2-D Compass Codes

Muyuan Li,[1] Daniel Miller,[2] Michael Newman,[3,*] Yukai Wu,[4] and Kenneth R. Brown[3,†]

[1] *School of Computational Science and Engineering,*
Georgia Institute of Technology, Atlanta, GA 30332, USA
[2] *Institut für Theoretische Physik III, Heinrich-Heine-Universität Düsseldorf, D-40225 Düsseldorf, Germany*
[3] *Departments of Electrical and Computer Engineering, Chemistry,*
and Physics, Duke University, Durham, NC, 27708, USA
[4] *Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA*

(Dated: November 21, 2018)

The compass model on a square lattice provides a natural template for building subsystem stabilizer codes. The surface code and the Bacon-Shor code represent two extremes of possible codes depending on how many gauge qubits are fixed. We explore threshold behavior in this broad class of local codes by trading locality for asymmetry and gauge degrees of freedom for stabilizer syndrome information. We analyze these codes with asymmetric and spatially inhomogeneous Pauli noise in the code capacity and phenomenological models. In these idealized settings, we observe considerably higher thresholds against asymmetric noise. At the circuit level, these codes inherit the bare-ancilla fault-tolerance of the Bacon-Shor code.

I. INTRODUCTION

At the heart of scalable quantum computing is fault-tolerance. The celebrated quantum threshold theorem [1–3] ensures that with sufficiently accurate components, we can perform arbitrarily long quantum computations with polylogarithmic overhead. For physical systems that prefer local interactions, topological codes have emerged as leading candidates for fault-tolerant quantum computation [4–11]. Among these, the surface code is a particularly enticing choice, offering depolarization accuracy thresholds in excess of 15% assuming noiseless error-correction with a planar architecture [12].

Another code family which has generated significant interest are the subsystem Bacon-Shor codes [13]. These codes have many desirable properties: their gauge group is 2-local, measurements can be performed with bare ancilla with virtually no loss in performance [14, 15], and they support fault-tolerance schemes that avoid costly magic-state distillation [16]. Unfortunately, while Bacon-Shor codes offer some of the highest concatenated thresholds [14], they fail to have any threshold when grown as a local family on a lattice without concatenation [17].

In the present article, we investigate codes derived from the quantum compass model on a square lattice [18]. This model provides a natural framework for constructing subsystem and subspace stabilizer codes. These codes can be viewed as different gauge-fixes of the Bacon-Shor code, and so include the (rotated) surface codes, as well as codes with certain topological defects, as members [6, 7]. While we focus on a subfamily of (generalized) surface codes [19] with desirable fault-tolerance properties, the design space for these codes is much larger. Two advantages of this family are its malleability, making it suitable for correcting asymmetric noise, and fault-tolerant bare-ancilla syndrome extraction inherited from measuring along the gauges of the template Bacon-Shor code.

Tailoring codes and decoders to specific noise models can often yield fruitful improvements in threshold scaling. For biased noise models, one can choose fault-tolerance schemes and gates that take advantage of asymmetric error rates [20–22]. Indeed, simply choosing the right decoder can yield tremendous gains in the effective threshold [23–26]. One can even customize codes directly to device level noise [27] or biased error-rates [28, 29]. Such asymmetric noise models are motivated experimentally by the observation that dephasing noise dominates certain quantum computing architectures [30]. By modifying the stabilizers and boundaries of a planar code directly, one can also obtain denser packings of logical qubits [19] and optimized performance with respect to erasures [31].

We similarly modify the geometry of planar codes using the convenient language of compass codes, adapting the density of the syndrome information to better correct biased and spatially dependent Pauli noise. To quantify the value of this adapted syndrome data, we consider randomized [4, 32], minimum-weight perfect matching [4, 12], and union-find [33] decoders that treat X- and Z-type errors independently. We choose different decoders depending on the context, but generally observe similar performance across all three. In particular, we expect that tuning correlated decoders to account for these different noise models will boost code performance even further [23].

The idea is simple: one should tesselate a lattice according to the relative likelihood of errors in that part of the lattice. Although these codes remain local, there is a trade-off between the locality of their stabilizers and their robustness against asymmetric noise, similar to [22]. We analyze these codes numerically in the code capacity and phenomenological noise models, and observe considerably higher thresholds against asymmetric noise in these idealized settings. We leave a discussion of the

[*] michael.newman@duke.edu
[†] kenneth.r.brown@duke.edu

challenges posed by circuit-level noise to the conclusion.

The paper is structured as follows. In Section II, we introduce 2-D compass codes and the noise models we consider. In Section III, we determine the threshold behavior in two randomized families of codes interpolating between Bacon-Shor codes, surface codes, and Shor's code. In Section IV, we quantify the threshold of 2-D compass codes tailored for different asymmetric noise models. In Section V, we demonstrate fault-tolerance for the compass code family using only bare-ancilla syndrome extraction. We conclude with some discussion in Section VI.

II. BACKGROUND

A. 2-D Compass Codes

The quantum compass model on a square lattice is defined generally by the Hamiltonian [34, 35],

$$H = \sum_i \sum_{j \neq L-1} J_X X_{i,j} X_{i,j+1} + \sum_{i \neq L-1} \sum_j J_Z Z_{i,j} Z_{i+1,j}.$$

Here, (i, j) indexes a qubit according to its displacement from the top-left corner of the lattice. Closely connected with this model are Bacon-Shor codes, which are stabilizer subsystem codes with gauge operators realized by the two-body interaction terms of the compass model [13]. This family is a standard example of codes requiring only local measurements for error-correction, but which are *not* topological, with stabilizers that extend the length of the lattice.

The gauge group of a Bacon-Shor code is generated by $\mathcal{G} = \langle X_{i,j} X_{i,j+1}, Z_{i,j} Z_{i+1,j} \rangle$, with stabilizer group $\mathcal{S} = \langle \prod_j X_{i,j} X_{i+1,j}, \prod_i Z_{i,j} Z_{i,j+1} \rangle$. When defined on an $L \times L$ lattice, these $2L$-body stabilizer generators leave us with $(L-1)^2$ gauge degrees of freedom to format as we please.

Our tool for constructing compass codes will be the method of gauge-fixing, by which we can insert gauge transformations into the stabilizer group [5, 36]. Operationally, this corresponds to inserting a gauge operator g into $\mathcal{S}$ and then removing the set of all gauge operators h which anticommute with g from $\mathcal{G}$. Note that as Bacon-Shor codes are CSS codes [37, 38], if we perform fixes of either X- or Z-type, we will preserve the CSS structure.

We focus on a subclass of surface codes that are easy to specify via a coloring of the lattice, see Figure 1. In that graphical language, red plaquettes correspond to "cuts" in the vertical Z-type stabilizers. We index plaquettes according to the index of their top-left qubit; then, for a red plaquette in the (i, j)-th cell of the lattice, we fix the gauge operator $\prod_{k=0}^{i} Z_{k,j} Z_{k,j+1}$, whereas for a blue plaquette, we fix $\prod_{k=0}^{j} X_{i,k} X_{i+1,k}$. Bacon-Shor codes correspond to an empty coloring, whereas the standard surface code correspond to a red and blue checkerboard. Note that a plaquette can be colored either red or blue, but not both, as the resulting stabilizers would not commute.

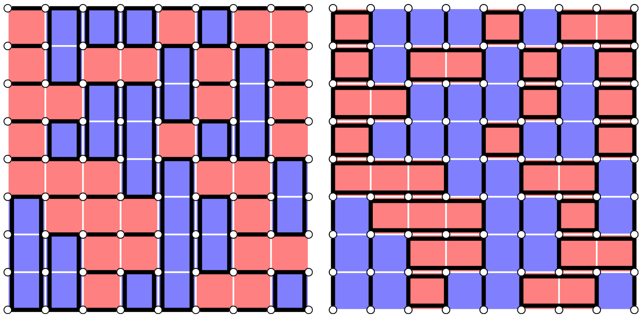

FIG. 1: An example of a compass code on a 9 × 9 lattice. Red and blue plaquettes represent cuts in the Z-type and X-type stabilizers, respectively. The bold lines outline the Z- and X-type stabilizers in the left- and right-side pictures, respectively. As there are no blank plaquettes, all of the gauge degrees of freedom are fixed.

B. Noise Models

In order to carry out numerical simulations, we restrict ourselves to asymmetric Pauli noise. We consider the *η-biased depolarizing channel with error rate p*, defined as

$$\mathcal{E}(\rho) = (1-p)\rho + p_X X \rho X + p_Y Y \rho Y + p_Z Z \rho Z,$$

where $p = p_X + p_Y + p_Z$ and $\eta = p_Z/(p_X + p_Y)$. We make the simplifying assumption that $p_X = p_Y$, matching the definition in [23]. The notion of a physical error rate p is then well-defined, as the fidelity of such a channel to the identity is independent of η.

We consider both symmetric but biased noise, in which each qubit experiences the same error channel, as well as spatially inhomogeneous noise models, in which the error channel may depend on the qubit's position in the lattice. In the latter case, we define the error-rate and bias of the channel on the lattice as a whole as the average fidelity and bias over each qubit in the lattice. Note that we must be careful in comparing such models. For example, concentrating noise on a small subset of qubits might always produce perfectly correctable errors, whereas distributing that noise symmetrically will not.

Finally, depending on the context, we consider either the *code capacity* setting, in which syndrome measurements are assumed perfect, or the *phenomenological setting*, in which the syndrome measurements can be faulty.

C. Decoders

Inherent to any discussion on thresholds is a choice of decoder. We focus on three decoders: randomized, minimum-weight perfect matching, and union-find decoding. We choose among these according to our computational needs.

Each of these decoders corrects X- and Z- type errors independently. Thus, any gains in threshold scaling are a product of the tailored syndrome information alone; it is these gains we aim to quantify. For example, we expect that using a correlated decoder with X- and Y-type stabilizers would augment the threshold further [23].

1. *Decoder Graph*

For independent correction of X- and Z-type errors on a CSS code, the relevant decoding information is captured in the *decoder (hyper-)graph.* The decoder graph for phase errors is constructed by associating a vertex to each X-type stabilizer and a (hyper-)edge to each qubit, where the edges connects all stabilizers incident to that qubit. The decoder graph for bit-flip errors is defined analogously. Note that for the subspace codes we consider, the decoder graph corresponds to a cellulation realizing that code as a homological surface code. For an example of the phase-error decoder graph of a compass code, see Figure 4.

The task of a decoder is then, given some syndrome information in the form of marked nodes, identify the corresponding edge configuration producing those marked nodes *up to homology.* The question we consider in this paper is,

What threshold gains do we obtain by modifying the phase and bit-flip decoder graphs according to asymmetrically distributed edge failure probabilities?

2. *Maximum-Likelihood and Randomized Decoding*

The *maximum-likelihood decoder* is one that takes in syndrome information, and chooses the most likely *error-class* producing that syndrome. Formally, its probability of success is given by $p_{succ} = \sum_s \Pr\big(\overline{E}_s\big)$, where the sum runs over all syndromes s and $\overline{E}_s$ is the most likely error-class conditioned on syndrome s. This decoder will yield optimal thresholds, but is often inefficient to implement.

The related *randomized decoder* is a probabilistic decoder that, conditioned on syndrome s, chooses to correct error-class $\overline{E}$ with probability $\Pr\big(\overline{E}|s\big)$. Thus, its probability of success is given by $p_{succ} = \sum_E \Pr(E) \Pr\big(\overline{E}|s_E\big)$, where s_E is the syndrome associated to E. The randomized decoder can well-approximate the maximum-likelihood decoder in the limit of large lattices. Its threshold can also often be estimated thanks to a well-established connection to statistical mechanics [4, 39].

3. *Minimum-Weight Perfect Matching Decoding*

The minimum-weight perfect matching (MWPM) decoder assigns to each syndrome the error-class corresponding to a most-likely *individual error* producing that syndrome. Its probability of success is then $p_{succ} = \sum_s \Pr\big(\overline{E_s}\big)$ where E_s is a most-likely error producing syndrome s.

This decoder is implemented by constructing a minimum-weight perfect matching among the marked vertices in the decoder graph. The edge weights between two marked vertices correspond to the most probable path between them; for symmetric noise, this is simply the shortest-path, but for asymmetric noise it need not be. Fortunately, Edmond's blossom algorithm runs efficiently on graphs without hyper-edges, taking $O(n^3)$ time on a graph with n nodes [40].

Within the subfamily of compass codes we focus on, each qubit participates in at most two stabilizer generators. As a result, the corresponding decoder graphs contain no hyper-edges, and so compass codes inherit the efficient MWPM decoder of the surface code.

When dealing with boundary conditions, some care must be taken to ensure a perfect-matching exists, since the parity of the marked nodes may no longer be even. We use the techniques of [12] to estimate the logical error rates in the presence of boundaries.

4. *Asymmetrically-Weighted Union-Find Decoding*

The final decoder we will use is an asymmetrically-weighted variant of the union-find decoder recently proposed in [33]. This decoder is guaranteed to performed optimally on errors of weight at most $\lfloor d/2 \rfloor$, and has been shown empirically to perform almost as well as MWPM on toric codes with respect to its threshold.

For simplicity, our simulations are run with a periodic north-south boundary condition, which suffices for threshold comparison [41]. However for completeness, we summarize the decoder on lattices with boundary, along with our modifications to account for asymmetric error rates. Decoding proceeds in two steps.

(1) Asymmetrically-Weighted Syndrome Validation. The first step is (weighted) syndrome validation. In this step, we form an erasure that is consistent with the observed syndrome *and* which accounts for the asymmetric error rates. To satisfy the first property, we save each node as a cluster, growing all clusters with an odd number of marked nodes by half-edges. After each growth, we fuse those clusters that intersect. The cluster growth terminates when each cluster has an even number of marked nodes, indicating that we can form a hypothetical erasure that is consistent with the observed syndrome. Furthermore, we use the weighted-growth heuristic, growing only those odd clusters in each step whose boundary is smallest. We refer to the reader to [33] for a more lengthy description of syndrome-validation.

In the case of a decoder graph with boundary, we no longer have a guarantee that there are an even number of syndromes in our graph. This is because some of the syndromes might condense at the boundary. To accommodate for this, we simply treat the boundary as a sink in which every cluster that fuses with the boundary is assigned an even parity.

After this, syndrome verification concludes by choosing a spanning forest within the clusters. We asymmetrically weight syndrome verification by using Kruskal's algorithm to choose a maximum-weight spanning forest, where each edge is weighted according to its probability of failure [42]. This increases the probability of identifying the erroneous qubits.

(2) Peeling With Boundaries. Having associated to the graph an erasure forest that is consistent with the observed syndrome and asymmetric error rates, the second step is to apply maximum-likelihood erasure decoding in the form of an altered peeling decoder [43].

To each leaf node of the resulting erasure forest, we apply the rules:

(i) If the leaf node is marked, apply a phase flip to the corresponding edge and flip the mark of the connected node. Then remove the leaf node and edge from the erasure tree.

(ii) If the leaf node is unmarked, remove it and the corresponding edge from the erasure tree.

At this stage, we have an erasure forest with no leaf nodes and potentially some open edges connecting to the open boundaries. Unfortunately, these open edges are missing their leaf nodes, and so we cannot peel. In [43], this is avoided by growing the spanning forest so that each tree has at most one open edge, and then peeling towards that edge. However, for asymmetrically-distributed noise, a maximum-weight spanning forest might not take this form.

Instead, we can use dynamic programming to find a maximum-probability error configuration consistent with syndrome information in linear time. Fix any tree inside the forest, with edges weighted according to their error probabilites, and root the tree at any node. Each node in this tree corresponds to a stabilizer, which will be either marked or unmarked. Our aim is to identify a subset S of edges that is both consistent with the syndrome information, and has maximal failure probability. We will then apply our phase-error correction to this set.

We proceed recursively. To each node v, we will associate two values. First, we compute the maximum weight of S_v for the subtree rooted at v over all sub-spanning trees that include the parent edge. Second, we compute the same maximum weight of S_v over all sub-spanning trees that do not include the parent edge. Each of these updates takes constant time, assuming that the children were previously evaluated and that v has bounded degree. Iterating over all vertices in the tree and trees within the forest, this terminates in linear time, and can be used to produce the desired S.

By using a tree structure, the union-find growth algorithm takes $O(n \cdot \alpha(n))$ time, where α is the exceptionally slow-growing inverse Ackermann's function. However, because we find a maximum-weight spanning tree, this variant requires $O(n \log(n))$-time preprocessing. The union-find decoder is the most time-efficient of the three decoders we consider.

For a pictoral skeleton of the decoder, see Figure 2. A comparison of the decoder error-rates with and without the asymmetric alteration on the surface code is shown in Figure 3. There, the error model is generated by choosing a error probability $p_i \in_r [0, 2p]$ for each physical qubit i uniformly at random. The value p_i is passed to the asymmetric decoder to inform Kruskal's algorithm. This additional information results in an improvement on the error-rate, but with little effect on the threshold.

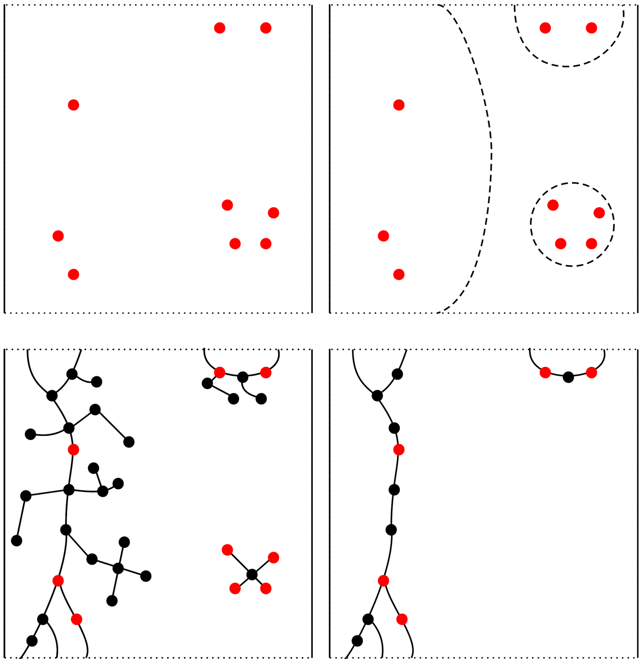

FIG. 2: The square represents the decoder graph (unseen) with north-south open boundary conditions. First, the clusters (dashed enclosures) are grown to an erasure consistent with the (red) marked syndromes. We then find maximum-weight spanning trees, with unmarked syndromes in black. After peeling the leaf nodes, we decode the trees using dynamic programming.

III. THRESHOLD SCALING

Before we consider asymmetric noise models, we ask the more fundamental question, *how does the threshold behave in these compass codes?* In particular, Bacon-Shor codes have no threshold while surface codes boast some of the highest thresholds. Compass codes provide a framework for interpolating between these two, and so we examine the threshold scaling here first.

We use the code's CSS structure to argue directly about phase-flip errors of probability p; bit-flip errors can be decoded analogously and independently. To correct phase errors, the relevant information about the code

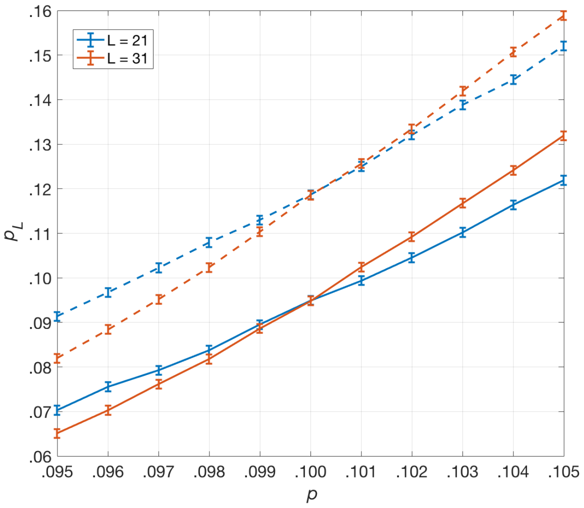

FIG. 3: A comparison of logical error-rates for the asymmetric decoder (solid lines) versus the symmetric decoder (dashed lines). Each data point was generated with 10^6 independent Monte-Carlo trials.

consists of $\mathcal{G}_Z$ and $\mathcal{S}_X$, the Z-type gauge subgroup and the X-type stabilizer subgroup, respectively.

A. Surface-Density Codes

The first family of codes we consider are the (randomized) *surface-density codes*, which interpolate between the Bacon-Shor and surface code. Each code is determined stochastically according to a *surface-density* q_{surf} in the following way. Given a square lattice, for each plaquette of one color in the checkerboard configuration of the surface code, we cut the corresponding X-type stabilizer at that plaquette with probability q_{surf}. Correspondingly, $q_{\mathrm{surf}} = 0$ is equivalent to the Bacon-Shor family (with respect to phase errors) and $q_{\mathrm{surf}} = 1$ is equivalent to the surface code.

1. *Ising Models Associated to Quantum Codes*

We identify the scaling of the threshold with the surface-density under a randomized decoder. To do so, we exploit a connection between the threshold of quantum codes and critical temperatures of associated Ising models [4, 32, 39].

We summarize this connection briefly. Let $\mathcal{G}_0$ be a minimal generating set of $\mathcal{G}_Z$. Let the $g_i \in \mathcal{G}_0$ be indexed by i, and associate to each generator an Ising spin $s_i = \pm 1$. Index the physical qubits by $j \in \{1, \ldots, L^2\}$ and define

$$g_i(j) := \begin{cases} 1 & \text{if } g_i \text{ is supported on site } j \\ 0 & \text{otherwise.} \end{cases}$$

Then for any vector $\tau \in \{+1, -1\}^{L^2}$, we define the classical spin Hamiltonian

$$H_\tau(s) = -\sum_{j=1}^{L^2} \tau_j \prod_{i=1}^{|\mathcal{G}_0|} s_i^{g_i(j)}.$$

For any Pauli Z-error E, define $(\tau_E)_k$ to be -1 if E is supported on site k, and $+1$ otherwise. For physical error-rate p, we can define the virtual temperature β_p according to the *Nishimori line* [44] so that

$$\beta_p := \frac{\log(1-p) - \log(p)}{2}.$$

Define τ to be a quenched vector-valued random variable that takes value τ_E with probability $p^{|E|}(1-p)^{L^2-|E|}$. Under this randomly-disordered statistical model, we can express our success probability using the randomized decoder as

$$\left\langle (1 + \exp\{-\beta_p \cdot (F(\beta_p, \tau_{E \cdot Z_L}) - F(\beta_p, \tau_E))\})^{-1} \right\rangle_p$$

where $\langle \cdot \rangle_p$ denotes the average over the random variable τ distributed according to p, F is the free energy, and Z_L is a Z-type representative of the logical-Z operator. In particular, finding a phase transition of the associated model at (p_c, β_{p_c}) indicates an accuracy threshold at p_c.

For an example of an Ising model associated to a compass code, see Figure 4. Note that, for decoder graphs without hyper-edges, the graph defining the Ising model is dual to the decoder graph.

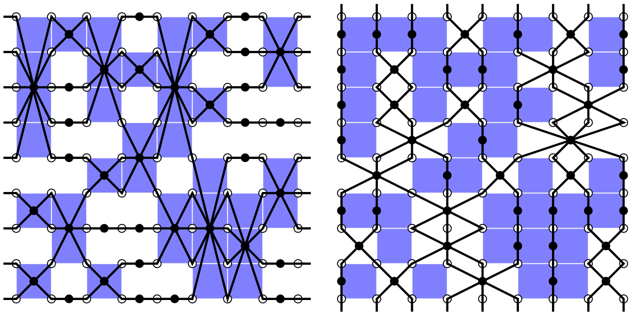

FIG. 4: The left-hand side represents the graph describing the two-body Ising model. The right-hand side represents its dual, the decoder graph. The blue squares represent cuts in the X-type stabilizers on a 9×9 lattice. The connectivity on the left-hand graph determines the sparsity on the right.

2. *Numerical Simulations*

Parameters of the Simulation. We map surface-density codes to their corresponding anisotropic Ising models on random graphs. We generate random samples of the model with the given q_{surf} and p for various system sizes L, with the temperature determined by the Nishimori line according to the disorder parameter p. For each random trial, we use a cluster algorithm [45] and improved estimator to compute the Binder cumulant [46]. Finally, we scan over p (at a separation of 0.1 for $\ln(p)$) and look for a transition point. The system size we use ranges from $L = 5$ to $L = 61$, the number of steps for the cluster update ranges from 10^6 to 10^8, and the number of random trials for each parameter set ranges from 200 to 10^4.

In general, as the transition point p_c increases with q_{surf}, it enhances the frustration in the system and so more steps are needed for convergence. This is verified by the autocorrelation of the observables. However, for larger q_{surf}, the slope of the Binder cumulant U with respect to $-\ln(p)$ also increases. As a result, less samples and smaller system sizes are required to achieve the same level of accuracy.

Numerical Results. Interestingly, simulations suggest that the threshold grows linearly with the surface density, see Figure 5. In particular, a positive density is both necessary and sufficient for the presence of a threshold. The linearity contrasts with the the threshold scaling of the less restricted code family that we consider next.

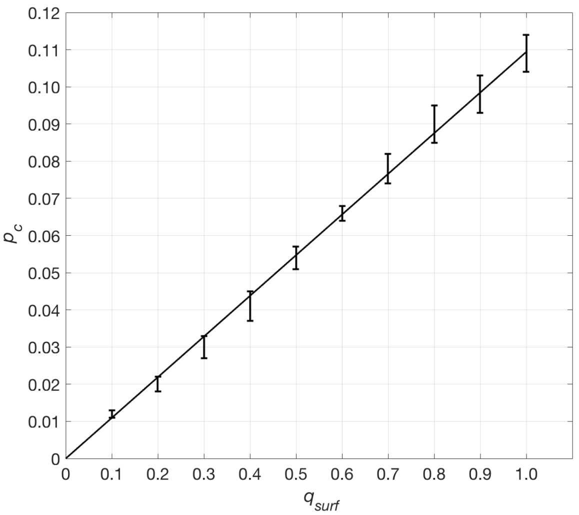

FIG. 5: Scaling of the critical disorder p_c with respect to the surface-density q_{surf}. Autocorrelation is checked using a binning analysis; fit is linear through the origin. The widest error bars are of total width ≈ 1%. At $q_{\mathrm{surf}} = 1$, the results closely match the established critical point at $p_c = 0.1094 \pm 0.0002$ [47].

B. Shor-Density Codes

We next turn our attention to *Shor-density codes*, which form a randomized family of codes that interpolate between Bacon-Shor codes and their full X-type gauge-fix, Shor's subspace code. These codes are defined similarly to surface-density codes according to a new parameter which we call the *Shor-density* q_{shor}. For these codes, X-type stabilizers are cut at *each* plaquette with probability q_{shor}. Thus, $q_{\mathrm{shor}} = 0$ again corresponds to the Bacon-Shor code whereas $q_{\mathrm{shor}} = 1$ corresponds to Shor's subspace code [48].

Of course, the thresholds for such codes are one-sided: more cuts for one type of stabilizer leaves less for the other. Consequently, such codes are best suited for asymmetric noise models. Note that these codes remain local, in the sense that the expected maximum stabilizer weight grows logarithmically in the lattice size for any fixed q_{shor}.

Because the associated graphs to these codes have a richer structure which may hinder convergence of the clustering algorithm, we instead study these codes using the union-find decoder. We generate a new decoder graph and error in each round, and perform 10^6 Monte Carlo trials for each data point. We then exploit the efficiency of the union-find decoder to run 300 Monte-Carlo trials on a 1001 × 1001 lattice to verify the thresholds, which should sharply converge to either $p_L = 0$ or $p_L = 0.5$ about the threshold. This large lattice size is necessary to mitigate the growing finite-size effects.

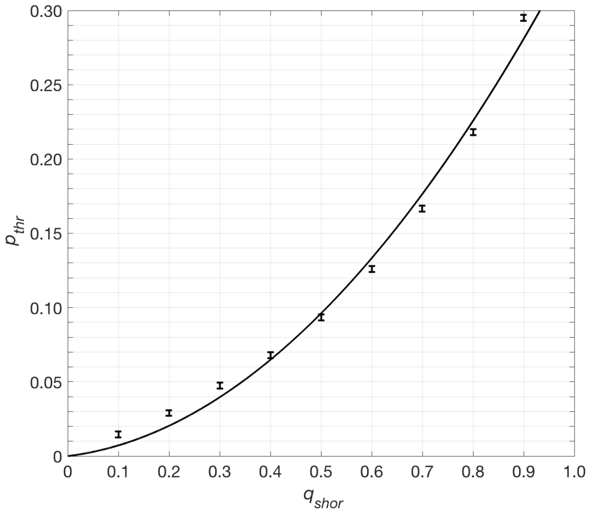

FIG. 6: Scaling of the estimated threshold p with respect to the Shor-density q_{shor}. The fit is quadratic through the origin; the finite-size effects are apparent. All points were obtained on 81 × 81 size lattices, except for $q_{\mathrm{shor}} = 0.9$. To emphasize finite-size effects, this was performed on a 631 × 631 lattice, which is greater than necessary for fault-tolerant computation [8].

The threshold scaling in Figure 6, nearly saturates the zero-rate quantum Gilbert-Varshamov bound [37],

$$H(p_x) + H(p_z) \leq 1,$$

mirroring results obtained on other lattice configurations [49, 50]. One thing to note is the normalizing finite-size effects at very high and very low densities. Note that, at $q_{\text{shor}} = 1$, we essentially have disjoint copies of a repetition code. This has a threshold of 50%, since the union-find decoder behaves optimally on errors of weight $< \lfloor L/2 \rfloor$ [33]. However, we observe a pseudothreshold of $\approx 45\%$ for union-find decoding on a 1001×1001 lattice, matching the analytical solution

$$p_{\text{logical}} = \frac{1}{2}(1 - (1 - 2p_{\text{rep}})^L)$$

where p_{rep} is the probability of failure of a repetition code of length L,

$$p_{\text{rep}} = \sum_{k=\lceil \frac{L}{2} \rceil}^{L} \binom{L}{k} p^k (1-p)^{L-k}.$$

Summary. These simulations suggest that the threshold is determined predominantly by the density of syndrome measurements, rather than their specific configuration, for symmetrically distributed noise. The usual surface code does not far outperform randomized codes of equal density by this metric; it does only slightly, as its symmetry will minimize the number of $\lceil L/2 \rceil$-weight errors that introduce a logical error. This is reinforced by the observation that the threshold appears to scale linearly with surface-density, but is strictly convex with respect to Shor-density.

IV. ASYMMETRIC NOISE

Next, we turn our attention to asymmetric noise. We consider two different types: biased noise that is symmetrically distributed throughout the lattice, and asymmetrically distributed noise. In both cases, we find that substantial gains can be made by tailoring the decoder graphs to the noise directly. We analyze these in both the code capacity and phenomenological models, and compute their thresholds under different noise biases.

A. Biased but Symmetric Noise

For biased noise that is symmetrically distributed, we construct a family of compass codes we call *elongated codes*. These codes are defined by a parameter $\ell \in \mathbb{N}^+$ we call the *elongation* of the code, and are constructed by cutting the Z-stabilizers at the (i, j)-th plaquettes for all $i - j \equiv 0 \pmod{\ell}$. The X-stabilizers are then cut at all remaining plaquettes, resulting in a subspace code. This is similar to the approaches of [14, 22], which consider concatenations with different phase-flip repetition codes.

Under this definition, we obtain Shor's code for $\ell = 1$ and the surface code for $\ell = 2$. For $\ell > 2$, we obtain an asymmetrization of Kitaev's toric code in the bulk with extended 2ℓ-body plaquette operators. This family illustrates a simple compass code is well-equipped to correct asymmetric noise, while sacrificing somewhat in locality.

It is worth noting that choosing asymmetric lattice dimensions as in [28] may alter the logical error-rate of a code family, but will not change the threshold, as it is a property of the bulk. Thus, the elongation of the code refers to a stretching of the bulk stabilizer geometry, not the lattice itself.

As the elongation grows, finite-size effects play a greater role. As such, we use MWPM decoding to perform simulations on smaller lattices at lower elongations, and union-find decoding to test larger lattices. While these larger lattices also suffer from finite-size effects, we use the efficiency of union-find decoding to simulate lattices of between 10^3 and 10^4 qubits, which is the estimated code-size required for full-scale fault-tolerant computation [8].

Furthermore, we estimate the phenomenological threshold by simulating $(2+1)$-D elongated codes. For a physical lattice of linear size L, this corresponds to performing L rounds of faulty syndrome extraction, followed by an ideal round, and then decoding. The corresponding decoder graph is then $L + 1$ copies of the initial decoder graph with L time-like slices of edges connecting the corresponding vertices in each space-like slice. These time-like edges represent faulty measurements.

Although the size of each stabilizer is independent of the lattice size, we scale the probability of failure for each stabilizer linearly with its weight. We assume the usual phenomenological normalization that plaquette stabilizers are faulty at the physical error rate p. Despite some increasing stabilizer weights, we observe substantial threshold gains in both the code capacity *and* phenomenological models.

Tables I and II show the code-capacity thresholds using the MWPM and union-find decoder, respectively, while Table III shows the phenomenological threshold using the union-find decoder. In these tables, η_{opt} refers to the optimal bias that realizes the threshold p_{thr}, while η_* is the bias above which the codes will outperform the surface code.

ℓ	η_{opt}	p_{thr}	η_*	p_z	p_x
2	0.5	15.5%	N/A	$10.3\% \pm 0.2\%$	$10.3\% \pm 0.2\%$
3	1.67	17.9%	1.39	$14.1\% \pm 0.3\%$	$6.5\% \pm 0.2\%$
4	3.00	20.0%	2.10	$17.5\% \pm 0.2\%$	$5.0\% \pm 0.2\%$
5	4.26	21.6%	2.78	$19.5\% \pm 0.1\%$	$4.1\% \pm 0.1\%$
6	5.89	22.8%	3.70	$21.1\% \pm 0.1\%$	$3.3\% \pm 0.1\%$

TABLE I: Thresholds for the MWPM decoder in the code-capacity model. Simulations were done on lattices of size at most 17×17.

Notably, a relatively *smaller* noise bias is required to outperform the surface code in the phenomenological setting. Unsurprisingly, the MWPM outperforms the union-find decoder as a whole, but surprisingly, displays lower

ℓ	η_{opt}	p_{thr}	η_*	p_z	p_x
2	0.5	15.0%	N/A	10.0% ± 0.2%	10.0% ± 0.2%
3	1.41	16.9%	1.14	13.4% ± 0.3%	7.0% ± 0.2%
4	2.40	18.4%	1.78	15.7% ± 0.2%	5.4% ± 0.2%
5	3.45	19.6%	2.41	17.4% ± 0.1%	4.4% ± 0.1%
6	4.45	20.7%	2.95	18.8% ± 0.1%	3.8% ± 0.1%
7	5.62	21.9%	3.55	20.2% ± 0.1%	3.3% ± 0.1%
8	6.23	22.8%	3.84	21.2% ± 0.1%	3.1% ± 0.1%
9	7.29	23.7%	4.17	22.2% ± 0.1%	2.9% ± 0.1%
10	8.36	24.0%	4.77	22.7% ± 0.1%	2.6% ± 0.1%
20	20.7	28.3%	10.5	27.6% ± 0.1%	1.3% ± 0.1%
50	55.3	33.8%	24.0	33.5% ± 0.1%	0.6% ± 0.1%

TABLE II: Thresholds for the union-find decoder in the code-capacity model. Simulations were done on lattices of size at most 81 × 81.

ℓ	η_{opt}	p_{thr}	η_*	p_z	p_x
2	0.5	3.98%	N/A	2.65% ± 0.2%	2.65% ± 0.2%
3	1.20	4.45%	0.99	3.4% ± 0.2%	2.0% ± 0.2%
4	1.88	4.60%	1.49	3.8% ± 0.2%	1.6% ± 0.2%
5	2.73	4.85%	2.06	4.2% ± 0.2%	1.3% ± 0.2%
6	3.17	5.00%	2.32	4.4% ± 0.2%	1.2% ± 0.2%

TABLE III: Thresholds for the union-find decoder in the phenomenological model. Simulations were done on lattices of size at most 35 × 35 × 35.

thresholds on lattices comprised of higher-weight stabilizers. This suggests that union-find decoding may better exploit the degeneracy of certain lattices; in particular, one should use MWPM for Z-type errors and union-find decoding for X-type errors on elongated lattices. Our estimates for established surface code thresholds match those found in [12] at 10.3% for MWPM decoding and in [33] at 9.95% and 2.65% for union-find decoding in 2- and $(2+1)$-D, respectively.

B. Spatially Dependent Noise

We conclude by considering noise that is asymmetrically distributed throughout the lattice. To illustrate the idea, we focus on a simple noise model in which dephasing noise decays linearly from the right-hand side of the lattice according to the function $p_z(i,j) = (w(j/L) + (1 - w)(1 - j/L)) \cdot p_{\text{tot}}/2$. Here, i and j are the coordinates of a qubit, L is the linear size of the lattice, and w is a constant that determines the degree of incline. We further assume that $p_x = p_{\text{tot}}/2$, so that p_{tot} is the total infidelity of the channel. Note that the average bias between the dephasing noise and bit-flip noise is symmetric.

The idea is simple: when the noise is distributed asymmetrically, the stabilizer information can be chosen to match the noise. Intuitively, lower weight stabilizers add more error-information about the qubits nearby. With this in mind, we define a randomized family of codes we call *(p_z-)tailored codes*. At each plaquette, we choose to cut the corresponding X-type stabilizer with probability $2p_z(i,j)/p_{\text{tot}}$, where i, j are the coordinates of the upper-left qubit at that plaquette. Then, in the presence of a high amount of dephasing noise, many low-weight X-type stabilizers will appear to aide in error-correction.

We observe that the tailoring of these codes to the noise model can augment error rates, see Figure 7. It is worth noting, however, that simply weighting the probability of each cut according to the surrounding qubits may not always be the optimal strategy. In particular, in the low error-rate limit, this will become an optimization problem that seeks to minimize the weights of uncorrectable paths of length $\lceil \frac{L}{2} \rceil$ in the decoder graph.

V. FAULT-TOLERANCE WITH BARE ANCILLA

One of the major advantages that comes with the locality of the Bacon-Shor code is fault-tolerant bare-ancilla syndrome extraction [14, 15]. Although this extraction scheme is the simplest and least resource-intensive, most codes incur some loss in effective distance due to high-weight correlated errors produced by errors on the ancilla. For the standard and rotated surface codes, these “hook” errors can be carefully designed to ensure no significant loss in performance [4, 6].

In the compass code framework, this resilience to correlated errors is a general phenomenon resulting from measuring stabilizers along the Bacon-Shor gauge operators. Using such a syndrome extraction scheme on any gauge-fix of the Bacon-Shor code, any collection of $d-1$ faults in the circuit produce an error of the form EG, where $|\text{supp}(E)| \leq d-1$ is minimal and G is a gauge operator of the initial Bacon-Shor code.

Divide the generators of the stabilizer group of any compass code into $\mathcal{S} = \langle \mathcal{S}_B, \mathcal{S}_F \rangle$, where $\mathcal{S}_B$ are the stabilizer generators of the Bacon-Shor code and $\mathcal{S}_F$ are those gauge operators that have been fixed. Then, for any error EG resulting from $d-1$ faults in the circuit, if $|\text{supp}(E)| = 0$, then either $G \in \mathcal{S}_F$ or there exists an $S \in \mathcal{S}_F : SG = -GS$. Else if $0 < |\text{supp}(E)| < d$, then there exists an $S \in \mathcal{S}_B : SE = -ES$. Since S must also commute with any gauge operator G, it follows that EG is detectable. Thus, any error resulting from $\leq d-1$ faults during syndrome extraction is either detectable, or trivial.

This demonstrates that there exists fault-tolerant de-

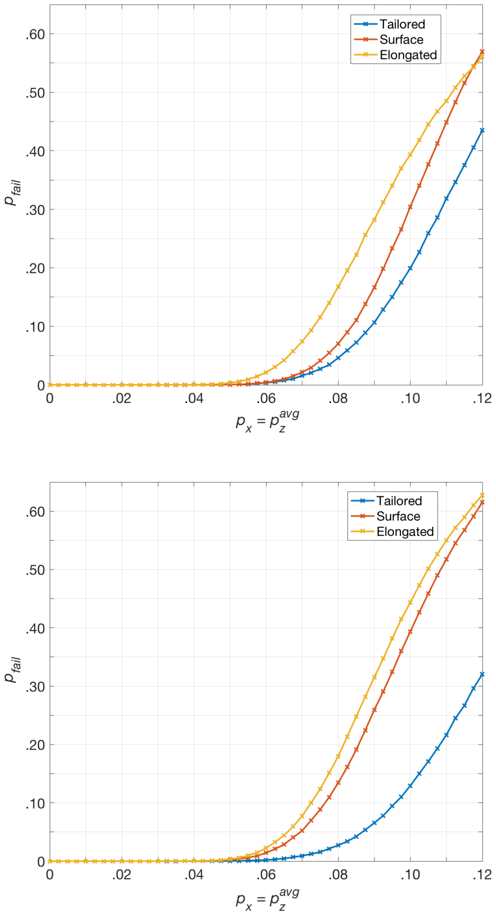

FIG. 7: Error-rates for gradually ($w = 0.25$, top) and steeply ($w = 0.10$, bottom) inclined linear noise, computed on a lattice of size 33×33 using union-find decoding in the high-noise regime. Here, p_{fail} represents the total probability of a failure in either the X- or Z-type decoders.

coding without a loss in effective distance. However, it is *not* necessarily maximum-likelihood decoding on the memory. One simple counter-example is Shor's code, where a single well-placed ancilla error can effect a weight d memory error that maximum-likelihood will misdiagnose as a weight $d-1$ memory error, resulting in failure. The above does imply that performing MWPM with respect to linear-probability faults in the decoder graph is fault-tolerant. Introducing these faults amount to triangulating the decoding graph, similar to hook errors in the surface code case [4, 12]. Determining circuit-level compass code performance in this model is the subject of future inquiry [51].

VI. CONCLUSIONS

In this article, we have described an ansatz for designing planar codes stemming from the 2-D compass model. We have provided evidence that simple subfamilies of this class may be useful for correcting biased noise in idealized code capacity and phenomenological noise models, particularly if that bias is distributed geometrically. In particular, one can bias the stabilizers locally towards correcting a certain error-type.

There are two central challenges for these codes in the more realistic circuit-level noise model. Although these codes are still local, there is a trade-off between the bias of the codes and the locality of the stabilizer measurements. We have demonstrated that fault-tolerant measurement in Bacon-Shor [14, 15] and surface codes [4, 6] using bare ancilla can be adapted to the compass model, if measurements are performed in the correct order. Nevertheless, these correlated errors will deteriorate code performance as higher-weight stabilizer outcomes become less reliable. This might be mitigated by using other flag-type schemes, or by preserving some gauge degrees of freedom. We would expect that these gains would persist, but at the expense of higher bias and code overhead. As such, we leave a more involved circuit-level analysis to future work.

The second concern is whether the biased noise model itself can persist at the circuit level. To remain experimentally motivated, one must choose operations that preserve the bias [20, 22, 23]. Consequently, the construction of *simple* and *bias-preserving* fault-tolerant gadgets is key to utilizing asymmetric noise.

Finally, we have only narrowly broached the design space offered by these codes. Exploring different configurations according to other geometrically-defined noise [31], generalizing to codes defined on the 3-D compass model, and using correlated decoders [23–26, 52] are all avenues to explore. More generally, finding other LDPC constructions adapted to biased noise may give the best of both worlds, mitigating the overhead of asymmetrization while taking advantage of the bias.

VII. ACKNOWLEDGEMENTS

The authors thank Dave Bacon, Nicolas Delfosse, and Pavithran Iyer for useful discussions, and Luming Duan for providing computational resources for simulations on the Flux cluster at the University of Michigan. Additionally, they thank anonymous referees for their helpful comments. This research was supported in part by NSF (1717523), ODNI/IARPA LogiQ program (W911NF-10-1-0231), ARO MURI (W911NF-16-1-0349), EPiQC - an NSF Expedition in Computing (1730104), and the Alexander von Humboldt Foundation.

[1] P. Aliferis, D. Gottesman, and J. Preskill, Quantum Inf. Comput. **6**, 97 (2005).

[2] E. Knill, R. Laflamme, and W. Zurek, arXiv preprint quant-ph/9610011 (1996).

[3] D. Aharonov and M. Ben-Or, in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (ACM, 1997), pp. 176–188.

[4] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Journal of Mathematical Physics **43**, 4452 (2002).

[5] H. Bombín, New Journal of Physics **17**, 083002 (2015).

[6] Y. Tomita and K. M. Svore, Physical Review A **90**, 062320 (2014).

[7] T. J. Yoder and I. H. Kim, Quantum **1**, 2 (2017).

[8] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Physical Review A **86**, 032324 (2012).

[9] B. M. Terhal, Reviews of Modern Physics **87**, 307 (2015).

[10] T. Obrien, B. Tarasinski, and L. DiCarlo, npj Quantum Information **3**, 39 (2017).

[11] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S.-T. Wang, L. Duan, and K. R. Brown, New Journal of Physics **20**, 043038 (2018).

[12] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, arXiv preprint arXiv:0905.0531 (2009).

[13] D. Bacon, Physical Review A **73**, 012340 (2006).

[14] P. Aliferis and A. W. Cross, Physical Review Letters **98**, 220502 (2007).

[15] M. Li, D. Miller, and K. R. Brown, arXiv preprint arXiv:1804.01127 (2018).

[16] T. J. Yoder, arXiv preprint arXiv:1705.01686 (2017).

[17] F. Pastawski, A. Kay, N. Schuch, and I. Cirac, arXiv preprint arXiv:0911.3843 (2009).

[18] Z. Nussinov and J. Van Den Brink, Reviews of Modern Physics **87**, 1 (2015).

[19] N. Delfosse, P. Iyer, and D. Poulin, arXiv preprint arXiv:1606.07116 (2016).

[20] P. Aliferis and J. Preskill, Physical Review A **78**, 052331 (2008).

[21] P. Webster, S. D. Bartlett, and D. Poulin, Physical Review A **92**, 062309 (2015).

[22] A. M. Stephens, W. J. Munro, and K. Nemoto, Physical Review A **88**, 060301 (2013).

[23] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, Physical review letters **120**, 050505 (2018).

[24] N. Delfosse and J.-P. Tillich, in *Information Theory (ISIT), 2014 IEEE International Symposium on* (IEEE, 2014), pp. 1071–1075.

[25] N. H. Nickerson and B. J. Brown, arXiv preprint arXiv:1712.00502 (2017).

[26] A. S. Darmawan and D. Poulin, arXiv preprint arXiv:1801.01879 (2018).

[27] A. Robertson, C. Granade, S. D. Bartlett, and S. T. Flammia, Physical Review Applied **8**, 064004 (2017).

[28] J. Napp and J. Preskill, arXiv preprint arXiv:1209.0794 (2012).

[29] P. Brooks and J. Preskill, Physical Review A **87**, 032310 (2013).

[30] P. Aliferis, F. Brito, D. P. DiVincenzo, J. Preskill, M. Steffen, and B. M. Terhal, New Journal of Physics **11**, 013061 (2009).

[31] N. Delfosse, P. Iyer, and D. Poulin, arXiv preprint arXiv:1611.04256 (2016).

[32] H. Bombín, Physical Review A **81**, 032301 (2010).

[33] N. Delfosse and N. H. Nickerson, arXiv preprint arXiv:1709.06218 (2017).

[34] J. Dorier, F. Becca, and F. Mila, Physical Review B **72**, 024448 (2005).

[35] K. Kugel and D. Khomskii, Zh. Éksp. Teor. Fiz **64**, 1429 (1973).

[36] A. Paetznick and B. W. Reichardt, Physical review letters **111**, 090505 (2013).

[37] A. R. Calderbank and P. W. Shor, Physical Review A **54**, 1098 (1996).

[38] A. M. Steane, Physical Review Letters **77**, 793 (1996).

[39] A. Kubica, M. E. Beverland, F. Brandão, J. Preskill, and K. M. Svore, Physical Review Letters **120**, 180501 (2018).

[40] J. Edmonds, in *Classic Papers in Combinatorics* (Springer, 2009), pp. 361–379.

[41] A. G. Fowler, Physical Review A **87**, 062320 (2013).

[42] J. B. Kruskal, Proceedings of the American Mathematical society **7**, 48 (1956).

[43] N. Delfosse and G. Zémor, arXiv preprint arXiv:1703.01517 (2017).

[44] H. Nishimori, Journal of the Physical Society of Japan **55**, 3305 (1986).

[45] V. S. Dotsenko, W. Selke, and A. Talapov, Physica A: Statistical Mechanics and its Applications **170**, 278 (1991).

[46] K. Binder, Zeitschrift für Physik B Condensed Matter **43**, 119 (1981).

[47] A. Honecker, M. Picco, and P. Pujol, Physical Review Letters **87**, 047201 (2001).

[48] P. W. Shor, Physical Review A **52**, R2493 (1995).

[49] K. Fujii and Y. Tokunaga, Physical Review A **86**, 020303 (2012).

[50] B. Röthlisberger, J. R. Wootton, R. M. Heath, J. K. Pachos, and D. Loss, Physical Review A **85**, 022313 (2012).

[51] S. Huang and K. R. Brown, In preparation (2018).

[52] N. Maskara, A. Kubica, and T. Jochym-O'Connor, arXiv preprint arXiv:1802.08680 (2018).