Technical description of hidden stabilizer circuits

qBitTensor Labs

Contents

1 Background

Previously, we opened Aaronson and Zhang's peaked circuits up to the Bittensor community as Phase 1 of our mission to democratize quantum computing. Although peaked circuits are a promising step toward verifiable quantum computation, we note some practical concerns in the context of Bittensor–the foremost of which is that they are computationally demanding to generate. Here, we introduce a modification of the peaked circuit problem to Subnet 63, one we call the "hidden stabilizers" problem.

This problem is built on well-known work by Daniel Gottesman, Emanuel Knill, and Scott Aaronson exploring the *stabilizer formalism* and the implications of the so-called *Gottesman-Knill theorem*. Both (to be defined below) are integral to the current and foremost understanding of quantum error correction, and involve several interesting mathematical structures we will hence leverage for extended benchmarking.

This section will give critical background for understanding the structure of the problem subsequently defined in Section 2 and suggested solution strategies given in Section 3. Finally, Section 4 will give the difficulty scaling functions for this problem.

1.1 Pauli operators and Clifford gates

Pauli operators are the first important element. They are defined as complex-valued 2×2 matrices

$$\sigma_1 \equiv X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_2 \equiv Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_3 \equiv Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1}$$

and are ubiquitous throughout quantum mechanics–for example, they act as Hermitian operators defining the properties of spin-1/2 particles, and the reader may recognize X and Z as the single-qubit "bit-flip" and "phase-flip" gates. It is also sometimes useful to define the identity matrix as a "zero-th" Pauli matrix, as we will do here:

$$\sigma_0 \equiv I \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2}$$

Mathematically, these four matrices form a basis for 2×2 Hermitian matrices over the real numbers, all complex-valued 2×2 matrices over the complex numbers, and a generating set for the Lie group SU(2). More concretely, these matrices satisfy the following product relations:

$$\sigma_i \sigma_j = \delta_{ij} \sigma_0 + i \varepsilon_{ijk} \sigma_k \qquad \sigma_k^2 = -i \sigma_1 \sigma_2 \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \sigma_0 \tag{3}$$

with δ_{ij} the Kronecker delta and ε_{ijk} the Levi-Civita symbol,

$$\delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \qquad \varepsilon_{ijk} = \begin{cases} +1 \text{ if } (i,j,k) \text{ is an even permutation of } (1,2,3) \\ -1 \text{ if } (i,j,k) \text{ is an odd permutation of } (1,2,3) \\ 0 \text{ otherwise} \end{cases}. \tag{4}$$

These product relations admit the structure of a *group* on the set

$$\mathcal{P} = \{ c\sigma_k \mid k \in \{0, ..., 3\}, c \in \{\pm 1, \pm i\} \}, \tag{5}$$

with ordinary matrix multiplication as the group operation; i.e. for all $\sigma, \rho \in \mathcal{P}$, the product and inverses of σ and ρ are also elements of $\mathcal{P}$. We can also construct a multiplication table

	I	X	Y	Z
I	I	X	Y	Z
X	X	I	iZ	$-iY$
Y	Y	$-iZ$	I	iX
Z	Z	iY	$-iX$	I

and note that every pair of Pauli matrices anticommutes ($\sigma_i \sigma_j = -\sigma_j \sigma_i$).

Of particular interest here is the group of N-ary tensor products of these Pauli matrices, corresponding to Pauli operations on N qubits,

$$\mathcal{P}_N \equiv \left\{ \bigotimes_{k=1}^{N} \sigma \,\middle|\, \sigma \in \mathcal{P} \right\}. \tag{6}$$

For convenience in what follows, we'll suppress the tensor product in the notation for individual elements of $\mathcal{P}_N$, e.g.

$$[-XYZ] \equiv -X \otimes Y \otimes Z \in \mathcal{P}_3. \tag{7}$$

Clifford gates are another special class of unitary transformations, and are in fact a superset of the Pauli matrices. The set of all Clifford gates is characterized by the fact that they act as transformations to take any element of $\mathcal{P}_N$ to any other element of $\mathcal{P}_N$. The total set of Clifford gates is notably larger than $\mathcal{P}_N$ and contains, for example, the Hadamard and phase shift gates

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad S \equiv \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{8}$$

Although the set $\mathcal{C}_N$ of N-qubit Clifford gates is very large,

$$|\mathcal{C}_N| = 2^{N^2+2N} \prod_{j=1}^{N} (4^j - 1) \tag{9}$$

(which identifies operators that differ only by an unphysical overall scalar together), we can easily describe it as being *generated* by (tensor products of) any element of $\mathcal{P}$, H, S, $S^\dagger$, and CNOT–that is, every element of $\mathcal{C}_N$ is describable as a quantum circuit on N qubits containing some combination of these gates. Usually when we talk about arbitrary elements of $\mathcal{C}_N$, we refer to these polynomial-depth circuits; various algorithms exist to perform uniform sampling over $\mathcal{C}_N$ [1,2].

1.2 The stabilizer formalism

We are building up to the mathematical framework that underpins much of quantum error correction [3], which is called the *stabilizer formalism*. Specifically, a "stabilizer" of a given quantum state $|\psi\rangle$ is any operator U for which $|\psi\rangle$ is an eigenstate with eigenvalue equal to $+1$,

$$U|\psi\rangle = +1|\psi\rangle. \tag{10}$$

From here, we can also note that the set of all stabilizers $\text{Stab}(|\psi\rangle)$ of $|\psi\rangle$ *also* form a group since, given two stabilizers U and V, we have

$$VU|\psi\rangle = V|\psi\rangle = |\psi\rangle \tag{11}$$

making VU a stabilizer, and we fix to unitary operators only, which guarantees the existence of $(VU)^{-1} = U^{-1}V^{-1}$.

However, things get interesting when we consider that $\text{Stab}(|\psi\rangle)$ can be used to identify $|\psi\rangle$ exactly: Two states $|\psi\rangle$ and $|\varphi\rangle$ are equal if and only if $\text{Stab}(|\psi\rangle)$ and $\text{Stab}(|\varphi\rangle)$ are. More interesting still are a particular class of states that are uniquely identified by the set of stabilizers that are also elements of $\mathcal{P}_N$. These states are called *stabilizer* or sometimes *Clifford* states, and are characterized by the following theorem from Ref. [4]:

Theorem 1.
Given an N-qubit state $|\psi\rangle$, the following are equivalent:

(i) $|\psi\rangle$ can be obtained from $|0\rangle^{\otimes N}$ by CNOT, Hadamard, and S gates only.
(ii) $|\psi\rangle$ can be obtained from $|0\rangle^{\otimes N}$ by CNOT, Hadamard, S, and measurement gates only.
(iii) $|\psi\rangle$ is stabilized by exactly 2^N N-qubit Pauli operators.
(iv) $|\psi\rangle$ is uniquely determined by $S(|\psi\rangle) \equiv \text{Stab}(|\psi\rangle) \cap \mathcal{P}_N$, or the group of Pauli operators that stabilize $|\psi\rangle$.

Hence, any circuit consisting of only Clifford and measurement gates is called a stabilizer circuit, and any state obtainable by applying the circuit to $|0\rangle^{\otimes N}$ a stabilizer state.

The biggest upshot of this formalism is the Gottesman-Knill theorem [5]:

Theorem 2. (Gottesman-Knill)
Any quantum computer performing only: a) Clifford group gates, b) measurements of Pauli group operators, and c) Clifford group operations conditioned on classical bits, which may be the results of earlier measurements, can be perfectly simulated in polynomial time on a probabilistic classical computer.

A constructive, informal proof of Theorem 2 is worth noting here. First, note that there are exactly four Pauli matrices, which means that any N-qubit operator consisting of only Pauli matrices and an overall $\pm$ sign (note that no Pauli operator with a $\pm i$ phase can be a stabilizer) can be encoded in exactly $2N+1$ bits. Further, a well-known theorem in group theory says that any finite group of $|G|$ elements has a generating set of at most $\log_2|G|$ elements. In our case, stabilizer states are stabilized by exactly 2^N N-qubit Pauli operators (statement (iii) in Theorem 1), which means that the set of stabilizers of any stabilizer state can be characterized by at most $\log_2 2^N = N$ N-qubit Pauli operators. Hence, it takes at most $N \times (N+1) = \text{poly}(N)$ bits to identify a stabilizer state.

Conventionally, these bits are arranged in what's known as a *tableau*, a three-part binary matrix

$$\left(\begin{array}{ccc|ccc|c} x_{11} & \cdots & x_{1N} & z_{11} & \cdots & z_{1N} & r_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N1} & \cdots & x_{NN} & z_{N1} & \cdots & z_{NN} & r_N \end{array}\right). \tag{12}$$

The binary elements x_{ij} and z_{ij} encode the Pauli matrix P_j in the j-th qubit's position of the stabilizer state's i-th stabilizer under the mapping

$$(x_{ij}, z_{ij}) \mapsto \begin{cases} I \text{ if } x_{ij} = 0 \text{ and } z_{ij} = 0 \\ X \text{ if } x_{ij} = 1 \text{ and } z_{ij} = 0 \\ Y \text{ if } x_{ij} = 1 \text{ and } z_{ij} = 1 \\ Z \text{ if } x_{ij} = 0 \text{ and } z_{ij} = 1 \end{cases}, \tag{13}$$

and r_i encodes the overall phase of the stabilizer as $(-1)^{r_i}$ with the added constraint that all stabilizers in the tableau commute. For example, the modified 5-qubit GHZ state $|00000\rangle + i|11111\rangle$ has stabilizers and corresponding tableau as follows:

$$\begin{matrix} [- & X & X & Y & Y & Y] \\ [+ & Z & Z & I & I & I] \\ [+ & Z & I & Z & I & I] \\ [+ & Z & I & I & Z & I] \\ [+ & Z & I & I & I & Z] \end{matrix} \quad \rightarrow \quad \left(\begin{array}{ccccc|ccccc|c} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right). \tag{14}$$

From here, (Clifford) gates can be applied by performing operations between the columns of the tableau. For example [4], applying an S gate to qubit a corresponds to the tableau update rule

$$z_{ia} \rightarrow z_{ia} \oplus x_{ia}, \quad r_i \rightarrow r_i \oplus x_{ia} z_{ia} \quad \forall i, \tag{15}$$

a Hadamard gate on qubit a corresponds to

$$x_{ia} \leftrightarrow z_{ia}, \quad r_i \rightarrow r_i \oplus x_{ia} z_{ia} \quad \forall i, \tag{16}$$

and a CNOT gate with control and target qubits c and t corresponds to

$$x_{it} \rightarrow x_{it} \oplus x_{ic}, \quad z_{ic} \rightarrow z_{ic} \oplus z_{it}, \quad r_i \rightarrow r_i \oplus x_{ic} z_{it} (x_{it} \oplus z_{ic} \oplus 1) \quad \forall i, \tag{17}$$

all of which are operations that have runtime $O(N)$. Measurement, on the other hand, is more complex, essentially requiring inversion of the tableau via Gaussian elimination (which is $O(N^3)$), but can be reduced to $O(N^2)$ by storing additional, complementary generators outside the stabilizer group as shown by Ref. [4] or to even to $O(N)$ by storing the entire inverse tableau as shown by Ref. [6] –all of which is to say that the stabilizer formalism permits classical simulations of Clifford circuits that are efficient in both time and space.

It is worth noting here that tableau representations of stabilizer states are not unique; multiple distinct tableaus may correspond to the same state. However, "canonical" tableau forms can be readily constructed–such that equal states have equal canonicalized tableaus–using variations of Gaussian elimination. Here we conform to the particular canonical form implemented in the well-known package Stim [6], with algorithm reproduced below (Algorithm 1) for use by miners with custom solution strategies. The canonical form of the stabilizer generators for the modified GHZ state above is:

$$\begin{matrix} [+ & X & X & X & X & Y] \\ [+ & Z & I & I & I & Z] \\ [+ & I & Z & I & I & Z] \\ [+ & I & I & Z & I & Z] \\ [+ & I & I & I & Z & Z] \end{matrix} \quad \rightarrow \quad \left(\begin{array}{ccccc|ccccc|c} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right). \tag{18}$$

Theorem 2 is often a surprising revelation to newcomers to quantum information. Clearly, since $\mathcal{C}_N$ contains both the Hadamard and CNOT gates on arbitrary qubits, the stabilizer formalism is computationally powerful enough to model large-scale entanglement in the quantum state–yet also somehow allows for significantly sub-exponential efficient classical simulation! Ref. [4] explains that, in fact, stabilizer circuits can actually be simulated probabilistically using only NOT and CNOT gates–which, of course, are purely classical operations–alone, which helps to explain this seeming contradiction. That is, stabilizer circuits reduce quantum entanglement to mere classically probabilistic behavior. As Aaronson and Gottesman note, this provides strong evidence that stabi-

lizer circuits are actually not even universal for *classical* computation, but here the reduction in complexity is precisely what makes this regime of simulation ideal for Subnet 63.

2 Problem definition and generation

Crucially, the tableau simulation scheme relies on a clear determination that a) the quantum state being simulated is a stabilizer state, and that b) the circuit being applied to it comprises only Clifford gates and measurements. If either condition is not satisfied then this scheme cannot be applied, and one must resort to other, more general simulation methods that incur exponential costs in the number of qubits.

Our hidden stabilizers problem is designed to take advantage of this clear delineation between efficient, stabilizer-only simulation and other methods (likely state vector, matrix product state, or tensor network). The essence of the problem is this: On the validators' side, simulate random element of $\mathcal{C}_N$, note the stabilizers of the output state as special "keys," and apply a set of randomized transformations to the circuit before passing it to the miners. These transformations are selected specifically to obfuscate the Clifford-ness of the circuit such that, when it is passed to miners, it cannot be readily simulated using the stabilizer formalism. Specifically, we decompose all one-qubit gates (i.e. X, Y, Z, H, S, $S^\dagger$) into a randomized series of R_X, R_Y, and R_Z gates with arbitrary rotation angles. Two-qubit gates are treated similarly, with the exception that a typical generated Clifford circuit will only have CNOT and SWAP. We therefore decompose CNOT in the same way as an X gate and add controls, and convert SWAPs to decompositions of three CNOTs. We then apply further obfuscations by fusing the first and last gates of all decompositions operating on overlapping qubits. As an example, we provide a source random Clifford circuit along with its obfuscated version in Listings 1 and 2.

Through all this, however, the *output* of the circuit will remain unchanged–it will still be a stabilizer state with N-qubit Pauli stabilizers , which the miners are tasked with finding.

Problem 1. (Hidden stabilizers)
Given an arbitrary unitary quantum circuit C on N qubits with guaranteed stabilizer state output $|\psi\rangle = C|0\rangle^{\otimes N}$, compute the N commuting stabilizers

$$\begin{aligned} S(|\psi\rangle) &\equiv \{S_1, ..., S_N\} \subset \mathcal{P}_N, \\ [S_i, S_j] &\equiv S_i S_j - S_j S_i = 0 \end{aligned} \tag{19}$$

that generate $\mathrm{Stab}(|\psi\rangle)$ in canonical form.

3 Solution strategies

Here, we offer a few observations that may be of use to innovative miners. First, an interesting property of Clifford gates is that, given an initial stabilizer state, they rotate any individual qubit between only the six "cardinal directions" of the Bloch sphere, $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |+i\rangle, |-i\rangle\}$, otherwise known as the eigenstates of the non-identity Pauli matrices. This, combined with the fact that each of the non-identity Pauli matrices have eigenvalues ± 1, means that, for any stabilizer state $|\psi\rangle$, we have

$$\langle \sigma_k \rangle \equiv \langle \psi | \sigma_k | \psi \rangle \in \{0, \pm 1\}, \tag{20}$$

which also extends to any tensor product of Pauli matrices $S \in \mathcal{P}_N$ as well:

$$\langle S \rangle \in \{0, \pm 1\}. \tag{21}$$

This provides a more memory-efficient check that S is a stabilizer of $|\psi\rangle$ than computing the full state $S|\psi\rangle$ and comparing the full vector to $|\psi\rangle$.

Additionally, by fixing the input to the generated circuit C as the all-zero state $|0\rangle^{\otimes N}$, we create a duality between C and the output state $|\psi\rangle = C|0\rangle^{\otimes N}$. This means, in some sense, the structure of C is encoded in $|\psi\rangle$, and hence can be elucidated through various means, all of which amount to finding a circuit that prepares $|\psi\rangle$. Given a restriction to only Clifford gates, there exist simple algorithms to find preparatory Clifford circuits for stabilizer states, which can then be simulated in the stabilizer formalism to find the appropriate stabilizers of the output state–see, for example, `Tableau.from_state_vector` from Stim [6]. This specific algorithm analyzes an input state vector and eliminates amplitudes by moving them to the $|0\rangle^{\otimes N}$ element in the vector via Clifford gates. `Tableau.from_state_vector` hence runs in roughly at least $O(2^N)$ time, but is likely more efficient than a $O(4^N)$ brute-force search through all possible N-qubit Pauli operators (which does not include the additional cost of checking that all stabilizers commute!).

Finally, another possible approach to solving the problem–one that does not rely on non-tableau simulation methods–is to analyze the given circuit C. In principle, C is still a Clifford circuit beneath the obfuscations we apply, which means there must exist a series of transformations one could perform to convert it back. Given knowledge and application of this transformation, it would be possible then to simply run an ordinary tableau simulator and obtain the stabilizers of the output state for free. Finding this inverse circuit transformation is a difficult task, however, and although several promising approaches to the more general problem of circuit simplification exist [7,8], we know of no existing methods that are guaranteed to uncover the obfuscated Clifford gates in our circuits. Any clever miners who are able to do so are encouraged to publish their findings!

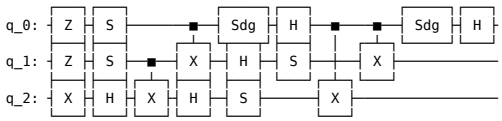
4 Difficulty scaling

Although we finely tuned scaling functions describing the number of qubits, depth, and output states of peaked circuits to allow for efficient problem generation, we are subject to significantly fewer constraints in this context. This is largely due to the power of the stabilizer formalism–approximately half of all computational constraints effectively disappear when validators can generate circuits in polynomial time while miners, absent powerful innovations, are constrained to exponential-time simulation.

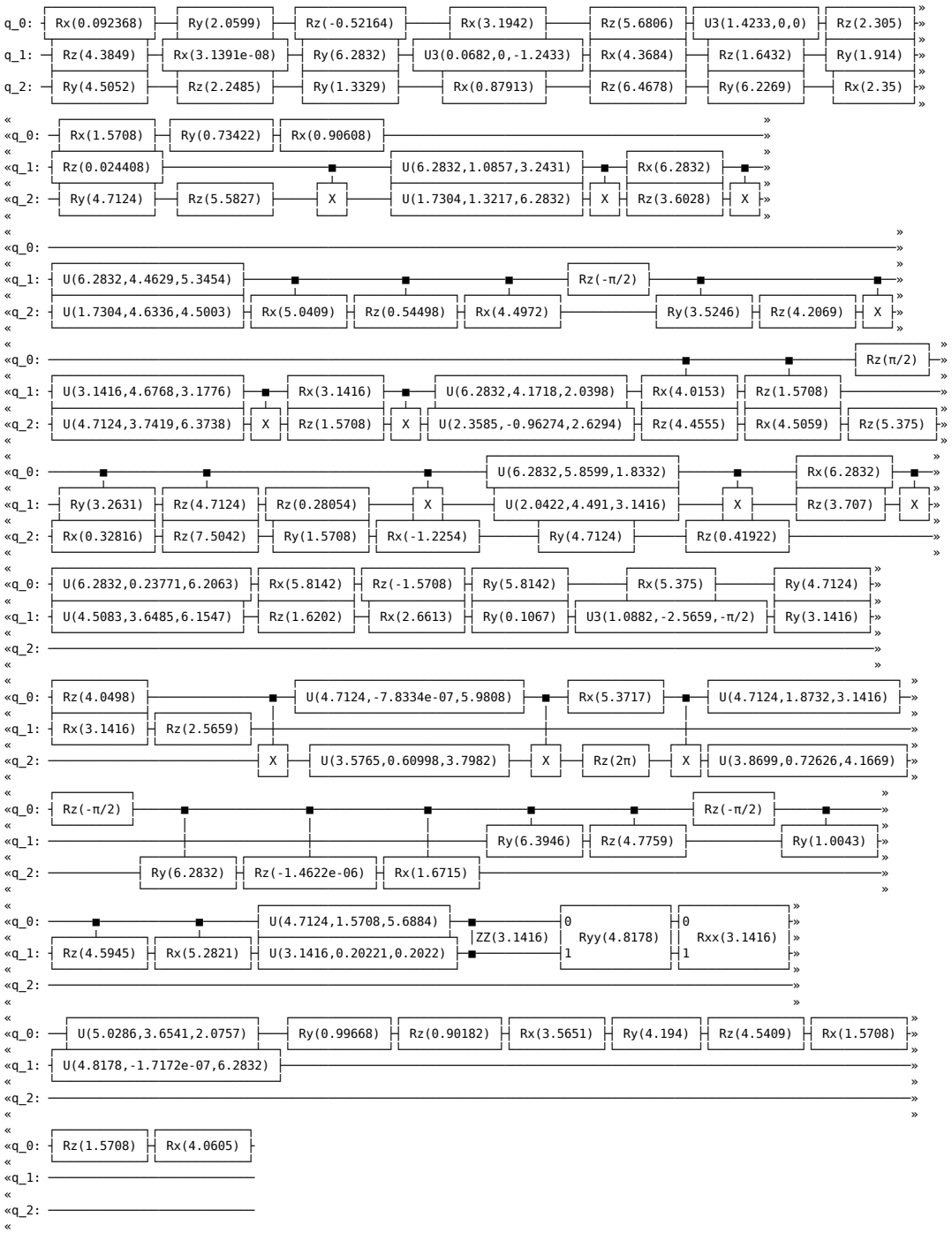
We therefore scale difficulties for this problem merely by the number of qubits N, and only describe here how the various particulars of the generated circuits scale with N. As previously noted (Equation 9), the number of physically distinct N-qubit Clifford operators is approximately double-exponential in N. For some perspective, we can note that $|\mathcal{C}_1| = 24$, while $|\mathcal{C}_2| = 11,520$, and $|\mathcal{C}_3| = 92,897,280$. However, the number of gates required to represent each of those elements scales only quadratically with N–in fact, the average number of gates happens to be quite close to exactly N^2. Our obfuscation procedure then decomposes each original Clifford gate into an average of ≈ 5.5 arbitrary rotation gates (some possibly with controls). This puts the final gate count at $\approx 5.5N^2$ and, unlike the brickwork structure of our peaked circuits, the two-qubit gates in these circuits will have arbitrary connectivity.

Example circuit

Below, we provide an example source random Clifford circuit for $N = 3$ qubits, as well as its obfuscated counterpart. Note that, while typical circuits passed to miners will have N significantly larger, we choose $N = 3$ here in order to keep the obfuscated circuit on one page!



Listing 1: A random pure-Clifford circuit on $N = 3$ qubits.



Listing 2: Obfuscated counterpart to the circuit in Listing 1.

Tableau canonicalization

CANONICALIZE
Inputs: Stabilizer tableau T with:
rows $T_i, i \in \{0, ..., N\}$, elements $x_{ij} \equiv T_{ij}$ and $z_{ij} \equiv T_{i,j+N}$ for $j \in \{0, ..., N\}$
Outputs: Canonicalized tableau T'

let $i_0 = 0$;
for $j = 0, ..., N$ **do**
  if $i_0 \geq N$ **then**
    return T;
  let $i_X \geq i_0$ be a row index s.t. $x_{i_X j} = 1$;
  if $\exists i_X$ **then**
    while $\exists i' \neq i_X$ s.t. $x_{i' j} = 1$ **do**
      $T_{i'} \leftarrow T_{i_X} T_{i'}$;
    if $i_X \neq i_0$ **then**
      swap T_{i_X}, T_{i_0};
    $i_0 \leftarrow i_0 + 1$;
  if $i_0 \geq N$ **then**
    return T;
  let $i_Z \geq i_0$ be a row index s.t. $z_{i_Z j} = 1$;
  if $\exists i_Z$ **then**
    while $\exists i' \neq i_Z$ s.t. $x_{i' j} = 1$ **do**
      $T_{i'} \leftarrow T_{i_Z} T_{i'}$;
    if $i_X \neq i_0$ **then**
      swap T_{i_Z}, T_{i_0};
    $i_0 \leftarrow i_0 + 1$;
return T;

References

[1] E. van den Berg, "A simple method for sampling random Clifford operators." 2021 IEEE International Conference on Quantum Computing and Engineering (QCE) 54 (2021).

[2] S. Bravyi, and D. Maslov, "Hadamard-free circuits expose the structure of the Clifford group." IEEE Transactions on Information Theory **67**, 4546 (2021).

[3] D. Gottesman, "Stabilizer Codes and Quantum Error Correction." arXiv:9705052 (1997).

[4] S. Aaronson, and D. Gottesman, "Improved simulation of stabilizer circuits." Phys. Rev. A **70**, 052438 (2004).

[5] D. Gottesman, "The Heisenberg Representation of Quantum Computers." Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics 32 (1999).

[6] C. Gidney, "Stim: A fast stabilizer circuit simulator." Quantum **5**, 497 (2021).

[7] A. Kissinger, and J. van de Wetering, "Reducing the number of non-Clifford gates in quantum circuits." Phys. Rev. A **102**, 022406 (2020).

[8] F. Lima, and A. C. Medina, "Clifford and Non-Clifford Splitting in Quantum Circuits: Applications and ZX-Calculus Detection Procedure." arXiv:2504.16004 (2025).