Gottesman-Kitaev-Preskill codes: A lattice perspective

Jonathan Conrad[1,2], Jens Eisert[1,2], and Francesco Arzani[1]

[1]Dahlem Center for Complex Quantum Systems, Physics Department, Freie Universität Berlin, Arnimallee 14, 14195 Berlin, Germany

[2]Helmholtz-Zentrum Berlin für Materialien und Energie, Hahn-Meitner-Platz 1, 14109 Berlin, Germany

February 9, 2022

We examine general Gottesman-Kitaev-Preskill (GKP) codes for continuous-variable quantum error correction, including concatenated GKP codes, through the lens of lattice theory, in order to better understand the structure of this class of stabilizer codes. We derive formal bounds on code parameters, show how different decoding strategies are precisely related, propose new ways to obtain GKP codes by means of glued lattices and the tensor product of lattices and point to natural resource savings that have remained hidden in recent approaches. We present general results that we illustrate through examples taken from different classes of codes, including scaled self-dual GKP codes and the concatenated surface-GKP code.

1 Introduction

It is known that quantum physics can be harnessed to speed-up certain computational tasks. However, it seems challenging to take advantage of such potential in practice. Arguably the biggest hindrance to achieving reliable quantum information processing is the difficulty to control systems and sufficiently isolate them from their environment. The field of *quantum error correction (QEC)* has been developed to cope with errors by choosing an appropriate encoding to constrain the action of environmental noise on the stored information and by repeated measurements of an associated stabilizer group to remove entropy from the system. With advancing experimental control over *superconducting* and *integrated optical photonic* circuits, recent years have witnessed growing interest in bosonic quantum error correction, where information is embedded into bosonic or continuous-variable (CV) systems exploiting the redundancy available in the associate infinite dimensional Hilbert space. In particular, so-called *Gottesman-Kitaev-Preskill (GKP)* codes [1] have been increasingly studied as promising components towards building a scalable quantum computer [2–6].

While the original proposal for the GKP code was made as early as in 2001, only in recent years experimental efforts of realizing such codes have gained momentum. Quantum error correction with single mode versions of the GKP code has recently been demonstrated experimentally in ion-traps [7] and superconducting hardware [8] while a growing number of theoretical proposals investigate the possibility to concatenate such codes with known qubit codes [9–12, 4] to achieve exponential error suppression by increasing the number of modes. Most works in this direction focus on understanding the single mode GKP code as an effective encoded qubit, with an effective noise channel, and proceed in their analysis using the well established toolbox of qubit quantum error correction. While it allowed rapid progress, this approach significantly limits the class of codes that can be considered and the analytical grasp on the continuous nature of the underlying system. In general, both the structure of the state and that of the stabilizer group of GKP codes can be understood in terms of lattices in phase-space, which is often used to introduce and visualize single-mode examples. But the deep connection with lattices also holds in the multi-mode case,

Jonathan Conrad: j.conrad1005@gmail.com

Francesco Arzani: frarzani@zedat.fu-berlin.de

including but, crucially, not limited to codes obtained from concatenating a single-mode GKP code with qubit stabilizer codes. The description of (multi-mode) GKP codes in terms of symplectic lattices was already formulated in the seminal work [1]. However, only a handful of examples can be found in the literature [12–15] that attempt to use it for a better understanding of the structure and properties of such codes, for constructing new codes or for devising decoding algorithms.

In this work we present an introduction to lattice theory for quantum error correction practitioners and show how the tools it provides can be fruitfully put to use for the analysis and design of GKP codes. The upshot is that, since lattices have long been studied in many branches of pure and applied mathematics, from number theory to string theory, they provide us with a rich toolbox whose potential remains largely untapped for quantum error correction. By elaborating on the connection with lattices, we move some further steps in this direction. We hope that this will motivate a wider adoption of lattice techniques in the study of GKP- and general quantum error correcting codes. The results in the present work suggest that this could lead to further insight as well as potential practical advantages towards fault-tolerant quantum computation with GKP codes.

To demonstrate the power of the lattice formalism, we derive a number of new results relating the parameters of GKP codes, such as the distance and encoding rate. We study the equivalence of codes under symplectic transformations, formulate the general decoding problem in terms of lattice quantities and provide two constructions leading to new codes that cannot be described (to the best of our knowledge) within the usual concatenated GKP-qubit paradigm. This highlights that the latter only captures very special cases, that, while easy to conceive, have no *a priori* reason to be considered optimal in taking advantage of the underlying bosonic nature of the system.

This article is structured as follows. In Section 2, we set the stage by introducing the relevant notions from lattice theory that will be used in the rest of this manuscript and we discuss their relation to GKP codes. Section 3 explores how the choice of the lattice basis corresponding to the stabilizer generators influences the properties of the latter. We focus in particular on suitable generalizations of *Calderbank-Shor-Steane (CSS)* codes and *low-density-parity-check (LDPC)* codes to the GKP setting. We show that significant savings in the number of stabilizer measurements with respect to the procedure usually employed for concatenated GKP-qubit codes are possible by measuring a minimal set of stabilizers at each error correction cycle. Furthermore, we show that concatenation of qubit stabilizer codes with single mode GKP codes provides a natural embedding of the qubit code into a lattice through the so-called *Construction A* [16], which leads to a no-go result that impacts the complexity of the decoding problem, in particular exact *minimum-energy-decoding*.

In Section 4, we prove a necessary and sufficient condition for two GKP codes to be equivalent up to a symplectic transformation, extending the single mode case mentioned in ref. [12]. This can be interpreted as code-switching by means of a Gaussian unitary. Section 5 is focused on defining a notion of *distance* for GKP codes via the length of the shortest non-trivial dual coset element of the lattice that describes the stabilizers. This definition is motivated by the structure of common noise models used for GKP codes in the literature. We discuss tradeoffs between the distance and other code parameters, such as encoding rate and the "length" of stabilizers. Furthermore, we show how to estimate the distance of a GKP code from the theta functions of the corresponding lattice, which are central quantities in lattice theory. Finally, we show that the distance of a CSS *qubit* stabilizer code is fully determined by the weight distribution of its stabilizer group, i.e., the number of stabilizers of particular Hamming weight in their representation by symplectic binary vectors. Although a similar statement can be derived from the seminal work of Shor, Laflamme [17] and Rains [18], it follows straightforwardly from the lattice theoretic concepts we introduce, showcasing that a lattice theoretic standpoint can also benefit the examination of qubit stabilizer codes.

Theta functions are central in the general formulation of the maximum likelihood (coset) decoding problem for general GKP codes, as we detail in Section 6. This hints at the existence of new decoding strategies that do not rely on the concatenated code structure, and are hence more general than those adapted from qubit codes. A detailed discussion of such strategies is beyond the scope of the present work and is therefore postponed to a separate publication.

Finally, we discuss in Section 7 two constructions, based on glued lattices and the lattice tensor product operation, respectively, to obtain GKP codes beyond scaled and concatenated codes, the two main code constructions featured in previous works and the rest of this work. We foresee that such alternative constructions may become of practical relevance in the near future, especially in setups with limited locality constraints, such as photonic architectures. Furthermore, we show that some properties of these codes can be computed easily, potentially making them also useful examples to probe the limits of GKP codes. Concluding remarks and future perspectives in Section 8 complete this work.

2 Preliminaries and notation

In this section, we review the basic setup of GKP codes as proposed in ref. [1] and introduce some basic concepts of lattice theory to analyze their structure. Good introductions to lattice theory are found in refs. [19, 16], from which many of the following definitions and lattice theoretic arguments presented here are adapted. A *stabilizer code* [20] embeds information into a physical Hilbert space by defining the code-space as fixed point of a finitely generated, abelian group $\mathcal{S}$, i.e., logical information is stored in terms of state vectors

$$|\psi\rangle: \; s\,|\psi\rangle = +1\,|\psi\rangle \; \forall s \in \mathcal{S}. \tag{1}$$

The *GKP code* [1] is a stabilizer code acting on the Hilbert space of n bosonic modes, where stabilizers are given by displacement operators

$$D\left(\boldsymbol{\xi}\right) = \prod_k \exp\left\{-i\sqrt{2\pi}u_k\hat{p}_k + i\sqrt{2\pi}v_k\hat{q}_k\right\} = \exp\left\{-i\sqrt{2\pi}\boldsymbol{\xi}^{\mathrm{T}}J\hat{\boldsymbol{x}}\right\}, \quad \boldsymbol{\xi} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix}, \quad \hat{\boldsymbol{x}} = \begin{pmatrix} \hat{\boldsymbol{q}} \\ \hat{\boldsymbol{p}} \end{pmatrix}, \tag{2}$$

where

$$J = J_{2n} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \otimes I_n = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix} \tag{3}$$

is the *symplectic form* and $\boldsymbol{\xi} \in \mathbb{R}^{2n}$ are phase-space vectors that specify the displacement. When no index is specified, we follow the convention $J = J_{2n}$ and $I = I_{2n}$. We also adopt the convention $\hbar = 1$, such that $[\hat{x}_k, \hat{x}_l] = iJ_{k,l}$ for $k, l = 1, \ldots, 2n$. Throughout this work, we use bold symbols to denote (column) vectors. *Displacement operators* implement a shift of the $2n-$quadratures

$$D^\dagger(\boldsymbol{\xi})\hat{\boldsymbol{x}}D(\boldsymbol{\xi}) = \hat{\boldsymbol{x}} + \sqrt{2\pi}\boldsymbol{\xi}, \tag{4}$$

commute as

$$D(\boldsymbol{\xi})D(\boldsymbol{\eta}) = e^{-i2\pi\boldsymbol{\xi}^T J\boldsymbol{\eta}}D(\boldsymbol{\eta})D(\boldsymbol{\xi}), \tag{5}$$

and satisfy

$$D(\boldsymbol{\xi})D(\boldsymbol{\eta}) = e^{-i\pi\boldsymbol{\xi}^T J\boldsymbol{\eta}}D\left(\boldsymbol{\xi} + \boldsymbol{\eta}\right). \tag{6}$$

Given the metaplectic representation of a symplectic operator [21] $S \mapsto U_S$, $S^TJS = J$, we have

$$U_S\hat{\boldsymbol{x}}U_S^\dagger = S\hat{\boldsymbol{x}}, \tag{7}$$

such that

$$U_{S^{-1}}D(\boldsymbol{\xi})U_{S^{-1}}^\dagger = \exp\left(-i\boldsymbol{\xi}^TJS^{-1}\hat{\boldsymbol{x}}\right) = \exp\left(-i\boldsymbol{\xi}^TS^TJSS^{-1}\hat{\boldsymbol{x}}\right) = D(S\boldsymbol{\xi}). \tag{8}$$

Displacement operators are orthogonal,

$$\mathrm{Tr}\left[D^\dagger\left(\boldsymbol{\xi}\right)D(\boldsymbol{\eta})\right] = \delta^{(2n)}\left(\boldsymbol{\xi} - \boldsymbol{\eta}\right), \tag{9}$$

and form a complete basis such that any operator $\hat{O} \in \mathcal{L}\left(\mathcal{H}\right)$ can be resolved as[1]

$$\hat{O} = \int d\boldsymbol{x} o(\boldsymbol{x})D\left(\boldsymbol{x}\right) = \int d\boldsymbol{x}\,\mathrm{Tr}\left[D^\dagger\left(\boldsymbol{x}\right)\hat{O}\right]D\left(\boldsymbol{x}\right). \tag{10}$$

We can now define the GKP stabilizer group.

[1] To be precise, this must be restricted to trace-class operators and one should formulate exceptions for operators such as $D\left(\beta\right)$ to satisfy eq. (9).

Definition 1 (GKP stabilizer group [1])**.** *The stabilizer group of a GKP code is given by a set of displacements*

$$\mathcal{S} := \langle D(\boldsymbol{\xi}_1), \ldots, D(\boldsymbol{\xi}_{2n})\rangle, \tag{11}$$

where $\{\boldsymbol{\xi}_i\}_{i=1}^{2n}$ *are linearly independent and we have* $\boldsymbol{\xi}_i^T J \boldsymbol{\xi}_j \in \mathbb{Z} \forall\, i,\, j$*.*

Exploiting the structure of the displacement operators, the GKP construction defines a stabilizer group isomorphic to a *lattice* with generator matrix

$$M = \begin{pmatrix} \boldsymbol{\xi}_1^T \\ \vdots \\ \boldsymbol{\xi}_{2n}^T \end{pmatrix}, \tag{12}$$

which is simply the set of integer linear combinations of basis elements

$$\mathcal{L} = \mathcal{L}(M) = \left\{\boldsymbol{\xi} \in \mathbb{R}^{2n} \middle|\, \boldsymbol{\xi}^T = \boldsymbol{a}^T M,\, \boldsymbol{a} \in \mathbb{Z}^{2n\times 2n}\right\}. \tag{13}$$

We follow the convention of ref. [1] such that the basis vectors of the lattice $\mathcal{L}$ constitute the *rows* of the generator matrix.

For $\mathcal{S}$ to constitute a stabilizer group, it needs to be (1.) a group and (2.) abelian. By construction in Definition 1, $\mathcal{S}$ is a group and by means of eq. (5) it can be observed that $\mathcal{S}$ is abelian if and only if the *symplectic Gram matrix* associated with *any* generator M of the lattice

$$A := MJM^T \tag{14}$$

has only *integer* entries. We then say that the lattice $\mathcal{L}(M)$ is *symplectically integral.* The symbol A will be reserved to the symplectic Gram matrix throughout the manuscript and although it depends on the particular generator M we will mostly leave this understood when the generator is clear from the context.

Using eq. (6), it can be verified that each element of the stabilizer group $\mathcal{S}$ can be written as

$$\prod_{i=1}^{2n} D(\boldsymbol{\xi}_i)^{a_i} = \prod_{i=1}^{2n} D(a_i \boldsymbol{\xi}_i) \tag{15}$$

$$= e^{i\pi \boldsymbol{a}^T A_{\text{◣}} \boldsymbol{a}} D\left((\boldsymbol{a}^T M)^T\right), \tag{16}$$

where $\boldsymbol{a} \in \mathbb{Z}^{2n}$ is an integer vector and $A_{\text{◣}} :\ (A_{\text{◣}})_{i,j} = A_{i,j} \,\forall\, i > j$ is the lower triangular matrix of A. The stabilizer group $\mathcal{S}$ is hence given by

$$\mathcal{S} = \left\{e^{i\phi_M(\boldsymbol{\xi})} D(\boldsymbol{\xi})\, | \boldsymbol{\xi} \in \mathcal{L}\right\}, \tag{17}$$

where

$$\phi_M(\boldsymbol{\xi}) = \pi \boldsymbol{a}^T A_{\text{◣}} \boldsymbol{a},\ \boldsymbol{a}^T = \boldsymbol{\xi}^T M^{-1} \tag{18}$$

denotes the phase-sector when we have chosen M to be the *pivot basis*, i.e. the set of basis vectors for which each associated displacement operator is fixed to eigenvalue +1 by definition 1. We will also use the notation $\mathcal{S} = \mathcal{S}(M, \phi_M)$ to specify GKP stabilizer groups from its generator M when the phase-sector is relevant. The pair (M, ϕ_N) specifies that the generator for the stabilizer group $D\left(M_i^T\right)$ is fixed by eigenvalue $e^{i\phi_N\left(M_i^T\right)}$ in code space. Consistently, we have $(M, \phi_M) = (M, 0)$.

When $\mathcal{L}$ is symplectically integral, we also have $e^{i\phi(\boldsymbol{\xi})} = \pm 1 \,\forall\, \boldsymbol{\xi} \in \mathcal{L}$, which is also already apparent from eq. (6). The additional phases in front of the displacement operators act trivially on the projective Hilbert space, such that they are irrelevant when considering stabilizer operators $\mathcal{S}$ as *symmetries* of the effective code. However, they play a role in error correction, where the goal is to fix each stabilizer $s \in \mathcal{S}$ to the +1 eigenvalue to define a code space and the phases determine to which ±1 eigenvalue the associated displacement operator needs to be fixed.

Remark (Definition of the GKP stabilizer group)**.** We remark that the definition of the GKP stabilizer group we make here differs from the one presented in ref. [1], which has been stated as $\mathcal{S}' = \{D(\boldsymbol{\xi})\,|\boldsymbol{\xi} \in \mathcal{L}\}$, where $\mathcal{L}$ is a symplectically integral lattice. Although this defines a commuting set of stabilizers, $\mathcal{S}'$ is in fact not closed in general due to the extra phases appearing when combining two displacement operators eq. (6) and would also not equal $\mathcal{S}$ as defined above. Definition 1 follows more closely the practical reality, where error correction is performed by measuring minimal generating set of the stabilizer group and by fixing (correcting) their eigenvalues to $+1$.

The definition $\mathcal{S}'$ is closed and identical to $\mathcal{S}$ in the special case where $A \in (2\mathbb{Z})^{2n\times 2n}$, i.e., the associated lattice $\mathcal{L}$ is *symplectically even*. We found, however, that this would restrict us too severely in the classes of codes to analyze (e.g. the concatenated GKP - surface code introduced later is described by a symplectically integral, but not symplectically even lattice). We illustrate the difference between the sets $\mathcal{S}$ and $\mathcal{S}'$ using the stabilizer group for the sensor state [22],

$$\mathcal{S}_{\text{sensor}} = \langle D\left(\boldsymbol{e}_1\right), D\left(\boldsymbol{e}_2\right)\rangle = \left\{(-1)^{mn} D\left(m\boldsymbol{e}_1 + n\boldsymbol{e}_2\right),\ (m,n) \in \mathbb{Z}^2\right\}, \tag{19}$$

where $\boldsymbol{e}_j$ are canonical basis vectors, so that code-states should have eigenvalue -1 on displacement for which mn is odd if they have eigenvalue $+1$ on each generator $D\left(\boldsymbol{e}_1\right), D\left(\boldsymbol{e}_2\right)$. Given the conventional definition $\mathcal{S}'$, $D\left(\boldsymbol{e}_1\right) D\left(\boldsymbol{e}_2\right) = (-1) D\left(\boldsymbol{e}_1 + \boldsymbol{e}_2\right)$ would formally not be included in the stabilizer group.

A system of n bosonic modes can be associated with an infinite dimensional Hilbert space and a $2n-$dimensional phase space, whose axes correspond to the quadratures of the system, each corresponding to an observable taking continuous values. To encode discrete quantum information, such as a qubit, the state space needs to be fully "discretized" by introducing suitable constraints. This is done by choosing a generating set for the stabilizer group with $2n$ linearly independent generators. Each linearly independent generator can be seen as "discretizing" one direction in phase space. This is the reason why we require the lattice $\mathcal{L}$ to be *full rank*, that is, we demand that M has full row-rank. It is always possible to specify a lattice in $\mathbb{R}^{2n}$ using more than $2n$ basis vectors, (we will see an example of such a scenario later), but they can always be reduced to $2n$ linearly independent vectors, a process for which a number of (efficient) algorithms are known [19]. We note that also non-full rank (or *degenerate*) lattices had recently been explored to define effective GKP codes [23, 15]. The code space of such codes retains undiscretized quadratures which allows to encode and perform error correction on general CV states. In this work we focus on encoding qubits or qudits defined via full-rank lattices while encoding of continuous information using degenerate lattices is briefly explained in appendix B.

Given a lattice $\mathcal{L}$, the lattice basis is not unique. Rather, two generator matrices M, M' generate the same lattice $\mathcal{L}(M) = \mathcal{L}(M')$ if and only if there exists a unimodular matrix $U \in GL\left(2n, \mathbb{Z}\right), |\det(U)| = 1$ such that

$$M' = UM. \tag{20}$$

Such transformation also transforms

$$A \mapsto A' = UAU^T. \tag{21}$$

Note that A' has even entries if and only if A does. Due to the phases appearing in eq. (17), when a different basis M' is used to fix the stabilizer group as in Definition 1, the generating set for the stabilizer group needs to be chosen as

$$(M',\, \phi_M) \tag{22}$$

to yield the same stabilizer group and, effectively, the same code and fix the same code-space. The symplectic Gram matrix is invariant under a symplectic transformation

$$M \mapsto MS^T = \begin{pmatrix} (S\boldsymbol{\xi}_1)^T \\ \vdots \\ (S\boldsymbol{\xi}_{2n})^T \end{pmatrix}. \tag{23}$$

Note that, unlike the change of basis, a symplectic transformation generally changes the lattice but also leaves the symplectic Gram matrix invariant. As a result, symplectic transformations can serve as useful tool to adapt the code to the properties of the noise while keeping the same encoding rate, as we will see in the following. When a GKP code is specified by its generator $\mathcal{S} = \mathcal{S}(M, 0)$, the corresponding phase sector transforms under symplectic transformation as

$$\phi_M \mapsto \phi_{MS^T}. \tag{24}$$

Any basis M of the full rank lattice $\mathcal{L}$ partitions $\mathbb{R}^{2n} = \mathbb{R}^{2n}M$ into lattice translates of the *fundamental parallelepiped* (note the row-vector convention adopted here),

$$\mathcal{P}(M) := \left\{\boldsymbol{x}^T M \,\middle|\, \boldsymbol{x} \in [0,1)^{2n}\right\} \tag{25}$$

such that every point $\boldsymbol{x}^T \in \mathbb{R}^{2n}$ is uniquely associated to a lattice point $\boldsymbol{x}^T \in \boldsymbol{a}^T M + \mathcal{P}(M)$. Equivalently, one can choose to partition $\mathbb{R}^{2n}$ using the *centered fundamental parallelepiped*

$$\mathcal{C}(M) := \left\{\boldsymbol{x}^T M \middle| \, \boldsymbol{x} \in \left[-\frac{1}{2}, \frac{1}{2}\right)^{2n}\right\}. \tag{26}$$

Any set that exactly partitions the space under lattice translations is called *fundamental* and has volume

$$\det(\mathcal{L}) := |\det(M)| = |\sqrt{\det(A)}| = |\sqrt{\det(G)}| \tag{27}$$

where $G := MM^T$ is the *euclidean* Gram matrix associated to the generator M. The determinant $\det(G)$ is a lattice invariant known as its *discriminant.* The determinant (discriminant) can be geometrically interpreted as measure of inverse density of lattice points.

A *sublattice* $\mathcal{L}' \subseteq \mathcal{L}$ is a subset of $\mathcal{L}(M)$ that is itself a lattice. A basis for any $d-$dimensional sublattice of $\mathcal{L}(M)$, where $d < 2n$, can be given by

$$S = BM, \tag{28}$$

where $B \in \mathbb{Z}^{d\times 2n}$ is an integer matrix of full row rank. The *symplectic dual* (simply "dual" in the following) of a lattice $\mathcal{L}$ is the lattice $\mathcal{L}^\perp$ that consists of all vectors that have integer symplectic inner product with any vector from $\mathcal{L}$

$$\mathcal{L}^\perp = \left\{\boldsymbol{\xi}^\perp \in \mathbb{R}^{2n} \mid \left(\boldsymbol{\xi}^\perp\right)^T J\boldsymbol{\xi} \in \mathbb{Z}\ \forall \boldsymbol{\xi} \in \mathcal{L}\right\}. \tag{29}$$

Within the GKP code construction vectors $\boldsymbol{\xi}^\perp \in \mathcal{L}^\perp$ correspond to the displacement amplitudes that are associated to the centralizer of the stabilizers within the set of displacement operators. A canonical choice of basis $M^\perp$ for the symplectic dual is specified by fixing $M^\perp$ to satisfy

$$M^\perp J M^T = I. \tag{30}$$

Since we will focus on full rank lattices, for which M is non-singular, we obtain the canonical dual basis as

$$M^\perp = (JM^T)^{-1} = M^{-T}J^T. \tag{31}$$

Together with the definition of A it can be shown that

$$M = AM^\perp, \tag{32}$$

that is, the symplectic Gram-matrix A describes how the sublattice $\mathcal{L} = \mathcal{L}(M) \subseteq \mathcal{L}^\perp = \mathcal{L}\left(M^\perp\right)$ associated to stabilizer operators embeds into the lattice associated to its centralizer. The *dual quotient* (or *glue group*) of $\mathcal{L}$, $\mathcal{L}^\perp/\mathcal{L}$ thus lists the logically distinct displacements admitted by the GKP code $\mathcal{S}$. Since we are interested in effective qubit encodings with logical dimension $d = 2^k$, these displacements form the effective logical Pauli group.

Finally, the number of logically distinct centralizer elements associated to $\mathcal{S}$ is

$$d^2 = |\mathcal{L}^\perp/\mathcal{L}| = |\det(M)|/|\det\left(M^\perp\right)| = |\det A| = |\det(M)|^2. \tag{33}$$

One can verify this formula geometrically by imagining the partition of $\mathcal{P}(M)$ with patches $\mathcal{P}\left(M^\perp\right)$. Under basis transformation of the direct lattice $M \mapsto UM$ the canonical dual basis transforms as

$$M^\perp \mapsto U^{-T}M^\perp, \tag{34}$$

and under symplectic transformations $M \mapsto MS^T$ we obtain

$$M^{\perp} \mapsto M^{\perp} S^T = \begin{pmatrix} (S\boldsymbol{\xi}^{\perp})_1^T \\ \vdots \\ (S\boldsymbol{\xi}_{2n}^{\perp})^T \end{pmatrix}. \tag{35}$$

The symplectic Gram matrix of the dual lattice can be shown to satisfy

$$A^{\perp} := M^{\perp} J \left(M^{\perp}\right)^T = A^{-1} = \frac{1}{|\det(A)|} \operatorname{adj}(A), \tag{36}$$

where $\operatorname{adj}(A)$ is the *adjugate* of A. It is similarly common to define the *euclidean* dual of a lattice $\mathcal{L}$ which we shall denote by $\mathcal{L}^*$. This is the lattice in $\mathbb{R}^{2n}$ consisting of all vectors with integer *euclidean* inner product with every vector in $\mathcal{L}$.

The euclidean dual is more common than the symplectic one in the lattice theory literature, and hence often simply called dual. Similar to the symplectic case, a canonical basis M^* for the dual lattice can be fixed to satisfy

$$M^* M^T = I \tag{37}$$

or equivalently (since we only deal with full rank lattices)

$$M^* = M^{-T}. \tag{38}$$

Since $J \in O(2n)$ is an orthogonal matrix, we can observe that the symplectic dual is equivalent to the euclidean dual up to an orthogonal rotation. In particular, the distribution of lengths of vectors in $\mathcal{L}^{\perp}$ is equal to the distribution of lengths of vectors in $\mathcal{L}^*$. We will take advantage of this fact when analyzing the code distance of GKP codes in Section 5.

We close this section by introducing the two most studied classes of GKP codes, which we will use for illustrations throughout this manuscript. The first class, which we will refer to as *scaled codes*, have been thoroughly examined in ref. [13] and build on symplectic *self-dual* lattices. A symplectic self-dual lattice is a symplectically integral lattice for which $|\det(M)| = |\det(A)| = 1$ and, consequently, $\mathcal{L} = \mathcal{L}^{\perp}$. The associated code-space is one-dimensional.

Scaled GKP codes. A *scaled GKP code* $\mathcal{L}(M)$ is obtained by rescaling a symplectically self-dual lattice $\mathcal{L}_0 = \mathcal{L}(M_0) \subset \mathbb{R}^{2n}$ as

$$M = \sqrt{\lambda} M_0, \ \lambda \in 2\mathbb{Z}. \tag{39}$$

The associated symplectic Gram matrix becomes $A = \lambda A_0$ and the dimension of the code-space is $d = |\sqrt{\det(\lambda A_0)}| = \lambda^n$. By e.g. choosing $\lambda = 2$ this yields a code with $k = n \log_2(\lambda) = n$. Let

$$\lambda_1(\mathcal{L}_0) := \min\{\|\boldsymbol{x}\|, \boldsymbol{x} \in \mathcal{L}_0\} \tag{40}$$

be the length of the shortest vector in $\mathcal{L}_0 = \mathcal{L}_0^{\perp}$. Note that throughout this manuscript $\|\cdot\| = \|\cdot\|_2$ denotes the euclidean *2-norm*. The rescaling implies $\lambda_1(\mathcal{L}) = \lambda^{\frac{1}{2}} \lambda_1(\mathcal{L}_0)$ and $\lambda_1\left(\mathcal{L}^{\perp}\right) = \lambda^{-\frac{1}{2}} \lambda_1(\mathcal{L}_0)$. For $\lambda > 1$ the symplectic dual vector corresponding to $\lambda_1\left(\mathcal{L}^{\perp}\right)$ cannot be in $\mathcal{L}$ and hence constitutes the shortest logically non-trivial displacement, the length of which decreases with the number of logical dimensions that are squeezed into the code. Prominent examples of scaled GKP codes that have already been discussed in ref. [1] are generated by

$$M_{\square} = \sqrt{2} I_2, \tag{41}$$

also known as the *square GKP code*, that encodes $k = 1$ qubit into $n = 1$ oscillator and

$$M_{\hexagon} = 3^{-\frac{1}{4}} \begin{pmatrix} 2 & 0 \\ 1 & \sqrt{3} \end{pmatrix}, \tag{42}$$

which is known as the *hexagonal GKP code*, similarly with $k = n = 1$ but with a different lattice geometry that requires a slightly larger displacement amplitude to implement a logical operator. Here we have $\lambda_1\left(\mathcal{L}^{\perp}\right) = 3^{-\frac{1}{4}}$ compared to $\lambda_1\left(\mathcal{L}^{\perp}\right) = 2^{-\frac{1}{2}}$ for the square GKP code. Further symplectic self-dual lattices with larger $\lambda_1(\mathcal{L}_0)$ and a numerical procedure to find symplectic self-dual lattices are detailed in ref. [13].

Concatenated square GKP codes. The second class that will be extensively discussed throughout this work is that of concatenated codes built from the square GKP code and a qubit quantum-error correcting code. Let $Q \subset \mathbb{Z}_2^{2n}$ be a set of binary symplectic vectors[2] such that $\mathcal{S}_Q = \left\{ Z_1^{q_1} Z_2^{q_2} \dots X_{n+1}^{q_{n+1}} \dots X_{2n}^{q_{2n}} \middle| \forall \boldsymbol{q} \in Q \right\}$ describes the stabilizer group of a $[\![n,k,d]\!]$ qubit quantum error correcting code. We can embed binary vectors in $\mathbb{R}^{2n}$ in the trivial way. The lattice associated to the concatenated GKP code will be given by

$$\mathcal{L} = \Lambda(Q) := \left\{ \boldsymbol{x} \in \mathbb{R}^{2n} \middle| \sqrt{2}\boldsymbol{x} \mod 2 \in Q \right\} \tag{43}$$

and equally describes an encoding of k logical qubits into n modes. Such lattices are known in the literature as *Construction A* lattices [16]. Given $r = n - k$ symplectic vectors associated to a set of generators for $\mathcal{S}_Q$ that we stack row-wise into a generator $B_Q \in \mathbb{Z}_2^{r \times 2n}$, we can write down the generator for the concatenated code

$$M_{\text{conc}} = \frac{1}{\sqrt{2}} \begin{pmatrix} B_Q \\ 2I_{2n} \end{pmatrix}. \tag{44}$$

This basis is overcomplete and can be row-reduced to only consist of $2n$ generators. We discuss basis choices in the next section.

3 Bases of GKP codes

The $2n$ rows of the generator matrix M of a GKP code stand in one-to-one correspondence with a minimal number of generators for the stabilizer group which are measured to deduce the error syndrome for recovery. Hence the choice of basis M strongly influences the physical implementation of the code and certain basis choices may be beneficial over others. Natural desiderata include the following properties.

CSS property. A *CSS code* is known in qubit quantum error correction as a code for which the set of stabilizer operators disjointly separates into one that only contains Pauli-Z type operators and one that only contains Pauli-X type operators (or any other pairing). In practice, this has the advantage that ancillary qubits used to measure the stabilizers only need to come in two different initial states and only need to couple with their associated data qubits in one particular way, i.e., via a $CNOT$ or CZ operations. Furthermore, for sufficiently simple error models, decoding algorithms only need to be tailored to the respective subcodes. The most studied quantum error correcting codes are indeed CSS codes, such as homological codes [24] including the celebrated *toric code* [25] or planar versions of it [26], which at the time of writing yield the best promise of protection from noise and allow for geometrically local implementations. A natural generalization to GKP codes is to call *CSS* a code whose lattice admits a basis that can be written as [1]

$$M = M_q \oplus M_p, \tag{45}$$

where $M_q, M_p \in \mathbb{R}^{n \times n}$ describe stabilizers that contain only $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ quadrature operators, respectively. Equivalentely, the corresponding lattice can be decomposed into the direct sum of two disjoint sublattices

$$\mathcal{L} = \mathcal{L}_q \oplus \mathcal{L}_p. \tag{46}$$

A trivial example of GKP-CSS code is the single-mode square GKP code. Concatenating a CSS qubit code with the square GKP code trivially preserves the CSS property. The decomposition eq. (45) holds concretely for the Hermite normal form [27, 28] of a lattice basis when M is (or can be rescaled to) integer, which is defined as follows.

Definition 2 (Hermite normal form (HNF))**.** *Let M be the basis for a lattice as described in the main text. M is in Hermite normal form if*

[2] That is $q^T J q' = 0 \mod 2$ for all $q,\ q' \in Q$ (considering addition over reals or in $\mathbb{Z}_2$ does not make any difference).

1. *M is upper triangular and of positive diagonal, and*
2. $0 \leq M_{i,j} \leq M_{j,j}\ \forall i, j$ *, i.e., the pivot of each row is the largest element of its column.*

The Hermite normal form is special in that, for integer M, the HNF is unique [29] and can be computed efficiently (see also ref. [19]). It can hence be used to verify the CSS property or compare whether two stabilizer groups are equivalent whenever we are given bases that can be scaled to be integer (and scaled back after applying the preferred HNF algorithm). In particular, this applies to concatenated square GKP codes such that we can always check whether a given code is CSS by computing its HNF.

Short basis vectors. Besides being CSS, the homological qubit codes mentioned above are *low-density-parity-check (LDPC)* codes, meaning that they admit a choice of generators such that the support of each generator is bounded and each qubit only interacts with a bounded number of generators. Here "low" typically means constant in the number of physical qubits. The LDPC property can be generalized in two different ways. Analogous to the qubit picture we can ask for the stabilizers of the GKP code to only act non-trivially on a low number of modes and that each mode be only touched upon by a low number of stabilizers. It is straightforward to see that this property is inherited whenever a qubit LDPC code is concatenated with single mode bosonic codes such as the square- or hexagonal-GKP code and can more generally be quantified by counting the number of non-zero elements of a choice of generator M per row/column. On the other hand, in the bosonic case, it would also be natural to demand for the lattice basis vectors to be *short* in the sense of the euclidean norm. If one imagines that the stabilizer checks are implemented by coupling in and subsequently measuring ancillary systems, the length of the basis vectors depends both on the number of modes that each ancilla is coupled to as well as on the interaction-strength/-time required. We will see shortly how a minimal basis (that is, consisting of only $2n$ generators) with short vectors can be constructed for concatenated GKP codes. In general, when no extra structure is specified, short bases can also be computed by means of *lattice reduction* algorithms, such as the *Lenstra-Lenstra-Lovász* (LLL) algorithm [30], which efficiently finds a short and almost orthogonal basis.

Orthogonality. On top of shortening the basis vectors, the LLL algorithm also attempts to find a basis that is *nearly* orthogonal. A basis M is orthogonal when MM^T is diagonal and we say a lattice $\mathcal{L}$ is orthogonal if such a basis exists.

We will see later in Sections 5, 6 how orthogonality is useful when attempting to quantify the distance of a GKP code or for the task of decoding. The single-mode square and rectangular GKP codes correspond to orthogonal lattices. However, when a qubit code is concatenated to the square-GKP code, leading to Construction A lattices, this can only be the case if the qubit stabilizer generators act disjointly on at most two quadratures at once [31], i.e., if the qubit stabilizer code decomposes as

$$Q = \bigoplus_i Q_i, \qquad Q_i = \{0, 1\} \text{ or } Q_i = \{00, 11\} \tag{47}$$

up to permutation of coordinates. Since stabilizer codes require stabilizer generators to overlap non-trivially to achieve long-range entangled states, we arrive at the following claim.

Claim 1 (Orthogonality of concatenated GKP codes)**.** *Concatenated GKP codes with long-range entangled code states have a non-orthogonal lattice $\mathcal{L}$.*

Phrased differently, this means that codes associated to orthogonal lattices cannot produce long-range entanglement. Finally, we can also relate the length of stabilizer generators to the possibly admitted encoding rate.

Theorem 1 (Hadamard's bound)**.** *Let M be a GKP code's generator, $C = \max_i \|M_i\|$ with M_i the ith row of M, and $d = 2^k = \sqrt{\det(A)}$ the logical dimension. We have*

$$k \leq 2n \log_2 C. \tag{48}$$

Proof. We have $\det(M^\perp) = \det(M)^{-1}$, thus by $M = AM^\perp$ we have $|\det(A)| = |\det(M)|/|\det(M^\perp)| = \det(M)^2$. Hadamard's inequality implies

$$|\det(M)| \leq \prod_{i=1}^{2n} \|M_i\| \leq C^{2n}, \tag{49}$$

which is also known as Hadamard's bound. Taking the logarithm on both sides yields the result. □

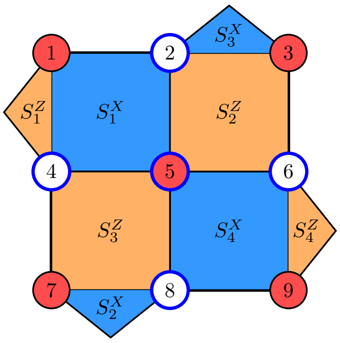
This simple inequality based on Hadamard's bound constrains the encoding rate of a GKP code by the maximum length of the stabilizers. For a fixed basis M, C quantifies the size of the displacement that needs to be measured to extract the syndrome, which can loosely be interpreted as the "experimental hardness" to implement a code with stabilizer generators described by the rows of M. C is large whenever many modes are involved in a single stabilizer generator (compare with the LDPC property) or the displacement to be measured is large (requiring stronger coupling). We note that the value of C is basis-dependent, while the LHS of the inequality is not. Hence any fixed basis M yields an upper bound on the encoding rate but the bound will only be saturated when all M_i are orthogonal and of similar length. We have already observed in Claim 1 that interesting concatenated codes do not yield orthogonal lattices, such that in these cases the bound will not hold tightly.

3.1 Minimal bases for concatenated GKP codes

As an example of an interesting application of lattice basis reduction, we present a procedure to find a minimal basis for the lattice obtained as concatenation of the single square-GKP code with the *Surface-17 code* in Fig. 1. This basis is derived by first noting that $2n$ stabilizers suffice to fully specify the code, in contrast to the usual $3n - k$ generators of the union individual GKP stabilizers and concatenated stabilizers. Starting from this set, it is clear that removing any of the $r = n - k$ independent qubit stabilizers from the union of generators necessarily changes the dimension of the code space. The redundant stabilizers, that can be omitted without affecting the encoded subspace, must hence be found among the single-mode GKP ones. We identify the redundant single-mode GKP stabilizers as follows. Squaring the concatenated qubit code stabilizers $S_i^{Z/X}$ yields a product of square GKP-code stabilizers, such that not all GKP stabilizer overlapping on $\operatorname{supp}\left(S_i^{Z/X}\right)$ need to be included on that local region to generate the full local stabilizer group, e.g. in Fig. 1 one can obtain S_2^p by multiplying $(S_3^X)^2$ with S_3^p, such that locally S_3^p and S_3^X suffice to reconstruct the full $p-$type syndrome. We note that the pattern of the included local GKP-code stabilizers depicted in Fig. 1 can be extended to larger surface code layouts.

In general, given a concatenated square GKP code we obtain a procedure to find a minimal generating set as detailed in Figs. 2, 3, 4. Let $B = B_Q \in \mathbb{Z}_2^{r\times 2n}$ be a minimal generator matrix for the qubit code. We iterate through all GKP-stabilizer generators and check whether they are already member of the lattice generated by B. If not, we append this generator to the basis and proceed to the next. This algorithm will terminate once $2n - r$ GKP stabilizer generators are added. Importantly, it preserves the qubit stabilizers and hence the LDPC property. If the qubit code had a generating set with maximal weight w, the longest basis vector will be of length $\max\{\sqrt{w/2}, \sqrt{2}\}$. Note that the membership test outlined in Fig. 4, that applies when dealing with rational lattice (in which case the lattice can be rescaled to integer, such that the HNF is unique), can be executed in polynomial time. The above procedure can trivially be generalized to the concatenation of a qubit code with a different single-mode GKP code. In particular if we use single mode GKP codes that are locally described by generator matrix M_{GKP}, the algorithm outlined in Fig. 2 is adapted by iterating through the rows of $M_{\text{GKP}}^{\oplus n}$ and matching the non-trivial entries in B_Q to elements in $\left(\mathcal{L}\left(M_{\text{GKP}}\right)^\perp\right)^{\oplus n}$. A *python* implementation of the algorithm can be found under ref. [32].

We conclude this section by noting a practical advantage following from the existence of minimal bases. In the literature, an error correction cycle for concatenated GKP codes typically contains two steps: first all single-mode GKP stabilizers are measured and the state is corrected to the corresponding sublattice. Then all qubit level stabilizers are measured and the syndrome is fed



$$
\begin{aligned}
S_1^X &= X_1^\dagger X_2 X_4 X_5^\dagger \\
S_2^X &= X_7^\dagger X_8 \\
S_3^X &= X_2 X_3^\dagger \\
S_4^X &= X_5^\dagger X_6 X_8 X_9^\dagger \\
S_1^Z &= Z_1 Z_4 \\
S_2^Z &= Z_2 Z_3 Z_5 Z_6 \\
S_3^Z &= Z_5 Z_5 Z_7 Z_8 \\
S_4^Z &= Z_6 Z_9
\end{aligned}
$$

Figure 1: The Surface-17 code with a minimal generating set. Multi qubit stabilizer generators are indicated in the figure. The minimal number of additional square-GKP stabilizer are given by $S_i^p = e^{-2i\sqrt{\pi}\hat{p}_i}$ on the modes labeled by red filled circles and $S_i^q = e^{i2\sqrt{\pi}\hat{q}_i}$ on the vertices drawn in blue. It suffices to measure only one type of square GKP generator on most modes, except for mode 5, where both S_5^q and S_5^p need to be measured. The $\cdot^\dagger$'s are chosen such that all qubit stabilizers commute on the full bosonic Hilbert space [10], i.e. such that $B_Q J B_Q^T = 0$. [3]We follow the convention for the stabilizers in ref. [11].

Figure 2: Greedy lattice completion algorithm

```
function LatticeCompletion($B_Q$)
    set $B = B_Q$
    for $i = 1, \dots, 2n$ do
        if not Member$\left(2\boldsymbol{e}_i^T; B\right)$ then
            $B \leftarrow \begin{pmatrix} B \\ 2\boldsymbol{e}_i^T \end{pmatrix}$
        end if
    end for
    Return $\frac{1}{\sqrt{2}} B$
end function
```

Figure 3: Membership test

```
function Member($\boldsymbol{x}$; $B$)
    Return $\exists \boldsymbol{x} \in \mathbb{Z}^{2n} : \boldsymbol{x}^T B = 2\boldsymbol{e}_i^T$
end function
```

Figure 4: Membership test (alternative)

```
function Member($\boldsymbol{x}$; $B$)
    Return HNF$\begin{pmatrix} B \\ 2\boldsymbol{e}_i^T \end{pmatrix}$ = HNF$(B)$
end function
```

to a modified qubit decoder, possibly taking into account the analogue measurement information. The existence of a minimal set of $2n$ lattice basis vectors implies that checking whether a given state is a code-state can be done with only $2n$ measurements. This has to be contrasted with the usual $2n + n - k$ measurements in typical two-step schemes. Using a minimal basis foregoes measuring the GKP stabilizers whose eigenvalues are already fixed by the results on the minimal set, and can thus lead to a reduction of about $1/3$ in the number of (GKP) ancillas required for an error correction cycle. Although the eigenvalues of the redundant, unmeasured, GKP stabilizers can be deduced from the other measurements, in the presence of measurement errors other strategies might be more effective.

4 Symplectic equivalence

By definiton of the symplectic Gram matrix, it remains invariant under symplectic transformations of the GKP code $M \mapsto N = MS^T$. Symplectic transformations are physically implemented by Gaussian unitaries which have a special role in bosonic quantum information processing, for a number of reasons, including the fact that they are often the easiest to implement in the laboratory. As far as error correction is concerned, the fact that they act linearly on the quadratures implies that potential errors, such as small displacements on the modes, remain bounded in their amplitude. By construction of the GKP code, the logical Clifford group can be implemented using only

[3] For homological codes derived from *Kitaev's quantum double* models [33], these can be identified by considering local groups $\mathbb{R}$ and looking at the direction of the arrows.

symplectic transformations.[4] Here, we show that two GKP codes are equivalent under symplectic transformations if and only if their standard form (introduced below) is equal and show that every code is symplectically equivalent to a standard form where each stabilizer acts on a single mode. This is a generalization of the symplectic equivalence of the square- and hexagonal GKP code noticed in ref. [12]. We say a symplectic Gram matrix is in standard form if

$$A = J_2 \otimes D, \tag{50}$$

where D is a positive diagonal $n \times n$ integer matrix with non-increasing diagonal elements. A symplectically integral matrix A can always be brought into standard form via basis transformation [1].

Theorem 2 (Symplectic equivalence)**.** *Let $M, N \in \mathbb{R}^{2n\times 2n}$ be generators of full rank lattices $\mathcal{L}(M), \mathcal{L}(N) \subset \mathbb{R}^{2n}$. Without loss of generality, assume the bases M, N are chosen such that $A_M = MJM^T = J_2 \otimes D_M$ and $A_N = NJN^T = J_2 \otimes D_N$ are in standard form. Then $M = NR^T$ for some symplectic matrix R if and only if $D_N = D_M =: D$.*

Proof. From the definition of A it is clear that $M = NS^T$ for some symplectic $S : S^T JS = J$ implies that $A_N = A_M$ or equivalently $D_N = D_M$. To prove the converse, let

$$Z := I_2 \otimes D^{-\frac{1}{2}}, \qquad \tilde{M} := ZM, \qquad \tilde{N} := ZN. \tag{51}$$

We have

$$\tilde{M}J\tilde{M}^T = \tilde{N}J\tilde{N}^T = J, \tag{52}$$

i.e., both $\tilde{M}, \tilde{N}$ are symplectic. Since symplectic matrices are invertible, we define $R^T := \tilde{N}^{-1}\tilde{M}$ such that $\tilde{M} = \tilde{N}R^T$. R is symplectic since the symplectic group is closed under transpose and inverse. Left-multiplying by Z^{-1} we obtain $M = NR^T$. □

It is important to note that since the symplectic equivalence above is established using the bases M, N chosen such that A_M, A_N are in standard form, one needs to be careful in accounting for possible phases in front of the associated stabilizer operators stemming from the basis transformation. When the above assumptions of the theorem above are satisfied we obtain that the stabilizer group $\mathcal{S}(N, 0)$ can, by symplectic transformation R^T, be transformed into $\mathcal{S}(M, \phi_{NR^T})$. I.e. code states $|\psi\rangle$ of the code $\mathcal{S}(N, 0)$ can be transformed into states $U_{R^{-T}} |\psi\rangle$ by means of a Gaussian unitary (see eq. (8)) which are stabilized by phased displacement operators

$$e^{i\phi_{NR^T}\left(M_i^T\right)} D\left(M_i^T\right), \, i = 1, \ldots, 2n. \tag{53}$$

Note that phases above are absent in the symplectic transformation between a single-mode square GKP code and hexagonal GKP code shown in ref. [12], which is because each of these codes have symplectically even lattices, such that the overall phase factors are trivial. The same discussion on phase-assignement is relevant for the application of the following immediate corollaries.

Corollary 1 (Normal form of generators)**.** *Let $M \in \mathbb{R}^{2n\times 2n}$ describe the basis of a full rank lattice with integral $A = MJM^T$. W.l.o.g. assume M is chosen such that $A = J_2 \otimes D$ is in standard form. The code specified by the generator*

$$N_\square = \oplus_{j=1}^n \sqrt{D_{jj}} I_2 \tag{54}$$

is symplectically equivalent to the one specified by M.

Note that here, we understand the direct sum as happening mode-wise, such that it corresponds to the usual matrix direct sum if the quadratures are ordered as $(\hat{q}_1, \hat{p}_1, \hat{q}_2, \hat{p}_2, \ldots)$ with symplectic form $J'_{2n} = I_n \otimes J_2$ but not if they are ordered as $(\hat{q}_1, \hat{q}_2, \ldots, \hat{p}_1, \hat{p}_2, \ldots)$ in the standard choice $J_{2n} = J_2 \otimes I_n$. This convention will be understood in the following whenever clear from the context. Note that $N_\square$ is diagonal, so its stabilizers act each on a single quadrature at a time. Since $N_\square$

[4] However, not every symplectic transformation necessarily corresponds to a Clifford operation on the logical level.

decomposes into a direct sum over each mode, we can always prepare a code state of M by locally preparing a code state of $N_{\square}$ and applying the corresponding symplectic transformation.

The same symplectically equivalent local decomposition can equally be done using the hexagonal GKP code on each local mode and is based on the fact that the square and hexagonal GKP code, as scaled GKP codes with scaling factor λ, have both $A_{\square} = A_{\hexagon} = \lambda J_2$. Similarly, any other scaled GKP codes can also be used. This is an example of *code switching*. Even more generally, this implies code switching between two GKP codes $\mathcal{L}\left(M\right)$ and $\mathcal{L}\left(N\right)$ on the same number of modes is possible whenever $D_N = D_M$ in their respective standard bases. Similarly, we can also deduce the following corollary.

Corollary 2 (Normal form in prime dimensions)**.** *For $d = \left|\det(M)\right|$ prime, the lattice $\mathcal{L} \subset \mathbb{R}^{2n}$ is symplectically equivalent to a code specified by*

$$N' = \sqrt{d} I_2 \oplus I_2^{\oplus(n-1)}, \tag{55}$$

Proof. For a basis of $\mathcal{L}$ in standard form, we have

$$d = \prod_{j=1}^{n} D_{j,j}. \tag{56}$$

Since D is a positive integer diagonal matrix, if d is prime, the eigenvalues of D are uniquely specified by $D = \mathrm{diag}(d, 1, 1, \ldots)$ up to permutations. Hence by Theorem 2 we have that $\mathcal{L}$ is symplectically equivalent to the code specified by eq. (55). □

From the proof above one can also see that the number of symplectically inequivalent classes of codes with given logical dimension d corresponds to the number of different factorizations of d. For example, for two modes encoding two qubits there are two inequivalent choices: $D = \mathrm{diag}\left(4, 1\right)$ or $D = \mathrm{diag}\left(2, 2\right)$, corresponding to two symplectically inequivalent classes of codes.

5 Distance of a GKP code

We need a benchmark to assess how well the logical content of a GKP state is protected from noise. In qubit codes, a simple standard assumption is that noise is stochastic i.i.d. for each physical qubit, such that the likelihood of an error decreases exponentially with its support or weight. This assumption makes the definition of a code distance as weight of the shortest non-trivial logical operator meaningful. This model no longer makes sense in the bosonic setting. In the bosonic setting, a reasonable assumption about the underlying error model is that weak coupling to the environment results in effectively small displacements. Concrete examples are loss, thermal noise [34], and to a limited extent, finite squeezing errors [35, 36, 2]. For simplicity, we assume a stochastic displacement noise model such that small displacements are more likely than large displacements, this is in particular the case for the Gaussian displacement noise model

$$\mathcal{N}\left(\rho\right) = \int d\boldsymbol{x}\; P_{\tilde{\sigma}}\left(\boldsymbol{x}\right) D\left(\boldsymbol{x}\right) \rho D^{\dagger}\left(\boldsymbol{x}\right), \tag{57}$$

which we will analyse more in-depth later, where

$$P_{\tilde{\sigma}}\left(\boldsymbol{x}\right) = G\left[0, \tilde{\sigma}^2 I_{2n}\right]\left(\boldsymbol{x}\right) \propto e^{-\frac{\|\boldsymbol{x}\|^2}{2\tilde{\sigma}^2}}, \tag{58}$$

that displaces the quadratures of the state by some random amplitude drawn from a centered Gaussian distribution with variance $\tilde{\sigma}^2$. Although this error model is widely used in the analysis of GKP codes [10–12] for its simplicity, one needs to be careful to note that in real implementations of GKP error correction, this is not a physically accurate model [2, 37] in general, but only reproduces the correct measurement statistics in specific cases in practice, such as when the finite squeezing error is applied to a perfect GKP state and interpreted as channel or when GKP states undergo a 0−photon loss event. The distance measure that we introduce in the following merely assumes that larger phase-space displacements are more likely than smaller ones, which is a natural choice and is sufficiently meaningful to indicate the robustness of the code with respect to realistic noise sources as also demonstrated in recent experiments [8].

Definition 3 (Euclidean distance of a GKP code)**.** *The (euclidean) distance Δ of a GKP code given by lattice $\mathcal{L}$ is the euclidean length of the shortest non-trivial logical operator, i.e.,*

$$\Delta = \Delta\left(\mathcal{L}\right) := \min_{0 \neq \boldsymbol{x} \in \mathcal{L}^{\perp}/\mathcal{L}} \|\boldsymbol{x}\|. \tag{59}$$

For CSS GKP codes, i.e., for which we have $\mathcal{L} = \mathcal{L}_q \oplus \mathcal{L}_p$ as specified earlier, we can further distinguish

$$\Delta_q := \min_{0 \neq \boldsymbol{x} \in \mathcal{L}_p^*/\mathcal{L}_q} \|\boldsymbol{x}\|, \tag{60}$$

$$\Delta_p := \min_{0 \neq \boldsymbol{x} \in \mathcal{L}_q^*/\mathcal{L}_p} \|\boldsymbol{x}\|. \tag{61}$$

5.1 Tradeoffs and bounds for GKP codes

Before discussing how the distance of GKP codes can be computed in general, we present a few results that relate the distance of GKP codes to other code properties such as number of modes n of the code, the encoded logical dimension $d = 2^k$ and size of stabilizers.

Lemma 1 (Distance bound)**.** *Remember that $\lambda_1\left(\mathcal{L}^{\perp}\right)$ denotes the shortest non-zero vector in the dual lattice. We have*

$$\Delta \geq \lambda_1\left(\mathcal{L}^{\perp}\right). \tag{62}$$

Proof. Because the lattice vector $0 \neq \boldsymbol{x} \in \mathcal{L}^{\perp}/\mathcal{L}$ for which $\|\boldsymbol{x}\|$ is minimal is also in $\mathcal{L}^{\perp}$ this holds trivially. □

We have already seen in the previous section that for GKP codes obtained from scaling a symplectically self-dual lattice $\mathcal{L}_0 = \mathcal{L}\left(M_0\right)$ to $\mathcal{L} = \sqrt{\lambda}\mathcal{L}_0$ we have

$$\Delta_{\text{scaled}} = \lambda_1\left(\mathcal{L}^{\perp}\right) = \lambda^{-\frac{1}{2}}\lambda_1\left(\mathcal{L}_0\right), \tag{63}$$

so the distance decreases while the size of the stabilizers and number of encoded logical dimensions increases. In particular, scaled codes are not straightforwardly *scalable* in the sense that we cannot increase the distance by adding generators while keeping the encoded logical dimensions fixed. It has been shown [38], however (see also ref. [13]), that symplectically self dual lattices in $\mathbb{R}^{2n}$ with $\lambda_1 \geq 2\sqrt{\frac{n}{\pi e}}$ exist, implicating that it is possible to find sequences of codes that can be scaled to *achieve* rates $R = \log_2 \lambda$,

$$\lambda = \lfloor \frac{\hbar}{2\tilde{\sigma}^2} \rfloor \tag{64}$$

for transmission through stochastic Gaussian displacement channels with variance $\tilde{\sigma}^2$ [14]. On the other hand, for the concatenation of a single mode GKP code into a $[\![n, k, d]\!]$ qubit quantum error correcting code, we have

$$\Delta_{\text{conc}} \geq \sqrt{d}\Delta_{\text{loc}}, \tag{65}$$

where Δ_{loc} is the distance of the local single mode GKP codes, such as $\Delta_{\square} = 2^{-\frac{1}{2}}$ for the square GKP code or $\Delta_{\hexagon} = 3^{-\frac{1}{4}}$ for the hexagonal GKP code. Equality in eq. (65) holds when the code we consider is CSS or the local code is the hexagonal GKP code (such that all shortest non-trivial logical vectors have the same distance). Eq. (65) is verified by decomposing a shortest representative non-trivial logical vector $\boldsymbol{L} = \oplus_{i=1}^{n}\boldsymbol{l}_i$ into n local sub-blocks, where $\boldsymbol{l}_i$ is a logical operator of the respective local code, such that we have $\|\boldsymbol{L}\|^2 \geq d\Delta_{\text{loc}}^2$.

The distance of a concatenated code is typically strictly larger than $\lambda_1\left(\mathcal{L}^{\perp}\right)$, because the shortest dual vectors would correspond to stabilizer displacements, in particular if the input qubit stabilizer code is LDPC . This is e.g. the case for the concatenation of the square GKP code with *surface-17* (see Fig. 1), where we have $\Delta = \sqrt{3/2}$, while $\lambda_1\left(\mathcal{L}^{\perp}\right) = \lambda_1\left(\mathcal{L}\right) = 1$.

The length of the shortest vector in a lattice $\lambda_1\left(\mathcal{L}\right)$ is the first of $2n$ successive minima of the lattice. Generally, the ith successive minimum $\lambda_i\left(\mathcal{L}\right)$, $i \leq 2n$ is defined to be the smallest $r > 0$, such that $\mathcal{L}$ contains i linearly independent vectors of length at most r. It holds that $\mu\left(\mathcal{L}\right) \geq \lambda_{2n}\left(\mathcal{L}\right)/2$, where $\mu\left(\mathcal{L}\right)$ is the *covering radius* of the lattice $\mathcal{L}$, i.e., the minimum radius μ, such that the union of closed balls $\mathcal{B}_{\mu}\left(\boldsymbol{x}\right)$, $\boldsymbol{x} \in \mathcal{L}$ centred around each lattice point of $\mathcal{L}$ cover the entire space $\mathbb{R}^{2n}$. A related quantity is the *packing radius* of the lattice $\rho(\mathcal{L}) = \lambda_1(\mathcal{L})/2$. Successive minima of the direct and dual lattice $\mathcal{L}^*$ are related by so-called transference theorems, in particular we have

$$1 \leq \lambda_1(\mathcal{L})\,\lambda_{2n}(\mathcal{L}^*) \leq 2n. \tag{66}$$

We have noticed earlier that $\mathcal{L}^\perp$ and $\mathcal{L}^*$ differ only by an orthogonal transformation, therefore it holds that

$$\lambda_i\left(\mathcal{L}^\perp\right) = \lambda_i(\mathcal{L}^*) \tag{67}$$

and we can apply eq. (66) to relate the distance to the length of stabilizer vectors.

Theorem 3 (Distance bound)**.** *For a GKP code with lattice $\mathcal{L}$, distance Δ and maximal length C of a basis vector for a fixed basis M we have*

$$\Delta \geq \lambda_1\left(\mathcal{L}^\perp\right) \geq \lambda_{2n}^{-1}(\mathcal{L}) \geq C^{-1}, \tag{68}$$

as well as

$$\Delta \leq \lambda_{2n}\left(\mathcal{L}^\perp\right) \leq \frac{2n}{\lambda_1(\mathcal{L})}. \tag{69}$$

Proof. The first bound follows immediately from Lemma 1 and eq. (66) by swapping the roles of $\mathcal{L}^\perp$ and $\mathcal{L}$, which is possible because $\left(\mathcal{L}^\perp\right)^\perp = \mathcal{L}$ and further from $C \geq \lambda_{2n}(\mathcal{L})$. Similar for the second bound. □

These bounds indicate an intimate relation between the lengths of the stabilizers in $\mathcal{L}$ and the distance of the GKP code, which we will expand on further in the next section.

5.2 Symplectic transformations

It is interesting to study how symplectic transformations change the code distance. In particular, since we have seen in Theorem 2 that GKP codes with equal standard form are symplectically equivalent, it is possible that for specific noise models a symplectic transformation of the stabilizers $M \mapsto MS^T$ can be used to improve the codes resilience to noise. It is clear that orthogonal (symplectic) transformations S satisfying $S^TS = I$ leave the distance invariant.

This is not the case for squeezing. In particular if we consider a uniform squeeze

$$S = \eta I_n \oplus \eta^{-1} I_n, \quad \eta \in (0, \infty) \tag{70}$$

applied to a CSS code, we have

$$\Delta_q \mapsto \Delta_{q,\eta} = \eta^{-1}\Delta_q, \qquad \Delta_p \mapsto \Delta_{p,\eta} = \eta\Delta_p, \tag{71}$$

so that we can adapt the squeezing to account for potential bias in the noise, e.g. when

$$P(\boldsymbol{x}) = G[0, \tilde{\sigma}^2\left(\eta^{-1} I_n \oplus \eta I_n\right)](\boldsymbol{x}). \tag{72}$$

For concatenated CSS GKP codes, we generally have

$$\Delta_q = \sqrt{d_Z}\,\Delta_{q,\mathrm{loc}}, \qquad \Delta_p = \sqrt{d_X}\,\Delta_{p,\mathrm{loc}}, \tag{73}$$

such that squeezing the local codes by η is equivalent to increasing (decreasing) the upper level $X/Z-$ distances $d_{X/Z}$ by a factor of η^2. On the other hand, for natively unbiased noise it is also possible to squeeze the local code and employ a qubit quantum error correcting codes tailored towards biased noise such that $\Delta_q = \Delta_p$ remains constant. Although such a setup leaves the distance invariant, it can still lead to improvements in the error correction procedure when dedicated decoders for the qubit error correcting codes are used, as was recently demonstrated in ref. [12].

Corollary 2 can also be used to derive an upper bound on the distance of a given code from symplectic equivalence as follows. Let us first consider a code $\mathcal{L}$ with generator M, encoding a single qubit within n modes. Suppose M is in canonical form, such that

$$A = MJM^T = J_2 \otimes D \text{ with } \quad D = \mathrm{diag}\{2, 1, \ldots, 1\}. \tag{74}$$

Corollary 2 implies that there exists a symplectic matrix S such that $M = N_\square S^T$ with $N_\square$ the generator of the code in the corollary with logical dimension two. Since $N_\square$ is diagonal, the corresponding lattice $\mathcal{L}_\square$ is trivially orthogonal, and so is the dual $\mathcal{L}_\square^\perp$. The shortest non-trivial logical operators are thus immediately found as $\boldsymbol{\eta}_{\square,1}^T = \left(1/\sqrt{2}, \mathbf{0}_{2n-1}^T\right)$ and $\boldsymbol{\eta}_{\square,2}^T = \left(0, -1/\sqrt{2}, \mathbf{0}_{n-2}^T,\right)$. We can now recall that commutation relations for the displacements are related to symplectic products of the corresponding phase space vectors and that S preserves symplectic products. Hence the transformation S maps all points in $\mathcal{L}_\square^\perp$ that correspond to stabilizers, $\boldsymbol{x} \in \mathcal{L}_\square \subset \mathcal{L}_\square^\perp$, to direct lattice points $S\boldsymbol{x} \in \mathcal{L}$, which are stabilizers of $\mathcal{L}$, and all non-trivial logical operators to non-trivial logical operators. Therefore, we can readily write down two logical operators, $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \mathcal{L}^\perp \setminus \mathcal{L}$, as $\boldsymbol{\eta}_j = S\boldsymbol{\eta}_{\square,j}$ whose lengths are

$$||\boldsymbol{\eta}_j|| = \left|\left|S\boldsymbol{\eta}_{\square,j}\right|\right| \geq \Delta. \tag{75}$$

Note that the bound is not necessarily tight. One can further relate the distance to the *squeezing* contained in S. Applying the Bloch-Messiah or Euler decomposition [39] we can write $S = R_2 K R_1$ with R_j symplectic orthogonal matrices and $K = \operatorname{diag}\{k_1, \dots, k_n, 1/k_1, \dots, 1/k_n\}$ with $k_l \in (0, \infty)$ denoting an effective squeezing operation. Similar to ref. [15], we denote by $\mathtt{sq}(S) = \sqrt{\mathrm{EV}_{max}\left(S^T S\right)}$ the root of largest eigenvalue of $S^T S$, or equivalently, the largest squeezing factor in the Bloch-Messiah decomposition of the associated symplectic matrix. Finally, we obtain

$$\Delta \leq \|S\boldsymbol{\eta}_{\square,j}\| \leq \|K\| \|\boldsymbol{\eta}_{\square,j}\| = \frac{1}{\sqrt{2}} \mathtt{sq}(S). \tag{76}$$

The bound presented above generalizes straightforwardly to codes with higher logical dimension, such that we obtain the following bound.

Theorem 4 (Squeezing bound to the distance)**.** *Let $\mathcal{L} = \mathcal{L}(M)$ specify a GKP code with symplectic Gram matrix $A = J_2 \otimes D$ in its canonical form. Further let S denote the symplectic matrix that transforms between M and the generator $N_\square$ as specified in Corollary 2. We have*

$$\Delta \leq \sqrt{\max_j D_{j,j}}^{-1} \mathtt{sq}(S). \tag{77}$$

This bound is interesting since it, similar to the bound presented in ref. [15], bounds a benchmark for error correcting capabilities of a code by the squeezing measure of the Gaussian unitary necessary to prepare the code state. In ref. [15] that bound implies a no-go theorem on the existence of thresholds for oscillator into (many-) oscillator codes by the assumption that the squeezing measure $\mathtt{sq}(S)$ is naturally bounded. Here, it is already known that code families with increasing distance exist, e.g., given by concatenation of single mode GKP codes with the surface code which have distance scaling as $\Delta \propto n^{1/4}$, where n is the number of modes. The crucial difference between our scenarios is, however, that ref. [15] focuses on the channel coding setup such that each code state necessarily is prepared by a Gaussian unitary encoding channel, while in the general quantum error correction setting such as considered here, code states can be formed by measurements of stabilizers (possibly of a LDPC code) and correction via application of displacement operators, forgoing the absolute constraint posed by the squeezing measure. Similar to the argument in ref. [15], Theorem 4, however, does restrict the possible effective protection a code state prepared via a Gaussian unitary acting on code states obtained from the associated normal form as in Corollary 2. Finally, it is worth mentioning that, by the Bloch-Messiah decomposition, a non-trivial squeezing content $\mathtt{sq}(S)$ as specified in Theorem 4 is a necessary ingredient to obtain codes with non-orthogonal lattices and, effectively, non-trivial distance scaling.

5.3 Computing the distance

In general, computing the distance of a GKP code given only the lattice $\mathcal{L}$ or a corresponding basis M is computationally hard. Given a non-trivial logical displacement $\boldsymbol{\xi}^\perp \in \mathcal{L}^\perp$, a candidate for the distance is computed by fully reducing $\boldsymbol{\xi}^\perp$ with respect to the lattice $\mathcal{L}$, that is one needs to find the length of the shortest *representative* of $\boldsymbol{\xi}^\perp$ modulo the lattice $\mathcal{L}$. Equivalently, it is necessary to solve the *optimization-closest vector problem (optCVP)* defined as

$$\mathrm{optCVP}(\boldsymbol{x}, \mathcal{L}) := \min_{\boldsymbol{\xi} \in \mathcal{L}} \|\boldsymbol{x} - \boldsymbol{\xi}\| = \mathrm{dist}(\boldsymbol{x}, \mathcal{L}). \tag{78}$$

A related problem is the *(search) closest vector problem (CVP)* that aims at identifying the closest lattice vector to the input

$$\mathrm{CVP}\,(\boldsymbol{x},\mathcal{L}) := \underset{\boldsymbol{\xi}\in\mathcal{L}}{\arg\min}\,\|\boldsymbol{x}-\boldsymbol{\xi}\|, \tag{79}$$

which given inputs $\boldsymbol{\xi}^{\perp}$ and $\mathcal{L}$, outputs the shortest representative of $\boldsymbol{\xi}^{\perp}$. These problems are known to be NP-hard. Naively, one may be tempted to try to identify the shortest logical vectors by going through the rows of $M_i^{\perp} = \boldsymbol{\xi}_i^{\perp\,T}$ and rounding it element-wise to the closest element in $\mathcal{L}$, i.e.

$$\left(\boldsymbol{\xi}_i^{\perp}/M\right)^T = \boldsymbol{\xi}_i^{\perp\,T} - \lfloor \boldsymbol{\xi}_i^{\perp\,T} M^{-1} \rceil M, \tag{80}$$

where $\lfloor x \rceil$ denotes the closest integer to x, [5] in other words identifying the shortest logical representatives with non-zero rows of

$$M^{\perp}/M = M^{\perp} - \lfloor M^{-T}J^{-1}M^{-1} \rceil M = M^{\perp} - \lfloor A^{-1} \rceil M = \left(I - \lfloor A^{-1} \rceil A\right) M^{\perp}. \tag{81}$$

Indeed, we have found numerically for the concatenated GKP code built on Surface-17 with M given in its HNF that this rounding method outputs the shortest logical representatives. This is, however, a lucky coincidence. It is known that the rounding method for solving CVP [40] perform poorly in general and only returns an accurate solution if the lattice basis is sufficiently well-behaved, e.g. when the rows of M are orthogonal. This is consistent with the fact that in general the solution of CVP is also hard to *approximate* better than up to an exponential factor in the dimension. In addition, it is not *a priori* granted that the basis for the symplectic dual $M^{\perp}$ consists of those vectors that correspond to the shortest logical displacements only up to stabilizers. Hence while this method allows to upper bound the distance Δ of a general GKP code by

$$\Delta \leq \tilde{\Delta} := \min_i \|\left(M^{\perp}/M\right)_i\|, \tag{82}$$

the r.h.s. is not guaranteed to yield an accurate estimate of the code distance. Here, we propose a different strategy to compute the distance that is feasible in particular for scaled GKP codes and concatenated codes. We introduce the *theta function* of a lattice [16]

$$\Theta_{\mathcal{L}}(z) = \sum_{\boldsymbol{x}\in\mathcal{L}} q^{\boldsymbol{x}^T\boldsymbol{x}} = \sum_{\delta\in\mathcal{D}} N_\delta q^{\delta}, \tag{83}$$

where $q = e^{i\pi z}$. $\mathcal{D} = \left\{\|\boldsymbol{x}\|_2^2,\, \boldsymbol{x}\in\mathcal{L}\right\}$ is the set of squared distances of $\mathcal{L}$. We have also introduced the number of lattice vectors of a given length

$$N_\delta = \#\left\{\boldsymbol{x}\in\mathcal{L} :\, \boldsymbol{x}^T\boldsymbol{x} = \delta\right\}. \tag{84}$$

For integral lattices, i.e., when the corresponding (euclidean) Gram matrix satisfies $G = MM^T \in \mathbb{Z}^{2n\times 2n}$, we can set $\mathcal{D} = \mathbb{N}$ and N_m is given by the number of integer solutions $\boldsymbol{n}\in\mathbb{Z}^{2n}$ to the equation $\boldsymbol{n}^T G\boldsymbol{n} = m$, which is an example of a *Diophantine equation*. We call the pair $(\mathcal{D}, N_\delta)$ the *distance distribution* of the lattice $\mathcal{L}$. The theta function converges and is holomorphic for $\mathrm{Im}(z) > 0$. The first summands are given by

$$\Theta_{\mathcal{L}}(z) = 1 + \tau q^{\lambda_1} + \ldots, \tag{85}$$

where $\lambda_1 = \lambda_1(\mathcal{L})$ is the length of the shortest vector of the lattice and τ is known as the *kissing number*, the number of minimal-length vectors of $\mathcal{L}$. It is known that the theta function for the (euclidean) dual of a lattice $\mathcal{L}\subset\mathbb{R}^{2n}$ is given by

$$\Theta_{\mathcal{L}^*}(z) = \det(\mathcal{L})\left(\frac{i}{z}\right)^n \Theta_{\mathcal{L}}\left(-\frac{1}{z}\right), \tag{86}$$

which follows from the Poisson summation formula (see also ref. [41]). In particular, since we have seen that $\mathcal{L}^*$ and $\mathcal{L}^{\perp}$ only differ by an orthogonal transformation, which does not change the length of lattice vectors, and hence the theta function, we also have

$$\Theta_{\mathcal{L}^{\perp}}(z) = \Theta_{\mathcal{L}^*}(z), \tag{87}$$

[5] acting element-wise

such that we can immediately obtain the theta function of $\mathcal{L}^{\perp}$ whenever the the theta function of the direct lattice $\Theta_{\mathcal{L}}(z)$ is known. By definition of the theta function, the distance of a GKP code specified by $\mathcal{L}$ is given by the smallest non-zero power of q in

$$Q_{\mathcal{L}}(z) := \Theta_{\mathcal{L}^{\perp}}(z) - \Theta_{\mathcal{L}}(z) = N_{\Delta^2} q^{\Delta^2} + \ldots, \tag{88}$$

and furthermore since we have that $\Theta_{\mathcal{L}}(z)$ is uniquely determined by the distance distribution $(\mathcal{D}, N_\delta)$ of $\mathcal{L}$, which by eq. (86) also fully specifies $\Theta_{\mathcal{L}^{\perp}}(z)$. Note that $Q_{\mathcal{L}}(z) > 0$ since $\mathcal{L} \subseteq \mathcal{L}^{\perp}$. We arrive at the following insight.

Theorem 5 (Code distance is specified by the distance distribution)**.** *The distance of a GKP code specified by $\mathcal{L}$ is uniquely determined by its distance distribution $(\mathcal{D}, N_\delta)$.*

The theta function of a scaled lattice is

$$\Theta_{\sqrt{\lambda}\mathcal{L}_0}(z) = \Theta_{\mathcal{L}_0}(\lambda z). \tag{89}$$

Many expressions of theta functions of symplectically self dual lattices, in particular those that also correspond to euclidean self-dual lattices are known in the literature [16] such that their corresponding distances can e.g. be estimated from a logarithmic fit for small $q \ll 1$ of eq. (88), or by expressing Q in a basis for which the distance distribution is known. We have verified that this method works numerically for some small examples including single mode scaled codes and the concatenated GKP-repetition code using the *weight-distribution* as outlined below.

For concatenated (square) GKP codes, where $\mathcal{L} = \Lambda(Q)$ is given by a Construction A lattice, we can express the theta function using the *weight enumerator*. We first introduce the *weight distribution* of a linear code Q, which is given by the numbers $\{A_i\}_{i=0}^{2n}$ of codewords q in $Q \subset \mathbb{Z}_2^{2n}$ with Hamming weight $\mathrm{wt}(q) = i$. Crucially, the weight distribution $\{A_i\}_{i=0}^{2n}$ here refers to the Hamming-weight distribution of the *symplectic representation* of the qubit stabilizers. The weight enumerator is given by

$$W_Q(x, y) = \sum_{q \in Q} x^{2n - \mathrm{wt}(q)} y^{\mathrm{wt}(q)} = \sum_{i=0}^{2n} A_i x^{2n-i} y^i. \tag{90}$$

Using this definition, we can express the theta function of a construction A lattice [16] by straightforward computation

$$\Theta_{\Lambda(Q)} = W_Q\left(\theta_3(2z), \theta_2(2z)\right), \tag{91}$$

where

$$\theta_3(z) = \Theta_{\mathbb{Z}}(z) = \sum_{m \in \mathbb{Z}} q^{m^2}, \tag{92}$$

$$\theta_2(z) = \Theta_{\mathbb{Z}+\frac{1}{2}}(z) = \sum_{m \in \mathbb{Z}} q^{\left(m+\frac{1}{2}\right)^2}. \tag{93}$$

Since we see that we can find the theta function of a Construction A lattice corresponding to a concatenated code by means of the (Hamming) weight distribution of its (qubit) stabilizer group, we have similar to Theorem 5 and by eq. (65),

Corollary 3 (Distance of CSS qubit stabilizer code from weight distribution)**.** *The distance d of a CSS qubit stabilizer code is fully determined by the weight distribution $\{A_i\}_{i=0}^{2n}$ of its stabilizers.*

We note that this corollary also follows from a quantum version of the weight enumerators defined by Shor and Laflamme [17] and Rains [18], which through their immediate relationship to the Quantum error correction conditions [42] impose strong restrictions on possible quantum error correcting codes. Generally, when $\mathcal{L}$ is integral (rational), $\Theta_{\mathcal{L}}(z)$ is a modular form[6], i.e., it remains invariant under certain Möbius transformations

$$z \mapsto \frac{az + b}{cz + d} \tag{94}$$

[6] if the lattice can be scaled to be integral, we consider the scaled version before doing the analysis and scale back accordingly afterwards.

and as a result can be expressed in a finite basis depending on the specific symmetry group of the theta function [43, 41, 16]. This avenue is interesting for the explicit computation of distances but extends beyond the scope of the present article. We thus defer further analysis of a modular forms approach to finding distances to future work.

6 Maximum likelihood decoding GKP codes

In this section, we discuss active quantum error correction and *maximum likelihood decoding (MLD)* for general GKP codes using the tools discussed in this work. The introduction to MLD follows closely the discussion in ref. [10]. We assume a stochastic Gaussian displacement noise channel as specified in eq. (57) with variance $\tilde{\sigma}^2$. For comparison with the literature, when the displacement operators are defined by a more "standard" convention without the overall constant $\sqrt{2\pi}$, this corresponds to a physical variance of $\sigma^2 = 2\pi\tilde{\sigma}^2$. Upon sampling an error $\boldsymbol{e}$ and measuring the stabilizers, a syndrome vector of the form

$$\boldsymbol{s}(\boldsymbol{e}) = MJ\boldsymbol{e} \mod 1 \tag{95}$$

is obtained as the phases of the eigenvalues of the stabilizer generators $\left\{D\left(M_i^T\right)\right\}_{i=1}^{2n}$ when acting on a code state vector $|\psi\rangle$ displaced by an error vector $\boldsymbol{e}$,

$$D\left(M_i^T\right) D\left(\boldsymbol{e}\right) |\psi\rangle = e^{i2\pi M_i J\boldsymbol{e}} D\left(\boldsymbol{e}\right) |\psi\rangle\,, \tag{96}$$

where $M_i = \boldsymbol{\xi}_i^T$ is the i'th row of M.

Since we are dealing with full rank lattices, given the syndrome, we can assign a pure error

$$\boldsymbol{\eta}(\boldsymbol{s}) = (MJ)^{-1}\boldsymbol{s} \tag{97}$$

that has the same syndrome $\boldsymbol{s}$ as $\boldsymbol{e}$ as "initial guess" for the correction. The net displacement of amplitude $\boldsymbol{e} - \boldsymbol{\eta}$ is necessarily in the dual lattice, and hence maps the code space to itself, but it may be logically non-trivial. It might thus be necessary to modify the initial guess by a logical vector (leaving the syndrome invariant), such that $\boldsymbol{e} - \boldsymbol{\eta}$ is a stabilizer. Note that any $\boldsymbol{\eta} + \boldsymbol{l}$ with $\boldsymbol{l} \in \mathcal{L}^\perp$ will lead to the same syndrome, so the choice of the additional logical displacement cannot be inferred from the syndrome alone. Rather, it is necessary to take the noise model into account.

To find the appropriate logical post-correction, for every $\boldsymbol{\xi}^\perp \in \mathcal{L}^\perp/\mathcal{L}$ we evaluate the probabilities that, given syndrome $\boldsymbol{s}$, the actual error is stabilizer equivalent to $\boldsymbol{\eta}(\boldsymbol{s}) + \boldsymbol{\xi}^\perp$, which is given by

$$P([\boldsymbol{\eta}(\boldsymbol{s}) + \boldsymbol{\xi}^\perp]|\boldsymbol{s}) = P^{-1}(\boldsymbol{s}) \sum_{\boldsymbol{\xi}\in\mathcal{L}} P_{\tilde{\sigma}}(\boldsymbol{\eta}(\boldsymbol{s}) + \boldsymbol{\xi}^\perp + \boldsymbol{\xi}), \tag{98}$$

where $[\boldsymbol{x}] = \{\boldsymbol{x} + \boldsymbol{\xi}, \boldsymbol{\xi} \in \mathcal{L}\}$ and $P_{\tilde{\sigma}}$ is as specified in eq. (57). This can be rewritten as

$$\begin{aligned} P([\boldsymbol{\eta}(\boldsymbol{s}) + \boldsymbol{\xi}^\perp]|\boldsymbol{s}) &= P^{-1}(\boldsymbol{s}) \sum_{\boldsymbol{\xi}\in\mathcal{L}+\boldsymbol{\eta}(\boldsymbol{s})+\boldsymbol{\xi}^\perp} P_{\tilde{\sigma}}(\boldsymbol{\xi}) & (99)\\ &= \sqrt{2\pi\tilde{\sigma}^{2n}}^{-1} P^{-1}(\boldsymbol{s})\Theta_{\mathcal{L}+\boldsymbol{\eta}(\boldsymbol{s})+\boldsymbol{\xi}^\perp}\left(\frac{i}{2\pi\tilde{\sigma}^2}\right), & (100)\end{aligned}$$

proportional to the theta series of the *packing* $\mathcal{P} = \mathcal{L} + \boldsymbol{\eta}(\boldsymbol{s}) + \boldsymbol{\xi}^\perp$ evaluated in $z = \frac{i}{2\pi\tilde{\sigma}^2}$ [7]. Let $\left(\tilde{\mathcal{D}}, N_{\tilde{\delta}}\right)$ denote the distance distribution of $\mathcal{P}$. We can write the coset probabilities above in a *small error* or *"low temperature expansion"*. [8]

$$\Theta_{\mathcal{L}+\boldsymbol{\eta}(\boldsymbol{s})+\xi^\perp}\left(\frac{i}{2\pi\tilde{\sigma}^2}\right) = \sum_{\tilde{\delta}\in\tilde{\mathcal{D}}} N_{\tilde{\delta}} q^{\tilde{\delta}}. \tag{101}$$

[7] $\mathcal{P}$ as the translate of a lattice $\mathcal{L}$ is formally not a lattice, in particular $\mathcal{P}$ may not contain the origin.

[8] Note that in contrast to the usual literature we do not distinguish a Nishimori line, such that these are the same things.

evaluated at $q = \exp\left(-1/2\tilde{\sigma}^2\right)$. Finally, MLD decoding is implemented by applying the total correction

$$\overline{\boldsymbol{\eta}} = \boldsymbol{\eta}(\boldsymbol{s}) + \underset{\boldsymbol{\xi}^{\perp}\in\mathcal{L}^{\perp}/\mathcal{L}}{\arg\max}\, P([\boldsymbol{\eta}(\boldsymbol{s}) + \boldsymbol{\xi}^{\perp}]|\boldsymbol{s}). \tag{102}$$

Methods to evaluate or sample the theta function of a lattice have been discussed in the literature [43]. Here, we find that, given an algorithm that reliably estimates the theta function of a packing, we can also approximately solve MLD decoding.

Minimum energy decoding (MED). In the limit $\tilde{\sigma}, q \to 0$ the sum (101) becomes sharply distributed around solutions with minimal $\tilde{\delta}$. That is, the bulk of the sum (101) is determined by

$$\underset{\xi\in\mathcal{L}}{\arg\min}\, \|\xi + \boldsymbol{\eta}(\boldsymbol{s}) + \xi^{\perp}\|. \tag{103}$$

The MLD correction then becomes

$$\begin{aligned}\overline{\boldsymbol{\eta}} &= \boldsymbol{\eta}(\boldsymbol{s}) + \underset{\xi^{\perp}\in\mathcal{L}^{\perp}/\mathcal{L}}{\arg\min}\ \min_{\xi\in\mathcal{L}} \|\boldsymbol{\eta}(\boldsymbol{s}) + \xi^{\perp} + \xi\| && (104)\\ &= \boldsymbol{\eta}(\boldsymbol{s}) + \underset{\xi^{\perp}\in\mathcal{L}^{\perp}}{\arg\min}\, \|\boldsymbol{\eta}(\boldsymbol{s}) + \xi^{\perp}\| && (105)\\ &= \boldsymbol{\eta}(\boldsymbol{s}) - \underset{\xi^{\perp}\in\mathcal{L}^{\perp}}{\arg\min}\, \|\boldsymbol{\eta}(\boldsymbol{s}) - \xi^{\perp}\|. && (106)\end{aligned}$$

The last line Eq (106) shows that this immediate version of MED requires solving CVP on the lattice $\mathcal{L}^{\perp}$. In the special case of a concatenated code, $\mathcal{L} = \Lambda(Q)$, we have that $\mathcal{L}_{\mathrm{GKP}} := \left\{\boldsymbol{x} : \sqrt{2}\boldsymbol{x} \mod 2 = 0\right\}$ with generator matrix $M_{\mathrm{GKP}} = \sqrt{2}I_{2n}$ is a sublattice of $\mathcal{L}$ and its dual contains $\mathcal{L}^{\perp} \subseteq \mathcal{L}_{\mathrm{GKP}}^{\perp}$. In the literature, a common approach to solve MLD for concatenated GKP codes is to apply the above outlined CVP decoder to the superlattice $\mathcal{L}_{\mathrm{GKP}}^{\perp}$, which leaves a net qubit level syndrome to be decoded with a standard qubit decoder [12, 11, 10]. Solving CVP on $\mathcal{L}_{\mathrm{GKP}}^{\perp}$ is easy since it is an orthogonal lattice and one can straightforwardly apply the rounding trick outlined in the previous section. However, as we see below in such a two-step approach the initial guess for the correction on the sublattice $\mathcal{L}_{\mathrm{GKP}}$ is arbitrary.

MLD decoding concatenated codes. For concatenated (square) GKP codes, we dissect the initial correction in two components

$$\boldsymbol{\eta}(\boldsymbol{s}) = \boldsymbol{\eta}^{\mathrm{GKP}} + \boldsymbol{\eta}^{Q}, \tag{107}$$

such that

$$\boldsymbol{\eta}^{\mathrm{GKP}} = (M_{GKP}J)^{-1}\boldsymbol{s}_{\mathrm{GKP}} \tag{108}$$

reproduces the syndrome detected by the single-mode GKP stabilizers $\mathcal{L}_{\mathrm{GKP}}$ and $\boldsymbol{\eta}^{Q} \in \mathcal{L}_{\mathrm{GKP}}^{\perp} = \mathbb{Z}^{2n}/\sqrt{2}$ is such that the composite initial guess is consistent with the overall syndrome. For notational clarity we suppress the explicit dependency on the syndrome $\boldsymbol{s}$. The relevant theta function evaluates to

$$\Theta_{\mathcal{L}+\boldsymbol{\eta}(\boldsymbol{s})+\boldsymbol{\xi}^{\perp}}(z) = \sum_{\boldsymbol{x}\in Q}\prod_{i=1}^{2n}\sum_{v_i\in\mathbb{Z}} q^{\left[\left(\boldsymbol{x}+\sqrt{2}\boldsymbol{\eta}^{\mathrm{GKP}}+\sqrt{2}\boldsymbol{\eta}^{Q}+\sqrt{2}\boldsymbol{\xi}^{\perp}\right)_i+2v_i\right]^2}, \tag{109}$$

where we see that we can w.l.o.g. reduce $\boldsymbol{\eta}^{Q}, \boldsymbol{\xi}^{\perp} \in \frac{1}{\sqrt{2}}\mathbb{Z}_2^{2n}$ and consider the sum in the exponent $\boldsymbol{x} + \sqrt{2}\boldsymbol{\eta}^{Q} + \sqrt{2}\boldsymbol{\xi}^{\perp}$ to be carried out modulo 2. [9] We interpret the individual terms

$$f\left(b_i|\eta_i^{\mathrm{GKP}}\right) \equiv \sum_{v_i\in\mathbb{Z}} q^{\left(\sqrt{2}\eta_i^{\mathrm{GKP}}+b_i+2v_i\right)^2} = e^{-\frac{1}{2\tilde{\sigma}^2}\left(\sqrt{2}\eta_i^{\mathrm{GKP}}+b_i\right)^2}\theta_3\left(\frac{i}{\tilde{\sigma}^2}\left(\sqrt{2}\eta_i^{\mathrm{GKP}}+b_i\right)\Big|\frac{2i}{\pi\tilde{\sigma}^2}\right) \tag{110}$$

[9] Any other choice just permutes the sum.

as proportional to an indicator for the likelihood of a local bit value $b_i = x_i + \sqrt{2}\eta_i^Q + \sqrt{2}\xi_i^\perp \in \mathbb{Z}_2^{2n}$ where we have used the definition

$$\theta_3(\xi|z) = \sum_{m\in\mathbb{Z}} e^{2mi\xi+\pi izm^2}, \;\; \mathrm{Im}(z) > 0. \tag{111}$$

Overall, eq. (109) can then be rewritten as

$$\Theta_{\mathcal{L}+\boldsymbol{\eta}(\boldsymbol{s})+\xi^\perp}\left(\frac{i}{2\pi\tilde{\sigma}^2}\right) = \sum_{\boldsymbol{x}\in Q}\prod_{i=1}^{2n} f\left(b_i|\eta_i^{GKP}\right), \tag{112}$$

reproducing a form for the MLD probability similar to those provided in ref. [12] for the surface-GKP code. It is clear that this derivation can be carried out similarly for the concatenation with other single-mode GKP codes or more general codes, such as those derived from glued lattices discussed in the next section. The MLD probability in eq. (112) can be expressed using a local tensor network whenever the underlying qubit code is LDPC as has been demonstrated in refs. [44, 12]. Although qualitatively similar, the effective local error probabilities in this formulation are not exactly the same as in refs. [11, 12] since normalization constants $\sum_{b_i\in\mathbb{Z}_2} f\left(b_i|\eta_i^{\mathrm{GKP}}\right)$ cannot be immediately pulled in from the global normalization $P(s)$ below each $f\left(b_i|\eta_i^{GKP}\right)$ term. Besides the generality of this formulation of the MLD for GKP codes, it would be interesting to test whether it also provides a quantitative advantage in decoding. We examine this question more in-depth in a forthcoming publication.

7 Other constructions of GKP codes

In this section we present two constructions of GKP codes that go beyond the scaled and concatenated GKP codes that are based on lattice glueing and the lattice tensor product. We expect that such alternative constructions allow for more flexibility in designing codes in the future, in particular when noise varies in strength for different modes or when no physical locality restrictions are present such as in photonic architectures. The following constructions are adapted from the discussion of glued (euclidean) lattices and the lattice tensor product in refs. [16, 45].

7.1 Glued codes beyond concatenation

We can understand concatenated GKP codes as GKP codes built on certain types of *glued lattices*, which include those obtained through Construction A. Here, we give a brief overview of glueing theory for symplectic lattices, leaning on the description in ref. [16]. We begin by dissecting a (glued) symplectic lattice with symplectic sublattice in a top-down approach to understand its structure and then move to a bottom-up approach to construct a glued lattice from a base lattice by appending an appropriate glue group. Let us assume that we have a $2n-$dimensional symplectic lattice $\mathcal{L}$ that has a (symplectic) sublattice $\mathcal{L}_0$ with direct sum structure

$$\mathcal{L}_0 = \bigoplus_{i=1}^{k}\mathcal{L}_i. \tag{113}$$

Vectors $\boldsymbol{v}\in\mathcal{L}$ can be written as

$$\boldsymbol{v} = \sum_i \boldsymbol{v}_i, \tag{114}$$

where $\boldsymbol{v}_i \in \mathbb{R}\otimes\mathcal{L}_i$. The symplectic inner product of any $\boldsymbol{v}_i$ with any vector of $\mathcal{L}_i$ is integer, such that it can be concluded that $\boldsymbol{v}_i \in \mathcal{L}_i^\perp$. Moreover we can add to any $\boldsymbol{v}_i$ a vector from $\mathcal{L}_i$ without changing the fact that $\boldsymbol{v}$ has integer symplectic inner product with any other vector of $\mathcal{L}$. It hence suffices to demand $\boldsymbol{v}_i \in \mathcal{L}_i^\perp/\mathcal{L}_i$. Such vectors are called *glue vectors* for $\mathcal{L}_i$, which in the coding language correspond to logical representatives of a local code. $\mathcal{L}_i^\perp/\mathcal{L}_i$ is also known as the (symplectic) dual quotient or *glue group* for $\mathcal{L}_i$. We can thus obtain symplectic lattices from a *base lattice* $\mathcal{L}_0 = \bigoplus_{i=1}^k \mathcal{L}_i$ by adding vectors $\boldsymbol{v}$ of the form in eq. (114), where each $\boldsymbol{v_i} \in \mathcal{L}_i^\perp/\mathcal{L}_i$. We also refer to the set of extra vectors $\boldsymbol{v}$ as the *glue group* G. Let $\mathcal{L} = \bigoplus_{i=1}^k \mathcal{L}_i \cup G$ be a glued lattice of this form. It is clear that we have $\mathcal{L}^\perp = \bigoplus_{i=1}^k \mathcal{L}_i^\perp \cap G^\perp$.

Generally, whenever we have *saturated* sublattices $\mathcal{L}_0 \subset \mathcal{L}$, i.e., $\mathbb{R} \otimes \mathcal{L}_0 = \mathbb{R} \otimes \mathcal{L}$, we have $|\mathcal{L}_0|/|\mathcal{L}| \in \mathbb{Z}$. For $g \in \mathcal{L}$ we consider the glue classes $[g] = g + \mathcal{L}_0$, which form an additive group that we denote

$$G = \langle g_1, \ldots, g_r \rangle. \tag{115}$$

Conversely, in a bottom-up approach, a glued lattice $\mathcal{L}$ can be constructed by considering a general glue group $G \subseteq \mathcal{L}_0^\perp/\mathcal{L}_0$ and forming $\mathcal{L} = \mathcal{L}_0[G] = \mathcal{L}_0 \cup G$. G is cyclic and isomorphic to $\mathbb{Z}_{n_1} \times .. \times \mathbb{Z}_{n_r}$, where $\mathbb{Z}_n$ is the cyclic group of order n, and each n_i is the order of the corresponding generator g_i, i.e the smallest positive integer such that $n_i g_i \in \mathcal{L}_0$. The determinant of the glued lattice $\mathcal{L}_0[G]$ can be computed as [45]

$$|\mathcal{L}_0[G]| = |\mathcal{L}_0|/|G| = |\mathcal{L}_0|/(\prod_{i=1}^r n_i). \tag{116}$$

To construct a *symplectic* glued lattice from a symplectic base lattice $\mathcal{L}_0$, it is important to take care that every g_i has integer symplectic inner product with every other g_j – i.e., G is itself a finite symplectic group and that each G has integer symplectic inner product with each $x \in \mathcal{L}_0$, i.e., $G \subset \mathcal{L}_0^\perp$. It is easy to see that the earlier considerations are reproduced for $\mathcal{L}_0 = \bigoplus_{i=1}^k \mathcal{L}_i$. Using (116) we can obtain the logical dimension of GKP codes associated to glued lattices $\mathcal{L}_0[G]$. E.g. for a concatenated GKP-qubit code, $G \subset \mathcal{L}_0^\perp/\mathcal{L}_0$ is identified with the outer code with, say, r lineary independent generators, each with order $n_i = 2$ in $\mathcal{L}_0$. $\mathcal{L}_0 = \mathcal{L}(\sqrt{2}I_{2n})$, such that we compute $|\mathcal{L}_0[G]| = 2^{n-r} = 2^k$, consistent with what we would expect. Similar to the derivation for concatenated codes, we do not expect it to be hard to derive the distance of a glued lattice $\mathcal{L}_0[G]$. However, this quantity strongly depends on the specifications of the glue group $\mathcal{L}_0^\perp/\mathcal{L}_0$ and G which makes it difficult to write down a general solution in closed form.

7.2 Tensor product codes

Aside from the glueing construction, it is also possible to obtain new codes by taking outer products of lattices. The idea behind this construction is akin to product constructions known for qubit quantum error correcting codes, namely the *hypergraph product codes* by Tillich and Zemor [46], where the defining structure of the code is a hypergraph, and Homological product codes by Bravyi and Hastings [47], where the code is defined via a cell complex. For GKP codes, the defining structure of the codes is given by a lattices, such that the tensor product for lattices serves as an immediate candidate for a similar construction.

Let $\mathcal{L}_1 = \mathcal{L}(M_1) \subset \mathbb{R}^2$ be a symplectic lattice with symplectic Gram matrix $A_1 = M_1 J_2 M_1^T$ and $\mathcal{L}_2 = \mathcal{L}(M_2) \subset \mathbb{R}^n$ an integral lattice with euclidean Gram matrix $G_2 = M_2 M_2^T$. The tensor product lattice is defined as $\mathcal{L}_\otimes = \mathcal{L}(M_1 \otimes M_2) = \mathcal{L}_1 \otimes \mathcal{L}_2 \subset \mathbb{R}^{2n}$, i.e., a basis for $\mathcal{L}_\otimes$ is given by $\{(M_1)_i \otimes (M_2)_j,\ i = 1, 2,\ j = 1, \ldots, n\}$. $\mathcal{L}_\otimes$ is a symplectic lattice due to the decomposition $J_{2n} = J_2 \otimes I_n$, and its symplectic Gram matrix reads

$$A_\otimes = (M_1 \otimes M_2) J_{2n} (M_1 \otimes M_2)^T = A_1 \otimes G_2, \tag{117}$$

which is integral by construction. The canonical dual basis is given by

$$M_\otimes^\perp = (J_{2n} M_\otimes^T)^{-1} = M_\otimes^{-T} J_{2n}^T = M_1^{-T} J_2^T \otimes M_2^{-T} = M_1^\perp \otimes M_2^*, \tag{118}$$

which forms a basis for the symplectically dual lattice $\mathcal{L}_\otimes^\perp = \mathcal{L}_1^\perp \otimes \mathcal{L}_2^*$. We have

$$|A_1 \otimes G_2| = |A_1|^n |G_2|^2, \tag{119}$$

$$k_\otimes = \frac{1}{2} \log_2 |A_1 \otimes G_2| = \frac{n}{2} \log_2 |A_1| + \log_2 |G_2|. \tag{120}$$

Theorem 6 (Distance of tensor product codes)**.** *The distance of the tensor product code*

$$\Delta_\otimes = \min_{0\neq x\in\mathcal{L}_\otimes^\perp/\mathcal{L}_\otimes} \|x\|$$

obeys

$$\max\left\{\frac{\Delta_1}{\lambda_n(\mathcal{L}_2)}, \frac{\Delta_2}{\lambda_2(\mathcal{L}_1)}\right\} \leq \Delta_\otimes \leq \Delta_1\Delta_2, \tag{121}$$

where

$$\Delta_1 = \min_{0\neq x\in\mathcal{L}_1^\perp/\mathcal{L}_1} \|x\|, \qquad \Delta_2 = \min_{0\neq x\in\mathcal{L}_2^*/\mathcal{L}_2} \|x\|. \tag{122}$$

Proof. The proof is analogous to that of Lemma 2 in ref. [47]. To prove the upper bound, let $x \in \mathcal{L}_1^\perp/\mathcal{L}_1$, $y \in \mathcal{L}_2^*/\mathcal{L}_2$, be minimal non-trivial logical representatives of each component codes. It is clear that $x\otimes y \in \mathcal{L}_\otimes^\perp$. Further we have $x\otimes y \notin \mathcal{L}_\otimes$ because we can pick $a \in \mathcal{L}_1^\perp$, $b \in \mathcal{L}_2^*$ such that $a^T J_2 x \in \mathbb{Q}_1$, $b^T y \in \mathbb{Q}_1$, where $\mathbb{Q}_1 = \mathbb{Q}\cap(0,1)$, yielding $(a\otimes b)^T J_{2n}(x\otimes y) \notin \mathbb{Z}$ (note that the symplectic inner product sets the commutation phase for the associated displacement operators). As such, we have obtained a non-trivial logical operator $x\otimes y \in \mathcal{L}_\otimes^\perp/\mathcal{L}_\otimes$ and $\Delta_\otimes \leq \|x\otimes y\| = \Delta_1\Delta_2$. Let $0 \neq \boldsymbol{\psi} \in \mathcal{L}_\otimes^\perp/\mathcal{L}_\otimes$ be a minimal length non-trivial vector. We can always choose $0 \neq c \in \mathcal{L}_1^\perp/\mathcal{L}_1$ and $d \in \mathcal{L}_2$ such that

$$(c\otimes d)^T(J_2\otimes I_n)\psi \notin \mathbb{Z}. \tag{123}$$

Let $vec^{-1}(\psi)$ be the un-vectorization of ψ, i.e., if $\psi = \sum_i x_i\otimes y_i$, $x_i \in \mathcal{L}_1^\perp$, $y_i \in \mathcal{L}_2^*$ we have $vec^{-1}(\psi) = \sum_i x_i y_i^T$. It holds that $vec^{-1}(\psi)d \in \mathcal{L}_1^\perp$ and $vec^{-1}(\psi)d \notin \mathcal{L}_1$ since otherwise $(c\otimes d)^T(J_2\otimes I_n)\psi = c^T J_2 vec^{-1}(\psi)d \in \mathbb{Z}$ for all choices of $\boldsymbol{c}, \boldsymbol{d}$, such that ψ is logically trivial, which is not the case by assumption. Hence, $vec^{-1}(\psi)d$ is a non-trivial representative of $\mathcal{L}_1^\perp/\mathcal{L}_1$. Using Cauchy-Schwartz we have $\|\psi\|\|d\| = \|vec^{-1}(\psi)\|_F\|d\| \geq \|vec^{-1}(\psi)d\| \geq \Delta_1$, where $\|\cdot\|_F$ is the Frobenius norm. We can always choose $d \in \mathcal{L}_2$ with length at most the covering radius $\lambda_n(\mathcal{L}_2)$ that satisfies eq. (123). This is because any $d \in \mathcal{L}_2$ can be written in a basis

$$d = \sum_{i=1}^n a_i\xi_i, \qquad a_i \in \mathbb{Z}, \qquad \xi_i \in \mathcal{L}_2 : \ \|\xi_i\| \leq \lambda_n\left(\mathcal{L}_2\right). \tag{124}$$

When d satisfies eq. (123), we have

$$\mathbb{Z} \not\ni \sum_{i=1}^n (c\otimes d)^T(J_2\otimes I_n)\psi = \sum_{i=1}^n a_i(c\otimes\xi_i)^T(J_2\otimes I_n)\psi. \tag{125}$$

There must be at least one summand $i = x$, for which $a_x(c\otimes\xi_x)^T(J_2\otimes I_n)\psi \notin \mathbb{Z}$. Since $a_x \in \mathbb{Z}$, it must hold that $(c\otimes\xi_x)^T(J_2\otimes I_n)\psi \notin \mathbb{Z}$. Finally, we obtain

$$\|\psi\| \geq \frac{\Delta_1}{\lambda_n(\mathcal{L}_2)}. \tag{126}$$

Following the same procedure, we choose $c \in \mathcal{L}_1, d \in \mathcal{L}_2^\perp/\mathcal{L}_2$ such that eq. (123) is satisfied to show that $\|\psi\|\|c\| \geq \Delta_2$. □

8 Conclusions and future work

In this work, we have shown how some of the core concepts of lattice theory can be fruitfully used to analyse the structure of GKP codes. We have introduced the notion of a euclidean code distance for GKP codes and shown its relation with other relevant code parameters, in particular the trade-off with the encoding rate and length of stabilizer operators. We have examined the structure of particular GKP code constructions, such as scaled and concatenated GKP codes and shown how extensions of those constructions can be obtained using tools already present in the lattice theory literature. Using a lattice theoretic perspective when discussing bounds on GKP code parameters, we observed that the distance of the code is intimately related to the length of the stabilizers. Leveraging the natural embedding of qubit codes in $\mathbb{R}^{2n}$ provided by the concatenation with GKP codes we have shown in Corollary 3 how this result leads to a new proof of a similar property for qubit stabilizer codes, which is simpler than those found in the literature. This highlights how concepts from lattice theory can also become valuable tools to understand qubit stabilizer codes. Concerning the role of symplectic operations, which are central both in the theory and experiments with bosonic systems, we have clarified which GKP codes are related through a symplectic transformation, extending previous results for the single-mode case. This can also be interpreted as code-switching via Gaussian unitaries. In particular, we have identified a necessary and sufficient condition in the equality of the symplectic Gram matrix in standard form.

We have discussed how the GKP codes highlighted in this work can be decoded in terms of a general maximum likelihood decoder for the GKP code, for which the defining logical coset probability is given by the lattice theta function. Algorithms to approximate or sample theta functions have been discussed in the literature [43], which we identify as possible route forward to build general purpose decoders for GKP codes. More details on the practical implementation of decoders are, however, to appear in a separate publication.

The theta function of the GKP code lattice is also proven to encode information about the distance of GKP codes. It is known that, depending on the lattice, the theta function may possess certain symmetries under conformal transformations, which allow it to be represented in a finitely generated basis. In other words, if the theta function has certain symmetries (e.g. invariance under $z \mapsto z + 2$ for integral lattices) there exists a finite basis classified by the symmetry which can be used to write the theta function. This can also potentially be useful for decoding. In general, it would be interesting to further investigate the connection between the code distance and possible conformal symmetries of the theta function of the lattice.

It is worth mentioning that the theta function of glued lattices also appears in place of the partition function of certain string theories [45]. Given that it represents the MLD probability discussed in the Section 6, which has also been interpreted in the past as the partition function of a *statistical mechanics model* [48, 10], it is an interesting open question whether a rigorous connection between string theories and the statistical mechanics model for decoding GKP codes can be leveraged to obtain a deeper understanding for either.

Finally, we have discussed constructions of GKP codes beyond scaled- and concatenated codes. We have seen that the toolbox of lattice theory allows for a remarkably versatile set of code constructions which have great potential for the search of good code families with potentially better parameters, better performance tailored to particular physical implementations or allowing for better decoders relying on the structure of the lattice, without reference to an intermediate qubit decoder. Further generalizations of GKP code constructions, such as subsystem variants by constructions on *shifted lattices* [45] also seem possible. Overall, we hope that this work will stimulate further interest in examining stabilizer codes, and in particular GKP codes, through the lens of lattice theory and will provide a foundation to import useful tools for the study of bosonic error correction.

Acknowledgments

J. C. thanks B. Terhal for asking whether product constructions for GKP codes can be found and N. P. Breuckmann, A. Ciani and J.-P. Seifert for helpful discussions. F. A. acknowledges the support of the Alexander von Humboldt foundation. This work has been funded by the BMBF (RealistiQ, for which it provides a key physics non-agnostic approach to quantum error correction, PhoQuant, QPIC-1, and QSolid) and the DFG (CRC 183, project B04, on entangled states of matter). We also gratefully acknowledge support from the Einstein Research Unit on quantum devices.

A Code space of GKP codes

In this section, we show how the code space of a general GKP code can be represented in terms of the Wigner function of the code space projector

$$\Pi_{GKP} = \sum_{g\in\mathcal{S}} g. \tag{127}$$

In particular, this illustrates in how far the code space is non-trivially influenced by the choice of pivot basis for the GKP code and the phase-sector ϕ_M. Consistent with our non-standard definition of the displacement operator, we define the generalized Wigner function of an operator O [49, 50]

$$W_O(\boldsymbol{x}) = \int_{\mathbb{R}^{2n}} d\boldsymbol{\eta}\, e^{-i2\pi\boldsymbol{x}^T J\boldsymbol{\eta}}\, \mathrm{Tr}\left[D(\boldsymbol{\eta})\, O\right], \tag{128}$$

such that we also have

$$\int_{\mathbb{R}^{2n}} d\boldsymbol{x} W_O(\boldsymbol{x}) = \mathrm{Tr}\left[O\right]. \tag{129}$$

And we equip us with the following lemma.

Lemma 2 (Dirac comb representation)**.** *Let $\mathcal{L}\subseteq\mathbb{R}^{2n}$ be a lattice with generator M with symplectic dual $\mathcal{L}^\perp$, we have*

$$\sum_{\boldsymbol{\xi}\in\mathcal{L}} e^{i2\pi\boldsymbol{\xi}^T J\boldsymbol{z}} = \frac{1}{\det(\mathcal{L})} \sum_{\boldsymbol{\xi}^\perp\in\mathcal{L}^\perp} \delta^{2n}\left(\boldsymbol{z}-\boldsymbol{\xi}^\perp\right). \tag{130}$$

Proof. We compute straightforwardly using one dimensional Poisson resummation $\sum_{n\in\mathbb{Z}} e^{i2\pi nx} = \sum_{k\in\mathbb{Z}} \delta(x-k)$

$$\begin{aligned}
\sum_{\boldsymbol{\xi}\in\mathcal{L}} e^{i2\pi\boldsymbol{\xi}^T J\boldsymbol{z}} &= \sum_{\boldsymbol{a}\in\mathbb{Z}^{2n}} e^{i2\pi\boldsymbol{a}^T MJ\boldsymbol{z}} && (131)\\
&= \prod_j \sum_{a_j\in\mathbb{Z}} e^{i2\pi a_j (MJ\boldsymbol{z})_j} && (132)\\
&= \prod_j \sum_{b_j\in\mathbb{Z}} \delta\left((MJ\boldsymbol{z})_j - b_j\right) && (133)\\
&= \frac{1}{\det(\mathcal{L})} \sum_{\boldsymbol{b}\in\mathbb{Z}^{2n}} \delta^{2n}\left(\boldsymbol{z} - \left(\boldsymbol{b}^T M^\perp\right)^T\right) && (134)\\
&= \frac{1}{\det(\mathcal{L})} \sum_{\boldsymbol{\xi}^\perp\in\mathcal{L}^\perp} \delta^{2n}\left(\boldsymbol{z}-\boldsymbol{\xi}^\perp\right). && (135)
\end{aligned}$$

□

We compute the Wigner function of the code space projector of a GKP code defined with primal basis M

$$\begin{aligned}
W_{\Pi_{GKP}}(\boldsymbol{x}) &= \int_{\mathbb{R}^{2n}} d\boldsymbol{\eta}\, e^{-i2\pi\boldsymbol{x}^T J\boldsymbol{\eta}}\, \mathrm{Tr}\left[D(\boldsymbol{\eta})\,\Pi_{GKP}\right] && (136)\\
&= \sum_{\boldsymbol{\xi}\in\mathcal{L}} e^{i\phi_M(\boldsymbol{\xi})} \int_{\mathbb{R}^{2n}} d\boldsymbol{\eta}\, e^{-i2\pi\boldsymbol{x}^T J\boldsymbol{\eta}}\, \mathrm{Tr}\left[D(\boldsymbol{\eta})\, D(\boldsymbol{\xi})\right] && (137)\\
&= \sum_{\boldsymbol{\xi}\in\mathcal{L}} e^{i\phi_M(\boldsymbol{\xi})} e^{i2\pi\boldsymbol{x}^T J\boldsymbol{\xi}}. && (138)
\end{aligned}$$

When $\phi_M = 0 \mod 2\pi$, such as when the GKP code is defined with symplectically even lattice, we can compute the sum using Lemma 2,

$$W_{\Pi_{GKP}}(\boldsymbol{x}) = \frac{1}{\det(\mathcal{L})} \sum_{\boldsymbol{\xi}^\perp\in\mathcal{L}^\perp} \delta^{2n}\left(\boldsymbol{x}-\boldsymbol{\xi}^\perp\right), \tag{139}$$

such that the Wigner function is precisely given by a generalized Dirac comb over the dual lattice $\mathcal{L}^{\perp}$. When $\phi_M \neq 0 \mod 2\pi$, however, this is no longer true. The above expression will also be substantially different from a mere shift of the Dirac comb over the lattice since ϕ_M is generally a *quadratic* function. We can express it compactly using the general *Riemann theta function* [51]

$$\vartheta\left(\boldsymbol{u} \,|\, F\right) = \sum_{\boldsymbol{m}\in\mathbb{R}^{2n}} e^{i2\pi\left(\frac{1}{2}\boldsymbol{m}^T F \boldsymbol{m} + \boldsymbol{m}^T \boldsymbol{u}\right)}, \tag{140}$$

as

$$W_{\Pi_{GKP}}\left(\boldsymbol{x}\right) = \frac{1}{\det\left(\mathcal{L}\right)} \vartheta\left(-MJ\boldsymbol{x} \,|\, A_{◣}\right). \tag{141}$$

Note, however, that $A_{◣}$ does not have positive definite imaginary part so that, similar to the Dirac comb above, this expression does not converge. This is a general artefact of exact GKP states discussed here which are unphysical as presented and, in practice, will be endowed with particular regularizing functions depending on the concrete physical implementation. When such regularization is present, the corresponding sum can be expressed using the following generalization of Poisson resummation.

Lemma 3 (Poisson resummation)**.** *Let $\mathcal{L} \subseteq \mathbb{R}^{2n}$ be a lattice with symplectic dual $\mathcal{L}^{\perp}$, we have for all Schwartz functions $f : \mathbb{R}^{2n} \to \mathbb{C}$,*

$$\sum_{\boldsymbol{\xi}\in\mathcal{L}} f\left(\boldsymbol{\xi}\right) = \frac{1}{\det\left(\mathcal{L}\right)} \sum_{\boldsymbol{\xi}^{\perp}\in\mathcal{L}^{\perp}} \hat{f}\left(\boldsymbol{\xi}^{\perp}\right), \tag{142}$$

where

$$\hat{f}\left(\boldsymbol{x}\right) = \int_{\mathbb{R}^{2n}} d\boldsymbol{y}\, f\left(\boldsymbol{y}\right) e^{i2\pi \boldsymbol{y}^T J \boldsymbol{x}} \tag{143}$$

is the symplectic Fourier transform of f.

Proof. We proof this fact following the analogous proof using the usual Fourier transform provided in [41]. First define

$$F\left(\boldsymbol{z}\right) = \sum_{\boldsymbol{\xi}\in\mathcal{L}} f\left(\boldsymbol{z} + \boldsymbol{\xi}\right). \tag{144}$$

This function is invariant under transformations $\boldsymbol{z} \mapsto \boldsymbol{z} + \mathcal{L}$, such that it descends to a function on $\mathbb{R}^{2n}/\mathcal{L}$ and has Fourier expansion

$$F\left(\boldsymbol{z}\right) = \sum_{\boldsymbol{y}\in\mathcal{L}^{\perp}} \hat{F}\left(\boldsymbol{y}\right) e^{i2\pi \boldsymbol{y}^T J \boldsymbol{z}}, \tag{145}$$

where

$$\hat{F}\left(\boldsymbol{y}\right) = \frac{1}{\det\left(\mathcal{L}\right)} \int_{\mathbb{R}^{2n}/\mathcal{L}} d\boldsymbol{x}\, F\left(\boldsymbol{x}\right) e^{i2\pi \boldsymbol{x}^T J \boldsymbol{y}}. \tag{146}$$

Using lemma 2 the correctness of the Fourier expansion can be straightforwardly verified. Let $\mathcal{P} \sim \mathbb{R}^{2n}/\mathcal{L}$ be a fundamental domain of $\mathcal{L}$. We compute for $\boldsymbol{y} \in \mathcal{L}^{\perp}$ (such that $e^{i2\pi \boldsymbol{x}^T J \boldsymbol{y}}$ is well defined on $\boldsymbol{x} \in \mathbb{R}^{2n}/\mathcal{L}$)

$$\begin{aligned}
\det\left(\mathcal{L}\right) \hat{F}\left(\boldsymbol{y}\right) &= \sum_{\boldsymbol{\xi}\in\mathcal{L}} \int_{\boldsymbol{x}\in\mathcal{P}} d\boldsymbol{x}\, f\left(\boldsymbol{x} + \boldsymbol{\xi}\right) e^{i2\pi \boldsymbol{x}^T J \boldsymbol{y}} && (147)\\
&= \sum_{\boldsymbol{\xi}\in\mathcal{L}} \int_{\boldsymbol{x}\in\mathcal{P}-\boldsymbol{\xi}} d\boldsymbol{x}\, f\left(\boldsymbol{x}\right) e^{i2\pi \boldsymbol{x}^T J \boldsymbol{y}} && (148)\\
&= \hat{f}\left(\boldsymbol{y}\right), && (149)
\end{aligned}$$

such that we have

$$F\left(\boldsymbol{z}\right) = \frac{1}{\det\left(\mathcal{L}\right)} \sum_{\boldsymbol{\xi}^{\perp}\in\mathcal{L}^{\perp}} \hat{f}\left(\boldsymbol{y}\right) e^{i2\pi \boldsymbol{y}^T J \boldsymbol{z}}. \tag{150}$$

Taking $z = 0$ completes the proof. □

We use Lemma 3 to compute the Wigner function of a GKP code, where we include generic Gaussian regularization parametrized by $\epsilon > 0$.

$$
\begin{aligned}
W_{\Pi_{GKP}}(\boldsymbol{x}) &= \left(\frac{\epsilon}{\pi}\right)^n \sum_{\boldsymbol{\xi}\in\mathcal{L}} e^{-\epsilon\boldsymbol{\xi}^T\boldsymbol{\xi}} e^{i\phi_M(\boldsymbol{\xi})} e^{i2\pi\boldsymbol{x}^T J\boldsymbol{\xi}} && (151)\\
&= \left(\frac{\epsilon}{\pi}\right)^n \sum_{\boldsymbol{\xi}\in\mathcal{L}} e^{-\boldsymbol{\xi}^T\left(\epsilon I - iM^{-1}A_{\text{◣}}M^{-T}\right)\boldsymbol{\xi} + i2\pi\boldsymbol{x}^T J\boldsymbol{\xi}} && (152)\\
&= \frac{1}{\det(\mathcal{L})}\frac{\epsilon^n}{\sqrt{\det(B)}} \sum_{\boldsymbol{\xi}^\perp\in\mathcal{L}^\perp} e^{-\pi^2\left(\boldsymbol{x}-\boldsymbol{\xi}^\perp\right)^T JB^{-1}J^T\left(\boldsymbol{x}-\boldsymbol{\xi}^\perp\right)}. && (153)
\end{aligned}
$$

where we have defined $B = \epsilon I - iM^{-1}A_{\text{◣}}M^{-T}$. We observe that for $A_{\text{◣}} = 0$ in the limit $\epsilon \to 0$ we again obtain eq. (139). However, in general for odd $A_{\text{◣}}$ the Wigner function is modified non-trivially. In this example the regularization was chosen generically without reference to a clear physical motivation other than that physical states need to be normalizable. It would be interesting to further study how the structure of a physical model of the code space changes with $A_{\text{◣}}$ in the Wigner representation for alternative regularization schemes such as ones discussed in ref. [52].

B GKP Codes from degenerate lattices

In this section we briefly discuss how degenerate symplectic lattices can be used for encoding continuous degrees of freedom into a collection of n oscillators, following the definition and analysis in [23, 15]. To simplify the notation we will work in the convention where $\hat{\boldsymbol{x}} = (\hat{q}_1, \hat{p}_1, \ldots, \hat{q}_n, \hat{p}_n,)$ and the symplectic form is $J = I_n \otimes J_2$. We encode a state vector $|\psi\rangle \in \mathcal{H}^{\otimes k}$ describing k oscillator modes into $\mathcal{H}^{\otimes n}$ by appending it with the ancillary state vector $|A\rangle = |GKP\rangle^{\otimes r}$, where $r = n - k$, and acting on the composite state with a Gaussian unitary U_S (see fig. 5). Each state vector $|GKP\rangle$ is the unique code state of the GKP stabilizer group described by $M_{GKP} = I_2$, the symplectic self-dual square lattice, such that the overall ancillary state vector $|A\rangle$ is stabilized by a GKP stabilizer group $\mathcal{S}_A$ associated to a lattice generated by $M_A = I_{2r}$ which we denote by $\mathcal{L}_A = \mathcal{L}(M_A)$. The code state is hence described by

$$
\begin{aligned}
|\overline{\psi}\rangle &= U_{S^{-1}}\left(|\psi\rangle \otimes |A\rangle\right)\\
&= U_{S^{-1}}\left(|\psi\rangle \otimes D(\boldsymbol{s})|A\rangle\right)\\
&= U_{S^{-1}} D(\boldsymbol{0}_{2k} \oplus \boldsymbol{s}) U_{S^{-1}}^\dagger |\overline{\psi}\rangle\\
&= D\left(S(\boldsymbol{0}_{2k} \oplus \boldsymbol{s})\right)|\overline{\psi}\rangle \qquad \forall \boldsymbol{s} \in \mathcal{L}_A && (154)
\end{aligned}
$$

and is stabilized by displacements associated to the lattice $\mathcal{L}_{osc}$ generated by the rows of

$$M_{osc} = \left[0_{2k} \oplus M_A\right] S^T, \tag{155}$$

where we used 0_{2k} to denote the $2k \times 2k$ zero-matrix. We denote the corresponding stabilizer group by $\mathcal{S}_{osc}$. It is immediate that M_{osc} is not a full rank matrix, such that there will be a continuum of solutions $\boldsymbol{x}$ to

$$MJ\boldsymbol{x} = 0 \mod 1. \tag{156}$$

In particular, we can obtain logical displacements by propagating a logical displacement $D(\boldsymbol{\alpha})$ on the data mode through the encoding unitary

$$U_{S^{-1}}\left(D(\boldsymbol{\alpha})|\psi\rangle \otimes |A\rangle\right) = D\left(S(\boldsymbol{\alpha} \oplus \boldsymbol{0}_{2r})\right)|\overline{\psi}\rangle. \tag{157}$$

Finally, we obtain that the centralizer of the stabilizer group $\mathcal{S}_{osc}$ is given by displacements within

$$\mathcal{L}_{osc}^\perp = \left\{S(\boldsymbol{\alpha} \oplus \boldsymbol{0}_{2r}) + \boldsymbol{\xi} \,\middle|\, \boldsymbol{\alpha} \in \mathbb{R}^{2k},\ \boldsymbol{\xi} \in \mathcal{L}_{osc}, \right\}. \tag{158}$$

In contrast to the encoding of qubits and qudits, where logical operations are encoded in the discrete collection of cosets of the quotient lattice, the oscillator-to-oscillator encoding does not immediately inherit error correction properties from an underlying notion of (euclidean) distance, where displacements of amplitude below a certain threshold are guaranteed to not harm the logical information. Instead, the protection stems from that fact that the encoding unitary can be tailored to correlate error displacements between the data- and the ancillary modes such that measuring the stabilizers $\mathcal{S}_A$ of the ancillary system allows to diagnose the error that incurred on the data modes. That is, when the error is promised to be sufficiently small, the modular quadrature information gained from measuring $\mathcal{S}_A$ reveals the actual displacement error value. Let $\mathcal{N}$ denote the Gaussian displacement channel. We then have that $\rho_0 = |\psi\rangle\langle\psi| \otimes |A\rangle\langle A|$ undergoes the evolution

$$\begin{aligned}\rho_0 \mapsto U^\dagger_{S^{-1}} \int d^{2n}\boldsymbol{x}\, e^{-\frac{1}{2}\boldsymbol{x}^T\left(\frac{I}{\sigma^2}\right)\boldsymbol{x}} D(\boldsymbol{x})\, U_{S^{-1}}\rho_0 U^\dagger_{S^{-1}} D(\boldsymbol{x})^\dagger\, U_{S^{-1}} \\ = \int d^{2n}\boldsymbol{x}\, e^{-\frac{1}{2}\boldsymbol{y}^T\left(\frac{S^TS}{\sigma^2}\right)\boldsymbol{y}} D(\boldsymbol{y})\,\rho_0 D(\boldsymbol{y})^\dagger\,, \qquad (159)\end{aligned}$$

that is ρ_0 effectively undergoes a Gaussian displacement channel with covariance $\Sigma = \sigma^2\left(S^TS\right)^{-1}$. This scheme can be understood as generalization and QEC application of the displacement-sensor setup developed in [53]. For concrete examples and further details on the oscillator-to-oscillator encoding we refer the reader to ref. [23] and for analysis of the possibility of a threshold to ref. [15].

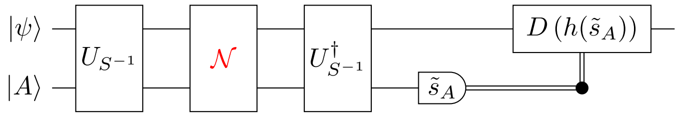


Figure 5: Oscillator-to-oscillator encoding of ref. [23]. $\tilde{s}_A$ denotes the Eigenvalues obtained from measuring the generating set of stabilizers in S_A and $h(\cdot)$ an estimator function that estimates the logical shift based on $\tilde{s}_A$.

To go beyond the encoding of either continuous or discrete information into a collection of oscillators, we remark that this scheme can be extended to host both continuous and discrete information. One way to do so is by choosing as input ancillary state vector $|A\rangle$ a code state of a non-trivial GKP code described by a lattice $\mathcal{L}(M_A)$ with $|\det\left(M_A J_{2r} M_A^T\right)| > 1$ such as one discussed in the main body of this work. Here, the ancillary state on the second register $\mathcal{H}^{\otimes r}$ would host all the discrete information with the addition that the errors applied to it may be (depending on the choice of S) strongly correlated with the errors on the first register $\mathcal{H}^{\otimes k}$ that hosts continuous information. The stabilizer measurements $\mathcal{S}_A$ would hence also reveal information about the exact displacement the first register $\mathcal{H}^{\otimes k}$ has undergone where $\lambda_1\left(\mathcal{L}_A^\perp\right)$ sets the scale for the variance up to which the modular measurement outcomes $\tilde{s}_A$ still accurately estimate the exact shift on the first register (it needs to lie inside the dual Voronoi cell $\mathcal{V}^\perp$ of $\mathcal{L}_A^\perp$). As far as we know (see, e.g., Theorem 3) this does not need to limit the euclidean distance Δ with which the discrete information is protected. We note that this hybrid scheme is equivalent to a setup where a GKP encoded discrete variable state is considered part of the CV input $|\psi\rangle$. A more in-depth analysis of such hybrid scheme extends beyond the scope of this work and is therefore deferred to future work.

References

[1] D. Gottesman, A. Kitaev, and J. Preskill. "Encoding a qubit in an oscillator". Phys. Rev. A **64**, 012310 (2001).

[2] B. M. Terhal, J. Conrad, and C. Vuillot. "Towards scalable bosonic quantum error correction". Quantum Science and Technology **5**, 043001 (2020).

[3] A. L. Grimsmo and S. Puri. "Quantum error correction with the Gottesman-Kitaev-Preskill code". PRX Quantum **2**, 020101 (2021).

[4] J. E. Bourassa, R. N. Alexander, M. Vasmer, A. Patil, I. Tzitrin, T. Matsuura, D. Su, B. Q. Baragiola, S. Guha, G. Dauphinais, and et al. "Blueprint for a scalable photonic fault-tolerant quantum computer". Quantum **5**, 392 (2021).

[5] S. Bartolucci, P. Birchall, H. Bombin, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kieling, N. Nickerson, M. Pant, F. Pastawski, T. Rudolph, and C. Sparrow. “Fusion-based quantum computation” (2021). arXiv:2101.09310.

[6] K. Noh, C. Chamberland, and F. G. S. L. Brandão. “Low-overhead fault-tolerant quantum error correction with the surface-GKP code”. PRX Quantum **3**, 010315 (2022).

[7] C. Flühmann, T. L. Nguyen, M. Marinelli, V. Negnevitsky, K. Mehta, and J. P. Home. “Encoding a qubit in a trapped-ion mechanical oscillator”. Nature **566**, 513–517 (2019).

[8] P. Campagne-Ibarcq, A. Eickbusch, S. Touzard, E. Zalys-Geller, N. E. Frattini, V. V. Sivak, P. Reinhold, S. Puri, S. Shankar, R. J. Schoelkopf, and et al. “Quantum error correction of a qubit encoded in grid states of an oscillator”. Nature **584**, 368–372 (2020).

[9] K. Fukui, A. Tomita, and A. Okamoto. “Analog quantum error correction with encoding a qubit into an oscillator”. Phys. Rev. Lett. **119**, 180507 (2017).

[10] C. Vuillot, H. Asasi, Y. Wang, L. P. Pryadko, and B. M. Terhal. “Quantum error correction with the toric Gottesman-Kitaev-Preskill code”. Phys. Rev. A **99**, 032344 (2019).

[11] K. Noh and C. Chamberland. “Fault-tolerant bosonic quantum error correction with the surface–Gottesman-Kitaev-Preskill code”. Phys. Rev. A **101**, 012316 (2020).

[12] L. Hänggli, M. Heinze, and R. König. “Enhanced noise resilience of the surface–gottesman-kitaev-preskill code via designed bias”. Phys. Rev. A **102** (2020).

[13] J. W. Harrington. “Analysis of quantum error-correcting codes: Symplectic lattice codes and toric codes”. PhD thesis. California Institute of Technology. (2004).

[14] J. Harrington and J. Preskill. “Achievable rates for the Gaussian quantum channel”. Phys. Rev. A **64**, 062301 (2001).

[15] L. Hänggli and R. König. “Oscillator-to-oscillator codes do not have a threshold”. IEEE Transactions on Information Theory **68**, 1068–1084 (2022).

[16] J. Conway and N. Sloane. “Sphere packings, lattices and groups”. Volume 290. Springer, New York, NY (1988).

[17] P. Shor and R. Laflamme. “Quantum analog of the macwilliams identities for classical coding theory”. Phys. Rev. Lett. **78**, 1600–1602 (1997).

[18] E. M. Rains. “Quantum weight enumerators”. IEEE Trans. Inf. Th. **44**, 1388–1394 (1998).

[19] D. Micciancio. “Cse 206a: Lattice algorithms and applications”. url: http://cseweb.ucsd.edu/classes/wi10/cse206a/.

[20] D. E. Gottesman. “Stabilizer codes and quantum error correction”. PhD thesis. California Institute of Technology. (1997).

[21] Arvind, B. Dutta, N. Mukunda, and R. Simon. “The real symplectic groups in quantum mechanics and optics”. Pramana **45**, 471–497 (1995).

[22] K. Duivenvoorden, B.M. Terhal, and D. Weigand. “Single-mode displacement sensor”. Phys. Rev. A **95**, 012305 (2017).

[23] K. Noh, S. M. Girvin, and L. Jiang. “Encoding an oscillator into many oscillators”. Phys. Rev. Lett. **125**, 080503 (2020).

[24] N. P. Breuckmann. “Homological quantum codes beyond the toric code”. PhD thesis. RWTH Aachen University. (2017).

[25] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary” (1998). arXiv:quant-ph/9811052.

[26] Y. Tomita and K. M. Svore. “Low-distance surface codes under realistic quantum noise”. Phys. Rev. A **90** (2014).

[27] D. Micciancio and B. Warinschi. “A linear space algorithm for computing the Hermite normal form”. In Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation. Page 231–236. ISSAC '01New York, NY, USA (2001). Association for Computing Machinery.

[28] D. Micciancio. “Improving lattice based cryptosystems using the Hermite normal form”. In Lecture Notes in Computer Science. Pages 126–145. Springer Berlin Heidelberg (2001).

[29] C. Hermite. “Sur l'introduction des variables continues dans la théorie des nombres.”. J. Reine Angew. Math **1851**, 191–216 (1851).

[30] A. Lenstra, H. Lenstra, and L. Lovász. “Factoring polynomials with rational coefficients”. Mathematische Annalen **261**, 515–534 (1982).

[31] K. Chandrasekaran, V. Gandikota, and E. Grigorescu. "Deciding orthogonality in Construction-A Lattices". In P. Harsha and G. Ramalingam, editors, 35th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2015). Volume 45 of Leibniz International Proceedings in Informatics (LIPIcs), pages 151–162. Dagstuhl, Germany (2015). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[32] "Lattice completion algorithm". `https://github.com/frarzani/GKP-lattice-reduction`.

[33] A. Y. Kitaev. "Fault-tolerant quantum computation by anyons". Ann. Phys. **303**, 2–30 (2003).

[34] K. Noh, V. V. Albert, and L. Jiang. "Quantum capacity bounds of Gaussian thermal loss channels and achievable rates with Gottesman-Kitaev-Preskill codes". IEEE Trans. Inf. Th. **65**, 2563–2582 (2019).

[35] S. Glancy and E. Knill. "Error analysis for encoding a qubit in an oscillator". Phys. Rev. A **73** (2006).

[36] I. Tzitrin, J. E. Bourassa, N. C. Menicucci, and K. K. Sabapathy. "Progress towards practical qubit computation using approximate Gottesman-Kitaev-Preskill codes". Phys. Rev. A **101**, 032315 (2020).

[37] J. Conrad. "Twirling and Hamiltonian engineering via dynamical decoupling for Gottesman-Kitaev-Preskill quantum computing". Phys. Rev. A **103** (2021).

[38] P. Sarnak and P. Buser. "On the period matrix of a Riemann surface of large genus (with an Appendix by J. H. Conway and N. J. A. Sloane)". Inventiones mathematicae **117**, 27–56 (1994). url: http://eudml.org/doc/144207.

[39] S. L. Braunstein. "Squeezing as an irreducible resource". Phys. Rev. A **71**, 055801 (2005).

[40] L. Babai. "On lovász' lattice reduction and the nearest lattice point problem". Combinatorica **6**, 1–13 (1986).

[41] N. D. Elkies. "Rational lattices and their theta functions". url: http://people.math.harvard.edu/ elkies/M272.19/index.html.

[42] E. Knill, R. Laflamme, and L. Viola. "Theory of quantum error correction for general noise". Phys. Rev. Lett. **84**, 2525–2528 (2000).

[43] Y. Tian, X. Zhu, and R. Sun. "Modular form approach to solving lattice problems". In T. V. Gopal, M Agrawal, A Li, and S. B. Cooper, editors, Theory and Applications of Models of Computation. Pages 401–421. Cham (2014). Springer International Publishing.

[44] S. Bravyi, M. Suchara, and A. Vargo. "Efficient algorithms for maximum likelihood decoding in the surface code". Phys. Rev. A **90** (2014).

[45] T. Gannon. "Lattices and theta functions". PhD thesis. McGill University (Canada). (1991). url: escholarship.mcgill.ca/concern/theses/rr171z009.

[46] J.-P. Tillich and G. Zemor. "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the block length". IEEE Trans. Inf. Th. **60**, 1193–1202 (2014).

[47] S. Bravyi and M. B. Hastings. "Homological product codes". In Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing. Page 273–282. STOC '14New York, NY, USA (2014). Association for Computing Machinery.

[48] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. "Topological quantum memory". J. Math. Phys. **43**, 4452–4505 (2002).

[49] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T.C. Ralph, J.H. Shapiro, and S. Lloyd. "Gaussian quantum information". Rev. Mod. Phys. **84**, 621–669 (2012).

[50] J. Eisert and M. B. Plenio. "Introduction to the basics of entanglement theory in continuous-variable systems". Int. J. Quant. Inf. **1**, 479 (2003).

[51] W. Wirtinger. "Untersuchungen über Thetafunctionen.". B. G. Teubner. (1895).

[52] L. J. Mensen, B. Q. Baragiola, and N. C. Menicucci. "Phase-space methods for representing, manipulating, and correcting Gottesman-Kitaev-Preskill qubits". Phys. Rev. A **104**, 022408 (2021).

[53] K. Duivenvoorden, B. M. Terhal, and D. Weigand. "Single-mode displacement sensor". Phys. Rev. A **95**, 012305 (2017).