arXiv:2303.17000v2 [quant-ph] 6 Feb 2024

Stabilizer Codes with Exotic Local-dimensions

Lane G. Gunderman

No affiliation for this work

February 5th, 2024

Traditional stabilizer codes operate over prime power local-dimensions. In this work we extend the stabilizer formalism using the local-dimension-invariant setting to import stabilizer codes from these standard local-dimensions to other cases. In particular, we show that any traditional stabilizer code can be used for analog continuous-variable codes, and consider restrictions in phase space and discretized phase space. This puts this framework on an equivalent footing as traditional stabilizer codes. Following this, using extensions of prior ideas, we show that a stabilizer code originally designed with a finite field local-dimension can be transformed into a code with the same n, k, and d parameters for any integral domain. This is of theoretical interest and can be of use for systems whose local-dimension is better described by mathematical rings, which permits the use of traditional stabilizer codes for protecting their information as well.

The necessity of generating protected quantum information is a central challenge in developing large, controlled quantum computing devices. One of the earliest theoretical methods for correcting quantum information is the stabilizer formalism which encapsulated many of the existing example schemes and put them into a general framework [1], [2]. This framework permitted classical codes to be imported in certain cases, now known as the Calderbank-Shor-Steane (CSS) Theorem [3], [4]. While the stabilizer formalism does not encapsulate all error-correction schemes, it provides a particularly useful mathematical framework, and so this work will focus on quantum error-correcting codes that formally are stabilizer codes, but may best be described with atypical local-dimensions–such as infinite, continuous, and those with structure best described by mathematical rings. This work provides some useful uses for the *local-dimension-invariant* (LDI) framework [5]–[7]. The final results in this work provide a general theorem somewhat similar to the CSS Theorem; whereas the CSS Theorem provides a method for bringing classical codes into quantum codes the results herein provide methods for using traditional quantum codes as quantum codes for other varieties of quantum systems.

There are two broad results that are focused on in this work. First, this work provides a solution to the long-standing open question of how to generally construct analog continuous-variable (CV) stabilizer codes [8], [9]. These codes map n physical continuous spaces into k protected continuous spaces with the ability to recover from up to $\lfloor \frac{d}{2} \rfloor$ registers being damaged. Second, this work also generally provides methods for bringing stabilizer codes from any field to any selection of integral domain local-dimension. This setting is useful for modular-qudit stabilizer codes and some other novel settings [10]–[14]. This latter result supersedes the former, but due to the independent utility of the prior we separate it out and discuss various cases related to it.

Lane G. Gunderman: lanegunderman@gmail.com

This work is organized as follows. First definitions are laid out, including the LDI framework, then we proceed to state our primary mathematical results. Following this, in Section 2 we proceed to prove these mathematical results in various cases. We begin by showing the ability to use stabilizer codes for analog CV codes, then consider the two non-trivial approximate versions of this setting–where the continuous space is not infinite, and where the infinite space is not continuous. The section then proceeds to prove the general result of being able to promise the distance of the code over certain mathematical rings. Then in Section 3, we discuss further possible extensions of the results to broader mathematical objects, including some discussion of topological codes which do not require the full distance promise of our proofs, as well as concluding remarks.

1 Definitions and Background

1.1 Mathematical Definitions

A mathematical, commutative field, in this work denoted by $\mathcal{F}$, is equipped with two invertible operations: addition and multiplication. A mathematical commutative ring [1], in this work denoted by $\mathcal{R}$, is equipped with an invertible addition operation and a multiplication operation, but not all elements have multiplicative inverses. Then in this work $\mathcal{F} \subset \mathcal{R}$, and in particular we will require $\mathcal{F}$ to have a finite characteristic. As this work extends to cases beyond qubits as well as qudits, we will use the more general term *registers* in place of these terms [15].

1.2 Definitions for Stabilizer Codes

In the discrete case with finite local-dimension q, the traditional qudit Pauli operators are generated by:

$$X|j\rangle = |(j+1) \mod q\rangle, \quad Z|j\rangle = \omega_q^j |j\rangle, \quad \omega_q = e^{2\pi i/q}. \tag{1}$$

We denote by $\mathbb{P}_q$ the single qudit Pauli operator group generated by the operators X and Z, while $\mathbb{P}_q^n$ indicates a tensor product of n such operators. In this work we will be varying the local-dimension, particularly focusing on the cases of formally infinite local-dimension, acting over $\mathbb{Z}$, and the continuous and simultaneously infinite local-dimension, acting over $\mathbb{R}$, cases. As our initial case we will consider q as a prime number so that all nonzero values have a unique multiplicative inverse.

Definition 1. *A stabilizer code, specified by its $n-k$ generators acting on n registers, is characterized by the following set of parameters:*

- *n: the number of (physical) registers that are used to protect the information.*
- *k: the number of encoded (logical) registers.*
- *d: the distance of the code, given by the lowest weight of an undetectable generalized Pauli error. An undetectable generalized Pauli error is an n-qudit Pauli operator which commutes with all elements of the stabilizer group, but is not in the group itself.*

[1] Throughout the work we use the term mathematical ring instead of simply ring in order to avoid any possible physical interpretable models that appear as rings, but do not have ring local-dimensions.

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

Working with tensors of Pauli operators can be challenging, and so we make use of the following well-known mapping from these to vectors in symplectic spaces, following the notation from [5]. This representation is often times called the symplectic representation for the operators [16], [17], but we use this notation instead to allow for greater flexibility, particularly in specifying the local-dimension of the mapping. This linear algebraic representation will be used for our proofs.

Definition 2 (ϕ representation of a qudit operator)**.** *We define the linear surjective map:*

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

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

$$I^{\otimes i-1} X^a Z^b I^{\otimes n-i} \mapsto \left(0^{i-1}\ a\ 0^{n-i} \middle| 0^{i-1}\ b\ 0^{n-i}\right), \tag{3}$$

which puts the power of the i-th X operator in the i-th position and the power of the i-th Z operator in the $(n+i)$-th position of the output vector. This mapping is defined as a homomorphism with: $\phi_q(s_1 s_2) = \phi_q(s_1) \oplus \phi_q(s_2)$, where $\oplus$ is component-wise addition mod q. We denote the first half of the vector as $\phi_{q,x}$ and the second half as $\phi_{q,z}$.

Equivalently we can state this mapping as:

$$\phi_q\left(\bigotimes_{t=1}^{n} X^{a_t} Z^{b_t}\right) = \left(\bigoplus_{t=1}^{n} a_t\right) \bigoplus \left(\bigoplus_{t=1}^{n} b_t\right), \tag{4}$$

where in the above $\bigoplus$ is a direct sum symbol. We will write a vertical bar between the vector for the X powers and the vector for the Z powers mostly for ease of reading.

In the infinite case we may still define Pauli operators, denoted as $\mathbb{P}_\infty$, although with some slight adjustments to avoid it being trivially true in the limit. We define the X and Z operators in the natural way, however, the root of unity used must be somewhat carefully selected so that it has infinite multiplicative order:

$$X|j\rangle = |j+1\rangle, \quad X^{-1}|j\rangle = |j-1\rangle, \quad Z|j\rangle = \omega_\infty^j |j\rangle, \quad \omega_\infty := e^{2\pi i/\sqrt{2}} \tag{5}$$

Note that we selected $\sqrt{2}$ as the denominator in the phase, but any irrational number would suffice. This is since we wish for ω_∞ to have unbounded multiplicative order and an inverse. If we kept with the traditional definition for ω_q but took the limit of $q \to \infty$, formally we would have $\omega_\infty = 1$, which would force X and Z to commute in the infinite case. In this case we have $XZ = \omega_\infty ZX$ and non-commutative properties still hold.

We may invert the map to return to the original n-qudit Pauli operator with the phase of each operator being undetermined. We make note of a special case of the ϕ representation:

Definition 3. *Let q be the dimension of the initial system. Then we denote by ϕ_∞ the mapping:*

$$\phi_\infty : \mathbb{P}_\infty^n \mapsto \mathbb{Z}^{2n} \tag{6}$$

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

The ability to still define ϕ_∞ as a homomorphism, upon quotienting out the leading phase of the operators, (and with the same mapping) is a portion of the results of [5]. Priorly this had only been defined so long as a new local-dimension was selected, however, in this work we formally show how this can be defined for codes with local-dimension given by the integers. ϕ_q is the standard choice for working over q bases, however, our ϕ_∞ allows us to avoid being dependent on the local-dimension of our system when working with our code.

The commutator of two operators in this picture is given by the following definition:

Definition 4. *Let s_i, s_j be two qudit Pauli operators over q bases, then these commute if and only if:*

$$\phi_q(s_i) \odot \phi_q(s_j) \equiv 0 \pmod{q} \tag{7}$$

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

$$\phi_q(s_i) \odot \phi_q(s_j) = \oplus_{k=1}^{n} [\phi_{q,z}(s_j)_k \cdot \phi_{q,x}(s_i)_k - \phi_{q,x}(s_j)_k \cdot \phi_{q,z}(s_i)_k] \tag{8}$$

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

Equivalently the commutator of two generalized Paulis, p_1 and p_2, is written and computed as $\phi(p_1) \odot \phi(p_2) = \vec{a}^{(1)} \cdot \vec{b}^{(2)} - \vec{b}^{(1)} \cdot \vec{a}^{(2)}$, with $\cdot$ as the standard dot product. This is not formally a commutator, but when this is zero, or zero modulo the local-dimension, the two operators commute, while otherwise it is a measure of the number of times an X operator passed a Z operator without a corresponding Z operator passing an X operator. When $q = 2$, this becomes the standard commutation relations between qubit Pauli operators and is particularly simplified since addition and subtraction mod 2 are identical.

In this work, we will assume that a stabilizer code is prepared in canonical form, given by $[I_{n-k}\ X_2 \mid Z_1\ Z_2]$, where X_2 is a $(n-k) \times k$ block from putting the code into this form, while Z_1 is a $(n-k) \times (n-k)$ block and Z_2 is a $(n-k) \times k$ block also resulting from putting $\mathcal{S}$ into canonical form. So long as the local-dimension is a prime number, this can be done, although not uniquely per se. These form our basic tools for analyzing stabilizer codes.

1.3 Local-dimension-invariant Codes

Before delving into new results we will provide a brief summary of prior results for our primary tool of local-dimension-invariant stabilizer codes as introduced in [5]. As an introductory example we may consider the qubit stabilizer code generated by the generators $\langle XX, ZZ \rangle$. Taking the symplectic product over the integers we find: $\phi_\infty(XX) \odot \phi_\infty(ZZ) = 2$. This symplectic product means that as a qubit code these operators commute, but they fail to commute as generators for a qutrit code. If we instead used the generator $\langle XX^{-1}, ZZ \rangle$, the symplectic product is now 0, so these generators commute regardless of the local-dimension choice. In particular, for any choice of local-dimension q, this produces a single stabilized codeword given by:

$$\frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} |j, q-j \mod q\rangle, \quad \text{with} \quad Z|j\rangle = e^{2\pi i j/q}|j\rangle \tag{9}$$

This then inspires the following definition:

Definition 5. *A stabilizer code $\mathcal{S}$ is called **local-dimension-invariant** iff:*

$$\phi_\infty(s_i) \odot \phi_\infty(s_j) = 0, \quad \forall s_i, s_j \in \mathcal{S} \tag{10}$$

While a few examples happened to have been known prior to this line of inquiry, they were neither known by this name nor framework. This is likely why the prime examples were either the short 5-register code [18] or the concatenated 9-register code [19]. These examples, along with the above and other simpler codes, are not the only local-dimension-invariant codes. The following theorem was shown in [5], but for completeness we provide it here as well since we will need to modify it later. This Theorem provides a prescriptive method for transforming any stabilizer code into an LDI form:

Theorem 6. *Any stabilizer code $\mathcal{S}$, with parameters $[[n,k,d]]_q$ for prime q, can be put into a local-dimension-invariant form.*

Proof. We provide a constructive method for transforming a given stabilizer code $\mathcal{S}$ into a local-dimension-invariant form $\mathcal{S}'$. The primitives for a finite field Gaussian elimination is obtained through the composition of generators in the ϕ representation, swapping rows, applying discrete Fourier transform gates (to swap X and Z powers, up to sign), and swapping registers. By performing this Gaussian elimination we put the code into canonical form: $[I_{n-k}\ X_2 \mid Z_1\ Z_2]$, where X_2 is a $(n-k)\times k$ block from putting the code into this form, while Z_1 is a $(n-k)\times(n-k)$ block and Z_2 is a $(n-k)\times k$ block also resulting from putting $\mathcal{S}$ into canonical form.

Now consider the modified set of generators given by $\mathcal{S}' := [I_{n-k}\ X_2 \mid Z_1 + L\ \ Z_2]$, with L the lower triangular matrix with $L_{ij} = \phi_\infty(s_i)\odot\phi_\infty(s_j)$ for $i>j$, and 0 otherwise. $\mathcal{S}'$ exhibits two properties:

1. $\mathcal{S}' \equiv \mathcal{S} \pmod q$, meaning that the code is unaltered over the original local-dimension choice.

2. For all $s'_i, s'_j \in \mathcal{S}'$, we have:

$$\begin{aligned}
\phi_\infty(s'_i)\odot\phi_\infty(s'_j) &= (\phi_\infty(s_i)+(0\mid L_i))\odot(\phi_\infty(s_j)+(0\mid L_j)) && (11)\\
&= \phi_\infty(s_i)\odot\phi_\infty(s_j) - L_i\cdot\phi_\infty(s_j)_x + L_j\cdot\phi_\infty(s_i)_x && (12)\\
&= \phi_\infty(s_i)\odot\phi_\infty(s_j) - L_{ij} + L_{ji} && (13)\\
&= 0, && (14)
\end{aligned}$$

   where the last step comes from L being lower-triangular.

Therefore $\mathcal{S}'$ is in LDI representation for $\mathcal{S}$ and still has $n-k$ generators. □

This is not the only way to form an LDI representation. Some other methods are briefly discussed in [6]. For the sake of simplicity we will use the lower-triangular L matrix method when using a prescriptive method. Throughout this work we will feature the example of the Steane code as this code is not a priori local-dimension-invariant [20].

Permitting a slight abuse of the ϕ notation, an LDI form for the Steane code, which in Pauli form we denote by H_3 due to its correspondence with the classical Hamming code, is given by the following [7]:

$$\phi_\infty(H_3) = \left[\begin{array}{ccccccc|ccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&1&1&0&0&0&0&0&0&0&0\\
0&0&1&1&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&-1&1&-1&0&0&0\\
0&0&0&0&0&0&0&0&1&-1&0&1&-1&0\\
0&0&0&0&0&0&0&0&0&1&-1&0&-1&1
\end{array}\right]. \qquad (15)$$

We will work with this LDI representation since it is also in CSS form and only differs from the traditional Steane code in signs.

$\phi_\infty(H_3)$ represents a set of generators that each have pairwise symplectic product zero, meaning that these generators commute regardless of the local-dimension of the underlying system.

The prior only ensures that there are generators over these local-dimensions. In order to ensure that the distance of the quantum error-correcting code of the LDI representation is at least as large as that of the original stabilizer code, we reproduce the theorem from [5]:

Theorem 7. *For all primes $p > p^*$, with p^* a cutoff value greater than q, the distance of an LDI form for a stabilizer code $[[n, k, d]]_q$ used for a system with p bases, $[[n, k, d']]_p$, has $d' \geq d$.*

To make claims about the distance of the code we begin by breaking down the set of undetectable errors into two sets. These definitions highlight the subtle possibility of the distance reducing upon changing the local-dimension.

Definition 8. *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 local-dimension for the code since they always exactly commute under the symplectic inner product with all stabilizer generators–and so all members of the stabilizer group. Since these errors are always undetectable we call them unavoidable errors since changing the local-dimension would not allow this code to detect this error.

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 local-dimension:

Definition 9. *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 local-dimension selected and could become detectable if a different local-dimension were used with this code. Generally there is no relation between artifact errors for different local-dimension choices. As we will see, however, for very large local-dimensions this set becomes empty. 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. Before proceeding, let us flip the signs of all entries in the Z component so that the symplectic product can be treated as a regular inner product, and symplectic kernels are replaced by traditional kernels of vector spaces.

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 (including the stabilizer group itself) or are artifact errors. We may rearrange the rows and columns so that the stabilizers and registers that generate these syndrome values that are nonzero multiples of q are the upper left $2d \times 2d$ minor, padding with identities (zero vectors in the ϕ representation) 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, upon taking the

product with an error, form the rows for a matrix minor for which there is a null space, and for which the number of columns used are given by the weight of the Pauli operator. In essence, we form a matrix minor from entries in the stabilizers in the support of the error, and suppose that such a matrix has a null space. This null space corresponds to artifact errors.

Now, consider the extension of the code to p bases. Building up the errors by Pauli weight, considering weight j, we consider the minors of the stabilizer matrix composed through all row and column swaps. This is an over complete method as this generates not only all minors with support corresponding to errors but any possible minor, however, it suffices. These minors of size $2j \times 2j$ can have a nontrivial null space in two possible ways:

- 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 local-dimension.
- 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 hence has 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:

$$p > p^* = B^{2(d-1)}(2(d-1))^{(d-1)} \tag{16}$$

where B is the maximal entry in ϕ_∞ [2]. Since we only need to ensure that the artifact induced null space is trivial for Pauli operators 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 $d < d'$. □

While the distance may increase if all weight d errors are artifact errors and are avoided through the local-dimension change, it cannot do so indefinitely.

Definition 10. *The distance d^* of a stabilizer code is given by the minimal weight unavoidable error.*

The value of d^* for a code effectively provides a limit to the distance of the quantum error-correcting code that cannot be overcome by increasing the local-dimension of the code. In the infinite local-dimension case, with or without continuous bases, we will always achieve this d^* value, as we will show. A large portion of the remaining results in this work are effectively variations of this proof with different settings, as well as some expanded settings for the results.

[2] Maximal here is taken as the largest absolute value over the reals. Depending on the method used for generating the LDI representation this can be bounded above by $\max_{ij} |\phi_\infty(s_i) \odot \phi_\infty(s_j)|$ or $(2 + k(q-1))(q-1)$.

1.4 Statement of Results

As a number of nuances in definitions must be made and the proof of these results may not be of particular interest to all readers, we separate out the primary results here, aside from this first Theorem which shows that if the local-dimension is a field then the results still hold. Being restricted to finite fields requires that the local-dimension just be a prime-power, but if one wishes to use stabilizer codes for other local-dimensions then we need to expand beyond fields to mathematical rings. The primary challenge of having mathematical ring local-dimensions is that the powers of the Pauli operators no longer are guaranteed to have multiplicative inverses. This next result generally provides that n and the number of linearly independent generators can be kept even if mathematical rings are the correct description for the local-dimension of a system. Unfortunately, since addition of generators in the ϕ representation corresponds to composition of generators, being linearly independent does not always suffice for being able to additively generate the full space of syndrome detection required. The common case of this is that of qudits with $q = p^m$, $m > 1$, whereby the syndromes that can be obtained are from $\mathbb{Z}_p$, so including each generator times all non-trivial powers of an irreducible factor permits the full error-correction capabilities [21]. Although other cases can occur, we will use this observation to obtain statements on the number of logical registers of the LDI code [22], [23]. In the case of finite, commutative mathematical ring local-dimensions, we may decompose the ring in a similar way such that each generator multiplies all non-trivial power of each irreducible polynomial–for instance for codes with local-dimension matching the structure of $\mathbb{Z}_6$ we would use all powers of the irreducible polynomials with order 2 and 3. Using the Chinese remainder theorem ring homomorphism then decomposes the $\mathbb{Z}_6$ local-dimension into $\mathbb{Z}_2 \oplus \mathbb{Z}_3$, which have characteristics 2 and 3, respectively.

Theorem 11. *Let $\mathcal{S}$ be a stabilizer code with parameters $[[n, k]]_{\mathcal{F}}$ for local-dimension given by finite field $\mathcal{F}$, with r linearly independent generators in the ϕ representation. Then there is a prescriptive method to transform $\mathcal{S}$ into a local-dimension-invariant form still with parameters $[[n, n-r]]_{\mathcal{R}}$ for any choice of a finite, commutative ring $\mathcal{R}$.*

Proof of Theorem 11. As in the initial proof of the canonical form [2][3], composition of generators corresponds to row additions, multiplying by a non-zero element of $\mathcal{F}$ generates an equivalent generator, rows may be swapped freely, as well as registers, and lastly X powers and Z powers for a register may be swapped (up to sign) by a discrete Fourier transform [24], generating a full set of elementary matrix operations. While in the prime power case formally only additivity is permitted, momentarily using linearity, finding an LDI form, then undoing the linearity used is permitted since the linearity would only be used to determine the entry changes required. Let there be r linearly independent generators. Using this we may perform Gaussian elimination to reduce the generators to canonical form: $[I_r \; X_2 \mid Z_1 \; Z_2]$.

The same lower-triangular L matrix as in the proof of Theorem 6 may be added to the Z_1 block to generate commuting generators which leave $\mathcal{S}$ alone over $\mathcal{F}$. Lastly, to preserve the independence number, r, when the vector space over $\mathcal{F}$ spanned by the generators is transformed into a module over a ring $\mathcal{R}$, it suffices to note that the first r entries are still independent for any ring $\mathcal{R}$, so all generators are still independent.

In the case of rings, since each generator, g, has syndrome values in $\text{char}(g)$, we generally must augment the generators from the canonical form in order to create an additively

[3]Whilst this initial proof did not form precisely this canonical form, it generates a form that differs by a discrete Fourier transform and some row additions, so we provide credit there.

spanning set. For a finite ring $\mathcal{R}$ the characteristic of any element divides $|\mathcal{R}|$, so including cosets of the generator g multiplied by powers of an irreducible factor for each characteristic factor, will create an additively spanning module. Then each generator will be replaced by $|\mathcal{R}|$ generators, which will result in $n - r$ logical registers–and thus an $[[n, n-r]]_{\mathcal{R}}$ quantum error-correcting code. □

First, it is worth noting that this reduces to the prior LDI construction theorem, here Theorem 6. It is also worth noting that this does not preclude stabilizer codes over broader mathematical objects from also being able to be put into an LDI representation. A particularly notable instance are codes over the integers, $\mathbb{Z}$, where the mathematical ring is infinite, however, the linearly independent generators still form an additively spanning set. The central element needed for other mathematical objects is the ability to put the code into canonical form, so even if multiplicative inverses do not exist for the initial local-dimension, if the code is in canonical form we may still put it into an LDI representation and ensure the rank. A possible method to include composite local-dimensions as well could be to use a Smith normal form such as that in [25].

The following result provides a plethora of stabilizer codes for the analog CV quantum error-correcting scheme laid out in [8], [9]. It broadly says that any stabilizer code can be used as an analog CV quantum error-correcting code and will have the same number of registers and logical registers, and at least the same distance.

Theorem 12 (CV Analog Case)**.** *Let q be a prime number and $\mathbb{R}$ be the real numbers. Then any stabilizer code $[[n, k, d]]_q$ can be transformed into a CV analog code with parameters $[[n, k, d']]_{\mathbb{R}}$ with $d' = d^* \geq d$.*

This result is shown in 2.1, with d^* as in Definition 10. Prior to this result, there were relatively few known non-trivial examples of analog CV codes [26], [27], so this provides a tremendous choice of codes. While the above Theorem is stated as requiring the initial local-dimension to be a prime number, this is not strictly required and any mathematical field $\mathcal{F}$ can be the initial local-dimension instead. The prior theorem is formally just a special case of the following theorem:

Theorem 13 ($\mathcal{F} \mapsto \mathcal{R}$ Case)**.** *Let $\mathcal{F}$ be a finite field and $\mathcal{R}$ be an integral domain. Then if $char(\mathcal{R}) = 0$ or $char(\mathcal{R}) > p^*$, with p^* some cutoff value that is a function of $char(\mathcal{F})$, n, k, and d, a stabilizer code $[[n, k, d]]_{\mathcal{F}}$, with r linearly independent generators in the ϕ representation, can be used as a $[[n, n-r, d']]_{\mathcal{R}}$ stabilizer code with $d' \geq d$.*

This Theorem, shown in 2.4, covers the various restrictions of the CV Analog code case: continuous-variable but bounded, integer local-dimension, and qudits. Additionally, it permits for codes without physical realizations but may be of theoretical interest which are best described by integral domains. Unfortunately, composite local-dimensions are not able to be shown here and so are left as a future direction. Lastly, through a synthesis of cases, rotor codes and their parameters are able to be obtained.

A single example family of this Theorem's use is to transform the $[[2^N - 1, 2^N - 1 - 2N, 3]]_2$ stabilizer code into a $[[2^N - 1, 2^N - 1 - 2N, 3]]_{\mathcal{R}}$ stabilizer code for any ring $\mathcal{R}$ with characteristic greater than 2, an example that will be delved into slightly greater depth later along with other examples. Having stated our achieved results we now proceed to prove them and provide some settings for their possible uses.

2 Local-dimensions: the continuous, the infinite, and the exotic cases

While the initial version of stabilizer codes operated on qubits, this was extended to the qudit case in time, typically under the constraint that the local-dimension is still a prime or prime-power. In this section we show that the same formalism can be extended to a number of broader mathematical settings. In particular, we begin by showing how using the LDI framework one may import stabilizer codes for use as analog CV error-correcting codes. Following this we specialize to the case whereby the underlying phase space is able to be interpreted as being a repeating cell, which is equivalent to having quadrature operators upper bounded by some value. In this case the distance of the generated code is only promised so long as the upper bound is sufficiently large. This reflects a somewhat more realistic phase space model for analog CV codes as they will have finite extent in reality. After this specialization, we consider the other approximation to analog CV codes whereby the local-dimension is allowed to be infinite, but only taking discrete values–in essence codes over $\mathbb{Z}$. We find that codes with this constraint are rather reminiscent of GKP encodings and form lattice points for their codewords which must be tapered to be physically realizable. The restriction to both finite cell and discrete values returns the codes to the traditional qudit case, so we do not examine those here.

Following these various cases of analog CV codes, we proceed to consider an extension of the stabilizer formalism using the LDI framework and the insights gained from considering codes over $\mathbb{Z}$. Using a similar proof technique we are able to show the ability to utilize stabilizer codes which originate over a finite field local-dimension to systems with integral domain local-dimensions, and give a condition under which the distance of such codes will be at least as large as the original stabilizer code. This scenario will primarily appeal to quantum error-correcting code theorists, but could be of use for topological systems exhibiting local-dimensions which are best described by mathematical rings.

Throughout the sections we provide some examples to illustrate the results and attempt to contextualize the results within some possible physical settings.

2.1 Analog CV QECC: Continuous and Infinite

In this section we show how to import arbitrary stabilizer codes, both qubit and qudit, and use them for analog CV codes, in the style of Lloyd and Slotine [8] and Braunstein [9]. In the continuous-variable setting the Heisenberg group, displacements of the quadratures, defines the Pauli operators and the errors. To begin, we define the "additive" form for Pauli operators which replaces a tensor product of many Pauli operators with a sum of single quadrature terms. Mathematically the quadrature operators, sometimes labeled as x and p, satisfy the commutation relation $[x, p] = 1$. Physically these may be position and momentum, electric and magnetic fields, or other dual variables and the domain of possible values for these quadratures form the phase space. A vector of these quadratures, representing a set of modes, is specified by $\vec{x}$ and $\vec{p}$. These quadrature sums form nullifiers which generate the code space [27]. The following Lemma shows this mapping and proves that Pauli operators represented this way observe the same commutation conditions [4].

Lemma 14 (Product to additive form)**.** *Let $\mathcal{S}$ be a stabilizer code, with $\phi_\infty(\mathcal{S})$ being the integer symplectic representation. Then there is a commutation conserving mapping such that:*

$$s_i := X^{\vec{a}} Z^{\vec{b}} \mapsto \vec{a} \cdot \vec{x} + \vec{b} \cdot \vec{p} := \mathcal{A}[s_i] \tag{17}$$

[4] This result was independently shown in [26], but we show it here in our language.

with $\mathcal{A}$ being the additive form of the analog nullifier.

Proof. To show this, we must show that $\phi_\infty(s_i) \odot \phi_\infty(s_j) = 0$ iff $[\mathcal{A}[s_i], \mathcal{A}[s_j]] = 0$. First, the symplectic side states that:

$$\vec{a}_i \cdot \vec{b}_j - \vec{a}_j \cdot \vec{b}_i = 0. \tag{18}$$

The additive form can be written as:

$$\begin{aligned}
[\mathcal{A}[s_i], \mathcal{A}[s_j]] &= [\vec{a}_i \cdot \vec{x} + \vec{b}_i \cdot \vec{p}, \vec{a}_j \cdot \vec{x} + \vec{b}_j \cdot \vec{p}] && (19)\\
&= [\vec{a}_i \cdot \vec{x}, \vec{b}_j \cdot \vec{p}] + [\vec{b}_i \cdot \vec{p}, \vec{a}_j \cdot \vec{x}] && (20)\\
&= \sum_k a_{ik} b_{jk} [x_k, p_k] - b_{ik} a_{jk} [x_k, p_k] && (21)\\
&= \vec{a}_i \cdot \vec{b}_j - \vec{a}_j \cdot \vec{b}_i, && (22)
\end{aligned}$$

and so the two are equal and so if the operators commute in one representation they will in the other. □

With this mapping from Pauli to additive form, we may still represent the same stabilizer code with the same symplectic matrix, only noting that now instead of the entries representing powers of Pauli operators they now represent coefficients of quadrature operators. This now permits us to show the following theorem:

Theorem 15. *Let $\mathcal{S}$ be a stabilizer code with parameters $[[n, k, d]]_q$, for a prime q, then $\mathcal{S}$ can be used as an analog CV quantum error-correcting code with parameters $[[n, k, d']]_{\mathbb{R}}$, with $d' = d^* \geq d$.*

The conditions for transforming $\mathcal{S}$ into a valid set of nullifiers for an analog CV error-correcting code are provided above. In order for the nullifiers to commute, all that must be done is to transform a given stabilizer code into an LDI representation then turn it into an "additive" form. These, however, only provide for ensuring that the analog CV code uses n physical continuous modes to generate k logical continuous modes. To show that the distance of the transformed code is at least as good requires some observations. Recall Definitions 8 and 9, which broke the set of undetectable errors into different varieties, for this proof.

Proof. The important insight here is that the local-dimension in this case is effectively infinite, and so artifact errors inherently cannot be induced. Digging into this a little more, the traditional argument for preserving the distance of an LDI form for a code involves ensuring that one cannot generate any new kernels for the minors of size up to $2(d-1) \times 2(d-1)$ in the LDI representation. This is guaranteed by bounding the determinant using Hadamard's inequality, providing requirements on how large the local-dimension must be to avoid incidentally introducing an artifact error which reduces the distance of the code. In this case, assuming infinite precision and infinitely large coefficients, there is no way to introduce an artifact error, and so the distance of the analog code must be some d' with $d' \geq d$. In fact, the distance of these codes is d^*, the minimal weight unavoidable error, since only unavoidable errors determine the distance of the code generated. □

The LDI framework provided for a relatively easy proof of this Theorem. This means that, were one to apply this with recent good quantum Low-Density Parity-Check (qLDPC) results, one would only need to ensure that at most a constant fraction of the continuous

modes are corrupted, and one will be able to have a constant fraction of logical continuous modes [28]–[30].

Physically implementing quadratures which correspond to x_1+27p_1, for instance, could be quite challenging as this may correspond to a carefully calibrated balance of electric and magnetic fields or other dual charges. Fortunately, if one begins with a CSS code, there is always a CSS LDI representation that can be prescriptively generated [5]. If one uses that method for generating an LDI representation one can be assured that the quadrature operators for the nullifiers are each only sums of x or p operators.

With this general result shown, we proceed now onto a non-trivial example. For this example, we will utilize some of the work from [26], which in essence showed that if one had LDI generators one could make an analog CV code, but neither specified how to make LDI generators nor mentioned the distance of such codes in general, besides just stating their existence. However, the tools laid out for specifying the code space is of particular use here.

Example 16. *We begin by recalling the following LDI form for the Steane code:*

$$\phi_\infty(H_3) = \left[\begin{array}{ccccccc|ccccccc} 1&1&1&1&0&0&0&0&0&0&0&0&0&0\\ 0&1&1&0&1&1&0&0&0&0&0&0&0&0\\ 0&0&1&1&0&1&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&-1&1&-1&0&0&0\\ 0&0&0&0&0&0&0&0&1&-1&0&1&-1&0\\ 0&0&0&0&0&0&0&0&0&1&-1&0&-1&1 \end{array}\right]. \tag{23}$$

As Pauli generators this code is given by:

$$\langle XXXXIII,\ IXXIXXI,\ IIXXIXX,\ ZZ^{-1}ZZ^{-1}III,\ IZZ^{-1}IZZ^{-1}I,\ IIZZ^{-1}IZ^{-1}Z\rangle. \tag{24}$$

As quadrature operators the nullifiers are then:

$$\begin{aligned}\langle x_1+x_2+x_3+x_4,\ x_2+x_3+x_5+x_6,\ x_3+x_4+x_6+x_7,\\ p_1-p_2+p_3-p_4,\ p_2-p_3+p_5-p_6,\ p_3-p_4-p_6+p_7\rangle\end{aligned} \tag{25}$$

We begin by finding the logical $|0\rangle$ state. We begin with the all $|0\rangle$ physical state and apply powers of the nullifiers to it. We then obtain:

$$|0_L\rangle = \int_{a,b,c} \Upsilon(a,b,c,d,e,f)|a,a+b,a+b+c,a+c,b,b+c,c\rangle \tag{26}$$

$$= \int_{a,b,c,e} e^{i\cdot ce}|a,a+b,a+b+c,a+c,b,b+c,c\rangle. \tag{27}$$

In this instance the function Υ ended up being a function only of c and e.

The logical operators are given by $\bar{X} = XXXXXXX$ and $\bar{Z} = ZZ^{-1}ZZ^{-1}ZZ^{-1}Z$. As quadratures these are:

$$\bar{x} = x_1+x_2+x_3+x_4+x_5+x_6+x_7, \quad \bar{p} = p_1-p_2+p_3-p_4+p_5-p_6+p_7. \tag{28}$$

Applying these with powers α for $\bar{x}$ and β for $\bar{p}$, the logical mode values are then given by:

$$|(\alpha,\beta)_L\rangle = \int_{a,b,c,e} e^{i\cdot ce}e^{i\beta\alpha}|a+\alpha,a+b+\alpha,a+b+c+\alpha,a+c+\alpha,b+\alpha,b+c+\alpha,c+\alpha\rangle, \quad \alpha,\beta\in\mathbb{R}. \tag{29}$$

An error which is undetectable but moves within the codespace is given by: $p_5-2p_6+p_7$ or $x_5+x_6+x_7$. These each have weight 3, as the results predict.

Having covered this example, we now move on to the repeating cell case.

2.2 The Continuous but Wrapping Case

Here we consider a slight modification for analog CV codes where the phase space is either repeating cells in the phase space, or equivalently the quadratures are both upper-bounded by some value p–meaning that the values can be taken to be in the range $[0, p)$ [5]. Mathematically we are considering codes with bases which are continuous functions but which are supported in $[0, p)$. These will have codewords which are L^2-integrable functions with domain $[0, p) \times [0, p)$. It is worth noting that due to Fourier constraints [6] [31]. this is not physically possible, however, we will consider this mathematical setting for now and consider the physical setting which corresponds best to this setting, *i.e.* rotors, later in 3.2.

The following Corollary follows from applying the included proofs for Theorems 7 and 15. In particular, the distance preservation argument from Theorem 7 still holds even if the underlying field is the reals modulo p, although artifact errors can now be induced if p is not sufficiently large. Notably, the linear algebraic argument for avoiding the inducement of kernels still holds.

Corollary 17. *Let $\mathbb{R}_p^+$ be the non-negative real numbers below p, and let p^* be some cutoff value. Then a stabilizer code with parameters $[[n, k, d]]_q$, for q prime, can be transformed into a $[[n, k, d']]_{\mathbb{R}_p^+}$ code with $d' \geq d$ if $p > p^*$.*

Given the current expressions for p^*, this would mean that the spatial extent needed would go as $B^{2(d-1)}(2(d-1))^{(d-1)}$ in order to guarantee the distance of the code, with B being the maximal entry in the LDI representation, generally satisfying $B \leq \max_{i,j} |\phi_\infty(s_i) \odot \phi_\infty(s_j)|$. There is also the alternative expression given by $(B(q-1)(d-1)(1+(d-1)^2(q-1)^{d-1}(d-2)^{(d-2)/2}))^{d-1}$ [6] and in the case of CSS codes $B^{d-1}(d-1)^{(d-1)/2}$ [7]. The values for p^* can be quite large, however, depending on the level of precision achievable in the determination of the quadratures this may not be a particularly limiting factor. Settings for this may include quadratures for repeating cells or on physical rings (or loops).

Example 18. *In [7] it was shown that $p^* = 2$ for all members of the family of quantum codes derived from the classical Hamming family–those with parameters $[[2^N - 1, 2^N - 1 - 2N, 3]]_2$. Given that, for any value of $p \geq 2$ and $N \geq 3$ we may generate a family $[[2^N - 1, 2^N - 1 - 2N, \geq 3]]_{\mathbb{R}_p^+}$, which could be a particularly promising early candidate if precision is limited.*

As this is a spatial truncation of the general analog CV code case, we do not analyze any further examples here and instead proceed to the other specialized case of infinitely large bases but only discrete values.

2.3 The Formally Infinite, Discrete Case

For this case we will actually consider two different settings. We will consider a discretized version of analog CV codes whereby the quadratures can only take discrete values. We will also consider a version more reminiscent of Pauli based stabilizer codes and discuss possible interpretations of this. Before we specialize to either formalism, we will prove general results for these codes. For notation, we will write both of these codes using the

[5] If they have unequal extent then the smaller of the two should be used for this result.

[6] Note that the Fourier transform of a periodic, bounded function has support over the integers, and so both the function and its Fourier transform cannot be periodic. This result in generality is covered by the Pontryagin Theorem

parameterized notation $[[n, k, d]]_{\mathbb{Z}}$ and note that the additive form mapping can be used in both directions to obtain analog-like codes as well as Pauli-like codes.

This is an example of stabilizer codes originating over a field and being still able to protect information over a mathematical ring. While multiplicative inverses may not exist for all members of the ring, there are still additive inverses, which still permits the correction of errors.

The following Theorem provides the proof for ensuring the distance of the code, while the generators are constructed through the same LDI framework:

Theorem 19. *Let $[[n, k, d]]_q$ be a stabilizer code, for a prime q, then we may transform it into a $[[n, k, d']]_{\mathbb{Z}}$ stabilizer code with $d' = d^* \geq d$.*

Proof. A commuting set of generators is provided from the LDI representation. This proof centers around the same argument, whereby we only need to worry about artifact errors being induced. For any $(2t) \times (2t)$ matrix minor, corresponding to a Pauli of weight between t and $2t$, the determinant will either be zero or not. The former case corresponds to either unavoidable errors or errors which already would have existed for the original code. The latter case has some nonzero integer value, which were we to consider the generators as being over the rationals, an extension ring of the integers, would have a matrix inverse and thus full rank. Given that it has full rank, there is no set of coefficients over the rationals which would reduce the rank, and therefore there are no coefficients over the integers which can reduce the rank, meaning that the rows and columns in this matrix minor are linearly independent. This then means that there is no new kernel induced, and so no artifact errors are induced. This then means that the distance of this code will be at least as good and by definition will be d^*. □

From this proof we trivially obtain the following Corollary which is less physically interpretable, but nonetheless mathematically defined:

Corollary 20. *Let $[[n, k, d]]_q$ be a stabilizer code, for a prime q, then we may transform it into a $[[n, k, d']]_{\mathbb{Q}}$ stabilizer code with $d' = d^* \geq d$.*

Of course, the codewords for a code over the integers is not normalizable, although taking a Gaussian envelope or something similar can generate highly accurate approximate codewords, much in the same fashion as GKP codewords [32]. Taking codes over $\mathbb{Z}$ is very similar to a discrete, lattice-based version of the GKP encoding scheme. In this setting the X and Z operators act as translations in this discrete "phasespace". In this case, much like the GKP encoding, the codewords are points on a lattice although with nearest "taxicab"-distances ($\|\cdot\|_1$) between different lattice point codewords being d^*. This results in a phasespace separation between any pair of encoded logical states of at least $\sqrt{d^*}$, with additional distance provided by the powers on the logical operators, discussed further in Remark 21.

For the Pauli-like generators in the infinite case, one should use Pauli operators as defined in 5, or other irrational induced phases from commuting X and Z. This avoids trivializing the commutation relations, but still forms a (non-trivial) symplectic set of operators over discrete space.

For the quadrature interpretation the same additive form from Lemma 14 may be used. The primary change from the analog CV case is that all integrals are replaced by sums. In this setting the quadratures x and p act as discrete displacements by a unit value as opposed to continuous displacements. In this way these codes are strongly reminiscent of an ideal GKP code and may be a good setting for further studying GKP stabilizer-like encodings such as those in [33].

Remark 21. *If we are interested in using these integer codes as codes for continuous variable systems, the distance metric provided for stabilizer codes is not the most prudent selection. The codewords for the integer code will form the lattice points in phase space assuming delta peaks, but the distance between the codewords is not simply described by the Pauli weight of the logical operator typically since the powers of the operators are not assumed to be uniformly distributed. Instead, the distance in phase-space between the codewords will be given by the minimal geometric distance between the lattice of codewords. Let us write d_{ps} as the distance in phase space, then:*

$$\sqrt{d} \leq d_{ps} := \min_{\vec{a},\vec{b}\in\mathbb{Z}^n \,:\, X^{\vec{a}}Z^{\vec{b}}\in\mathcal{L}} \sqrt{\|\vec{a}\|^2 + \|\vec{b}\|^2}, \tag{30}$$

with $\mathcal{L}$ being the logical operators for the integer code. An example is shown in Figure 1, assuming $\vec{b} = 0$, this increases the lattice spacing by a factor of $\sqrt{14} \approx 3.74$. For the case of the Steane code the increased lattice spacing is $\sqrt{3} \approx 1.73$, as there is a weight-three CSS logical operator with all terms having power 1. This scheme could aid in continuous variable encodings to discrete bases. This result is rather similar to those in [34], which provides a concrete and stabilizer-grounded approach to the same problem, providing the ability to use whichever stabilizer codes, although neglecting formal symplectic continuous variable analyses. Explicit unification could be valuable. Using good qLDPC codes in this provides an analytic solution for the open problem of a capacity achieving Gaussian encoding channel from [35]. In the work [36] they find a heuristic method for generating GKP-like encodings with phase space distance $\Theta(\sqrt{n})$ while having a constant rate. While there are also some interesting cryptographic applications from their work, it is a heuristic. This work is not a heuristic method but a deterministic, constructive method and so retains low weights for checks if derived from a "good" qubit qLDPC, especially if some weight reduction procedure such as [37] is applied before being transformed into this GKP-like encoding.

We have now exhausted the various ways of restricting analog CV codes, aside from the trivial, and priorly studied, restriction to qudits. We finished these cases by considering codes formally over a mathematical ring, the integers, using a slight variation of the proofs used throughout this work. Next we take the idea further to show that the destination local-dimension can be a mathematical ring and the distance is preserved so long as the characteristic of the ring is sufficiently large.

2.4 Ring Case

Throughout this work we have considered a myriad of local-dimension cases, but now we reach the most exotic, and general, case in this work. We will show that any stabilizer code originating over a field local-dimension can be used as an additive quantum code, much like a stabilizer code, over mathematical rings and under specific conditions have the distance of this code be at least as large. As before, we will prove our general result then show an example to highlight a possible construction. Some example uses of this result for codes include [10], [14], [17], [38]–[40], the majority of which can be obtained from, or supplemented by, the results herein. Beyond these, many anyonic condensation rules can be utilized within this setting, providing a framework for combining them with already known stabilizer codes.

For this result it helps to briefly discuss fields of fractions for commutative rings with integral domains. While the integers are a mathematical ring, the associated field of fractions for the integers is the field of rational numbers. This concept holds more broadly

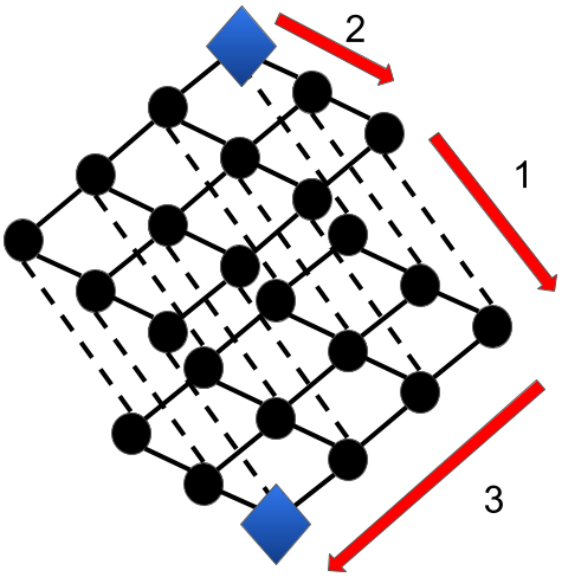

Figure 1: Above is a schematic of an integer code, assuming a code with lattice points only differing by X operators. The lattice would formally continue infinitely, but we only highlight this sublattice. Two codewords are placed on the blue diamonds. Their distance is $\sqrt{14}$, so their spacing is scaled by this factor and the inherent lattice spacing. This acts as an encoding of a discrete logical subspace in a possibly continuous phase space, or explicitly here a discretized phase space.

for certain mathematical rings. For an integral domain $\mathcal{R}$, which is a commutative ring within which $ab = ac$ $(a \neq 0)$ implies $b = c$, there is an associated field of fractions to it, denoted $Q(\mathcal{R})$, formed from ratios of elements of $\mathcal{R}$ and $\mathcal{R} - \{0\}$. The field of fractions is a field.

Theorem 22 ($\mathcal{F} \mapsto \mathcal{R}$ Case)**.** *Let $\mathcal{F}$ be a finite field and $\mathcal{R}$ be an integral domain. Then if char$(\mathcal{R}) = 0$ or char$(\mathcal{R}) > p^*$, with p^* some cutoff value that is a function of char$(\mathcal{F})$, n, k, and d, a stabilizer code $[[n, k, d]]_{\mathcal{F}}$, with r linearly independent generators in the ϕ representation, can be used as a $[[n, n-r, d']]_{\mathcal{R}}$ stabilizer code with $d' \geq d$.*

Before proving this Theorem it is worth noting that all finite integral domains are isomorphic to finite fields [7].

Proof. Theorem 11 provides a set of generators preserving full rank even upon the matrix rows being considered to generate a module. For the distance, one separates the set of undetectable errors into unavoidable errors and artifact errors and avoids inducing any artifact errors which would reduce the distance of the code. Consider errors with support in the symplectic module formed from $2j \times 2j$ entries, corresponding to support overlap with Pauli operators with weight j, randomly permuted. This module considered as a matrix may have zero determinant, which corresponds to either unavoidable errors or errors which already existed over $\mathcal{F}$. The zero value case is handled as before. Considering first the case of finite integral domains, where we use the isomorphism to finite fields. If the determinant is some value below the characteristic of the ring $\mathcal{R}$ then it will be some

[7] This can be seen from the requirement that for $c \neq 0$, $ac = bc$ implies $a = b$. Then due to being finite for each $r \in \mathcal{R}$ there must be some $n > m$ such that $r^n = r^m$ or equivalently $r^{n-m} = 1$, therefore each nonzero element has a multiplicative inverse and thus $\mathcal{R}$ is a field.

nonzero element in $\mathcal{R}$. If one considers the associated field of fractions for the ring, $Q(\mathcal{R})$, the matrix will be invertible, meaning that no new kernel is introduced, so the distance cannot be decreased in this way. Lastly a kernel can be induced if the determinant is a multiple of the characteristic of the ring $\mathcal{R}$. We may avoid this possibility by ensuring the characteristic of $\mathcal{R}$, $\mathrm{char}(\mathcal{R})$, satisfies:

$$\mathrm{char}(\mathcal{R}) > p^* = B^{2(d-1)}(2(d-1))^{(d-1)}, \tag{31}$$

where the expression is obtained from Hadamard's bound on the determinant of a $2(d-1)\times 2(d-1)$ matrix with maximal entry B. Therefore the distance in this case of the code will be at least as large as the original code's distance.

For infinite integral domains, all nonzero elements that the determinant may evaluate to has an associated inverse element in the field of fractions, $Q(\mathcal{R})$, and thus the rank of the module cannot be lowered. Therefore no artifact errors can be introduced and so the distance will be at least as large. □

In the above, one aspect worthy of some extra discussion is the expression for p^*, particularly the maximal entry B. If the field has finite characteristic then the prior bounds for B hold. If, however, the characteristic for the field is infinite, then one may replace the terms in the expression for B that are $(q-1)$ with the maximal value used. So for instance, were one using a code which originated over the rationals, $\mathbb{Q}$, and the largest entry in the canonical representation is $777/17$, then the $(q-1)$ terms would be replaced by this fraction in the bound. By adding this caveat, one may begin over a field local-dimension value, which does not have a finite characteristic. Alternatively, one may use the alternative bound for B based on the maximal symplectic product, if that LDI representation is used [6].

Some examples of other integral domains include: fields $\mathcal{F}$, prime-power local-dimensions p^m, polynomials, power series, and polynomials quotiented by prime ideals, p-adic integers, and the rings $\mathbb{Z}[\sqrt{n}]$ for non-square integers n. Note that the errors must also be from the same mathematical object as the code is written. While most of these local-dimensions are not physically observed, were one to find such a system or a theoretical framework where these would be considered, this would permit the distance promises to apply immediately.

Example 23. *In [7] it was shown that $p^* = 2$ for all members of the family of quantum codes derived from the classical Hamming family–those with parameters $[[2^N-1, 2^N-1-2N, 3]]_2$. Given that, for any integral domain $\mathcal{R}$ with char$(\mathcal{R}) \geq 2$, or char$(\mathcal{R}) = 0$, and $N \geq 3$ we may generate a family $[[2^N-1, 2^N-1-2N, \geq 3]]_{\mathcal{R}}$, so it could be a particularly promising early candidate for systems with this variety of local-dimension.*

3 Future Directions

In the cases where the distance of the code is only promised for sufficiently large local-dimensions, this method may be somewhat limited in its use. Below this value one would need to determine the distance through other means. One particularly readily analyzable setting is that of topological codes. We will briefly delve into this case now.

3.1 The Toric Code and Similar Topological Codes

All of the examples considered so far have been traditional stabilizer codes, however, topological codes can also be written in matrix form. In more near-term applications topological quantum error-correcting codes have become of greater interest. Additionally, topological

codes require some sense of spatially neighboring qubits, which we show can be leveraged here. Amongst the many topological codes is the toric code, proposed by Kitaev, which has a distance that goes as the square root of the number of qubits [41]. We examine the toric code here and show how we can put it into an LDI form that always preserves the distance, even if the local-dimension is a general commutative ring. This code was already considered in [42], but we put this result into the context of this work.

The toric code is constructed on a square grid of qubits and its dual square grid also as qubits. The code is CSS with the stabilizer generators given as X operators on the edges of each grid square, and Z operators on the cross from the intersection of squares, see Figure 2. We make the proposed modifications in the figure, flipping the power of opposing pairs of Pauli operators within each generator. This allows for compositions to still perfectly commute with all other generators. This simple change provides an LDI form for the code, but does not, yet, guarantee that the distance is preserved.

For the toric code, the logical operators are completely topologically winding strings of a single Pauli operator variety. This property gives the code the high distance as the weight of the logical operators is a lower bound on the distance for non-degenerate codes such as this. As an ansatz, we take these same logical operators and consider the coset equivalent forms for them upon interacting with the stabilizer generators. In Figure 3 we show two characteristic examples. The top example imagines applying the X stabilizer from below the middle X operator, which deforms the string. This is exactly the same action as the qubit case, except the left edge is a -1 power instead of $+1$. Consider, however, if we apply the same generator from above. Then the string is deformed like the qubit case but there is also an added *bridge* in the operator that is impossible in the qubit case (it disappears upon taking $\mod 2$). Now, this behavior could be extended and we could generate more bridges, however, no bridge can fully wind around the torus and generate a non-trivial action on the logical space. This means that the distance is still preserved, although these peculiar bridges appear for the non-qubit case. As of this time, it is not clear whether these bridges provide some possible other changes to the toric code, aside from singular strings not always being the logical operator, but rather strings with possible multiplicities in them.

A couple of notable features occur in this example which we can take for more general topological codes. Firstly, the generators and the logical operations required no properties of the underlying mathematical object other than additive inverses, so arbitrary rings can be used for the local-dimension, including composite valued local-dimensions and groups. This then provides the codes from [43], [44] as well as codes for broader mathematical objects than those considered. Having spatially local LDI generators ensures that one only has to satisfy the distance promise for subsets of these local-operators which provides for the global distance being preserved. Unfortunately as of this time there are no results on whether the same generators, just with uniform power changes for each generator, can always be obtained for topological codes, nor are there bounds on the level of locality loss from using a prescriptive method. However, this may prove a fruitful direction for future work. Additionally, the toric code example considered satisfies fusion rules for certain anyons. It is likely that changes to the powers in the generators and the shapes of the generators could obtain the results of [13]. This could aid in extending these results to non-topological codes and will also put these results more clearly within the context of stabilizer codes.

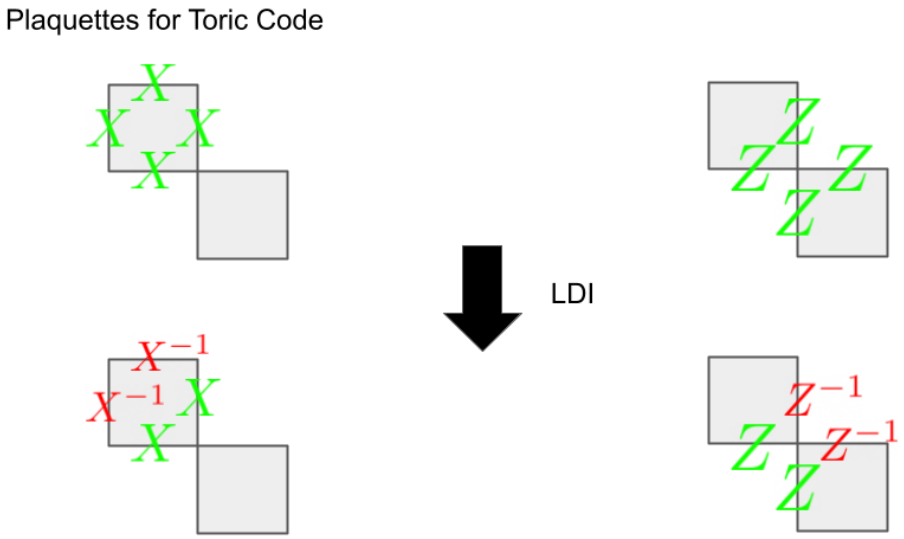

Figure 2: This figure shows the transformations on the stabilizing plaquettes for the qubit toric code to generate a toric code that works regardless of the local-dimension of the system. This transforms the code from a $[[2N^2, 2, N]]_2$ code into a $[[2N^2, 2, N]]$ code for any choice of local-dimension with additive inverses.

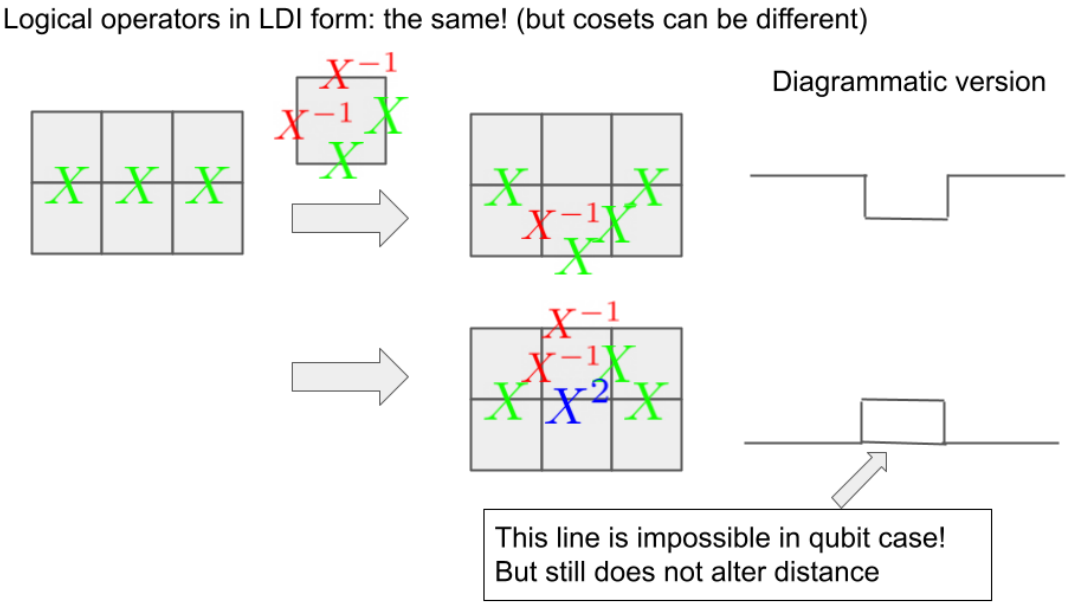

Figure 3: This figure illustrates the distance argument, showing that the distance of the code is preserved upon changing the local-dimension of the code. Notice that bridges generated are only existent in the non-qubit case, as modulo 2 they disappear and a mere distortion of the logical string operator occurs

.

3.2 Rotor Codes

An interesting possible use case for this work can be found in rotor systems and their codes. A rotor system has its two Pauli axes having different varieties of local-dimensions, in particular one is the truncated continuous space $\mathbb{R}^+_{2\pi} := [0, 2\pi)$, while the other is the integers $\mathbb{Z}$. There are a few physical settings where rotors best describe the underlying physical system. We direct interested readers to [45] for details on this. That work also described methods for generating codes for these systems through homological products, with some cases requiring the LDI condition. This work permits direct algebraic importing of stabilizer codes. In this work, we may combine Corollary 17 and Theorem 19 to cover systems like this, however, the distance for both axes can only be promised if $p^* < 2\pi$. This then leads to the following conjecture:

Conjecture 24. *Let $\phi_\infty(\mathcal{S})$ represent a good qubit LDPC code (with constant rate and relative distance). If $p^* < 2\pi$, $\phi_\infty(\mathcal{S})$ represents a good quantum LDPC rotor code.*

If the basis described by $\mathbb{Z}$ is more appropriately described by some finite portion of $\mathbb{Z}$ then the appropriate Theorem and p^* value should be used. The strongest way to show this conjecture would be to show that for a code with parameters $[[n, k, d]]_q$, the LDI form has its distance at least preserved for $p^* = q$, which would permit codes with local-dimensions 2, 3, and 5 to be used freely for rotor systems. However, showing this only for $q = 2$ would suffice for the above conjecture. The sparsity will be retained as the number of nonzero entries in L will be sparse. While the traditional stabilizer distance d is not always the best metric for continuous spaces, this value still provides information on what fraction of the registers may be lost and the information still reliably recovered.

3.3 Conclusion

In this work we have shown that the LDI extension of the traditional stabilizer formalism can be used to obtain some different settings of stabilizer-like codes. We showed that they permit the use of arbitrary stabilizer codes as analog CV codes and that these codes will keep their parameters. Additionally we considered the restriction to bounded phase space and integer precision, each of which correspond to different approximations of analog CV codes. In the case of only integer precision, forming a lattice, we find that the codes are remarkably reminiscent of GKP codes. Using these ideas we showed that stabilizer codes originating over a field local-dimension can have their distance still kept when used over integral domain local-dimension systems. These are of theoretical interest for quantum error-correcting code theorists, as they may exhibit interesting properties not observable for field local-dimension codes. These mathematical ring local-dimension codes can also be of use for physical systems whose Pauli operators best correspond to these mathematical objects.

Naturally, throughout this work one can pull in qLDPC results, although the sparsity, ρ nonzero entries at most per register or generator, would be roughly squared. These good qLDPC codes would still be good qLDPC codes upon being used in an LDI form since they would still have a constant number of nonzero entries as the index of the family increases. One may expect that a weight reduction technique similar to that of Hastings would exist in these other local-dimensions, and so could remove this quadratic increase in the check weights [37].

The full implications of the results in this work have not been explored and likely contain many other useful and interesting cases. Some possible future directions to carry this work would include connecting the LDI framework to anyonic theories, subsystem codes, and

GKP stabilizer codes, among other quantum error-correcting frameworks. It is possible that some of the other methods for quantum error-correction can be unified through the LDI framework. For more near-term uses properties such as fault-tolerance preservation could be useful features to show for LDI representations of codes. Additionally, for those results where p^* was involved, possible reductions in the expressions for this could prove very helpful. Lastly, we were able to show the ability to apply stabilizer codes over integral domains, but it could be interesting to extend the results to general commutative rings, which would cover composite local-dimension values as well as some other exotic possible local-dimensions.

Acknowledgments

We thank the following people for helpful discussions and ideas: E. Knill, A. Jena, M. Vasmer, G. Dauphinais, V. Albert, B. Terhal, C. Vuillot, and A. Ciani.

References

[1] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum hamming bound," *Physical Review A*, vol. 54, no. 3, p. 1862, 1996. DOI: 10.1103/PhysRevA.54.1862.

[2] D. Gottesman, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997. DOI: 10.48550/arXiv.quant-ph/9705052.

[3] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, p. 1098, 1996. DOI: 10.1103/PhysRevA.54.1098.

[4] A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, p. 793, 1996. DOI: 10.1103/PhysRevLett.77.793.

[5] L. G. Gunderman, "Local-dimension-invariant qudit stabilizer codes," *Physical Review A*, vol. 101, no. 5, p. 052 343, 2020. DOI: 10.1103/PhysRevA.101.052343.

[6] L. G. Gunderman, "Degenerate local-dimension-invariant stabilizer codes and an alternative bound for the distance preservation condition," *Physical Review A*, vol. 105, no. 4, p. 042 424, 2022. DOI: 10.1103/PhysRevA.105.042424.

[7] A. J. Moorthy and L. G. Gunderman, "Local-dimension-invariant calderbank–shor–steane codes with an improved distance promise," *Quantum Information Processing*, vol. 22, no. 1, p. 59, 2023. DOI: 10.1007/s11128-022-03792-3.

[8] S. Lloyd and J.-J. E. Slotine, "Analog quantum error correction," *Physical Review Letters*, vol. 80, no. 18, p. 4088, 1998. DOI: 10.1103/PhysRevLett.80.4088.

[9] S. L. Braunstein, "Error correction for continuous quantum variables," in *Quantum Information with Continuous Variables*, Springer, 1998, pp. 19–29. DOI: 10.1103/PhysRevLett.80.4084.

[10] A. Ashikhmin and E. Knill, "Nonbinary quantum stabilizer codes," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 3065–3072, 2001. DOI: 10.1109/18.959288.

[11] V. Gheorghiu, "Standard form of qudit stabilizer groups," *Physics Letters A*, vol. 378, no. 5-6, pp. 505–509, 2014. DOI: 10.1016/j.physleta.2013.12.009.

[12] S. S. Bullock and G. K. Brennen, "Qudit surface codes and gauge theory with finite cyclic groups," *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 13, p. 3481, 2007. DOI: 10.1088/1751-8113/40/13/013.

[13] T. D. Ellison, Y.-A. Chen, A. Dua, W. Shirley, N. Tantivasadakarn, and D. J. Williamson, "Pauli stabilizer models of twisted quantum doubles," *PRX Quantum*, vol. 3, no. 1, p. 010 353, 2022. DOI: 10.1103/PRXQuantum.3.010353.

[14] V. V. Albert, J. P. Covey, and J. Preskill, "Robust encoding of a qubit in a molecule," *Physical Review X*, vol. 10, no. 3, p. 031 050, 2020. DOI: 10.1103/PhysRevX.10.031050.

[15] J. Watrous, *The theory of quantum information*. Cambridge university press, 2018. DOI: 10.1017/9781316848142.

[16] D. A. Lidar and T. A. Brun, *Quantum error correction*. Cambridge university press, 2013. DOI: 10.1017/cbo9781139034807.

[17] A. Ketkar, A. Klappenecker, S. Kumar, and P. K. Sarvepalli, "Nonbinary stabilizer codes over finite fields," *IEEE transactions on information theory*, vol. 52, no. 11, pp. 4892–4914, 2006. DOI: 10.1109/TIT.2006.883612.

[18] H. Chau, "Five quantum register error correction code for higher spin systems," *Physical Review A*, vol. 56, no. 1, R1, 1997. DOI: 10.1103/PhysRevA.56.R1.

[19] H. Chau, "Correcting quantum errors in higher spin systems," *Physical Review A*, vol. 55, no. 2, R839, 1997. DOI: 10.1103/PhysRevA.55.R839.

[20] A. Steane, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, 1996. DOI: 10.1098/rspa.1996.0136.

[21] D. Gottesman, "Stabilizer codes with prime power qudits," in *invited talk at Caltech IQIM seminar (Pasadena, California)*, vol. 1, 2014, pp. 12–13. [Online]. Available: https://www.qec14.ethz.ch/slides/DanielGottesman.pdf.

[22] P. J. Nadkarni and S. S. Garani, "$\mathbb{F}_p$-linear and $\mathbb{F}_{p^m}$-linear qudit codes from dual-containing classical codes," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–19, 2021. DOI: 10.1109/TQE.2021.3078152.

[23] S. S. Garani, P. J. Nadkarni, and A. Raina, "Theory behind quantum error correcting codes: An overview," *Journal of the Indian Institute of Science*, pp. 1–47, 2023. DOI: 10.1007/s41745-023-00392-7.

[24] D. Gottesman, "Fault-tolerant quantum computation with higher-dimensional systems," in *Quantum Computing and Quantum Communications: First NASA International Conference, QCQC'98 Palm Springs, California, USA February 17–20, 1998 Selected Papers*, Springer, 1999, pp. 302–313. DOI: 10.1007/3-540-49208-9_27.

[25] R. Sarkar and T. J. Yoder, "The qudit pauli group: Non-commuting pairs, non-commuting sets, and structure theorems," *arXiv preprint arXiv:2302.07966*, 2023. DOI: 10.48550/arXiv.2302.07966.

[26] R. L. Barnes, "Stabilizer codes for continuous-variable quantum error correction," *arXiv preprint quant-ph/0405064*, 2004. DOI: 10.48550/arXiv.quant-ph/0405064.

[27] V. V. Albert, "Bosonic coding: Introduction and use cases," *arXiv preprint arXiv:2211.05714*, 2022. DOI: 10.48550/arXiv.2211.05714.

[28] P. Panteleev and G. Kalachev, "Asymptotically good quantum and locally testable classical ldpc codes," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022, pp. 375–388. DOI: 10.1145/3519935.3520017.

[29] A. Leverrier and G. Zémor, "Quantum tanner codes," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, 2022, pp. 872–883. DOI: 10.1109/FOCS54457.2022.00117.

[30] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, "Good quantum ldpc codes with linear time decoders," in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 2023, pp. 905–918. DOI: 10.1145/3564246.3585101.

[31] M. Stroppel, *Locally compact groups*. European Mathematical Society, 2006, vol. 3. DOI: 10.4171/016.

[32] D. Gottesman, A. Kitaev, and J. Preskill, "Encoding a qubit in an oscillator," *Physical Review A*, vol. 64, no. 1, p. 012 310, 2001. DOI: 10.1103/PhysRevA.64.012310.

[33] K. Noh, S. Girvin, and L. Jiang, "Encoding an oscillator into many oscillators," *Physical Review Letters*, vol. 125, no. 8, p. 080 503, 2020. DOI: 10.1103/PhysRevLett.125.080503.

[34] J. Conrad, J. Eisert, and F. Arzani, "Gottesman-kitaev-preskill codes: A lattice perspective," *Quantum*, vol. 6, p. 648, 2022. DOI: 10.22331/q-2022-02-10-648.

[35] J. Harrington and J. Preskill, "Achievable rates for the gaussian quantum channel," *Physical Review A*, vol. 64, no. 6, p. 062 301, 2001. DOI: 10.1103/PhysRevA.64.062301.

[36] J. Conrad, J. Eisert, and J.-P. Seifert, "Good gottesman-kitaev-preskill codes from the ntru cryptosystem," *arXiv preprint arXiv:2303.02432*, 2023. DOI: 10.48550/arXiv.2303.02432.

[37] M. B. Hastings, "On quantum weight reduction," *arXiv preprint arXiv:2102.10030*, 2021. DOI: 10.48550/arXiv.2102.10030.

[38] A. Niehage, "Quantum goppa codes over hyperelliptic curves," *arXiv preprint quant-ph/0501074*, 2005. DOI: 10.48550/arXiv.quant-ph/0501074.

[39] A. L. Grimsmo, J. Combes, and B. Q. Baragiola, "Quantum computing with rotation-symmetric bosonic codes," *Physical Review X*, vol. 10, no. 1, p. 011 058, 2020. DOI: 10.1103/PhysRevX.10.011058.

[40] P. Faist, S. Nezami, V. V. Albert, *et al.*, "Continuous symmetries and approximate quantum error correction," *Physical Review X*, vol. 10, no. 4, p. 041 018, 2020. DOI: 10.1103/PhysRevX.10.041018.

[41] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of physics*, vol. 303, no. 1, pp. 2–30, 2003. DOI: 10.1016/S0003-4916(02)00018-0.

[42] L. Gunderman, "Collective spin-cavity ensembles and the protection of higher-dimensional quantum information," 2022. [Online]. Available: http://hdl.handle.net/10012/18836.

[43] H. Watanabe, M. Cheng, and Y. Fuji, "Ground state degeneracy on torus in a family of zn toric code," *Journal of Mathematical Physics*, vol. 64, no. 5, 2023. DOI: https://doi.org/10.1063/5.0134010.

[44] M. Mathur and A. Rathor, "Su (n) toric code and non-abelian anyons," *Physical Review A*, vol. 105, no. 5, p. 052 423, 2022. DOI: 10.1103/PhysRevA.105.052423.

[45] C. Vuillot, A. Ciani, and B. M. Terhal, "Homological quantum rotor codes: Logical qubits from torsion," *arXiv preprint arXiv:2303.13723*, 2023. DOI: 10.48550/arXiv.2303.13723.