1D Ising model, and an additional protocol needs to be implemented in order to correct thermal errors. While dissipation cannot protect the 1D Ising model at finite temperature, it is instructive to understand the details of how thermal fluctuations lead to instability in this simple case, because very similar processes are responsible for the instability of many other stabilizer codes at finite temperature. In previous work, we examined the dynamics of this model, as well as of the toric code, at finite temperature [18]. In particular, we identified a low temperature regime where the dynamics are well described by a simple random walk model. We briefly summarize the analysis below.

When studying the error dynamics, it is convenient to consider the dual lattice of the Ising model: we imagine a new 1D lattice with sites interleaved between the sites of Eq. (5) and associate auxiliary spin values b_i with them. The auxiliary site's spin values are uniquely determined by the products $b_i = \sigma_z^i \sigma_z^{i+1}$, where site b_i defined by this equation sits between site i and $i + 1$. We can identify these extra variables with *domain walls*. If adjacent spin variables disagree, then the auxiliary site sitting between them will have $b_i = -1$. If all but a contiguous block of spins disagree, then all auxiliary sites will have $b_i = 1$, except for those two sites that sit at the two boundaries of the contiguous blocks of spins. Describing the dynamics of these domain walls is equivalent to describing the spin dynamics, because if one knows all the auxiliary variables plus any single spin value, σ_z^i, one can reconstruct all of the remaining spin variables σ_z^j.

For simplicity, we assume a bath that operates on the system only by creating, destroying, or translating domain walls. Then, for sufficiently low temperatures, occasionally the bath will cause an adjacent domain wall pair to appear in the system. Bath fluctuations will cause this pair of domain walls to fluctuate across the system, effectively causing the domain walls to undergo a 1D random walk. When domain walls are adjacent, it is energetically favorable for them to be dissipated. If domain walls fuse *before* traversing the length of the system, the encoded state will be preserved. But if domain walls undergo a random walk such that one winds entirely around the system, this effectively performs an uncorrectable error on the encoded qubit because the system will have transitioned from one encoded ground state to the other encoded ground state [18].

C. Microscopic master equation

When the bath operates on the system with purely local spin-flip errors, which only create, destroy, and translate domain walls, the Lindblad operators are of the form

$$\{c_\omega\} = \{\sqrt{\gamma_0} T_b, \sqrt{\gamma_+} D_b^\dagger, \sqrt{\gamma_-} D_b\}. \tag{6}$$

When resolved in the Pauli basis and factored, these operators take a simple form:

$$D_b^\dagger = \tfrac{1}{4}(I\sigma_x I)(1 + I\sigma_z\sigma_z)(1 + \sigma_z\sigma_z I),$$

$$D_b = \tfrac{1}{4}(I\sigma_x I)(1 - I\sigma_z\sigma_z)(1 - \sigma_z\sigma_z I),$$

$$T_b = \tfrac{1}{4}(I\sigma_x I)(1 - I\sigma_z\sigma_z)(1 + \sigma_z\sigma_z I). \tag{7}$$

A short calculation verifies $\sum_i L_i L_i^\dagger = I$. Physically, these operators represent the creation of a domain wall pair at dual lattice sites b and $b + 1$ ($D_b^\dagger$), annihilation of a pair of domain walls at dual lattice sites b and $b + 1$ (D_b), and the translation of a domain wall from b to $b + 1$ (the adjoint translates $b + 1$ to b) (T_b). Intuitively, these operators are built from projectors ($1 \pm I\sigma_z\sigma_z$) which pick out those configurations relevant to each process. For example, the combination $(1 + I\sigma_z\sigma_z)(1 + \sigma_z\sigma_z I)$ projects onto the subspace with no domain walls between the first and second or second and third spins. The leftmost ($I\sigma_x I$) term then performs the spin flip, effectively creating a domain wall.

Additionally, these operators only connect diagonal elements of the density matrix to other diagonal elements. This reduces the time evolution of the diagonal matrix elements to a classical master equation:

$$\frac{dP_n}{dt} = \gamma_0 \sum_{n_0} \left(P_{n_0} - P_n \right) + \sum_{n_+} (\gamma_- P_{n_+} - \gamma_+ P_n)$$
$$+ \sum_{n_-} (\gamma_+ P_{n_-} - \gamma_- P_n), \tag{8}$$

where $P_n := \rho_{nn}$, and the indices n_0, n_+, and n_- run over those eigenstates connected to nth state by a single application of the operator T_b, $D_b^\dagger$, or D_b, respectively. The rates with which these operators are applied, i.e., γ_0, γ_+, and γ_-, are set by the specific choice of bath model. For simplicity, we consider here a Markovian bath. The rates of such a bath are determined by

$$\gamma(\omega) = \xi \left| \frac{\omega^n}{1 - e^{-\beta\omega}} \right|, \tag{9}$$

where ω is the amount of energy exchanged when a particular Lindblad operator acts on the system. The relevant rates for our study are γ_-, γ_+, and γ_0, corresponding to domain wall pair annihilation, pair creation, and translation, with energy scales of -4Δ, 4Δ, and 0, respectively. Different n correspond to different types of baths—for $n = 1$ the bath is ohmic, and for $n \geqslant 2$, the bath is superohmic [13]. For our purposes, it will be more convenient to treat γ_0 as a tunable parameter to study the scaling behavior of our protocol. Qualitatively, γ_0 scales linearly with T for ohmic baths and equals zero for superohmic baths. For simplicity, we work in units where $\xi = 1$.

For more details of the master equation approach used to study this model, see Ref. [18].

IV. THE PROTOCOL

A. Protocol considerations for the 1D Ising model with measurements

While we are interested in developing protocols that do not use measurements, we first review a conventional
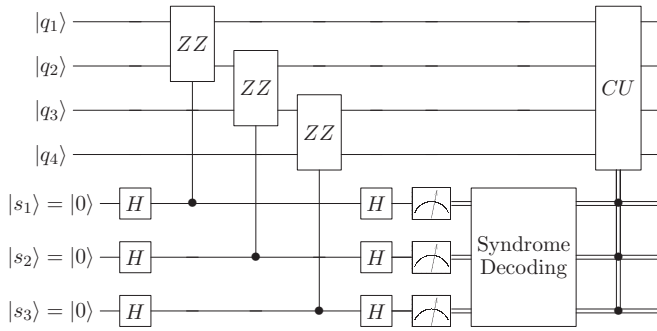
FIG. 2. Circuit for performing error suppression for a subregion of the 1D Ising model. Qubits q_1 through q_4 are lattice sites on an Ising chain. s_1 through s_3 are ancillas used to read out the syndrome measurements of ZZ on the nearest-neighbor Ising lattice sites. Based on the results of the syndrome measurements, the conditional unitary operator CU corrects the errors present. A table that defines CU is provided in Appendix A. This entire circuit represents a possible realization of one such operator O from Fig. 1.

measurement-based protocol for performing error correction in the 1D Ising model.

In the absence of resource constraints, it is straightforward to construct the operators that correct errors in the 1D Ising model. According to the schematic shown in Fig. 1, the $\lambda = 2$ analog of O is simply the domain-wall annihilation operator, D_b from Eq. (7). In more generality, for larger λ the corresponding O is the operator which, given an even number of domain walls, annihilates all domain wall pairs in the region being operated upon. For example, the circuit for the $\lambda = 3$ version of this operator is depicted in Fig. 2. Note that for an odd number of domain walls, there is not an unambiguous choice for how to annihilate domain walls because a free, unpaired domain wall is always left over.

Thus, the most straightforward conventional error-correcting protocol is simply to measure the system's stabilizers often enough that one can unambiguously locate pairs of domain walls and then perform correction operations, as indicated in Fig. 2. This can be represented by a sequence of measurement operators, the stabilizer for the 1D Ising model, S_i, interleaved by conditional application of corrective unitaries: DSWAP and DWALL. These operators have the following representation in the Pauli basis:

$$\text{DWALL} = \tfrac{1}{2}(\text{III} + \text{I}\sigma_x\text{I} - \sigma_z\text{I}\sigma_z + \sigma_z\sigma_x\sigma_z), \qquad (10)$$

$$\text{DSWAP} = \tfrac{1}{2}(\text{III} + \text{I}\sigma_x\text{I} + \sigma_z\text{I}\sigma_z - \sigma_z\sigma_x\sigma_z), \qquad (11)$$

where there is a pair of these operators for each triple of lattice sites. In Fig. 2, the details of the four-qubit operator CU are abstracted away (see Appendix A), but it can be decomposed into applications of DSWAPs and DWALLs, conditioned on syndrome measurements. A simple calculation shows $[H_{\text{Ising}}, \text{DSWAP}] = 0$ and $[H_{\text{Ising}}, \text{DWALL}] = +4\Delta$.

Intuitively, DWALL destroys (creates) a domain wall pair at the dual lattice site in-between the three qubits being operated on if and only if a domain wall pair is present (or, all of the spins are aligned), respectively. DSWAP translates a domain wall, either left or right, if and only if a single domain wall exists between the three spins being operated on.

Error correction protocols for the 1D Ising model can thus be understood as procedures for efficiently dissipating domains walls before they propagate too far. We leverage this intuition in the following sections to build a procedure that dissipates domain walls *without* explicit knowledge of the locations of those domain walls.

B. A protocol without measurements

Because we seek a protocol *without* explicit measurements, the natural operators for such a procedure are DWALL and DSWAP. DWALL is inconvenient, both because the bath already acts to dissipate excitations and because it can lead to the generation of extra, uncontrolled domain walls more easily than the DSWAP operator. Consequently, we only use DSWAPs in our protocols.

If we restrict our attention to the low-temperature regime with the rate assumption $L\gamma_+ \ll 1$, then the lifetime of the Ising chain is governed by the dynamics of single pairs of defects. For error correcting purposes, it is convenient to classify the common geometries of pairs of domain walls. First, *correctable errors* are those errors for which the pair of domain walls is not yet separated by $L/2$ or more. *Noncorrectable* errors are those domain wall configurations in the complement of this set. In the language of error correction, the distance for this code is $\lfloor L/2 \rfloor$—more transparently, correctable errors are those errors that will be correctly matched by a perfect decoder. Furthermore, we need to distinguish between *trivial* and *nontrivial* defect pairs. A domain wall pair is trivial if two domain walls sit on neighboring dual lattice sites. Again, assuming we operate in the low-temperature regime, these trivial defect pairs annihilate with rate γ_-—that is, much faster than other time scales of the problem. Nontrivial pairs are those pairs that are not on neighboring dual lattice sites.

Designing a successful protocol for the Ising model amounts to designing a sequence of DSWAPs that efficiently dissipates nontrivial, correctable defect pairs. If we let χ be the rate at which DSWAPs can be applied, then we expect an enhanced lifetime given the following rate assumptions:

$$\gamma_- \gg \chi/O[\text{poly(L)}] \sim \gamma_0 > \gamma_+. \qquad (12)$$

To wit, DSWAPs are applied at a rate much slower than the inherent annihilation rate of the system—this is so DSWAPs do not turn trivial defect pairs into nontrivial defect pairs. Furthermore, χ is chosen to be close to the inherent translation rate so that correctable, nontrivial defect pairs can be brought adjacent to one another and then be dissipated by the bath before they have time to translate out of the correctable range of the protocol. The $O[\text{poly}(L)]$ factor multiplying χ accounts for the fact that different protocol require some polynomial in L number of swaps to sweep across the entire lattice. For a proof of the polynomial scaling in L; see Appendix B.

In the absence of a corrective protocol, this intrinsic hopping rate of the Ising model gives rise to a simple, background error rate [18,40],

$$\Gamma_0 = \frac{\gamma_0}{1 + e^{\Delta/T}}. \qquad (13)$$

Thus, the natural lifetime of the system in the absence of any corrective protocols can be defined as $1/\Gamma_0$.
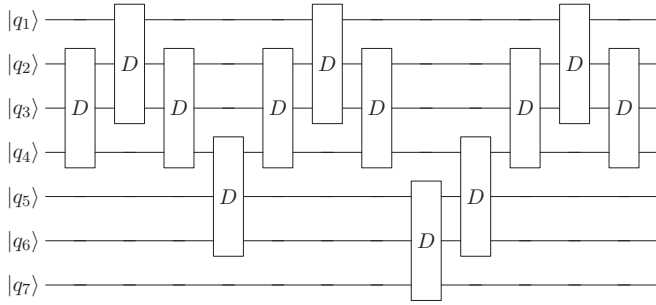
FIG. 3. Sequence of DSWAPs, denoted C, for a $\lambda = 3$ λ-mixing protocol. If a pair of domain walls exist anywhere between sites q_1 through q_7, then they will necessarily be brought adjacent to each other by this sequence of swaps. Gates are applied sequentially with waiting time $1/\chi$ between each gate.

C. Protocol construction

In this section, we construct an autonomous error correction protocol for the 1D Ising model with a variable length-scale λ. The design of the protocol reduces to attempting to perform a sequence of DSWAPs that will necessarily cause any arbitrarily placed pair of domain walls within a region of length 2λ to become neighbors. We refer the reader to Appendix B for a more complete discussion of this strategy.

There are a variety of ways to construct protocols that achieve this in a number of DSWAPs that scales polynomially in the length of the system. Here we focus on protocols that we call λ-mixing. By definition, these are protocols which, in the absence of errors, never translate domain walls a distance λ or greater. For an Ising model of length L, λ runs from 1 to $\lfloor L/2 \rfloor$. In the language of error correction, the protocol can be designed to correct errors of distance 1 to distance $\lfloor L/2 \rfloor$.

First, the dual lattice is subdivided into nonintersecting subregions of length λ. Then, two adjacent regions are chosen, and a λ-mixing protocol is applied over that subregion of total length 2λ. DSWAPs are chosen to move defects toward the shared boundary of the two regions but not to mix defects between the boundaries. The nonintersection of the two regions is crucial: if the protocol did not have this feature, it would actually *increase* the error rate, effectively increasing the inherent translation rate, and thus diffusion rate of defects in the system. Figure 3 depicts a circuit for this protocol for $\lambda = 3$, and Fig. 4 depicts the same circuit acting on the
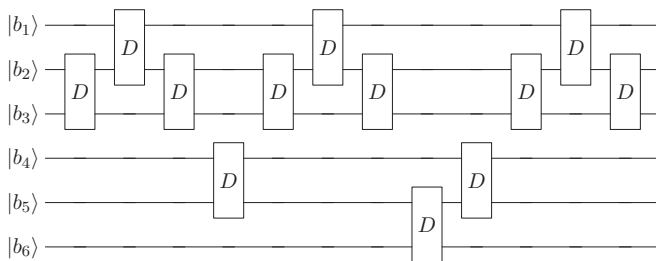

FIG. 4. The same sequence from Fig. 3 but shown acting on domain-wall variables. Here, it is clear that the sequence of DSWAPs is designed not to mix domain walls between the two regions of size $\lambda = 3$. Site b_1 sits between q_1 and q_2, b_2 between q_2 and q_3, etc.
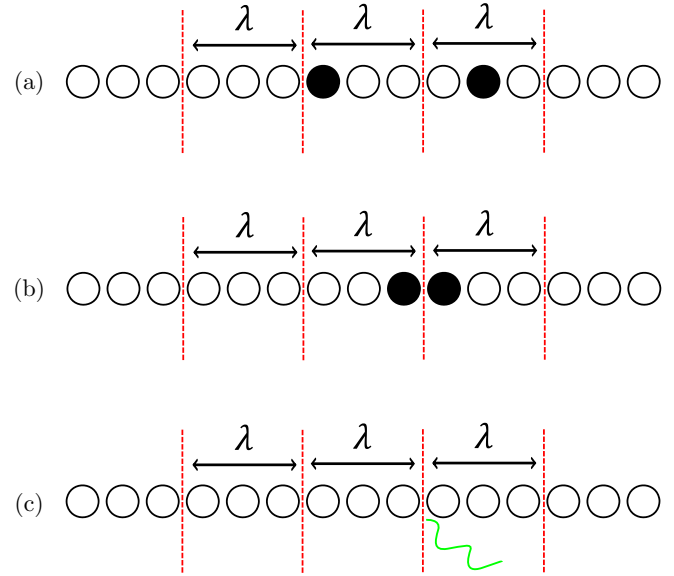

FIG. 5. One possible snapshot of the error correction process. (a) A system with two domain walls present, each sitting in adjacent λ-domains. (b) The state of the system after the protocol has been applied—domain walls have been shuttled to the shared boundary. In (c), the bath dissipates the domain walls, and the system returns to the ground state.

domain wall variables. Figure 5 illustrates a snapshot of this entire procedure for a representative error process involving two domain walls sitting in neighboring λ domains.

This circuit should be reminiscent of the illustration in Fig. 1. For our purposes, the operator O from Fig. 1 is the full sequence of DSWAPs in Fig. 3 dressed by the probabilistic action of creation, annihilation, and translation operators by the bath on the system.

We provide code for this algorithm in the Appendix C, including how the λ-mixing subprotocols are constructed.

D. Error modes and scaling

In this section, we examine how uncontrollable thermal errors lead to loss of the qubit in the presence of the protocol.

In the presence of a corrective protocol, and assuming the correction rate χ is close to the translation rate of the system but still much less than the annihilation rate, the rate of the lowest-order error process is given by

$$\tilde{\Gamma} = L\gamma_+ \frac{\gamma_0}{\lambda\chi} \frac{\gamma_0}{\gamma_-} \frac{1}{L - 2 - 2\lambda} \frac{f(\lambda)}{L}. \tag{14}$$

This rate is the product of (i) the baseline production rate of defect pairs, $L\gamma_+$, (ii) the probability of a defect pair not immediately annihilating, $\frac{\gamma_0}{\gamma_-}$, (iii) the probability of a defect exiting a corrective region, $\frac{\gamma_0}{\lambda\chi}$, (iv) the probability of a nontrivial random walk across the chain $\frac{1}{L-2-2\lambda}$, divided by a factor proportional to the number of correcting regions on the lattice. Thus, for fully parallel application $f(\lambda) \propto \lambda$. Without the protocol, the probability that a pair of domain walls undergoes a random walk that winds around the entire system scales like $\frac{1}{L-2}$, but when the protocol is implemented, the effective lattice size is slightly reduced: the particle need

only come within approximately a distance 2λ of its partner for the protocol to fuse them. Thus, $\tilde{\Gamma}$ is the leading order estimate for the rate at which domain wall pairs are "missed" by the protocol and cause uncontrolled transitions between the two ground states of the model.

This effective rate is valid as long as χ is fast enough to compete with γ_0, but not so fast as to compete with pair annihilation, γ_-, and other higher-order processes in $\frac{\gamma_0}{\gamma_-}$ and $\frac{\gamma_0}{\chi}$. Further, we assume that random walks occur much faster than the intrinsic creation rate, or $\gamma_0/L^2 \gg \gamma_+$, and that we can model the random walk as occurring instantaneously—for sufficiently large lattice sizes, the breakdown of this assumption would introduce an additional γ_0 dependence into this error rate to account for the nonzero amount of time it takes defects to traverse the lattice. For regimes studied here, γ_0 and χ run from 10 000 to 100 000 times faster than the intrinsic pair creation rate, γ_+.

It might be tempting to examine the form of Eq. (14) and expect that errors vanish in the limit of $\gamma_0 \to 0$, but a new effective translation rate appears once $\gamma_0 \ll \gamma_+$. In this regime, two pairs of domain walls can appear next to one another, and a consecutive annihilation event produces a lone pair of domain walls separated by two dual-lattice sites. In this way, an effective translation rate is set by the rate at which these doubled-pair creation events occur. We do not consider this limit further, but it is the natural error process for superohmic baths at low temperature.

To model the breakdown of Eq. (14) as χ is varied, we can approximate the error timescale, $\frac{1}{\tilde{\Gamma}}$, as being effectively reduced by some factor proportional to χ:

$$\frac{1}{\tilde{\Gamma}} \to \frac{1}{\tilde{\Gamma}} \left\{ 1 - \chi g(\lambda, L, \gamma_0, \gamma_-, \gamma_+) + O\left[\left(\frac{\chi}{\gamma_-}\right)^2 + \left(\frac{\chi}{\gamma_0}\right)^2 \right] \right\},$$
(15)

with $g(\lambda, L)$ a protocol-dependent scaling function. Heuristically, for fixed λ, one expects that g should scale linearly with the number of parallel domains of size λ because, for twice as many domains, twice as many pairs will be pulled apart by the protocol that would have otherwise fused. At the same time, for a fixed number of domains, i.e., fixed $\frac{L}{\lambda}$, any given pair of lattice sites is only ever operated on by a DSWAP for a fraction of the corrective cycle. So, for fixed χ and fixed $\frac{L}{\lambda}$, as λ is increased, domain walls may spend a longer amount of time sitting on a boundary before being caught by the protocol. For the protocol used in this paper, this is cubic in λ. Thus,

$$g(\lambda, L, \gamma_0, \gamma_-, \gamma_+) \propto g(\gamma_0, \gamma_-, \gamma_+)\lambda^3 \frac{L}{\lambda} = g(\gamma_0, \gamma_-, \gamma_+)\lambda^2 L.$$
(16)

This scaling behavior suggests a critical cycling rate, χ_c, at which the lifetime is maximally improved by the protocol. Differentiating Eq. (15) with respect to λ yields the critical rate, up to the rate function g,

$$\chi_c = \frac{1}{2\lambda^2 L g(\gamma_0, \gamma_-, \gamma_+)},$$
(17)

where any residual prefactors and terms involving γ_0, γ_+, and γ_- have been absorbed into g.

E. Memory enhancement and scaling

We now present numerical results demonstrating the enhanced lifetime of the Ising model when subjected to λ-mixing protocols in serial and in parallel. For serial application, only a single corrective operation was applied every $1/\chi$ units of time. For parallel application, $L/(2\lambda)$ simultaneous corrective operations were applied every $1/\chi$, where each operation acted on a nonintersecting region of length (2λ).

For the following analysis, we define the *lifetime* as the average time it takes a 1D Ising model initialized to the spin-up state to transition to the spin-down state. In the absence of the protocol, that is, in the low-χ limit, this lifetime asymptotes to approximately the lifetime given by Eq. (13).

For the details of the Monte Carlo algorithm, see Ref. [18]. The only nontrivial choice required at the level of simulation is how to treat the competition between the application DSWAPs and bath operators. For simplicity, we assume if a bath operator takes longer than $1/\chi$ to occur, that the DSWAP occurs unhindered. Likewise, if a bath operator takes less than $1/\chi$ to occur, the transformation associated with that bath operator occurs unhindered, be that a pair creation, pair annihilation, or single translation. More complicated choices could be made, like choosing a probabilistic failure rate of a DSWAP as a function of the ratio of the competing timescales, but we do not expect the result of a such a treatment to greatly affect our analysis.

Figure 6 depicts the scaling of the 1D Ising model's lifetime with λ at fixed L, where a smaller λ results in more domains being operated on in parallel. Specifically, for parallel simulations, the protocol was performed simultaneously on $L/(2\lambda)$ domains. These domains were chosen such that DSWAPs were only being applied on nonoverlapping regions of characteristic
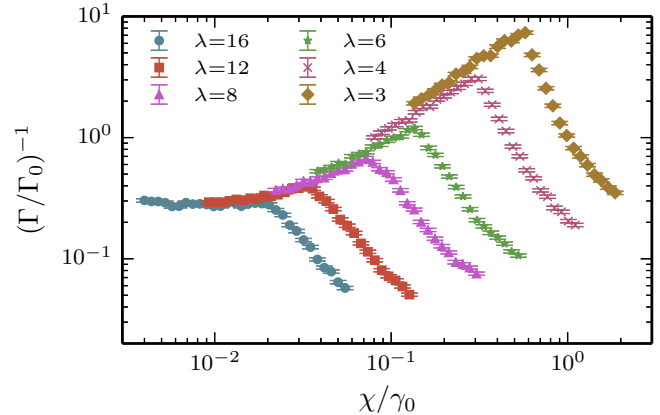

FIG. 6. Measured lifetime of the Ising model, Γ^{-1}, expressed in terms of inverse units of Γ_0 (i.e., Eq. (13)) as a function of the correction rate, χ, rescaled by the translation rate, γ_0, for different values of λ and for fixed system size $L = 96$, temperature, $T = .07$, and translation rate, $\gamma_0 = .0007$. Protocols were implemented in parallel on $48/\lambda$ blocks (see text). In the absence of the protocol, the lifetime of the Ising model for these parameters corresponds to approximately Γ_0^{-1}, i.e., Eq. (13). This is the value which all three protocols converge toward in the limit of $\chi/\gamma_0 \ll 1$. Note the decrease in lifetime for $\chi/\gamma_0 \approx 1$, as well as the universal scaling in the lifetime up until the λ-dependent cutoff (i.e., Eq. (17)), which increases as λ decreases. The only remaining mismatch between the curves is due to the residual λ dependence in the $1/(L - 2 - 2\lambda)$ term of Eq. (15).
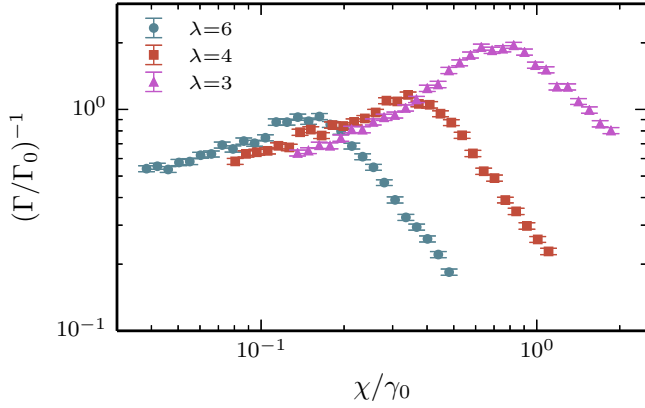
FIG. 7. Lifetime of the Ising model as a function of the correction rate, χ, rescaled by the translation rate, γ_0, for different values of λ, and for fixed system size, $L = 96$, temperature, $T = .07$, and translation rate, $\gamma_0 = .0007$. Protocols were implemented serially (see text). The scaling of lifetime with χ is characteristically similar to the parallel case; however, the maximal lifetime is correspondingly smaller for the serial implementation. Note that smaller λ still yields a larger enhanced lifetime.

size λ. Here, increasing parallelization manifestly increases the lifetime of the model. For small χ, the protocol does nothing, and the memory converges to the value of the memory in the absence of any corrective protocol, i.e., Eq. (13). For χ approaching γ_-, the protocol begins to compete with the process of pair annihilation, and begins turning trivial defect pairs into nontrivial pairs. This actually reduces the lifetime below that of the protocol-free value. In the intermediate regime, the optimal lifetime grows linearly with the number of parallel blocks employed in the algorithm. For this particular protocol, the number of parallel blocks was $48/\lambda$.

Figure 7 depicts the scaling of lifetime with λ, as in Fig. 6, but for a serial application of the protocol. For serial application, only a single DSWAP operator ever operates on the system over a timescale χ^{-1}. Decreasing λ also manifestly
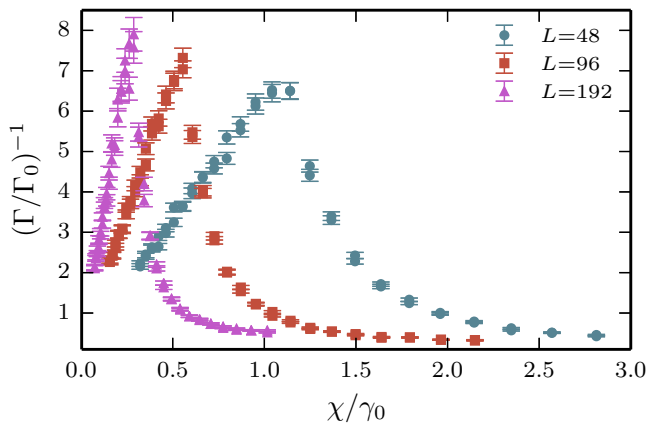

FIG. 8. Lifetime of the Ising model as a function of the correction rate, χ, rescaled by the translation rate, γ_0, for different values of system size, L, and for fixed $\lambda = 3$, $T = .07$, $\gamma_0 = .0007$. Protocols were implemented in parallel on $L/2\lambda$ blocks (see text). Note the linear scaling in χ for small values, as well as the shift in the maximum of the lifetime as a function of L.
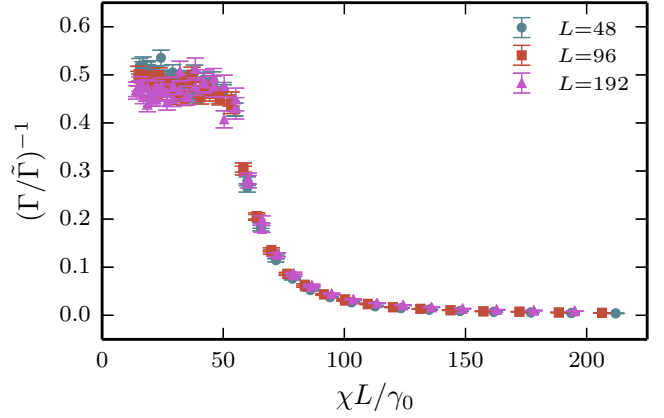

FIG. 9. This figure contains the same data as Fig. 8, but with the χ axis rescaled to χL, and the $1/\Gamma$ axis rescaled by Eq. (14). Hence, the linear scaling in χ, and the slight residual system size dependence have been removed. Note the steep, sudden dropoff in lifetime after $\chi L \sim 55$.

increases the maximum enhanced lifetime of the protocol. Thus, for fixed-resource architectures, smaller λ necessarily outperforms larger λ implementations.

Figures 8 and 9 depict the scaling of the lifetime with L at fixed λ for parallel application. Remarkably, the L dependence of the models can be completely removed by rescaling the data by Eq. (14), and rescaling χ to χL as depicted in Fig. 9. This rescaling reveals the turnaround in the scaling of the lifetime for $\chi L/\gamma_0 = 54.2 \pm 2.8$, whereafter it transitions from linear scaling in χ to a power law decay.

Figure 10 depicts the scaling of the critical cycling rate χ_c versus the inverse system length $1/L$. This scaling agrees with the error ansatz of Eq. (17).

We note that substituting χ_c from Eq. (17) into Eq. (14) yields $1/\tilde{\Gamma} \sim \frac{L-2-2\lambda}{L}$—or that the maximum enhanced lifetime is asymptotically independent of system size. Thus, even for much larger system sizes, the maximum achievable lifetime will not greatly exceed the maximum lifetime for the $L = 192$ result in Fig. 8.
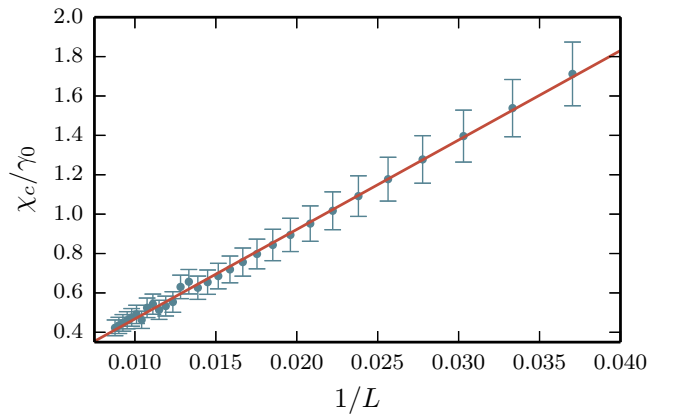

FIG. 10. The critical cycling rate, χ_c, rescaled by the translation rate, γ_0, as a function of $\frac{1}{L}$ for $\lambda = 3$, $T = 0.07$, $\gamma_0 = 0.0007$. Protocols were implemented in parallel on $L/2\lambda$ blocks. This scaling is consistent with the error model ansatz in Eq. (17). Fit to $1/L$ in red. Errors are dominated by systematic effects, not sampling error.

V. HIGHER DIMENSIONS AND GENERALIZATION

A. The toric code

The argument and construction from the previous section immediately generalizes to any higher-dimensional stabilizer codes with stringlike error operators. The immediate analog is Kitaev's toric code, whose Hamiltonian is defined as a sum over vertex and plaquette operators acting on the edges of a square lattice,

$$H_{\mathrm{TC}} = -J_e \sum_v A_v - J_m \sum_p B_p, \qquad (18)$$

$$A_v \equiv \prod_{j \in v} \sigma_j^z, \quad B_p \equiv \prod_{j \in p} \sigma_j^x. \qquad (19)$$

The low-temperature dynamics of the toric code are governed by the proliferation of localized excitations that are created by stringlike error operators, with dynamics similar to to those of the 1D Ising model. However, toric code dynamics differ in two ways: first, there are now two types of defects in the toric code—defined as -1 eigenstates of the A_v and B_p operators, located on the vertices and plaquettes of the square lattice, respectively. Because of the nontrivial braiding statistics of these two defects, a nontrivial winding resulting from the interaction with the bath gives rise to uncontrolled errors that can be mapped onto logical Z *and* logical X operations, depending on which type of defect incurs the nontrivial winding. These erroneous operations can be suppressed by operating at low temperature and by tuning the relative strength of the plaquette and star terms in the Hamiltonian. Second, both of these defects undergo two-dimensional random walks rather than one-dimensional random walks. This difference in dimension gives rise to a modified form of the toric code's finite temperature error rate, due to the differing nontrivial topological random walk probability for two dimensions versus one.

Operationally, these differences only require small modifications of the autonomous protocol. Namely, there need be two DSWAP operators:

$$
\begin{aligned}
\mathrm{DSWAP}_{vv'}^e = {} & \tfrac{1}{4}\sigma_{vv'}^x(1 - A_v)(1 + A_{v'}) \\
& + \tfrac{1}{4}\sigma_{vv'}^x(1 + A_v)(1 - A_{v'}) \\
& + \tfrac{1}{4}(1 - A_v)(1 - A_{v'}) \\
& + \tfrac{1}{4}(1 + A_v)(1 + A_{v'}), \\
\mathrm{DSWAP}_{pp'}^m = {} & \tfrac{1}{4}\sigma_{pp'}^z(1 - B_p)(1 + B_{p'}) \\
& + \tfrac{1}{4}\sigma_{pp'}^z(1 + B_p)(1 - B_{p'}) \\
& + \tfrac{1}{4}(1 + B_p)(1 + B_{p'}) \\
& + \tfrac{1}{4}(1 - B_p)(1 - B_{p'}).
\end{aligned} \qquad (20)
$$

These unitary operators translate an A-type (B-type) excitation from a vertex v (plaquette p) to an adjacent vertex v' (plaquette p'). Second, the λ-mixing protocol shuttles defects towards a shared boundary of length λ between subdomains of characteristic area λ^2.

Because subregions share a boundary of length λ rather than a single site, as in the one-dimensional case, the cycling protocols require at most a factor of λ more swaps to complete a cycle. The protocol then takes the following simple form:

(1) Choose a species of quasiparticle.

(2) Divide the lattice into domains of characteristic area λ^2.

(3) Pick two λ-domains which share a boundary.

(4) Pick two candidate defect locations within these two λ-domains.

(5) If these defect locations are within the same λ-domain, apply DSWAPs until they would be nearest neighbors. If they are in different λ-domains, apply DSWAPs until they meet at the shared boundary.

(6) Repeat step 5 until all pairs of defect locations are exhausted.

(7) Repeat steps 3 through 6 until all pairs of λ domains which share a boundary are exhausted.

(8) Repeat steps 1 through 7 until all species of quasiparticle are exhausted.

This protocol is also highly parallelizable, both by operating on multiple pairs of λ domains and by acting on simultaneous pairs of defect sites within pairs of λ domains.

B. The general problem

We can always divide a d-dimensional lattice into $N \equiv L^d/\lambda^d$ domains and try to devise an algorithm that fuses defects between adjacent domains. For our protocol, defects are shuttled toward $d - 1$-dimensional boundaries between adjacent domains of volume λ^d. From this, we can generalize the low-temperature dynamics of Eq. (14) to the d-dimensional case as follows:

$$\tilde{\Gamma} \propto L^d \gamma_+ \frac{\gamma_0}{\lambda \chi} \frac{\gamma_0}{\gamma_-} P_\Omega^d(L, \lambda) \frac{f(\lambda^d)}{L^d}, \qquad (21)$$

where L is the edge length of the d-dimensional volume enclosed by the system, $P_\Omega^d(L, \lambda)$ encodes the probability of a nontrivial topological random walk of a pair of defects in d dimensions with system size L and domain lengthscale λ, and $f(\lambda^d) \propto \lambda^d$ is a protocol-dependent function that depends on the implementation details of the algorithm.

Theorem Eq. (B1) guarantees that an algorithm exists that can perform the cycling in a number of steps polynomial in the dimension of the lattice. However, this theorem does not guarantee that a λ-mixing protocol exists which solves the problem. In general, for higher dimensions, there are always defect patterns of distance $O(\lambda)$, which are uncorrectable by our λ-mixing protocol. The design strategy is then to try and maximize this minimum uncorrectable distance by careful tiling of the graph of interest.

To be more explicit, if a single pair of defects appears on the graph, uncorrectable errors are generated only when one of the defects escapes to an adjoining region that does not share a boundary with its pair. An illustration of this process is depicted in Fig. 11. For one dimension, this cannot happen in one step after an adjacent pair of defects appears. More specifically, if a single pair of adjacent defects appears on the lattice, no single DSWAP will cause such an error to occur, by design, and no single bath operation will cause an adjacent pair of defects to be in nonadjacent regions of size λ. In one dimension, at worst a pair will be created, shuttled around by
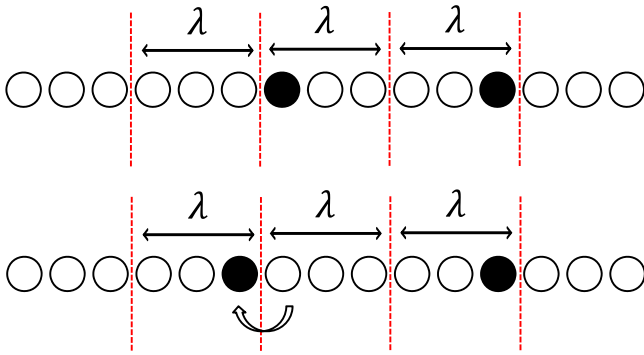
FIG. 11. A single random application of the bath-hopping operator causes a defect to move between adjacent regions of size λ. Once a pair of defects are separated by this distance, the protocol will not be able to correct them with certainty.

the protocol, and *then* translate by a bath operator across a boundary, resulting in an uncorrectable error.

This distinction is important because poor choice of tiling in higher dimension *can* result in uncorrectable errors that occur in a single step after pair creation. For example, compare the single hop in the upper half of Fig. 12 to the lower half. A defect pair appearing at a corner can transition to an uncorrectable configuration in a single step, whereas in the lower tiling, this is not possible for any initial configuration of adjacent defect pairs. This can be checked by simple enumeration of the possible defect locations and single-hop geometries.

This shifted square lattice tiling depicted in the lower half of Fig. 12 generalizes to three dimensions and is necessary for Eq. (21) to describe the leading order error process.

C. Hybrid DSWAP-stabilizer codes

While we have demonstrated that our protocol gives rise to an enhanced lifetime for a topological code with string-like
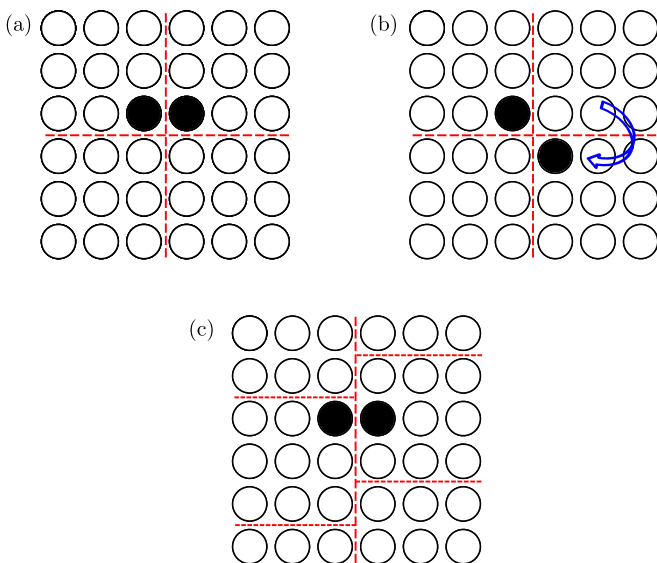

FIG. 12. A single random application of the bath-hopping operator causes a defect to move between nonintersecting regions of size λ, depicted in (a) and (b). (c) A new lattice tiling where no single swap can move defects into two λ regions that do not share a boundary.

error operators, it is also possible and desirable to use our DSWAP cycling protocol with a more traditional stabilizer detection-correction scheme simultaneously. We postpone numerical analysis of such a scheme for future work, but we sketch such a protocol in this section.

For concreteness, we specialize here to the case of the toric code. Error detection and correction in the toric code requires (1) measurement of all stabilizer syndrome operators and (2) application of a perfect-matching algorithm to determine which pairs of defects to fuse. Whether or not such an algorithm will be successful depends on the density of errors at the time of measurement. Given a stabilizer measurement rate γ, in the "infinite temperature" limit (i.e., $T \to \infty$ in Eq. (9)), each edge is acted upon by an error operator with an error probability p. Operationally, the bath is equally likely to create a pair of defects as it is to dissipate a pair. The resulting dynamics are analogous to an uncorrelated "white noise" model.

It is well known that if p is below some critical value, p_c, it is possible to correct the errors in the toric code with certainty. Equivalently, p_c sets the minimum rate at which measurement must occur so that detection is possible in principle. Call this rate γ_c.

For the protocol to have an effect, we must operate in a regime where pair annihilation is favored over pair creation. For simplicity, we work in the low-temperature regime where single-defect pairs dominate. In this regime, an uncorrectable error has occurred when a single pair of defects becomes separated by more than half the linear lattice dimension. In the presence of the DSWAP cycling protocol, the rate associated with such an event occurring is modified by some constant factor:

$$\frac{1}{\Gamma_{\text{Toric Code Cycling}}} = g \frac{1}{\Gamma_{\text{Toric Code}}}, \qquad (22)$$

where $\Gamma_{\text{Toric Code Cycling}}$ is the error rate of the toric code in the presence of a cycling protocol, and $\Gamma_{\text{Toric Code}}$ is the error rate in the absence of the protocol.

For $g > 1$, i.e., when our protocol actually enhances the lifetime of the code, this effectively reduces the critical detection rate γ_c by the same factor. This is because the protocol effectively reduces the rate at which undetectable pairs are created.

Thus, if a physical realization of a stabilizer error detection and correction cycle is rate limited due to hardware or fundamental noise constraints, the DSWAP cycling protocol provides one avenue toward reducing the critical measurement and detection rate purely by application of local unitaries.

VI. DISCUSSION

In summary, we have provided a dissipative error correction protocol that enhances the lifetime for models with stringlike defects (see Sec. IV C). In particular, we have combined a local dissipative thermal protocol with carefully designed unitaries, in order to further encourage defect dissipation. Furthermore, we have derived an enhanced lifetime for the one-dimensional Ising model in the presence of our protocol, i.e., Eq. (14), determined the scaling behavior of the optimal protocols, i.e., Eq. (17), and provided numerical evidence for the enhanced lifetime of the Ising model in the presence of our protocol

given certain rate assumptions (see Figs. 6–10). Practically, this algorithm increases the lifetime of the system linearly with system size up to a system-size independent cutoff, illustrated in Fig. 8, as anticipated from no-go theorems. Furthermore, we have sketched how this protocol can be generalized to higher-dimensional models like the toric code and used in conjunction with traditional stabilizer error detection and correction schemes (see Secs. V A and V C).

The efficacy of these sorts of protocols is intimately related to the scaling of the protocol with system size and protocol parameters, as we have demonstrated. Notably, the best performing versions of our protocol have small λ, and, in a sense, only correct the shortest distance errors. This may seem counterintuitive from the perspective of designing protocols, which correct as many errors as possible. For example, suppose we wish to compare a $\lambda = 3$ protocol with total cycle time τ to a $\lambda = 4$ protocol with the same cycle time τ. Note that by fixing total cycle time, we are implicitly requiring that the $\lambda = 4$ protocol be performed more quickly at the level of individual application of DSWAPs (because there are more DSWAPs in a complete $\lambda = 4$ cycle than a $\lambda = 3$ cycle), but we require that the complete error-correcting cycle of each protocol is completed in the same amount of time. Naively, we would expect the $\lambda = 4$ protocol to do better, because it is dissipating errors over a longer length scale, but in the same amount of time. Figure 7 indicates a narrow region where this is the case, but, generically, this is not the case.

This can be traced to the poor scaling of the maximal lifetime with λ, as represented by Eq. (17). Essentially, the protocols which correct larger distance errors—i.e., large λ-fixing protocols—employ so many gates that all of the gains of correcting longer distance errors are erased by the time it takes to actually perform the protocol, even when implementing the protocol in parallel.

Additionally, the form of Eq. (14) suggests that larger systems manifestly have lower error rates, because $\tilde{\Gamma} \propto \frac{1}{\chi L}$ for low temperature and $L \gg \lambda$. For sufficiently large systems this once again breaks down due to Eq. (17) (see also, Fig. 10). Namely, for a fixed cycling rate χ_0, making the system larger only increases the lifetime so long as $\chi_0 < \chi_c$. Thus, the more favorable scaling of the lifetime with χ at larger system sizes is precisely offset by the poor scaling of χ_c. This can be seen immediately by inserting χ_c from Eq. (17) into Eq. (14). This replacement yields the expected scaling of our "best" measurement-free protocol, which scales asymptotically as $\sim \frac{L - \text{constant}}{L}$—i.e., upper bounded by a constant.

These shortfalls could be circumvented by allowing for longer-range unitaries. For example, the DSWAP operator could be replaced by a generalized operator DSWAP$_\lambda$, which transports domain walls over longer distances. Our insistence on building the protocol entirely out of local DSWAP gates was to perform as honest an analysis as possible with respect to the power of this type of protocol. But if a particular architecture could exchange defects over long distances just as easily as short ones, this would immediately allow for algorithms with better scaling. We hope to examine the optimality of these sorts of protocols in future work.

In the long term, this program is meant to identify the simplest possible set of ingredients necessary to provide protection for a stabilizer code based quantum memory. Many partial ingredients are known, like the no-go theorems mentioned in Sec. I. Practically, the goal is a protocol designed around the dynamics of the excitations of the stabilizer codes of interest with minimal usage of resources but which still results in an error threshold so that a state can be preserved indefinitely. With this work, we have demonstrated a constant factor improvement with only local unitaries dressing the system.

APPENDIX A: SYNDROME DECODING FOR 1D ISING MODEL

The corrective operator CU in Fig. 2 can be written as a collection of conditional applications of DSWAP and DWALL, where the applications of the operators are conditioned on the measurements of the stabilizers. We adopt the notation U_{123} to indicate the application of the unitary U on qubits 1, 2, and 3. The operator U is given explicitly in Table I for several syndrome measurements of the stabilizers.

APPENDIX B: MATCHSEQ AND ERROR CORRECTION

1. Polynomial scaling

Define the game MATCHSEQ as follows: two nonadjacent vertices on a simply connected graph G are colored black, called defects, the rest are white. The player is allowed to perform a conditional swap, or DSWAP, on any two adjacent vertices, which exchanges black vertices and white vertices, and does nothing to pairs of white vertices. If black vertices become adjacent, they immediate *fuse* and become white vertices. Crucially, the player does not know which vertices are colored black.

"Winning" MATCHSEQ amounts to performing a sequence of moves that guarantees that a pair of arbitrarily placed vertices fuses.

Define the pairing sequence $M(G_v)$ to be the sequence of conditional swaps necessary to bring any configurations of two defects adjacent to one another at least once on a graph G with v vertices. Define the pairing number $|M(G_v)|$ to be the pairing

TABLE I. Corrective operations given certain measurements of the stabilizers s_1 through s_3 in Fig. 2. If an odd number of domain walls are detected, the identity is applied.

s_1	s_2	s_3	U
1	1	1	I
1	1	-1	I
1	-1	1	I
1	-1	-1	DWALL$_{234}$
-1	1	1	I
-1	1	-1	DSWAP$_{123}$DWALL$_{234}$
-1	-1	1	DWALL$_{123}$
-1	-1	-1	I

TABLE II. Minimum number of DSWAPs required to necessarily fuse any two defects on a linear chain with open boundary conditions. Computed via breadth first search.

$|M(3)|$	1
$|M(4)|$	3
$|M(5)|$	6
$|M(6)|$	10
$|M(7)|$	18

sequence with minimal length. Table II tabulates the first few nontrivial pairing numbers for the special case of G equal to a linear chain of length L. See Fig. 13 for an illustration of the application of a winning pairing sequence to a graph with four vertices.

Theorem B.1. The number of DSWAPs necessary to win *MATCHSEQ* for an arbitrary finite, connected graph G is polynomial in the number of vertices in the graph G.

Proof. Let $M^*(G)$ be a winning strategy on an arbitrary graph G. Suppose an arbitrary vertex is added to G, called v^*, with up to $|G|$ edges. Call this modified graph G'. Then, performing $M^*(G)$ on G' either fuses two arbitrarily placed defects, or there's a single defect on the new vertex, and the remaining vertex has just been permuted around in G. A candidate $M(G')$ is then: 1. Perform M^* on G. 2. Pick a vertex, v', on G. Supposing a defect is on v', perform a sequence of DSWAPs that brings that defect adjacent to the new vertex, v^*. 3. Perform the reverse of the sequence of DSWAPs in step 2, and repeat step 2 with a new v'. 4. Repeat steps 2 and 3 until all vertices in G are exhausted.

The number of DSWAPs needed for step 2 is at most $|v|$, i.e., the number of vertices in G. Thus, the total complexity of steps 2 through 4 is $O(|v|^2)$. This admits a recurrence relation:

$$|M(G_{v+1})| \leqslant |M(G_v)| + b * |v|^2, \tag{B1}$$

where b is a constant $\leqslant 1$. $b = 1$ corresponds to the case that the new vertex is only connected to one vertex in the original graph G. Solving this recurrence relation in the limit that the inequality is always saturated yields $|M(G_v)| \leqslant O(|v|^3)$. It is worth emphasizing that this is not the minimal such solution to MATCHSEQ, just one that is easily provably polynomial in $|G|$. The likely graph structures of interest to an experimentalist, i.e., linear chains, square lattices, admit more favorable algorithms with softer polynomial scaling.

2. Strategies

For a given winning strategy, $M^*(G)$, it will be convenient to classify the strategy based on the maximum distance that any given defect is moved. In the sequel, we will construct $M^*(G)$ out of a concatenation of $M^*(G_i)$, where G_i are subgraphs of G. Thus, if only a single defect happens to be in the subgraph G_i, we would like to bound the maximum displacement of that defect by the strategy.

Let $d(M)$ be the maximum distance any given defect is moved by a given strategy. For winning strategies, $d(M)$ is at least half of the maximum distance between defects and at most permutes defects around the entire graph, so $|v| \geqslant d(M) \geqslant |v|/2$. Define a strategy $M^*(G)$ to be *k-mixing* if $d(M) = k$.

We introduce this terminology because most physical realizations of MATCHSEQ will have a background rate of uncontrollable DSWAPs, driven by coupling to a bath. k-mixing strategies are necessary in such cases to be partially resilient to these random "error" DSWAPs.

3. Mapping onto 1D Ising model

Vertices in the problem setup for MATCHSEQ correspond to the dual lattice of the Ising chain, and the process of fusion is simply dissipation of adjacent domain wall pairs by the bath.

However, we caution that the mapping onto MATCHSEQ is only partial: defects on the Ising chain hop in the absence of any experimental intervention, so the Ising chain is more akin to a game of MATCHSEQ with a random, background DSWAP rate. Further, there can be more than two pair of excitations on the Ising chain, but for low temperature, the regime where the protocol works best, this is exceedingly rare. Last, defect pairs don't necessarily fuse immediately—fusion happens at the timescale set by the system-bath coupling, the type of bath model, and the temperature, so rate at which DSWAPs are applied must be chosen carefully for optimal lifetime enhancement.

APPENDIX C: ALGORITHM FOR 1D ISING MODEL

Here we provide python code for a λ-mixing algorithm for the Ising chain. The output of the algorithm is a sequence of locations. Our convention is such that location i indicates a DSWAP should be applied that exchanges defects between sites i and $i + 1$. Heuristically, the algorithm attempts to shuffle defects toward the shared boundary of the disjoint sites 0, 1, 2,..., $\lambda - 1$ and λ, $\lambda + 1$,..., $2\lambda - 1$. That is, it attempts to translate defects so that they are adjacent to each other at sites $\lambda - 1$ and λ. After completing this cycle, the algorithm repeats for the next two adjacent domains, λ,...,$2\lambda - 1$ and 2λ,...,$3\lambda - 1$. This continues until the lattice has been exhausted.

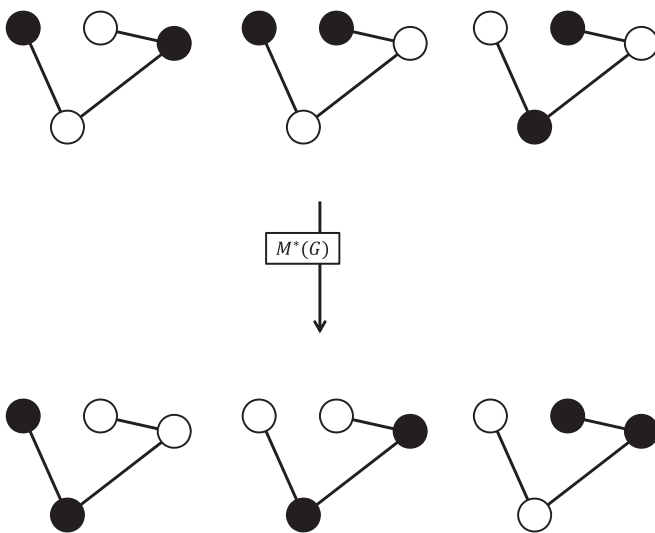The following python code generates a complete sequence of DSWAPs given a lattice size and λ length.

FIG. 13. An example of the result of applying a winning sequence $M(G)$ to a graph with defect pairs present. No matter where the defects are, the sequence of DSWAPs brings pairs adjacent, whereupon they immediately fuse.

```python
def SwapProtocol(L, lamb):
  prot = []
  numofdomains = L / lamb
  for d in xrange(numofdomains):
    for k in xrange(lamb):
      for m in xrange(k):
        prot.append((lamb-1-k+m+d*lamb)%L)
      for i in xrange(lamb):
        for j in xrange(i):
          prot.append((lamb+i-j-1+d*lamb)%L)
  return prot
```

The protocol is parallelized by operating simultaneously on specific pairs of domains. To be more explicit: denote the first λ sites as λ_1, the next λ sites λ_2, and so on. The algorithm can be naturally partitioned into a sequence of DSWAPs that translates defects to the shared boundary of λ_1 and λ_2 (call this sequence (λ_1,λ_2)), followed by a sequence that translates defects to the shared boundary between λ_2 and λ_3 (call this sequence (λ_2,λ_3)), etc. To parallelize, apply the sequence (λ_1,λ_2) simultaneously with (λ_3,λ_4), (λ_5,λ_6), etc. When complete, apply the sequence (λ_2,λ_3) with (λ_4,λ_5), etc. This exhausts the protocol.

[1] A. G. Fowler, Phys. Rev. Lett. **109**, 180502 (2012).

[2] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, Quant. Inf. Comput. **10**, 456 (2010); arXiv:0905.0531 [quant-ph].

[3] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).

[4] K. Fujii, M. Negoro, N. Imoto, and M. Kitagawa, Phys. Rev. X **4**, 041039 (2014).

[5] D. Nigg, P. Schindler, T. Monz, and R. Blatt, Science **345**, 302 (2014).

[6] S. Bravyi and A. Kitaev, arXiv:quant-ph/9811052, translation of Quantum Computers and Computing **2**, 43 (2001).

[7] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).

[8] J. R. Wootton and D. Loss, Phys. Rev. Lett. **109**, 160503 (2012).

[9] Z. Nussinov and G. Ortiz, Phys. Rev. B **77**, 064302 (2008).

[10] Z. Nussinov and G. Ortiz, Ann. Phys. **324**, 977 (2009).

[11] C. Castelnovo and C. Chamon, Phys. Rev. B **76**, 184442 (2007).

[12] R. Alicki, M. Fannes, and M. Horodecki, J. Phys. A: Math. Theor. **42**, 065303 (2009).

[13] S. Chesi, B. Röthlisberger, and D. Loss, Phys. Rev. A **82**, 022305 (2010).

[14] S. Chesi, D. Loss, S. Bravyi, and B. M. Terhal, New J. Phys. **12**, 025013 (2010).

[15] B. Yoshida, Ann. Phys. **326**, 2566 (2011).

[16] O. Viyuela, A. Rivas, and M. A. Martin-Delgado, New J. Phys. **14**, 033044 (2012).

[17] F. H. E. Watson and S. D. Barrett, New J. Phys. **16**, 093045 (2014).

[18] C. D. Freeman, C. M. Herdman, D. J. Gorman, and K. B. Whaley, Phys. Rev. B **90**, 134302 (2014).

[19] B. J. Brown, D. Loss, J. K. Pachos, C. N. Self, and J. R. Wootton, Rev. Mod. Phys. **88**, 045005 (2016).

[20] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).

[21] H. Bombin, R. W. Chhajlany, M. Horodecki, and M. A. Martin-Delgado, New J. Phys. **15**, 055023 (2013).

[22] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, Phys. Rev. A **90**, 022305 (2014).

[23] C. M. Herdman, K. C. Young, V. W. Scarola, M. Sarovar, and K. B. Whaley, Phys. Rev. Lett. **104**, 230501 (2010).

[24] H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, and H. P. Büchler, Nat. Phys. **6**, 382 (2010).

[25] J. Dengis, R. König, and F. Pastawski, New J. Phys. **16**, 013023 (2014).

[26] A. Hutter, J. R. Wootton, B. Röthlisberger, and D. Loss, Phys. Rev. A **86**, 052340 (2012).

[27] F. L. Pedrocchi, A. Hutter, J. R. Wootton, and D. Loss, Phys. Rev. A **88**, 062313 (2013).

[28] E. Kapit, J. T. Chalker, and S. H. Simon, Phys. Rev. A **91**, 062324 (2015).

[29] C.-E. Bardyn and T. Karzig, Phys. Rev. B **94**, 094303 (2016).

[30] S. Bravyi and B. Terhal, New J. Phys. **11**, 043029 (2009).

[31] M. B. Hastings, Phys. Rev. Lett. **107**, 210501 (2011).

[32] O. Landon-Cardinal and D. Poulin, Phys. Rev. Lett. **110**, 090502 (2013).

[33] K. Temme, arXiv:1412.2858v1 (2014).

[34] K. Temme and M. J. Kastoryano, arXiv:1505.07811 (2015).

[35] D. Gottesman, Phys. Rev. A **57**, 127 (1998).

[36] D. Lidar and T. Brun, *Quantum Error Correction*, 1st ed. (Cambridge University Press, Cambridge, 2013).

[37] K. C. Young, M. Sarovar, J. Aytac, C. M. Herdman, and K. B. Whaley, J. Phys. B **45**, 154012 (2012).

[38] J. W. Harrington, Ph.D. thesis, California Institute of Technology, Pasadena, 2004.

[39] J. Edmonds, J. Res. Nat. Bur. Standards B **69B**, 125 (1965).

[40] R. J. Glauber, J. Math. Phys. **4**, 294 (1963).