Kerdock Codes Determine Unitary 2-Designs

Trung Can*, Narayanan Rengaswamy*, Robert Calderbank, and Henry D. Pfister

Abstract—The non-linear binary Kerdock codes are known to be Gray images of certain extended cyclic codes of length $N = 2^m$ over $\mathbb{Z}_4$. We show that exponentiating these $\mathbb{Z}_4$-valued codewords by $i = \sqrt{-1}$ produces stabilizer states, that are quantum states obtained using only Clifford unitaries. These states are also the common eigenvectors of commuting Hermitian matrices forming maximal commutative subgroups (MCS) of the Pauli group. We use this quantum description to simplify the derivation of the classical weight distribution of Kerdock codes. Next, we organize the stabilizer states to form $N + 1$ mutually unbiased bases and prove that automorphisms of the Kerdock code permute their corresponding MCSs, thereby forming a subgroup of the Clifford group. When represented as symplectic matrices, this subgroup is isomorphic to the projective special linear group $\mathrm{PSL}(2,N)$. We show that this automorphism group acts transitively on the Pauli matrices, which implies that the ensemble is Pauli mixing and hence forms a unitary 2-design. The Kerdock design described here was originally discovered by Cleve et al. (2016), but the connection to classical codes is new which simplifies its description and translation to circuits significantly. Sampling from the design is straightforward, the translation to circuits uses only Clifford gates, and the process does not require ancillary qubits. Finally, we also develop algorithms for optimizing the synthesis of unitary 2-designs on encoded qubits, i.e., to construct logical unitary 2-designs. Software implementations are available at https://github.com/nrenga/symplectic-arxiv18a, which we use to provide empirical gate complexities for up to 16 qubits.

Index Terms—Heisenberg-Weyl group, Pauli group, quantum computing, Clifford group, symplectic geometry, Kerdock codes, Delsarte-Goethals codes, Gray map, stabilizer states, mutually unbiased bases, unitary $t$-designs

I. INTRODUCTION

Quantum computers promise enormous computational speedups over the best classical supercomputers in certain problems. It has been established that even constant-depth quantum circuits provide an advantage over classical computation [2]. Today these devices are moving out of the lab and becoming generally programmable [3]. Since quantum computers are noisy, they will most likely employ quantum error-correcting codes (QECCs) to ensure reliability of computation. Classical error-correcting codes inspired the discovery of the first QECC by Shor [4], the development of CSS codes by Calderbank, Shor and Steane [5], [6], and the development of stabilizer codes [7], [8] by Calderbank, Rains, Shor and Sloane and by Gottesman. A QECC protects $m-k$ logical qubits by embedding them in a system comprising $m$ physical qubits. In fault-tolerant computation, any desired operation on the $m-k$ logical (protected) qubits must be implemented as a physical operation on the $m$ physical qubits that preserves the code space. The most common QECCs are stabilizer QECCs, that are derived from commutative subgroups of the Heisenberg-Weyl group $\mathrm{HW}_N$ (also known as the Pauli group), and the simplest quantum circuits are composed of unitary operators from the Clifford group $\mathrm{Cliff}_N$ (see Section III for formal definitions). Note that $N = 2^m$ throughout this paper. The Clifford group normalizes $\mathrm{HW}_N$, so each element of $\mathrm{Cliff}_N$ induces an automorphism of $\mathrm{HW}_N$ under conjugation. Since $\mathrm{HW}_N$ elements can be efficiently represented as binary length-$2m$ matrices, Clifford operators can be efficiently represented as $2m \times 2m$ binary symplectic matrices that preserve the group structure of $\mathrm{HW}_N$ under the binary mapping. A central theme in this paper is that beyond inspiring the construction of QECCs, interactions between the classical and quantum worlds still prove mutually beneficial for other purposes, especially via the binary representations of $\mathrm{HW}_N$ and $\mathrm{Cliff}_N$.

Prior to running applications, a quantum computer needs to be assessed for its quality, in terms of the fidelity of the quantum operations executed on the system. The randomized benchmarking protocol, introduced by Emerson et al. [9], [10], is a well-established scheme that estimates the fidelity of the noise in the system by twirling the underlying channel through a randomized sequence of gates and calculating the fidelity of the resulting depolarizing channel. The depolarizing channel can be interpreted as the quantum analogue of the binary symmetric channel in classical communication theory. Since the fidelity is shown to be an invariant of this twirling process, the protocol indeed estimates the fidelity of the actual noise (under some additional assumptions). This protocol works on the physical operations in the computer, so the fidelity estimates then need to be translated into the quality of the logical operations (on the protected qubits) for an error-corrected quantum computer. Since this translation might not be reliable, the procedure has recently been extended to the logical randomized benchmarking protocol [11] that directly estimates the fidelity of the logical operations more reliably.

Randomized benchmarking requires that the sequence of gates be sampled from an ensemble of unitaries that form a unitary 2-design. A unitary $t$-design is an ensemble of unitaries, bestowed with a probability distribution, that ap-

*These two authors contributed equally to this work. This work was presented in part at the 2019 IEEE International Symposium on Information Theory [1].

T. Can is currently with the Department of Mathematics, California Institute of Technology, Pasadena, CA 91125, USA. Most of this work was conducted while he was with the Department of Mathematics, Duke University, Durham, North Carolina 27708, USA. Email: tcan@caltech.edu

N. Rengaswamy, R. Calderbank and H. D. Pfister are with the Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, USA. Email: {nrenga, recalderbank, henry.pfister}@duke.edu

This work was supported in part by the National Science Foundation (NSF) under Grant Nos. 1718494 and 1908730. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these sponsors.

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.
proximates the Haar distribution on the unitary group up to the $t$-th moment. Section [VI] makes this precise by discussing linear maps called twirls over the full unitary ensemble and the finite ensemble. When the two linear maps coincide the finite ensemble is a unitary $t$-design. It is easy to analyze protocols that randomly sample unitary matrices with respect to the Haar measure, but such sampling is infeasible, hence the interest in finite ensembles of unitary matrices that approximate the Haar distribution. Cliff$_N$ is known to be a unitary 3-design [12], and the proof by Webb involves the concepts of Pauli mixing and Pauli 2-mixing, which we introduce in Section [VI]. However, Cliff$_N$ has size $2^{m^2+2m} \prod_{j=1}^{m} (4^j - 1)$ (up to scalars $e^{2\pi i\theta}, \theta \in \mathbb{R}$) which is much larger than the lower bound of $\approx 2^{4m}$ for any unitary 2-design, established by Gross, Audenaert and Eisert [13], and by Roy and Scott [14]. They also discuss the existence of Clifford ensembles that saturate this bound. Although random circuits are known to be exact or approximate unitary 2-designs [15], [16], deterministic constructions of such ensembles facilitate practical realizations. Cleve et al. [17] have recently found an explicit subgroup of Cliff$_N$ that is a unitary 2-design. The central contribution of this paper is to use the classical Kerdock codes to simplify the construction of this quantum unitary 2-design and its translation to circuits (see Section [VI]).

There are many other applications of unitary 2-designs. The first is quantum data hiding (the LOCC model described in [18]), where the objective is to hide classical information from two parties who each share part of the data but are only allowed to perform local operations and classical communication. The data hiding protocol can be implemented by sampling randomly from the full unitary group but it is sufficient to sample randomly from a unitary 2-design. Other applications of unitary 2-designs are the fidelity estimation of quantum channels [19], quantum state and process tomography [20], and more recently minimax quantum state estimation [21]. In quantum information theory, they have been used extensively in the analysis of decoupling [14], [16], [22], [23].

The next section discusses the flow of ideas in the rest of the paper, and points to related work and applications.

II. MAIN IDEAS AND DISCUSSION

The first result in the paper is to use the (quantum) commutative subgroups of HW$_N$ to simplify the derivation of weight distributions for the (classical) Kerdock codes.

Recall that Hermitian matrices that commute can be simultaneously diagonalized. The Kerdock and Delsarte-Goethals binary code are unions of cosets of the first order Reed-Muller (RM) code RM$(1, m)$, the cosets are in one-to-one correspondence with symplectic forms, and the weight distribution of a coset is determined by the rank of the symplectic form (see [24] Chapter 15) for more details). Section [IV-A] reviews the $\mathbb{Z}_4$ description of these codes given in [25]. Section [IV-B] shows that when we exponentiate codewords in these cosets of RM$(1, m)$ we obtain a basis of common eigenvectors for a maximal commutative subgroup of Hermitian Pauli matrices. The operators that project onto individual lines in a given eigenbasis are invariants of the corresponding subgroup [26], and the distance distribution of the code is determined by the inner product of pairs of such eigenvectors (see [Section IV] Lemma[10]). When calculating weight distributions, this property makes it possible to avoid using Dickson’s Theorem [24] Chapter 15] to choose an appropriate representation of the symplectic form. The correspondence between classical and quantum worlds simplifies the calculation (of the weight distribution) given in [24] significantly. In order to demonstrate this simplicity, we provide the proof of the weight distribution here. For most parts of the proof, we only need a brief description of the Kerdock set, Kerdock code, and the Gray map. See Section [IV] for formal definitions, constructions, and results used in the following proof.

In [27], the Kerdock set $P_k(m)$ is defined to be a collection of $N$ binary $m \times m$ symmetric matrices that is characterized by the following properties: it is closed under binary addition, and if $P, Q \in P_k(m)$ are distinct, then $P + Q$ is non-singular. The codewords of the length-$N$ $\mathbb{Z}_4$-linear Kerdock code $K(m)$ can be expressed as $[xP \pm 2wz + \kappa]$ (mod 4), where $x \in \mathbb{F}_2^m$ chooses the symbol index in a codeword, and the codeword is determined by the choice of $w \in \mathbb{F}_2^m$, $P \in P_k(m)$ and $\kappa \in \mathbb{Z}_4 = \{0, 1, 2, 3\}$. One can obtain complex vectors of quaternary phases $\{1, i, -1, -i\}$ by raising $i = \sqrt{-1}$ to the integer powers in the $(\mathbb{Z}_4$-valued) codeword vector. We will refer to this operation as exponentiating the codeword by $i$. The Gray map is an isometry from (length-$N$ vectors of quaternary phases, Euclidean metric) to $(\mathbb{Z}_2^N, \text{Hamming metric})$ defined by $g(1) \triangleq 00, g(i) \triangleq 01, g(-1) \triangleq 11, \text{and } g(-i) \triangleq 10$. Note that the domain of the map can equivalently be taken as $\mathbb{Z}_4$ in which case the metric is the Lee metric (defined in Section [IV-B]). The codewords of the binary non-linear Kerdock code of length $2^{m+1}$ are obtained as Gray images of the $\mathbb{Z}_4$-valued codewords described above.

**Theorem 1:** Let $m$ be odd. The weight distribution $A_i, i = 0, \ldots, 2^m+1$ of the classical binary Kerdock code of length $2^{m+1}$ is as follows.

| $i$ | $2^m - 2^{(m-1)/2}$ | $2^m$ | $2^m + 2^{(m-1)/2}$ | $2^{m+1}$ |
|-----|----------------|------|----------------|----------|
| $A_i$ | 1 | $2^m + 1 - 2^{m+1}$ | $2^{m+2} - 2$ | $2^{2m+1} - 2^{m+1}$ | 1 |

**Proof:** We explicitly use Lemma[10] and Corollary[16] to calculate the weight distribution. Fix a vector $v$ of quaternary phases obtained by exponentiating a codeword $c_v = [xP_2x^T + 2wz + 2\kappa] \in \mathbb{F}_2^m$ in the $\mathbb{Z}_4$-linear Kerdock code. Consider any vector $u$ of quaternary phases obtained by exponentiating a second codeword $c_u = [xP_1x^T + 2wz + 2\kappa] \in \mathbb{F}_2^m$.

(Lemmas [11] and [15] consider $\kappa_1 = \kappa_2 = 0$ so that the eigenvalues are $\pm 1$ (and not in $\{\pm 1, \pm i\}$), but we account for these factors here.)

Let $n_j$ be the number of indices $x \in \mathbb{F}_2^m$ for which $(c_u)x - (c_v)x = j$, for $j \in \mathbb{Z}_4$. Since the Gray map preserves the Lee metric, the Hamming distance between the Gray images of $c_u$ and $c_v$ is $d_H(g(c_u), g(c_v)) = n_1 + 2n_2 + n_3$. Since $n_0 + n_1 + n_2 + n_3 = 2^m$ we simply have to relate
Now we observe three distinct cases for the codeword $c_u - c_v$.

(i) $P_1 = P_2, w_1 = w_2$: If $\kappa_1 - \kappa_2 = 0$ then we have the all-zeros codeword, and if $\kappa_1 - \kappa_2 = 2$ then we have the all-ones codeword. However, if $\kappa_1 - \kappa_2 \in \{1, 3\}$ then $n_0 - n_2 = 0$ and this determines two codewords of weight $2^m$ (more precisely, at distance $2^m$ from $c_u$).

(ii) $P_1 = P_2, w_1 \neq w_2$: From Corollary \ref{corollary} irrespective of $\kappa_1, \kappa_2$, we have $\langle u, v \rangle = 0$, which implies $n_0 - n_2 = 0$ and hence the distance is $2^m$. This determines another $(2^m - 1)^2 = 2^{m+2} - 4$ codewords of weight $2^m$.

(iii) $P_1 \neq P_2$: From Corollary \ref{corollary} we have $\|u, v\|^2 = 2^m$, which implies $(n_0 - n_2)^2 + (n_1 - n_3)^2 = 2^m$. Since $m$ is odd, and $n_3$ are non-negative integers, direct calculation shows that this means $(n_0 - n_2)^2 = (n_1 - n_3)^2 = 2^{m-1}$ and therefore $n_0 - n_2 = \pm 2^{(m-1)/2}$. More formally, since the Gaussian integers $\mathbb{Z}[i]$ is a unique factorization domain, we have $(n_0 - n_2) + (n_1 - n_3)i = (\pm 1 \pm i)2^{(m-1)/2}$ and this gives weights $2^m \pm 2^{(m-1)/2}$. Thus, we have $2^{2m+2} - 2^{m+2}$ codewords remaining and it is easy to see that the signs occur equally often. Hence there are $2^{2m+1} - 2^{m+1}$ codewords of each weight.

Lemma \ref{lemma} shows that the (normalized) length-$N$ vectors of quaternary phases obtained by exponentiating Pauli, or more generally Delsarte-Goethals, codewords are common eigenvectors of maximal commutative subgroups of $HW_N$. These eigenvectors are called stabilizer states in the quantum information literature \cite{stabilizer} (also see end of Section \ref{section}). They can equivalently be described as those quantum states obtainable by applying Clifford unitaries to the basis state $|0\rangle^{\otimes m} = e_0 \otimes \cdots \otimes e_0 = |1, 0, 0, \ldots, 0\rangle^T$ \cite{Pauli}. Clifford elements act by conjugation on $HW_N$, permuting the maximal commutative subgroups, and fixing the ensemble of stabilizer states (see Section \ref{section}). The Clifford group is strongly symmetric, and it approximates the full unitary group in a way that can be made precise by comparing irreducible representations \cite{Kueng, Gross}. Kueng and Gross \cite{Kueng} have shown that the ensemble of stabilizer states is a complex projective 3-design; given a polynomial of degree at most 3, the integral over the $N$-sphere can be calculated by evaluating the polynomial at stabilizer states, and taking a finite sum. Stabilizer states also find application as measurements in the important classical problem of phase retrieval, where an unknown vector is to be recovered from the magnitudes of the measurements (see Kueng, Zhu and Gross \cite{Kueng}). A third application is unsourced multiple access, where there is a large number of devices (messages) each of which transmits (is transmitted) infrequently. This provides a model for machine-to-machine communication in the Internet-of-Things (IoT), including the special case of radio-frequency identification (RFID), as well as neighbor discovery in ad-hoc wireless networks. Here, Thompson and Calderbank \cite{Thompson} have shown that stabilizer states associated with Delsarte-Goethals codes support a fast algorithm for unsourced multiple access that scales to $2^{100}$ devices (arbitrary 100-bit messages).

Section \ref{section} constructs the $N+1$ eigenbases of $N$ stabilizer states each) determined by the Kerdock code of length $N$ over $\mathbb{Z}_4$, and shows that the corresponding maximal commutative subgroups partition the non-identity Hermitian Pauli matrices. The eigenbases are mutually unbiased, so that unit vectors $u, v$ in different eigenbases satisfy $\|u, v\| = N^{-\frac{1}{2}}$, and hence each eigenbasis looks like noise to the other eigenbases. The Kerdock ensemble of $N(N+1)$ complex lines is extremal; Calderbank et al. \cite{Calderbank} have shown that any collection of unit vectors for which pairwise inner products have absolute value 0 or $N^{-\frac{1}{2}}$ has size at most $N^2 + N$, and that any extremal example must be a union of eigenbases. The group of Clifford symmetries of this ensemble, represented as binary symplectic matrices, is shown to be isomorphic to the projective special linear group $\text{PSL}(2, N)$, and hence its size is $(N+1)(N-1) = 2^{2m} - 2^m$. We note that the Kerdock ensemble also appears in the work of Tirkkonen et al. \cite{Tirkkonen}.

Section \ref{section} defines a graph $\mathbb{H}_N$ whose vertices are labeled by (scalar multiples of) all non-identity (Hermitian) Pauli matrices, where two matrices (vertices) are joined (by an edge) if and only if they commute. This graph is shown to be strongly regular; every vertex has the same degree, and the number of vertices joined to two given vertices depends only on whether the two vertices are joined or not joined. The automorphism group of this graph is the group of binary symplectic matrices $Sp(2m, \mathbb{F}_2)$. A subgroup of $\text{Cliff}_N$ containing $HW_N$ is proven to be Pauli mixing if it acts transitively on vertices, and Pauli 2-mixing if it acts transitively on edges and on non-edges. These properties imply that Pauli mixing ensembles are unitary 2-designs and Pauli 2-mixing ensembles are unitary 3-designs \cite{Pauli}. The Clifford symmetries of the Kerdock ensemble (of stabilizer states), again represented as symplectic matrices, are shown to be transitive on the vertices of $\mathbb{H}_N$ and hence a unitary 2-design. Since the Clifford symmetries include all Hermitian Paulis, in addition to PSL(2, N), the size of the Kerdock unitary 2-design is $N^3 - N^2 \approx 2^{2m}$, which almost saturates the bound by Gross et al. \cite{Gross} discussed above. The next step is to translate these symmetry elements into circuits for, say, randomized benchmarking.

T. C. Can has developed an algorithm \cite{Can} that factors a $2m \times 2m$ binary symplectic matrix into a product of at most 6 elementary symplectic matrices of the type shown in Table \ref{table}. The target symplectic matrix maps the (Hadamard) dual basis $X_N = E([I_m | 0])$, $Z_N = E([0 | I_m])$ (see Section \ref{section}) for notation) to a dual basis $X_N', Z_N'. $ Then, row and column operations by the elementary matrices return $X_N', Z_N'$ to the original pair $X_N, Z_N$ thereby producing a decomposition of the target symplectic matrix.

Section \ref{section} uses this decomposition to simplify the translation of the Kerdock unitary 2-design into circuits. The elementary symplectic matrices appearing in the product can be related to the Bruhat decomposition of the symplectic group $Sp(2m, \mathbb{F}_2)$ (see \cite{Bruhat}). When the algorithm is run in reverse it produces a random Clifford matrix that can serve as an approximation to a random unitary matrix. This is an
instance of the subgroup algorithm [38] for generating uniform random variables. The algorithm has complexity $O(m^3)$ and uses $O(n^2)$ random bits, which is order optimal given the order of the symplectic group $Sp(2m, \mathbb{F}_2)$ (cf. [39]). We note that the problem of selecting a unitary matrix uniformly at random finds application in machine learning (see [40] and the references therein). The algorithm developed by Can is similar to that developed by Jones, Osipov and Rokhlin [41] in that it alternates (partial) Hadamard matrices and diagonal matrices; the difference is that the unitary 3-design property of the Clifford group [42] provides randomness guarantees.

Finally, Section VII constructs logical unitary 2-designs that can be applied in the logical randomized benchmarking protocol of Combes et al. [11]. In prior work [42], we have developed a mathematical framework for synthesizing all physical circuits that implement a logical Clifford operator (on the encoded qubits) for stabilizer codes (up to equivalence classes and ignoring stabilizer freedom). Circuit synthesis is enabled by representing the desired physical equivalence classes and ignoring stabilizer freedom). Circuit synthesis is enabled by representing the desired physical Clifford operator as a $2m \times 2m$ binary symplectic matrix. For an $[m, m-k]$ stabilizer code, every logical Clifford operator is shown to have $2^{k(k+1)/2}$ symplectic solutions, and these are enumerated efficiently using symplectic transvections, thus enabling optimization with respect to a suitable metric. See https://github.com/nrenga/symplectic-arxiv18a for implementations.

It is now well-known that different codes yield efficient (e.g., low-depth) implementations of different logical operators. However, computing environments can change dynamically so that qubits or qubit links might have varying fidelity, and thus low-depth alone might not be desirable. Under such circumstances it is necessary to leverage all degrees of freedom in implementing a logical operator, and a compiler might use the above framework for this purpose. More generally, a compiler might usefully switch between several codes [43] dynamically, depending on the state of the system. Then this algorithm enables the compiler to be able to determine logical operators for a code quickly depending on the user-input circuit (on the protected qubits).

Section VII provides a proof of concept implementation of the Kerdock unitary 2-design on the protected (logical) qubits of the $[6,4,2]$ CSS code using the above logical Clifford synthesis algorithm. The logical randomized benchmarking protocol requires a unitary 2-design on the logical qubits, and Combes et al. use the full Clifford group for this purpose, which is much larger than the Kerdock design as shown above.

In summary, the purpose of this paper is to emphasize that interactions between the classical and quantum domains still prove mutually beneficial, as much as they helped inspire the first QECC more than two decades back. Specifically, we make four main theoretical contributions:

1) Use of quantum concepts to simplify the calculation of classical weight distributions of several families of non-linear binary codes [24], [43]–[50].
2) Elementary description of symmetries of the Kerdock code, and the $N^2 + N$ stabilizer states determined by this code [27], [33], [35], [51].
3) Demonstration that the symmetry group of the Kerdock code is a unitary 2-design and that sampling from it is straightforward. Introduction of elementary methods for translation to circuits without using ancillary qubits.
4) Provide a proof of concept construction for unitary 2-designs on the logical qubits of a stabilizer code [11], [42].

We also provide software implementations of all algorithms, at https://github.com/nrenga/symplectic-arxiv18a. Using this utility, we provide empirical estimates of the gate complexity for circuits obtained from the Kerdock design. We believe this paves the way for employing this design in several applications, specifically in randomized benchmarking [10], [52].

III. THE HEISENBERG-WEYL AND CLIFFORD GROUPS

Quantum error-correcting codes serve to protect qubits involved in quantum computation, and this section summarizes the mathematical framework introduced in [5], [7], [8], [53], and described more completely in [54] and [42]. In this framework for fault-tolerant quantum computation, Clifford operators on the $N$-dimensional complex space afforded by $m$ qubits are represented as $2m \times 2m$ binary symplectic matrices. This is an exponential reduction in size, and the symplectic matrices serve as a binary control plane for the quantum computer.

Remark 2: Throughout the paper, we adopt the convention that all binary vectors are row vectors, and $\mathbb{Z}_4$, real- or complex-valued vectors are column vectors, where $\mathbb{Z}_4$ is the ring of integers modulo 4. The values $i^*$, where $i \triangleq \sqrt{-1}, \kappa \in \mathbb{Z}_4$, are called quaternary phases.

A single qubit is a 2-dimensional Hilbert space, and a quantum state $v$ is a superposition of the two states $e_0 \triangleq [1, 0]^T, e_1 \triangleq [0, 1]^T$ which form the computational basis. Thus $v = \alpha e_0 + \beta e_1$, where $\alpha, \beta \in \mathbb{C}$ satisfy $|\alpha|^2 + |\beta|^2 = 1$ as per the Born rule [55] Chapter 3. The Pauli matrices are

$$I_2, X \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Z \triangleq \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Y \triangleq iXZ = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},$$

where $i \triangleq \sqrt{-1}$ and $I_2$ is the $2 \times 2$ identity matrix [56] Chapter 10. We may express an arbitrary pure quantum state $v$ as $v = (\alpha_0 I_2 + \alpha_1 X + \alpha_2 Z + \alpha_3 Y) e_0$, where $\alpha_i \in \mathbb{R}$. (3)

We describe $m$-qubit states by (linear combinations of) $m$-fold Kronecker products of computational basis states, or equivalently by $m$-fold Kronecker products of Pauli matrices. Given row vectors $a, b \in \mathbb{F}_2^m$ define the $m$-fold Kronecker product

$$D(a, b) \triangleq X^{a_1} Z^{b_1} \otimes \cdots \otimes X^{a_m} Z^{b_m} \in \mathbb{U}_N, \quad N \triangleq 2^m,$$ (4)

where $\mathbb{U}_N$ denotes the group of all $N \times N$ unitary operators. The Heisenberg-Weyl group $HW_N$ (also called the $m$-qubit Pauli group) consists of all operators $\kappa^* D(a, b)$, where $\kappa \in \mathbb{Z}_4 \triangleq \{0, 1, 2, 3\}$. The order is $|HW_N| = 4N^2$ and the center of this group is $< i I_N > \triangleq \{I_N, iI_N, -I_N, -iI_N\}$, where $I_N$ is
the $N \times N$ identity matrix. Multiplication in $HW_N$ satisfies the identity

$$D(a,b)D(a',b') = (-1)^{a'b^T+b'a^T} D(a',b') D(a,b). \tag{5}$$

The standard symplectic inner product in $F_2^{2m}$ is defined as

$$\langle [a,b], [a',b'] \rangle_s \triangleq a'b^T + b'a^T = [a,b] \Omega [a',b']^T, \tag{6}$$

where the symplectic form $\Omega \triangleq \begin{bmatrix} 0 & I_m \\ I_m & 0 \end{bmatrix}$ (see [7], [42]).

Therefore, two operators $D(a,b)$ and $D(a',b')$ commute if and only if $\langle [a,b], [a',b'] \rangle_s = 0$.

The homomorphism $\gamma : HW_N \to F_2^{2m}$ defined by

$$\gamma(g^\kappa D(a,b)) = [a,b] \quad \forall \kappa \in \mathbb{Z}_4 \tag{7}$$

has kernel $\langle I_N \rangle$ and allows us to represent elements of $HW_N$ (up to multiplication by scalars) as binary vectors.

The Clifford group Cliff$_N$ consists of all unitary matrices $g \in \mathbb{C}^{N \times N}$ for which $gD(a,b)g^\dagger \in HW_N$ for all $D(a,b) \in HW_N$, where $g^\dagger$ is the Hermitian transpose of $g$ [54]. Cliff$_N$ is the normalizer of $HW_N$ in the unitary group $U_N$. The Clifford group contains $HW_N$ and its size is $|Cliff_N| = 2^{m^2+2m} \prod_{j=1}^m (4^j - 1)$ (up to scalars $2^{2m^2 \theta}$, $\theta \in \mathbb{R}$) [7]. We regard operators in Cliff$_N$ as physical operators acting on quantum states in $\mathbb{C}^N$, to be implemented by quantum circuits. Every operator $g$ in Cliff$_N$ induces an automorphism of $HW_N$ by conjugation. Note that the inner automorphisms induced by matrices in $HW_N$ preserve every conjugacy class $\{ \pm D(a,b) \}$ and $\{ \pm tD(a,b) \}$, because (5) implies that elements of $HW_N$ either commute or anti-commute. Matrices $D(a,b)$ are symmetric or anti-symmetric according as $ab^T = 0$ or 1, hence the matrix

$$E(a,b) \triangleq t^{ab^T} D(a,b) \tag{8}$$

is Hermitian. Note that $E(a,b)^2 = I_N$. The automorphism induced by a Clifford element $g$ satisfies

$$gE(a,b)g^\dagger = \pm E ([a,b]F_g), \text{ where } F_g = \begin{bmatrix} A_g & B_g \\ C_g & D_g \end{bmatrix} \tag{9}$$

is a $2m \times 2m$ binary matrix that preserves symplectic inner products: $\langle [a,b]F_g, [a',b']F_g \rangle_s = \langle [a,b], [a',b'] \rangle_s$. Hence $F_g$ is called a binary symplectic matrix and the symplectic property reduces to $F_g \Omega F_g^T = \Omega$, or equivalently

$$A_g B_g^T = B_g A_g^T, \quad C_g D_g^T = D_g C_g^T, \quad A_g D_g^T + B_g C_g^T = I_m. \tag{10}$$

(See [57] for an extensive discussion on general symplectic geometry and quantum mechanics.) The symplectic property encodes the fact that the automorphism induced by the Clifford element $g$ must respect commutativity in $HW_N$. Let $Sp(2m, F_2)$ denote the group of symplectic $2m \times 2m$ matrices over $F_2$. The map $\phi : Cliff_N \to Sp(2m, F_2)$ defined by

$$\phi(g) \triangleq F_g \tag{11}$$

is a homomorphism with kernel $HW_N$, and every Clifford operator projects onto a symplectic matrix $F_g$. Thus, $HW_N$ is a normal subgroup of Cliff$_N$ and Cliff$_N/HW_N \cong Sp(2m, F_2)$. This implies that the size is $|Sp(2m, F_2)| = 2^{m^2} \prod_{j=1}^m (4^j - 1)$.

**Table I:** A Generating Set of Symplectic Matrices and Their Corresponding Unitary Operators.

| Symplectic Matrix $F_g$ | Clifford Operator $g$ |
|------------------------|------------------------|
| $\Omega = \begin{bmatrix} 0 & I_m \\ I_m & 0 \end{bmatrix}$ | $H_N = H^{\otimes m}$ |
| $L_Q = \begin{bmatrix} Q & 0 \\ 0 & Q^{-1} \end{bmatrix}$ | $\ell_Q : e_v \mapsto e_vQ$ |
| $T_P = \begin{bmatrix} I_m & P \\ 0 & I_m \end{bmatrix} : P = P^T$ | $t_P = \diag \left( i^{P \mod 4} \right)$ |
| $G_t = \begin{bmatrix} L_{m-t} & U_t \\ U_t & L_{m-t} \end{bmatrix}$ | $g_t = H_{2t} \otimes I_{2m-t}$ |

(Also see [7].) Table I lists elementary symplectic transformations $F_g$, that generate the binary symplectic group $Sp(2m, F_2)$, and the corresponding unitary automorphisms $g \in Cliff_N$, which together generate $HW_N$ from Cliff$_N$.

Remark: Since all elements of $S$ are unitary, Hermitean and commute with each other, they can be diagonalized simultaneously with respect to a common orthonormal basis, and their eigenvalues are $\pm 1$ with algebraic multiplicity $N/2$. We refer to such a basis as the common eigenbasis or simply eigenspace of the subgroup $S$, and to the subspace of eigenvectors with eigenvalue $\pm 1$ as the $\pm 1$ eigenspace of $S$.

If the subgroup $S$ is generated by $E(a_i, b_i), i = 1, \ldots, k$, then the operator

$$\frac{1}{2^k} \prod_{i=1}^k (I_N + E(a_i, b_i)) \tag{12}$$

projects onto the $2^{m-k}$-dimensional subspace $V(S)$ fixed pointwise by $S$, i.e., the $+1$ eigenspace of $S$. The subspace $V(S)$ is the stabilizer code determined by $S$. We use the notation $[m, m-k]$ to represent that $V(S)$ encodes $m-k$ logical qubits into $m$ physical qubits.

Let $\gamma(S)$ denote the subspace of $F_2^{2m}$ formed by the binary representations of the elements of $S$ using the homomorphism $\gamma$ in (7). A generator matrix for $\gamma(S)$ is

$$G_S \triangleq [a_i, b_i]_{i=1, \ldots, k} \in F_2^{k \times 2m} \text{ s.t. } G_S \Omega G_S^T = 0, \tag{13}$$

where 0 is the $k \times k$ matrix with all entries zero.

Given a stabilizer $S$ with generators $E(a_i, b_i), i = 1, \ldots, k$, we can define $2^k$ subgroups $S_{\epsilon_1 \cdots \epsilon_k}$ where the index $(\epsilon_1 \cdots \epsilon_k)$
represents that $S_{e_1 \cdots e_k}$ is generated by $e_i E(a_i, b_i)$, for $e_i \in \{\pm 1\}$. Note that the operator

$$
\Pi_{e_1 \cdots e_k} \triangleq \frac{1}{2^k} \prod_{i=1}^{k} (I_N + e_i E(a_i, b_i))
$$

(14)

projects onto $V(S_{e_1 \cdots e_k})$, and that

$$
\sum_{(e_1, \ldots, e_k) \in \{\pm 1\}^k} \Pi_{e_1 \cdots e_k} = I_N.
$$

(15)

Hence the subspaces $V(S_{e_1 \cdots e_k})$, or equivalently the subgroups $S_{e_1 \cdots e_k}$, provide a resolution of the identity, and elements (errors) in $HW_N$ simply permute these subspaces (under conjugation).

Given an $[m, m - k]$ stabilizer code, it is possible to perform encoded quantum computation in any of the subspaces $V(S_{e_1 \cdots e_k})$ by synthesizing appropriate logical Clifford operators (see [42] for algorithms). If we think of these subspaces as threads, then a computation starts in one thread and jumps to another when an error (from $HW_N$) occurs. Quantum error-correcting codes enable error control by identifying the jump that the computation has made. Identification makes it possible to modify the computation in flight instead of returning to the initial subspace and restarting the computation. The idea of tracing these threads is called as Pauli frame tracking in the literature (see [58] and references therein).

A stabilizer group $S$ defined by $k = m$ generators is called a maximal commutative subgroup of $HW_N$ and $\gamma(S)$ is called a maximal isotropic subspace of $F_{2^m}$. The generator matrix $G_S$ has rank $m$ and can be row-reduced to $[0 \mid I_m]$ if $S = Z_N \triangleq \{E(0, b) : b \in F_{2^m}^2\}$, or to the form $[I_m \mid P]$ if $S$ is disjoint from $Z_N$. We will denote these subgroups as $E([0 \mid I_m])$ and $E([I_m \mid P])$, respectively. The condition $G_S G_S^T \equiv 0$ implies $P = P^T$, and any element of $\gamma(S)$ can be expressed in the form $[a, aP]$ for some $a \in F_{2^m}^2$. Note that $E([I_m \mid 0]) = X_N \triangleq \{E(a, 0) : a \in F_{2^m}^2\}$. Since $\dim V(S) = 2^{m-m} = 1$, the subgroup $S$ fixes exactly one vector. The $N$ eigenvectors in an orthonormal eigenbasis for $S$ are defined up to an overall phase and called stabilizer states [28], [29]. The number of non-zero entries in a stabilizer state is determined by the intersection of $S$ with $Z_N$ [59].

IV. Weight Distributions of Kerdock Codes

Kerdock codes were first constructed as non-linear binary codes [45], as was the Goethals code [47] and the Delsarte-Goethals codes [49]. In this section, we describe the Kerdock and Delsarte-Goethals codes as linear codes over $\mathbb{Z}_4$, the ring of integers modulo 4. These $\mathbb{Z}_4$-linear codes were constructed by Hammons et al. [25] as Hensel lifts of binary cyclic codes, and this description requires Galois rings. The description given in Section IV-A requires finite field arithmetic, but is entirely binary and follows [27]. Our construction of unitary 2-designs in Section VI uses the matrices that are defined in Section IV-A. In Section IV-B we make a connection between the Kerdock and Delsarte-Goethals codes and maximal commutative subgroups of $HW_N$ via stabilizer states, use this relation to compute inner products between stabilizer states, and hence calculate weight distributions of Kerdock codes.

A. Kerdock and Delsarte-Goethals Sets [27]

The finite field $\mathbb{F}_{2^m}$ is obtained from the binary field $\mathbb{F}_2$ by adjoining a root $\alpha$ of a primitive irreducible polynomial $p(x)$ of degree $m$ [60]. The elements of $\mathbb{F}_{2^m}$ are polynomials in $\alpha$ of degree at most $m - 1$, with coefficients in $\mathbb{F}_2$, and we will identify the polynomial $z_0 + z_1\alpha + \ldots + z_{m-1}\alpha^{m-1}$ with the binary (row) vector $[z_0, z_1, \ldots, z_{m-1}]$.

The Frobenius map $f : \mathbb{F}_{2^m} \to \mathbb{F}_{2^m}$ is defined by $f(x) \triangleq x^2$, and the trace map $Tr : \mathbb{F}_{2^m} \to \mathbb{F}_2$ is defined by

$$
Tr(x) \triangleq x + x^2 + \ldots + x^2^{m-1}.
$$

(16)

Since $$(x+y)^2 = x^2 + y^2$$ for all $x, y \in \mathbb{F}_{2^m}$, the trace is linear over $\mathbb{F}_2$. The trace inner product $\langle x, y \rangle_{\mathbb{F}} = Tr(xy)$ defines a symmetric bilinear form, so there exists a binary symmetric matrix $W$ for which $Tr(xy) = xW^T y$. In fact

$$
W_{ij} = Tr(\alpha^i \alpha^{j}) , \quad i, j = 0, 1, \ldots, m - 1.
$$

(17)

The matrix $W$ is non-singular since the trace inner product is non-degenerate (if $Tr(xz) = 0$ for all $z \in \mathbb{F}_{2^m}$ then $x = 0$). Observe that $W$ is a Hankel matrix, since if $i + j = b + k$ then $Tr(\alpha^i \alpha^j) = Tr(\alpha^b \alpha^k)$. The matrix $W$ can be interpreted as the primal-to-dual-basis conversion matrix for $\mathbb{F}_{2^m}$, with the primal basis being $\{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ (see [17]).

The Frobenius map $f(x) = x^2$ is linear over $\mathbb{F}_2$, so there exists a binary matrix $R$ for which $f(x) = xR$. Since

$$
f(x_0 + x_1\alpha + \ldots + x_{m-1}\alpha^{m-1})
= x_0 + x_1\alpha^2 + \ldots + x_{m-1}\alpha^{2(m-1)},
$$

(18)

the rows of $R$ are the vectors representing the field elements $\alpha^{2i}, i = 0, \ldots, m - 1$. Note that square roots exist for all elements of $\mathbb{F}_{2^m}$ since $R$ is invertible.

We write multiplication by $z \in \mathbb{F}_{2^m}$ as a linear transformation $xz \equiv x A_z$. For $z = 0, A_0 = 0,$ and for $z = \alpha^i$ the matrix $A_z = A^i$ for $i = 0, 1, \ldots, 2^m - 2$, where $A$ is the matrix that represents multiplication by the primitive element $\alpha$. The matrix $A$ is the companion matrix of the primitive irreducible polynomial $p(x) = p_0 + p_1 x + \ldots + p_{m-1} x^{m-1} + x^m$ over the binary field. Thus

$$
A \triangleq \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
& \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
p_0 & p_1 & p_2 & \cdots & p_{m-1}
\end{bmatrix},
$$

(19)

and we have chosen $A$ rather than $A^T$ as the companion matrix since we are representing field elements in $\mathbb{F}_{2^m}$ by row vectors (rather than column vectors).

Lemma 4: The matrices $A_z$, $W$, and $R^i$, for $i \in [m]$, satisfy:

(a) $A_z A_{z'} = A_z A_{z'}$;
(b) $A_z + A_{z'} = A_{z+z'}$;
(c) $A_z W = W A_z^T$;
(d) $R^i A^2 = A_x R^i$, $R^i A^2 = A_x^{2i} R^i$, and $R^i A^{-2} = A_x^{-2i} R^{-i}$;
(e) $R^i W = W R^{-1} W = (R^i)^T$.
Proof: Identities (a) through (d) follow directly from the arithmetic of \( \mathbb{F}_{2^m} \). Specifically, for (c), observe that
\[
(xA_z)Wy^T = \text{Tr}(xyz) = \text{Tr}(yA_z)x^T = xW(yA_z)^T,
\]
and (d) can be proven similarly. To prove part (e) we observe \( \text{Tr}(x^2) \) and verify that for all \( x, y \in \mathbb{F}_{2^m} \),
\[
(xR^T)y^T = \text{Tr}(x^2y) = \text{Tr}(xy^T) = xW(R^{-1})y^T.
\]

Definition 5: For \( 0 \leq r \leq (m-1)/2 \) and for \( \underline{z} = (z_0, z_1, \ldots, z_r) \in \mathbb{F}_{2^{r+1}} \) define the bilinear form \( \beta_{\underline{z}, r}(x, y) \equiv \text{Tr}[z_0xy + z_1(x^2y + xy^2) + \cdots + z_r(x^2y + xy^2)] \).
Note that \( \beta_{\underline{z}, r}(x, y) \) is represented by the binary symmetric matrix
\[
P_{\underline{z}, r} \triangleq A_{z_0}W + \sum_{i=1}^{r}[A_{z_i}W(R^{i}) + R^{i}WA_{z_i}^T].
\]

The Delsarte-Goethals set \( P_{DG}(m, r) \) consists of all such matrices \( P_{\underline{z}, r} \). The Kerdock set \( P_{K}(m) \equiv P_{DG}(m, 0) \) consists of all matrices \( P_{\underline{z}, 0} \equiv P_{\underline{z}_0} \), where \( \underline{z}_0 = (z, z) \in \mathbb{F}_{2^m} \).

Lemma 6: The Delsarte-Goethals set \( P_{DG}(m, r) \) is an \((m+1)\)-dimensional vector space of symmetric matrices. If \( z \neq 0 \) then \( \text{rank}(P_{\underline{z}, r}) \geq m - 2r \). Matrices in the Kerdock set \( P_{K}(m) \) are non-singular.

Proof: Closure under addition follows from part (b) of Lemma 4. Observe \( \text{Tr}(x^2) = \cdots = \text{Tr}(x^{1/2}) \). If \( x \) is in the nullspace of \( P_{\underline{z}, r} \), i.e., using its vector representation \( xP_{\underline{z}, r} = 0 \), then \( \beta_{\underline{z}, r}(x, y) = 0 \) for all \( y \in \mathbb{F}_{2^m} \) and we obtain
\[
0 = \text{Tr}[z_0xy + z_1(x^2y + xy^2) + \cdots + z_r(x^2y + xy^2)]
\]
\[
= \text{Tr}[z_0x^2y^r + (z_1^2 x^{2r+1} y^r + (z_1 x)^{2r-1} y^r) + \cdots + (z_r^2 x^{2r} y^r + (z_r x)y^r)]
\]
\[
= \text{Tr}[y^r ((z_0 x)^2r + (z_1 x)^{2r-1} + (z_1 x)^{2r-1}) + \cdots + (z^r x^{2r} + z x)].
\]
This holds for all \( y \), so \( (z_0 x)^{2r} + (z_1^2 x^{2r+1} + (z_1 x)^{2r-1}) + \cdots + (z^r x^{2r} + z x) \) must be identically 0, i.e., \( x \) is a root of the polynomial. Since the polynomial has at most \( 2^r \) roots, the nullspace of \( P_{\underline{z}, r} \) has dimension at most \( 2r \), which implies that \( \text{rank}(P_{\underline{z}, r}) \geq m - 2r \).

Remark 7: Note that since the dimension of the vector space of all binary \( m \times m \) symmetric matrices is \( m(m+1)/2 \), the set \( P_{DG}(m, (m-1)/2) \) contains all possible symmetric matrices. For the remainder of this paper we represent a general symmetric matrix as simply \( P \), thereby dropping the subscripts \( \underline{z}, r \) unless necessary. We will continue to represent Kerdock matrices as \( P_z \).

B. Delsarte-Goethals Codes and Weight Distributions

Hammons et al. [25] showed that the classical nonlinear Kerdock and Delsarte-Goethals codes, defined by quadratic forms in [45], are images of linear codes over \( \mathbb{Z}_4 \) under the Gray map. In this section, we begin by reviewing this construction using the Kerdock and Delsarte-Goethals sets of matrices, and demonstrate that exponentiating these \( \mathbb{Z}_4 \)-valued codewords entry-wise by \( i \) produces stabilizer states. For stabilizer states of \( E(\{I_m \mid P_{\underline{z}, r}\}) \) and \( E(\{I_m \mid P_{\underline{z}_0, r}\}) \), we calculate their Hermitian inner products using the trace of certain projection operators, and show that the distribution of inner products depends only on \( \text{rank}(P_{\underline{z}, r} + P_{\underline{z}_0, r}) \). Then, since \( \text{rank}(P_z) \in \{0, m\} \) for all \( P_z \in P_K(m) \), we compute the weight distribution of Kerdock codes by relating these Hermitian inner products to the histogram of values in the difference between two \( \mathbb{Z}_4 \)-valued codewords. In order to calculate the weight distribution of Delsarte-Goethals codes, we would need to determine the distribution of ranks in the Delsarte-Goethals sets \( P_{DG}(m, r) \). While this question is straightforward for \( P_K(m) \), it remains open for general \( P_{DG}(m, r) \) and will be investigated in future work.

Definition 8: The \( \mathbb{Z}_4 \)-linear Delsarte-Goethals code is given by
\[
\text{DG}(m, r) \equiv \{[xP_{x}, 2w_{xT} + \kappa]|x \in \mathbb{F}_{2^m}: P \in P_{DG}(m, r), \quad w \in \mathbb{F}_{2^m}, \kappa \in \mathbb{Z}_4\}.
\]
Lemma 10: Let \( u, v \in \mathbb{C}^N \) be two length-\( N \) vectors of quaternary phases. Then
\[
\langle u - v, u - v \rangle = 2d_H(g(u), g(v)),
\]
where \( d_H \) denotes the Hamming distance.

Next, we prove a lemma establishing the relation between \( \text{DG}(m, r) \) and the common eigenspace of \( E([I_m \mid P]) \) determined by a binary symmetric matrix \( P \) (see Remark 2). Note that we denote the maximal commutative subgroup determined by the rows of \( [I_m \mid P] \) as \( E([I_m \mid P]) \), and that we do not normalize eigenvectors (stabilizer states) in this section, since the Gray map needs to be applied to quaternary phases.

Lemma 11: Given a binary symmetric matrix \( P \), the (column) vectors \([x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m}\) are common eigenvectors of the maximal commutative subgroup \( E([I_m \mid P]) \). Each eigenvector has Euclidean length \( \sqrt{N} = 2^{m/2} \).

Proof: It is possible to prove this result by direct calculation, i.e., by calculating \( E(a, ap) \cdot [x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m} \) for some \( a \in \mathbb{F}_2^m \), but the following argument uses the mathematical framework described in Section III. Note that
\[
[x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m} = \sum_{x \in \mathbb{F}_2^m} x^TPx^T + 2wx^T e_x,
\]
where \( e_x \) denotes the standard basis vector in \( \mathbb{C}^N \) with a 1 in the position \( x \) and 0 elsewhere.

The columns \([x^TPx^T]_{x \in \mathbb{F}_2^m} = [(-1)^{wx}]_{x \in \mathbb{F}_2^m}\) of the Walsh-Hadamard matrix \( W_N \), where \( w \in \mathbb{F}_2^m \) indexes the column, are common eigenvectors of the maximal commutative subgroup \( X_N = E([I_m \mid 0]) \). The Clifford operator \( t_{P} = \text{diag} \left( x^TPx^T \right) \) corresponds to the binary symplectic matrix \( T_{P} = \begin{bmatrix} I_m & P \\ 0 & I_m \end{bmatrix} \) (see Table 1). Hence conjugation by \( t_{P} \) maps \( E([I_m \mid 0]) \) to \( E([I_m \mid P]) \), i.e., \( t_{P}E(a,0)t_{P}^{\dagger} = E(a,ap) \), and so the common eigenvectors of \( E([I_m \mid P]) \) are
\[
t_{P} \cdot [x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m} = [x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m}.
\]
It is easily verified that for any \( a \in \mathbb{F}_2^m \),
\[
\left( t_{P}E(a,0)t_{P}^{\dagger} \right) \cdot [x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m} = \pm [x^TPx^T + 2wx^T]_{x \in \mathbb{F}_2^m}.
\]

Now we have the following important observation. Given \( v \in \mathbb{C}^N \), define
\[
S_v \triangleq \{ g \in HW_N : gv = \alpha v, \text{ where } \alpha \in \mathbb{C} \text{ and } |\alpha| = 1 \},
\]
(24)

Lemma 12:
(a) \( S_v \) is commutative.
(b) There is a 1-1 correspondence between maximal commutative subgroups of \( HW_N \) and stabilizer states.

Proof:
(a) If \( E(a, b), E(a', b') \in S_v \) then
\[
E(a, b)E(a', b')v = \alpha \alpha' v = \alpha' \alpha v = E(a', b')E(a, b)v.
\]
(b) If \( v \) is a stabilizer state then \( S_v \) is a maximal commutative subgroup of \( HW_N \).

Remark 13: Any stabilizer state of a maximal commutative subgroup of \( HW_N \) disjoint from \( Z_N \) can be obtained by exponentiating Delsarte-Goethals codewords, and multiplying the quaternary phase vector by \( N^{-s/2} \). The maximal commutative subgroups \( E([I_m \mid P_j]) \) determined by the Kerdock matrices \( P_j \) intersect trivially. Together with \( Z_N = E([0 \mid I_m]) \), they partition all \( (N^2 - 1) \) non-identity Hermitian Pauli matrices. Hence, given a non-identity Hermitian Pauli matrix \( E(a, b) \), it follows that there is a sign \( \epsilon \in \{ \pm 1 \} \) such that \( \epsilon E(a, b) \) is in one of the \( N + 1 \) subgroups determined by all \( P_j \in P_K(m) \) and \( Z_N \).

Therefore, stabilizer states connect the classical world of Kerdock and Delsarte-Goethals codes and the quantum world of maximal commutative subgroups of \( HW_N \). This synergy has proven successful in several applications 27, 31–33, 35, 51 and our construction of a unitary 2-design here, from stabilizer states, is yet another instance.

Remark 14: Note that in Lemma 11 we only considered \( \kappa = 0 \) while exponentiating codewords \( c_u \in \text{DG}(m, r) \). This is to ensure that the resulting eigenvector corresponds to a \( \pm 1 \) eigenvalue (and not a value in \( \{ \pm 1, \pm i \} \)). However, Theorem 7 considers all \( \kappa \in \mathbb{Z}_4 \) while calculating the weight distribution.

Given \( P \in P_{DG}(m, r) \), we scale the common eigenvectors of \( E([I_m \mid P]) \) by \( \sqrt{N} \) to obtain a set \( V(P) \) of length-\( N \) vectors of quaternary phases. (Note the similarity to the notation \( V(S) \) used in Section III and observe that here we consider all eigenvectors, albeit unnormalized, and not just the +1 eigenspace.) Therefore, if we can compute the Hermitian inner products between these unnormalized stabilizer states then we can use Lemma 10 to calculate the weight distribution of Kerdock and Delsarte-Goethals codes. Note that despite being non-linear codes the weight and distance distributions of these codes coincide, as shown in 24 Chapter 15 (which follows from \( Z_4 \)-linearity and Gray isometry).

Lemma 15: Let \( P_1, P_2 \in P_{DG}(m, r) \) be distinct. Fix \( v \in V(P_2) \) and let \( u \) run through \( V(P_1) \). If \( \text{rank}(P_1 + P_2) = k \) then
\[
|\langle u, v \rangle|^2 = \begin{cases} 2^{2m-k} & \text{for } 2^k \text{ eigenvectors } u, \\ 0 & \text{for } 2^m - 2^k \text{ eigenvectors } u. \end{cases}
\]

Proof: Let \( Q = [I_m \mid P_1] \cap [I_m \mid P_2] \) represent a basis for the subspace formed by intersecting the spaces generated by \([I_m \mid P_1]\) and \([I_m \mid P_2]\). Then \( \dim(Q) = m - k \). Let \( [a_1, b_1], \ldots, [a_{m-k}, b_{m-k}] \) be a basis for \( Q \), and complete to bases for \( P_1 \) and \( P_2 \) by adding vectors \( [c_j, d_j], j = m - k + 1, \ldots, m \) and \( [c_j', d_j'], j = m - k + 1, \ldots, m \) respectively. Since \( v \) is fixed, using (14), there are \( f_i, t_j \in \{ \pm 1 \} \) such that
\[
\left( \frac{1}{N} \right)^v \left( \frac{1}{N} \right)^u = \prod_{i=1}^{m-k} \left( I_N + f_i E(a_i, b_i) \right) \prod_{j=m-k+1}^{m} \left( I_N + t_j E(c_j', d_j') \right),
\]
and since the only constraint for \( u \) is to be from \( V(P_1) \),
\[
\left( \frac{1}{N} \right)^u \left( \frac{1}{N} \right)^u = \prod_{i=1}^{m-k} \left( I_N + f_i E(a_i, b_i) \right) \prod_{j=m-k+1}^{m} \left( I_N + t_j E(c_j', d_j') \right),
\]
where \( e_i, s_i \in \{\pm 1\} \) are variable. Since \(|\langle u, v \rangle|^2 = \text{Tr}(uu^\dagger vv^\dagger)\) it only remains to calculate \(\text{Tr}(uu^\dagger vv^\dagger)\). If \( e_i = f_i \quad \forall \ i\), then
\[
\langle I_N + e_i E(a_i, b_i) | I_N + f_i E(a_i, b_i) \rangle = 2(I_N + e_i E(a_i, b_i))
\]
so that
\[
\text{Tr}(uu^\dagger vv^\dagger) = 2^{m-k} \prod_{i=1}^{m-k} (I_N + e_i E(a_i, b_i)) \prod_{j=m-k+1}^{m} (I_N + s_j E(c_j, d_j)).
\]

Expanding the right hand side, the only term with nonzero trace is the identity with trace \(2^m\). Hence in this case \(\text{Tr}(uu^\dagger vv^\dagger) = 2^{2m-k}\). The \(k\) eigenvalues \(s_j\) can be freely chosen, so there are \(2^k\) eigenvectors in this case.

If \( e_i \neq f_i \) for some \( i\), then
\[
\langle I_N + e_i E(a_i, b_i) | I_N + f_i E(a_i, b_i) \rangle = 0
\]
and \(\text{Tr}(uu^\dagger vv^\dagger) = 0\). There are \(2^m - 2^k\) such eigenvectors.

Finally, if \( k = m \) then \(\text{Tr}(uu^\dagger vv^\dagger) = \text{Tr}(I_N) = 2^m \forall u\).

Corollary 16: For \(P_1, P_2 \in P_k(m)\), since \(\text{rank}(P_1 + P_2) \in \{0, m\}\) the inner products are
\[
|\langle u, v \rangle|^2 = \begin{cases} 
0 & \text{if } P_1 = P_2 \text{ and } u \neq v \\
2^m & \text{if } P_1 \neq P_2, \\
2^2m & \text{if } (P_1 = P_2 \text{ and } u = v). 
\end{cases} \tag{26}
\]

This result is the primary tool that allowed us to simplify the derivation of the weight distribution of Kerdock codes in Theorem 1. Note that Theorem [1] is the only result in this section that is restricted to Kerdock codes but not general Delsarte-Goethals codes (and requires \(m = 0\) to be odd).

V. MUTUALLY UNBIASED BASES FROM \(P_k(m)\)

In this section, we will organize the columns of \(I_N\) (i.e., the common eigenbasis of \(Z_N\)) and all stabilizer states determined by \(P_z \in P_k(m)\) into a matrix to form mutually unbiased bases, and analyze its symmetries. This symmetry group will eventually lead to the construction of the unitary 2-design. We first state a result that holds for stabilizer states determined by matrices from general Delsarte-Goethals sets.

Definition 17: Given a collection \(M\) of unit vectors in \(\mathbb{C}^N\) (Grassmannian lines) the chordal distance \(\text{chor}(S)\) is given by
\[
\text{chor}(M) \triangleq \min_{(u,v) \in M} \sqrt{1 - |\langle u, v \rangle|^2}. \tag{27}
\]

It follows from Lemma [15] that the Delsarte-Goethals set \(P_{DG}(m, r)\) determines \(2^{m(r+2)}\) complex lines (stabilizer states) in \(\mathbb{C}^N\) with chordal distance \(\sqrt{1 - 2^{-(m-2r)}}\) (cf. [61]).

Definition 18: Two \(N \times N\) unitary matrices \(U\) and \(V\) are said to be mutually unbiased if \(| \langle u, v \rangle | = N^{-\frac{1}{2}}\) for all columns \(u\) of \(U\), and all columns \(v\) of \(V\). Each matrix is interpreted as an orthonormal basis and collections of such unitary matrices that are pairwise mutually unbiased are called mutually unbiased bases (MUBs). Vectors in each orthonormal basis look like noise to the other bases (due to the small inner product).

Corollary 16 when applied to normalized eigenvectors, shows that the \(N\) eigenbases determined by the Kerdock set \(P_k(m)\) are mutually unbiased (also see Remark [13]). Let \(B_k(m)\) denote the collection of these \(N\) eigenbases (of \(E(\{0 | I_m\})\) for all \(P \in P_k(m)\)) along with the eigenbasis \(I_N\) of \(E(\{0 | I_m\})\). This is a set of \(N + 1\) mutually unbiased bases and they determine an ensemble of \((N + 1)\) complex lines (stabilizer states) that is extremal [34]. In this section, we provide an elementary description of their group of Clifford symmetries.

A. The Kerdock MUBs

Recollect from Section [IV-A] that the Kerdock matrices are \(P_z = A_z W\), where \(W\) is a symmetric Hankel matrix with binary entries that satisfies \(\text{Tr}(xy) = xWy^T\) and \(A_z\) represents multiplication by \(z\), both in \(\mathbb{F}_{2^m}\). Using the result of Lemma [11] define \(M\) mutually unbiased bases
\[
M_z \triangleq t_{P_z} H_N = \text{diag}(t^{xP_zx^T}) H_N, \quad z \in \mathbb{F}_{2^m}, \tag{28}
\]
where \([H_N]_{x,y} = \frac{1}{\sqrt{N}} (-1)^{x+y^T}\), for \(x, y \in \mathbb{F}_{2^m}\), is the Walsh-Hadamard matrix of order \(N\). Note that \(M_0 = H_N\) and that all columns of \(M_z\) have Euclidean length \(\frac{1}{\sqrt{N}}\). Complete the MUBs by appending the matrix \(M_{\infty} \triangleq I_N\). The common eigenspaces of the maximal commutative subgroup \(E(\{0 | I_m\})\) are the columns of \(M_z\) and the common eigenspaces of \(E(\{0 | I_m\})\) are the standard unit coordinate vectors. Hence the set of Kerdock MUBs
\[
B_k(m) \triangleq \{ I_N, M_z : z \in \mathbb{F}_{2^m} \} \tag{29}
\]
is a maximal collection of mutually unbiased bases [34].

B. Symmetries of Kerdock MUBs

Let \(M\) be the \(N \times (N + 1)\) matrix given by
\[
M \triangleq \begin{bmatrix} M_\infty & M_0 & \cdots & M_z & \cdots \end{bmatrix}. \tag{30}
\]
Note that \(M_\infty = I_N\) and \(M_0 = H_N\).

Definition 19: A symmetry of \(M\) is a pair \((U,G)\) such that \(UMG = M\), where \(U\) is an \(N \times N\) unitary matrix, and \(G\) is a generalized permutation matrix, i.e., \(G = \Pi D\) where \(\Pi\) is a permutation matrix and \(D\) is a diagonal matrix of quaternary phases.

Observe that for any such symmetry, \(G\) can undo the action of \(U\) if and only if \(U\) induces a (generalized) permutation on the columns of \(M\). Moreover, since \(U\) is unitary it has to preserve inner products, so Corollary 16 implies that \(U\) can only permute the bases \(M_z\) and permute columns within each basis, or equivalently permute the corresponding maximal commutative subgroups and permute elements within each subgroup, respectively, by conjugation.

Lemma 20: For any symmetry \((U,G)\) of \(M\), the unitary matrix \(U\) is an element of the Clifford group \(\text{Cliff}_N\).
Proof: A Pauli matrix \( E(a, b) \in E([I_m \mid P_z]) \) that fixes \( M_z \) can be written as \( E(a, b) = \sum_{v \in M_z} \epsilon_v v^v \), where \( \epsilon_v = \pm 1 \) for all \( v \). Since \( U \) permutes the eigenbases \( M_z \), it follows that \( U v \in M_z \), for some \( z' \in F_{2m} \cup \{\infty\} \), is fixed by \( U E(a, b) U^T \) which must again be a Pauli matrix. Hence \( U \) is a Clifford element.

We first observe the symmetries induced by elements of \( HW_N \).
1) Pauli Matrices \( E(a, 0) \) and \( E(0, b) \): The group \( E([I_m \mid 0]) \) of Pauli matrices \( E(a, 0) \) fixes each column of \( M_0 = H_N \) and acts transitively on the columns of each of the other \( N \) blocks. The group \( E([0 \mid I_m]) \) of Pauli matrices \( E(0, b) \) fixes each column of \( M_\infty = I_N \) and acts transitively on the columns of each of the remaining blocks.

Definition 21: The projective special linear group \( PSL(2, 2^m) \) is the group of all transformations
\[
f(z) = \frac{az + b}{cz + d}, \quad a, b, c, d \in F_{2m}, \quad ad + bc = 1,
\]
acting on the projective line \( F_{2m} \cup \{\infty\} \). The transformation \( f \) is associated with the action of the \( 2 \times 2 \) matrix \( \begin{bmatrix} a & b \\ c & d \end{bmatrix} \) on 1-dimensional spaces, since
\[
f: \begin{bmatrix} z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} az + b \\ cz + d \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} a \pm i b \\ c \pm i d \end{bmatrix}.
\]
The projective linear group \( PTL(2, 2^m) \) is the group of all transformations
\[
f(z) = \frac{az^{2^{-i}} + b}{cz^{2^{-i}} + d}, \quad a, b, c, d \in F_{2m}, \quad ad + bc = 1,
\]
and \( i \in \{0, 1, \ldots, m - 1\} \). The orders are
\[
|PSL(2, 2^m)| = (N + 1)N(N - 1) = 2^{2m} - 2^m,
\]
\[
|PTL(2, 2^m)| = (N + 1)N(N - 1)m.
\]

Now we analyze the symmetries induced by elements of the binary symplectic group \( Sp(2m, F_2) \).
2) Clifford Symmetries of \( M \): The group \( PSL(2, 2^m) \) is generated by the transformations \( z \mapsto z + x, z \mapsto z x, z \mapsto 1/z \). The group \( PTL(2, 2^m) \) is \( PSL(2, 2^m) \) enlarged by the Frobenius automorphisms \( z \mapsto z^{2^{-i}} \equiv z^{R^{-i}} \) discussed in Section [V-A].

We realize each of these transformations as a symmetry of \( M \). We recall that \( A_z W A_z^T = A_x W \) from part (c) of Lemma [4] and for convenience we work with maximal commutative subgroups \( E([I_m \mid A_z^2 W]) \), i.e., the Kerdock matrices are \( P_z = A_z^2 W \). Note that every field element \( \beta \in F_{2m} \) is a square, so this is equivalent to \( P_z = A_z^2 W \).

(i) \( z \mapsto z + x \) becomes \( \begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} \mapsto \begin{bmatrix} I_m \mid A_{x+z}^2 W \end{bmatrix}: \)
\[
\begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} = \begin{bmatrix} Z_x Z_y W \end{bmatrix} = \begin{bmatrix} I_m \mid (A_z^2 + A_x^2) W \end{bmatrix} \equiv \begin{bmatrix} I_m \mid A_z^2 (A_x^2 + z) W \end{bmatrix}.
\]

(ii) \( z \mapsto xz \) becomes \( \begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} \mapsto \begin{bmatrix} I_m \mid A_{xz}^2 W \end{bmatrix}: \)
\[
\begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} = \begin{bmatrix} A_x^{-1} A_z W A_x \end{bmatrix} = \begin{bmatrix} I_m \mid A_z^2 (A_x z + b) W \end{bmatrix}.
\]

(iii) \( z \mapsto 1/z \) becomes \( \begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} \mapsto \begin{bmatrix} I_m \mid A_x^{-2} W \end{bmatrix}: \)
\[
\begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} = \begin{bmatrix} Z_x W^{-1} A_z W \end{bmatrix} = \begin{bmatrix} I_m \mid A_z^2 (A_x^{-1} z) W \end{bmatrix}.
\]

(iv) \( z \mapsto z^i \equiv z^{R^{-i}} \) becomes \( \begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} \mapsto \begin{bmatrix} I_m \mid A_x^{2^{-i}} W \end{bmatrix}: \)
\[
\begin{bmatrix} I_m \mid A_x^2 W \end{bmatrix} = \begin{bmatrix} (R^{-i}) W^{-1} A_z W \end{bmatrix} = \begin{bmatrix} I_m \mid A_z^2 (R^{-i}) W \end{bmatrix}.
\]

Let \( \mathcal{P}_{K,m} \) be the group of symplectic transformations generated by (i), (ii) and (iii) above, and let \( \mathcal{P}_{K,m}^* \) be the group \( \mathcal{P}_{K,m} \) enlarged by the generators (iv). Thus, using notation in Table II, we have
\[
\mathcal{P}_{K,m} \triangleq (T_{A_z^2 W}, L_{A_z^{-1}}, \Omega L_{W^{-1}}; x \in F_{2m}) \cong PSL(2, 2^m),
\]
\[
\mathcal{P}_{K,m}^* \triangleq (T_{A_z^2 W}, L_{R^{-i} A_z^{-1}}, \Omega L_{W^{-1}}; x \in F_{2m}) \cong PTL(2, 2^m).
\]

Although \( T_{A_z^2} = I_{2m} \) in the unitary group we have \( T_{A_z^2 W} = E(0, dW) \). Therefore the corresponding Clifford elements will generate a group larger than \( PSL(2, 2^m) \) and \( PTL(2, 2^m) \). Each symplectic matrix in the above groups can be transformed into a quantum circuit (or simply a unitary matrix) by expressing it as a product of standard symplectic matrices from Table II (see [22 Section II]).

Remark 22: Note that \( \Omega \notin \mathcal{P}_{K,m} \) but \( \Omega L_{W^{-1}} \in \mathcal{P}_{K,m} \), which means \( H_N \) does not permute columns of \( M \) but \( H_N \Omega L_{W^{-1}} \) does. Hence, for example, to map \( [0, a] \) to \( [a, 0] \) one sequence would be \( (\Omega L_{W^{-1}}) \cdot L_{A_z^{-1}} \), where \( b \) satisfies \( aW^{-1} A_b^{-1} = a \). This does not force \( A_{b^{-1}} = W \).

Lemma 23: Any element of \( \mathcal{P}_{K,m} \) as a product of at most 4 basis symplectic matrices given in Section [V-B2].

Proof: Using the results in this Section, a general block permutation \( \begin{bmatrix} I_m \mid A_z W \end{bmatrix} \mapsto \begin{bmatrix} I_m \mid A_z z + b \end{bmatrix} \) is realized as
\[
\begin{bmatrix} I_m \mid A_z^2 W \end{bmatrix} W^{-1} A_z^{-1} (R^{-i} A_z) = \begin{bmatrix} I_m \mid A_z^{2^{-i}} (A_z^{-1} z) W \end{bmatrix}.
\]

It can be verified that the above is a valid symplectic matrix and satisfies all conditions in (10). We now show that this matrix can be decomposed as a product of 4 basis matrices.

We use the results in Lemma [4] and observe the following:
\[
\begin{bmatrix} A_x^{-2} W \end{bmatrix} (R^{-i} A_z) = \begin{bmatrix} W^{-1} A_x^{-2} W \end{bmatrix}.
\]
\[
\begin{bmatrix}
I_m & A_x^2 W \\
0 & I_m
\end{bmatrix}
\begin{bmatrix}
0 & A_x^{-2} W \\
W^{-1} A_x^2 & 0
\end{bmatrix}
= \begin{bmatrix}
A_x^2 y & A_x^{-2} W \\
W^{-1} A_x^2 & 0
\end{bmatrix}:
\]

\[
\begin{bmatrix}
A_x^2 y & A_x^{-2} W \\
W^{-1} A_x^2 & 0
\end{bmatrix}
\begin{bmatrix}
I_m & A_y^2 W \\
0 & I_m
\end{bmatrix}
= \begin{bmatrix}
A_x^2 y & A_x^{-2} W \\
W^{-1} A_x^2 & 0
\end{bmatrix}
\begin{bmatrix}
A_x^2 y & A_x^{-2} W \\
W^{-1} A_x^2 & 0
\end{bmatrix}(A_{xy})^T,
\]

Hence we set \( x \triangleq c, w \triangleq a/c, y \triangleq d/c \) so that \( wxy + x^{-1} = ad/c + c = b \) and the resultant matrix matches the general symplectic matrix given above.

**Corollary 24:** Let \( a, b, c, d \in \mathbb{F}_2 \) be such that \( ad + bc = 1 \).

The isomorphism \( \tau : \text{PSL}(2, 2^m) \to \mathfrak{P}_{K,m} \) can be defined as

\[
\tau \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) \triangleq T_{A_{xy}} W \cdot L_{A_x^{-2}} \cdot \Omega L_{W^{-1}} \cdot T_{A_{xy}} W. 
\]

Observe that this provides a systematic procedure to sample from the group \( \mathfrak{P}_{K,m} \). By choosing \( \alpha, \beta, \delta \in \mathbb{F}_2 \) uniformly at random, a symmetry element can be constructed as

\[
F_{\alpha, \beta, \delta} \triangleq T_{A_{xy}} W \cdot L_{A_x} \cdot (\Omega L_{W^{-1}} \cdot T_{A_{xy}} W). \tag{41}
\]

The first two factors provide transitivity on the Hermitian matrices of all maximal commutative subgroups except \( Z_N = E([0 \mid I_m]) \), and the last factor enables exchanging any subgroup \( E([I_m \mid P_z]) \) with \( E([0 \mid I_m]) \) (see Lemma 21).

We complete this section by observing that this symmetry group can be enlarged by including the Frobenius automorphisms \( R \) from Section 10.

**Lemma 25:** An arbitrary element from \( \mathfrak{P}_{K,m} \) specified by \( a, b, c, d \in \mathbb{F}_2 \) and \( t \in \{0, \ldots, m-1\} \) takes the form

\[
F = \begin{bmatrix}
R^{-1} A_y^2 & R^{-1} A_x^2 W \\
W^{-1} R^{-1} A_y^2 & (R^t)(A_x^2)^T
\end{bmatrix}, \tag{42}
\]

with \( ad + bc = 1 \), and realizes the block permutation

\[
[I_m \mid A_x^2 W] \rightarrow [I_m \mid A_{x^2y^2} W], \quad z \rightarrow z^{2^{-1}}. \tag{43}
\]

**Proof:** See Appendix A.

VI. UNITARY 2-DESIGNS FROM THE KERDOCK MUBS

In this section, we show that the unitary transformations determined by \( \mathfrak{P}_{K,m} \), along with Pauli matrices \( D(a, b) \in HW_N \) form a unitary 2-design. We first define a graph on Pauli matrices, where Clifford elements act as graph automorphisms. We then show that a group of automorphisms that acts transitively on vertices forms a unitary 2-design. Finally we show that a group of automorphisms that acts transitively on vertices, on edges, and on non-edges forms a unitary 3-design.

**Definition 26:** The Heisenberg-Weyl graph \( \mathbb{H}_N \) has \( N^2 - 1 \) vertices, labeled by pairs \( \pm E(a, b) \) with \( [a, b] \not\equiv [0, 0] \) where vertices labeled \( \pm E(a, b) \) and \( \pm E(c, d) \) are joined if \( E(a, b) \) commutes with \( E(c, d) \). We use \([a, b] \) to represent the vertex labeled \( \pm E(a, b) \) and \([a, b] \rightarrow [c, d] \) to represent an edge between two vertices.

**Remark 27:** Elements of the Clifford group act by conjugation on \( HW_N \), inducing automorphisms of the graph \( \mathbb{H}_N \). We shall distinguish two types of edges in \( \mathbb{H}_N \). Type 1 edges connect vertices from the same maximal commutative subgroup \( E([I_m \mid P_z]), z \in \mathbb{F}_2^m \), and from \( E([0 \mid I_m]) \).

2 edges connect vertices from different maximal commutative subgroups.

To see that \( \text{Aut}(\mathbb{H}_N) = \text{Sp}(2m, \mathbb{F}_2) \), determine a symplectic matrix to reduce an arbitrary graph automorphism to an automorphism \( \pi \) that fixes \( [e_i, 0], [0, e_i], i = 1, \ldots, m \), then show that \( \pi \) fixes every vertex. This essentially amounts to solving for a symplectic matrix satisfying a linear system.

**Definition 28** (Def. 2.4): A strongly regular graph with parameters \( (n, t, \lambda, \mu) \) is a graph with \( n \) vertices, where each vertex has degree \( t \), and where the number of vertices joined to a pair of distinct vertices \( x, y \) is \( \lambda \) or \( \mu \) according as \( x, y \) are joined or not joined respectively.

**Lemma 29:** The Heisenberg-Weyl graph \( \mathbb{H}_N \) is strongly regular with parameters

\[
n = N^2 - 1, \quad t = \frac{N^2}{2} - 2, \quad \lambda = \frac{N^2}{4} - 3, \quad \mu = \frac{N^2}{4} - 1. \tag{44}
\]

**Proof:** A vertex \([c, d] \) joined to a given vertex \([a, b] \) is a solution to \([a, b] \Omega[c, d]^T = 0 \) (due to (3)). This is a linear system with a single constraint, and after eliminating the solutions \([0, 0] \) and \([a, b] \), we are left with \( t = \frac{N^2}{2} - 2 \) distinct vertices \([c, d] \) joined to \([a, b] \).

[The rest of the document continues with proofs and definitions, including the calculation of symplectic matrices and the properties of the Heisenberg-Weyl graph.]
are called $k$-fold twirls. A unitary $k$-design is defined by the property that the ensemble twirl coincides with the full unitary twirl.

We define the Kerdock twirl to be the linear transformation of $(C^N)^\otimes 2$ determined by the uniformly weighted ensemble consisting of $\phi^{-1}(\mathbb{P}_{K,m})$ along with Pauli matrices $D(a,b)$, where $\phi: \text{Cliff}_N/\text{HW}_N \rightarrow \text{Sp}(2m,\mathbb{F}_2)$ (from Section III). Similarly, we define the 2-fold action (in \[45\]) of the ensemble determined by $\mathbb{P}_{K,m}$ as the enlarged Kerdock twirl.

**Definition 34:** An ensemble $E = \{p_i, U_i\}_{i=1}^c$ of Clifford elements $U_i$ is Pauli mixing if for every vertex $[a,b]$ the distribution $\{p_i, U_i E(a,b) U_i^\dagger\}$ is uniform over vertices of $\mathbb{H}_N$. The ensemble $E$ is Pauli 2-mixing if it is Pauli mixing and if for every edge (resp. non-edge) $([a,b],[c,d])$ the distribution $\{p_i, (U_i E(a,b) U_i^\dagger, U_i E(c,d) U_i^\dagger)\}$ is uniform over edges (resp. non-edges) of $\mathbb{H}_N$.

**Theorem 35:** Let $G$ be a subgroup of the Clifford group containing all $D(a,b) \in \text{HW}_N$, and let $E = \{\phi(U), U \in G\}$ be the ensemble defined by the uniform distribution. If $G$ acts transitively on vertices of $\mathbb{H}_N$ then $E$ is a unitary 2-design, and if $G$ acts transitively on vertices, edges and non-edges then $E$ is a unitary 3-design.

**Proof:** Transitivity means a single orbit so that random sampling from $G$ results in the uniform distribution on vertices, edges, or non-edges. Hence transitivity on vertices implies $E$ is Pauli mixing and transitivity on vertices, edges and non-edges implies $E$ is Pauli 2-mixing. It now follows from [12] or [17] that Pauli mixing (resp. Pauli 2-mixing) implies $E$ is a unitary 2-design (resp. unitary 3-design).

**Corollary 36:** Random sampling from the Clifford group gives a unitary 3-design and random sampling from the groups $\mathbb{P}_{K,m}$, $\mathbb{P}_{K,m}$, followed by a random Pauli matrix $D(a,b)$ gives unitary 2-designs.

Puchała and Miszczak have developed a useful Mathematica package IntU [64] for symbolic integration with respect to the Haar measure on unitaries. For small $m$, we used this utility to verify the equality in \[45\] explicitly for the Kerdock 2-design.

Sampling uniformly from the groups $\mathbb{P}_{K,m}$ can be achieved using the `systematic` procedure shown in [44]. The resultant symplectic matrix can be transformed into a quantum circuit (or simply a unitary matrix) by expressing it as a product of standard symplectic matrices from Table II (see [42] Section II). Although our unitary 2-design is equivalent to that discovered by Cleve et al. [17], the methods we use to translate design elements to circuits are very different and much simpler. While they use sophisticated methods from finite fields to propose a circuit translation that is tailored for the design, our algorithm from [36] (whose details are discussed in [42] Appendix II) is a general purpose procedure that can be used to translate arbitrary symplectic matrices to circuits. They have been able to show Clifford-gate-complexity $O(m \log m \log \log m)$ assuming the extended Riemann hypothesis is true, or $O(m \log^2 m \log \log m)$ unconditionally, both of which are near-linear when compared to the $O(m^2)$ gate-complexity for general Clifford elements. Our open-source implementation allows one to construct the design for a specified number of qubits, and we use this utility to calculate worst-case gate complexities on up to 16 qubits.

In our sampling procedure [44] we have three elementary forms $T_{A_{\alpha}}$, $L_{A_{\beta}}$, and $\Omega$, which translate to phase and controlled-Z gates, permutations and controlled-NOT gates, and Hadamard gates on all qubits, respectively (see Appendix I). Note that $L_{W^{-1}}$ has the same elementary form as $L_{A_{\beta}}$, although $W$ is fixed for a given $m$. The Hadamard gates add only $O(m)$ complexity. Figures 2, 3, and 4 plot the worst-case complexities of the gates $T_{A_{\alpha}}$, $L_{A_{\beta}}$, and $L_{W^{-1}}$ obtained using our procedure. The only form that seems to grow faster than $O(m \log m \log \log m)$ is $L_{A_{\beta}}$, and we are currently investigating methods to calculate this gate complexity via analytical arguments that leverage results in the classical computation literature. A curious data point is $m = 15$ in Fig. 4 where the matrix $W$ has zeros everywhere except the anti-diagonal, which translates to a single permutation of the qubits. Since the decomposition in [44] involves a constant number of factors, the overall complexity is that of the factor with largest order term. We will also investigate if our circuits can always be organized to give a depth of $O(\log m)$ just as Cleve et al.

Hence, we have provided an alternative perspective to the quantum unitary 2-design discovered by Cleve et al. by establishing a connection to classical Kerdock codes, and simplified the description of the design as well as its translation to circuits. Since we also appear to achieve competitive Clifford-gate-complexities, and provide implementations for our methods, we believe this paves the way for employing this 2-design in several applications, specifically in randomized benchmarking [10, 52].

**VII. LOGICAL UNITARY 2-DESIGNS**

In this section, we apply our synthesis algorithm [42] to logical randomized benchmarking, which was recently proposed by Combes et al. [11] as a more precise protocol to reliably
estimate performance metrics for an error correction implementation, as compared to the standard approach of physical randomized benchmarking. Using this procedure, they are able to quantify the effects of imperfect logical gates, crosstalk, and correlated errors, which are typically ignored. They use the full logical Clifford group to perform benchmarking as this group forms a (logical) unitary 2-design (as well as 3-design). Our construction can be used to implement their protocol with a much smaller 2-design (\(P_{K,4}\)) and our synthesis algorithm can be used to realize the design at the logical level.

Here we show by example that we can efficiently translate any unitary 2-design of Clifford elements into a logical unitary 2-design, i.e., for a given \([m, m-k]\) stabilizer code, produce \(m\)-qubit physical Clifford circuits that form a unitary 2-design on the \(m-k\) protected qubits. As a proof of concept, we translate the Kerdock design \(P_{K,m-k}\) on the \(m-k = 4\) logical qubits of the \([6,4,2]\) CSS code [53, 65] into their physical 6-qubit implementations. The stabilizers of this code are \(S \triangleq \langle X^{\otimes 6}, Z^{\otimes 6} \rangle\)

\[
\langle E(111111, 000000), E(000000, 111111) \rangle.
\]

The logical Pauli operators are

\[
\begin{align*}
\hat{X}_1 & \triangleq E(110000, 000000) & \hat{Z}_1 & \triangleq E(000000, 010001) \\
\hat{X}_2 & \triangleq E(101000, 000000) & \hat{Z}_2 & \triangleq E(000000, 001001) \\
\hat{X}_3 & \triangleq E(100100, 000000) & \hat{Z}_3 & \triangleq E(000000, 000101) \\
\hat{X}_4 & \triangleq E(100010, 000000) & \hat{Z}_4 & \triangleq E(000000, 000011)
\end{align*}
\]

Consider \(F_{16}\) constructed by adjoining to \(F_2\) a root \(\alpha\) of the primitive polynomial \(p(x) = x^2 + x + 1\). Consider the element \(F_{abcd}\) in \(P_{K,4}\) identified by the tuple \((a = \alpha^3, b = \alpha^8, c = \alpha^7, d = 0)\), or equivalently \((a, b, c, d, i = 0)\) in \(P_{K,4}\). In this case the matrices defined in Section [41-A] are

\[
W = \begin{bmatrix}
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1
\end{bmatrix},
W^{-1} = \begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0
\end{bmatrix},
\]

\[
R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix},
A = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}.
\]

Then, using the isomorphism in Corollary [24] and the direct form in Lemma [23] we can express the element in \(P_{K,4}\) as

\[
F_{abcd} = T_{A_8^2} \cdot L_{E_7^{-2}} \cdot \Omega L_{E_8^{-1}} \cdot T_{A_{211}^2} W
\]

\[
= I_8 \cdot L(A^7)^{-2} \cdot \Omega L(A^8)^{-1} \cdot T(A^{11})^2 W
\]

\[
= \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix},
W_{-1} A^2 = \begin{bmatrix}
A^2 W \\
(A^2)^T H
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix},
W_{-1} A^4 = \begin{bmatrix}
A^{16} W \\
(A^4)^T H
\end{bmatrix}
\]

Using the explicit decomposition in [49], we map the elementary symplectic matrices to standard Clifford gates (as discussed in [42]) and get the following circuit (CKT1)\(^2\)

\[
\begin{align*}
\text{‘Permute’} & \quad [4,1,2,3] \\
\text{‘CNOT’} & \quad [1,2] \\
\text{‘Permute’} & \quad [4,3,2,1] \\
\text{‘CNOT’} & \quad [1,4] \\
\text{‘H’} & \quad [1,2,3,4] \\
\text{‘P’} & \quad [1,2,3,4] \\
\text{‘CZ’} & \quad [1,3] \\
\text{‘CZ’} & \quad [2,4] \\
\text{‘CZ’} & \quad [3,4]
\end{align*}
\]

Here the indices corresponding to ‘Permute’ imply the cycle permutation (1432), i.e., the first qubit has been replaced by

\(^2\text{We list circuits instead of giving their circuit representation to align with the \texttt{MATLAB}® cell array format we adopt in our implementations.}\)
the fourth, the fourth by the third, and the second by the first. (Note that we do not simplify circuits to their optimal form here but simply report the results of our synthesis algorithm.) An alternative procedure is to directly input the final symplectic matrix \( M \) to the symplectic decomposition algorithm in [36] (also see [42, Section II]), yielding the following circuit (CKT2).

\[
\text{‘Permute’} [3,2,1,4], \quad \text{‘CNOT’} [4,3], \quad \text{‘CNOT’} [1,4], \quad \text{‘H’} [1,2,3,4], \quad \text{‘P’} [1,2,3,4], \quad \text{‘CZ’} [1,3], \quad \text{‘CZ’} [2,4], \quad \text{‘CZ’} [3,4].
\]

The difference in depth of the two circuits is very small in this case, but we found that for about half of the elements in \( \mathcal{P}_{k,n-k} \), the explicit form in Corollary [24] had smaller depth, while for those remaining, the direct decomposition was better.

Next we translate this logical circuit into its physical implementation for the \([6, 4, 2]\) CSS code. We apply our synthesis algorithm [42], which can be summarized as follows.

1) Compute the action of either of the above circuits on the Pauli matrices \( X_i, Z_i \), for \( i = 1, 2, 3, 4 \), under conjugation, i.e., compute \( gE(e_i, 0)g^\dagger, gE(0, e_i)g^\dagger \) where \( g \) represents the circuit.
2) Translate these into logical constraints on the desired physical circuit \( \bar{g} \in \text{Cliff}_{6^2} \) by interpreting \( X_i, Z_i \) above as their logical equivalents \( \bar{X}_i, \bar{Z}_i \in \text{HW}_{6^2} \).
3) Rewrite the above conditions as linear constraints on the desired symplectic matrix \( F_\bar{g} \) using [9]. Add constraints to normalize (or just centralize) the stabilizer \( S \).
4) Solve for all symplectic solutions, compute their corresponding circuits and identify the best solution in terms of smallest depth, with respect to the decomposition in [36] (also discussed in [42, Section II]).
5) Verify the constraints imposed in step 2 and check for any sign violations (due to signs in [24]). In case of violations, identify a Pauli matrix to fix the signs.

Using this algorithm, we computed the circuit with smallest depth (relative to other solutions decomposed using the same algorithm) and obtained the following solution (CKT3).

\[
\text{‘Permute’} [1,6,3,5,4,2], \quad \text{‘CNOT’} [3,1], \quad \text{‘CNOT’} [4,1], \quad \text{‘CNOT’} [5,1], \quad \text{‘CNOT’} [6,1], \quad \text{‘CNOT’} [6,4], \quad \text{‘CNOT’} [6,5], \quad \text{‘CNOT’} [4,5], \quad \text{‘CNOT’} [3,6], \quad \text{‘CNOT’} [2,3], \quad \text{‘H’} [1,2,3,4,5,6], \quad \text{‘P’} [3,4,5], \quad \text{‘CZ’} [1,3].
\]

We used the same procedure to translate all 4080 elements of \( \mathcal{P}_{k,n-k} \) into their (smallest depth) physical implementations for the \([6, 4, 2]\) code, and the synthesis took about 25 minutes on a laptop running the Windows 10 operating system (64-bit) with an Intel® Core™ i7-5500U @ 2.40GHz processor and 8GB RAM. Note that for this case each element of \( \mathcal{P}_{k,n-k} \) translates to \( 2^{k(k+1)/2} \) = 8 symplectic solutions during the above procedure, and we need to compare depth after calculating circuits for all solutions. In future work, we will try to optimize directly for depth without computing all solutions as this procedure is expensive for codes with large redundancy \( (k) \). However, since this translation needs to be done only once for a given set \( \mathcal{P}_{k,n-k} \) and an \( [m, m-k] \) stabilizer code, the circuits can be precomputed and stored in memory.

**VIII. Conclusion**

In this paper, we provided a simpler calculation of the Hamming weight distribution of the classical binary non-linear Kerdock codes by appealing to the connection with maximal commutative subgroups of the Heisenberg-Weyl group from the quantum domain. Using this connection, we also described the group of Clifford symmetries of the mutually unbiased bases determined by the Kerdock sets of symmetric matrices. We then used this result in the classical domain to construct small unitary 2-designs. Finally, we demonstrated an efficient procedure for translating any unitary 2-design consisting of Clifford elements into a logical unitary 2-design for a given stabilizer code. This work demonstrates yet again that interactions between the classical and quantum domains are still mutually beneficial, both theoretically and practically.

**References**

[1] T. Can, N. Rengaswamy, R. Calderbank, and H. D. Pfister, “Kerdock codes determine unitary 2-designs,” in Proc. IEEE Int. Symp. Inform. Theory, pp. 2908–2912, 2019.

[2] S. Bravyi, D. Gosset, and R. König, “Quantum advantage with shallow circuits,” Science, vol. 362, no. 6412, pp. 308–311, 2018.

[3] E. Conover, “Quantum computers are about to get real,” Science News, vol. 191, no. 13, p. 28, 2017. https://www.sciencenews.org/article/quantum-computers-are-about-get-real.

[4] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” Phys. Rev. A, vol. 52, no. 4, pp. R2493–R2496, 1995.

[5] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” Phys. Rev. A, vol. 54, pp. 1098–1105, Aug 1996.

[6] A. M. Steane, “Simple quantum error-correcting codes,” Phys. Rev. A, vol. 54, no. 6, pp. 4741–4751, 1996.
Now we have the following simplifications for the three terms.

\[
I_m \rightarrow A^2_{w}W + R^{-i}\left( A^{2\rightarrow i}_{y+x} + A^{2\rightarrow i}_{wz} \right) W
\]

\[
I_m \rightarrow A^2_{w}W + R^{-i}\left( A^{2\rightarrow i}_{y+x} + A^{2\rightarrow i}_{wz} \right) W
\]

\[
I_m \rightarrow \left( A^2_{w} + A^2_{(y+z)^{-2\rightarrow i}} \right) W
\]

\[
I_m \rightarrow A^2_{w}W + R^{-i}\left( A^{2\rightarrow i}_{y+x} + A^{2\rightarrow i}_{wz} \right) W
\]

Now we define \( x \triangleq c, w \triangleq \frac{a}{c} \) and \( y \triangleq \left( \frac{a}{c} \right)^{2i} \). Then we get

\[
w + (y + z)^{-2\rightarrow i} = \frac{a}{c} \left( z + \left( \frac{d}{c} \right)^2 \right)^{-2i} \frac{1}{c^2}
\]

\[
= \frac{a}{c} \left( z^{2i} - \frac{d}{c} \left( 2^{i} \cdot 2^{-i} \right) \right)^{-1} \frac{1}{c^2}
\]

\[
= \frac{1}{c} \left[ a + \frac{1}{c^{2i+1} + c \cdot \frac{a}{c}} \right]
\]

\[
= \frac{1}{c} \left[ a z^{2i} + ad + 1 \right]
\]

\[
= \frac{a z^{2i} + (ad + 1)}{c^{2i+1} + d}
\]

\[
= \frac{a z^{2i} + b}{c^{2i+1} + d} ; b = \frac{ad + 1}{c}
\]

Hence we have proved that \( F \) performs the permutation

\[
[I_m | A^2_{w}W] \mapsto \left[ I_m | A^2_{wz}W \right], \quad z' \triangleq z^{2^{-i}}
\]

We note that the above definitions for \( x, w, y \) also satisfy the special case of \( i = 0 \) that corresponds to the proof in Lemma 23. We now substitute these back in \( F \) and observe the following simplifications.

\[
(A^2_{y}R^{-i} A^2_{y}) A^2_{w} = R^{-i} A^2_{y+w} W + A^2_{wz}(R^i)T(A^2_{w})T
\]

\[
(A^2_{y}R^{-i} A^2_{wz} + R^{-i} A^2_{wz}) W = R^{-i} A^2_{y+w} W + A^{-i} A_{wz}^2 W
\]

\[
(R^i)T(A^2_{wz})T = (R^i)T(A^2_{wz})T
\]

These imply that the general form of an element in \( \mathfrak{U}_{k,m} \) is

\[
F = \left[ \begin{array}{c}
I_m | A^2_{w}W \\
I_m | A^2_{wz}W
\end{array} \right] = \left[ \begin{array}{c}
\left( A^2_{y}R^{-i} A^2_{wz} + R^{-i} A^2_{wz} \right) W \\
\left( R^i \right)^T(A^2_{wz})T
\end{array} \right]
\]

\[
= \left[ \begin{array}{c}
R^{-i} A^2_{y} \\
R^{-i} A^2_{wz}
\end{array} \right] \left[ \begin{array}{c}
\left( A^2_{y}R^{-i} A^2_{wz} + R^{-i} A^2_{wz} \right) W \\
\left( R^i \right)^T(A^2_{wz})T
\end{array} \right]
\]

\[
= \left[ \begin{array}{c}
R^{-i} A^2_{y} \\
R^{-i} A^2_{wz}
\end{array} \right] \left[ \begin{array}{c}
\left( A^2_{y}R^{-i} A^2_{wz} + R^{-i} A^2_{wz} \right) W \\
\left( R^i \right)^T(A^2_{wz})T
\end{array} \right]
\]
**Trung Can**, a native of Ho Chi Minh City, Vietnam, received a B.A. in Math from Duke University in 2018 and is currently a Ph.D. student at the California Institute of Technology. 

In high school, he received a Gold medal at the International Mathematical Olympiad, representing his home country, Vietnam. At Duke, he served as the vice-president of the Math Union. His undergraduate thesis on the applications of the Heisenberg-Weyl group and finite symplectic geometry received the highest distinction from the math department. He founded and currently serves as the president of PiMA, a nonprofit organization that organizes summer camps and seminars on math topics for high school students in Vietnam. 

His current research interests are algebraic geometry and number theory; in particular, the Schottky problem and Shimura varieties.

**Narayanan Rengaswamy** received his Ph.D. in Electrical Engineering from Duke University, USA, in May 2020 where he was supervised by Prof. Henry Pfister and Prof. Robert Calderbank. He is currently a research associate at Duke University continuing to work with his Ph.D. supervisors. His primary training is in classical information and coding theory, and compressed sensing. During his Ph.D. he actively learned about quantum computing and quantum error correction and this is now his primary area of research. Prior to this, he received his Master of Science (M.S.) degree, also in Electrical Engineering, from Texas A&M University, USA, in December 2015. Earlier he obtained his Bachelor of Technology (B. Tech.) degree in Electronics and Communication Engineering from Amrita University, Coimbatore, India in May 2013, where he ranked first in the college and third in the university. 

He has studied the problem of realizing logical operations on quantum error correcting codes in a fault-tolerant manner and produced new insights via a perspective rooted in classical coding theory. His paper on developing conditions for a general stabilizer code to support transversal T gates was one among the 73 out of 283 submissions that were accepted as talks to the esteemed 2020 Quantum Information Processing (QIP) conference. 

His recent work on Joseph Renes’ belief propagation algorithm that passes quantum messages provides a potential near-term application with a quantum advantage for a special-purpose all-photonic quantum processor. In general, he is excited about problems in the intersection of classical and quantum information processing and mathematical tools that help address problems of this sort. He has been a reviewer for IEEE Transactions on Information Theory, IEEE Transactions on Vehicular Technology, Quantum, Quantum Science and Technology, and Proceedings of the Royal Society A, besides the conferences IEEE International Symposium on Information Theory (2018, 2020) and IEEE Information Theory Workshop (2018). He also helped organize the 2016 North American School of Information Theory at Duke University.

**Dr. Calderbank** served as Editor in Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY from 1995 to 1998, and as Associate Editor for Coding Techniques from 1986 to 1989. He was a member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996 and from 2006 to 2008. Dr. Calderbank was honored by the IEEE Information Theory Prize Paper Award in 1995 for his work on the Z4 linearity of Kendock and Preparata Codes (joint with A.R. Hammons Jr., P.V. Kumar, N.J.A. Sloane, and P. Sole), and again in 1999 for the invention of space-time codes (joint with V. Tarokh and N. Seshadri). He has received the 2006 IEEE Donald G. Fink Prize Paper Award, the IEEE Millennium Medal, the 2013 IEEE Richard W. Hamming Medal, and the 2015 Shannon Award. He was elected to the US National Academy of Engineering in 2005.

**Henry Pfister** received his Ph.D. in Electrical Engineering in 2003 from the University of California, San Diego and is currently a professor in the Electrical and Computer Engineering Department of Duke University with a secondary appointment in Mathematics. Prior to that, he was an associate professor at Texas A&M University (2006-2014), a post-doctoral fellow at the Ecole Polytechnique Fédérale de Lausanne (2005-2006), and a senior engineer at Qualcomm Corporate R&D in San Diego (2003-2004). His current research interests include information theory, error-correcting codes, quantum computing, and machine learning.

He received the NSF Career Award in 2008 and a Texas A&M ECE Department Outstanding Professor Award in 2010. He is a coauthor of the 2007 IEEE COMSOC best paper in Signal Processing and Coding for Data Storage and a coauthor of a 2016 Symposium on the Theory of Computing (STOC) best paper. He has served the IEEE Information Theory Society as a member of the Board of Governors (2019-2022), an Associate Editor for the IEEE Transactions on Information Theory (2015-2016), and a Distinguished Lecturer (2015-2016). He was also the General Chair of the 2016 North American School of Information Theory.

**Robert Calderbank** (M’89–SM’97–F’98) received the B.S. degree in 1975 from Warwick University, England, the M.S. degree in 1976 from Oxford University, England, and the Ph.D. degree in 1980 from the California Institute of Technology, all in Mathematics. Dr. Calderbank is Professor of Electrical and Computer Engineering at Duke University where he directs the Rhodes Information Initiative at Duke (i2D). Prior to joining Duke in 2010, Dr. Calderbank was Professor of Electrical Engineering and Mathematics at Princeton University. Prior to joining Princeton in 2004, he was Vice President for Research at AT&T, responsible for directing the first industrial research lab in the world where the primary focus is data at scale. At the start of his career at Bell Labs, innovations by Dr. Calderbank were incorporated in a progression of voiceband modem standards that moved communications practice close to the Shannon limit. Together with Peter Shor and colleagues at AT&T Labs he developed the mathematical framework for quantum error correction. He is a co-inventor of space-time codes for wireless communication, where correlation of signals across different transmit antennas is the key to reliable transmission.