Kitaev's $\mathbb{Z}_d$-code threshold estimates

Guillaume Duclos-Cianci[*] and David Poulin[†]
Département de Physique, Université de Sherbrooke, Sherbrooke, Québec J1K 2R1, Canada

(Received 18 February 2013; published 27 June 2013)

We study the quantum error correction threshold of Kitaev's toric code over the group $\mathbb{Z}_d$ subject to a generalized bit-flip noise. This problem requires special decoding techniques, and for this purpose we generalize the renormalization-group method we introduced previously [G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010) and IEEE Information Theory Workshop, Dublin (2010), p. 1] for $\mathbb{Z}_2$ topological codes.

DOI: 10.1103/PhysRevA.87.062338 PACS number(s): 03.67.Pp

I. INTRODUCTION

Kitaev's topological code (KTC) [1] on qubits is the archetypical topological code and has been extensively studied. As explained in Kitaev's original paper [1], this construction applies to any group. Much less is known about these generalizations, and in this paper we investigate the quantum error correction (QEC) thresholds of the KTCs built with the groups $\mathbb{Z}_d$ [2–5], where $d \geqslant 2$. We label these $\mathbb{Z}_d$-KTCs, so the original code on qubits corresponds to $\mathbb{Z}_2$-KTCs. These codes are well suited for arrays of d-level quantum systems, qudits. Qudits occur very naturally in nature, e.g., superconducting qubits [6], Rydberg atoms [7], and orbital states of light [8], have multiple levels and their Hilbert space needs to be truncated to obtain qubits.

As explained in [9], $\mathbb{Z}_2$-KTCs can be decoded by a binary perfect matching algorithm [10], since every particle is its own antiparticle in this model. Because this is not the case for $d > 2$, other techniques are required and for this purpose we generalize the renormalization-group (RG) soft decoder that we introduced in [11] and [12]. Our numerical simulations show that the threshold increases monotonically with d and appears to follow the general trend of the qudit hashing bound.

This paper is organized as follows. First, we introduce a generalized Pauli group (see [13] and [14] for more details), stabilizer codes, and $\mathbb{Z}_d$-Kitaev's toric code. Next, we briefly review the decoding problem of these systems and show how the RG decoder applies in this case. Finally, we present the numerical results and close with a discussion.

II. $\mathbb{Z}_d$ GENERALIZATION OF KITAEV'S TORIC CODE

In this section, we review the definition of $\mathbb{Z}_d$-KTCs and show that many features of KTCs on qubits extend to them. Since we will be working with qudits, we introduce a generalized Pauli group. The Hilbert space of a qudit, $\mathcal{H}_d$, is spanned by the states $\{|0\rangle,|1\rangle,\ldots,|d-1\rangle\}$. We define the operators X and Z such that

$$X|g\rangle = |g \oplus 1\rangle, \quad Z|g\rangle = \omega^g|g\rangle, \tag{1}$$

where $0 \leqslant g < d$, "$\oplus$" denotes addition modulo d, and $\omega = \mathrm{e}^{i2\pi/d}$. The generalized Pauli group is generated by X, Z, and a phase, i.e., $\mathcal{P}_d = \langle \omega, X, Z\rangle$ if d is odd and $\mathcal{P}_d = \langle \omega^{1/2}, X, Z\rangle$ if d is even (XZ has order $2d$ in this case). From the definitions of Eq. (1), we deduce the following properties:

$$\begin{aligned} X^a|g\rangle &= |g \oplus a\rangle, \quad Z^a|g\rangle = \omega^{ag}|g\rangle, \\ ZX|g\rangle &= \omega XZ|g\rangle, \quad Z^aX^b|g\rangle = \omega^{ab}X^bZ^a|g\rangle. \end{aligned} \tag{2}$$

Finally, we define the n-qudit Pauli group $\mathcal{P}_d^n \equiv \mathcal{P}_d^{\otimes n}$ as the n-fold tensor product of $\mathcal{P}_d$.

The stabilizer group $\mathcal{S}$ is an abelian subgroup of $\mathcal{P}_d^n$. The code is defined as the simultaneous $+1$ eigenspace of all stabilizers. Note that even though the generalized Pauli operators are unitary, they are not Hermitian in general so do not correspond to physical observables. However, the operator $\frac{1}{2}(s + s^\dagger)$ is Hermitian and can be measured. Since s has eigenvalues ω^a, $\frac{1}{2}(s+s^\dagger)$ has eigenvalues $\frac{1}{2}(\omega^a + \omega^{-a}) = \cos(2\pi a/d)$, which are in one-to-one correspondence with the eigenvalues of s.

With these definitions in place, we present a generalization of the KTC on qudits, which we call $\mathbb{Z}_d$-KTC, using Kitaev's original construction [1] on the cyclic groups $\mathbb{Z}_d$ with $d \geqslant 2$. The system is a square lattice of linear size L with periodic boundary conditions. Each edge is occupied by a qudit, so there are $n = 2L^2$ qudits in total. We define vertex operators A_v and plaquette operators B_p as shown in Fig. 1. There is one such operator for each vertex and each plaquette. We verify that they commute using the last line of Eq. (2). These operators generate the stabilizer group $\mathcal{S} = \langle A_v, B_p\rangle$ and the code is spanned by the simultaneous $+1$ eigenstates of the stabilizer generators.

Figure 2 illustrates how applying some power of X on a code state creates defects on the lattice. Indeed, X^a applied on some qudit does not commute with the two plaquette operators involving that qudit. The eigenvalues of the plaquettes to the north or east of the error will change from 1 to ω^a, and those of the plaquettes to the south or west will change from 1 to ω^{-a}. One can show that the defects thus created are topological charges; we associate the charge a with a plaquette defect corresponding to an eigenvalue ω^a of that plaquette. With this choice of labeling, the charge group restricted to plaquettes is $\mathbb{Z}_d$ with addition.

From these simple facts, it follows that string operators can be built with defects attached only to their end points (these strings actually live on the dual lattice, just like in KTC). This requires a careful choice of the powers of X on the qudits along the string such that the total charge in each plaquette is 0 except on its end points. For instance, one can adopt the convention that power a is used when heading north or east,

[*] Guillaume.Duclos-Cianci@USherbrooke.ca
[†] David.Poulin@USherbrooke.ca

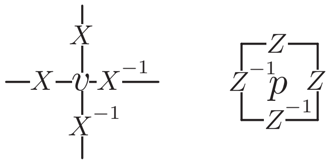

FIG. 1. $\mathbb{Z}_d$-KTC stabilizer generators. With each vertex v, we associate an operator A_v (left) and with each plaquette p, we associate an operator B_p (right).

and $-a$ when heading south or west. Moreover, we can verify that nontrivial cocycles (loops on the dual lattice; see Fig. 2) of any power of X obeying this convention commute with the stabilizer. These operators are not in the stabilizer, as all the vertex generators in Fig. 1 are trivial cocyles. It follows that such operators, e.g., the one at the bottom of Fig. 2, are logical operators (for any value of a).

A similar analysis holds for defects created by powers of Z operators. In this case, the defects live on vertices and string operators, on the direct lattice. Also, nontrivial cycles of any power of Z are logical operators. From the form of the logical operators, we directly deduce that there are two qudits encoded in the code space. Again, this is analogous to the case of KTC.

III. $\mathbb{Z}_d$-KTC DECODING

We are now interested in the problem of error correcting $\mathbb{Z}_d$-KTCs for $d > 2$. In our study, we consider a simple noise model that generalizes the independent symmetric bit-flip channel to qudits: with probability $1 - p_{\text{phys}}$, the qudit remains unaffected and with probability p_{phys}, we apply at random (uniformely distributed) one of $X, X^2, \ldots, X^{d-1}$ (see the Appendix). Suppose an error $E \in \mathcal{P}_d^n$ occurs on a code state. It creates defects on the lattice, and by measuring the eigenvalues of every $\frac{1}{2}(A_v + A_v^\dagger)$ and $\frac{1}{2}(B_p + B_p^\dagger)$ we can learn the position and charge of each defect. The role of the decoder is to bring the system back in the code space by applying a correcting Pauli operator, $C \in \mathcal{P}_d^n$. However, care must be taken in choosing an appropriate correcting operation. Indeed, if the operator CE resulting from the combination of the error and the recovery is an element of $\mathcal{S}$, the state is unaffected. However, if CE is a nontrivial logical operator, then the system is returned to the code space but potentially in a different code state, so the information is corrupted.

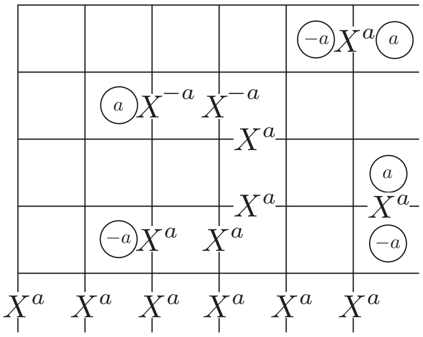

FIG. 2. Plaquette defects created by the application of some power of X. The values a ($-a$) in the plaquettes are such that the eigenvalue of the corresponding B_p is ω^a (ω^{-a}). By choosing appropriately the powers of X, we can build string operators with defects only on their end points. Nontrivial cocyles of X^a correspond to $\overline{X}^a$ logical operators.

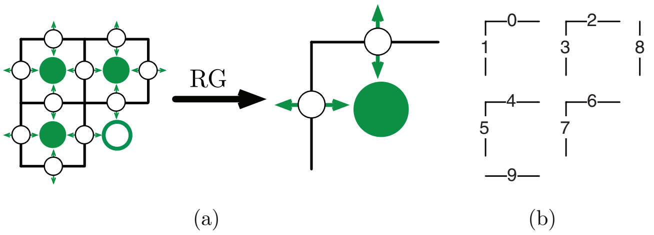

FIG. 3. (Color online) (a) The lattice is cut into unit cells containing 10 qudits (edges). The renormalization process takes the defect configuration and the noise model on a unit cell as inputs and outputs a two-qudit distribution (white circles) which corresponds to a probability on the charge flow through the corresponding boundaries. Filled (green) circles represent plaquette operators. The plaquette corresponding to the filled (green) circle is replaced by the product of all four plaquettes of the unit cell, such that its eigenvalue gives the total charge of the cell. This value is only going to be used in the next round of RG [larger filled (green) circle]. (b) Labeling convention for qudits in Eq. (3).

Any operator $E \in \mathcal{P}_d^n$ creating the measured configuration of defects is a potential error. However, we classify these operators by their logical effect on the code space: two operators E_1, E_2 with the same configuration of defects are equivalent iff $E_2^\dagger E_1$ has a trivial effect on the code, i.e., $E_1 \sim E_2$ iff $E_2^\dagger E_1 \in \mathcal{S}$. Note that since E_1 and E_2 lead to the same defect configuration, $E_2^\dagger E_1$ creates no defect, or equivalently, E_1 creates some defects that $E_2^\dagger$ annihilates.

Given a measured defect configuration, the decoder seeks the best correction among the set of all errors which would lead to this defect configuration. One strategy would be to identify the error from this set that has the highest probability $\mathcal{P}(E)$, where the probability of an error is specified by the physical noise model, in our case the symmetric bit-flip channel. This turns out not to be optimal, however, because some errors have equivalent effects on all code states. Thus, the decoder should instead seek the most likely equivalence class of errors. The probability of an equivalence class of errors is obtained by summing over the probability of each error within a class. Given these probabilities, the optimal correction consists of applying the adjoint of any representative of the class with maximal probability.

IV. RG DECODER GENERALIZATION TO $\mathbb{Z}_d$-KTC

Unfortunately, the above procedure cannot be realized efficiently in general since the number of errors in each equivalence class scales exponentially with the system size. In [11,12], we introduced a RG soft decoder (RG decoder) that efficiently approximates the exact calculation (see [15] for a related scheme). The general idea is to cut the lattice into small unit cells (e.g., 2 × 2 sublattices) and to "distill" from each cell an effective two-qubit noise model [cf. Fig. 3(a)]. This is realized by keeping track of the flow of charges through the cell and summing over the microscopic details leading to this flow. This has the effect of shrinking the lattice linear size by a constant factor (k for cells of size $k \times k$). Recursing on this process, one can shrink the lattice to a constant, manageable, size where the exact decoding can be performed. With appropriate simple modifications, this method can be used for charges over $\mathbb{Z}_d$.

There are two technical difficulties in realizing the above heuristic description, which are both caused by charge conservation. First, because the unit cells share boundaries, the flow of charge through one boundary of a cell should be equal and opposite to the flow of charge of the corresponding boundary of the neighboring cell. Thus, the variables corresponding to charge flows in each cell are highly constrained. This problem is easily circumvented by keeping track only of the flow of charge through the northern and the western boundary of each cell, i.e., by eliminating this redundancy.

Second, the sum of the charge flow through the boundaries of a cell must be equal to its total charge, revealed by the syndrome measurement. This once again sets a hard constraint between the variables corresponding to the charge flows, which would, in principle, require a probability distribution that correlates all the variables of the system. This cannot be realized efficiently, so we must resort to some approximation. As the first approximation, we choose to ignore the cross-cell correlations and keep only marginal probabilities on the flows associated with a given cell (we keep a probability distribution that involves the northern and western boundary only). To diminish the effect of these correlations we are neglecting, we let the charge inside a unit cell fluctuate. For each unit cell, we measure all but one of the plaquettes it encloses. This remaining plaquette thus determines the total charge of the unit cell, and indeed we can substitute the corresponding stabilizer generator with a plaquette enclosing the entire unit cell (obtained by multiplying all the plaquette operators contained in the unit cell). This new stabilizer generator represents a renormalized charge.

This procedure is illustrated in Fig. 3(a), where filled (green) circles represent plaquettes that are measured and the bold-outline circle represents the plaquette that is left fluctuating. This circle is replaced by the larger, renormalized filled (green) circle (at the right) that is used in the next RG step. The white circles in this figure each represent a probability distribution on charge flow or, equivalently, a two-qudit probability distribution. Thus, after one round of RG, we are left with a smaller lattice and both renormalized charges and renormalized noise models.

Equation (3) lists a set of generators for all X operators living on a unit cell [see Fig. 3(b) for labeling]. This basis is used to decompose any X-type error contained on the unit cell. These operators are defined in accordance to the renormalization process itself as we now explain. The T_i operators are used to build a representative error with the appropriate defect configuration. Indeed, only the T_i operators of Eq. (3) do not commute with all three plaquette operators in the unit cell [filled (green) circles in Fig. 3(a)]. Label the defect configuration on a unit cell $\vec{a} = (a_0,a_1,a_2)$, where a_0 is the charge of the northwest plaquette, a_1 is the charge of the northeast one, and a_2 is the charge of the southwest one. Then the Pauli operator $t(\vec{a}) = T_0^{a_0} T_1^{a_1} T_2^{a_2}$ creates the defect configuration $\vec{a}$. Moreover, given a defect configuration $\vec{a}$, every potential error has to contain this product in its decomposition on basis Eq. (3) since only the T_i operators do not commute with plaquettes. The L_i operators characterize the flow of charge through the northern and western boundaries, so the two-qudit ouput distribution of a RG round is precisely the probability distribution over these two operators. The S_i operators are stabilizer operators (or parts of stabilizer generators supported on the unit cell). They only deform strings without changing their defect configuration or their associated charge flow. Finally, the E_i operators correspond to charge flowing through the southern and eastern boundaries into the plaquette operator that is left out. Thus, they are responsible for the charge fluctuation inside the unit cell and they are summed over.

$$\begin{aligned} S_0 &= X_0 X_2^{-1} X_3^{-1}, \quad T_0 = X_4 X_7^{-1}, \\ S_1 &= X_1 X_4^{-1} X_5^{-1}, \quad T_1 = X_6, \\ S_2 &= X_3 X_4 X_6^{-1} X_7^{-1}, \quad T_2 = X_7^{-1}, \\ E_0 &= X_6 X_8, \quad L_0 = X_2 X_6, \\ E_1 &= X_7^{-1} X_9^{-1}, \quad L_1 = X_5 X_7. \end{aligned} \tag{3}$$

With these definitions, we can formally describe an RG round that starts with a defect configuration $\vec{a}$ and computes the marginal probability of each $l \in \langle L_0, L_1 \rangle$ conditioned on the measured defect configuration,

$$\mathcal{P}(l) = \sum_{e \in \langle E_0, E_1 \rangle} \sum_{s \in \langle S_0, S_1, S_2 \rangle} \mathcal{P}(tles), \tag{4}$$

where $t = T^{a_0} T^{a_1} T^{a_2}$ is given by the defect configuration and $\mathcal{P}(tles)$ is the probability assigned to the error $E = tles$ by the noise model. The complexity of decoding a unit cell is given by the number of operators that are considered in Eq. (4): $|\langle L_0, L_1 \rangle||\langle E_0, E_1 \rangle||\langle S_0, S_1, S_2 \rangle|$. Since all L_i, E_i, and S_i have order d, the complexity is the constant d^7. For different unit cell sizes, the complexity is still a power of d, but with a different exponent, which depends on the number of qudits in the cell and the number of measured stabilizer generators. Moreover, the number of unit cells to decode in a given round of RG is given by $(L/k)^2$, where k and L are the linear sizes of the unit cell and the global lattice, respectively. Thus, the complexity of a step of RG goes as $d^c (L/k)^2$ for some constants c and k that depend on the choice of unit cell. Of course, the RG calculations on different cells can be executed in parallel.

The procedure we have described above to evade the correlations caused by local charge conservation is only a heuristic and can be improved using belief propagation (BP). Roughly, the role of BP is to ensure consistency between the marginal probability of qubits located at the boundary of two or more unit cells, e.g., qudits 0, 1, 8, and 9 [see Fig. 3(b) for labeling]. First, given a defect configuration inside a unit cell, one can compute the marginal error probability $\mathcal{P}_q(tles|_q)$ for each qudit q, obtained by taking a marginal of $\mathcal{P}(tles)$. These are called messages and denoted $m_q^{\text{out}}(p)$, where q labels a qudit and p is a one-qudit Pauli operator. These outgoing messages are then exchanged between neighboring cells and become incoming messages, e.g., a cell c sends to its northern neighbor c' the message m_0^{out}, which becomes m_9^{in} in c' and receives from c' the message m_9^{out}, which becomes m_0^{in} in c. Subsequent rounds of messages can be calculated using the

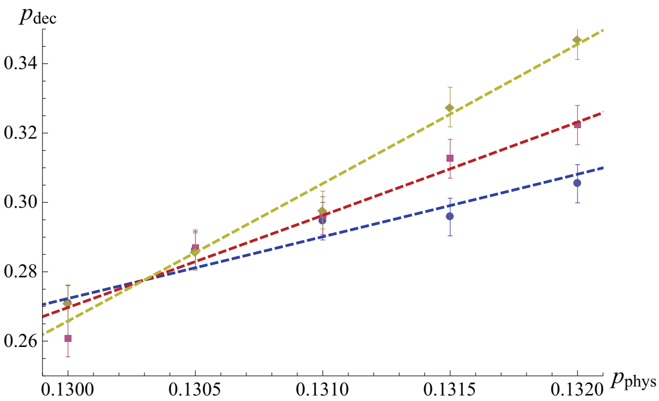

FIG. 4. (Color online) Threshold estimation for $\mathbb{Z}_3$-KTC. The x axis represents the physical error rate and the y axis, the decoding error rate. The (blue) circles, (red) squares, and (yellow) diamonds correspond to $L = 32$, $L = 64$, and $L = 128$ respectively. The fitting curve used is $p_{\text{dec}} = (p_{\text{phys}} - p_{\text{th}})L^{1/\nu}$. In this case, we find $p_{\text{th}} = 0.13(0)$.

received messages, following the prescription

$$m_q^{\text{out}}(p) \leftarrow \sum_{l,s,e} \delta(tles|_q,p) \frac{\mathcal{P}(tles)}{\mathcal{P}_q(tles|_q)} \prod_{q' \neq q} m_{q'}^{\text{in}}(tles|_{q'}). \qquad (5)$$

Here, $q,q' \in \{0,1,8,9\}$, $tles|_q$ is the restriction to qudit q of the Pauli operator $tles$ and $\mathcal{P}_q$ is the marginal on qudit q of the noise model as above. BP can be iterated a few times (e.g., three rounds) before executing an RG step. This has the effect of replacing Eq. (4) with

$$\mathcal{P}(l) = \sum_{e \in \langle E_0,E_1 \rangle} \sum_{s \in \langle S_0 S_1 S_2 \rangle} \mathcal{P}(tles) \prod_q m_q^{\text{in}}(tles|_q). \qquad (6)$$

V. NUMERICAL RESULTS

In this section, we present our numerical estimates of the thresholds of $\mathbb{Z}_d$-KTCs for $2 \leqslant d \leqslant 6$ subject to the generalized bit-flip noise model introduced in the previous section. The threshold is defined as the value of the physical noise rate p_{phys}, below which the decoding error probability p_{dec} can be made arbitrarily small by increasing the lattice size L.

The simulations were performed as follows. For various values of d, L, and p_{phys}, specifiying a $\mathbb{Z}_d$-KTC of linear size L subject to a noise of parameter p_{phys}, we performed a Monte Carlo simulation to estimate the decoding error probability p_{dec}. We used sample sizes of the order of 10^4. For a fixed value of d, we plotted estimates of p_{dec} vs p_{phys} for different values of L. We then used the fitting model $p_{\text{dec}} = (p_{\text{phys}} - p_{\text{th}})L^{1/\nu}$ (see [9] and [16] for more details) to estimate the value of the threshold. As an example, we plot the results and the fits for $\mathbb{Z}_3$-KTC in Fig. 4.

Repeating this for $3 \leqslant d \leqslant 6$ (2 was studied in [11] and [12]), Fig. 5 shows p_{th} as a function of d. Heuristically, we did expect that the value of p_{th} increases with d. Indeed, if we imagine simulating a qudit using $\log_2 d$ qubits, a fixed noise rate for increasing values of d translates into a decreased noise rate per qubit. Moreover, it was reported in [17] that the performance of BP for $\mathbb{Z}_d$-KTC, which is very poor in the qubit case, is greatly increased as d grows.

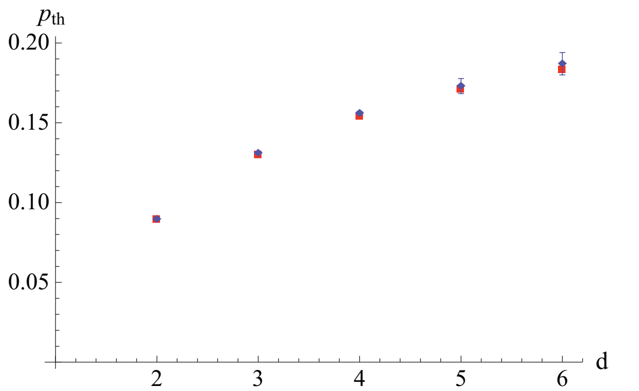

FIG. 5. (Color online) The (blue) diamonds are the values extracted by fitting the threshold values for $2 \leqslant n \leqslant 6$ (see Fig. 4, for example). The (red) squares were obtained via the generalized hashing bound (see text) rescaled by a common factor $\alpha = p_{\text{th}}(2)/C_2 \approx 0.81$. Error bars are (pessimistically) obtained, e.g., by replacing each line in Fig. 4 with a stripe of width equal to the statistical error bars and determining the values of p_{phys} above and below the crossing point where the strips cease to overlap. We do not report the fitting parameter ν because they are too sensitive to statistical fluctuations and therefore unreliable in our study.

It is intringuing to note that for $\mathbb{Z}_2$-KTC subject to bit-flip or depolarizing noise, p_{th} is numerically very close to the hashing bound [9,16,18] (and so do other topological codes [19]). The hashing bound, obtained by a simple packing argument [20], states that for nondegenerate CSS codes,

$$0 \leqslant 1 - 2H_2(p), \qquad (7)$$

where H_2 is the binary entropy: $H_2(p) = -(1-p)\log_2(1-p) - \log_2 p$. From Eq. (7), one can calculate the saturating point $C_2 \approx 0.110$, which is indeed quite close to the optimal threshold of the $\mathbb{Z}_2$-KTC subject to independent bit-flip and phase-flip errors, $p_{\text{th}}(2) \approx 0.109(4)$ [9,16]. This near-coincidence is intriguing given that topological codes are highly degenerate, so there is no reason they should obey the hashing bound. Of course, the decoder we are using here is suboptimal, so the threshold we find, $p_{\text{th}}(2) \approx 0.89(6)$, is a smaller fraction, $\alpha = p_{\text{th}}(2)/C_2 \approx 0.81(4)$, of the hashing bound.

For qudits, the hashing bound is

$$0 \leqslant 1 - 2H_d(p),$$
$$\text{with} \quad H_d(p) = -(1-p)\log_d(1-p) - p\log_d \frac{p}{d-1}. \qquad (8)$$

In this case, we find $C_3 \approx 0.159, C_4 \approx 0.189$, and so on. Figure 5 shows the threshold $p_{\text{th}}(d)$ obtained with the RG decoder as well as a rescaled hashing bound αC_d, where α is determined by the $\mathbb{Z}_2$ fit. The agreement is both unexplained and surprisingly good. Note also that even though our decoder is suboptimal, $p_{\text{th}}(d+1) > C_d$ for all d we have studied, which strongly supports the claim that the threshold increases with d.

VI. CONCLUSION

In this paper, we have presented a generalization of the RG decoder of [11] and [12] to KTCs built with the groups $\mathbb{Z}_d$. Our numerical results show that the threshold value increases as a function of the local dimension d. Moreover, its behavior is in very good agreement with a scaling predicted by the hashing bound. This trend could be confirmed by more accurate numerical estimates using a mapping to a statistical mechanics model, which does not require solving the decoding problem [9,18]. A theoretical understanding of this behavior is also desirable. Finally, estimating the threshold in the presence of measurement error and detailed syndrome measurement circuits on qudits remains an interesting open question.

ACKNOWLEDGMENTS

We would like to thank Jonas Anderson for useful discussions regarding the generalized hashing bound. We also thank Simon Burton, Courtney Brell, and Stephen Bartlett for enlightening discussions of Kitaev's construction [1]. Computational resources were provided by Calcul Québec and Compute Canada. This work was partially funded by NSERC and by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior (DoI) National Business Center Contract No. D11PC20167. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

APPENDIX: GENERALIZED BIT-FLIP CHANNEL

The noise model described in Sec. III can be seen as emerging from a qudit depolarizing channel. To show this, consider the depolarizing channel which has the same form for qudits as for qubits:

$$\begin{aligned}\epsilon_d(\rho) &= (1-p)\rho + p\frac{I}{d} \\ &= (1-p)\rho + \frac{p}{d^2}\sum_{i,j=0}^{d-1} X^i Z^j \rho Z^{-j} X^{-i}. \end{aligned} \qquad \text{(A1)}$$

From this last expression, we see that the probability that error X^i and Z^j occur simultaneously is $\mathcal{P}(X^i,Z^j) = \frac{p}{d^2}$, except when $i = j = 0$ when $\mathcal{P}(I,I) = (1 - p + \frac{p}{d})$. Because we use a CSS code, we can correct X-type and Z-type errors independently. The code correcting X-type errors sees a noise model that is the marginal of $\mathcal{P}(X^i,Z^j)$, i.e., $\mathcal{P}(X^i) = \sum_j \mathcal{P}(X^i,Z^j)$, resulting in the channel

$$\epsilon_x(\rho) = (1 - p + \tfrac{p}{d})\rho + \tfrac{p}{d}\textstyle\sum_{i,j=1}^{d-1} X^i \rho X^{-i}, \qquad \text{(A2)}$$

and similarly for Z.

The generalized bit-flip channel introduced in Sec. III has the form

$$\epsilon_{bf}(\rho) = (1-p)\rho + \tfrac{p}{d-1}\textstyle\sum_{i,j=0}^{d-1} X^i \rho X^{-i}. \qquad \text{(A3)}$$

Comparing Eq. (A2) and Eq. (A3), we see that performing the following subsitution in ϵ_{bf} yields ϵ_x:

$$p \to (1 - d^{-1})p.$$

This type of noise model is typical in information theory to assess the robustness of different codes and architectures. While the depolarizing noise model, Eq. (A1), is well motivated physically by its symmetry—i.e., it is the least biased noise model—its marginal, the generalized bit-flip channel, Eq. (A3), is somewhat more artificial. However, a coding scheme capable of correcting a generalized bit-flip channel of strength p for both X- and Z-type errors will also error correct a depolarizing channel of strength $p' = 3p/2$. Moreover, a coding scheme that exploits the correlations between X- and Z-type errors may even tolerate a depolarizing rate higher than this. For more discussions on physically justified noise model, see [5].

[1] A. Kitaev, Ann. Phys. **303**, 2 (2003).
[2] S. S. Bullock and G. K. Brennen, J. Phys. A Math. Gen. **40**, 3481 (2007).
[3] H. Bombin and M. A. Martin-Delgado, J. Math. Phys. **48**, 052105 (2007).
[4] M. D. Schulz, S. Dusuel, R. Orús, J. Vidal, and K. P. Schmidt, New J. Phys. **14**, 025005 (2012).
[5] O. Viyuela, A. Rivas, and M. A. Martin-Delgado, New J. Phys. **14**, 033044 (2012).
[6] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. A **76**, 042319 (2007).
[7] A. Nussenzweig, J. Hare, A. M. Steinberg, L. Moi, M. Gross, and S. Haroche, Europhys. Lett. **14**, 755 (1991).
[8] A. Vaziri, J.-W. Pan, T. Jennewein, G. Weihs, and A. Zeilinger, Phys. Rev. Lett. **91**, 227902 (2003).
[9] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[10] J. Edmonds, Can. J. Math. **17**, 449 (1965).
[11] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).
[12] G. Duclos-Cianci and D. Poulin, IEEE Information Theory Workshop, Dublin (2010), p. 1.
[13] E. Knill, arXiv:quant-ph/9608048.
[14] D. Gottesman, Chaos Solitons Fractals **10**, 1749 (1999).
[15] S. Bravyi and J. Haah, arXiv:1112.3252.
[16] J. W. Harrington, Ph.D. thesis, California Institute of Technology, 2004.
[17] I. Andriyanova, D. Maurice, and J.-P. Tillich, arXiv:1202.3338.
[18] H. Bombin, R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, Phys. Rev. X **2**, 021004 (2012).
[19] H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, Phys. Rev. Lett. **103**, 090501 (2009).
[20] A. Ekert and C. Macchiavello, Phys. Rev. Lett. **77**, 2585 (1996).