The Classification of Clifford Gates over Qubits

Daniel Grier[1] and Luke Schaeffer[2]

[1]University of Waterloo, Cheriton School of Computer Science
[2]University of Waterloo, Department of Combinatorics and Optimization

We examine the following problem: given a collection of Clifford gates, describe the set of unitaries generated by circuits composed of those gates. Specifically, we allow the standard circuit operations of composition and tensor product, as well as ancillary workspace qubits as long as they start and end in states uncorrelated with the input, which rule out common "magic state injection" techniques that make Clifford circuits universal. We show that there are exactly 57 classes of Clifford unitaries and present a full classification characterizing the gate sets which generate them. This is the first attempt at a quantum extension of the classification of reversible classical gates introduced by Aaronson et al., another part of an ambitious program to classify all quantum gate sets.

The classification uses, at its center, a reinterpretation of the tableau representation of Clifford gates to give circuit decompositions, from which elementary generators can easily be extracted. The 57 different classes are generated in this way, 30 of which arise from the single-qubit subgroups of the Clifford group. At a high level, the remaining classes are arranged according to the bases they preserve. For instance, the CNOT gate preserves the X and Z bases because it maps X-basis elements to X-basis elements and Z-basis elements to Z-basis elements. The remaining classes are characterized by more subtle tableau invariants; for instance, the T_4 and phase gate generate a proper subclass of Z-preserving gates.

1 Introduction

A common thread throughout quantum computing is the manner in which a few elementary gates often suffice for universal computation. This "pervasiveness of universality" is explored in recent work of Aaronson, Grier, and Schaeffer [1]. There, the authors give a complete classification of *classical* reversible gates in terms of the functions over bits they generate and find that a rich structure emerges.

Of course, the ultimate goal would be a complete classification of *quantum* gate sets based on the functions over qubits they generate. Unfortunately, not even a full classification of the subgroups of a three-qubit system is known.[1] Since each class of gates is a subgroup, this suggests that a complete classification remains out of reach. This might be surprising given how well we understand random gate sets, and even those that contain particular gates such as CNOT

Daniel Grier: daniel.grier@uwaterloo.ca
Luke Schaeffer: lrschaeffer@gmail.com

[1]The difficulty in classifying the subgroups of $\mathrm{SU}(N)$ arises not from the infinite classes but from the finite ones. In fact, even the finite subgroups of $\mathrm{SU}(5)$ remain unclassified. This motivates our focus on finite, discrete classes such as the Clifford group.

[2, 3]. However, a full classification begets a complete understanding of *all* possible behaviors, despite their strangeness or rarity (see, for example, the sporadic gate sets in the lattice of Aaronson et al. [1]). Nevertheless, there has been some encouraging progress on classification problems: for Hamiltonians, Bouland, Mančinska, and Zhang [4] classified all 2-qubit commuting Hamiltonians, while Childs et al. [5] characterized all 2-qubit Hamiltonians when restricted to circuits over two qubits; for linear optics, Aaronson and Bouland [6] completed a classification for linear optics of 2-mode beamsplitters; and for Clifford+T circuits, Amy, Glaudell, and Ross [7] give a type of classification based on the elements appearing in their representations as unitary matrices.

This paper contributes a new classification of quantum gate sets by giving a complete classification of the Clifford gates, the set of unitaries normalizing the Pauli group. To provide some context, the Clifford gates are generated by the CNOT gate, the Hadamard gate, and the $\frac{\pi}{4}$-phase gate. It is not hard to see that the Clifford operations on n qubits are a discrete, finite set, so it has always been widely assumed that they do not suffice for universal quantum computation. Indeed, Gottesman and Knill [8] showed that they could be efficiently simulated with a classical computer, using a binary matrix representation of states called a *stabilizer tableau*. Using a slightly larger version of these tableaux, Aaronson and Gottesman [9] were able to further improved the efficiency of measurements in this algorithm.[2] In fact, a reinterpretation of their tableau representation is integral to our classification.

Clifford circuits are somewhat remarkable in that they may in fact be integral to our eventual development of a general-purpose quantum computer. For instance, the stabilizer formalism, which tracks state evolution through conjugated Pauli elements, underlies many of the important quantum error correcting codes [10]. In fact, the Clifford operations are exactly those operations which can be easily computed transversally in many fault-tolerant schemes of quantum computing (e.g., the Shor code [11] or the [[7,1,3]] Steane code [12]).

Our model is motivated in part by the use of Clifford circuits as subroutines of a general quantum computation, much like the transversal gates in a fault-tolerant scheme. We regard the creation of complicated ancilla states as an inherently difficult task, and therefore require that all ancillary qubits used during the computation be returned to their initial state at the end of the computation. This restriction eliminates schemes in which much of the difficulty of the computation is offloaded to the creation of "magic states" which are subsequently consumed by the computation [11, 13, 14]. Unlike these schemes in which the ancillas boost weak gate sets to computational universality, Clifford operations cannot be boosted in our model to generate anything outside the Clifford group.[3]

We also regard the classification of Clifford gates as an important step towards a full classification of quantum gate classes. Although the complete inclusion lattice for general quantum gates will be significantly more complicated than the one we present for the Clifford gates, the classes described here provide a testbed for the techniques used for general quantum gates. This is due to the fact that our lattice for Clifford gates must appear as a sublattice in the complete quantum gate classification. This situation contrasts with the reversible gate classification of Aaronson et al., in which much of the complexity of the lattice is due to the fact that only $|0\rangle$ and $|1\rangle$ ancillas were allowed. Indeed, we show in Appendix B that the classical reversible classification collapses significantly under quantum ancillas. Because we allow for arbitrary ancillas in our model, our classification does not suffer from the same issue.

[2] In terms of complexity classes, Aaronson and Gottesman [9] show that Clifford circuits can be simulated in the class $\oplus\mathsf{L}$ (pronounced "parity ell"). This class is most easily understood as capturing those problems reducible to solving linear equations mod 2. To be clear, we have that $\oplus\mathsf{L} \subseteq \mathsf{P}$.

[3] This was first noticed by Anderson [15].

1.1 Results

We wish to determine the set of Clifford operations that can be realized as circuits consisting of gates from a given gate set. Let us briefly explain the circuit building operations we allow (full details and justification in Section 5). First, we can combine gates in series or parallel, i.e., their composition or tensor product. We also assume for simplicity that swapping qubits is allowed at any point in the circuit. Each circuit also has access to arbitrary quantum ancillas provided that they are returned to their initial states by the end of the computation. Finally, we adopt the standard practice of ignoring global phase in circuits. Under this model, our main result is the classification of Clifford gates below:

Theorem 1. *Any set of Clifford gates generate one of 57 distinct classes of Clifford operations. There are 30 classes (depicted in Figure 1) generated by single-qubit gates. The remaining 27 non-degenerate classes are shown in Figure 2. Notation for the generators of the classes depicted in those diagrams is given in Section 3.*

We list some consequences and highlights of the classification below:

(1) **Invariants.** Every class can be defined by a collection of invariants, i.e., properties of the Clifford gates which are preserved under our circuit building operations. Formally, we define each invariant based on the tableau representation of the Clifford gate (see sections 4 and 6). We now describe the broad themes behind the main invariants of the classification. First, there is a three-fold symmetry in the classification, corresponding to the symmetry of the X, Y, and Z elements of the Pauli group. For example, the CNOT gate behaves classically in the X-basis and Z-basis (i.e., it permutes the four 2-qubit X-basis vectors, and likewise in the Z-basis), but not the Y-basis. By symmetry, there are gates like CNOT which are classical in each pair of bases. There is even a nontrivial class of gates which act classical in all three bases (up to sign).

When two classes cannot be distinguished by their high-level basis behavior, we need more refined invariants to separate them. For example, some invariants correspond to the specific action the gate has in some basis: the CNOT gate can generate any reversible linear transformation in the Z-basis; there is another gate which can only perform *orthogonal* linear transformations; and we also have gates like CZ, which can only change the sign of the basis element.[4] In fact, these three cases and the available collection of single-qubit gates are enough to determine the class.

(2) **Finite Generation.** Every class can be generated by a single gate on at most four qubits. Also, given a set of gates generating some class, there always exist three gates from that set that generate the same class. Moreover, the classification implies that the canonical set of Clifford generators—CNOT, Hadamard, and phase—is *not* a minimal set of generators in our model. It turns out that with the aid of ancilla qubits, CNOT and Hadamard generate a phase gate. This is well-known [16, 17], but comes as a simple consequence of our classification theorem.

(3) **Ancillas.** In general, giving a Clifford gate access to ancillary qubits often increases the set of functions it can compute. A priori, one might suspect that extracting all functionality from a large entangling Clifford gate would require large highly-entangled ancilla states. Nevertheless, our classification shows that only a constant number of one- and two-qubit ancillary states are ever needed. In fact, an even stronger result is true. Namely, our

[4] The CNOT (controlled-X) gate is sometimes written as CX, and we similarly write the controlled-Z gate as CZ.

classification holds even when we allow the ancillas to change in an input-independent manner,[5] as would be natural for a Clifford subroutine in a general quantum computation. See Section 5 for further discussion.

(4) **Canonical Forms.** It has long been known that there exists canonical forms for Clifford circuits [9, 18–20]. Our classification theorem also reveals explicit canonical forms for most *subclasses* of Clifford circuits (see Section 7). Furthermore, we give an explicit canonical form for 2-qubit Clifford circuits using the generators from our classification (see Appendix E).

(5) **Enumeration.** For each class $\mathcal{C}$ and for all n, we give explicit formulas for the number of gates in $\mathcal{C}$ on n-qubits. This enumeration is often derived from our explicit canonical forms discussed above. One consequence is that every class is exponentially smaller than any class strictly containing it. See Appendix A for details.

(6) **Algorithms.** Our classification implies a linear time algorithm which, given the tableau of a gate G, identifies which class G belongs to. In fact, to witness that G generates some class in the classification, one only needs to view a constant number of bits of the tableau. These details are discussed in sections 9 and 10.

(7) **Sporadic Gates.** The process of classification unearthed certain strange classes, which arise from the interaction of the various invariants. For example, four of the classes containing the T_4 gate require a generator on at least four qubits. Surprisingly, the class $\langle \mathrm{T}_4, \Gamma, \mathcal{P} \rangle$ has a three-qubit generator.[6] We investigate such a gate in Appendix C.

[5] Aaronson et al. called this the "Loose Ancilla Rule," and it *does* affect their classification of classical reversible circuits.

[6] Interestingly, there are no affine gate sets in the classification of *classical* reversible gates which admit a generator over three bits and no smaller.

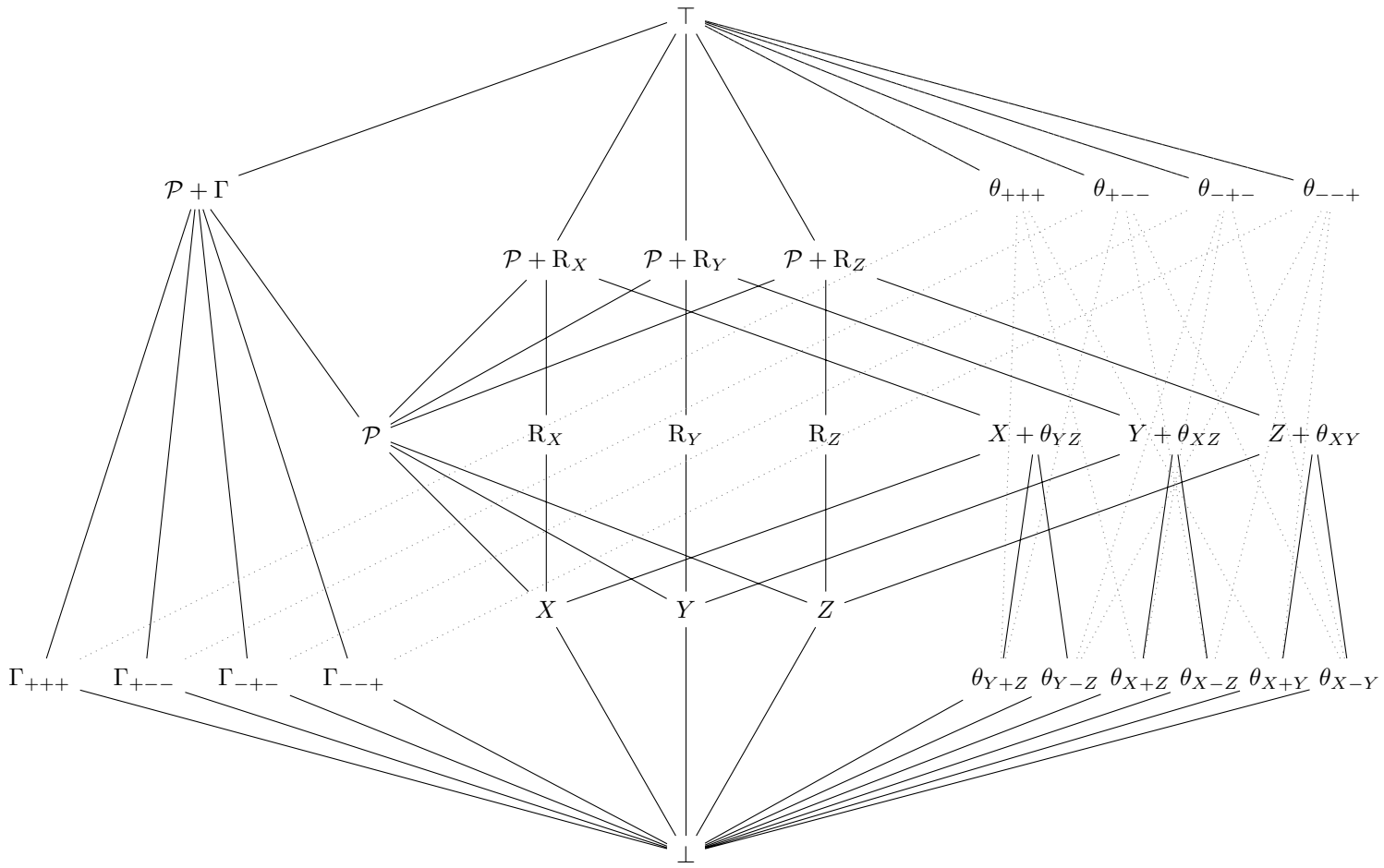


Figure 1: The inclusion lattice of degenerate Clifford gate classes. To improve readability, some lines are gray and dotted, but they have the same meaning as their counterparts. The gates in this diagram are listed in Section 3.1 and some class names have been abbreviated as listed in Section 5.

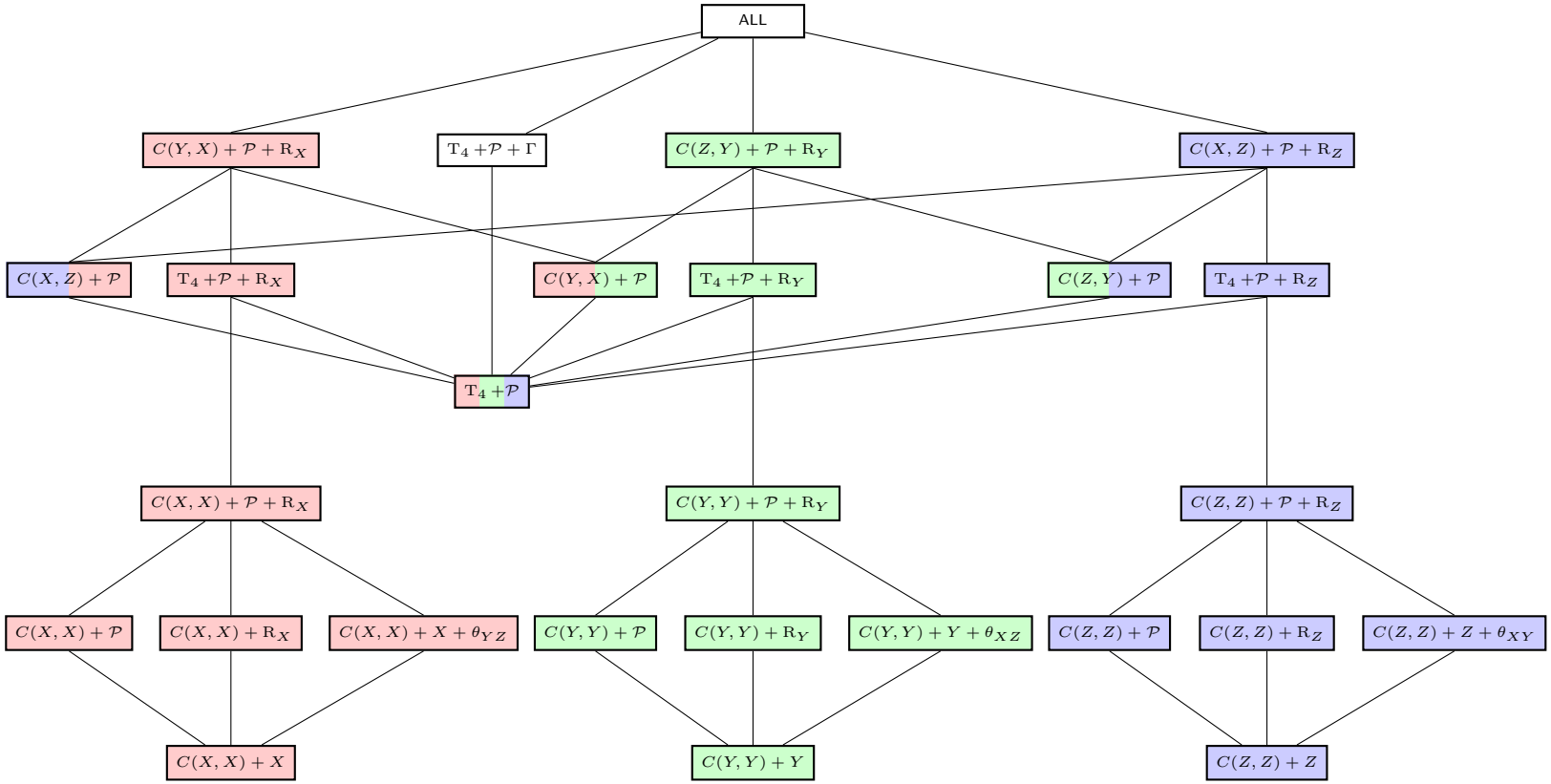


Figure 2: The inclusion lattice of non-degenerate Clifford gate classes. Red, green, blue denote X-, Y-, and Z-preserving, respectively.

1.2 Proof Outline

We can divide the proof into a few major steps. First, we introduce the notion of a tableau, a binary matrix representation of a Clifford circuit. We then present all the classes in the classification and designate them by their generators. An examination of the tableaux of the gates in these classes reveals candidate invariants. We then prove that these candidate invariants are indeed invariant under the circuit building operations in our model. That is to say, if we have two gates whose tableaux satisfy the invariant, then the tableau of their composition satisfies the invariant, and so on for all the other ways to build circuits from gates—tensor products, ancillas, swapping.

At this point, we will have shown that each class has a corresponding invariant, which implies that each class in our lattice is distinct. That is, for any two classes, there is a generator of one that fails to satisfy the invariant of the other. Next, we will show that this correspondence is complete. The generators of a class can construct *any* gate which satisfies the invariant for that class.

The challenge remains to show that our list of classes is exhaustive. Suppose we are given some gate set G, and we wish to identify the class it generates. Clearly, the class generated by G is contained in some class in the lattice, and let $\mathcal{C}$ be the smallest such class. The hope is to show that G generates all of $\mathcal{C}$. To do this, we use the minimality of $\mathcal{C}$. That is, for each class $\mathcal{S} \subset \mathcal{C}$, there must be some gate $g \in G$ which is not in $\mathcal{S}$, otherwise $\mathcal{S}$ would be a smaller class containing G. We now wish to use g to generate a simpler gate, also violating some invariant of $\mathcal{S}$. This is accomplished via the "universal construction," which is a particular circuit built from g and SWAP gates. Finally, we combine the simpler gates to construct the canonical generators for the class $\mathcal{C}$ itself.

2 Stabilizer Formalism

The one-qubit unitary operations

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

are known as *Pauli matrices*. The Pauli matrices are all involutions ($X^2 = Y^2 = Z^2 = I$), and have the following relations between them

$$\begin{aligned} XY &= iZ & YZ &= iX & ZX &= iY \\ YX &= -iZ & ZY &= -iX & XZ &= -iY. \end{aligned}$$

It follows that the Pauli matrices generate a discrete group (under multiplication), called the *Pauli group* $\mathcal{P}$, which consists of sixteen elements: $\{I, X, Y, Z\}$ with phases ± 1 or $\pm i$. The *Pauli group on n qubits*, $\mathcal{P}_n$, is the set of all n-qubit tensor products of elements from $\mathcal{P}$. We define a *Pauli string* as any element of $\mathcal{P}_n$ with positive phase (i.e., a tensor product of the matrices I, X, Y, Z). We frequently omit the tensor product symbol from Pauli strings and write, e.g., $P_1 \cdots P_n$ where we mean $P_1 \otimes \cdots \otimes P_n$.

The *Clifford group on n qubits*, $\mathcal{C}_n$, is the set of unitary operations which *normalize* $\mathcal{P}_n$ in the group-theoretic sense. That is, $U \in \mathcal{C}_n$ if $UpU^\dagger \in \mathcal{P}_n$ for all $p \in \mathcal{P}_n$. We leave it as a simple exercise to check that $\mathcal{C}_n$ is indeed a group.

A *Clifford gate* is any unitary in $\bigcup_{n \geq 1} \mathcal{C}_n$. A *Clifford circuit* is a quantum circuit of Clifford gates implementing a unitary transformation on some set of qubits, designated the *input/output qubits*, while preserving the state of the remaining *ancilla qubits*. We say that a state $|\psi\rangle$ is

stabilized by an operation U if and only if $U|\psi\rangle = |\psi\rangle$. In other words, $|\psi\rangle$ is in the $+1$ eigenspace of U. The Pauli elements and their corresponding stabilized states are below:

$$\begin{array}{ll} X : |+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}} & -X : |-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}} \\ Y : |i\rangle = \frac{|0\rangle+i|1\rangle}{\sqrt{2}} & -Y : |-i\rangle = \frac{|0\rangle-i|1\rangle}{\sqrt{2}} \\ Z : |0\rangle & -Z : |1\rangle \end{array}$$

We call the vectors stabilized by non-identity Pauli elements P and $-P$ the *P-basis*. A *stabilizer state* is any state $U|0\dots0\rangle$ where U is a Clifford gate. For example, $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$, $|i\rangle$, and $|-i\rangle$ are the 6 stabilizer states on one qubit. Multi-qubit stabilizer states include $\frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$ and $\sum_{x\in\{0,1\}^n} |x\rangle$. In general, stabilizer states are of the form (unnormalized) $\sum_{x\in A}(-1)^{q(x)} i^{\ell(x)}|x\rangle$ where A is an affine space over $\mathbb{F}_2$, $q(x)$ is a quadratic form, and $\ell(x)$ is a linear form [21, 22].

3 Gates

Let us introduce some common Clifford gates used throughout the classification.

3.1 Single-qubit Gates

We start with the single-qubit Clifford gates, which by definition permute (up to phases) the X, Y, and Z bases. In fact, the single-qubit Clifford gates correspond to symmetries of the cube (see Figure 3).[7] We group the gates by the type of rotation to emphasize this geometric intuition.

Face rotations: The Pauli matrices X, Y, and Z (as gates) correspond to 180° rotations about the X, Y, and Z axes respectively. Similarly, we define R_X, R_Y, and R_Z to be 90° rotations (in the counterclockwise direction) about their respective axes. Formally,

$$\mathrm{R}_X = \frac{I - iX}{\sqrt{2}}, \qquad \mathrm{R}_Y = \frac{I - iY}{\sqrt{2}}, \qquad \mathrm{R}_Z = \frac{I - iZ}{\sqrt{2}},$$

although in the case of R_Z (also known as the *phase gate* and often denoted by S or P), a different choice of phase is more conventional. The clockwise rotations are then $\mathrm{R}_X^\dagger$, $\mathrm{R}_Y^\dagger$, and $\mathrm{R}_Z^\dagger$.

Edge rotations: Another symmetry of the cube is to rotate one of the edges 180°. Opposing edges produce the same rotation, so we have six gates: θ_{X+Y}, θ_{X-Y}, θ_{X+Z}, θ_{X-Z}, θ_{Y+Z}, θ_{Y-Z}. We define

$$\theta_{P+Q} = \frac{P + Q}{\sqrt{2}}, \qquad \theta_{P-Q} = \frac{P - Q}{\sqrt{2}},$$

for all Pauli matrices $P \neq Q$. Note that θ_{X+Z} is the well-known *Hadamard gate*, usually denoted by H.

Vertex rotations: The final symmetry is a 120° counterclockwise rotation around one of the diagonals passing through opposite vertices of the cube. The cube has eight vertices,

[7] Or equivalently, symmetries of the octahedron, which is dual to the cube.

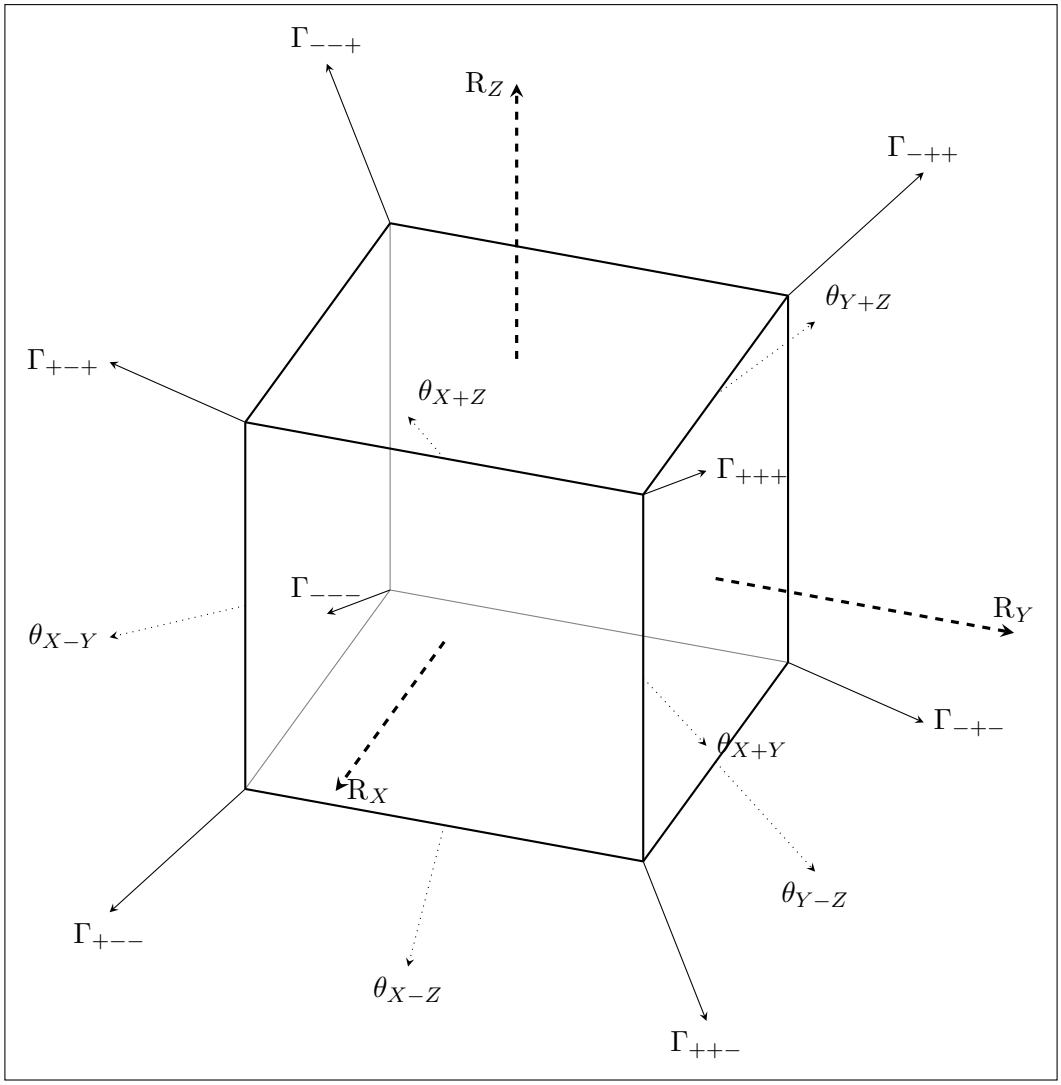


Figure 3: Single-qubit gates as symmetries of the cube.

$(\pm 1, \pm 1, \pm 1)$, and we denote the corresponding single-qubit gates Γ_{+++}, Γ_{++-}, …, Γ_{---}. Algebraically, we define

$$\begin{aligned}
\Gamma_{+++} &= \frac{I - iX - iY - iZ}{2}, \\
\Gamma_{++-} &= \frac{I - iX - iY + iZ}{2}, \\
&\vdots \\
\Gamma_{---} &= \frac{I + iX + iY + iZ}{2}.
\end{aligned}$$

We also define Γ (without subscripts) to be the first gate, Γ_{+++}, since it is the most convenient; conjugation by Γ maps X to Y, Y to Z, and Z to X.

Gate	Tableau	Unitary Matrix
X	$\left(\begin{array}{cc\|c} 1 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right)$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Y	$\left(\begin{array}{cc\|c} 1 & 0 & 1 \\ 0 & 1 & 1 \end{array}\right)$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Z	$\left(\begin{array}{cc\|c} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right)$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
R_X	$\left(\begin{array}{cc\|c} 1 & 0 & 0 \\ 1 & 1 & 1 \end{array}\right)$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$
R_Y	$\left(\begin{array}{cc\|c} 0 & 1 & 1 \\ 1 & 0 & 0 \end{array}\right)$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$
$\mathrm{R}_Z = S$	$\left(\begin{array}{cc\|c} 1 & 1 & 0 \\ 0 & 1 & 0 \end{array}\right)$	$\frac{1-i}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
$\theta_{XZ} = \theta_{X+Z} = H$	$\left(\begin{array}{cc\|c} 0 & 1 & 0 \\ 1 & 0 & 0 \end{array}\right)$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
θ_{X-Z}	$\left(\begin{array}{cc\|c} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right)$	$\frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$
$\Gamma_{+++} = \Gamma$	$\left(\begin{array}{cc\|c} 1 & 1 & 0 \\ 1 & 0 & 0 \end{array}\right)$	$\frac{1-i}{2}\begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}$

Table 1: Single-qubit gates represented as tableaux (Section 4) and as complex unitary matrices.

3.2 Multi-qubit Gates

We now introduce the multi-qubit Clifford gates relevant to the classification.[8] The *SWAP gate*, for instance, simply exchanges two qubits. A more interesting example is the *controlled-NOT* or *CNOT gate*, and the *generalized CNOT gates*.

A *generalized CNOT gate* is a two-qubit Clifford gate of the form

$$C(P,Q) := \frac{I \otimes I + P \otimes I + I \otimes Q - P \otimes Q}{2},$$

where P and Q are Pauli matrices. If the first qubit is in the $+1$ eigenspace of P then $C(P,Q)$ does nothing, but if it is in the -1 eigenspace of P then $C(P,Q)$ applies Q to the second qubit. Of course, the definition is completely symmetric, so you can also view it as applying P to the first qubit when the second qubit is in the -1 eigenspace of Q.

Observe that $C(Z,X)$ is actually the CNOT gate; it applies a NOT gate to the second qubit when the first qubit is $|1\rangle$ and does nothing when the first qubit is $|0\rangle$. Figure 4 shows this equivalence, and illustrates our circuit diagram notation for generalized CNOT gates. Also note that $C(X,Z)$ is a CNOT, but with the opposite orientation (i.e., the second bit controls the first). The rest of the *heterogeneous generalized CNOT gates* (i.e., $C(P,Q)$ where $P \neq Q$) are the natural equivalents of CNOT in different bases.

Similarly, $C(Z,Z)$ sometimes known as the *controlled-sign gate* or CZ, which flips the sign on input $|11\rangle$, but does nothing otherwise. The *homogeneous generalized CNOT gates* (i.e., $C(P,P)$

[8] Like the single-qubit gates, it turns out that the two-qubit Clifford gates can also be interpreted as symmetries of a polyhedron, in particular, the six-dimensional hyperoctahedron [23, 24]. However, we find it easier to reason about the group from a set of simple generators.

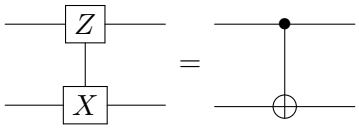


Figure 4: CNOT expressed as a $C(Z, X)$ gate.

for some P) are quite different from heterogeneous CNOT gates. For instance, when one CNOT targets the control qubit of another CNOT then it matters which gate is applied first. On the other hand, two CZ gates will always commute, whether or not they have a qubit in common.

It turns out that every two-qubit Clifford gate is equivalent (up to a SWAP) to some combination of one qubit Clifford gates and at most one generalized CNOT gate (see Appendix E). Although most classes of Clifford gates can be specified by such two-qubit generators, there are five classes which require a larger generator such as the T_4 gate [1, 25].

For all $k \geq 1$, let T_{2k} be a $2k$-qubit gate such that for all $x = (x_1, \ldots, x_{2k}) \in \{0,1\}^{2k}$

$$\mathrm{T}_{2k} |x_1, \ldots, x_{2k}\rangle = |x_1 \oplus b_x, x_2 \oplus b_x, \ldots, x_{2k} \oplus b_x\rangle$$

where $b_x = x_1 \oplus x_2 \oplus \ldots \oplus x_{2k}$. Intuitively, T_{2k} outputs the complement of the input when the parity of the input is odd and does nothing when the parity of the input is even. In particular, T_2 is the lowly SWAP gate. Notice that this is an orthogonal linear function of the input bits, and therefore has a $2k \times 2k$ matrix over $\mathbb{F}_2$ with orthogonal rows and columns:

$$\mathrm{T}_{2k} = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 0 \end{pmatrix}$$

4 Tableaux

Observe that the matrices I, X, Y, Z are linearly independent, and therefore form a basis for the 2×2 complex matrices. It follows that $\mathcal{P}_n$ spans all $2^n \times 2^n$ complex matrices. Hence, any unitary operation on n qubits can be characterized by its action on $\mathcal{P}_n$. In particular, any gate is characterized by how it acts on

$$p_1 = XI \cdots I,\ p_2 = ZI \cdots I,\ \ldots,\ p_{2n-1} = I \cdots IX,\ p_{2n} = I \cdots IZ.$$

We call this list the *Pauli basis on n qubits*, since one can write any element of $\mathcal{P}_n$ as a product of basis elements times a phase (± 1 or $\pm i$).

Now suppose we are given a Clifford gate, $U \in \mathcal{C}_n$. By definition, Clifford gates map each Pauli basis element to something in $\mathcal{P}_n$, which can be written as a product of basis elements times a phase. That is,

$$U p_j U^\dagger = \alpha_j \prod_{k=1}^{2n} p_k^{M_{jk}}$$

Gate	Tableau	Unitary Matrix on computational basis
$C(X,X)$	$\left(\begin{array}{cc\|cc} 1&0&0&0\\ 0&1&1&0\\ \hline 0&0&1&0\\ 1&0&0&1 \end{array}\right)$	$\frac{1}{2}\begin{pmatrix} 1&1&1&-1\\ 1&1&-1&1\\ 1&-1&1&1\\ -1&1&1&1 \end{pmatrix}$
$C(Y,Y)$	$\left(\begin{array}{cc\|cc} 1&0&1&1\\ 0&1&1&1\\ \hline 1&1&1&0\\ 1&1&0&1 \end{array}\right)$	$\frac{1}{2}\begin{pmatrix} 1&-i&-i&1\\ i&1&-1&-i\\ i&-1&1&-i\\ 1&i&i&1 \end{pmatrix}$
$C(Z,Z)$	$\left(\begin{array}{cc\|cc} 1&0&0&1\\ 0&1&0&0\\ \hline 0&1&1&0\\ 0&0&0&1 \end{array}\right)$	$\begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&-1 \end{pmatrix}$
$C(Y,X)$	$\left(\begin{array}{cc\|cc} 1&0&1&0\\ 0&1&1&0\\ \hline 0&0&1&0\\ 1&1&0&1 \end{array}\right)$	$\frac{1}{2}\begin{pmatrix} 1&1&-i&i\\ 1&1&i&-i\\ i&-i&1&1\\ -i&i&1&1 \end{pmatrix}$
$C(Z,Y)$	$\left(\begin{array}{cc\|cc} 1&0&1&1\\ 0&1&0&0\\ \hline 0&1&1&0\\ 0&1&0&1 \end{array}\right)$	$\begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&-i\\ 0&0&i&0 \end{pmatrix}$
$C(X,Z) = \mathrm{CNOT}$	$\left(\begin{array}{cc\|cc} 1&0&0&0\\ 0&1&0&1\\ \hline 1&0&1&0\\ 0&0&0&1 \end{array}\right)$	$\begin{pmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0 \end{pmatrix}$

Table 2: Two qubit gates. The sign bits are all 0 in the above tableaux so they are omitted.

for some bits $M_{j1}, \ldots, M_{j(2n)} \in \{0,1\}$ and some phase[9] $\alpha_j \in \{\pm 1, \pm i\}$. The *tableau* for U is a succinct representation for U consisting of the binary matrix $M = [M_{jk}]$, and some representation of the phases $\alpha_1, \ldots, \alpha_{2n}$.

It turns out that U maps p_j (or any Pauli string) to ± 1 times a Pauli string. This follows from the fact that each Pauli string has ± 1 eigenvalues, which are preserved under conjugation by a unitary. However, α_j may still be any one of $\{\pm 1, \pm i\}$. This is because the product of $p_{2k-1}p_{2k}$ is $I \cdots I(XZ)I \cdots I = -iI \cdots IYI \cdots I$, with an awkward $-i$ phase. Once we cancel the extra factors of i from α_j, we are left with

$$(-1)^{v_j} := \alpha_j \prod_{k=1}^{n} (-i)^{M_{j(2k-1)} M_{j(2k)}},$$

where $v_j \in \{0,1\}$ is the *phase bit for row j*. For example, if $Up_1U^\dagger$ is $YI \cdots I$ then we have $M_{11} = M_{12} = 1$ and $v_1 = 0$. Then the complete tableau for U is the matrix $M = [M_{jk}]$ and the vector of bits $v = [v_j]$, which we typically write as

$$\left(\begin{array}{ccc|c} M_{11} & \cdots & M_{1(2n)} & v_1 \\ \vdots & \ddots & \vdots & \vdots \\ M_{(2n)1} & \cdots & M_{(2n)(2n)} & v_{2n} \end{array}\right).$$

Our ordering of the basis elements (which differs from other presentations [9]) puts Pauli strings on the same qubit (e.g., $XI \cdots I$ and $ZI \cdots I$) side-by-side in the matrix, so the 2×2 submatrix

$$\begin{pmatrix} M_{2i-1,2j-1} & M_{2i-1,2j} \\ M_{2i,2j-1} & M_{2i,2j} \end{pmatrix}$$

completely describes how the ith qubit of the input affects the jth qubit of the output. In fact, it will be fruitful to think of the tableau as an $n \times n$ matrix of 2×2 blocks, along with a vector of $2n$ *phase bits*. To be clear, the blocks come from $\mathsf{R} := \mathbb{F}_2^{2\times 2}$, the ring of 2×2 matrices over the field of two elements, $\mathbb{F}_2$. Then the tableau is a matrix in $\mathsf{R}^{n\times n}$ (the $n \times n$ matrices over the ring R), combined with a vector of phase bits in $\mathbb{F}_2^{2n}$. Each row of the matrix is associated with a pair of phase bits from the vector.

However, not every matrix in $\mathsf{R}^{n\times n}$ corresponds to a unitary operation and therefore to a Clifford gate. To help express valid tableau, we define a unary operation * on R such that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} d & b \\ c & a \end{pmatrix}.$$

The * operator has the property that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} d & b \\ c & a \end{pmatrix} = \begin{pmatrix} ad+bc & 0 \\ 0 & ad+bc \end{pmatrix} = I\begin{vmatrix} a & b \\ c & d \end{vmatrix}.$$

Additionally,

$$\begin{aligned} I^* &= I, \\ (M+N)^* &= M^* + N^*, \\ (MN)^* &= N^*M^*, \\ (M^*)^* &= M, \end{aligned}$$

[9]Note that the order of the terms in the product matters, since the Pauli basis elements do not necessarily commute, so we assume the terms are in the natural order from $p_1^{M_{j1}}$ up to $p_{2n}^{M_{j(2n)}}$.

	$\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}0&1\\1&1\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right)$	I	$\mathrm{R}_X^\dagger$	θ_{X+Z}	R_Z	Γ_{+++}	Γ_{---}
$\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$	X	R_X	$\mathrm{R}_Y^\dagger$	θ_{X+Y}	Γ_{--+}	Γ_{+-+}
$\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$	Z	θ_{Y+Z}	R_Y	$\mathrm{R}_Z^\dagger$	Γ_{-+-}	Γ_{-++}
$\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)$	Y	θ_{Y-Z}	θ_{X-Z}	θ_{X-Y}	Γ_{+--}	Γ_{++-}

Table 3: Invertible tableau elements and the corresponding single-qubit gates produced by the universal construction. Row of the table corresponds to the sign bit of the row of the tableau in which the element occurs.

so * makes R a *-*ring* or *ring with involution*. We also extend * to an operation on matrices (over R) which applies * to each entry and then transposes the matrix. It turns out that a tableau represents a unitary operation if and only if the matrix $M \in \mathsf{R}^{n\times n}$ satisfies $MM^* = M^*M = I$. This (intentionally) resembles the definition of a unitary matrix ($UU^\dagger = U^\dagger U = I$), but we will call this the *symplectic condition*. This follows the traditional presentation of the tableau as a symplectic matrix over $\mathbb{F}_2$.

4.1 Correspondence between Gates and Tableaux

We will find it useful to switch between gates and tableaux, as one notion often informs the other. Since the phase bits will often be irrelevant, let us denote the matrix part of the tableau for g as $\mathcal{M}(g)$. Most non-degenerate gate sets generate the Pauli group, which alone suffices to set the phase bits of the tableau arbitrarily by applying gates at the beginning of the circuit as follows: applying X to qubit j negates v_{2j} and applying Z to qubit j negates v_{2j-1}. Furthermore, there is a surprising connection between individual entries of tableaux and elementary Clifford operations that can be extracted from them (see Section 9).

If $a \in \mathsf{R}$ is invertible, then let $\mathcal{G}(a)$ be the single-qubit gate with $\mathcal{M}(\mathcal{G}(a)) = a$ and zeros for phase bits. These gates are well-defined since every 2×2 invertible matrix over $\mathbb{F}_2$ is itself a symplectic matrix over $\mathbb{F}_2$. They are shown in the first row of Table 3. Let $\mathcal{G}(a, i)$ be the gate $\mathcal{G}(a)$ applied to the ith qubit.

We also define gates from the noninvertible elements of R. We see that the tableau of each generalized CNOT gate consists of unique noninvertible elements $b \in \mathsf{R}$ and $b^* \in \mathsf{R}$ along the off-diagonal, as shown in Table 2. We summarize this correspondence more succinctly in Table 4. Therefore, if $b \in \mathsf{R}$ is noninvertible, define $\mathrm{CNOT}(b, i, j)$ to be the generalized CNOT on qubits i and j corresponding to the noninvertible matrix b. The tableau for $\mathrm{CNOT}(b, i, j)$ is the identity tableau except for b^* and b in positions (i, j) and (j, i), respectively. We use the circuit in Figure 5 to designate such a gate.

Element	$\left(\begin{smallmatrix}0&0\\1&0\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right)$ / $\left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&0\\1&0\end{smallmatrix}\right)$ / $\left(\begin{smallmatrix}0&0\\1&1\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)$ / $\left(\begin{smallmatrix}0&1\\0&1\end{smallmatrix}\right)$
Gen. CNOT	$C(X, X)$	$C(Y, Y)$	$C(Z, Z)$	$C(X, Z)$	$C(Y, X)$	$C(Z, Y)$

Table 4: Noninvertible tableau elements and the corresponding generalized CNOT gates produced by the universal construction.

Finally, we would like to have a direct way to compose two circuits by a simple operation on their tableaux. Suppose we wish to compute the composition of circuits C_1 and C_2. To compute the tableau of $C_2 \circ C_1$, we must compute, for each Pauli basis element p_j, the product

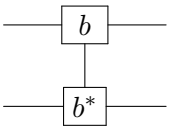


Figure 5: Circuit diagram for $\mathrm{CNOT}(b, 1, 2)$ gate.

$C_2 C_1 p_j C_1^\dagger C_2^\dagger$. First consider the jth row of the tableau for C_1, which gives

$$C_1 p_j C_1^\dagger = \alpha_j \prod_{k=1}^{2n} p_k^{M_{jk}^{(1)}},$$

where $M^{(1)}$ is the binary representation of $\mathcal{M}(C_1)$, and α_j is the phase. Similarly,

$$C_2 p_j C_2^\dagger = \beta_j \prod_{k=1}^{2n} p_k^{M_{jk}^{(2)}}.$$

Therefore,

$$\begin{aligned} C_2 C_1 p_j C_1^\dagger C_2^\dagger &= C_2 \left(\alpha_j \prod_{k=1}^{2n} p_k^{M_{jk}^{(1)}} \right) C_2^\dagger = \alpha_j \prod_{k=1}^{2n} \left(C_2 p_k^{M_{jk}^{(1)}} C_2^\dagger \right) = \alpha_j \prod_{k=1}^{2n} \left(\beta_k \prod_{\ell=1}^{2n} p_\ell^{M_{k\ell}^{(2)}} \right)^{M_{jk}^{(1)}} \\ &= \alpha_j \prod_{k=1}^{2n} \left(\beta_k^{M_{jk}^{(1)}} \prod_{\ell=1}^{2n} p_\ell^{M_{jk}^{(1)} M_{k\ell}^{(2)}} \right) \propto \prod_{\ell=1}^{2n} \prod_{k=1}^{2n} p_\ell^{M_{jk}^{(1)} M_{k\ell}^{(2)}} \propto \prod_{\ell=1}^{2n} p_\ell^{\bigoplus_{k=1}^{2n} M_{jk}^{(1)} M_{k\ell}^{(2)}} \\ &= \prod_{\ell=1}^{2n} p_\ell^{[M^{(1)} M^{(2)}]_{j\ell}} \end{aligned}$$

Notice that this implies $\mathcal{M}(C_2 \circ C_1) = \mathcal{M}(C_1)\mathcal{M}(C_2)$. Since it is cumbersome to write out explicitly, we did not include the exact phases in the above calculation. Nevertheless, one can compute the phase bits by tracking the intermediate steps in the above calculation, which includes the multiplication of Pauli basis elements.

5 Classes

The class generated by a set of Clifford gates is the collection of Clifford operations which can be constructed from circuits built from those gates. Formally, define a *class* $\mathcal{C}$ to be a set of Clifford gates satisfying the following four rules:

1. **Composition Rule** $\mathcal{C}$ is closed under composition of gates. If $f, g \in \mathcal{C}$ are gates on the same number of qubits, then $f \circ g \in \mathcal{C}$.

2. **Tensor Rule** $\mathcal{C}$ is closed under tensor product of gates. If $f, g \in \mathcal{C}$, then $f \otimes g \in \mathcal{C}$.

3. **Swap Rule** $\mathcal{C}$ contains the SWAP gate.

4. **Ancilla Rule** $\mathcal{C}$ is closed under ancillas. If $f \in \mathcal{C}$ and there exists g such that

   $$f(|x\rangle \otimes |\psi\rangle) = g(|x\rangle) \otimes |\psi\rangle$$

   for some $|\psi\rangle$ and for all inputs $|x\rangle$ (up to a global phase), then $g \in \mathcal{C}$.

Given that each class is supposed to capture those gates which can be built from a circuit, the operations of composition and tensor product are completely natural. Let us now spend some time to justify the swap and ancilla rules.

First, the swap rule allows us to consider gates without needing to specify the qubits on which they must be applied. Indeed, we can relabel the input wires however we like. There are natural alternatives to this rule, e.g., allow reordering the qubits on each gate. The classification will be more complicated under these definitions, and we feel that adding the SWAP gate is cleanest way to address the fact that a reordering of qubits is inherently uninteresting.[10]

Second, the ancilla rule allows us to have useful workspace for our computation. Notice that the other rules are "additive" in the sense that the number of qubits in the circuit never decreases—we need a rule to reduce the number of qubits, otherwise there is no way that, e.g., a 3-qubit gate could ever generate a 2-qubit gate. This would lead to hierarchies of classes that differ only on gates with a small number of qubits (e.g., classes that are identical except on 1-qubit and 2-qubit gates), solely because of the size of the available generators.

Instead, we permit quantum ancillas which can be used by the circuit, but must lead to a unitary transformation on the remaining non-ancilla (input) qubits. To guarantee unitarity, notice that the ancilla qubits must not be entangled with the input qubits at the end of the circuit. Furthermore, the ancillas cannot change in an input-dependent manner (e.g., consider the circuit that simply swaps in the input qubits with the ancilla qubits). The ancilla rule attempts to capture these restrictions: any ancilla qubits used during the computation must be returned to their initial configuration at the end of the computation. Notice that this rule coincides with the notion of a *catalyst* (see, e.g., [27, 28]).

One might assume that we could increase the power of ancillas by letting them change over the course of the computation, as long as the change is independent of the input (that is, from some constant initial state to some possibly different constant final state). Indeed, in the classification of reversible gates [1], these "loose ancillas" collapse a few pairs of classes. Nevertheless, we show that even with loose ancillas, our classification (as presented) still holds.[11]

Next, we have the question of how the ancillas are initialized. The weakest assumption one could make is that we have no control: the ancillas are initialized to an unknown state, and must be returned to this state at the end of the circuit. This is unreasonably harsh, since it is almost always practical to initialize a qubits to the $|0\rangle$ state, and there is reason to believe the classification is *dramatically* more difficult without some control over the state of the ancillas.[12] A slightly stronger assumption would be to allow ancillas initialized to computational basis states, but this would break symmetry by introducing a bias towards the Z-basis.[13]

A next natural step would be to permit ancillas initialized to arbitrary stabilizer states. Although this would appear to be circular (i.e., Clifford gates are necessary to implement stabilizer states, which we then use as ancillas in Clifford circuits), the reusability of the ancilla states implies that even if the states are difficult to construct, at least we only have to construct them once. Unfortunately, we are unable to complete the classification in its entirety under this ancilla model. However, we have reduced the problem to finding a single stabilizer state which

[10] From a category-theoretic perspective, the addition of the SWAP gate implies that our classes correspond to dagger *symmetric* monoidal groupoids, which may feel more natural to some readers [26].

[11] When we formally define class invariants in Section 6 (see Theorem 3 in particular), we will see that the invariants hold under the loose quantum ancilla model, and therefore hold for all weaker models.

[12] For example, the lattice of classical many-to-one functions over bits is finite when we allow 0/1 inputs to any function, but infinite when we do not allow such freedom [1, 29].

[13] We can fix the bias by allowing all single-qubit ancillary states: $|0\rangle, |1\rangle, |+\rangle, |-1\rangle, |i\rangle, |-i\rangle$. This introduces new classes such as $\langle \theta_{X+Z} \otimes \theta_{X+Z} \rangle$, but we leave the classification under these assumptions as an open problem.

is stabilized by Γ and a permutation, and we conjecture that such a state exists (see Section 11). Moreover, the conjectured classification matches the one we will present (under a stronger ancilla model).

Finally, we arrive at our chosen model, that is, ancillas initialized to arbitrary quantum states. A priori, these states could be arbitrarily large, and arbitrarily complicated to construct, which is clearly undesirable. It turns out, however, the classification only requires finitely many one or two qubit states, in particular, the eigenstates of the single-qubit Clifford gates and states that are locally equivalent to the Bell state. For comparison, we have determined the reversible gate lattice under quantum ancillas in Appendix B, and observe that, in some cases, arbitrarily large, entangled states are necessary.

It is worth noting that there is a long line of work showing that weak gate sets, including Clifford gates, are universal for quantum computation when given access to magic states [11, 13, 14]. Importantly, these magic states do *not* need to be preserved after the computation. Conversely, under our model, we show that arbitrary quantum ancillas cannot boost the power of Clifford gates beyond the Clifford group.

Let's now see how the rules justify natural properties one would want from a class:

Proposition 2. *Let $\mathcal{C}$ be a class of Clifford gates. Then $\mathcal{C}$ contains the n-qubit identity gate for all $n \geq 1$, and $\mathcal{C}$ is closed under inverses.*

Proof. All classes contain SWAP, and by composition it contains SWAP $\circ$ SWAP, which is the two-qubit identity. By the ancilla rule, we can remove a qubit to get the one-qubit identity, and then by the tensor rule we get the n-qubit identity.

Now suppose $g \in \mathcal{C}$ is an n-qubit Clifford gate. Since the n-qubit Clifford group is finite, g has finite order: $g^r = I$. If $r > 1$ then we can construct the $(r-1)$-fold composition, g^{r-1}, which is the inverse since $g \circ g^{r-1} = g^r = I$. Otherwise, $g = g^{-1} = I$. □

The most practical way to talk about a class $\mathcal{C}$ is by its *generators*. We say a set of gates G *generates* a class $\mathcal{C}$ if $G \subseteq \mathcal{C}$ and every class containing G also contains $\mathcal{C}$. We introduce the notation $\langle \cdot \rangle$ for the class generated by a set of gates. Similarly, we say that G generates a specific gate g if $g \in \langle G \rangle$.

Our goal is therefore to identify all Clifford gate classes, determine their generators, and diagram the relationships between classes. As it turns out, there are 57 different classes, which we have split across Figure 1 (which contains the classes with single-qubit generators) and Figure 2 (which contains the multi-qubit classes). Each class is labelled by a set of generators for that class, except for ALL, the class of all Clifford gates; $\top$, the class of all single-qubit Clifford gates; and $\bot$, the class generated by the empty set. Additionally, we abbreviate some class names in Figure 1:

- θ_{+++}, θ_{+--}, θ_{-+-}, θ_{--+} denote the single-qubit classes containing Γ_{+++}, Γ_{+--}, Γ_{-+-}, and Γ_{--+} respectively, and three θ gates each, as indicated by the gray lines.
- θ_{xy} abbreviates θ_{x+y} or θ_{x-y} (it contains both) and similarly for θ_{xz} and θ_{yz}.

In Figure 2 each class includes the label of the single-qubit subgroup, even when not all of the single-qubit generators are necessary to generate the class. This is intended to make the relationship between the degenerate and non-degenerate lattices clearer. For example, T_4 generates the Pauli group, $\mathcal{P}$, on its own (Lemma 11), but we label the class $\langle \mathrm{T}_4, \mathcal{P} \rangle$ to make it clear that the class $\langle \mathrm{T}_4 \rangle$ is above $\langle \mathcal{P} \rangle$ in the lattice.

6 Invariants

Until now, we have defined each class in terms of the generators for that class. It turns out that each class can also be characterized as the set of all gates satisfying a collection of invariants. Section 7 formalizes this equivalence. This section focuses on the form of the invariants themselves.

Informally, an *invariant* is a property of gates, readily apparent from their tableaux, which is preserved by the circuit building operations. In other words, if a collection of gates all satisfy a particular invariant then any circuit constructed from those gates must also satisfy the invariant. All our invariants are formally defined from the tableaux, but for now we give the following informal descriptions to make the intuition for each invariant clear.

X-, Y-, or Z-preserving: We say a Clifford gate is *Z-preserving* if it maps Z-basis states to Z-basis states, possibly with a change of phase. The Z-preserving gates include all (classical) reversible gates (e.g., X, CNOT, and T_4), gates which only manipulate the sign (e.g., R_Z and CZ), and combinations of the two.

Symmetrically, there are X-preserving gates mapping X-basis states to X-basis states, and Y-preserving gates gates mapping Y-basis states to Y-basis states. Our definitions of classes, gates, invariants, etc., are completely symmetric with respect to X, Y and Z basis, so if some gate or class is X-preserving (resp. Y-preserving or Z-preserving), then there must be a corresponding gate or class which is Y-preserving (resp. Z-preserving or X-preserving). We will often appeal to this symmetry to simplify proofs.

Note that a gate can be any combination of X-, Y-, and Z-preserving. For instance, T_4 is X-, Y-, and Z-preserving; CNOT is X-preserving and Z-preserving but not Y-preserving (similarly $C(X, Y)$ and $C(Y, Z)$ fail to be Z-preserving and X-preserving, respectively); R_Z is Z-preserving only (similarly R_X is X-preserving and R_Y is Y-preserving); and Γ is not X-, Y-, or Z-preserving.

Egalitarian We say an n-qubit gate U is *egalitarian* if

$$\mathcal{M}(\Gamma^{\otimes n} U (\Gamma^\dagger)^{\otimes n}) = \mathcal{M}(U).$$

Egalitarian gates have no preferred basis, since conjugation by Γ cycles X to Y, Y to Z, and Z to X. More concretely, if an egalitarian gate U maps Pauli string P to $Q = UPU^\dagger$ under conjugation, then U maps $\Gamma^{\otimes n} P (\Gamma^\dagger)^{\otimes n}$ to $\pm\Gamma^{\otimes n} Q (\Gamma^\dagger)^{\otimes n}$. The Pauli matrices, Γ, and T_4 are examples of egalitarian gates.

Degenerate: We say a gate is *degenerate* if each input affects only one output. More precisely, when applying the gate to a string of Paulis, changing one Pauli in the input will change exactly one Pauli in the output. All single-qubit gates are degenerate, and all degenerate gates can be composed of single-qubit gates and SWAP gates.

X-, Y-, or Z-degenerate: A gate is *Z-degenerate* if it is Z-preserving and flipping any bit of a classical (Z-basis) input to the gate causes exactly one bit of the output to flip. The gate may or may not affect the phase. This class includes several Z-preserving single-qubit gates, like R_Z, the Pauli operations, and θ_{X+Y}. It also includes CZ because this gate *only* affects phase, but CNOT is not Z-degenerate because flipping the control bit changes both outputs. Notice that CZ is Z-degenerate, but not degenerate. We define X-degenerate and Y-degenerate symmetrically.

X-, Y-, or Z-orthogonal: A gate G is *Z-orthogonal* if it can be built from T_4 and Z-preserving single-qubit gates. The term "orthogonal" comes from the fact that T_4 is an orthogonal linear transformation in the Z-basis, but not all Z-orthogonal gates are literally orthogonal transformations in the Z-basis (see, for example, Lemma 11). Similarly for X-orthogonal and Y-orthogonal.

Single-Qubit Gates: There are thirty different classes of single-qubit gates. All of these classes are degenerate, and some can be distinguished by the other invariants above. However, many single-qubit invariants depend on the phase bits of the tableau. For instance, the tableau of θ_{X+Y}, θ_{X-Y}, and R_Z all have the same matrix part, $\left(\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right)$, but generate three distinct classes. One can write down explicit invariants for these classes where the phase bits are correlated to the tableau entries, but in most cases we present a single-qubit class as a subgroup of the symmetries of the cube/octahedron, as shown in Figure 3.

6.1 Formal invariants

An *invariant* is a property of tableaux which is preserved by the four circuit-building rules.

Swap Rule: Every class contains the SWAP gate, so every invariant we propose must be satisfied by the tableau for SWAP.

Composition Rule: If the invariant holds for two gates, then it must hold for their composition. We have seen that the tableau for the composition of two gates is essentially the matrix product of the two tableau, except for the phase bits (which are significantly more complicated to update).

Tensor Rule: The tensor product of two gates satisfying the invariant must also satisfy the invariant. Note that the tableau of the tensor product is the direct sum of the tableaux, and phase bits are inherited from the sub-tableaux in the natural way.

Ancilla Rule: The invariant must be preserved when some qubits are used as ancillas. It turns out the ancilla operation reduces the tableau to a submatrix (of non-ancilla rows and columns) and under certain conditions, the corresponding subset of the phase bits. This is somewhat technical, so we prove it in Theorem 3 below.

Theorem 3. *Let G be a Clifford gate on n qubits, and suppose there exist states $|\psi\rangle$ and $|\psi'\rangle$ such that*

$$G(|x\rangle \otimes |\psi\rangle) = H(|x\rangle) \otimes |\psi'\rangle$$

for all $|x\rangle$, for some unitary H on m-qubits. In particular, this is true if we use the ancilla rule to reduce G to H, where $|\psi\rangle = |\psi'\rangle$ is the ancilla state. Then

1. *H is a Clifford operation,*
2. *$\mathcal{M}(H)$ is obtained by removing the rows and columns corresponding to the ancilla bits from $\mathcal{M}(G)$,*
3. *If a row of $\mathcal{M}(H)$ is obtained by removing only zeros, then the phase bit for that row is the same in G and H.*

Proof. Let $P \in \mathcal{P}_m$. Then for any $|x\rangle$,

$$G(P|x\rangle \otimes |\psi\rangle) = H(P|x\rangle) \otimes |\psi'\rangle.$$

On the other hand, G is a Clifford gate, so conjugating the Pauli string $P \otimes I^{n-m}$ by G produces $\alpha Q \otimes R$ for Pauli strings $Q \in \mathcal{P}_m$ and $R \in \mathcal{P}_{n-m}$ and phase $\alpha \in \pm 1$. Equivalently,

$$G(P \otimes I^{n-m}) = \alpha(Q \otimes R)G.$$

It follows that

$$\begin{aligned} H(P|x\rangle) \otimes |\psi'\rangle &= G(P|x\rangle \otimes |\psi\rangle) \\ &= \alpha(Q \otimes R)G(|x\rangle \otimes |\psi\rangle) \\ &= \alpha(Q \otimes R)\left(H(|x\rangle) \otimes |\psi'\rangle\right) \\ &= \alpha_1 QH(|x\rangle) \otimes \alpha_2 R|\psi'\rangle, \end{aligned}$$

for some choice of $\alpha_1, \alpha_2 \in \mathbb{C}$ for which $\alpha_1\alpha_2 = \alpha$, $H(P|x\rangle) = \alpha_1 QH(|x\rangle)$, and $|\psi'\rangle = \alpha_2 R|\psi'\rangle$. We see that $|\psi'\rangle$ is an eigenvector with eigenvalue α_2^{-1}, so our choice of α_2 (and hence α_1) is unique. Since R is a Pauli with eigenvalues ± 1, it follows that $\alpha_2 = \pm 1$.

Therefore, we have $\alpha_1 \in \{\pm 1\}$ and $HP = \alpha_1 QH$, which implies $HPH^\dagger = \alpha_1 Q$. That is, since P was arbitrary, the conjugation of a Pauli element by H is always another Pauli element, so H is a Clifford gate. In the special case that P (and therefore $P \otimes I^{n-m}$) is a Pauli basis element, then $\alpha Q \otimes R$ is represented in the row of the binary tableau of G. We keep the bits representing Q in the tableau for H, which is why $\mathcal{M}(H)$ is a submatrix of $\mathcal{M}(G)$. The phase in the new tableau is α_1. In the special case $R = I^{n-m}$, the only eigenvalue is 1 so $\alpha_2 = 1$ and hence $\alpha_1 = \alpha$. That is, the phase for the corresponding row of the tableau for H is inherited from G. □

As a direct consequence of these rules, our invariants take on a distinctly algebraic flavor. Let us consider, for the sake of illustration, invariants that depend only on the matrix part of the tableau and ignore the phase bits. Then an invariant is equivalent to a set of matrices closed under the four rules above. In particular, the matrices do form a group under multiplication as a consequence of the composition rule (and the fact that every gate has finite order).

On the other hand, not every group of matrices will correspond to an invariant. For instance, due to the swap rule, the group of matrices must also be closed under arbitrary reordering of the rows and columns. This eliminates, e.g., the group of upper triangular matrices. Similarly, the ancilla rule excludes the special orthogonal group. In the end, we are left with just two kinds of matrix groups which lead to invariants:

Subring Invariants Matrices with elements restricted to a particular subring of R (analogous to the real matrices, integer matrices, etc.)

Permutation Invariants Permutation matrices, except where each 1 entry can be any one of a subset of invertible elements, and each 0 entry comes from a collection of non-invertible elements.

Now we are ready to present formal definitions for these invariants, and show that they really are preserved by the circuit-building rules.

6.2 Subring invariants

The first kind of invariant restricts the entries of the tableau to a subring of R. That is, given a subring $\mathsf{S} \subseteq \mathsf{R}$, a gate satisfies the invariant $\mathcal{I}(\mathsf{S})$ if and only if all entries of the tableau are in

S.[14] There are twelve classes, all near the top of the lattice, of the form

$$\mathcal{C} = \{\text{All gates satisfying } \mathcal{I}(\mathsf{S})\},$$

corresponding to all 12 subrings of R listed below.

- The entire ring, R, is technically a subring of itself, and $\mathcal{I}(\mathsf{R})$ is the trivial invariant satisfied by all Clifford gates. Notice that not *every* matrix over R gives a valid tableau because it must still be symplectic.

- There are four maximal proper subrings of R:

$$\begin{aligned}
\mathsf{R}_X &= \{\left(\begin{smallmatrix} a & 0 \\ c & d \end{smallmatrix}\right) : \{a, c, d\} \in \{0, 1\}\}, \\
\mathsf{R}_Y &= \{\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) : \{a, b, c, d\} \in \{0, 1\}, a + b + c + d = 0\}, \\
\mathsf{R}_Z &= \{\left(\begin{smallmatrix} a & b \\ 0 & d \end{smallmatrix}\right) : \{a, b, d\} \in \{0, 1\}\}, \\
\mathsf{R}_E &= \{\left(\begin{smallmatrix} a & b \\ b & a+b \end{smallmatrix}\right) : \{a, b\} \in \{0, 1\}\}.
\end{aligned}$$

  Our formal definition for Z-preserving gates is the invariant $\mathcal{I}(\mathsf{R}_Z)$. The fact that the lower left entry is 0 implies that the gate maps Pauli strings of I and Z to strings of I and Z. Hence, Z-basis strings are mapped to Z-basis strings. Similarly, the X-preserving and Y-preserving invariants are $\mathcal{I}(\mathsf{R}_X)$ and $\mathcal{I}(\mathsf{R}_Y)$ respectively. The egalitarian invariant, $\mathcal{I}(\mathsf{R}_E)$, comes from the subring R_E.

- The intersection of two subrings is itself a subring, giving us exactly four more subrings ($\mathsf{R}_X \cap \mathsf{R}_Y$, $\mathsf{R}_X \cap \mathsf{R}_Z$, $\mathsf{R}_Y \cap \mathsf{R}_Z$, and $\mathsf{R}_X \cap \mathsf{R}_Y \cap \mathsf{R}_Z$) since the intersection of R_E with any of the others is

$$\mathsf{R}_X \cap \mathsf{R}_Y \cap \mathsf{R}_Z = \{\left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)\},$$

  the trivial ring.

- Three more subrings are obtained by taking only self-conjugate elements of R_X, R_Y, and R_Z respectively. An element $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ is *self-conjugate* if

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)^*,$$

  or equivalently, $a = d$. These invariants correspond to the X-orthogonal (i.e., $\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_X \rangle$), Y-orthogonal (i.e., $\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_Y \rangle$), and Z-orthogonal (i.e., $\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_Z \rangle$) classes respectively.

Theorem 4. *For any subring $\mathsf{S} \subseteq \mathsf{R}$, the property $\mathcal{I}(\mathsf{S})$ is an invariant. That is, the set of matrices over S respect the circuit building operations.*

Proof. Every subring contains $\left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ by definition, and therefore the tableau (phase bits omitted) of the SWAP gate,

$$\left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right)$$

satisfies $\mathcal{I}(\mathsf{S})$.

[14] There are several interesting works that connect the elements of the matrix representation of a unitary, to the set of gates that generate it. For example, every multi-qubit unitary with elements in $\mathbb{Z}[1/\sqrt{2}, i]$ corresponds to a circuit built from Clifford + T gates [30, 31]. There is a classification of gates corresponding to subrings of $\mathbb{Z}[1/\sqrt{2}, i]$ [7]. We stress, however, that our classification depends on the the ring elements of the *tableau*.

Matrix multiplication is a polynomial in the entries of the two matrices, so composition cannot produce entries outside the subring. Similarly, combining tableaux with tensor products or reducing tableaux to submatrices via ancillas does not introduce any new ring elements (see Theorem 3); those operations only use elements already present in the tableau. We conclude that $\mathcal{I}(\mathsf{S})$ is an invariant for any subring S. □

6.3 Permutation invariants

The *permutation invariants* get their name from the matrix part of their tableaux, which is required to have the structure of a permutation matrix. That is, every row (or column) has exactly one element which is invertible, and the others are non-invertible. Permutation invariants are also sensitive to phase bits. It is natural to associate the unique invertible element in a row with the phase bits for that row, giving the tableau of a single-qubit gate. A permutation invariant $\mathcal{P}(G, S)$ is defined by the set of single-qubit gates G which can be obtained in this way, and the set of non-invertible elements S used to fill the rest of the tableau. In other words, a tableau satisfies $\mathcal{P}(G, S)$ if all entries are from S except exactly one entry per row which, when combined with the phase bits for the row, is the tableau of some gate in G.

Note that not all pairs of sets (G, S) produce an invariant. For instance, circuit-building operations will fail to preserve $\mathcal{P}(G, S)$ if G is not a group. The exact relationship between G and S required to produce an invariant is difficult to write down. Roughly speaking, products of elements in S should be zero, products of elements in G should remain in G, and products between S and $\mathcal{M}(G)$ should be manageable in some sense. Theorem 5 gives a list of $\mathcal{P}(G, S)$ invariants, which will turn out to be exhaustive by Theorem 19, the culminating theorem of this paper.

Theorem 5. *We prove that the following permutation invariants are indeed invariant under the circuit-building operations. Let G be a group of single-qubit gates.*

1. *Then*
$$\mathcal{P}(G, \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right)\})$$
is an invariant for $\langle G\rangle$. All thirty degenerate classes are characterized by invariants of this form.

2. *If $\langle X\rangle \subseteq \langle G\rangle \subseteq \langle \mathcal{P}, \mathrm{R}_X\rangle$ then*
$$\mathcal{P}(G, \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&0\\1&0\end{smallmatrix}\right)\})$$
is an invariant for $\langle C(X, X), G\rangle$. These invariants characterize the five X-degenerate classes.

3. *If $\langle Y\rangle \subseteq \langle G\rangle \subseteq \langle \mathcal{P}, \mathrm{R}_Y\rangle$ then*
$$\mathcal{P}(G, \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\right)\})$$
is an invariant for $\langle C(Y, Y), G\rangle$. These invariants characterize the five Y-degenerate classes.

4. *If $\langle Z\rangle \subseteq \langle G\rangle \subseteq \langle \mathcal{P}, \mathrm{R}_Z\rangle$ then*
$$\mathcal{P}(G, \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)\})$$
is an invariant for $\langle C(Z, Z), G\rangle$. These invariants characterize the five Z-degenerate classes.

Proof. Let $\mathcal{P}(G,S)$ be one of the invariants above. Let $M = \{\mathcal{M}(g) : g \in G\}$ be the set of matrices from tableaux in G. In all cases, S contains $\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right)$, and G contains the single-qubit identity operation,

$$\left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right),$$

so SWAP satisfies the invariant. And clearly the direct sum of two tableaux in $\mathcal{P}(G,S)$ is still in $\mathcal{P}(G,S)$ for any G and S.

Now consider the composition of two gates. Each entry in the tableau is a dot product of some row from one tableau with some column from the other. Hence, the entry is a sum of $S \times S$, $S \times M$, $M \times S$, or $M \times M$ products. Observe that the $S \times S$ products are all zero (for the particular sets S above), so we may ignore those products. Recall that the row and column each contain exactly one entry in M, so depending on whether those entries align, we get either $SM + MS \subseteq S$ or $M^2 = M$. Furthermore, for any row in one tableau there is exactly one column in the other such that the invertible entries line up. Therefore, exactly one entry in any row (or column) of the composition is in M and the rest are in S. Clearly the matrix part of the tableau has the correct form for the invariant.

We must also consider phase bits under composition. Recall that the phase bits associate with the invertible entries of the matrix to produce single-qubit gates. When we multiply two tableau, these single-qubit gates multiply to produce elements in G (since G is a group), as you would expect. If the non-invertible elements are all zero, then this is the only factor in determining phase bits, so the invariant is preserved by composition.

Now consider the phase bits in the case where S contains nonzero elements, for instance,

$$S = \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)\}.$$

Notice that in this case, both matrices in S have zeros in the bottom row, and the invertible matrices are of the form $\left(\begin{smallmatrix}1&a\\0&1\end{smallmatrix}\right)$. Hence, every even-indexed row of the tableau (as a binary matrix) is all zeros except for one entry. Using the method of tableau composition in Section 4, one can easily show that for these even-indexed rows, the phase bits are exactly what one would get by composing the invertible elements as gates in G. For the other half of the rows, the non-invertible elements may flip the phase bits. But we assume G contains the Pauli element (in this case Z) which flips that sign, so the invertible elements and associated phase bits are still in G, therefore the invariant is preserved. The X- and Y-degenerate cases are similar.

Last, we show that $\mathcal{P}(G,S)$ is preserved under ancilla operations. Recall that when we use ancillas, we remove the rows and columns corresponding to those bits. Clearly the elements of the submatrix are still in M and S. There is a risk that the invertible element for some row could be in one of the removed columns, but if the submatrix is missing an invertible element in some row then the submatrix violates the symplectic condition and the ancilla rule must have been misapplied. Hence, only elements in S are removed in the non-ancilla part of the tableau, and each row still contains exactly one entry in M.

We appeal to Theorem 3 for the phase bits. The theorem says that removing the ancillas can only change the sign for a row if there is a nonzero entry in the non-ancilla bits of the row that are removed. For example, if $S = \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)\}$ then only the top phase bit can change. But changing the top phase bit is the same as applying a Z, and for this case Z is assumed to be in G, so the combination of the element in M and the phase bits is still in G. Therefore the Z-degenerate $\mathcal{P}(G,S)$ are invariants, and the X-degenerate and Y-degenerate invariants follow by symmetry.

□

7 Equivalence of Generator and Invariant Definitions

We have now defined each class by a set of generators, and by an invariant, but have not yet shown that these definitions coincide. Below are a collection of lemmas which prove this for all classes in our lattice. Note that one direction is always trivial: it is easy to check that the generators defining a class satisfy a particular invariant, and therefore everything they generate (i.e., the class) must satisfy the invariant. We encourage the reader to check these invariants against, say, the tableaux in Table 2.

For the other inclusion (i.e., every gate satisfying the invariant can be generated by the given generators), we start with an arbitrary gate g satisfying the invariant, and apply gates in the class to g to simplify its tableau step-by-step until it is the tableau of the identity operation. It follows that $AgB = I$ for circuits A and B in the class, which proves $g = A^{-1}B^{-1}$ is in the class. In many cases, the circuit derived this way is a *canonical form* for the gate, and can be used to count the number of gates on n qubits in a class.

Let us start with the degenerate classes.

Lemma 6. *Let G be a group of single-qubit gates, and let g be a gate satisfying the permutation invariant $\mathcal{P}(G, \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right)\})$. Then there is a circuit for g consisting of a permutation of the inputs followed by layer of single-qubit gates in G.*

Proof. Consider the tableau for g. Each row or column has exactly one invertible element, so we can read off a permutation π from the positions of those elements. Apply SWAP gates to g to remove this permutation, and put the invertible elements on the diagonal. When we pair a diagonal element with the phase bits for that row, we get a single-qubit gate g_i in G. Applying the inverse of this gate to qubit i will zero the phase bits for that row, and make $\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ the diagonal entry. Once we do this for each row, we have the identity tableau, therefore g is in $\langle G\rangle$ and has a circuit of the desired form. □

Next, we consider the Z-degenerate classes and, by symmetry, the X- and Y-degenerate classes.

Lemma 7. *Let G be a group of Z-preserving single-qubit gates such that $\langle Z\rangle \subseteq \langle G\rangle \subseteq \langle \mathcal{P}, \mathrm{R}_Z\rangle$. Let g be any gate satisfying the permutation invariant $\mathcal{P}(G, \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)\})$. Then there is a circuit for g consisting of a layer of single-qubit gates (from G), a layer of $C(Z,Z)$ gates, and a permutation.*

Proof. Consider the tableau of g. We can read off a permutation π, and a single-qubit gate for each input. Assume we have removed those gates (i.e., we now consider the tableau of $g\pi^{-1}g_1^{-1}\cdots g_n^{-1}$), so the tableau has $\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ on the diagonal, all other entries are either $\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)$ or zero, and the phase bits are zero.

The non-zero, off-diagonal entries in the matrix indicate the positions of $C(Z,Z)$ gates. Specifically, if the entry in row i and column j is nonzero then there is a $C(Z,Z)$ on qubits i and j. Note that because the matrix part of the tableau is symplectic, the symmetric entry in row j and column i must also be non-zero. The remainder of the circuit consists of the set of $C(Z,Z)$ gates indicated by the non-zero, off-diagonal entries. Notice that $C(Z,Z)$ gates always commute, so their ordering does not matter. □

Now let us consider four Z-preserving classes which, when we consider symmetry (i.e., the X-preserving and Y-preserving equivalents) cover all but two of the remaining classes.

Lemma 8. *Each of the classes* $\langle \mathrm{T}_4, \mathcal{P}\rangle$, $\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_Z\rangle$, $\langle C(Z,X), \mathcal{P}\rangle$, *and* $\langle C(Z,X), \mathcal{P}, \mathrm{R}_Z\rangle$ *is the set of all gates corresponding to a subring invariant, where the subrings are*

$$\begin{aligned}
\mathsf{S}_1 &= \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\},\\
\mathsf{S}_2 &= \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)\},\\
\mathsf{S}_3 &= \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\},\\
\mathsf{S}_4 &= \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)\}.
\end{aligned}$$

Proof. In all four classes, elements of the tableau are of the form $\left(\begin{smallmatrix}a&b\\0&d\end{smallmatrix}\right)$. Suppose $\left(\begin{smallmatrix}?&?\\0&d_1\end{smallmatrix}\right)$ is the ith entry of some row, and $\left(\begin{smallmatrix}?&?\\0&d_2\end{smallmatrix}\right)$ is the jth entry in the same row, where entries labeled by "?" are unconstrained. If we apply a CNOT gate from qubit i to j, these entries will be of the form $\left(\begin{smallmatrix}?&?\\0&d_1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}?&?\\0&d_1+d_2\end{smallmatrix}\right)$ respectively. That is, the bottom right bits change as though we applied the CNOT gate to those bits. Since a T_4 gate can be built from CNOT gates, it will (similarly) affect the bottom right bits as though we are applying a T_4.

Our strategy is to use either CNOT or T_4 gates (depending on the class) to perform Gaussian elimination on the bottom right entries of the matrix elements. If we have access to CNOT gates then we literally apply Gaussian elimination, using CNOT to add one column to another, and using SWAP to exchange columns.

If we only have T_4 gates then we are in subring $\mathsf{S}_1 \subseteq \mathsf{S}_2$ or S_2, so $\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$ are the only elements with a 1 in the bottom right position, and also the only invertible elements. It follows that the number of bottom right bits set to 1 in a row is the same as the number of invertible elements, which must be odd because the matrix is symplectic. To reduce the number, we apply a T_4 to three 1 bits and a 0 bit (note: we may add a zero bit by adding an ancilla, if necessary), which changes the 0 to a 1 and the 1's to 0's, reducing the number of 1's (or invertible elements) in the row by two. When there is a single 1 left in the row, symplectic conditions imply that it is also the only 1 left in that column, so we may ignore that row and column for the moment and continue to eliminate the rest of the matrix.

Now suppose we have row reduced the matrix, using either CNOT or T_4, so that the bottom right entry of every element is 0, except along the diagonal where that bit 1. At this point, the diagonal element is the only element in a row that can possibly be invertible, therefore the diagonal elements are of the form $\left(\begin{smallmatrix}1&b\\0&1\end{smallmatrix}\right)$. Similarly, the symplectic conditions imply that the off-diagonal elements are of the form $\left(\begin{smallmatrix}0&b\\0&0\end{smallmatrix}\right)$. In other words, the remaining tableau is Z-degenerate, since there is only one invertible element per row or column, and the off-diagonal elements are in $I = \{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)\}$. We can use either Lemma 6 or Lemma 7 to find a circuit from the remainder, which is in either $\langle \mathcal{P}\rangle$ or $\langle C(Z,Z), \mathcal{P}, \mathrm{R}_Z\rangle$, depending on the class. □

There are only two classes remaining, ALL and $\langle \mathrm{T}_4, \mathcal{P}, \Gamma\rangle$, which we handle specially. For the first, we appeal to Aaronson and Gottesman [9] who give an explicit decomposition for any Clifford gate into layers of CNOT, Hadamard (θ_{X+Z} in our notation), and phase (R_Z) gates.

Lemma 9. *Any egalitarian gate* g *can be generated in* $\langle \mathrm{T}_4, \mathcal{P}, \Gamma\rangle$.

Proof. Egalitarian gates satisfy the invariant that all elements are in the subring

$$\{\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right), \left(\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right), \left(\begin{smallmatrix}0&1\\1&1\end{smallmatrix}\right).\}$$

In fact, this subring is isomorphic to $\mathbb{F}_4$, so it is a field. In particular, we will use that the only noninvertible element of the subring is zero. The symplectic conditions on the tableau translate to it being unitary as a matrix over $\mathbb{F}_4$.

Like the other T_4 classes, we use Gaussian elimination on the tableau of g. Consider a row of the tableau. If the entry in some column is not the identity, then apply Γ or Γ^{-1} to the

corresponding qubit to make it the identity. By the symplectic condition, there are an odd number of identity elements in the row. We may remove pairs of identity elements with a T_4, similar to Lemma 8, until there is only one left and the rest of the row is zero. The symplectic condition implies the column below the identity element is also zero, and we proceed to eliminate the rest of the tableau. Once the matrix part of the tableau is a permutation, we apply SWAP gates so that it becomes the identity and apply Pauli matrices to zero out the phase bits.

We conclude that all egalitarian gates are in $\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle$. □

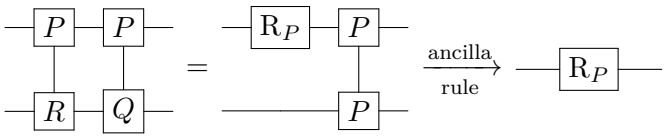
8 Circuit Identities

In this section, we give necessary tools to prove that a set of gates generates, in some sense, "all that one could hope for." Formally, we wish to prove that the gate set generates a particular class in the classification lattice when it is contained in that class but fails to satisfy the invariants of all classes below it. To this end, we give several useful circuit identities that will be used extensively in Section 9. For instance, one can show that any circuit on two qubits can be reduced to an equivalent circuit containing at most one generalized CNOT gate (see Appendix E). The following lemma gives only the aspect of that theorem that is necessary to the classification, that is, the ability to extract single-qubit Clifford operations from the composition of generalized CNOT gates.

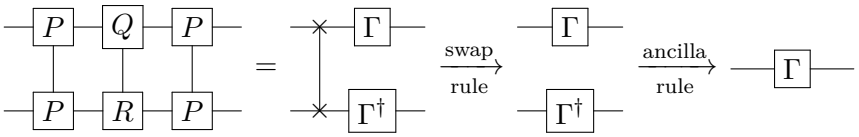
Lemma 10. *Let $P, Q, R \in \mathcal{P}$, and let $\Gamma P \Gamma^\dagger = Q$ and $\Gamma Q \Gamma^\dagger = R$. Then*

- *$C(P,Q)$ and $C(P,R)$ generate R_P.*
- *$C(P,P)$ and $C(P,Q)$ generate R_P.*
- *$C(P,P)$ and $C(Q,R)$ generate Γ.*
- *$C(P,P)$ and $C(Q,Q)$ generate θ_{P+Q}.*

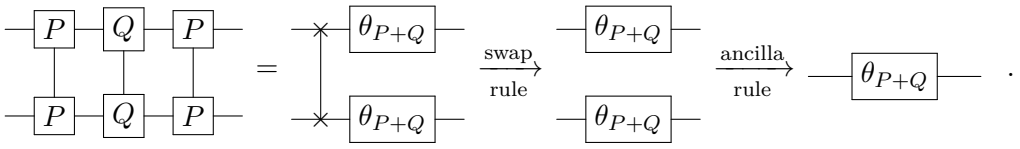
Proof. The first inclusion comes from the following identity:



where we have used that $QR = iP$ is a Pauli. Conjugating the second qubit by Γ in the diagram above, gives the second identity. For the third identity, we have



and for the final identity

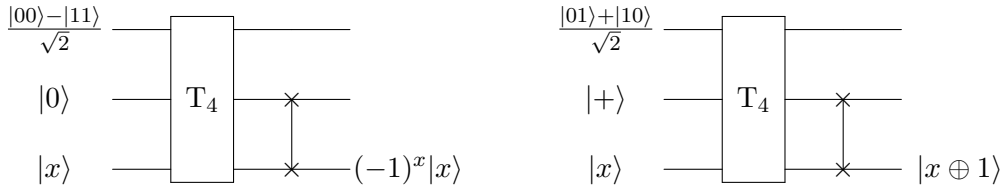


□

It might seem strange to reduce non-degenerate gates into less powerful single-qubit gates, but we will eventually see that the single-qubit generators are crucial. Once we have shown that a particular set of gates generates all single-qubit operations, then that set of gates will generate the class of *all* Clifford operations provided it contains any non-degenerate gate. All non-degenerate gates generate at least one Pauli, often the entire Pauli group, which is why some single-qubit classes do not appear as the single-qubit subgroup of a non-degenerate class. For instance, consider the CNOT gate where the first qubit controls the second qubit. If we let the first input be $|1\rangle$, then a Pauli X operation is always applied to the second qubit. Similarly, if we let the input to the second qubit be $|-\rangle$, then a Pauli Z operation is always applied to the first qubit. Under the ancilla rule, we now have Pauli X and Z operations, so we can generate Y and the entire Pauli group. Clearly, the same is true for any heterogeneous CNOT gate. However, surprisingly, the following lemma shows that even the T_4 gate suffices to generate the entire Pauli group.

Lemma 11. T_4 *generates the Pauli group.*

Proof. Consider the following two circuits:



Under the ancilla rule, the first generates a Pauli Z operation while the second generates a Pauli X, from which we can clearly generate the Pauli group. □

Lemma 12. T_4 *and* $C(P,P)$ *generate* R_P.

Proof. Figure 6 shows how to generate R_Z with $C(Z,Z)$. To generate R_X with $C(X,X)$, we simply appeal to the symmetry of T_4. Concretely, consider conjugating the entire circuit in Figure 6 by $\Gamma^{\otimes 4}$. We must now check: the circuit generates R_X; and it is in the class $\langle \mathrm{T}_4, C(X,X)\rangle$. For the former, note that we can still apply the ancilla rule on the first three qubits (i.e., by multiplying the original ancillas by Γ on each qubit). Therefore, the transformation on the final qubit is $\Gamma\,\mathrm{R}_Z\,\Gamma^\dagger = \mathrm{R}_X$.

To check containment in the class, we simply conjugate all the gates in the circuit by $\Gamma^{\otimes 4}$. SWAP is unchanged, and $C(Z,Z)$ is mapped to $C(X,X)$. Finally, recall that T_4 is egalitarian, so

$$\mathcal{M}(\Gamma^{\otimes 4}\,\mathrm{T}_4(\Gamma^\dagger)^{\otimes 4}) = \mathcal{M}(\mathrm{T}_4).$$

That, it is mapped to is itself followed by layer of Pauli operations (technically, we have the identity $\Gamma^{\otimes 4}\,\mathrm{T}_4(\Gamma^\dagger)^{\otimes 4} = \mathrm{T}_4\,X^{\otimes 4}$). Since T_4 generates the Pauli group by Lemma 11, the entire conjugated circuit is in $\langle \mathrm{T}_4, C(X,X)\rangle$. Repeating the entire argument for $C(Y,Y)$ and R_Y completes the proof. □

The following lemmas make precise our working assumption that single-qubit gates can significantly bolster the power of non-degenerate gate sets.

Lemma 13. *Suppose we have any* $C(P,Q)$ *gate with any single-qubit gate* G *that does not preserve the* P*-basis and any single-qubit gate* H *that does not preserve the* Q*-basis. Then* $\langle C(P,Q), G, H\rangle = \mathsf{ALL}$.

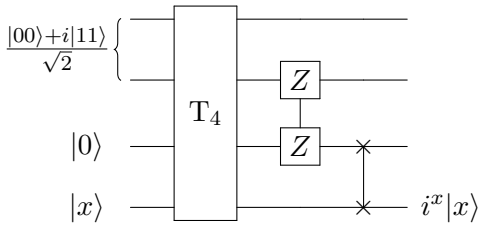


Figure 6: Generating R_Z with T_4 and $C(Z,Z)$.

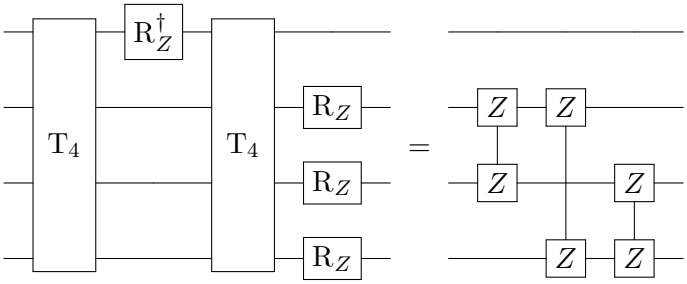


Figure 7: Generating $C(Z,Z)$ with T_4 and R_Z.

Proof. We will prove that the class $\langle C(P,Q), G, H\rangle$ contains all single-qubit gates. Then, to prove that the class generates all Clifford operations, it is be sufficient to show that it contains a CNOT gate. However, since all generalized CNOT gates are conjugates of each other, this is immediate.

First suppose $P = Q$. Since G does not preserve P-basis, we can use G to create a $C(R,R)$ gate where $R \neq P$. By Lemma 10, we can generate a θ_{P+R} gate. Conjugating $C(P,P)$ by θ_{P+R} on the second qubit yields a $C(P,R)$ gate. Once again leveraging Lemma 10, $C(P,R)$ and $C(P,P)$ generate an R_P gate. Referring to the single-qubit lattice (see Figure 1), we see that the class $\langle \mathcal{P}, \theta_{P+R}, \mathrm{R}_P\rangle$ contains all single-qubit gates.

Now suppose that $P \neq Q$. Once again, since G does not preserve P-basis, we can use G to create a $C(R,Q)$ gate. If $R = Q$, then by the logic above, we can use H to generate all single-qubit gates, so suppose $R \neq Q$. By Lemma 10, we can use $C(P,Q)$ and $C(R,Q)$ to generate an R_Q gate. Conjugating both $C(P,Q)$ and $C(R,Q)$ by H appropriately, gives a $C(P,S)$ and $C(R,S)$ for some $S \neq Q$, which we can once again generate an R_S gate. Referring to the single-qubit lattice, we see that the class $\langle \mathcal{P}, \mathrm{R}_S, \mathrm{R}_Q\rangle$ contains all single-qubit gates. □

Lemma 14. T_4 *with the class of all single-qubit gates generates* ALL.

Proof. It is well known that CNOT, θ_{X+Z}, and R_Z generate all Clifford circuits. Therefore, it will be sufficient to show that T_4 plus all single-qubit gates generate CNOT. Under the ancilla rule, it is clear by Figure 7 that T_4 and R_Z suffice to generate $C(Z,Z)$. Conjugating one qubit of $C(Z,Z)$ by θ_{X+Z} yields a $C(Z,X) = \mathrm{CNOT}$ gate, completing the proof. □

9 Universal Construction

Suppose G is an n-qubit Clifford gate. It turns out there is a single circuit $\mathfrak{C}(G)$, the *universal construction*, which can help us extract useful generators (e.g., single-qubit gates, generalized

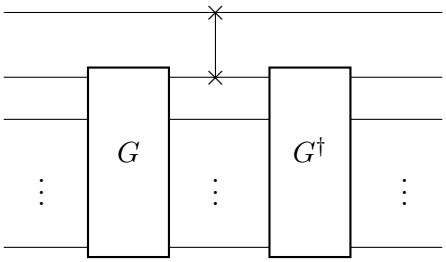


Figure 8: Universal Construction $\mathfrak{C}(G)$

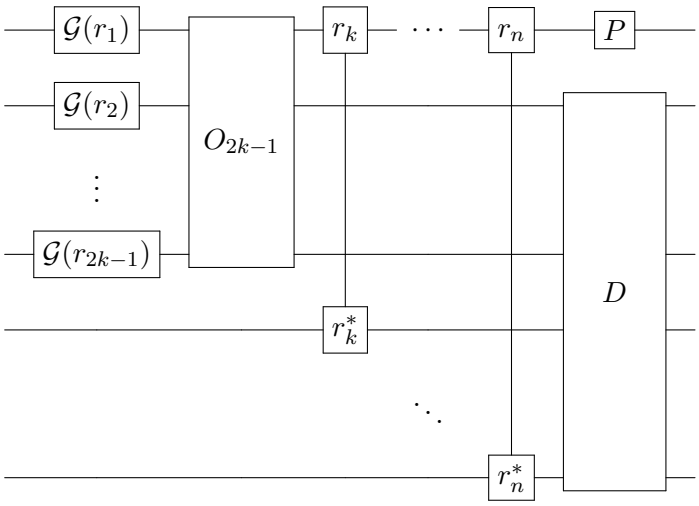


Figure 9: A diagram of the decomposition in Lemma 15. For convenience, $S = \{1, \ldots, 2k-1\}$ and $i = 1$, so O_{2k-1} is on the first $2k-1$ qubits and no swap is necessary.

CNOTs, etc.) from G. Specifically, the circuit $\mathfrak{C}(G)$ (shown in Figure 8) applies G to qubits 2 through $n+1$, swaps qubits 1 and 2, then applies $G^\dagger$ to qubits 2 through $n+1$.

The intuition is that since we apply G and $G^\dagger$ back-to-back on all but one of the same qubits, they "mostly" cancel and the result depends on the influence of the swapped qubit. In fact, we will show that the universal construction is equivalent to a circuit derived from the column of the tableau associated with that qubit. If that part of the tableau violates a particular invariant, then this circuit contains some gate *also* violating that invariant, and in the next section we show how to extract it. For now, we focus on proving our claims about the universal construction.

Our main tool for analyzing $\mathfrak{C}(G)$ is a novel canonical form for Clifford circuits. For all $k \geq 1$, let O_{2k-1} be a $(2k-1)$-qubit gate of the form

$$\mathcal{M}(O_{2k-1}) = \begin{pmatrix} I & I & I & \cdots & I \\ I & \alpha & \alpha^* & \cdots & \alpha^* \\ I & \alpha^* & \alpha & & \alpha^* \\ \vdots & \vdots & & \ddots & \vdots \\ I & \alpha^* & \alpha^* & \cdots & \alpha \end{pmatrix}$$

where $\alpha = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right), I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \in \mathsf{R}$. That is, I in the first row and column, otherwise α on the diagonal and α^* off-diagonal. We note that the three-qubit gate O_3 generates $\langle \mathrm{T}_4, \Gamma, \mathcal{P} \rangle$ (see discussion in Appendix C for a similar gate up to single-qubit gates).

Lemma 15. *Let G be an arbitrary n-qubit Clifford gate and suppose there are $2k-1$ invertible elements in the first column of G. Then G can be decomposed into the following sequence of gates (see Figure 9):*

1. *single-qubit gates on a subset S of the qubits,*
2. *an O_{2k-1} gate on the same subset S,*
3. *a generalized CNOT from a fixed $i \in S$ to each qubit outside S,*
4. *a* SWAP *on qubit 1 and qubit i (or no swap if $i = 1$),*
5. *a Pauli P on qubit 1,*
6. *an $(n-1)$-qubit Clifford gate D on qubits $2, \ldots, n$.*

Proof. Our goal is to show that G decomposes into these layers of gates:

$$G = (P \otimes D) \circ \mathrm{SWAP}(1, i) \circ \left(\prod \mathrm{CNOT}(r_j)\right) \circ O_{2k-1} \circ \left(\bigotimes \mathcal{G}(r_j)\right).$$

If we look at the matrix part only and isolate $P \otimes D$, we have

$$\mathcal{M}(\mathrm{SWAP}(1,i))\mathcal{M}\left(\prod \mathrm{CNOT}(r_j)\right)\mathcal{M}(O_{2k-1}^\dagger)\mathcal{M}\left(\bigotimes \mathcal{G}(r_j)^\dagger\right)\mathcal{M}(G) = \mathcal{M}(P \otimes D).$$

To prove the decomposition, we start with the first column of $\mathcal{M}(G)$ and show that this sequence of gates/matrices simplifies it. This process looks a bit like one step of Gaussian elimination: we pick a pivot row i, perform operations (i.e., gates) to zero out all other entries of the column, and swap the pivot row with the first row. Let $(r_1, \ldots, r_n)$ be the first column of $\mathcal{M}(G)$. Take $S = \{j \in \{1, \ldots, n\} : r_j \text{ is invertible in } \mathsf{R}\}$ to be the indices of the invertible elements, and let the *pivot* $i \in S$ be an arbitrary index (e.g., take the minimum element of S as the canonical choice). We note that S must have an odd number of elements because the tableau is symplectic.

For each invertible $r_j \in \mathsf{R}$, there exists a gate $\mathcal{G}(r_j)$ (unique up to Paulis/phase) which maps I to r_j. Thus, when we apply $\mathcal{G}(r_j)^\dagger$ to qubit j, it maps r_j to I. That is, the layer of single-qubit gates multiplying the rows such that the invertible elements of the column are sent to I. Next, we apply O_{2k-1} to the invertible qubits S to further simplify the column. Notice that O_{2k-1} is not completely symmetric—the first qubit is special, but all the others are interchangeable. Let the pivot row, i, be the special qubit when we apply O_{2k-1}. We leave it as an exercise to check that O_{2k-1} maps $(I, 0, \ldots, 0)$ to $(I, I, \ldots, I)$, and therefore $O_{2k-1}^\dagger$ will simplify the column to $(I, 0, \ldots, 0)$ on indices S.

Next, for each $r_j \notin S$, there exists a generalized CNOT gate $\mathrm{CNOT}(r_j)$ that maps (I, r_j) to $(I, 0)$. Since the ith entry of the column is now I, and the jth entry is r_j, applying $\mathrm{CNOT}(r_j)$ zeros out another entry of the column. After all generalized CNOTs have been applied, the column vector is all 0's except for an I in the pivot row. We apply a SWAP on qubit 1 and i to move this pivot row to the top, so the column vector must be $(I, 0, \ldots, 0)$. By the symplectic conditions, one can check that the first row must also be $(I, 0, \ldots, 0)$, and hence the tableau is of the form

$$\left(\begin{array}{c|ccc} I & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A & \\ 0 & & & \end{array}\right)$$

In other words, we are left with a Pauli on qubit 1 (since there may be sign bits) and a Clifford gate on qubits 2 through n. We choose P and D to be these components of the tableau, and that completes the decomposition. □

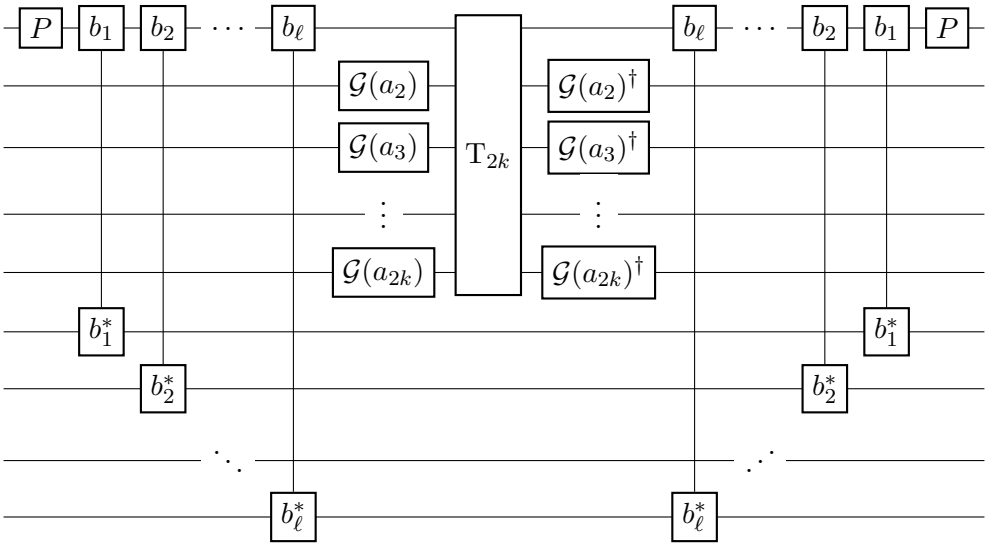


Figure 10: Decomposition of $\mathfrak{C}(G)$.

Corollary 16. *Let G be a Clifford gate on n qubits where the first column of the tableau is $(a_2, \ldots, a_{2k}, b_1, \ldots, b_\ell)$ for invertible $a_2, \ldots, a_{2k} \in \mathsf{R}$ and noninvertible $b_1, \ldots, b_\ell \in \mathsf{R}$. Then $\mathfrak{C}(G)$ is as shown in Figure 10: a T_{2k} on qubits 1 through $2k$, conjugated by single-qubit gates (except on qubit 1) derived from $a_2, \ldots, a_{2k}$, an array of generalized CNOT gates derived from $b_1, \ldots, b_\ell$ on either side between qubit 1 and qubits $2k+1$ through $n+1$, and conjugated by a Pauli P on qubit 1.*

Proof. Apply the decomposition (Lemma 15) to G. This decomposition ends with an $(n-1)$-qubit gate D; in the universal construction, D appears innermost in the circuit, next to the SWAP, acting on qubits 3 through $n+1$. Since D and the SWAP act on disjoint qubits, they commute, so D cancels with $D^\dagger$ on the other side (from the inverted decomposition for $G^\dagger$).

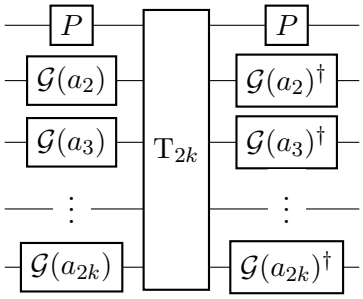
After cancelling D with $D^\dagger$, the next innermost gates are the Pauli P and generalized CNOT gates acting on qubit 2. Let us push these gates through the SWAP (from both sides) so that they all act on qubit 1 instead of 2, i.e., as they appear in Figure 10. At this point the SWAP is flanked by O_{2k-1} and $O_{2k-1}^\dagger$. It is a straightforward calculation to check that $\mathfrak{C}(O_{2k-1}) = \mathrm{T}_{2k}$, so we replace SWAP, O_{2k-1}, and $O_{2k-1}^\dagger$ with T_{2k}. Finally, the decomposition has single-qubit gates $\mathcal{G}(a_2), \ldots, \mathcal{G}(a_{2k})$, which are matched by their inverses $\mathcal{G}(a_2)^\dagger, \ldots, \mathcal{G}(a_{2k})^\dagger$ in $G^\dagger$, i.e., T_{2k} is conjugated by the appropriate single-qubit gates on qubits 2 through $2k$. □

We are finally ready to prove the main theorem of this section.

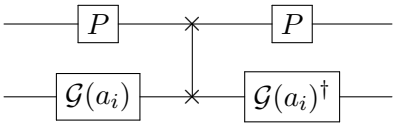
Theorem 17. *Let G be a Clifford gate on n qubits. Furthermore, let $(a_2, a_3, \ldots, a_{2k}, b_1, \ldots, b_\ell) \in \mathsf{R}^n$ be some column of $\mathcal{M}(G)$, where each a_i is invertible and each b_i is noninvertible. Then G generates a gate $\mathcal{G}_i$ such that $\mathcal{M}(\mathcal{G}_i) = a_i$ for each $i \in \{2, \ldots, 2k\}$, generates $\mathrm{CNOT}(b_j)$ for all $j \in \{1, \ldots, \ell\}$, and generates a T_{2k} gate.*

Proof. From Corollary 16, the universal construction $\mathfrak{C}(G)$ can be decomposed as shown in Figure 10. The proof will proceed in the following manner. Starting with the decomposition of $\mathfrak{C}(G)$, we show that it generates some elementary gate. We then use that gate to further simplify the original decomposition of $\mathfrak{C}(G)$ until we have generated all the gates specified in the theorem.

First notice that for each input $i \in \{2k+1, \ldots, n\}$, there exists a single-qubit Clifford state $|b_i\rangle$ that eliminates the generalized CNOT(b_i) gate (e.g., $|0\rangle$ on the control of a CNOT gate). Therefore, we can generate the following gate:

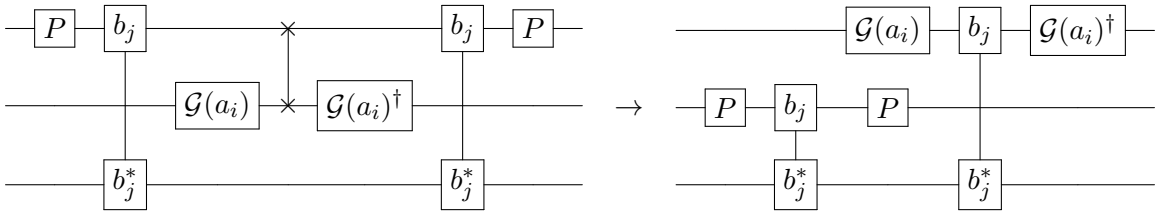


Now let $|\phi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ be the Bell state on two qubits. Notice that we can use $|\phi\rangle$ as an ancilla to remove two bits from $\mathrm{T}_{2\ell}$ (i.e., leaving a $\mathrm{T}_{2\ell-2}$). Therefore, the state $\mathcal{G}(a_i)^\dagger \otimes \mathcal{G}(a_j)^\dagger|\phi\rangle$ on bits i and j (for $i, j \neq 1$) is fixed by the circuit and simplifies the T_{2k} gate. We can iterate this procedure on the circuit above until it has been reduced to just two qubits. In particular, the T_{2k} gate in the middle is now a T_2, otherwise known as the SWAP gate. This remaining circuit is



Commute the first layer of gates through the swap and apply the swap rule. This leaves a tensor product of single-qubit gates, i.e., $P \circ \mathcal{G}(a_i)$ and its inverse, each of which we can isolate using the ancilla rule.

Now let us repeat the procedure above starting from $\mathfrak{C}(G)$, but stop short of applying the ancilla rule to qubit $2k+j$. The result is the first circuit depicted below, which is then simplified by swapping the first two qubits, and adding gates $P \circ \mathcal{G}(a_i), \mathcal{G}(a_i)^\dagger \circ P$:

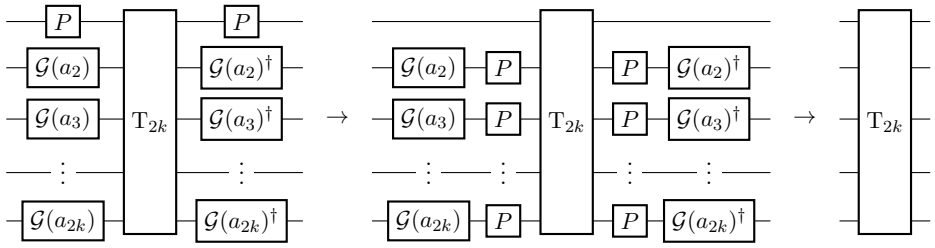


Notice that for the circuit on the right, we can apply the ancilla rule using the state $\mathcal{G}(a_i)^\dagger|b_j\rangle$ on the topmost qubit, leaving CNOT(b_j) conjugated by $P \otimes I$. Let us assume that CNOT(b_j) is a $C(Q, R)$ gate for some Paulis Q and R. If $Q = P$, then conjugation by $P \otimes I$ does nothing, and we obtain a CNOT(b_j) gate. Otherwise, the conjugation results in the gate $C(Q, R) \circ (I \otimes R)$ from which we can generate a Pauli R using the ancilla rule and the state stabilized by Q on the first qubit. In either case, we eventually generate a CNOT(b_j) gate.

Finally, to generate the T_{2k} we exploit the identity

$$\mathrm{T}_{2k}(P \otimes I^{\otimes 2k-1})\,\mathrm{T}_{2k} = I \otimes P^{\otimes 2k-1},$$

which holds up to a global phase. We have the following chain of consequences:



where the last implication follows by applying the gates $P \circ \mathcal{G}(a_i)$ and $\mathcal{G}(a_i)^\dagger \circ P$, which we generated previously. □

10 Completing the Classification

The final step in the classification is to demonstrate that the classes we have defined are in fact the only classes that exist. First, we give a simple consequence of Theorem 17 that will make it easier to talk about gate sets that violate some invariant.

Lemma 18. *Given that a gate set G violates some invariant I (either, a subring invariant or a permutation invariant), there is either a single-qubit gate, a generalized CNOT gate, or a T_4 gate in $\langle G \rangle$ that also violates the invariant.*

Proof. Let $g \in G$ be the gate violating the invariant. There are two cases:

I is a subring invariant: Since g violates the invariant, there is some entry $\mathcal{M}(g)_{ij}$ of the tableau outside the subring. Apply Theorem 17 to this column of g and it will extract either a generalized CNOT (if $\mathcal{M}(g)_{ij}$ is non-invertible) or single-qubit gate (if $\mathcal{M}(g)_{ij}$ is invertible) having the same entry, and thus violating the invariant.

I is a permutation invariant: The gate g could violate the invariant in one of two ways. The may be more than one invertible entry in some column of g, in which case applying Theorem 17 extracts a T_{2k} for $k \geq 2$, which generates a T_4, and T_4 violates all permutations invariants. Otherwise, there is some entry of $\mathcal{M}(g)_{ij}$ outside the allowed subring of elements, in which case we generate a single-qubit gate or generalized CNOT as in the other case.

□

Theorem 19. *Let G be any set of Clifford gates. If $\mathcal{S}$ is the smallest (with respect to lattice order) class of our classification containing G, then $\langle G \rangle = \mathcal{S}$.*

Proof. There is a very general strategy for proving that the class $\langle G \rangle$ is equal to the smallest $\mathcal{S}$ containing it. Since $\langle G \rangle \not\subseteq \mathcal{S}'$ for all $\mathcal{S}' \subsetneq \mathcal{S}$, there exists a gate $g \in G$ such that $g \notin \mathcal{S}'$. That is, for each such class $\mathcal{S}'$, there is an invariant (described in Section 6) that $\langle G \rangle$ fails to preserve. Using the universal construction, we can generate a simple gate in $\langle G \rangle$ which also fails to satisfy that invariant by Lemma 18. For the purposes of this proof let us call this set of gates (i.e., the single-qubit gates, generalized CNOT gates, and T_4 gates) the *elementary gates*. Finally, we will show that these elementary gates generate $\mathcal{S}$ itself (sometimes requiring the identities from Section 8).

We now give a complete sequence of tests to identify the class $\langle G \rangle$. It will be simpler to address the degenerate and non-degenerate gate classes separately, so let us assume for now that $\langle G \rangle$ is non-degenerate; we will tackle the degenerate classes at the end. The rest of the

case analysis depends on $\mathcal{S}$, the smallest class containing G. At the highest level, we separate these classes based on which of the X-, Y-, Z-preserving invariants hold for $\mathcal{S}$.

Suppose first that $\mathcal{S}$ is X-, Y-, and Z-preserving. There is only one non-degenerate class with these properties, so $\mathcal{S} = \langle \mathrm{T}_4, \mathcal{P} \rangle$. Since $\langle G \rangle$ is non-degenerate, there is some generalized CNOT gate or T_4 gate in $\langle G \rangle$ by Lemma 18. However, all generalized CNOT gates fail to preserve at least one of the bases and $\langle G \rangle \subseteq \mathcal{S}$ is X-, Y-, and Z-preserving. Therefore, the non-degenerate gate must be a T_4. Since T_4 generates the Pauli group (Lemma 11), we have that $\langle G \rangle \supseteq \mathcal{S}$, and so $\langle G \rangle = \mathcal{S}$.

Suppose now that $\mathcal{S}$ is P- and Q-preserving but not R-preserving for distinct Pauli operations P, Q, and R. Again, there is only one class with these properties, so $\mathcal{S} = \langle C(P,Q), \mathcal{P} \rangle$. Therefore, there exists some gate $g \in G$ which is not in $\mathcal{S}' = \langle \mathrm{T}_4, \mathcal{P} \rangle$. In other words, g fails to be R-preserving. Note now there is no single-qubit gate which is both P- and Q-preserving but not R-preserving. Furthermore, $C(P,Q)$ is the only generalized CNOT gate with this property. Therefore, by Lemma 18, g generates a $C(P,Q)$ gate. Clearly, $C(P,Q)$ generates both Pauli P and Q, and therefore the whole Pauli group. Hence, $\langle G \rangle = \mathcal{S}$.

Suppose now that $\mathcal{S}$ is P-preserving but not Q- and R-preserving. This is the most involved case and will require several more subdivisions. First suppose $\mathcal{S} = \langle C(P,Q), \mathrm{R}_P, \mathcal{P} \rangle$. We will use that G generates an elementary gate which does not preserve the Q-basis ($\mathcal{S}' = \langle C(P,Q), \mathcal{P} \rangle$), a gate which does not preserve the R-basis ($\mathcal{S}' = \langle C(P,R), \mathcal{P} \rangle$), and a gate that is not P-orthogonal ($\mathcal{S}' = \langle \mathrm{T}_4, \mathrm{R}_P, \mathcal{P} \rangle$). Notice that all P-preserving single-qubit gates are also P-orthogonal, and recall that $C(P,P)$ is P-orthogonal. Therefore, by Lemma 18, G generates $C(P,Q)$ or $C(P,R)$, both of which fail to be P-orthogonal. Assume without loss generality it is $C(P,Q)$. Let us now consider the gates that fail to be Q-preserving: if G generates a $C(P,R)$ or $C(P,P)$ gate, then G also generates a R_P gate by Lemma 10; otherwise, G generates a non-Q-preserving single-qubit gate, which must be R_P up to multiplication by Pauli elements. Since $C(P,Q)$ generates the Pauli group, we get that G generates an R_P gate, and so $\langle G \rangle = \langle C(P,Q), \mathrm{R}_P, \mathcal{P} \rangle = \mathcal{S}$.

Suppose now that $\mathcal{S}$ is the P-orthogonal class $\langle \mathrm{T}_4, \mathrm{R}_P, \mathcal{P} \rangle$. First, G generates a gate which is not P-degenerate ($\mathcal{S}' = \langle C(P,P), \mathcal{P}, \mathrm{R}_P \rangle$). Since the only P-orthogonal single-qubit gates are also P-degenerate, we have that G generates a T_4 by Lemma 18 ($C(P,P)$ is P-degenerate and neither $C(P,Q)$ nor $C(P,R)$ are P-orthogonal). Next, G generates a gate which is not both R- and Q-preserving ($\mathcal{S}' = \langle \mathrm{T}_4, \mathcal{P} \rangle$). This is either a $C(P,P)$ gate or an R_P gate up to Pauli operations. Since T_4 generates the Pauli group (Lemma 11) and T_4 plus $C(P,P)$ generate an R_Z gate (Lemma 12), then in either case $\mathrm{R}_P \in \langle G \rangle$, implying that $\langle G \rangle = \langle \mathrm{T}_4, \mathrm{R}_P, \mathcal{P} \rangle$.

Suppose now that $\mathcal{S}$ is P-degenerate. Let us handle all five possible P-degenerate classes together. First, since $\langle G \rangle$ is non-degenerate, it must contain a $C(P,P)$ gate. There are five P-degenerate classes correspond to the five P-preserving single-qubit classes containing P. Unlike previous cases, such a diversity of classes exists because $C(P,P)$ does not suffice to generate the Pauli group on its own. We now use the normal form for P-degenerate gates given in Lemma 7; that is, each gate can be expressed as a layer of single-qubit gates, a layer of $C(P,P)$ gates, and a permutation. We can extract all the single-qubit gates (let's call them G_1) by canceling the $C(P,P)$ gates (with more $C(P,P)$ gates), canceling the SWAP gates (with more SWAP gates), and using eigenstate ancillas. We have that $\langle G \rangle \subseteq \langle C(P,P), G_1 \rangle$ since every gate in G is composed of such gates, and $\langle G \rangle \supseteq \langle C(P,P), G_1 \rangle$ since we have shown how to generate all such gates. Therefore, $\langle G \rangle = \langle C(P,P), G_1 \rangle$.

Suppose now that $\mathcal{S}$ is the egalitarian class $\langle \mathrm{T}_4, \Gamma, \mathcal{P} \rangle$, which is neither P-,Q-, nor R-preserving. Since T_4 is the only egalitarian non-degenerate elementary gate, it can be generated in $\langle G \rangle$. Furthermore, it must contain a single-qubit gate that is egalitarian, but not P-,Q-, or R-preserving ($\mathcal{S}' = \langle \mathrm{T}_4, \mathcal{P} \rangle$). The only such single-qubit gates are Γ and $\Gamma^\dagger$ up to

Pauli operations. Therefore $\langle G\rangle = \langle \mathrm{T}_4, \Gamma, \mathcal{P}\rangle$ since T_4 generates the Pauli group (Lemma 11).

Finally, suppose that $\mathcal{S}$ is the class of all Clifford gates, violating all invariants. Let us organize the proof by which non-degenerate gates we generate from Lemma 18. The first case is that we get no generalized CNOT gates, only a T_4 gate. The immediate subclasses of $\mathcal{S}$ satisfy the X-, Y-, Z-preserving and egalitarian invariants, so G generates a gate violating each of these invariants. Since this cannot be T_4 (it satisfies all invariants), and we are assuming Lemma 18 gives us no generalized CNOT gates, these must be single-qubit gates. Clearly these single-qubit gates generate *all* single-qubit gates, so by Lemma 14, $\langle G\rangle = \mathsf{ALL}$.

Another possibility is that Lemma 18 produces one or more generalized CNOT gates. Our main tool is Lemma 13—if we have $C(P,Q)$, a non-P-preserving single-qubit gate, and a non-Q-preserving single-qubit gate (note: P and Q may be the same in this lemma!), then $\langle G\rangle = \mathsf{ALL}$. In some cases, the generalized CNOT gates themselves can provide the single-qubit gates, so any collection of generalized CNOTs containing one of the following subsets generates ALL.

- $\{C(P,P), C(Q,Q)\}$: These gates generate θ_{P+Q} by Lemma 10, which preserves neither the P nor Q basis.
- $\{C(P,P), C(Q,R)\}$: These gates generate Γ by Lemma 10, which does not preserve any basis.
- $\{C(P,Q), C(P,R), C(Q,R)\}$: These gates generate R_P, R_Q, and R_R by Lemma 10, which together do not preserve any basis.

In fact, the only exceptions are collections of generalized CNOT gates such that all gates are P-preserving (for at least one Pauli basis P), so let us suppose Lemma 18 gives us only P-preserving generalized CNOT gates and a non-P-preserving single-qubit gate. We are done if one of the generalized CNOT gates is $C(P,P)$ by Lemma 13—otherwise, we have one of the following sets:

- $\{C(P,Q)\}$: G generates a single-qubit gate which is not P-preserving and a single-qubit gate which is not Q-preserving by Lemma 18.
- $\{C(P,Q), C(P,R)\}$: G generates a single-qubit gate that is not P-preserving by Lemma 18. We note that any single-qubit gate which fails to be P-preserving must also fail to be either Q- or R-preserving.

This finishes the $\mathcal{S} = \mathsf{ALL}$ case analysis.

Let us now return to the degenerate classes where the argument will be similar to the P-degenerate gate classes. We will associate every degenerate class with a subgroup of the single-qubit Clifford gates. Explicitly, we decompose every degenerate gate into a circuit of single-qubit gates and SWAP gates by Lemma 6, and then extract all the single-qubit gates by canceling the SWAP gates (with more SWAP gates), and using eigenstate ancillas. Therefore, $\langle G\rangle$ is associated with a class of single-qubit gates, for which the classification is well known (see Figure 1). □

Corollary 20. *Given any set of gates G, there is a subset $S \subseteq G$ of at most three gates such that $\langle S\rangle = \langle G\rangle$.*

Proof. Suppose to the contrary that there exists a set of gates G such that $\langle G\rangle \neq \langle S\rangle$ for any subset $S \subseteq G$ of size ≤ 3. Without loss of generality, let G be a minimal set of gates with this property. If there are subsets $A, B \subseteq G$ such that $\langle A\rangle \subseteq \langle B\rangle$ but $A \not\subseteq B$ then we could delete $A \backslash B$ from G and it will generate the same class, but have strictly fewer subsets of size 3. It

follows that in a minimal G, all subsets of G generate distinct classes. But then we can further shrink G to any 4-element subset, say $G = \{g_1, g_2, g_3, g_4\}$, since that is distinct from all size ≤ 3 subsets.

Now we think of $\langle\cdot\rangle$ as a map embedding the power set lattice for 4 elements into our class lattice. We rule this out by case analysis. The first case is when $\langle G\rangle$ is degenerate. Notice that an ascending chain like

$$\varnothing \subseteq \{g_1\} \subseteq \{g_1, g_2\} \subseteq \{g_1, g_2, g_3\} \subseteq \{g_1, g_2, g_3, g_4\}$$

must map to a similar ascending chain in the degenerate class lattice (Figure 1). All such ascending chains go through $\mathcal{P}+\Gamma$, $\mathcal{P}+\mathrm{R}_X$, $\mathcal{P}+\mathrm{R}_Y$, or $\mathcal{P}+\mathrm{R}_Z$, and end at $\top$, so the 4 distinct classes $\langle g_1, g_2, g_3\rangle, \ldots, \langle g_2, g_3, g_4\rangle$ must correspond to those 4 cases. Let's say $\langle g_1, g_2, g_3\rangle = \langle\mathcal{P}+\Gamma\rangle$. However, there are 6 ascending chains through $\{g_1, g_2, g_3\}$, and only 3 chains going through $\mathcal{P}+\Gamma$, so two subsets of G map to identical classes, a contradiction.

The more involved case is when $\langle G\rangle$ is non-degenerate. One of the gates must generate a non-degenerate class (otherwise $\langle G\rangle$ would be degenerate). Again, we consider ascending chains in the lattice. For instance, $\langle g_1\rangle, \ldots, \langle g_4\rangle$ cannot contain $C(P, Q)$ (for Paulis $P \neq Q$) because the chains above those classes are insufficiently long (i.e., at best two hops up to ALL—we need three). With the exception of ALL, no class containing $C(P, Q)$ is the join of two classes *not* containing $C(P, Q)$. Therefore no proper subset of G generates any $C(P, Q)$ gate.

Now imagine removing classes (except ALL) which contain $C(P, Q)$. With those classes gone, the ascending chains above the classes containing T_4 are too short for any of them to be a generator, just like we argued for $C(P, Q)$. Similarly, these T_4-containing classes (excluding ALL) are not the join of any classes *not* containing T_4. Once we remove *those*, the only remaining non-degenerate classes contain $C(P, P)$ for some P, so suppose $C(P, P) \in \langle g_1\rangle$. If $\langle g_2\rangle$ is not also P-preserving, then we see from the lattice that $\langle g_1, g_2\rangle = \mathsf{ALL} = \langle g_1, g_2, g_3\rangle$, a contradiction. Hence, all of the classes are P-preserving. But there are only 5 P-preserving classes left in the lattice, and 7 distinct classes generated by g_1 and proper subsets of $\{g_2, g_3, g_4\}$. Hence, $\langle G\rangle$ cannot be non-degenerate either.

We conclude that for any gate set G, there is a subset of at most 3 gates that generates it. This is tight, since $C(X, Y)$, $C(Y, Z)$, and $C(Z, X)$ generate all Clifford gates, but any pair of them cannot generate the third. □

11 Open Problems

Our classification of Clifford gates resolves an open problem of Aaronson et al. [1], but leaves their central question, the classification of arbitrary quantum gates, completely open. It is unclear whether there is another piece of the full quantum gate classification that can be peeled off. Other discrete quantum gate sets are known, but none are known to have the rich structure and entanglement of Clifford gates (aside from conjugated Clifford gates). So we ask: are there other interesting discrete gate sets, and can they be classified like Clifford gates?

Another source of open problems is the choice of ancilla rule. As discussed, we permit ancillas initialized to arbitrary quantum states. We have determined that the classification continues to hold under a stabilizer state ancilla model if the following conjecture holds:

Conjecture 21. *For any single-qubit Clifford gate g, there exists a stabilizer state $|\psi\rangle$ and circuit of* SWAP *gates π such that $g \circ \pi|\psi\rangle = |\psi\rangle$.*

This is sufficient to remove single-qubit gates in situations where we would otherwise use an eigenstate.

For many single-qubit gates, there is a trivial stabilizer eigenstate. For instance, X stabilizes $|+\rangle$, R_Z stabilizes $|0\rangle$, and many other single-qubit Clifford gates are conjugate to one of these cases. Now consider the gate θ_{X+Z}, whose eigenstates (unnormalized) $(1 \pm \sqrt{2})|0\rangle + |1\rangle$ are *not* stabilizer states. How then, given the gate $\theta_{X+Z} \otimes \theta_{X+Z}$, does one generate the gate θ_{X+Z} which acts only on one qubit? Han-Hsuan Lin discovered the first explicit nine qubit stabilizer state for this task.

Let π be a circuit that cyclically permutes qubits 2 through 9, and suppose θ_{X+Z} is applied to qubit 1. Let $|\psi\rangle$ be the state stabilized by the following commuting Pauli strings,

$$\begin{array}{llll} XXZXZIIII, & ZIXZXZIII, & XIIXZXZII, & ZIIIXZXZI, \\ XIIIIXZXZ, & ZZIIIIXZX, & XXZIIIIXZ, & ZZXZIIIIX, \\ YYIIIYIII, & -YIYIIIYII, & YIIYIIIYI, & -YIIIYIIIY, \end{array}$$

9 of which are independent. One can check that conjugating each generator by $\theta_{X+Z} \circ \pi$ yields another element of the stabilizer group, so $(\theta_{X+Z} \circ \pi)|\psi\rangle = |\psi\rangle$. In other words, Conjecture 21 holds for θ_{X+Z}, and for all conjugates θ_{P+Q} by symmetry.

To verify the conjecture, it suffices to find a stabilizer state $|\psi\rangle$ and circuit C, constructed of SWAP gates and a single Γ gate, such that $C|\psi\rangle = |\psi\rangle$.

12 Acknowledgments

Much on this work was completed at MIT, where both authors were graduate students. DG additionally acknowledges the support an NSF Graduate Research Fellowship under Grant No. 1122374. We would like to thank Scott Aaronson for his guidance throughout this project. Also, thanks to Han-Hsuan Lin and Adam Bouland for useful discussions. Finally, we thank the many anonymous reviewers that have helped to improve the manuscript. In particular, we thank the reviewer that pointed out an error in an attempted proof of Conjecture 21.

References

[1] Scott Aaronson, Daniel Grier, and Luke Schaeffer. "The classification of reversible bit operations". In 8th Innovations in Theoretical Computer Science Conference. Volume 67 of Leibniz International Proceedings in Informatics, pages 23:1–23:34. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2017).

[2] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. "Elementary gates for quantum computation". Physical Review A **52**, 3457–3467 (1995).

[3] Yaoyun Shi. "Both Toffoli and controlled-NOT need little help to do universal quantum computing". Quantum Information & Computation **3**, 84–92 (2003).

[4] Adam Bouland, Laura Mančinska, and Xue Zhang. "Complexity classification of two-qubit commuting Hamiltonians". In 31st Conference on Computational Complexity. Volume 50 of Leibniz International Proceedings in Informatics, pages 28:1–28:33. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2016).

[5] Andrew M. Childs, Debbie Leung, Laura Mancinska, and Maris Ozols. "Characterization of universal two-qubit Hamiltonian". Quantum Information & Computation **11**, 19–39 (2011).

[6] Adam Bouland and Scott Aaronson. "Generation of universal linear optics by any beam splitter". Physical Review A **89**, 062316 (2014).

[7] Matthew Amy, Andrew N Glaudell, and Neil J Ross. "Number-theoretic characterizations of some restricted Clifford+T circuits". Quantum **4**, 252 (2020).

[8] Daniel Gottesman. "The Heisenberg representation of quantum computers" (1998). arXiv:quant-ph/9807006.
[9] Scott Aaronson and Daniel Gottesman. "Improved simulation of stabilizer circuits". Physical Review A **70**, 052328 (2004).
[10] Daniel Gottesman. "Stabilizer codes and quantum error correction". PhD thesis. California Institute of Technology. (1997).
[11] Peter W. Shor. "Fault-tolerant quantum computation". In Proceedings of 37th Conference on Foundations of Computer Science. Pages 56–65. (1996).
[12] Andrew Steane. "Multiple-particle interference and quantum error correction". Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551–2577 (1996).
[13] Sergey Bravyi and Alexei Kitaev. "Universal quantum computation with ideal Clifford gates and noisy ancillas". Physical Review A **71**, 022316 (2005).
[14] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. "Measurement-based quantum computation on cluster states". Physical Review A **68**, 022312 (2003).
[15] Jonas T. Anderson. "On the power of reusable magic states" (2012). arXiv:1205.0289.
[16] Panos Aliferis. "Level reduction and the quantum threshold theorem". PhD thesis. California Institute of Technology. (2007).
[17] N. Cody Jones, Rodney Van Meter, Austin G. Fowler, Peter L. McMahon, Jungsang Kim, Thaddeus D. Ladd, and Yoshihisa Yamamoto. "Layered architecture for quantum computing". Physical Review X **2**, 031007 (2012).
[18] Sergey Bravyi and Dmitri Maslov. "Hadamard-free circuits expose the structure of the Clifford group". IEEE Transactions on Information Theory **67**, 4546–4563 (2021).
[19] Dmitri Maslov and Martin Roetteler. "Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations". IEEE Transactions on Information Theory **64**, 4729–4738 (2018).
[20] Peter Selinger. "Generators and relations for n-qubit Clifford operators". Logical Methods in Computer Science**11** (2015).
[21] Jeroen Dehaene and Bart De Moor. "Clifford group, stabilizer states, and linear and quadratic operations over GF(2)". Physical Review A **68**, 042318 (2003).
[22] Maarten Van den Nest. "Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond". Quantum Information & Computation **10**, 258–271 (2010).
[23] Andrew N Glaudell, Neil J Ross, and Jacob M Taylor. "Optimal two-qubit circuits for universal fault-tolerant quantum computation". npj Quantum Information **7**, 1–11 (2021).
[24] Wikipedia. "Hyperoctahedral group". `http://en.wikipedia.org/w/index.php?title=Hyperoctahedral%20group&oldid=1079812980` (2022). [Online; accessed 22-April-2022].
[25] Daniel Gottesman. "Theory of fault-tolerant quantum computation". Physical Review A **57**, 127 (1998).
[26] Peter Selinger. "Dagger compact closed categories and completely positive maps". Electronic Notes in Theoretical Computer Science **170**, 139–163 (2007).
[27] Michael Beverland, Earl Campbell, Mark Howard, and Vadym Kliuchnikov. "Lower bounds on the non-Clifford resources for quantum computations". Quantum Science and Technology **5**, 035009 (2020).
[28] Daniel Jonathan and Martin B Plenio. "Entanglement-assisted local manipulation of pure quantum states". Physical Review Letters **83**, 3566–3569 (1999).
[29] Emil L. Post. "The two-valued iterative systems of mathematical logic". Number 5 in Annals of Mathematics Studies. Princeton University Press. (1941).

[30] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. "Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates". Quantum Information & Computation **13**, 607–630 (2013).

[31] Brett Giles and Peter Selinger. "Exact synthesis of multiqubit Clifford+T circuits". Physical Review A **87**, 032332 (2013).

[32] G. E. Wall. "On the conjugacy classes in the unitary, symplectic and orthogonal groups". Journal of the Australian Mathematical Society **3**, 1–62 (1963).

A Enumeration

Theorem 22. *Let $\#\langle\cdot\rangle_n$ denote the number of n-qubit gates in a class. Then*

$$\begin{aligned}
\#\langle G\rangle_n &= |G|^n\, n! && \text{for } G \text{ a group of single-qubit gates,}\\
\#\langle C(Z,Z), G\rangle_n &= |G|^n\, 2^{n(n-1)/2} n! && \text{for } \langle Z\rangle_1 \subseteq G \subseteq \langle \mathcal{P}, \mathrm{R}_Z\rangle_1 \text{ a group,}\\
\#\langle C(Z,X), \mathcal{P}\rangle_n &= 4^n 2^{n(n-1)/2} \prod_{i=1}^{n} (2^i - 1),\\
\#\langle C(Z,X), \mathcal{P}, \mathrm{R}_Z\rangle_n &= 8^n 2^{n(n-1)} \prod_{i=1}^{n} (2^i - 1),\\
\#\langle \mathrm{T}_4, \mathcal{P}\rangle_n &= 4^n a(n),\\
\#\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_Z\rangle_n &= 8^n 2^{n(n-1)/2} a(n),\\
\#\langle \mathrm{T}_4, \mathcal{P}, \Gamma\rangle_n &= 4^n 2^{n(n-1)/2} \prod_{i=1}^{n} (2^i - (-1)^i),\\
\#\langle \mathsf{ALL}\rangle_n &= 4^n 2^{n^2} \prod_{i=1}^{n} (4^i - 1),
\end{aligned}$$

where

$$a(n) = \begin{cases} 2^{m^2} \prod_{i=1}^{m-1} (2^{2i} - 1), & \text{if } n = 2m,\\ 2^{m^2} \prod_{i=1}^{m} (2^{2i} - 1), & \text{if } n = 2m+1. \end{cases}$$

Proof. Most of these numbers follow from the lemmas above. For example, consider the class $\langle C(Z,X), \mathcal{P}, \mathrm{R}_Z\rangle$. It follows from Lemma 8 that any gate in this class has a circuit consisting of a layer of $C(Z,X)$ gates, then a layer of $C(Z,Z)$ gates, then a layer of single-qubit gates in G.

We would like to count the number of possible gates by multiplying the number of possibilities for each layer, but we must be careful that there is no gate with two circuit representations. Suppose for a contradiction that g_1 and g_2 generate the same gate, but some layer of g_1 differs from g_2. Then $g_1^{-1} g_2$ is the identity, since g_1 and g_2 generate the same transformation.

On the other hand, the $C(Z,X)$ layers of g_1 and g_2 meet in the middle of the circuit for $g_1^{-1} g_2$. If those layers do not generate the same linear transformation, then the combination is some non-trivial linear transformation which is, in particular, not Z-degenerate. The other layers of g_1 and g_2 *are* Z-degenerate, so we conclude that $g_1^{-1} g_2$ is not Z-degenerate (if it were, we could invert the outer layers to show that the two middle layers are Z-degenerate). But $g_1^{-1} g_2 = I$ is clearly Z-degenerate, therefore the $C(Z,X)$ layers of g_1 and g_2 must generate the same linear transformation.

The $C(Z,X)$ layers of g_1 and g_2 cancel (since we have shown they are equivalent), so they effectively disappear, and we make a similar argument about the $C(Z,Z)$ layers, and then the

single-qubit layers. That is, if the $C(Z,Z)$ layers do not contain the same set of $C(Z,Z)$ gates, then we obtain a contradiction because they produce a non-degenerate layer in the middle, implying that $g_1^{-1}g_2 = I$ is non-degenerate. Once we remove the $C(Z,Z)$ layers, the single-qubit layers must be the same or they would leave behind a non-trivial single-qubit gate. We conclude that all layers of g_1 and g_2 are actually the same, so the number of gates is the product of the number of choices for each layer.

Now the problem is to count the number of choices for each layer. For the single-qubit layer, this is clearly just n independent choices of single-qubit gate from $\langle \mathcal{P}, \mathrm{R}_Z \rangle$, or 8^n. For the $C(Z,Z)$ layer, there is a choice whether or not to place a $C(Z,Z)$ gate in each of the $\binom{n}{2}$ possible positions, so $2^{n(n-1)/2}$ choices for the layer. For the $C(Z,X)$ layer, observe that $C(Z,X)$ generate precisely the set of invertible linear transformations, of which there are

$$2^{n(n-1)/2} \prod_{i=1}^{n} (2^i - 1)$$

by a classical argument. Multiplying the three layers, we have a total of

$$\#\left\langle C(Z,X), \mathcal{P}, \mathrm{R}_Z \right\rangle_n = 8^n 2^{n(n-1)} \prod_{i=1}^{n} (2^i - 1)$$

n-qubit transformations generated by $C(Z,X)$, $\mathcal{P}$, and R_Z.

The numbers for $\langle G \rangle$, $\langle C(Z,Z), G \rangle$, $\langle C(Z,X), \mathcal{P} \rangle$, $\langle \mathrm{T}_4, \mathcal{P} \rangle$, and $\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_Z \rangle$ follow by a similar argument, although for the last two classes we need the fact that T_4 generates

$$a(n) = \begin{cases} 2^{m^2} \prod_{i=1}^{m-1} (2^{2i} - 1), & \text{if } n = 2m, \\ 2^{m^2} \prod_{i=1}^{m} (2^{2i} - 1), & \text{if } n = 2m+1. \end{cases}$$

orthogonal transformations on n qubits.

For the final two classes, we use known expressions (see, e.g., equations (2.6.1) and (2.6.2) from [32]) for the number of $n \times n$ unitary matrices over $\mathbb{F}_4$ (in the case of $\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle$) and for the number of $2n \times 2n$ symplectic matrices over $\mathbb{F}_2$ (in the case of ALL). We multiply by 4^n in both cases to account for the phase bits, which are completely independent of the matrix part. □

Theorem 23. *The asymptotic size of each class is as follows.*

$$\begin{aligned}
\log_2 \#\left\langle G \right\rangle_n &= n \log_2(|G|) + n \log_2 n - n \log_2 e + \frac{1}{2} \log_2 2\pi + O\left(\frac{1}{n}\right), \\
\log_2 \#\left\langle C(Z,Z), G \right\rangle_n &= n \log_2(|G|) + \frac{n(n-1)}{2} + n \log_2 n - n \log_2 e + \frac{1}{2} \log_2 2\pi + O\left(\frac{1}{n}\right), \\
\log_2 \#\left\langle C(Z,X), \mathcal{P} \right\rangle_n &= n^2 + 2n - \alpha + O(2^{-n}), \\
\log_2 \#\left\langle C(Z,X), \mathcal{P}, \mathrm{R}_Z \right\rangle_n &= \frac{3}{2}n^2 + \frac{5}{2}n - \alpha + O(2^{-n}), \\
\log_2 \#\left\langle \mathrm{T}_4, \mathcal{P} \right\rangle_n &= \frac{1}{2}n^2 + \frac{3}{2}n - \beta + O(2^{-n}), \\
\log_2 \#\left\langle \mathrm{T}_4, \mathcal{P}, \mathrm{R}_Z \right\rangle_n &= n^2 + 3n - \beta + O(2^{-n}), \\
\log_2 \#\left\langle \mathrm{T}_4, \mathcal{P}, \Gamma \right\rangle_n &= n^2 + 2n + \gamma + O(2^{-n}), \\
\log_2 \#\left\langle \mathsf{ALL} \right\rangle_n &= 2n^2 + 3n - \beta + O(4^{-n}).
\end{aligned}$$

where G is the same as in Theorem 22, and

$$\alpha = -\sum_{i=1}^{\infty} \log_2(1 - 2^{-i}) \approx 1.7919,$$
$$\beta = -\sum_{i=1}^{\infty} \log_2(1 - 4^{-i}) \approx 0.53839,$$
$$\gamma = \sum_{i=1}^{\infty} \log_2(1 - (-2)^{-i}) \approx 0.27587.$$

Proof. We take the logarithm of each class size, which we can separate into the logarithm of each layer comprising that class, as in Theorem 22. For most layers this is straightforward, except for the layer of permutations, orthogonal transformations, or general linear transformations. The first we handle with Stirling's approximation. For the other two, we factor out powers of two leaving a partial sum of a convergent series, which we analyze with a Taylor expansion. The classes $\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle_n$ and $\langle \mathsf{ALL} \rangle_n$ follow by similar techniques. □

Corollary 24. *Let $\mathcal{C}$ be any class, and let G be an n-qubit gate chosen uniformly at random from $\mathcal{C}$. Then*

$$\Pr[G \text{ generates } \mathcal{C}] = 1 - O(2^{-n}).$$

B Classical reversible gates with quantum ancillas

In this section we describe what the classical reversible gate lattice of Aaronson et al. [1] would look like under quantum ancillas. We extend all classical gates discussed in that paper to the quantum setting in the most natural way. Figure 12 shows the new (dramatically simpler) lattice.

Some of the collapses in the lattice are immediate. For instance, the class $\langle \mathrm{NOT} \otimes \mathrm{NOT} \rangle$ collapses with the class $\langle \mathrm{NOT} \rangle$ because $\mathrm{NOT}\,|+\rangle = |+\rangle$. A similar collapse occurs been all classes where the parity issue arises, such as between the classes $\langle \mathrm{CNOTNOT} \rangle$ and $\langle \mathrm{CNOT} \rangle$.

A more interesting collapse occurs between all mod-k-preserving classes for $k \geq 2$. Consider the following gate $G : \{0,1\}^k \to \{0,1\}^k$ of order 2 which preserves Hamming weight mod k:

$$G(0^k) = 1^k$$
$$G(1^k) = 0^k$$
$$G(1^a 0^b 0) = 1^{a-1} 0^{b+1} 1$$
$$G(1^a 0^b 1) = 1^{a+1} 0^{b-1} 0$$

where G acts as the identity on all other inputs. Since G preserves the Hamming weight mod k, it must appear in the class. We will show that G can generate a NOT gate. Let

$$|\psi_k\rangle := \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} \left| 1^i 0^{k-i-1} \right\rangle$$

so, for example

$$|\psi_4\rangle = \frac{|000\rangle + |100\rangle + |110\rangle + |111\rangle}{2}.$$

Now, for $b \in \{0,1\}$, $G(|\psi_k\rangle|b\rangle) = |\psi_k\rangle|b \oplus 1\rangle$. Therefore, each mod-$k$-preserving class for $k \geq 2$ collapses to the $\langle \mathrm{Fredkin}, \mathrm{NOT} \rangle$ class. Furthermore, Figure 11 shows that the Fredkin and NOT

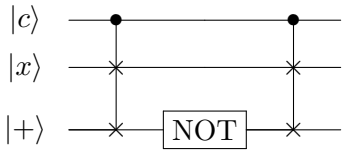


Figure 11: Generating CNOT from Fredkin and NOT gates

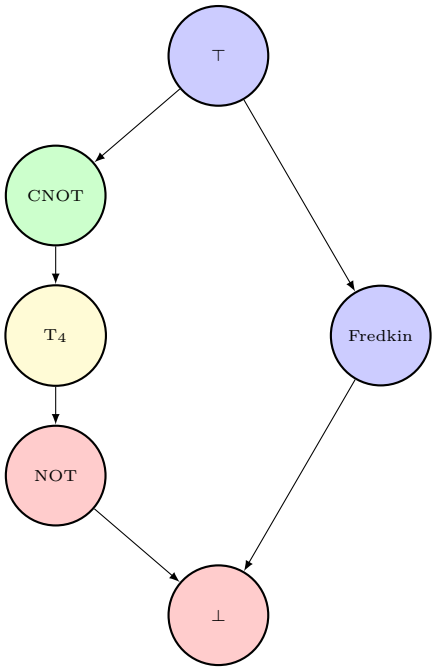


Figure 12: The inclusion lattice of classical gates using quantum ancillas.

gates are sufficient to generate a CNOT. Therefore, every non-conservative non-affine class generates all classical reversible transformations.

We now only need to prove that the classes appearing in Figure 12 are distinct. Notice, however, that the classes $\langle \mathrm{CNOT} \rangle$, $\langle \mathrm{T}_4 \rangle$, and $\langle \mathrm{NOT} \rangle$ all have Clifford gate generators, which by the results of this paper, generate distinct classes. We only need to show that the $\langle \mathrm{Fredkin} \rangle$ class is distinct from the remaining classes. However, the invariant in [1] more or less functions to prove this separation. Namely, Fredkin conserves the Hamming weight of its input. Therefore the sum of the Hamming weights of the computational basis states of the input state is conserved. However, the NOT gate necessarily changes this sum, witnessing that $\mathrm{NOT} \notin \langle \mathrm{Fredkin} \rangle$, and therefore that the lattice is complete.

C Three-qubit generator for $\langle \mathrm{T}_4, \Gamma, \mathcal{P} \rangle$

The T_4 gate is a minimal generator for a class of orthogonal gates, both in our classification (i.e., $\langle \mathrm{T}_4, \mathcal{P} \rangle$), and in the Aaronson et al. [1] classification. Surprisingly, when we add Γ gates to this class, it has a *three*-qubit generator. This is most easily seen by counting (using the enumeration results from Appendix A):

$$\begin{aligned}
&\#\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle_3 = 2^{3(3-1)/2+2(3)} \prod_{i=1}^{3} (2^i - (-1)^i) = 41472, \\
&\#\langle \Gamma, \mathcal{P} \rangle_3 = 12^3 3! = 10368, \\
&\#\langle \mathrm{T}_4, \mathcal{P} \rangle_3 = 4^3 2^{1^2} \prod_{i=1}^{1} (2^{2i} - 1) = 384, \\
&\#\langle \mathcal{P} \rangle_3 = 4^3 3! = 384.
\end{aligned}$$

We see that $\langle \mathrm{T}_4, \mathcal{P} \rangle$ and $\langle \mathcal{P} \rangle$ have the same number of gates on three qubits, but $\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle$ has substantially more gates than $\langle \Gamma, \mathcal{P} \rangle$. Notice that there are 4 cosets of $\langle \Gamma, \mathcal{P} \rangle_3$ in $\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle_3$ by Lagrange's Theorem, corresponding to 4 gates that are nonequivalent up to applications of elements in $\langle \Gamma, \mathcal{P} \rangle_3$. If we let $\alpha = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right) \in \mathsf{R}$, then one such gate is described by the following tableau

$$\left(\begin{array}{ccc|c} \alpha & I & I & 0 \\ I & \alpha & I & 0 \\ I & I & \alpha & 0 \end{array}\right).$$

This is equal to the gate O_3 (up to single-qubit gates) that appears in Section 9. By Theorem 19, this gate generates all of $\langle \mathrm{T}_4, \mathcal{P}, \Gamma \rangle$.

D Canonical form from Section 9

Recall the decomposition for an n-qubit Clifford gate G in Lemma 15:

$$G = (P \otimes D) \circ \mathrm{SWAP}(1, i) \circ \left(\prod \mathrm{CNOT}(r_j) \right) \circ O_{2k-1} \circ \left(\bigotimes \mathcal{G}(r_j) \right). \tag{1}$$

It decomposes as single-qubit gates, an O_{2k-1} gate, generalized CNOT gates, an optional SWAP, a Pauli on the first qubit, and an arbitrary Clifford gate D on the remaining $(n-1)$ qubits. Conceptually, the decomposition is a mapping $\mu \colon \mathcal{C}_n \to \mathcal{C}_{n-1} \times Q_n$, taking an n-qubit Clifford to an $(n-1)$-qubit Clifford (i.e., D) and some subset of Clifford gates Q_n (i.e., all gates except D in the decomposition). In this appendix we will argue that μ is actually a bijection, leading to a nice canonical form for Clifford gates.

First, μ is clearly one-to-one since we can multiply the decomposition out to recover the original Clifford G. To show that μ is bijective, we simply need to argue that the domain and codomain have the same size. We conveniently have an expression for the number of Clifford operations from Appendix A, and can compute the ratio

$$\frac{|\mathcal{C}_n|}{|\mathcal{C}_{n-1}|} = \frac{4^n 2^{n^2} \prod_{i=1}^{n}(4^i - 1)}{4^{n-1} 2^{(n-1)^2} \prod_{i=1}^{n-1}(4^i - 1)} = 2^{2n+1}(4^n - 1).$$

Lemma 25. *The decomposition produces $|Q_n|$ possible n-qubit Clifford circuits (excluding D) where*

$$|Q_n| = 2^{2n+1}(4^n - 1).$$

Proof. We will count the number of configurations for the gates in each layer of the decomposition. A key parameter is the number of invertible elements in the first column of $\mathcal{M}(G)$, which determines the number of single-qubit gates we apply. Let S be the set of indices for these invertible elements with $s := |S|$.

There are 6 single-qubit gates (corresponding to the 6 invertible elements of R), and we apply one such gate to each qubit in S for a total of 6^s possible single-qubit layers given S. Next, we consider the non-invertible elements in the column, corresponding to the 10 choices of generalized CNOT gate (including the identity gate). Since we apply a CNOT targeting each qubit *not* in S, this gives 10^{n-s} choices for this layer. Finally, we have four choices for a Pauli P. The remaining degrees of freedom in the circuit all have a canonical choice: take i as the first invertible element in S, perform the generalized CNOT gates (which may not commute) in the order of their target (not i), and likewise for the single-qubit gates (except they *do* commute). The targets of the SWAP are determined by i, and we omit it if and only if $i = 1$. The targets of the O_s gate are S, and the orientation is fixed by i.

Now we sum over s and count the number of choices for S (i.e., $\binom{n}{s}$), the single-qubit gates (6^s), and generalized CNOT gates (10^{n-s})).

$$\begin{aligned}
|Q_n| &= 4 \cdot \sum_{s \text{ odd}} \binom{n}{s} 6^s 10^{n-s} \\
&= 2 \cdot \left(\sum_{s=0}^{n} \binom{n}{s} 6^s 10^{n-s} - \sum_{s=0}^{n} \binom{n}{s} (-6)^s 10^{n-s} \right) \\
&= 2\left((6+10)^n - (10-6)^n\right) \\
&= 2^{2n+1}(4^n - 1).
\end{aligned}$$

□

It follows that $|\mathcal{C}_n|/|\mathcal{C}_{n-1}| = 2^{2n+1}(4^n - 1) = |Q_n|$, so the decomposition is a bijection.

E Canonical form for 2-qubit circuits

In this section, we describe a very clean canonical form for 2-qubit Clifford circuits.

Theorem 26. *Let G be any Clifford circuit on two qubits. Then, G is equivalent to a circuit of at most depth 3 composed of the following sequence of gates*

1. *a* SWAP*, and*
2. *a tensor product of single-qubit gates, and*
3. *a generalized CNOT gate,*

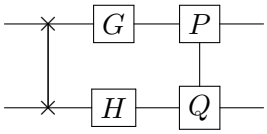


Figure 13: Canonical form of a 2-qubit circuit: optional SWAP gate, single-qubit gates G and H, and $C(P, Q)$ gate.

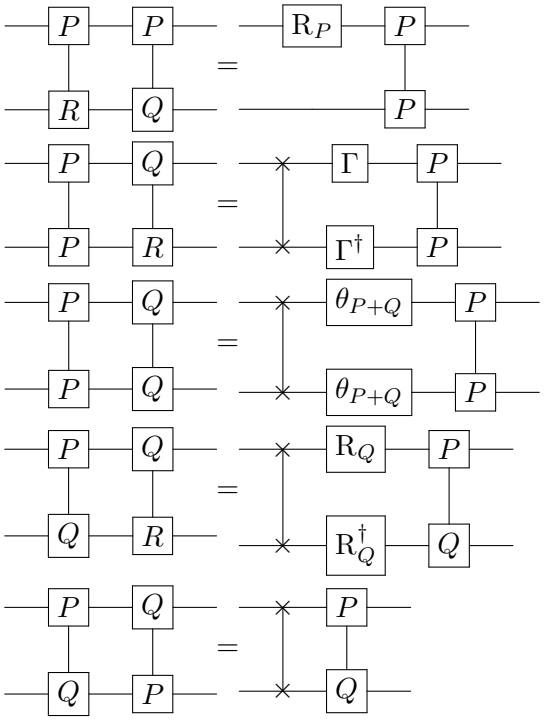


Table 5: Rules for coalescing generalized CNOT gates, assuming $\Gamma P \Gamma^\dagger = Q$ and $\Gamma Q \Gamma^\dagger = R$.

where we can choose at each step whether or not to include the gate. That is, G is of the form of the circuit depicted in Figure 13.

Proof. Since G is a Clifford circuit, it can be written as a product of CNOT, θ_{X+Z} (Hadamard), and R_Z gates. Recall that conjugating a generalized CNOT gate by a single-qubit gate is simply another generalized CNOT gate. Therefore, we can push all the single-qubit gates left and all the generalized CNOT gates right. All that remains to show is that we can coalesce the generalized CNOT gates into a single CNOT gate. We refer to Table 5 for those equivalences, and note that identical generalized CNOT gates cancel. Eventually, what remains is a circuit composed of single-qubit gates, SWAP gates, and at most one generalized CNOT gate. We can push the SWAP gates to the left (they collapse to either a single SWAP gate or the identity) and combine the single-qubit gates, which completes the proof. □