


Erasure decoding of two-dimensional color codes

Arun B. Alosious^{*} and Pradeep Kiran Sarvepalli[†]

Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600 036, India

 (Received 20 May 2019; published 14 October 2019)

The quantum erasure channel models phenomena such as loss or leakage of qubits. Using quantum codes, we can recover from such errors. In this paper, we are interested in studying the performance of two-dimensional color codes over the quantum erasure channel. Our approach makes use of the local equivalence between color codes and surface codes. We propose a variety of decoding algorithms for color codes over the erasure channel. First, we propose algorithms that decode by projecting the erasures on the color to surface codes. Then, instead of directly decoding on the color code or on the equivalent copies of surface codes, we decode jointly on the color code and the equivalent surface codes. We observe a threshold of 44.3% for the color code on the square octagonal lattice.

DOI: [10.1103/PhysRevA.100.042312](https://doi.org/10.1103/PhysRevA.100.042312)

I. INTRODUCTION

Loss of qubits and leakage errors occur in many realizations of qubits; see, for instance [1–4]. The loss of qubits in quantum systems can be modeled as a quantum channel known as the quantum erasure channel [5,6,8]. The erasure channel also shows up in the context of quantum cryptographic protocols such as secret sharing [7]. Quantum error correcting codes help in protecting information in the presence of qubit losses and leakage [6,8–18].

There have been many studies on the design and performance of quantum codes for erasure channels, ranging from algebraic codes such as the quantum Bose-Chaudhuri-Hocquenghem (BCH) codes [8] to quantum low density parity check codes [14]. In particular, many researchers have investigated the performance of topological codes over the erasure channel [11–13,18–22].

The performance of a code over the erasure channel can often give an indication of its performance over other channels [22]. In the context of topological codes, one can also view erasures as defects in a lattice, leading to interesting connections between topological codes and percolation theory [11]. Thus, studying the performance of quantum codes over the erasure channel is not only important in its own right, but also for the additional insights it provides.

Toric codes, also called surface codes [23], and color codes [24] are two of the most studied classes of topological codes in view of their relevance for fault tolerant quantum computing. While erasure decoding of toric codes has been studied extensively for many years, an algorithm that is optimal both in running time and performance was proposed only recently in [20]. There is very limited work on the erasure decoding of two-dimensional (2D) color codes [19]. So, in this paper we study the decoding of color codes over the erasure channel.

The erasure decoding problem is to find the error estimate that satisfies the measured syndrome and whose support lies entirely within the erased qubits. When the qubits are lost, we have additional information about the positions of the lost qubits. Hence to get the best performance, decoders over the erasure channel must exploit this additional information.

The decoding problem, as in the classical erasure case, can be reduced to the solution of a linear system of equations. The system of linear equations can be solved in time that is cubic in the length of the code. We seek a more efficient algorithm that is either linear or almost linear time in the length of the code.

One approach to decoding the color codes involves modifying the stabilizers so that they do not involve the erased qubits. This approach was taken in [19]; see below for further discussion on this approach. In this paper, we take an alternative route in which we retain all the stabilizers including those that involve the erased qubits. At the heart of our approach is the fact that the color codes are equivalent to copies of toric codes [25–29]. To perform the decoding of color codes through the toric code decoder, we need to map the erasure positions appropriately. We propose two different maps for erasures on the color code to the toric code, which are consistent with the map of operators given in [27]. We can then decode the erasures on the toric codes and then lift them back to the color code. Such a decoder can be significantly improved. So instead of decoding the erasures solely on the toric codes, we decode the erasures jointly on the color code and the toric codes. This modification led to a performance of $\gtrsim 44.3\%$ for the color code on the square octagonal lattice. This is closer to the best possible threshold of 50%, which is restricted by the no-cloning theorem.

In a recent paper, Vodola *et al.* [19] also studied the problem of qubit losses in color codes. They approached this problem from a different point of view from the one taken in this paper. When a qubit is erased, the stabilizers that involve that qubit are modified so that the erased qubit does not participate in the modified stabilizers. Hence the erased qubits are completely removed, and new stabilizers are used

^{*}aloshious.sp@gmail.com

[†]pradeep@ee.iitm.ac.in

to encode the logical information that is not erased. This modification is such that it preserves the information encoded as long as the support of a logical operator is not erased. This is in effect equivalent to a code of shorter length with a lower number of stabilizer generators. This is in contrast to our approach, in which every erased qubit is replaced by a qubit in a completely mixed state, and all the stabilizer generators are retained.

Secondly, the approach taken in [19] decodes on a modified lattice for the color code. In this paper, we present multiple approaches: we propose an algorithm that decodes the color code by mapping it to a pair of surface codes, and another that decodes jointly on both the color code and the surface codes.

Another difference between our work and [19] lies in the support of stabilizer generators. In [19], the new set of stabilizer generators, which do not contain the erased qubits, are typically of higher weight than the original stabilizer generators. Further, they may not always be local. Since we keep all the stabilizer generators unmodified, the code continues to be a local stabilizer code. In the noiseless setting, their approach leads to a threshold of 46.1% for the color code, while we were able to achieve a threshold of $\gtrsim 44.3\%$ (for the color code on the square octagonal lattice).

We organized the paper as follows: In the next section, we review the background for our proposed decoders. We review the mapping between color codes and surface codes [27]. Before we can apply the map for erasure decoding, our approach needs to extend the map in [27] to the mapping of syndromes on color codes to surface codes. We explore this in Sec. III. Then in Sec. IV, we map the erasures on the color codes to erasures on the surface codes. We then use this map to decode color codes over the erasure channel. Finally, we conclude with a brief summary in Sec. V.

II. BACKGROUND

The two-dimensional toric code (or a surface) is defined on an arbitrary 2D lattice embedded on a torus in which the qubits are on edges. We can also define the code on a closed or open surface instead of a torus. We use the terms toric code and surface code interchangeably. Stabilizer generators are defined on vertices and faces as follows:

$$A_v = \prod_{e \in \partial(v)} X_e \quad \text{and} \quad B_f = \prod_{e \in \partial(f)} Z_e. \quad (1)$$

Color codes are defined on a 2-face colorable trivalent lattice called a 2-colex. Qubits are placed on the vertices of this 2-colex. The stabilizer generators for color codes are defined on its faces and given by

$$B_f^X = \prod_{v \in f} X_v \quad \text{and} \quad B_f^Z = \prod_{v \in f} Z_v. \quad (2)$$

We have modeled the qubit loss as the quantum erasure channel [5],

$$\rho \rightarrow (1 - \gamma)\rho + \gamma \frac{I}{2}. \quad (3)$$

A qubit in the completely mixed state is placed in the position of an erased qubit. Now if we make a stabilizer measurement, then it projects the erasure error onto one of the single-qubit

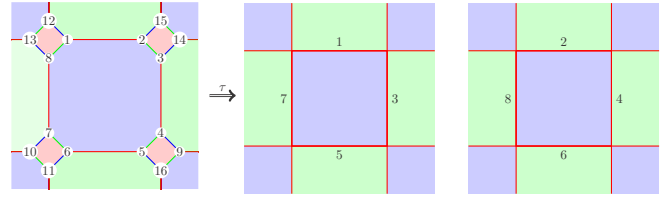


FIG. 1. Color code and its equivalent toric codes. Qubits are numbered accordingly.

Pauli errors with equal probability. This is because of the following relation:

$$\frac{I}{2} = \frac{1}{4}\rho + \frac{1}{4}X\rho X + \frac{1}{4}Y\rho Y + \frac{1}{4}Z\rho Z \quad (4)$$

for arbitrary state ρ . We do this for all erased qubits. This approach is used in some of the decoders for quantum codes over the erasure channel [16,18,20,22].

Now we summarize the map between the color code and two copies of the toric code given in [27]. We assume that the color code is defined on a 2-colex Γ . The vertices, edges, and faces of Γ are denoted $V(\Gamma)$, $E(\Gamma)$, and $F(\Gamma)$, respectively. We use the notation i or v_i to represent a qubit placed on the vertex numbered i ; see Fig. 1.

Pauli operators along the edges of Γ are called hopping operators. One basis for the Pauli operators on the color code is the set $\{X_u, Z_u \mid u \in V(\Gamma)\}$. Another generating set that is useful is the set of weight-2 operators acting along the c and c' colored edges of Γ and two single-qubit operators X_u, Z_v for two (distinct) vertices of each c'' colored face. Specifically, $X_u X_v$ and $Z_u Z_v$ for each edge $(u, v) \in E_c(\Gamma) \cup E_{c'}(\Gamma)$ and X_{2l_f}, Z_{2m_f} for each c'' colored face f , where f contains $2l_f$ vertices numbered 1 to $2l_f$ and $2m_f \leq 2l_f$. These weight-2 operators are called hopping operators and some of them are dependent. In fact, exactly one X type and one Z type hopping operator for each c'' face are dependent. If the vertices of a c'' colored face f are numbered from 1 to $2l_f$, the hopping operators $Z_1 Z_{2l_f}$ and $X_{2m_f} X_{2m_f+1}$ are dependent. For more details, see [27].

Now we define a map π on the Pauli operators on the color code. This map is defined in a recursive form for each c'' colored face as follows. First the initialization is done as

$$\pi(Z_1) = X_2 \prod_{i=1}^{m_f} Z_{2i-1}, \quad (5)$$

$$\pi(X_1) = X_1, \quad (6)$$

$$\pi(X_{2l_f}) = X_{2l_f-1}, \quad (7)$$

where $1 \leq m_f \leq l_f$. For each c'' colored face f , the parameter m_f can be chosen to be any integer satisfying the above constraint. Furthermore, the action of π on other qubits is given by

$$\pi(Z_{2j}) = \pi(Z_{2j-1})Z_{2j-1}, \quad (8)$$

$$\pi(Z_{2j-1}) = \pi(Z_{2j-2})X_{2j-2}X_{2j}. \quad (9)$$

For Z operators the recursion is twofold. First for $1 \leq j \leq m_f$,

$$\pi(X_{2j}) = \pi(X_{2j-1})Z_{2j}, \quad (10)$$

$$\pi(X_{2j-1}) = \pi(X_{2j-2})X_{2j-3}X_{2j-1}. \quad (11)$$

For $m_f + 1 \leq j \leq l_f$,

$$\pi(X_{2j-1}) = \pi(X_{2j})Z_{2j}, \quad (12)$$

$$\pi(X_{2j}) = \pi(X_{2j+1})X_{2j-1}X_{2j+1}. \quad (13)$$

If we map the Pauli operators by Eqs. (5)–(13) on all c'' colored faces of the color code, then the color code will be mapped to exactly two copies of the toric code. One toric code is defined by the odd-numbered qubits and the other is defined by the even-numbered qubits. The toric code lattice Γ_c is obtained by shrinking the c colored faces into vertices. Hence the toric codes are constructed on the lattice Γ_c , which contains c colored faces of Γ as vertices, c colored edges of Γ as edges, and c' and c'' colored faces of Γ as faces. The qubits on odd-numbered vertices are moved to the c colored edges incident on these vertices of the first toric code, while the qubits on even-numbered vertices are moved to the c colored edges of the second toric code. The details can be seen in [27]. The action of obtaining the toric code lattice from the color code lattice is performed by a map τ . The map τ acts on the c colored faces of 2-colex Γ and converts them into vertices in the toric code lattice Γ_c . Similarly, it converts the c' and c'' colored faces of 2-colex to faces of the toric code lattice. The vertices of the color code are converted using the action of τ as the edges on Γ_c . An example of the map τ on qudits of the color code to the toric codes is as shown in the Fig. 1. We use $[A]_i$ to denote the operator A acting on the i th copy of the toric code.

We can easily obtain the inverse map π^{-1} . We have the following:

$$\pi^{-1}([Z_{2i-1}]_1) = Z_{2i-1}Z_{2i}, \quad (14)$$

$$\pi^{-1}([Z_{2i}]_2) = X_{2i-1}X_{2i}. \quad (15)$$

For $1 \leq i \leq m_f$, we have

$$\pi^{-1}([X_{2i-1}]_1) = X_1X_2, \dots, X_{2i-1}, \quad (16)$$

$$\pi^{-1}([X_{2i}]_2) = Z_{2i}Z_{2i+1}, \dots, Z_{2m_f}. \quad (17)$$

For $m_f + 1 \leq i \leq l_f$, we compute the inverse mapping as

$$\pi^{-1}([X_{2i-1}]_1) = X_{2i}X_{2i+1}, \dots, X_{2l_f}, \quad (18)$$

$$\pi^{-1}([X_{2i}]_2) = Z_{2m_f+1}Z_{2m_f+2}, \dots, Z_{2i-1}. \quad (19)$$

III. MAPPING SYNDROME FROM COLOR CODE TO TORIC CODES

To decode a color code, the information we have is the positions of erasures and the syndrome measurement. So, to design a decoder for a color code by mapping it to the toric codes, we need two additional mappings: one for mapping syndrome information from the color code to the toric codes, and another to map the information about erasure positions. These two mappings should be consistent with the map of

operators π between the color code and the toric codes. In this section, we map syndromes from color code to toric codes.

The syndrome for a stabilizer code is obtained by measuring the stabilizer generators. A color code is a Calderbank-Shor-Steane (CSS) code [30] and every face can be associated with an X -type and a Z -type stabilizer. In projecting the syndrome on the color code to the surface codes, we require the following to be satisfied:

(i) Consistency: Suppose E is an error on the color code and s_E is its syndrome. We require the syndrome mapped on the surface code to be the syndrome of $\pi(E)$, i.e., it should be the syndrome of the image of the error on the surface codes.

(ii) Locally computable: Given s_E , the syndrome of an error E on the color code, we want to be able to compute the syndrome on the surface codes only from s_E and not from E . Further, we want to be able to compute the syndrome in a local fashion. In other words, the syndrome computation on the surface code must depend only on the syndromes in the “neighborhood of the syndrome.” This condition is imposed to ensure that the complexity of syndrome computation is linear.

Consider a stabilizer generator B_f^σ for $\sigma \in \{X, Z\}$ on the color code. Let $\pi(B_f^\sigma) = \prod_{i=1}^{n_1} A_{v_i} \prod_{j=1}^{n_2} B_{f_j}$, where A_{v_i} and B_{f_j} are vertex and face stabilizer generators on the surface codes. Suppose an error E produces the syndrome s with respect to B_f^σ . Then we want $\pi(E)$ to produce the same syndrome with respect to $\pi(B_f^\sigma)$. On the surface codes, the syndrome is determined by whether A_{v_i} and B_{f_j} commute with $\pi(E)$ or not.

Recall that the syndrome associated with a stabilizer generator S with respect to an error is given by $SE = (-1)^s ES$, where $s = 0$ if E commutes with S and $s = 1$ otherwise. Let $B_f^\sigma E = (-1)^s EB_f^\sigma$, $A_{v_i} E = (-1)^{s_i} EA_{v_i}$, and $B_{f_j} E = (-1)^{s'_j} EB_{f_j}$. Therefore, we must have

$$(-1)^s = \prod_i (-1)^{s_i} \prod_j (-1)^{s'_j}, \quad (20)$$

$$s = s_1 + \dots + s_{n_1} + s'_1 + \dots + s'_{n_2}. \quad (21)$$

The sum of these syndromes s_i and s'_j must be the same as the syndrome for E with respect to B_f^σ . Here we know s , while s_i and s'_j are unknown. From the preceding discussion, we see that the syndromes on the surface codes can be found by solving a system of linear equations. This can lead to a superlinear overhead in general. Fortunately, we can solve this in linear time since some of the stabilizers B_f^σ are mapped to exactly one stabilizer generator on the surface code. In this case, the right-hand side of Eq. (21) contains only one variable, in other words the syndrome on the color code is projected directly onto the surface code. This happens whenever $f \in F_{c'}(\Gamma) \cup F_{c''}(\Gamma)$, as will be shown in Theorem 1. If $f \in F_c(\Gamma)$, then, as will be shown in Theorem 2, the right-hand side of Eq. (21) contains only one unknown variable, and all other variables will be known from faces in $F_{c'}(\Gamma) \cup F_{c''}(\Gamma)$. Thus we are able to map the syndromes from the color code to the surface codes. The locality of this map follows from the fact that π is local so each B_f^σ is mapped onto a collection of local stabilizers in each copy of the surface code. We shall now prove this formally in the rest of the section. We shall use the notation s_f^σ to denote the syndrome of B_f^σ . First, we need to

find the images of the stabilizers on the color code for which we define a dependent edge set.

Definition 1. Dependent edge set. Let D_X be the set of c' colored edges on Γ that are defined as the dependent edges for X hopping operators under π . Likewise, D_Z represents the c' colored edges that are dependent for Z hopping operators.

Let $f_{c'}^{\otimes e}$ be the unique c' -face that contains the c' -edge e in its boundary.

Lemma 1. Images of stabilizers. Let $f \in F_{c'}(\Gamma) \cup F_{c''}(\Gamma)$. Then

$$\pi(B_f^Z) = [B_{\tau(f)}]_1, \quad (22)$$

$$\pi(B_f^X) = [B_{\tau(f)}]_2. \quad (23)$$

If $f \in F_c(\Gamma)$, then

$$\pi(B_f^X) = [A_{\tau(f)}]_1 \prod_{e \in \partial(f) \cap D_X} [B_{\tau(f_{c'}^{\otimes e})}]_2, \quad (24)$$

$$\pi(B_f^Z) = [A_{\tau(f)}]_2 \prod_{e \in \partial(f) \cap D_Z} [B_{\tau(f_{c'}^{\otimes e})}]_1, \quad (25)$$

where D_X and D_Z are the set of dependent c' -edges for magnetic and electric hopping operators, respectively.

Proof. Let $f \in F_{c'}(\Gamma) \cup F_{c''}(\Gamma)$. Let us number the vertices of f as $1, 2, \dots, 2\ell_f$ such that $(1,2)$ is a c colored edge. Let $\tau(v_{2i}) = \tau(v_{2i-1}) = e_i$, where e_i is an edge in Γ_c . Then

$$\pi(B_f^X) = \pi\left(\prod_{i=1}^{\ell_f} X_{v_{2i-1}} X_{v_{2i}}\right) = \prod_{i=1}^{\ell_f} \pi(X_{v_{2i-1}} X_{v_{2i}}) \quad (26)$$

$$\stackrel{(a)}{=} \prod_{i=1}^{\ell_f} [Z_{\tau(v_{2i})}]_2 \stackrel{(b)}{=} \prod_{i=1}^{\ell_f} [Z_{e_i}]_2 \stackrel{(d)}{=} [B_{\tau(f)}]_2, \quad (27)$$

where (a) comes from the definition of π , (b) comes from the definition of τ , and (d) comes from the structure of the toric code. Arguing similarly, we have $\pi(B_f^Z) = [B_{\tau(f)}]_1$.

Now let $f \in F_c(\Gamma)$. Number the vertices as $1, 2, \dots, 2\ell_f$ such that $(1,2)$ is a c' colored edge. Then,

$$\pi(B_f^X) = \pi\left(\prod_{i=1}^{\ell_f} X_{v_{2i-1}} X_{v_{2i}}\right) = \prod_{i=1}^{\ell_f} \pi(X_{v_{2i-1}} X_{v_{2i}}).$$

If $e = (v_{2i-1}, v_{2i})$ is an independent c' -edge, then $\pi(X_{v_{2i-1}} X_{v_{2i}}) = [X_{\tau(v_{2i-1})} X_{\tau(v_{2i})}]_1$, otherwise $\pi(X_{v_{2i-1}} X_{v_{2i}}) = [X_{\tau(v_{2i-1})} X_{\tau(v_{2i})}]_1 [B_{\tau(f_{c'}^{\otimes e})}]_2$. In this case, the dependent edge e is in D_X , and we can write

$$\begin{aligned} \pi(B_f^X) &= \prod_{i=1}^{\ell_f} [X_{\tau(v_{2i-1})} X_{\tau(v_{2i})}]_1 \prod_{e \in \partial(f) \cap D_X} [B_{\tau(f_{c'}^{\otimes e})}]_2 \\ &= \prod_{i=1}^{\ell_f} [X_{\tau(v_{2i-1})} X_{\tau(v_{2i})}]_1 \prod_{e \in \partial(f) \cap D_X} \pi(B_{f_{c'}^{\otimes e}}^X). \end{aligned}$$

Now note that $\{\tau(v_i) | 1 \leq i \leq 2\ell_f\}$ is the set of edges incident on the vertex $\tau(f)$ in Γ_c . This implies that

$$\pi(B_f^X) = [A_{\pi(f)}]_1 \prod_{e \in \partial(f) \cap D_X} [B_{\tau(f_{c'}^{\otimes e})}]_2.$$

We can prove Eq. (25) in a similar fashion. We omit the proof for brevity.

Now we show how to compute the syndromes on the surface codes given the syndromes on the color codes.

Theorem 1. Face syndromes on surface codes. Let $f \in F_{c'}(\Gamma) \cup F_{c''}(\Gamma)$ and $s_{f,i}$ be the syndrome associated with $[B_{\tau(f)}]_i$ of the i th surface code. Then

$$s_{f,1} = s_f^Z \quad \text{and} \quad s_{f,2} = s_f^X, \quad (28)$$

where s_f^σ is the syndrome for B_f^σ on the color code.

Proof. From Lemma 1 we have $\pi(B_f^X) = [B_{\tau(f)}]_2$. So the syndrome of an error E with respect to B_f^X must be exactly the same as the syndrome of $[B_{\tau(f)}]_2$ with respect to $\pi(E)$.

Similarly, $\pi(B_f^Z) = [B_{\tau(f)}]_1$. Therefore, the syndrome of E with respect to B_f^Z is identical to the syndrome of $[B_{\tau(f)}]_1$ with respect to $\pi(E)$.

Theorem 2. Vertex syndromes on surface codes. Let $f \in F_c(\Gamma)$ and $s_{f,i}$ be the syndrome associated with $[A_{\tau(f)}]_i$ of the i th surface code. Then

$$s_{f,1} = s_f^X \bigoplus_{e \in \partial(f) \cap D_X} s_{f_{c'}^{\otimes e}}^X, \quad (29)$$

$$s_{f,2} = s_f^Z \bigoplus_{e \in \partial(f) \cap D_Z} s_{f_{c'}^{\otimes e}}^Z, \quad (30)$$

where s_f^σ is the syndrome for B_f^σ .

Proof. From Eq. (25) we obtain

$$\pi\left(B_f^X \prod_{e \in \partial(f) \cap D_X} B_{f_{c'}^{\otimes e}}^X\right) = [A_{\pi(f)}]_1.$$

From this we immediately obtain the syndrome $s_{f,1}$ as stated. The proof for $s_{f,2}$ is similar, and we skip the details.

We make two important remarks about the map on the syndromes. First, note that the syndrome computations are efficient as they only depend on the syndromes of faces adjacent to a given face on the color code. Second, π preserves the CSS structure of the color code. A bit flip error causes nonzero syndromes on the vertices of the first copy and faces of the second copy of the surface code and vice versa for a phase flip error.

We illustrate the syndrome computation with the following example. Figure 2 is an example of a syndrome map for phase flip errors. It shows a portion of the square octagonal lattice defining the color code and the associated syndromes, and how they are mapped onto the surface codes. We show a similar computation for the bit flip errors in Fig. 3. This example also illustrates how π preserves the CSS structure of the color code. Syndromes due to Z errors are mapped to syndromes due to Z -type errors only on the first surface code and X errors only on the second surface code. Before we present the simulation results, we study the error model on the surface codes.

IV. PERFORMANCE OVER THE QUANTUM ERASURE CHANNEL

In this section, we study the performance of color codes over the quantum erasure channel. As in the case of the bit flip channel, we shall apply the equivalence between the color

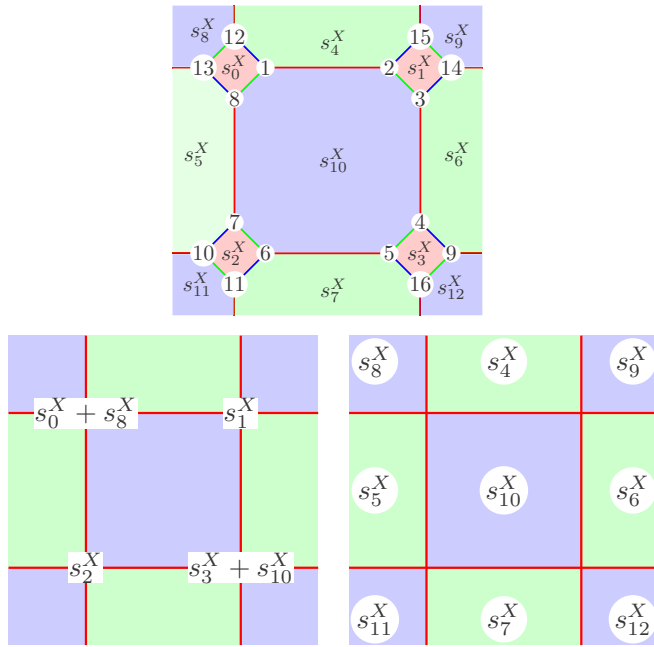


FIG. 2. Example of a syndrome map for Z errors; we assume that X_4X_5 and $X_{12}Z_{13}$ are dependent hopping operators.

code and the copies of surface codes for decoding the color code over the erasure channel.

We model the erasure channel as follows (see also [20]): We replace each of the erased qubits in the color code by a qubit in the completely mixed state $I/2$. Then we perform the syndrome measurement. Since $I/2 = \frac{\rho+X\rho X+Z\rho Z+Y\rho Y}{4}$, for

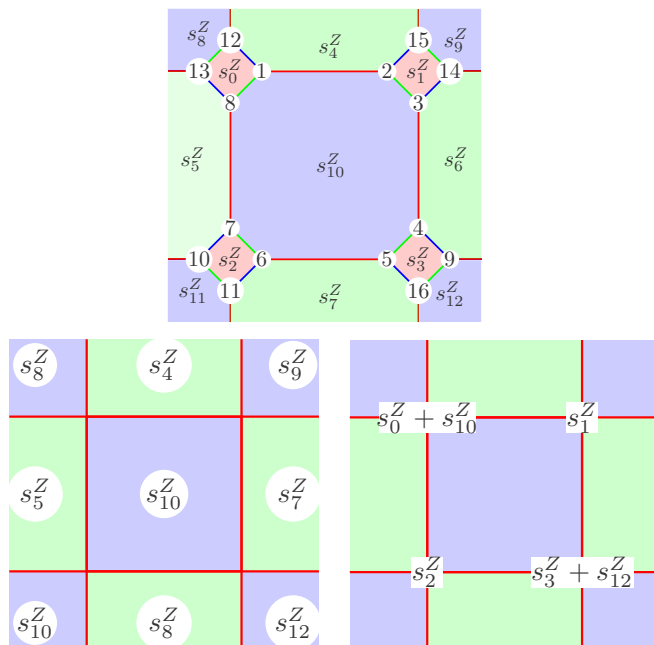


FIG. 3. Example of a syndrome map for X errors; we assume that Z_1Z_8 and Z_9Z_{16} are dependent hopping operators.

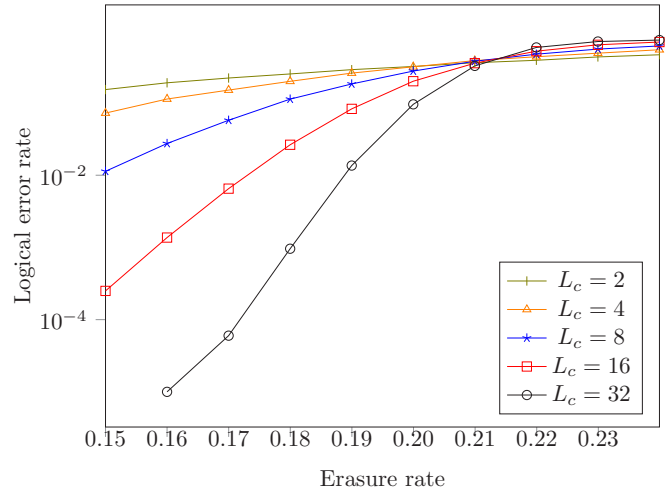


FIG. 4. Threshold for the quantum erasure channel of a color code using the naive erasure map. The code parameters are given by $[[16L_c^2, 4, 4L_c]]$.

any single-qubit density matrix ρ we can project an erasure error onto one of the Pauli errors I, X, Y , or Z with equal probability. We can measure the syndrome on the color code and map it onto the surface codes using Theorems 1 and 2. Finally, we estimate the error using the projected syndrome on the surface codes and lift the error back to the color code.

A subtle point must be borne in mind when dealing with the erasures. While the locations of erasures on the color code can be explicitly specified, the locations of the erasures on the copies of the induced surface code are not obvious. An erasure on the color code can affect multiple qubits of the surface codes.

A. Naive erasure map

A naive method to map the erasure locations onto the surface codes is as follows. Suppose the i th qubit is erased. Then, on the surface codes erase those qubits that are in support of $\pi(X_i)$, $\pi(Z_i)$, or $\pi(Y_i)$. However, observe that $\text{supp}(\pi(Y_i)) = \text{supp}(\pi(X_i)) \cup \text{supp}(\pi(Z_i))$, and hence for any Pauli error $P_{\mathcal{E}}$ acting nontrivially on the set of qubits \mathcal{E} , $\text{supp}(\pi(P_{\mathcal{E}})) \subseteq \cup_{i \in \mathcal{E}} \text{supp}(\pi(Y_i))$. Suppose \mathcal{E} is the erasure pattern and $\pi(\mathcal{E})$ is the set of qubits on which the erasure \mathcal{E} is mapped. Then we can extend the map π from color codes to surface codes to erasures as follows:

$$\pi(\mathcal{E}) = \cup_{i \in \mathcal{E}} \text{supp}(\pi(Y_i)). \tag{31}$$

While this map is conceptually simple, it places more erasures on the surface codes than required. Certain combinations of erasures lead to a smaller set of erasures on the surface codes than this map requires. A decoder based on such an erasure mapping does not perform well. We observed a threshold of about $\approx 21\%$ with such an approach (Fig. 4).

Algorithm 1: Erasure decoding of 2D color codes using a naive erasure map.

Input: A 2-colex Γ and an erasure pattern \mathcal{E} .

Output: Error estimate \hat{E}

- 1: Replace each erased qubit by a qubit in the completely mixed state.
- 2: Perform syndrome measurement on the color code.
- 3: Map the syndromes obtained on the color code using Eqs. (28)–(30).
- 4: //Map erasures onto the surface codes as follows.
- 5: **for** all erasure v_i **do**
- 6: Put erasures on $\text{supp}(\pi(Y_{v_i}))$.
- 7: **end for**
- 8: Decode both the surface codes and obtain the error estimate on surface codes \hat{E}_s .
- 9: Lift \hat{E}_s to color code using Eqs. (14)–(19).

$$\hat{E} = \pi^{-1}(\hat{E}_s).$$

B. An improved erasure map

Suppose a qubit is erased. Then we replace this qubit with a completely mixed state and measure stabilizer generators. As we discussed earlier, this measurement induces one of the Pauli errors on the qubit with equal probabilities. Our objective is to estimate this error from the measured syndrome and the position of erasures. Since we are dealing with a CSS code and the two copies of surface codes, both bit flip errors and phase flip errors lead to two instances of decoding each. In the naive map, we used the same erasure positions in each of these instances. See Fig. 5 for an illustration of this when one qubit is erased.

Suppose that an X error was induced by the stabilizer measurement. This error will then cause a nonzero syndrome on only two instances of decoding. Therefore, it would be unnecessary to place erasures on the remaining instances. Similarly, if a Z error is induced, then it will be decoded using the remaining two instances and there is no need to place the erasures on those instances that are used for bit flip decoding.

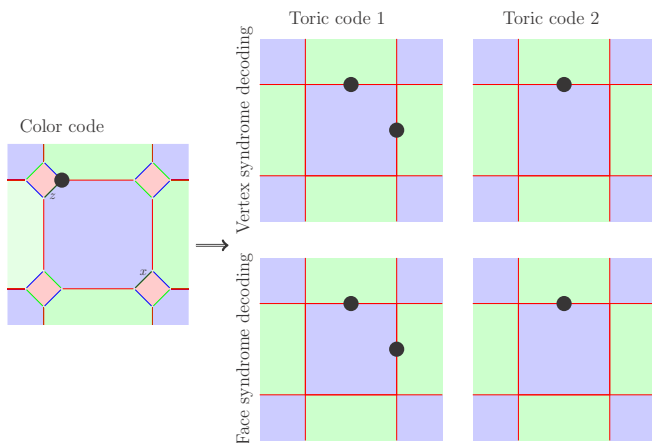


FIG. 5. Map of a single erasure on a color code based on the naive method given in Eq. (31).

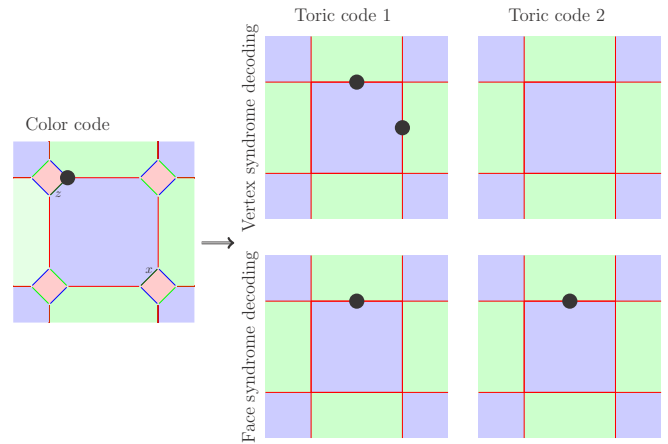


FIG. 6. Map of a single erasure on a color code based on the improved method given in Eqs. (32) and (33).

In other words, we map the erasures to each of the instances consistent with the image of error on that decoding instance (see Fig. 6). For the decoding instances corresponding to bit flip errors, we have

$$\pi(\mathcal{E}) = \cup_{i \in \mathcal{E}} \text{supp}(\pi(X_i)). \quad (32)$$

For the decoding instances corresponding to the phase flip errors, we have

$$\pi(\mathcal{E}) = \cup_{i \in \mathcal{E}} \text{supp}(\pi(Z_i)). \quad (33)$$

This method is described in Algorithm 2, and for the square octagonal lattice we obtained a threshold of $\approx 30.8\%$. Simulation results are shown in Fig. 7.

C. Jointly decoding on the color code and surface codes

Consider the face f , which contains only one erasure on vertex v of the color code. If the syndrome s_f^X on this face is nonzero, then we can be sure that the qubit placed on v has undergone a Z-type error. This is because in the erasure.

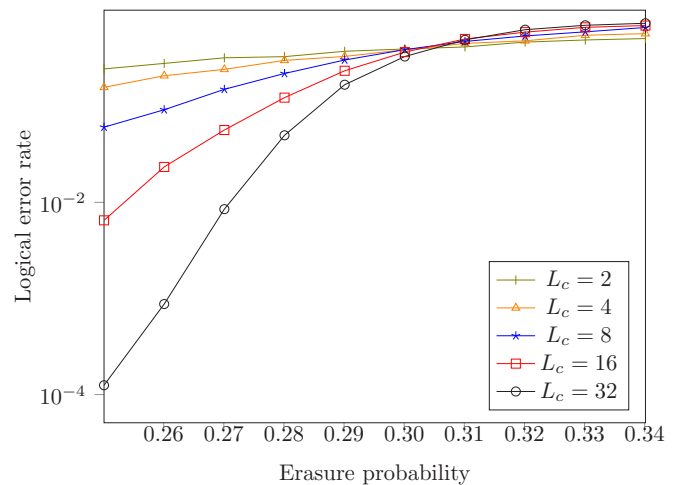


FIG. 7. Threshold for the quantum erasure channel of a color code using the erasure map in Eqs. (32) and (33). The code parameters are given by $[[16L_c^2, 4, 4L_c]]$.

Algorithm 2: Erasure decoding of 2D color codes.

Input: A 2-colex Γ and an erasure pattern \mathcal{E} .
Output: Error estimate \hat{E}

- 1: Replace each erased qubit by a qubit in the completely mixed state.
- 2: Perform syndrome measurement on the color code.
- 3: Map the syndromes obtained on the color code using Eqs. (28)–(30).
- 4: //Map erasures onto the surface codes as follows.
- 5: For vertex syndromes: If there an erasure on qubit v_{2i-1} or v_{2i} place an erasure on qubit $\tau(v_{2i-1})$ on both copies of the surface codes.
- 6: For face syndromes:
- 7: //Map erasures onto the first copy of surface code
- 8: **if** $1 \leq i \leq 2m_f$ **them**
- 9: **if** there is an erasure among $v_{2i}, v_{2i+1}, \dots, v_{2m_f}$, **then** place an erasure on $\tau(v_{2i-1})$
- 10: **end if**
- 11: **else**
- 12: **if** there is an erasure among $v_{2m_f+1}, v_{2m_f+2}, \dots, v_{2i-1}$, **then** place an erasure on $\tau(v_{2i-1})$
- 13: **end if**
- 14: **end if**
- 15: //Map erasures onto the second copy of surface code
- 16: **if** $1 \leq i \leq 2m_f$ **them**
- 17: **if** there is an erasure among $v_{1i}, v_{2i}, \dots, v_{2i-1}$, **then** place an erasure on $\tau(v_{2i-1})$
- 18: **end if**
- 19: **else**
- 20: **if** there is an erasure among $v_{2i}, v_{2i+1}, \dots, v_{2l_f}$, **then** place an erasure on $\tau(v_{2i-1})$
- 21: **end if**
- 22: **end if**
- 23: Decode both the surface codes and obtain the error estimate on surface codes \hat{E}_s .
- 24: Lift \hat{E}_s to color code using Eqs. (14)–(19).

$$\hat{E} = \pi^{-1}(\hat{E}_s).$$

channel model, there will not be any error in the unerased qubits. Using the same procedure for s_f^Z , we can estimate the X -type error undergone by this qubit v . With this, we can confirm the presence or absence of the X and Z error on this qubit. A Y -type error is equivalent to the product of X - and Z -type errors, hence detecting both X and Z on this vertex implies the presence of a Y -type error on this qubit. With this procedure, we can identify the error estimate on qubit v . After identifying the appropriate error estimate at the vertex v , we can remove this vertex from the set of erasures. Now again we can check for faces with only one erased vertex, and we can use the same procedure. We can repeat this until no faces with only one erasure exist

The process we described above is called peeling, and it has been used in the context of decoding erasures [18]. We do peeling as long as possible, in other words as long as there is a check in which there is exactly one erased qubit. At some point we will encounter a situation in which all the faces (checks) are supported on at least two vertices from the erased qubits. In such a situation, we would be unable to do

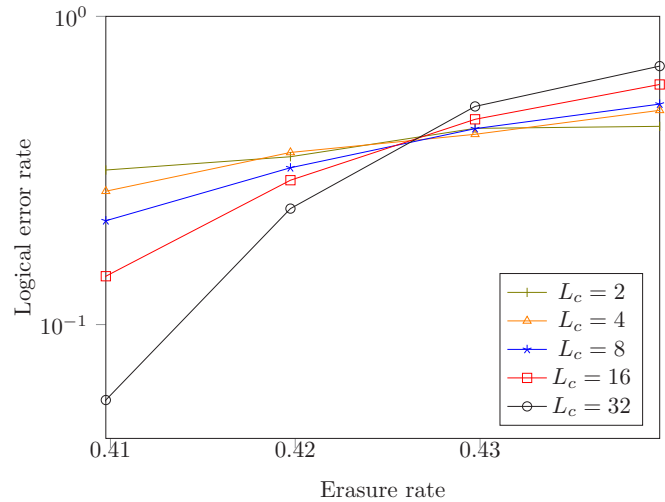


FIG. 8. Threshold for the quantum erasure channel of a color code using the naive erasure map with joint decoding. The code parameters are given by $[[16L_c^2, 4, 4L_c]]$.

peeling, thus we would have to apply our earlier decoders. We summarize this procedure in the Algorithm 3.

We have simulated this joint decoder with both Algorithms 1 and 2. We obtained improved thresholds of $\approx 42.6\%$ and $\approx 44.3\%$, respectively (Figs. 8 and 9), for the color code defined on the square octagonal lattice. We also note that the peeling decoder by itself gives a poor performance on the color code and is worse than either of the algorithms.

We obtained the thresholds based on the logical error rate. The color code (in the simulations) encodes four logical qubits. We count the number of logical errors made by the decoder in each run. Each data point in the threshold curves is obtained by running the decoder until we accumulate 2000 logical errors or 10 000 runs, whichever occurs first. We also simulated the performance using a block error rate. In this case, one or more logical errors in a single run are counted as

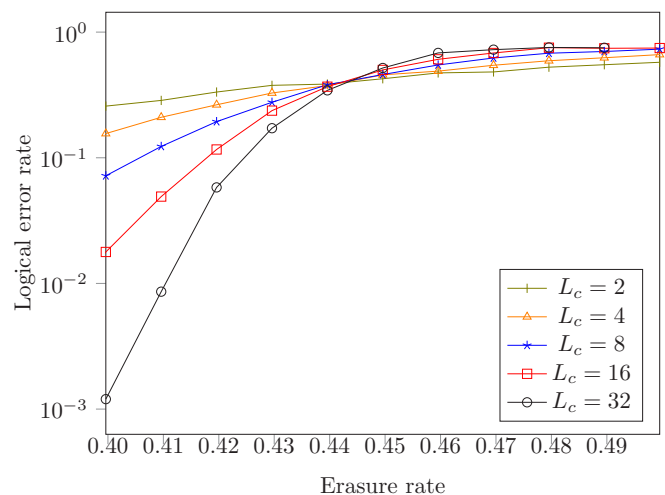


FIG. 9. Threshold for the quantum erasure channel of a color code using the erasure map in Eqs. (32) and (33) with joint decoding. The code parameters are given by $[[16L_c^2, 4, 4L_c]]$.

Algorithm 3: Jointly decoding on the color code and surface codes.

Input: A 2-colex Γ and an erasure pattern \mathcal{E} .

Output: Error estimate \hat{E}

- 1: Initialize $\hat{E} = I$
 - 2: Replace each erased qubit by a qubit in the completely mixed state.
 - 3: Perform syndrome measurement on the color code.
 - 4: **while** There exists a face f that carries only one erasure v **do**
 - 5: **if** X -type syndrome on face f is nonzero **then**
 - 6: Update the error estimate $\hat{E} = \hat{E}Z_v$
 - 7: **end if**
 - 8: **if** Z -type syndrome on face f is nonzero **then**
 - 9: Update the error estimate $\hat{E} = \hat{E}X_v$
 - 10: **end if**
 - 11: Remove v from erasure set.
 - 12: **end while**
 - 13: Use Alg. 1 or Alg. 2 to estimate the remaining error \hat{E}_R
 - 14: Update the error estimate $\hat{E} = \hat{E}\hat{E}_R$
-
-

one block error. The thresholds for the block error rate are about 0.1–0.8% lower.

In our simulations, the toric code erasure decoder that we used in line 8 of Algorithm 1 and line 23 of Algorithm 2 is that of Delfosse *et al.* [20]. As can be seen from our channel model in Eq. (3), we have considered a pure erasure channel. In other words, the channel has only erasures and no errors. Therefore, nonzero syndromes must be adjacent to erasures. In this decoding algorithm, we choose a nonzero syndrome as the root node, and we construct a tree from the set of erased qubits. We repeat this process with another nonzero syndrome that is not in the constructed tree. This leads to a forest in the subset of erasures that contains all the nonzero syndromes. Then we use the peeling algorithm to estimate the errors on the erased qubits, which are then lifted back to the color code.

The map π gives rise to two toric codes. Although the error model on the color is independent, the error model on the toric codes is not. In our decoding algorithms, we decoded the errors on these copies independently. Clearly, this is suboptimal. Therefore, the performance of our decoding algorithms can be improved by decoding the two copies together. Alternatively, the erasures can be mapped to reflect the dependencies between the two copies of toric codes.

V. CONCLUSION

We have proposed methods to decode color codes over the quantum erasure channel. Along the way we have extended the results of [27], which shows how to map Pauli errors on color codes to Pauli errors on two copies of toric codes. More precisely, we have shown how to project syndromes and erasures on a color code to the associated pair of toric codes. While mapping the erasures, we used a naive method and an improved method reducing the number of erasures. Finally, we improved our decoders by jointly decoding on the color code and toric codes. Using this method, we obtained a threshold of $\gtrsim 44.3\%$ for the color code defined on the square octagonal lattice. We also observed that the joint decoding with the color code increases the threshold considerably even while using the naive erasure map. Our result of $\gtrsim 44.3\%$ is comparable to the result of Vodola *et al.* [19]. Considering that we are decoding the toric codes independently, it seems possible to improve the proposed decoders.

ACKNOWLEDGMENTS

This research was supported by the Science and Engineering Research Board, Department of Science and Technology under Grant No. EMR/2017/005454. We would like to thank Arjun Nitin Bhagoji for valuable discussions.

-
- [1] R. M. Gingrich, P. Kok, H. Lee, F. Vatan, and J. P. Dowling, *Phys. Rev. Lett.* **91**, 217901 (2003).
 - [2] J. Vala, K. B. Whaley, and D. S. Weiss, *Phys. Rev. A* **72**, 052318 (2005).
 - [3] B. H. Fong and S. M. Wandzura, *Quantum Inf. Comput.* **11**, 1003 (2011).
 - [4] J. J. Wallman, M. Barnhill, and J. Emerson, *Phys. Rev. Lett.* **115**, 060501 (2015).
 - [5] M. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
 - [6] C. H. Bennett, D. P. DiVincenzo, and J. A. Smolin, *Phys. Rev. Lett.* **78**, 3217 (1997).
 - [7] R. Cleve, D. Gottesman, and H.-K. Lo, *Phys. Rev. Lett.* **83**, 648 (1999).
 - [8] M. Grassl, T. Beth, and T. Pellizzari, *Phys. Rev. A* **56**, 33 (1997).
 - [9] T. C. Ralph, A. J. F. Hayes, and A. Gilchrist, *Phys. Rev. Lett.* **95**, 100501 (2005).
 - [10] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997.
 - [11] T. M. Stace, S. D. Barrett, and A. C. Doherty, *Phys. Rev. Lett.* **102**, 200501 (2009).
 - [12] T. M. Stace and S. D. Barrett, *Phys. Rev. A* **81**, 022317 (2010).
 - [13] A. C. Whiteside and A. G. Fowler, *Phys. Rev. A* **90**, 052316 (2014).
 - [14] S. Lloyd, P. Shor, and K. Thompson, [arXiv:1703.00382](https://arxiv.org/abs/1703.00382).
 - [15] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill, *J. High Energy Phys.* **06** (2015) 149.
 - [16] A. J. Moncy and P. K. Sarvepalli, in *International Symposium on Information Theory and Its Applications, ISITA 2018, Singapore, 2018* (IEEE, Piscataway, NJ, 2018), pp. 334–338.
 - [17] S. Muralidharan, C.-L. Zou, L. Li, J. Wen, and L. Jiang, *New J. Phys.* **19**, 013026 (2017).
 - [18] A. Kulkarni and P. K. Sarvepalli, *Phys. Rev. A* **100**, 012311 (2019).
 - [19] D. Vodola, D. Amaro, M. A. Martin-Delgado, and M. Müller, *Phys. Rev. Lett.* **121**, 060501 (2018).
 - [20] N. Delfosse and G. Zémor, [arXiv:1703.01517](https://arxiv.org/abs/1703.01517).
 - [21] N. Delfosse and G. Zémor, *Quantum Inf. Comput.* **13**, 793 (2013).

- [22] N. Delfosse, P. Iyer, and D. Poulin, [arXiv:1611.04256](#).
- [23] A. Kitaev, *Ann. Phys.* **303**, 2 (2003).
- [24] H. Bombin and M. A. Martin-Delgado, *Phys. Rev. Lett.* **97**, 180501 (2006).
- [25] B. Yoshida, *Ann. Phys.* **326**, 15 (2011).
- [26] H. Bombin, G. Duclos-Cianci, and D. Poulin, *New J. Phys.* **14**, 073048 (2012).
- [27] A. Bhagoji and P. Sarvepalli, in *2015 IEEE International Symposium on Information Theory (ISIT)* (IEEE, 2015).
- [28] A. Kubica, B. Yoshida, and F. Pastawski, *New J. Phys.* **17**, 083026 (2015).
- [29] J. Haah, *Rev. Colomb. Matemat.* **50**, 299 (2016).
- [30] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, *IEEE Trans. Inf. Theor.* **44**, 1369 (1998).