Local-dimension-invariant qudit stabilizer codes

Lane G. Gunderman [*]
Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1

(Received 5 March 2020; accepted 6 May 2020; published 28 May 2020)

Protection of quantum information from noise is a massive challenge. One avenue people have begun to explore is reducing the number of particles needing to be protected from noise and instead use systems with more states, so-called qudit quantum computers. These systems will require codes which utilize the full computational space. Many prior qudit codes are very restrictive on relations between the parameters of number of qudits, number of logical qudits, distance of the code, and number of bases. In this paper we show that codes for these systems can be derived from already known codes, often relaxing the constraints somewhat, a result which could prove to be very useful for fault-tolerant qudit quantum computers.

DOI: 10.1103/PhysRevA.101.052343

I. BACKGROUND

The ability to perform classical computation within an arbitrarily small error rate was shown by Shannon in the 1940s [1]. He provided a theoretical framework showing that modern classical computation would be possible. From that point, there arose a new challenge of finding actual codes that could best implement Shannon's result. This in turn pushed coding theory into a new realm, inspiring codes such as the Hamming code family and Bose-Chaudhuri-Hocquenghem (BCH) codes [2], and later leading to incredible ideas such as polar codes [3] and turbo codes [4].

As computational power progressed, there began to be investigations into the potential power of using quantum phenomena as a computational tool. This brought those same questions explored for classical computers back into question. This led to various ideas to try to bring over classical codes in some form or another. Among some of the earlier ideas was the stabilizer formalism [5], Calderbank-Shor-Steane (CSS) codes [6,7], and teleportation [8]. Many classical coding theory methods have been generalized into this new quantum setting, such as polynomial codes (a generalization of BCH and cyclic codes) [9,10], polar codes [11], and turbo codes [12,13]—including results such as a complete list of all perfect codes [14].

Building a quantum computer out of qudits (quantum objects with more than two levels) instead of qubits (quantum objects with only two levels) is an appealing option, since such a system would need comparatively few qudits to perform large quantum computations due to the larger computational space of each particle in such a system. In addition, context being the cause for the magic in quantum computation in the qudit case has been shown, whereas the case for qubits is still an open problem [15]. This has led to the characterization of magic-state distillation regions for qudits as well as fault-tolerant methods for such [16–18].

This means that we also need error-correction methods for these qudit systems. Prior work on qudit codes often depends on having a classical code which satisfies the conditions needed for CSS code construction, or a similar orthogonality requirement (such as [19–21]). This allows for the generation of many qudit quantum codes; however, at times these codes can require very strict relations between the number of bases for the particles (sometimes called the local dimension of the system) as defined in Definition 1, the number of particles, and the number of logical qudits. This can result in these codes being less useful for constructed qudit systems. This work aims to tackle this problem by working to reduce this level of restriction by allowing codes to be used for qudits of a different number of bases than they were initially designed for. In some regards one may consider this a tool somewhat similar in nature to CSS code construction: CSS allows classical to quantum code construction, whereas this allows for quantum to quantum code construction. In addition, this work may provide an avenue for determining whether a code is utilizing the qudit space particularly well.

Experimental realizations of qudit quantum computers have been progressing, as well as the theory of making such systems [22–24]. As these systems come online and grow there will be a need to have more flexibility in the set of codes that can be used to protect the information in these systems. In this article we primarily explore the ability to apply quantum error-correcting codes in smaller dimensional spaces onto systems with larger alphabets without having to discover codes for those systems through other methods, thus creating extensions of these already known codes into larger spaces.

Before we move on to discussing this problem, we must first define our mathematical language for working on these problems. Following that we introduce our results showing the ability to apply codes in larger spaces and then show the conditions required for preserving the distance of such codes, as well as a region where the distance of these codes can be preserved. We then propose some directions to carry out this work.

[*]lgunderman@uwaterloo.ca

II. DEFINITIONS

In this section we define the majority of the tools used in this paper. We recall common definitions and results for qudit operators.

A qubit is defined as a two-level system with states $|0\rangle$ and $|1\rangle$. We define a qudit as being a quantum system over q levels, where q is prime.

Definition 1. Generalized Paulis for a space over q orthogonal levels, where we assume q is prime, are given by

$$X_q|j\rangle = |(j+1) \mod q\rangle, \quad Z_q|j\rangle = \omega^j|j\rangle, \tag{1}$$

with $\omega = e^{2\pi i/q}$, where $j \in \mathbb{Z}_q$. These Paulis form a group, denoted $\mathbb{P}_q$.

When $q = 2$, these are the standard qubit operators. This group structure is preserved over tensor products, since each of these Paulis has order q.

Definition 2. An n-qudit stabilizer s is an n-fold tensor of generalized Pauli operators, such that there exists at least one state, $|\psi\rangle$, such that

$$s|\psi\rangle = |\psi\rangle, \tag{2}$$

where $|\psi\rangle \in \mathbb{C}^{q^n}$.

Definition 3. A stabilizer group $\mathbf{S}$ with commuting *generators* $\{s_i\}$ is defined as the subgroup of all n-qudit generalized Paulis formed from all multiplicative compositions ($\circ$) of these generators. This subgroup must not contain a nontrivial multiple of the identity.

Definition 4. We call a basis of orthonormal states $|\psi\rangle$ which satisfy the condition in Definition 2 for a stabilizer group $\mathbf{S}$ the codewords of the stabilizer.

Since each operator has order q, a collection of k compositionally independent generators for this stabilizer group will have q^k elements. Measuring the eigenvalues of the members in our stabilizer group, called the *syndrome*, of our state gives us a way to determine what error might have occurred and then undo the determined error. We recall for the reader, the well-known result:

Theorem 5. For any stabilizer code with k qudit stabilizers and n physical qudits, there will be q^{n-k} mutually orthogonal basis stabilizer states, or codewords.

This differs from the standard convention of k being the number of encoded qudits, since throughout this work we focus ourselves on the number of stabilizer generators. When discussing the errors that occur to our system, the standard choice of the depolarizing channel model focuses on the weights of the errors:

Definition 6. The weight of an n-qudit operator is given by the number of nonidentity operators in it.

Definition 7. A stabilizer code, specified by its stabilizers and stabilizer states, is characterized by a set of values:

(1) n: the number of qudits that the states are over

(2) $n-k$: the number of encoded (logical) qudits, where k is the number of stabilizers

(3) d [for nondegenerate codes (where all stabilizer group members have weight at least d)]: the distance of the code, given by the lowest weight of an undetectable generalized Pauli error (commutes with all stabilizer generators)

These values are specified for a particular code as: $[[n, n-k, d]]_q$, where q is the dimension of the qudit space.

We note that as long as no ambiguity exists, we suppress $\otimes$. We include $\otimes$ only to make register changes explicit.

Working with tensors of operators can be challenging, and so we make use of the following well-known mapping from these to vectors. This mapping is sometimes referred to as the symplectic representation, but we use alternative notation in this work to provide some notational flexibility utilized in this work. By representing these operators as vectors at times, the solution to a problem can become far more tractable.

Definition 8. (ϕ representation of a qudit operator). We define the surjective map

$$\phi_q : \mathbb{P}_q^n \mapsto \mathbb{Z}_q^{2n}, \tag{3}$$

which carries an n-qudit Pauli in $\mathbb{P}_q^n$ to a $2n$ vector mod q, where we define this map as

$$\begin{aligned}&\phi_q(\omega^\alpha \otimes_{i-1} I \otimes X_q^a Z_q^b \otimes_{n-i} I)\\ &= (0^{i-1}\, a\, 0^{n-i}|0^{i-1}\, b\, 0^{n-i}),\end{aligned} \tag{4}$$

which puts the power of the ith X operator in the ith position and the power of the ith Z operator in the $(i+n)$th position of the output vector. Throughout we will assume that $\mathbb{Z}_q$ takes values in $\{0, \ldots, q-1\}$. This mapping is defined as a homomorphism with $\phi_q(s_1 \circ s_2) = \phi_q(s_1) \oplus \phi_q(s_2)$, where $\oplus$ is a componentwise addition mod q. We denote the first half of the vector as $\phi_{q,x}$ and the second half as $\phi_{q,z}$.

When $q = 2$ this is the standard mapping used in the qubit stabilizer formalism. We may invert the map ϕ_q to return to the original n-qudit Pauli operator with the global phase being undetermined. We make note of a special case of the ϕ representation:

Definition 9. Let q be the dimension of the initial system. Then we denote by ϕ_∞ the mapping

$$\phi_\infty : \mathbb{P}_q^n \mapsto \mathbb{Z}^{2n}, \tag{5}$$

where no longer are any operations taken mod some base, but instead carried over the full set of integers.

As a clear example showing how these two are different, consider these mappings acting on $X \otimes X^{-1}$:

$$\phi_2(X \otimes X^{-1}) = (1\,1\,|\,0\,0), \quad \phi_\infty(X \otimes X^{-1}) = (1\ -1\,|\,0\,0). \tag{6}$$

The first one takes -1 to 1, since we have required the entries to be either 0 or 1 since $q = 2$, whereas ϕ_∞ is allowed to have negative values for entries since we are no longer performing any mod operations, meaning that each entry just has to be an integer.

The ability to define ϕ_∞ as a homomorphism still (and with the same rule) is a portion of the results of this paper, shown in Theorem 12. Our definition of ϕ_q is the standard choice for working with stabilizers over q bases; however, our ϕ_∞ allows us to avoid being dependent on the number of bases our system has when working with our stabilizers. In general we will write a stabilizer as ϕ_q, perform some operations, then write it in ϕ_∞. We shorten this to write it as ϕ_∞ and can later select to write it as $\phi_{q'}$ for some prime q' by taking elementwise mod q'. When we provide no subscript for the representation, that implies that the choice is irrelevant. The commutator of two operators in this picture is given by the following definition:

Definition 10. Let s_i, s_j be two qudit Pauli operators over q bases. Then these commute if and only if

$$\phi_q(s_i) \odot \phi_q(s_j) = 0 \mod q, \tag{7}$$

where $\odot$ is the symplectic product, defined by

$$\phi_q(s_i) \odot \phi_q(s_j) = \oplus_k[\phi_{q,z}(s_j)_k\, \phi_{q,x}(s_i)_k - \phi_{q,x}(s_j)_k\, \phi_{q,z}(s_i)_k], \tag{8}$$

where standard integer multiplication mod q is used and $\oplus$ is addition mod q.

When $q = 2$, this becomes the standard commutation relations between qubit Pauli's and is particularly simplified since addition and subtraction mod 2 are identical. Before finishing, we make a brief list of some possible operations we can perform on our ϕ representation for a stabilizer group:

1. As remarked above, we may add rows of the stabilizer generator matrix together, which corresponds to composition of operators
2. We may swap rows, corresponding to permuting the stabilizers
3. We may multiply each row by any number in $\{1, \dots, q-1\}$, corresponding to composing a stabilizer with itself. Since all operations are done over a prime number of bases, each number has an inverse.
4. We may swap registers (qudits) in the following ways:
   (a) We may swap columns $(Reg\ i, Reg\ i+n)$ and $(Reg\ j, Reg\ j+n)$ for $0 < i, j \leqslant n$, corresponding to relabelling qudits.
   (b) We may swap columns $Reg\ i$ and $(-1) \cdot Reg\ i + n$, for $0 < i \leqslant n$, corresponding to conjugating by a Hadamard gate on register i (or discrete Fourier transforms in the qudit case [25]), thus swapping X and Z's roles on that qudit.

All of these operations leave all parameters of the code alone but can be used in proofs. At this point we have all the necessary definitions to prove our results and have a solid base in qudit operators.

III. INVARIANT CODES

In this section we begin by defining *invariant* codes, which are codes that can be used for systems over any number of bases. Prior to this, only a few examples of invariant codes were known. Then we proceed to show that all qudit codes are invariant codes. This shows only that codes are valid over other spaces, so we then show that at least for sufficiently sized spaces all parameters of the code—particularly the distance—are at least preserved, if not even improved. We provide an argument about when the distance of the code will be improved. We finish by showing how to find the corresponding logical operators for these codes.

Definition 11. (Invariant codes). A stabilizer code is invariant if and only if

$$\phi_q(s_i) \odot \phi_q(s_j) = 0, \quad \forall i, j \tag{9}$$

holds for all primes q.

Qubit stabilizers *need* to commute only in the symplectic sense for $q = 2$ but *could* commute in the symplectic sense for $q = 3$, for example. To be an invariant code the stabilizers need to commute in the symplectic sense for all prime values of q. This is satisfied if $\phi_\infty(s_i) \odot \phi_\infty(s_j) = 0$ for all stabilizers s_i and s_j in the stabilizer group $\mathbf{S}$.

A. Motivating examples

Consider the following example of generators for a stabilizer group: $\langle XX, ZZ\rangle$. As a qubit code this forms a valid stabilizer code with codeword

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{10}$$

and the commutator of these generators can be seen to be $(1) + (1) = 2 \equiv 0 \mod 2$. Now suppose we wish to use this code for a qutrit system. In order to do that we must transform these generators into ones which have commutator 0; this can be achieved with $\langle XX^{-1}, ZZ\rangle$, whose powers are congruent mod 2 to the original code. In this case $\phi_\infty(X \otimes X^{-1}) \odot \phi_\infty(Z \otimes Z) = 0$. This means that not only can this be used for qutrits but for all prime numbers of bases. The codeword in the qutrit case is

$$\frac{|00\rangle + |12\rangle + |21\rangle}{\sqrt{3}}, \tag{11}$$

and the generalization of this for the codewords of a q-level system is a simple extension. We simply make each term in the codeword have the entries sum to a multiple of the qudit dimension so that the ZZ operator has a $+1$ eigenvalue:

$$\frac{1}{\sqrt{q}}\left(\sum_{j=1}^{q} |j \mod q, q-j \mod q\rangle\right). \tag{12}$$

If we look at the generators of this code, there is no single qudit operator that commutes with the generators; thus the distance of this invariant form of the code is still $d = 2$.

This is not the only example of a code that can be turned into invariant form. Another great example is the five-qubit code [26]. In fact, no changes are needed:

$$\langle XZZXI, \quad IXZZX, \quad XIXZZ, \quad ZXIXZ\rangle. \tag{13}$$

From inspection this can be seen to have commutators 0, and so this is a valid stabilizer code for qudits, and it can also be checked that this code will always have distance 3.

It is helpful to have a couple of examples; however, it has been unknown whether it is always possible to put stabilizer codes into an invariant form. We move forward from here to show that this can always be done and discuss a method of how to do this.

B. Embedding theorem statement and proof

We now show that all qudit stabilizer codes can be written in an invariant form.[1] This shows that we can form valid stabilizer groups over any number of bases but says nothing about the distance of these codes. This aspect is treated in the section immediately following.

[1]We acknowledge Andrew Jena for his contributions in the form of the following theorem and corollary.

Theorem 12. All qudit stabilizer codes can be transformed into invariant codes.

Proof. Let $\{s_1, \cdots, s_k\}$ be a set of stabilizer generators for a qudit code over q levels, with $k \leqslant n$ and q prime. We must construct a set of stabilizers, $\{s'_1, \cdots, s'_k\}$, such that

1. $\phi_\infty(s'_i) \equiv \phi_q(s_i) \mod q$, for all i
2. $\phi_\infty(s'_i) \odot \phi_\infty(s'_j) = 0$, for all $i \neq j$.

Without loss of generality, we assume that our stabilizers are given in canonical form:

$$\begin{pmatrix} \phi(s_1) \\ \vdots \\ \phi(s_k) \end{pmatrix} = (I_k\ X_2 | Z_1\ Z_2). \tag{14}$$

We define the strictly lower diagonal matrix L with entries

$$L_{ij} = \begin{cases} 0 & i \leqslant j \\ \phi(s_i) \odot \phi(s_j) & i > j \end{cases} \tag{15}$$

and define $s'_1, \cdots, s'_k$ such that

$$\begin{pmatrix} \phi(s'_1) \\ \vdots \\ \phi(s'_k) \end{pmatrix} = (I_k \quad X_2 \,|\, Z_1 + L \quad Z_2). \tag{16}$$

We show that $s'_1, \cdots, s'_k$ satisfies the conditions.

1. Since $\phi(s_i) \odot \phi(s_j) \equiv 0 \mod q$ for all $i \neq j$, we observe that $L_{ij} \equiv 0 \mod q$ for all entries. By adding rows of L to our stabilizers, we have not changed the code modulo q.
2. For $i > j$, we observe the following:

$$\begin{aligned} &\phi(s'_i) \odot \phi(s'_j) \\ &\quad = [\phi(s_i) + (0 \mid L_i\ 0)] \odot [\phi(s_j) + (0 \mid L_j\ 0)] \\ &\quad = \phi(s_i) \odot \phi(s_j) + \phi(s_i) \odot (0 \mid L_j\ 0) \\ &\qquad + (0 \mid L_i\ 0) \odot \phi(s_j) + (0 \mid L_i\ 0) \odot (0 \mid L_j\ 0) \\ &\quad = \phi(s_i) \odot \phi(s_j) + 0 - L_{ij} + 0 \\ &\quad = 0. \end{aligned}$$

■

Example 13. Consider the seven-qubit Steane code with parameters $[[7, 1, 3]]_2$, and denote it by Ξ [27]. The ϕ representation is given by

$$\phi_2(\Xi) = \begin{bmatrix} H & | & 0 \\ 0 & | & H \end{bmatrix}, \tag{17}$$

where H is the parity-check matrix for the classical Hamming code, given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{18}$$

We will make this an invariant code using the method shown in Theorem 12. We begin by putting this in standard form, performing operations mod 2:

$$\phi_2(\Xi) = \left[\begin{array}{ccccccc|ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{19}$$

Now that we have the code expressed in standard form, we construct the matrix containing the symplectic inner products, no longer taking operation over mod 2. The antisymmetric matrix $[\odot]$ representing the symplectic inner products between the stabilizers and the resulting L matrix for this code are given below:

$$[\odot] = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{20}$$

Adding this to our standard form, we have an invariant form for the Steane code given by

$$\phi_\infty(\Xi) = \left[\begin{array}{ccccccc|ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{21}$$

Since now all stabilizer generators from $\phi_\infty(\Xi)$ commute, this form of the code is a valid stabilizer code over any number of bases. We do not know, however, what the distance of this code is from this. We address this in Example 20.

We will want to know the size of the maximal entry in this invariant form for our bound on ensuring the distance of the code is at least preserved. The bound on the maximal entry is provided from the above proof:

Corollary 14. The maximal element in $\phi_\infty(S)$, B, is upper bounded by

$$[2+(n-k)(q-1)](q-1). \tag{22}$$

Proof. As before, we begin with S in standard form. For any $i \neq j$, there are at most $n-k$ entries, those entries which are not part of the identity portion of the standard form, in which both $\phi_{q,x}(s_i)$ and $\phi_{q,z}(s_j)$ are nonzero and bounded above by $q-1$, and a single entry, corresponding to the sole nonzero entry in the identity portion of the standard form, in which one is 1 whereas the other is bounded above by $q-1$. This gives us a bound on the inner product of $(n-k)(q-1)^2+(q-1)$. This is a bound on the size of an entry in our invariant stabilizer of $q-1+(n-k)(q-1)^2+(q-1)=[2+(n-k)(q-1)](q-1)$. ■

Example 15. In this example we show that CSS codes remain CSS codes under this transformation. Consider a general CSS code given by

$$\phi(\Xi)=\begin{bmatrix} I_{k_1} & X_{k_2} & X_{n-(k_1+k_2)} & | & 0 & 0 & 0 \\ 0 & 0 & 0 & | & Z_{k_1} & I_{k_2} & Z_{n-(k_1+k_2)} \end{bmatrix}, \tag{23}$$

where we have put the two block matrices into approximately standard form. We now perform Hadamards (or discrete Fourier transforms) on the k_2-sized middle blocks. We then have

$$\phi(\Xi)=\begin{bmatrix} I_{k_1} & 0 & X_{n-(k_1+k_2)} & | & 0 & X_{k_2} & 0 \\ 0 & I_{k_2} & 0 & | & Z_{k_1} & 0 & Z_{n-(k_1+k_2)} \end{bmatrix}. \tag{24}$$

Now, we note that the first k_1 stabilizers exactly commute with each other, i.e., inner product 0 in the ϕ_∞ sense, and likewise for the k_2 other stabilizer generators. Now we simply need to consider the case where we pick generators from each of the halves. We consider the matrix $[\odot]$, as above. This has nonzero entries for rows in k_2 when the columns are in k_1. Likewise for when the rows are in k_1, the entries are nonzero for columns in k_2. Thus we only add entries to Z_{k_1} and X_{k_2} with $[\odot]$ and, hence, certainly also for our L matrix. In fact, the L matrix adds entries only to Z_{k_1} since it is lower triangular. Given the new invariant form matrix, we may now invert our initial step of applying discrete Fourier transforms, and we will still have a CSS code.

C. Distance-preserving condition

Now that we know that all qudit codes can be put into an invariant form, we now prove that at least for most sizes of the space we can ensure that the distance of the code is at least preserved. We find a cutoff on the number of bases in the underlying space needed to at least preserve the distance.

Theorem 16. For all primes $p > p^*$, with p^* a cutoff value greater than q, the distance of an embedding of a nondegenerate stabilizer code $[[n, n-k, d]]_q$ into p bases, $[[n, n-k, d']]_p$, has $d' \geqslant d$.

Before proving this theorem we make a couple of nuanced definitions:

Definition 17. An unavoidable error is an error that commutes with all stabilizers and produces the $\vec{0}$ syndrome over the integers.

These correspond to undetectable errors that would remain undetectable regardless of the number of bases for the code, since they always exactly commute under the symplectic inner product with all stabilizer generators and thus all members of the stabilizer group. Since these errors are always undetectable, we call them unavoidable errors, since changing the number of bases would not allow this code to detect this error. This then provides the following insight:

Remark 18. The distance of a code over the integers is given by the minimal weight member in the set of unavoidable errors. The distance over the integers is represented by d^*, and so $d^* \geqslant d$. This value is also the minimum number of columns of the stabilizer generator matrix that are linearly dependent over the integers (or equivalently, over the rationals) in the symplectic sense.

We also define the other possible kind of undetectable error for a given number of bases, which corresponds to the case where some syndromes are multiples of the number of bases:

Definition 19. An artifact error is an error that commutes with all stabilizers but produces at least one syndrome that is only zero modulo the base.

These are named artifact errors, as their undetectability is an artifact of the number of bases selected and could become detectable if a different number of bases was used with this code. Each undetectable error is either an unavoidable error or an artifact error. We utilize this fact to show our theorem.

Proof. The ordering of the stabilizers and the ordering of the registers does not alter the distance of the code. With this, ϕ_∞ for the stabilizer generators over the integers can have the rows and columns arbitrarily swapped.

Let us begin with a code over q bases and extend it to p bases. The errors for the original code are the vectors in the kernel of ϕ_q for the code. These errors are either unavoidable errors or are artifact errors. We may rearrange the rows and columns so that the stabilizers and registers that generate these entries that are nonzero multiples of q are the upper left $2d \times 2d$ minor, padding with identities if needed. The factor of 2 occurs due to the number of nonzero entries in ϕ_∞ being up to double the weight of the Pauli. The stabilizer(s) that generate these multiples of q entries in the syndrome are members of the null space of the minor formed using the corresponding stabilizer(s).

Now, consider the extension of the code to p bases. Building up the qudit Pauli operators by weight j, we consider the minors of the matrix composed through all row and column swaps. These minors of size $2j \times 2j$ can have a nontrivial null space in two possible ways:

(1) If the determinant is 0 over the integers, then this is either an unavoidable error or an error whose existence did not occur due to the choice of the number of bases.

(2) If the determinant is not 0 over the integers but takes the value of some multiple of p, then it's $0 \mod p$ and so a null space exists.

Thus we can only introduce artifact errors to decrease the distance. By bounding the determinant by p^*, any choice of $p > p^*$ will ensure that the determinant is a unit in $\mathbb{Z}_p$, and will hence have a trivial null space, since the matrix is invertible.

Now, in order to guarantee that the value of p is at least as large as the determinant, we can use Hadamard's inequality to obtain

$$p > p^* = B^{2(d-1)}[2(d-1)]^{(d-1)}, \tag{25}$$

where B is the maximal entry in ϕ_∞. Since we only need to ensure that the artifact-induced null space is trivial for Paulis with weight less than d, we used this identity with $2(d-1) \times 2(d-1)$ matrices.

When $j = d$, we can either encounter an unavoidable error, in which case the distance of the code is d, or we could obtain an artifact error, also causing the distance to be d. It is possible that neither of these occur at $j = d$, in which case the distance becomes some d' with $d < d' \leqslant d^*$. ∎

Example 20. In our example of the Steane code, Corollary 14 tells us that the maximal entry is at most 3, but from our application of the method given in Theorem 12 we have $B = 1$, so we defer to this value since it is the true maximal entry value. The original distance was $d = 3$. This means that for all primes larger than $1^{2\times 2}(2\times 2)^2 = 16$ we are guaranteed that the distance is preserved. For primes below that value, we can manually check and apply alternate manipulations if needed. Given that all entries are ± 1, we know that the determinant of all the minors of interest are bounded by 4, and all primes at least as large as 5 preserve the distance. Through manual checking 3 is also not a possible minor determinant, so all primes preserve the distance for our invariant form of the Steane code.

To determine p^* for a given invariant code one can compute all the determinants of the minors; then so long as p is larger than the largest of these (or just not in this set), the distance will be preserved. We note that if p is in the set of minor determinants, the distance may still be preserved, since when considering the minors we have allowed them to be composed of entries from the X portion and the Z portion arbitrarily. If computing all these determinants is too computationally costly, one could just select the maximal entry in the invariant form instead of using the upper bound for B shown in Corollary 14, which will also generally greatly reduce the value of p^*.

We previously alluded to this proof that the code over the integers has distance at least as large. To determine how many bases are needed to ensure we have distance d^*, we simply extend our above result to obtain the cutoff expression, whereby no further distance improvements can be obtained from embedding the code—suggesting that another code ought to be used.

Corollary 21. For a nondegenerate stabilizer code we obtain the integer distance d^* when

$$p > B^{2(d^*-1)}[2(d^*-1)]^{d^*-1}. \tag{26}$$

After this value the distance cannot be improved through embedding. If d^* is unknown, this can be upper bounded by using k in place of d^*.

Proof. This follows from the above proof. The looser bound comes from $d^* \leqslant k$, so we can evaluate this at $d^* = k$ to obtain the loosest condition. ∎

The above provides a condition on the number of bases needed to ensure the distance of the code is at least preserved, but one could also ask, given an invariant code, whether that code can be used over fewer bases. We provide a bound on this with the following:

Lemma 22. For a nondegenerate code, for all $p < p^{**}$, with p^{**} a cutoff value less than q (possibly $\leqslant 2$), the distance of $[[n, n-k, d]]_q$ over p bases, $[[n, n-k, d']]_p$, must have $d' < d$.

Proof. Let $t = \lfloor \frac{d-1}{2} \rfloor$. The qudit quantum Hamming bound requires the initial code to satisfy

$$\sum_{j=0}^{t} \binom{n}{j} (q^2-1)^j \leqslant q^k. \tag{27}$$

Now we consider applying the code over p levels. Then we may bound

$$\binom{n}{t}(p^2-1)^t \leqslant \sum_{j=0}^{t} \binom{n}{j}(p^2-1)^j. \tag{28}$$

Likewise, when $p \geqslant 2$ we may bound

$$p^k \leqslant (p^2-1)^k. \tag{29}$$

Combining these we have

$$\binom{n}{t}(p^2-1)^t \leqslant (p^2-1)^k. \tag{30}$$

Then we violate the initial inequality if

$$p < \sqrt{1 + \binom{n}{t}^{1/(k-t)}} = p^{**}. \tag{31}$$

This means that p^{**} is only a valid bound when it is larger than 2, otherwise this result is trivially true since we no longer have a quantum code. ∎

Combining these results mean that distance *may* be preserved for $p^{**} \leqslant p < p^*$, while for $p > p^*$ it is guaranteed to have the distance preserved. For the region of values of p where the distance might be preserved, one can manually check and attempt another invariant form to try to make the distance preserved for the desired number of bases.

D. Invariant logical operators

Besides from the stabilizers, we also need logical operators to perform computations over the encoded qudits. Now we show how to construct such invariant logical operators.

Lemma 23. We may define invariant logical operators, $\mathcal{L}_\infty$, for the stabilizer code $\mathbf{S}$ as well.

Proof. Each logical operator is in $N(\mathbf{S})/\mathbf{S}$, the normalizer of $\mathbf{S}$ excluding $\mathbf{S}$, and there are $n-k$ X logical operators and $n-k$ Z logical operators. This means that we could, if we desired, generate a code $\mathbf{S}'$ whose generators are $\mathbf{S} \cup L_X$. This will have rank n and can be written in standard form as

$$[I_n|*], \tag{32}$$

meaning that L_X may be diagonalized within the last $n-k$ qudits. This can also be done with L_Z.

Then, since these logical operators are compositionally independent, they must be linearly independent in the ϕ representation, meaning $rank(L_X \cup L_Z) = 2(n-k)$. Now, if we take the standard form for $\mathbf{S}$ and append L_X, L_Z as additional rows we have

$$\begin{bmatrix} \mathbf{S} \\ L_X \\ L_Z \end{bmatrix} = \begin{bmatrix} I_k & A & | & B & C \\ 0 & D & | & E & F \\ 0 & G & | & H & J \end{bmatrix}. \quad (33)$$

From the above observation it is possible to compose the generators for L_X, L_Z to generate the matrix

$$\begin{bmatrix} I_k & A & | & B & C \\ 0 & I_{n-k} & | & E' & F' \\ 0 & G' & | & H' & I_{n-k} \end{bmatrix}. \quad (34)$$

At this point we focus on fixing the commutators between the elements of L_X and L_Z. Since the first k qudits will always contribute 0 to the commutator we drop those columns:

$$\begin{bmatrix} I_{n-k} & | & F' \\ G' & | & I_{n-k} \end{bmatrix}. \quad (35)$$

We can further reduce this to

$$\begin{bmatrix} I_{n-k} & | & 0 \\ 0 & | & I_{n-k} \end{bmatrix}. \quad (36)$$

This trivially satisfies the following required relations:

$$\phi_q(\bar{X}_i) \odot \phi_q(\bar{Z}_j) = \delta_{ij}, \quad (37)$$

$$\phi_q(\bar{X}_i) \odot \phi_q(\bar{X}_j) = \phi_q(\bar{Z}_i) \odot \phi_q(\bar{Z}_j) = 0, \quad \forall i, j. \quad (38)$$

Throughout these computations we have updated E' and H'. We now simply apply Theorem 12 to each logical operator in turn appended to $\phi(\mathbf{S})$. ■

Remark 24. This process does not alter our invariant stabilizer form, so our bound from earlier still holds.

IV. CONCLUSION AND DISCUSSION

This work introduces and lays the groundwork for qudit codes that can be used on systems with local dimension different than initially designed. This helps ease the restrictions that some qudit codes suffer from. We showed one method for generating these invariant codes but bring up the following example to motivate additional work on this:

Example 25. Throughout we have considered the method of creating invariant codes given by Theorem 12. With the following simple example we can show that $p^* = q$ for this method is not always possible. Consider the $[[4, 2, 2]]_2$ code generated by

$$\Xi = \langle XZXX, ZXZZ \rangle. \quad (39)$$

Following the method prescribed we obtain

$$\phi_\infty(\Xi) = \begin{bmatrix} 1 & 0 & 1 & 1 & | & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & | & -3 & 0 & 1 & 1 \end{bmatrix}. \quad (40)$$

This means that if we were to use this as a qutrit code the distance will drop to 1. This cannot be resolved by changing the choices of generators through compositions. If, however, we select as our generators

$$\Xi' = \langle XZXX, ZXZZ^{-1} \rangle, \quad (41)$$

then Ξ' is an invariant code and the distance of this code remains 2. Determining whether such a modification is always possible and whether it's possible to achieve this with a simple procedure are other open problems.

In this paper we have shown that qudit codes can be embedded into larger spaces, and at least for a sufficiently large number of bases, all parameters of the code are at least preserved. This result provides another tool for error-correction schemes for qudit quantum computers by providing immediate codes for these devices using modifications of already known codes.

Although in this work we find some critical value, p^*, above which all primes preserve the distance of the code, we believe that this result carries to all primes at least as large as the initial dimension if one uses other procedures to make the code invariant. Proving this, or at least tightening the bound on the critical value, seems like an important extension of this result, since the current bound can be quite large. In addition, there is the question of whether these results also hold for degenerate codes.

Some additional directions to carry these results include the following. Determining whether the prescribed method for generating invariant codes, or some other method, allows for transversality preservation—a crucial tool in fault-tolerant quantum computation. We also ask whether it is possible to take codes already known over q levels, and not a perfect code, and preserve the distance while using the code over $p < q$ levels.

ACKNOWLEDGMENTS

We would like to thank David Cory for useful suggestions, Daniel Gottesman for reading over an earlier draft and providing some useful directions and caveats to consider, and Andrew Jena for his help in proving the aforementioned theorem and corollary. We are grateful for financial support from the Canada First Research Excellence Fund, Industry Canada, CERC (Grant No. 215284), NSERC (Grant No. RGPIN-418579), CIFAR, Province of Ontario, and Mike & Ophelia Lazaridis.

[1] C. E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. **27**, 379 (1948).

[2] R. C. Bose and D. K. Ray-Chaudhuri, On a class of error correcting binary group codes, Information and control **3**, 68 (1960).

[3] E. Arikan, Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, IEEE Trans. Inf. Theory **55**, 3051 (2009).

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: Turbo-codes 1,

in *Proceedings of the ICC'93-IEEE International Conference on Communications* (IEEE, New York, 1993), Vol. 2, pp. 1064–1070.

[5] D. E. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology, 1997, https://resolver.caltech.edu/CaltechETD:etd-07162004-113028.

[6] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A **54**, 1098 (1996).

[7] A. M. Steane, Error Correcting Codes in Quantum Theory, Phys. Rev. Lett. **77**, 793 (1996).

[8] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, Nature (London) **402**, 390 (1999).

[9] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, arXiv:quant-ph/9906129.

[10] R. Cleve, D. Gottesman, and H.-K. Lo, How to Share a Quantum Secret, Phys. Rev. Lett. **83**, 648 (1999).

[11] M. M. Wilde and J. M. Renes, Quantum polar codes for arbitrary channels, in *2012 IEEE International Symposium on Information Theory Proceedings* (IEEE, New York, 2012), pp. 337–338.

[12] D. Poulin, J.-P. Tillich, and H. Ollivier, Quantum serial turbo codes, IEEE Trans. Inf. Theory **55**, 2776 (2009).

[13] M. M. Wilde, M.-H. Hsieh, and Z. Babar, Entanglement-assisted quantum turbo codes, IEEE Trans. Inf. Theory **60**, 1203 (2013).

[14] Z. Li and L. Xing, Classification of q-Ary perfect quantum codes, IEEE Trans. Inf. Theory **59**, 631 (2012).

[15] M. Howard, J. Wallman, V. Veitch, and J. Emerson, Contextuality supplies the 'magic' for quantum computation, Nature (London) **510**, 351 (2014).

[16] H. Anwar, E. T. Campbell, and D. E. Browne, Qutrit magic state distillation, New J. Phys. **14**, 063006 (2012).

[17] H. Dawkins and M. Howard, Qutrit Magic State Distillation Tight in Some Directions, Phys. Rev. Lett. **115**, 030501 (2015).

[18] E. T. Campbell, H. Anwar, and D. E. Browne, Magic-State Distillation in All Prime Dimensions Using Quantum Reed-Muller Codes, Phys. Rev. X **2**, 041021 (2012).

[19] A. Ketkar, A. Klappenecker, S. Kumar, and P. K. Sarvepalli, Nonbinary stabilizer codes over finite fields, IEEE Trans. Inf. Theory **52**, 4892 (2006).

[20] Y. Liu, R. Li, G. Guo, and J. Wang, Some nonprimitive BCH codes and related quantum codes, IEEE Trans. Inf. Theory **65**, 7829 (2019).

[21] X. Kai, S. Zhu, and P. Li, Constacyclic codes and some new quantum MDS codes, IEEE Trans. Inf. Theory **60**, 2080 (2014).

[22] P. J. Low, B. M. White, A. Cox, M. L. Day, and C. Senko, Practical trapped-ion protocols for universal qudit-based quantum computing, arXiv:1907.08569.

[23] P. Imany, J. A. Jaramillo-Villegas, M. S. Alshaykh, J. M. Lukens, O. D. Odele, A. J. Moore, D. E. Leaird, M. Qi, and A. M. Weiner, High-dimensional optical quantum logic in large operational spaces, npj Quantum Inf. **5**, 59 (2019).

[24] R. Sawant, J. A. Blackmore, P. D. Gregory, J. Mur-Petit, D. Jaksch, J. Aldegunde, J. M. Hutson, M. R. Tarbutt, and S. L. Cornish, Ultracold polar molecules as qudits, New J. Phys. **22**, 013027 (2020).

[25] D. Gottesman, Fault-tolerant quantum computation with higher-dimensional systems, in *NASA International Conference on Quantum Computing and Quantum Communications* (Springer, New York, 1998), pp. 302–313.

[26] H. F. Chau, Five quantum register error correction code for higher spin systems, Phys. Rev. A **56**, R1 (1997).

[27] A. Steane, Multiple-particle interference and quantum error correction, Proc. R. Soc. London A **452**, 2551 (1996).