Numerical Analysis of the Collision Probability in Pseudo-Random Quantum Circuits

Matthew Khoury

Abstract

We introduce two conjectures that predict how the expected value of the collision probability in a pseudo-random quantum circuit behaves as a function of both the depth of the circuit and the number of qubits in the circuit. We then present an algorithm to compute the collision probability of a stabilizer state in polynomial time. Lastly, we show that some of our numerical results strongly agree with the predictions made by the conjectures while others only loosely agree with those predictions.

1 Introduction

Current research suggests that quantum computers may be able to outperform classical computers in certain classes of algorithms. For example, as described in [3] and [1], certain sampling tasks may be more efficient on small quantum computers than on current classical computers. One such task is to apply a set of random gates to n qubits in the state $|0\rangle^{\otimes n}$, then measure the qubits in the computational basis.

One feasible way to accomplish this task is to lay the n qubits out onto a lattice or a complete graph and then apply a polynomial number of random two qubit gates. We will refer to such a set of gates as a *pseudo-random quantum circuit*. In this paper, we will describe numerical simulations that we have used to test one of the statistical properties of these pseudo-random quantum circuits.

2 Statistical Property of Pseudo-Random Quantum Circuits

2.1 Definition of Collision Probability

Imagine we prepare two identical quantum states of n qubits represented as $|\psi\rangle \in \mathbb{C}^{2^n}$. We then measure both states in the computational basis, letting random variables M_1 and M_2 denote the outcomes of the two measurements. We then define the *collision probability* P_c as the probability of measuring the same state twice, so we have

$$P_c = P(M_1 = M_2) \tag{2.1}$$

Using the basic rules of probability along with the fact that the two measurements are independent, we can simplify P_c by writing

$$P_c = P(M_1 = M_2) \tag{2.2}$$

$$= P\big(\big\{\{M_1 = z_1\} \cap \{M_2 = z_1\}\big\} \cup \cdots \cup \big\{\{M_1 = z_{2^n}\} \cap \{M_2 = z_{2^n}\}\big\}\big) \tag{2.3}$$

$$= \sum_{z \in \mathbb{F}_2^n} P(\{M_1 = z\} \cap \{M_2 = z\}) \tag{2.4}$$

$$= \sum_{z \in \mathbb{F}_2^n} P(M_1 = z) P(M_2 = z) \tag{2.5}$$

$$= \sum_{z \in \mathbb{F}_2^n} (p(z))^2 \tag{2.6}$$

Where $p(z)$ is the probability of measuring state $|z\rangle$ from $|\psi\rangle$. We know that we can write this probability as

$$p(z) = |\langle z|\psi\rangle|^2 = \langle z|\psi\rangle \langle z|\psi\rangle^* = \langle z|\psi\rangle \langle \psi|z\rangle \tag{2.7}$$

Likewise, we can rewrite the collision probability as

$$P_c = \sum_{z \in \mathbb{F}_2^n} (p(z))^2 = \sum_{z \in \mathbb{F}_2^n} (\langle z|\psi\rangle \langle \psi|z\rangle)^2 \tag{2.8}$$

2.2 Collision Probability in Pseudo-Random Quantum Circuits

If we have applied a total of $N = O(\text{poly}(n))$ gates in a pseudo-random quantum circuit, then the *depth* d of the circuit is defined as $d \sim N/n$. The statistical property of pseudo-random quantum circuits we are interested in testing is how the collision probability changes as a function of both the depth of the circuit d and the number of qubits in the circuit n.

We note that in a pseudo-random quantum circuit, the final quantum state of the circuit can be modeled as a random variable. Likewise, the collision probability in a pseudo-random quantum circuit is also a random variable. With that said, we would like test the following conjectures.

Conjecture 1. The collision probability in a pseudo-random quantum circuit arranged in a w-dimensional lattice will have an expected value

$$\langle P_c \rangle = \frac{2^{O(n/d^w)}}{2^n} \tag{2.9}$$

Equivalently, we will have

$$-\log_2(\langle P_c \rangle) = n - O\left(\frac{n}{d^w}\right) = n\left(1 - O\left(\frac{1}{d^w}\right)\right) \tag{2.10}$$

We also say that the collision probability is *saturated* once $-\log_2(\langle P_c \rangle) = n-1$. Thus, the conjecture also predicts that collision probability saturates when

$$O\left(\frac{1}{d^w}\right) = \frac{1}{n} \quad \Rightarrow \quad d \sim n^{1/w} \tag{2.11}$$

Conjecture 2. The collision probability in a pseudo-random quantum circuit arranged in a complete graph will have an expected value

$$\langle P_c \rangle = \frac{2^{O(n/e^{N/n})}}{2^n} \tag{2.12}$$

Similarly, we will have

$$-\log_2(\langle P_c \rangle) = n - O\left(\frac{n}{e^{N/n}}\right) = n\left(1 - O\left(\frac{1}{e^{N/n}}\right)\right) \tag{2.13}$$

And the circuit will be saturated when

$$O\left(\frac{1}{e^{N/n}}\right) = \frac{1}{n} \quad \Rightarrow \quad N \sim n\ln(n) \tag{2.14}$$

2.3 Underlying Intuition

The intuition underlying Conjectures 1 and 2 is as follows. We start with the idea that in a w-dimensional lattice, we expect a qubit to be able to interact with $O(d^w)$ of its surrounding qubits after the circuit reaches a depth of d. Likewise, we expect that after the circuit has reached a depth d, there will be clusters of Haar Random qubits of size $O(d^w)$[1]. Moreover, there will be $n/O(d^w)$ of these clusters and we expect these clusters to be independent of one another. As a result, the expected collision probability in a w-dimensional lattice will be

$$\langle P_c \rangle = \left(\frac{2}{2^{O(d^w)}}\right)^{n/O(d^w)} = \frac{2^{O(n/d^w)}}{2^n} \tag{2.15}$$

To generalize, we would expect that after a depth d, we will have $n/O(f(d))$ independent clusters of size $O(f(d))$ that are each Haar Random. We also expect that the function $f(d)$ will be determined by the geometry of the circuit. Likewise, the intuition behind Conjecture 2 would be to set $f(d) = e^d = e^{N/n}$, as we would expect the cluster sizes in a complete graph to be much larger than those in a lattice.

3 The Stabilizer Formalism

3.1 Motivation

In general, simulating a pseudo-random quantum circuit on a classical computer will take both exponentially large storage and an exponentially large runtime. Thus, in order to simulate a pseudo-random quantum circuit to numerically test Conjectures 1 and 2, we make use of the Gottesman-Knill Theorem.

The Gottesman-Knill Theorem states that a quantum circuit composed only of Clifford Gates can be efficiently simulated on a classical computer. In this section, we review some of the mathematical principles underlying the Gottesman-Knill Theorem, as they will be necessary for explaining how our simulations work. Many concepts in this section can be found more thoroughly described in Chapter 10.5 of [6].

[1] A Haar Random quantum circuit of n qubits will have an expected collision probability of $2/2^n$.

3.2 The Pauli Group

We let G_n denote the Pauli Group on n qubits. Formally, we define

$$G_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\} \tag{3.1}$$

Where

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.2}$$

Likewise, the set G_n consists of all the n-fold tensor product of the matrices in G_1. We see that the Pauli Group is closed under matrix multiplication, as these matrices satisfy

$$X^2 = Y^2 = Z^2 = I \tag{3.3}$$
$$XY = iZ \quad YX = -iZ \tag{3.4}$$
$$YZ = iX \quad ZY = -iX \tag{3.5}$$
$$ZX = iY \quad XZ = -iY \tag{3.6}$$

From this we also see that all of the elements in G_n either commute or anti-commute. We also note that any $g \in G_n$ can be written as

$$g = (-1)^a \; i^b \; g_1 \otimes g_2 \otimes \cdots \otimes g_n \quad \text{for} \quad a, b \in \mathbb{F}_2 \quad \text{and} \quad g_j \in \{I, X, Y, Z\} \tag{3.7}$$

3.3 Isomorphism to $\mathbb{F}_2$

We use $\mathbb{F}_2$ to refer to the finite field $\mathbb{F}_2 \in \{0, 1\}$. All operations are done modulo 2, so we will have $1 + 1 = 0$ and $x = -x$. We can also see that $\mathbb{F}_2$ defines a group closed under addition modulo 2.

We define the map $M : \mathbb{F}_2^2 \to \{I, X, Y, Z\}$ as

$$00 \to I, \quad 01 \to Z, \quad 10 \to X, \quad 11 \to Y \tag{3.8}$$

Likewise, from 3.7, if we let $u = (u_s, u_i, u_{x_1}, \ldots, u_{x_n}, u_{z_1}, \ldots, u_{z_n}) \in \mathbb{F}_2^{2n+2}$, we can write any element $g \in G_n$ as

$$g = (-1)^{u_s} \; i^{u_i} \; M(u_{x_1}, u_{z_1}) \otimes \cdots \otimes M(u_{x_n}, u_{z_n}) \tag{3.9}$$

In shorthand, we will write that $v = (v_{x_1}, \ldots, v_{x_n}, v_{z_1}, \ldots, v_{z_n}) = (v_x, v_z) \in \mathbb{F}_2^{2n}$ and that

$$M(v) = M(v_x, v_z) = M(v_{x_1}, v_{z_1}) \otimes \cdots \otimes M(v_{x_n}, v_{z_n}) \tag{3.10}$$

We also let $[A]$ be the set of operators that are the same up to a global phase so that

$$[A] = \{\beta A : \beta \in \mathbb{C}, |\beta| = 1\} \tag{3.11}$$

Using 3.3 – 3.6, we can see that the map M induces an isomorphism $[M] : \mathbb{F}_2^{2n} \to [G_n]$ because addition of vectors in $\mathbb{F}_2^{2n}$ corresponds to the multiplication of matrices in G_n up to a global phase

$$[M(u + v)] = [M(u)][M(v)] \quad \text{for} \quad u, v \in \mathbb{F}_2^{2n} \tag{3.12}$$

3.4 The Stabilizer Group

A quantum state $|\psi\rangle$ is said to be *stabilized* by a unitary operator g if and only if $g|\psi\rangle = |\psi\rangle$. The *stabilizer group* S is a subgroup of G_n, which we denote as $S \leq G_n$. We then define a *stabilizer subspace* V_S to be the set of states that is stabilized by S, or more formally

$$V_S = \{|\psi\rangle : g|\psi\rangle = |\psi\rangle, \forall g \in S\} \tag{3.13}$$

Here we note that if $g, h \in S$, then we must also have $gh \in S$ and $g^{-1} \in S$ because

$$gh|\psi\rangle = g|\psi\rangle = |\psi\rangle \quad \text{and} \quad |\psi\rangle = g^{-1}g|\psi\rangle = g^{-1}|\psi\rangle \tag{3.14}$$

We also note that if we would like V_S to be a non-trivial vector space[2], S must satisfy two conditions

(1) The elements of S must commute

(2) $-I$ cannot be an element of S

We see that condition (1) must hold because if $-I \in S$, then we will have that $-I|\psi\rangle = |\psi\rangle$, which can only be satisfied by $|\psi\rangle = 0$. We see that condition (2) must also hold because all of the elements in S either commute or anti-commute, as $S \leq P_n$. This means that if two elements $g, h \in S$ do not commute, then they must anti-commute so that $gh = -hg$. But if this is the case, then $-|\psi\rangle = -hg|\psi\rangle = gh|\psi\rangle = |\psi\rangle$, which can also only be satisfied by $|\psi\rangle = 0$.

Moreover, if $g \in S$, then g cannot have an overall phase of $\pm i$. This follows because if we let

$$g = \pm i\ g_1 \otimes g_2 \otimes \cdots \otimes g_n \quad \text{for} \quad g_j \in \{I, X, Y, Z\} \tag{3.15}$$

Then using 3.3 and 3.14, we see that

$$|\psi\rangle = g|\psi\rangle = g^2|\psi\rangle = (\pm i)^2 I^{\otimes n}|\psi\rangle = -|\psi\rangle \tag{3.16}$$

Again, this is only satisfied when $|\psi\rangle = 0$. Thus, if S stabilizes a non-trivial vector space, then we can write any element $g \in S$ as

$$g = \pm 1\ g_1 \otimes g_2 \otimes \cdots \otimes g_n \quad \text{for} \quad g_j \in \{I, X, Y, Z\} \tag{3.17}$$

In terms of the $\mathbb{F}_2$ linear algebra perspective, this means that we can represent any $g \in S$ as a vector $u = (u_s, u_x, u_z) \in \mathbb{F}_2^{2n+1}$ where using the shorthand notation in 3.10 we have

$$g = (-1)^{u_s} M(u_x, u_z) \tag{3.18}$$

3.5 Clifford Gates

It has been shown in [6] and [2] that we can actually make use of the $\mathbb{F}_2$ linear algebra perspective to efficiently simulate quantum circuits composed of the following gates.

$$H(\alpha|0\rangle + \beta|1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle \tag{3.19}$$

$$P(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + i\beta|1\rangle \tag{3.20}$$

[2]The trivial vector space is $V_s = \{0\}$

$$C_{a,b}\left(|a\rangle \otimes |b\rangle\right) = |a\rangle \otimes |a+b\rangle \quad \text{for} \quad a, b \in \mathbb{F}_2 \tag{3.21}$$

Any quantum circuit that can be decomposed into these three gates is said to only contain *Clifford Gates*. In order to simulate a quantum circuit composed only of Clifford Gates, we keep track of the generators of the stabilizer group of $|\psi\rangle$, which we initialize as $|0\rangle^{\otimes n}$. We then update the generators as described in [2] each time we apply one of the gates above. Moreover, it has been shown that a state $|\psi\rangle$ created by applying Clifford Gates to $|0\rangle^{\otimes n}$ is uniquely determined by a stabilizer group $S \leq P_n$ that contains 2^n elements. Likewise, we only need a total of n generators, or vectors in $\mathbb{F}_2^{2n+1}$, in order to uniquely determine such a state.[3]

3.6 Projectors

If $W \subseteq \mathbb{C}^{2^n}$ is a w-dimensional vector subspace, then a *projector* Π_W onto the vector subspace W is a linear operator that satisfies two conditions

(1) For all $|\phi\rangle \in \mathbb{C}^{2^n}$, $\Pi_W |\phi\rangle \in W$

(2) $\Pi_W^2 = \Pi_W$

Likewise, if W has an orthonormal basis $|1\rangle, |2\rangle, \ldots, |w\rangle$, then

$$\Pi_W = \sum_{j=1}^{w} |j\rangle \langle j| \tag{3.22}$$

Claim 1. *Π_S given by*

$$\Pi_S = \frac{1}{|S|} \sum_{g \in S} g \tag{3.23}$$

is a projector onto the stabilizer subspace V_S.

Proof. First we note that if $g \in S$, then using 3.14

$$g\Pi_S = g\frac{1}{|S|} \sum_{h \in S} h = \frac{1}{|S|} \sum_{h \in S} gh = \frac{1}{|S|} \sum_{h'=gh \in S} h' = \Pi_s \tag{3.24}$$

From this, we see that Π_S satisfies condition (1) because for all $|\phi\rangle \in \mathbb{C}^{2^n}$ and for all $g \in S$ we have $g\Pi_S |\phi\rangle = \Pi_S |\phi\rangle$. By definition, this means that $\Pi_S |\phi\rangle \in V_S$. We also see that Π_S satisfies condition (2) because

$$\Pi_S^2 = \left(\frac{1}{|S|} \sum_{g \in S} g\right) \Pi_S = \frac{1}{|S|} \sum_{g \in S} g\Pi_S = \frac{1}{|S|} \sum_{g \in S} \Pi_S = \frac{1}{|S|} \; |S| \; \Pi_S = \Pi_S \tag{3.25}$$

□

Here we also note that if V_S contains only one vector so that $V_S = \{|\psi\rangle\}$, then combining 3.22 and 3.23, we see that

$$\Pi_S = \frac{1}{|S|} \sum_{g \in S} g = |\psi\rangle \langle\psi| \tag{3.26}$$

[3] If G is a group closed under multiplication, then a set of elements $g_1, \ldots, g_l \in G$ is said to generate a group G if every element in G can be written as a product of the elements in $g_1, \ldots, g_l$. Moreover, it is a known fact from Group Theory that a finite group G of size $|G|$ has a set of at most $\log_2(|G|)$ generators.

4 Efficiently Computing Collision Probability

4.1 Preliminaries

As described in Section 3.5, we can efficiently simulate pseudo-random quantum circuits composed only of Clifford Gates by keeping track of the n generators of the stabilizer group S that uniquely determine a stabilizer state $V_S = \{|\psi\rangle\}$. Moreover, keeping track of these n generators is equivalent to keeping track of n vectors in $\mathbb{F}_2^{2n+1}$. In this section, we will present an algorithm to compute the collision probability of a stabilizer state $V_S = \{|\psi\rangle\}$ given the n vectors in $\mathbb{F}_2^{2n+1}$ corresponding to the n generators of S.

Throughout this section we will always let $|\psi\rangle$ be a state that has been created by applying a set of Clifford Gates to the initial state $|0\rangle^{\otimes n}$. Equivalently, we let $|\psi\rangle$ be the state uniquely determined by the stabilizer group S so that $V_S = \{|\psi\rangle\}$. Moreover, we let $S = \langle g^{(1)}, \dots, g^{(n)} \rangle$, meaning S is generated by the n matrices $g^{(1)}, \dots, g^{(n)}$. For every element $g \in S$, we also let $u = (u_s, u_x, u_z) \in \mathbb{F}_2^{2n+1}$ be the vector representation of g as written in 3.18.

By construction, we can use the fact that $|S| = 2^n$ along with 3.26 to write

$$|\psi\rangle\langle\psi| = \Pi_S = \frac{1}{2^n} \sum_{g \in S} g \tag{4.1}$$

Here, we also note some properties of the trace of a matrix. Specifically for a vector $|\phi\rangle$ and matrices A, B of the appropriate shape, we will have

$$\mathrm{tr}(A\,|\phi\rangle\langle\phi|) = \langle\phi|A|\phi\rangle \tag{4.2}$$

$$\mathrm{tr}(A + B) = \mathrm{tr}(A) + \mathrm{tr}(B) \tag{4.3}$$

$$\mathrm{tr}(A \otimes B) = \mathrm{tr}(A)\mathrm{tr}(B) \tag{4.4}$$

The last tool we need is the following:

Claim 2. *If $a \in \mathbb{F}_2^n$ and we let $Z^a = Z^{a_1} \otimes \cdots \otimes Z^{a_n}$, then*

$$\sum_{z \in \mathbb{F}_2^n} |z\rangle\langle z| \otimes |z\rangle\langle z| = \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} Z^a \otimes Z^a \tag{4.5}$$

Proof. First, we note that

$$|0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \frac{I + Z}{2} \qquad |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \frac{I - Z}{2} \tag{4.6}$$

As a result, we can write that for any $z \in \mathbb{F}_2^n$

$$|z\rangle\langle z| = |z_1\rangle\langle z_1| \otimes \cdots \otimes |z_n\rangle\langle z_n| \tag{4.7}$$

$$= \frac{I + (-1)^{z_1} Z}{2} \otimes \cdots \otimes \frac{I + (-1)^{z_n} Z}{2} \tag{4.8}$$

$$= \frac{1}{2^n} \Big(I \otimes I \otimes \cdots \otimes I \;+\; (-1)^{z_1} Z \otimes I \otimes \ldots \otimes I \;+\; I \otimes (-1)^{z_2} Z \otimes \cdots \otimes I \\ + \;\cdots\; + \; (-1)^{z_1} Z \otimes (-1)^{z_2} Z \otimes \cdots \otimes (-1)^{z_n} Z \Big) \tag{4.9}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} (-1)^{\langle a,z \rangle} Z^a \tag{4.10}$$

Second, we also note that for any $w \in \mathbb{F}_2^n$

$$\sum_{z \in \mathbb{F}_2^n} (-1)^{\langle w,z \rangle} = 2^n \ \delta_{w=0} \tag{4.11}$$

We see this is the case whenever $w = 0$ because $\langle 0, z \rangle = 0$ for all values of z.[4] However, if w has a 1 in one or more entries, then for every z such that $\langle w, z \rangle = 1$, there will be a z' such that $\langle w, z' \rangle = 0$ that can be constructed by flipping one bit in z that is in the same entry as a 1 in w. Likewise, it follows that half of the values in the sum will evaluate to (-1) while the other half evaluate to 1, meaning the whole sum is just 0.

Using 4.10 and 4.11, we can now prove the original claim by writing

$$\sum_{z \in \mathbb{F}_2^n} |z\rangle \langle z| \otimes |z\rangle \langle z| = \sum_{z \in \mathbb{F}_2^n} \left(\frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} (-1)^{\langle a,z \rangle} Z^a \right) \ \otimes \ \left(\frac{1}{2^n} \sum_{b \in \mathbb{F}_2^n} (-1)^{\langle b,z \rangle} Z^b \right) \tag{4.12}$$

$$= \frac{1}{4^n} \sum_{a \in \mathbb{F}_2^n} \sum_{b \in \mathbb{F}_2^n} \sum_{z \in \mathbb{F}_2^n} (-1)^{\langle a,z \rangle + \langle b,z \rangle} \ Z^a \otimes Z^b \tag{4.13}$$

$$= \frac{1}{4^n} \sum_{a \in \mathbb{F}_2^n} \sum_{b \in \mathbb{F}_2^n} \left(\sum_{z \in \mathbb{F}_2^n} (-1)^{\langle a+b,z \rangle} \right) \ Z^a \otimes Z^b \tag{4.14}$$

$$= \frac{1}{4^n} \sum_{a \in \mathbb{F}_2^n} \sum_{b \in \mathbb{F}_2^n} (2^n \ \delta_{a+b=0}) \ Z^a \otimes Z^b \tag{4.15}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \sum_{b \in \mathbb{F}_2.} \delta_{a=b} \ Z^a \otimes Z^b \tag{4.16}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} Z^a \otimes Z^a \tag{4.17}$$

□

4.2 Simplifying Collision Probability

We now use 4.1 – 4.5 to simplify the expression for P_c given in 2.8 by writing

$$P_c = \sum_{z \in \mathbb{F}_2^n} \left(\langle z|\psi\rangle \langle \psi|z\rangle \right)^2 \tag{4.18}$$

$$= \sum_{z \in \mathbb{F}_2^n} \left(\langle z|\Pi_S|z\rangle \right)^2 \tag{4.19}$$

$$= \sum_{z \in \mathbb{F}_2^n} \left(\mathrm{tr} \left(\Pi_S \, |z\rangle \langle z| \right) \right)^2 \tag{4.20}$$

[4] In these notes, the function δ_v evaluates to 1 if v is true and evaluates to 0 if v is false

$$= \sum_{z \in \mathbb{F}_2^n} \text{tr}\left((\Pi_S \otimes \Pi_S) \left(|z\rangle \langle z| \otimes |z\rangle \langle z| \right) \right) \tag{4.21}$$

$$= \text{tr}\left((\Pi_S \otimes \Pi_S) \sum_{z \in \mathbb{F}_2^n} |z\rangle \langle z| \otimes |z\rangle \langle z| \right) \tag{4.22}$$

$$= \frac{1}{2^n} \text{tr}\left((\Pi_S \otimes \Pi_S) \sum_{a \in \mathbb{F}_2^n} Z^a \otimes Z^a \right) \tag{4.23}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \text{tr}\left((\Pi_S \otimes \Pi_S) \left(Z^a \otimes Z^a \right) \right) \tag{4.24}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\text{tr}\left(\Pi_S \; Z^a \right) \right)^2 \tag{4.25}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\text{tr}\left(\frac{1}{2^n} \sum_{g \in S} g \; Z^a \right) \right)^2 \tag{4.26}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\frac{1}{2^n} \sum_{g \in S} \text{tr}\left(g \; Z^a \right) \right)^2 \tag{4.27}$$

We now find an expression for the trace in 4.27 by writing that for any $g \in S$ and $a \in \mathbb{F}_2^n$ we have

$$\text{tr}\left(g \; Z^a \right) = \text{tr}\left((-1)^{u_s} M(u_x, u_z) Z^a \right) \tag{4.28}$$

$$= (-1)^{u_s} \prod_{j=1}^{n} \text{tr}\left(M(u_{x_j}, u_{z_j}) Z^{a_j} \right) \tag{4.29}$$

$$= (-1)^{u_s} \prod_{j=1}^{n} (2 \; \delta_{u_{x_j}=0} \; \delta_{u_{z_j}=a_j}) \tag{4.30}$$

$$= (-1)^{u_s} \; (2^n) \; \delta_{u_x=0} \; \delta_{u_z=a} \tag{4.31}$$

Where in 4.30 we have used that the trace in 4.29 is 2 only when the inner term evaluates to I. Otherwise, the trace in 4.29 is just 0.

We now combine 4.27 with 4.31 to write

$$P_c = \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\frac{1}{2^n} \sum_{g \in S} (-1)^{u_s} \; (2^n) \; \delta_{u_x=0} \; \delta_{u_z=a} \right)^2 \tag{4.32}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\sum_{g \in S} (-1)^{u_s} \; \delta_{u_x=0} \; \delta_{u_z=a} \right)^2 \tag{4.33}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\delta_{(0,0,a) \in S} - \delta_{(1,0,a) \in S} \right)^2 \tag{4.34}$$

Where in 4.34 we use the shorthand notation $(u_s, u_x, u_z) \in S$ to mean that the matrix $g = (-1)^{u_s} M(u_x, u_z) \in S$. We now note that we cannot have both $(0, u_x, u_z) \in S$ and $(1, u_x, u_z) \in S$ because $V_S = \{|\psi\rangle\}$, which is a non-trivial vector space. We see this is true because if both

$(0, u_x, u_z) \in S$ and $(1, u_x, u_z) \in S$, then both $g \in S$ and $-g \in S$. As a result this would require that

$$|\psi\rangle = g\,|\psi\rangle = -g\,|\psi\rangle = -\,|\psi\rangle \tag{4.35}$$

And this can only be satisfied by the trivial vector space. Thus, we can further simplify 4.34 by writing

$$P_c = \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \left(\pm\delta_{(0,0,a)\in S \text{ or } (1,0,a)\in S}\right)^2 \tag{4.36}$$

$$= \frac{1}{2^n} \sum_{a \in \mathbb{F}_2^n} \delta_{(0,0,a)\in S \text{ or } (1,0,a)\in S} \tag{4.37}$$

$$= \frac{|\{(u_s, u_x, u_z) \in S : u_x = 0\}|}{2^n} \tag{4.38}$$

We now define U, V, W as

$$U = \{(u_x, u_z) \in \mathbb{F}_2^{2n} : (0, u_x, u_z) \in S \text{ or } (1, u_x, u_z) \in S\} \tag{4.39}$$

$$V = \{(v_x, v_z) \in \mathbb{F}_2^{2n} : v_x = 0\} \tag{4.40}$$

$$W = U \cap V \tag{4.41}$$

Again, because only $(0, u_x, u_z) \in S$ or $(1, u_x, u_z) \in S$, we will have that $|U| = |S|$. Moreover, it has been shown in Chapter 10.5 of [6] that the n vectors $u^{(1)}, \ldots, u^{(n)} \in U$ corresponding to the n generators $g^{(1)}, \ldots, g^{(n)}$ of S are linearly independent. Using this along with 3.12, we see that taking linear combinations of basis vectors in U is isomorphic to matrix multiplying the generators of S. Thus, U, V, W are actually all subspaces of $\mathbb{F}_2^{2n}$, which we denote as $U, V, W \subseteq \mathbb{F}_2^{2n}$.

If we let $\dim(W)$ be the number of basis vectors of W, then we see that $|W| = 2^{\dim(W)}$, meaning we can write

$$P_c = \frac{|W|}{2^n} = \frac{2^{\dim(W)}}{2^n} = \frac{1}{2^{n-\dim(W)}} \tag{4.42}$$

4.3 An Algorithm to Efficiently Compute Collision Probability

Assume we are given a set of basis vectors for the subspace U, which we denote as $u^{(1)}, \ldots, u^{(n)}$. To re-iterate, these n basis vectors correspond to the n generators $g^{(1)}, \ldots, g^{(n)}$ of the stabilizer group S that uniquely determines the stabilizer state $V_S = |\psi\rangle$ so that $g^{(j)} = \pm M(u_x^{(j)}, u_z^{(j)})$. We now provide an algorithm to efficiently compute the collision probability of the state $|\psi\rangle$

Algorithm 1. *Given the vectors $u^{(1)}, \ldots, u^{(n)}$, do the following*

(1) Define the following vectors where each $v^{(j)} \in \mathbb{F}_2^{2n}$

$$v^{(1)} = (0, \ldots, 0, 1, 0, \ldots, 0) \tag{4.43}$$

$$v^{(2)} = (0, \ldots, 0, 0, 1, \ldots, 0) \tag{4.44}$$

$$\vdots$$

$$v^{(n)} = (0, \ldots, 0, 0, 0, \ldots, 1) \tag{4.45}$$

Note that these form a basis for the vector space V as defined in 4.40 while $u^{(1)}, \ldots, u^{(n)}$ form a basis for the vector space U as defined in 4.39

(2) Define the following matrices $A, B \in \mathbb{F}_2^{2n \times n}$ *and* $C \in \mathbb{F}_2^{2n \times 2n}$

$$A = \begin{bmatrix} u^{(1)^T} & u^{(2)^T} & \dots & u^{(n)^T} \end{bmatrix} \tag{4.46}$$

$$B = \begin{bmatrix} v^{(1)^T} & v^{(2)^T} & \dots & v^{(n)^T} \end{bmatrix} \tag{4.47}$$

$$C = \begin{bmatrix} A & B \end{bmatrix} \tag{4.48}$$

(3) Let r *be the number of pivot columns in REF*(C)*. We then have* $dim(W) = dim(Ker(C)) = 2n - r$*, so return*

$$P_c = \frac{1}{2^{n-dim(W)}} = \frac{1}{2^{n-(2n-r)}} = \frac{1}{2^{r-n}} \tag{4.49}$$

Proof. As shown in Section 4.2, the column vectors of matrices A and B are the basis vectors of the respective subspaces U and V. Moreover, the expression for the collision probability in 4.49 was also derived in section 4.2. Likewise, the only thing left to justify is the claim that $\dim(W) = \dim(\text{Ker}(C)) = 2n - r$.

To see this, we note that the Kernel of $C = \begin{bmatrix} A & B \end{bmatrix}$ will be given by the vectors $(x, y) = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) \in \mathbb{F}_2^{2n}$ such that

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x^T \\ y^T \end{bmatrix} = 0 \tag{4.50}$$

Thus, all vectors $(x, y)^T \in \text{Ker}(C)$ will have

$$Ax^T + By^T = 0 \quad \Rightarrow \quad Ax^T = By^T \quad \Rightarrow \quad xA^T = yB^T \tag{4.51}$$

Where we have used that $y^T = -y^T$ when working in $\mathbb{F}_2^n$. Equivalently, this means that every vector in the $\text{Ker}(C)$ forms a simultaneous linear combination of the basis vectors in U and V because expanding 4.51 gives us

$$x_1 u^{(1)} + x_2 u^{(2)} + \dots + x_n u^{(n)} = y_1 v^{(1)} + y_2 v^{(2)} + \dots + y_n v^{(n)} \tag{4.52}$$

Likewise, we see that by definition $(x, y)^T \in \text{Ker}(C)$ if and only if $xA^T = yB^T \in U \cap V$. We also see that if $(x^{(1)}, y^{(1)})^T, (x^{(2)}, y^{(2)})^T \in \text{Ker}(C)$, then

$$\alpha(Ax^{(1)^T} + By^{(1)^T}) + \beta(Ax^{(2)^T} + By^{(2)^T}) = 0 \tag{4.53}$$

And we can rewrite this as

$$(\alpha x^{(1)} + \beta x^{(2)})A^T = (\alpha y^{(1)} + \beta y^{(2)})B^T \tag{4.54}$$

From this we see that taking linear combinations of vectors in $\text{Ker}(C)$ is equivalent to taking linear combinations of vectors in $U \cap V = W$. Thus, the basis vectors of the $\text{Ker}(C)$ will also form a basis for the vectors in W. As a result, we will have

$$\dim(\text{Ker}(C)) = \dim(W) \tag{4.55}$$

Lastly, by the Rank-Nullity Theorem, for the $2n \times 2n$ matrix C, we will have

$$\dim(\text{Im}(C)) + \dim(\text{Ker}(C)) = 2n \tag{4.56}$$

In step (3) of the algorithm, we set r equal to the number of pivot columns in $\text{REF}(C)$, which is the same as the $\dim(\text{Im}(C))$. Thus, combining 4.55 and 4.56 we see

$$\dim(W) = \dim(\text{Ker}(C)) = 2n - \dim(\text{Im}(C)) = 2n - r \tag{4.57}$$

□

5 Simulations and Numerical Results

5.1 Simulation Protocol

As described in Section 2.2, we are specifically interested in the relationship the expected value of the collision probability as a function of the depth of the circuit and the number of qubits in the circuit. In order to explore this relationship, we simulate pseudo-random quantum circuits in 1D, 2D, and 3D lattices as well as in a complete graph. Likewise, we restrict our simulations to stabilizer states that are created by applying Clifford Gates to the computational basis $|0\rangle^{\otimes n}$, as described in Section 3.5

Our simulations use a tableau as described in [2] to keep track of the vector representations of the generators of the stabilizer group S of the stabilizer state $V_S = \{|\psi\rangle\}$. However, we do not keep track of the phase bits, as they will not be necessary in order to compute the collision probability of the state $|\psi\rangle$. See [2] for details on how to initialize and update the tableau.

In order to generate random Clifford Gates, we use the algorithm described in [4]. Again, we ignore the phase bits as they will not affect the collision probability. Likewise, in order to generate a random Clifford Gate that acts on two qubits, we can equivalently generate a 4×4 Symplectic Matrix. We then decompose these Symplectic Matrices into Clifford Gates using the algorithm described in [2].

Lastly, we use Algorithm 1 in order to efficiently compute the collision probability of the stabilizer state $|\psi\rangle$ using the tableau that keeps track of the generators of the stabilizer group.

Using these algorithms, all of our simulations use the following protocol to collect data in order to test Conjectures 1 and 2.

Protocol 1. *Our simulations perform the following steps for a fixed number of qubits n and fixed number of samples m*

(1) For a w-dimensional lattice, let $x_{\max} = O(n^{1/w})$ and for a complete graph let $x_{\max} = O(n\ln(n))$. Initialize a vector of evenly spaced integers $x = [0, \ldots, x_{\max}]$ which corresponds to values of depth d in a lattice or number of gates N in a complete graph. Initialize a matrix K of all zeros and size $m \times len(x)$. Initialize a counter $i = 1$.

(2) Initialize a tableau to keep track of the generators of our stabilizer state $|\psi\rangle$. This tableau corresponds to the state $|0\rangle^{\otimes n}$. For a lattice, initialize $d = 0$ and for a complete graph initialize $N = 0$. Initialize a counter $j = 2$.

(3) Based on the geometry of the circuit, create a list L that contains pairs of qubits. In a lattice, we construct L so that after applying two qubit gates to all pairs of qubits in L, the depth of the circuit will increase by one. In a complete graph, L only contains one pair of qubits.

(4) For every pair of qubits in L, do the following. Pick a uniformly random 4×4 Symplectic Matrix and decompose it into Clifford Gates. Apply these two qubit gates to the pair of qubits by appropriately modifying the tableau.

(5) Update the values $d \leftarrow d + 1$ for a lattice and $N \leftarrow N + 1$ for a complete graph.

(6) If $x[j] = d$ for a lattice or $x[j] = N$ for a complete graph, do the following. Compute the collision probability of the state using the tableau. Update the matrix so that $K[i, j] \leftarrow$

$-\log_2(P_c)$ *and update the value* $j \leftarrow j+1$.

(7) Repeat steps (2) – (6) until $j = len(x)+1$. *At this point the* i^{th} *row of* K *is complete, so update* $i \leftarrow i+1$.

(8) Repeat steps (2) – (7) until $i = m+1$. *At this point, we are finished collecting data for matrix* K *for a fixed value of* n.

We have written code in Python 3.5 to implement the steps in Protocol 1, which can be found in [5]. This code also includes implementations of the algorithms described in [2] and [4] as well as an implementation of Algorithm 1. In order to enhance performance, our code includes some methods to parallelize some of the steps in Protocol 1. This implementation also precomputes the decomposition of all 4×4 Symplectic Matrices in order to speed up the simulations.

Our code also uses the following protocol to process the data for a given matrix K.

Protocol 2. *Given a matrix* K, *perform the following protocol for each column in the matrix*

(1) Initialize a value s *such that* m *is an integer multiple of* s.

(2) Exponentiate every data point such that if k *is an element of* K, *we have* $k \leftarrow (1/2)^k$

(3) Divide the m *exponentiated data points in the column vector into* m/s *sets of size* s.

(4) Let μ *and* σ *be vectors of size* m/s. *Store values in the vectors such that* $\mu[i]$ *and* $\sigma[i]$ *are the mean and standard deviation of the exponentiated data points in the* i^{th} *set of size* s.

(5) Store our estimates for the expected value of the collision probability so that $\langle P_c \rangle = mean(\mu)$ *and error* $Err(\langle P_c \rangle) = std(\mu)$

(6) Store our estimates for the standard deviation of the collision probability so that $\Delta P_c = mean(\sigma)$ *and error* $Err(\Delta P_c) = std(\sigma)$.

(7) Convert these points to a logarithmic scale so our estimates are $-\log_2(\langle P_c \rangle)$ *with error* $Err(-\log_2(\langle P_c \rangle)) = Err(P_c)/(\langle P_c \rangle \ln(2))$ *and* $-\log_2(\Delta P_c)$ *with error* $Err(-\log_2(\Delta P_c)) = Err(\Delta P_c)/(\Delta P_c \ln(2))$

Our code in [5] also implements Protocol 2 in Python 3.5. In doing so, we worked in the logarithmic domain in order to avoid underflow errors, meaning the exponentiation occurred in the last steps rather than the first steps.

In the following sections, we describe how to obtain L for 1D, 2D, and 3D lattices as well as for a complete graph. We also provide some of our numerical results obtained using our implementation of Protocols 1 and 2 on 1D, 2D, and 3D lattices as well as for a complete graph. Here, we note that all of our simulations have $m = 25$ and $s = 5$.

5.2 1D Lattice

The pairs of qubits in a 1D lattice that correspond to a depth of 1 can be seen in Figure 1. Specifically, we simply alternate between two lists of pairs of qubits, which we refer to as two separate rounds. Likewise, each round corresponds to a depth of 1. Figure 2 shows our results for $-\log_2(\langle P_c \rangle)$ as a function of depth in a 1D lattice with $n = 625$.

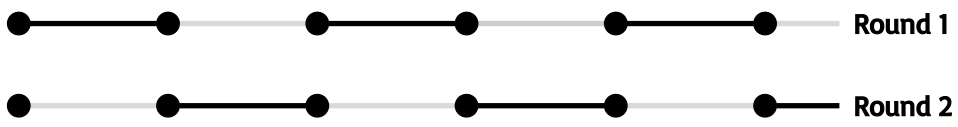

Figure 1: Qubits in a 1D Lattice. We alternate between two rounds and each round corresponds to a depth of 1. Black dots represent qubits, black lines represent random two qubit gates, grey lines represent inactive gates.

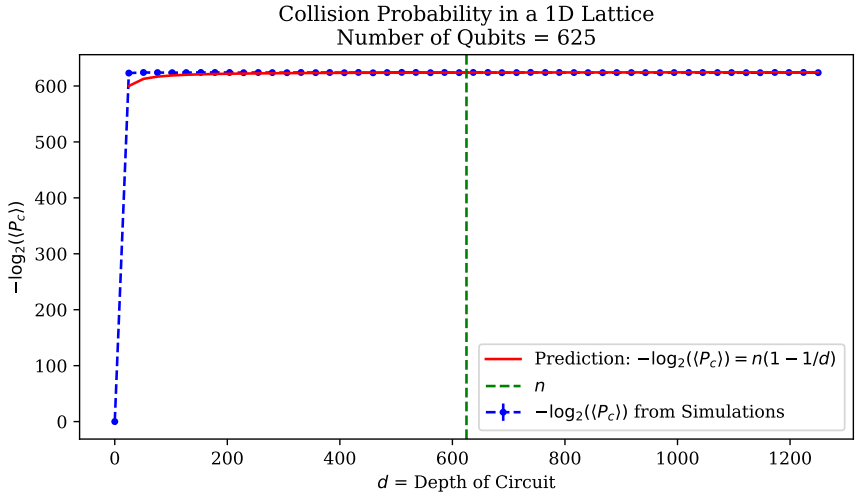

Figure 2: $-\log_2(\langle P_c \rangle)$ as a function of d in a 1D lattice with $n = 625$ qubits.

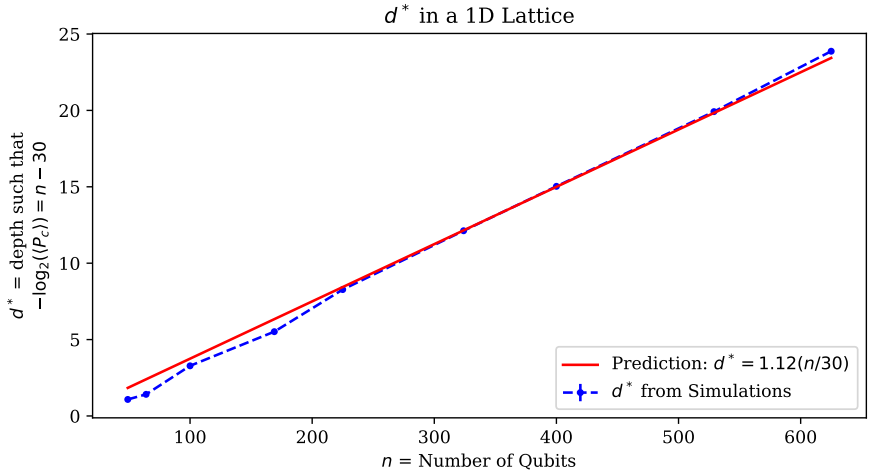

Figure 3: d^* as a function of n in a 1D lattice.

We would like to note here that in the 1D lattice we did not compute the collision probability at each depth. Rather, we only computed the collision probability for 50 different values of the depth, which were approximately evenly spaced. The reasoning for this is two-fold. First, computing the collision probability given a tableau is the most expensive operation in our simulations. Second, in the 1D lattice, we needed to compute the collision probability for large values of depth on the order

of $O(n)$. Likewise, limiting the number of data points we collected saved a significant amount of time while running our simulations.

We see that the numerical results obtained from our simulations support Conjecture 1, which predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - O\left(\frac{1}{d}\right)\right) \tag{5.1}$$

Moreover, for a 1D lattice, we would expect the circuit to saturate at a depth of $d \sim n$. Likewise, looking at Figure 2, it is clear that we will have $-\log_2(\langle P_c \rangle) = O(n - n/d)$ and the circuit clearly saturates well before $d = n$.

To provide further evidence in support of Conjecture 1, we define d^* to be the depth such that $-\log_2(\langle P_c \rangle) = n - 30$. Likewise, if we let $O(1/d) = c/d$ for some constant c, the conjecture predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - \frac{c}{d^*}\right) = n - 30 \quad \Rightarrow \quad d^* = c\left(\frac{n}{30}\right) \tag{5.2}$$

In Figure 3 we have plotted d^* as a function of n and chosen a value of c that fits the data well. As we can see, the values of d^* computed from our simulations show a clear linear trend, meaning our simulations in a 1D lattice largely agree with the predictions of Conjecture 1.

5.3 2D Lattice

The pairs of qubits in a 2D lattice that correspond to a depth of 1 can be seen in Figure 4. Similar to the 1D lattice, we perform a round robin cycling through four different rounds when applying our sets of two qubit gates. Each round in this round robin increases the depth of the circuit by 1. Figure 5 shows our results for $-\log_2(\langle P_c \rangle)$ as a function of depth in a 2D lattice with $n = 900$.

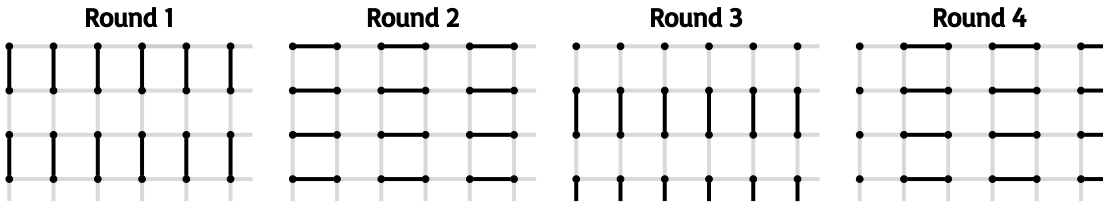

Figure 4: Qubits in a 2D Lattice. We cycle through four rounds and each round corresponds to a depth of 1. Black dots represent qubits, black lines represent random two qubit gates, grey lines represent inactive gates.

In Figure 5, we collect 36 data points corresponding to the values of depth $d = 0, 1, 2, \ldots, 35$ so that we can see how $-\log_2(\langle P_c \rangle)$ behaves as a function of d for depths on the order of $O(n^{1/2})$.

For a 2D lattice, Conjecture 1 predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - O\left(\frac{1}{d^2}\right)\right) \tag{5.3}$$

We also expect that for a 2D lattice the circuit will saturate at a depth of $d \sim n^{1/2}$. Looking at Figure 5, we can see that $-\log_2(\langle P_c \rangle) = O(n - n/d^2)$ and the circuit saturates before $d = n^{1/2}$. However, for small values of d, we see that the prediction provides a somewhat loose upper bound. Likewise, Conjecture 1 can likely be modified to provide a tighter bound for small values of d.

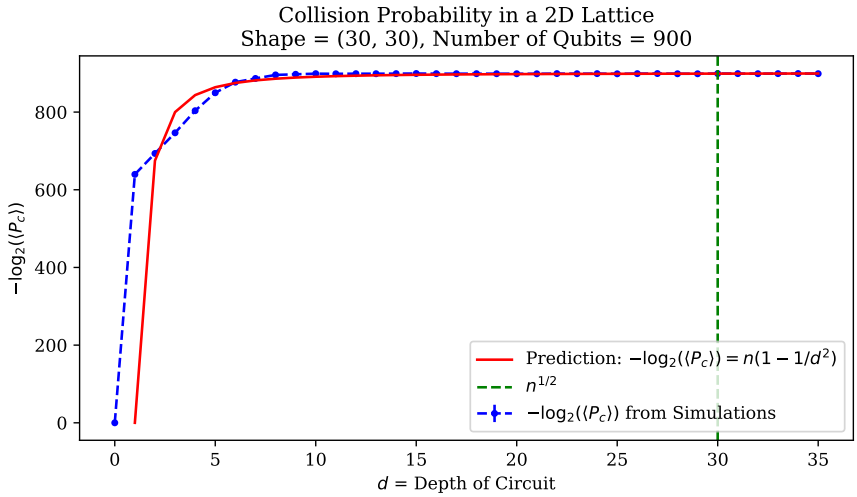

Figure 5: $-\log_2(\langle P_c \rangle)$ as a function of d in a 2D lattice with $n = 900$ qubits.

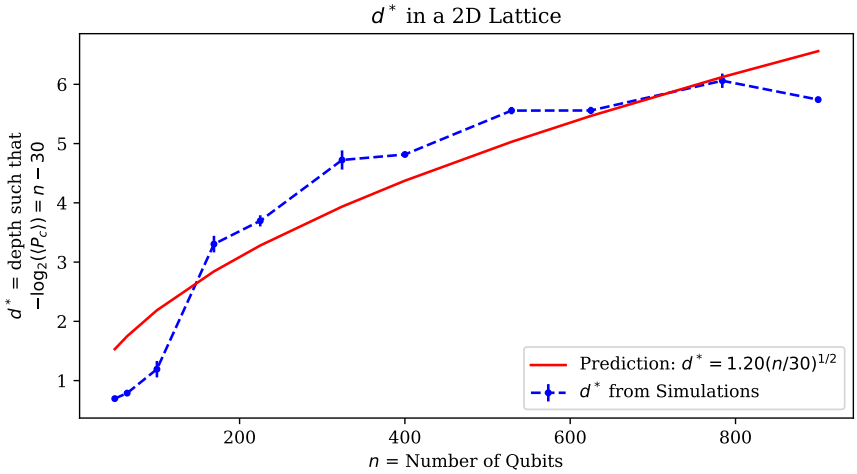

Figure 6: d^* as a function of n in a 2D lattice.

Again, to provide further evidence for Conjecture 1, we look at how d^* changes as a function of n. In this case, the conjecture predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - \frac{c}{(d^*)^2}\right) = n - 30 \quad \Rightarrow \quad d^* = c^{1/2}\left(\frac{n}{30}\right)^{1/2} \tag{5.4}$$

In Figure 6, we have plotted d^* as a function of n in a 2D lattice. Here, we have chosen a value for $c^{1/2}$ that fits the data well. We see that the data collected from our simulations roughly behaves like $n^{1/2}$. However, the data seems to be a bit noisy and it may be the case that a different shaped curve will fit the data better. Modifications to Conjecture 1 that provide tighter bounds for smaller depth will likely address this issue. Nonetheless, Conjecture 1 still provides a meaningful upper bound and asymptotic behavior that largely agree with our data.

5.4 3D Lattice

The pairs of qubits in a 3D lattice that correspond to a depth of 1 can be seen in Figure 7. Analogous to the 2D lattice, we perform a round robin cycling through six rounds and each round increases the depth of the circuit by 1. Figure 8 shows our result for $-\log_2(\langle P_c \rangle)$ as a function of depth in a 3D lattice with $n = 1000$.

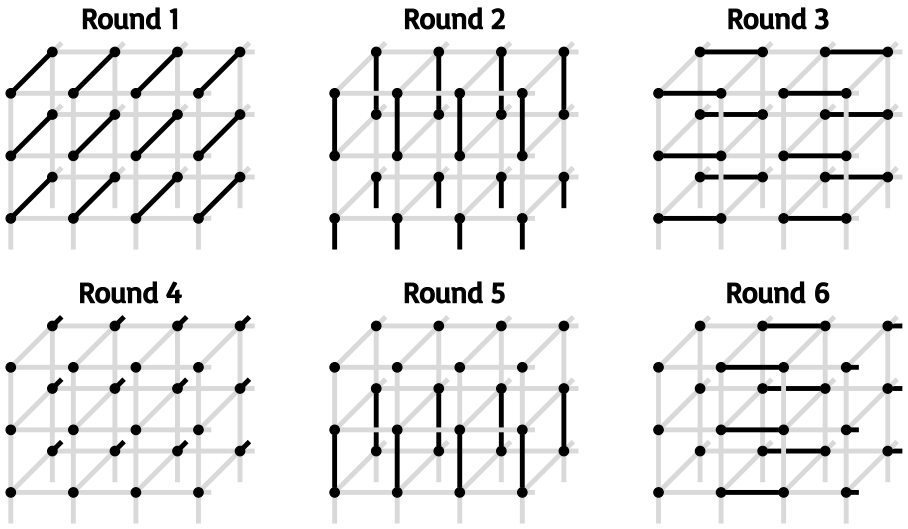

Figure 7: Qubits in a 3D Lattice. We cycle through six rounds and each round corresponds to a depth of 1. Black dots represent qubits, black lines represent random two qubit gates, grey lines represent inactive gates.

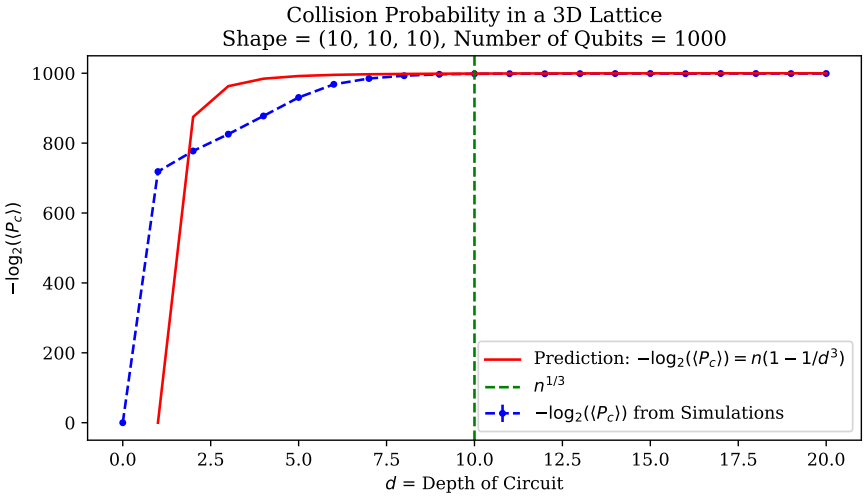

Figure 8: $-\log_2(\langle P_c \rangle)$ as a function of d in a 3D lattice with $n = 1000$ qubits.

In Figure 8, we collect 21 data points corresponding to the values of depth $d = 0, 1, 2, \ldots, 20$ so that we can see how $-\log_2(\langle P_c \rangle)$ behaves as a function of d for depths on the order of $O(n^{1/3})$.

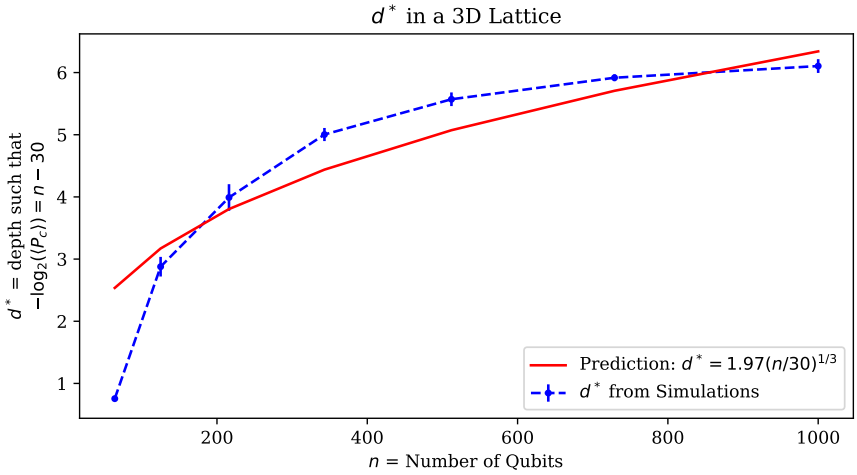

Figure 9: d^* as a function of n in a 3D lattice with $m = 25$.

For a 3D lattice, Conjecture 1 predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - O\left(\frac{1}{d^3}\right)\right) \tag{5.5}$$

We also expect that for a 3D lattice the circuit will saturate at a depth of $d \sim n^{1/3}$. Looking at Figure 8, we can see that $-\log_2(\langle P_c \rangle) = O(n - n/d^3)$ and the circuit tends to saturate around $d = n^{1/3}$. Similar to the case of the 2D lattice, we see that the prediction provides a loose upper bound for small values of d.

We also see how d^* changes as a function of n. Here, Conjecture 1 predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - \frac{c}{(d^*)^3}\right) = n - 30 \quad \Rightarrow \quad d^* = c^{1/3}\left(\frac{n}{30}\right)^{1/3} \tag{5.6}$$

In Figure 9, we have plotted d^* as a function of n and have chosen a value of $c^{1/3}$ that fits the data well. Similar to the case of the 2D lattice, we see that the values of d^* collected from our simulations roughly display a $n^{1/3}$ behavior. However, there may be a better curve that fits the function. Again, we hypothesize that modifications to Conjecture 1 that provide tighter bounds for small values of d will likely address these issues.

5.5 Complete Graph

In a complete graph, we look at $-\log_2(\langle P_c \rangle)$ as a function of the total number of gates applied to the circuit N. Likewise, we simply choose two qubits uniformly at random and apply a gate to those two qubits. After doing so, N increases by 1. An illustration of this process can be seen in Figure 10. Figure 11 shows our results for $-\log_2(\langle P_c \rangle)$ as a function of N in a complete graph with $n = 625$.

Similar to the 1D lattice, we have also chosen to only compute the collision probability for 50 different values of N. The reasons for doing this are the same as those described in Section 5.2. These 50 data points are also approximately evenly space out and we look at the number of gates N on the order of $n\ln(n)$.

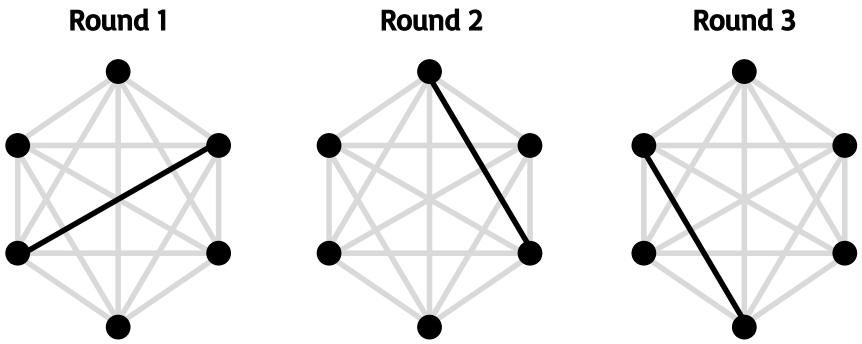

Figure 10: Qubits in a Complete Graph. This is an illustration of 6 qubits laid out in a complete graph. Each round corresponds to a single gate. In each round, two qubits are chosen uniformly at random. Black dots represent qubits, black lines represent random two qubit gates, grey lines represent inactive gates.

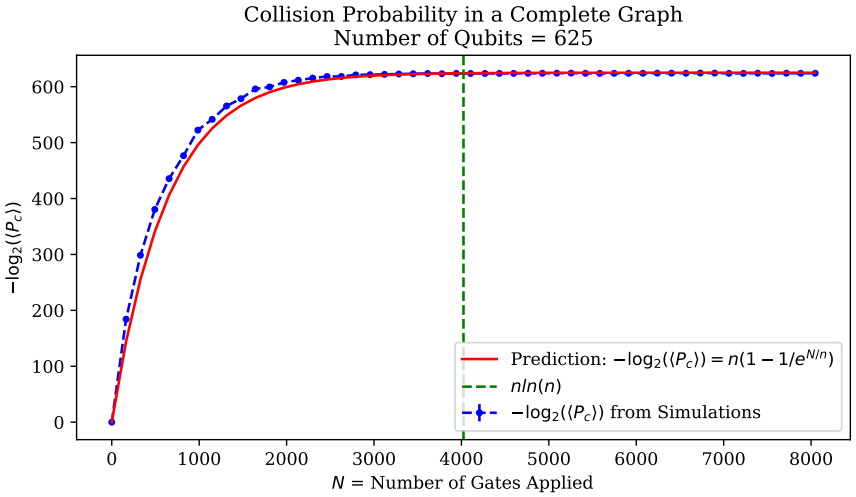

Figure 11: $-\log_2(\langle P_c \rangle)$ as a function of N in a complete graph with $n = 625$ qubits.

We see that the numerical results obtained from our simulations support Conjecture 2, which predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - O\left(\frac{1}{e^{N/n}}\right)\right) \tag{5.7}$$

This conjecture also predicts that the circuit will saturate around $N \sim n\ln(n)$. Looking at Figure 11, we see that $-\log_2(\langle P_c \rangle) = O(n - n/e^{N/n})$ and the circuit saturates around $N = n\ln(n)$.

Similar to the 1D, 2D, and 3D lattices, we provide further support for Conjecture 2 by defining a new quantity N^*. However, in this case, we define N^* to be the number of gates such that $-\log_2(\langle P_c \rangle) = n - n/30$. Likewise, the conjecture predicts

$$-\log_2(\langle P_c \rangle) = n\left(1 - \frac{c}{e^{N^*/n}}\right) = n - \frac{n}{30} \quad \Rightarrow \quad e^{N^*/n} = 30c \quad \Rightarrow \quad N^* = \ln(30c)(n) \tag{5.8}$$

In Figure 12, we have plotted N^* as a function of n. Here, we have adjusted the value of $\ln(30c)$ so the prediction fits the data well. Similar to the case of the 1D lattice, we see a clear linear trend in

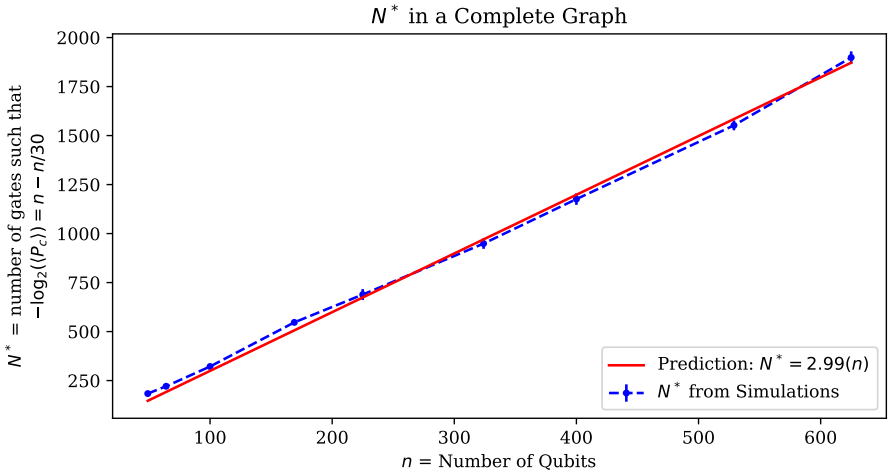

Figure 12: N^* as a function of n in a complete graph.

the values of N^* from our simulations that line up very closely with our prediction. Both Figures 11 and 12 provide data that strongly agrees with the predictions of Conjecture 2. Likewise, the upper bounds provided in Conjecture 2 are quite tight.

5.6 Fluctuations in Collision Probability

So far, we have shown that our numerical results largely support Conjecture 1 in a 1D lattice and Conjecture 2 in a complete graph. We have also shown that Conjecture 1 provides meaningful, but loose, upper bounds in a 2D and 3D lattice. Overall, these numerical results are mostly concerned with the expected value of the collision probability $\langle P_c \rangle$. In this section, we present some numerical results concerned with the standard deviation of the collision probability ΔP_c.

Figures 13, 14, 15, and 16 show our numerical results for $-\log_2(\Delta P_c)$ in a 1D lattice, 2D lattice, 3D lattice, and complete graph respectively. These plots show how $-\log_2(\Delta P_c)$ changes as a function of depth for fixed values of n.

We see some striking similarities when comparing Figures 13, 14, 15, and 16 with Figures 2, 5, 8, 11 respectively. That is, the data collected from our simulations suggest that the values of $-\log_2(\langle P_c \rangle)$ and $-\log_2(\Delta P_c)$ grow similarly as functions of depth and number of qubits.

Furthermore, in Figure 17 we plot both $-\log_2(\langle P_c \rangle)$ and $-\log_2(\Delta P_c)$ as a function of n in a 1D lattice. In these plots, we have chosen large depths for each value of n so that the circuits have been fully saturated. We see that both $-\log_2(\langle P_c \rangle)$ and $-\log_2(\Delta P_c)$ are almost exactly equal to n when the circuits have reached a steady state. Moreover, although not included in this paper, we have found nearly the same exact results when creating the same plots for a 2D and 3D lattice as well as in a complete graph.

These results are quite interesting as they imply that as a circuit saturates, we expect fluctuations in the collision probability to be approximately $\Delta P_c \approx (1/2)^n$. This means that the steady state fluctuations in the collision probability become exponentially small as the size of the circuit grows.

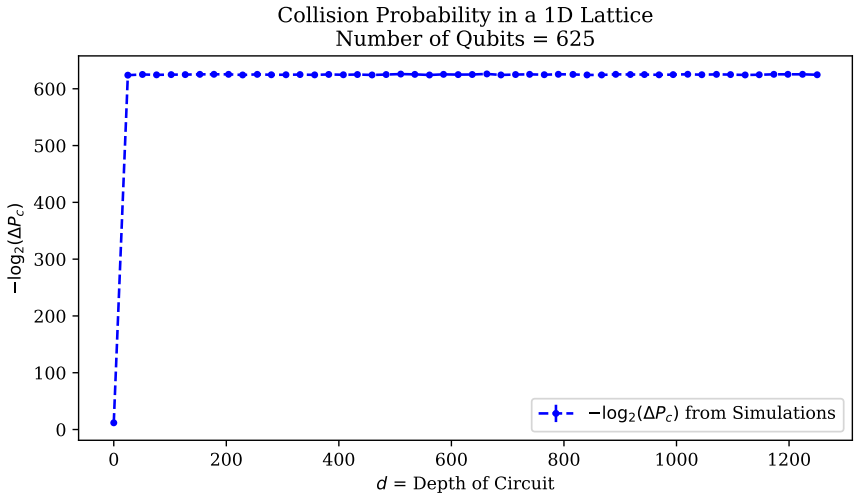

Figure 13: $-\log_2(\Delta P_c)$ as a function of d in a 1D lattice with n = 625 qubits.

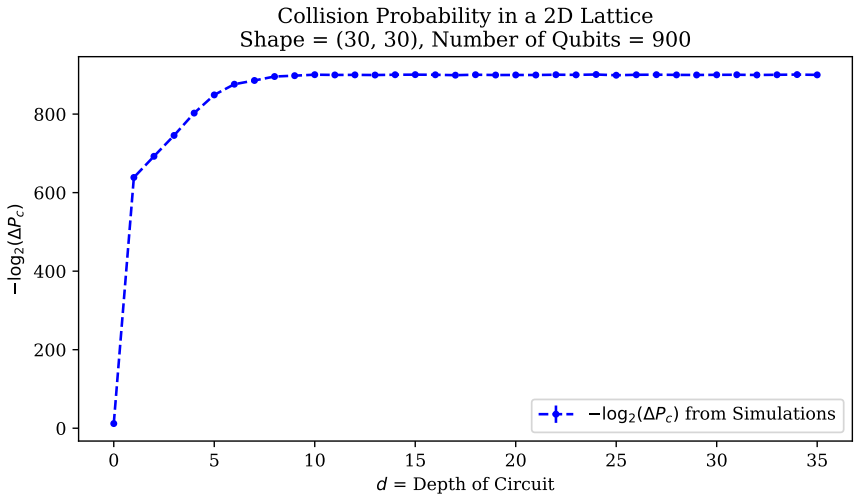

Figure 14: $-\log_2(\Delta P_c)$ as a function of d in a 2D lattice with n = 900 qubits.

5.7 Summary of the Numerical Results

We have tested Conjectures 1 and 2 on three separate accounts: the behavior of $-\log_2(\langle P_c \rangle)$ as function of d or N, the depths or number of gates at which the circuits saturate, and the behavior of d^* or N^* as a function of n. Overall, we found that our data strongly agrees with the predictions of Conjecture 1 in a 1D lattice and Conjecture 2 in a complete graph.

We found that our data roughly agreed with Conjecture 1 in a 2D and 3D lattice. Specifically, we saw that the upper bounds provided by the conjecture seemed a bit loose for small values of depth. Likewise, we hypothesize that modifications to Conjecture 1 will likely be able to address these issues. Nonetheless, our data still suggests that Conjecture 1 provides meaningful upper bounds for a 2D and 3D lattice.

Lastly, we diverged slightly from our original conjectures to discuss fluctuations we observed in the collision probability. Our numerical results suggest that $-\log_2(\langle P_c \rangle)$ and $-\log_2(\Delta P_c)$ grow

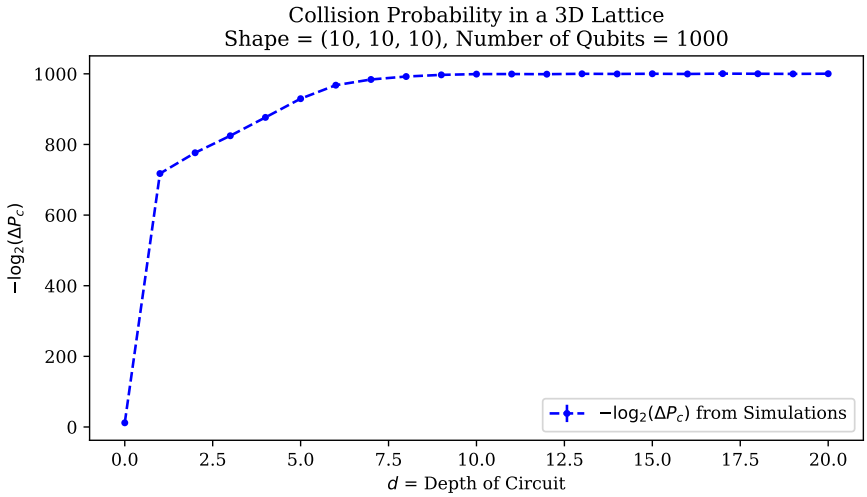

Figure 15: $-\log_2(\Delta P_c)$ as a function of d in a 3D lattice with n = 1000 qubits.

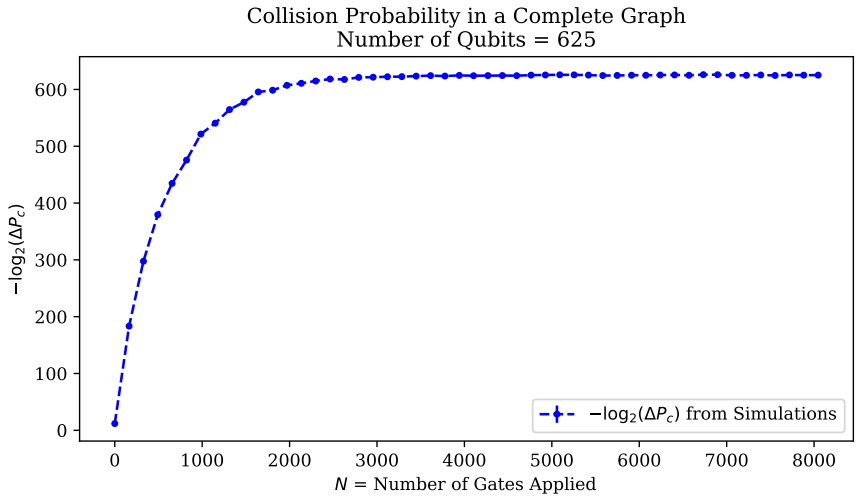

Figure 16: $-\log_2(\Delta P_c)$ as a function of d in a complete graph with n = 625 qubits.

similarly as functions of depth and number of qubits. We also observed that the standard deviation of the collision probability $\Delta P_c \approx (1/2)^n$ for large depths where the circuit is saturated, regardless of its geometry.

6 Conclusion

Throughout this paper, our main focus has been on studying one statistical property of pseudo-random quantum circuits: how the expected value of the collision probability behaves as a function of the depth of the circuit and the number of qubits in the circuit. In doing so, we began by introducing Conjectures 1 and 2, which we have tested numerically via simulations of pseudo-random quantum circuits. Before presenting the structure and results of our simulations, we described some of the mathematical principles allowing us to efficiently simulate quantum circuits composed only of Clifford Gates. We also presented an algorithm to efficiently compute the collision probability

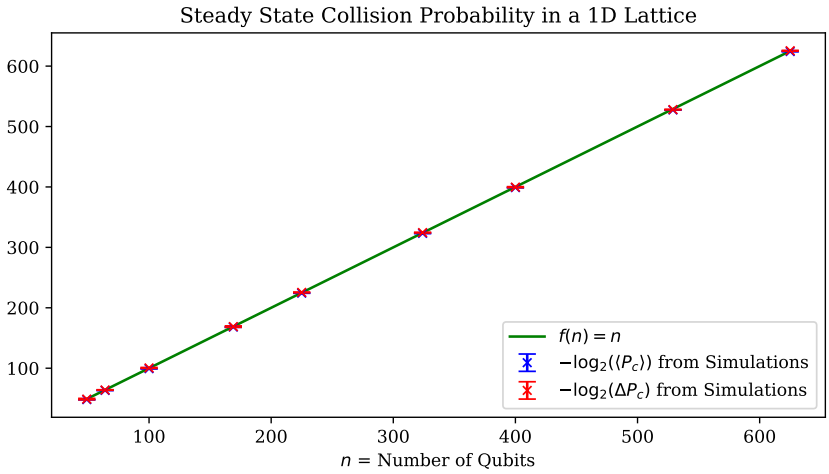

Figure 17: $-\log_2(\langle P_c \rangle)$ and $-\log_2(\Delta P_c)$ as a function of n in a 1D lattice at saturated depths.

of such circuits.

Overall, our numerical results for a 1D lattice and a complete graph strongly agree with the predictions made by Conjectures 1 and 2 respectively. Our numerical results also suggest that Conjecture 1 likely provides meaningful, but somewhat loose, upper bounds for a 2D and 3D lattice. We also found that $-\log_2(\langle P_c \rangle)$ and $-\log_2(\Delta P_c)$ behave similarly as functions of depth and number of qubits and that steady state fluctuations in the collision probability are approximately $\Delta P_c \approx (1/2)^n$.

References

[1] S. Aaronson and L. Chen. Complexity-Theoretic Foundations of Quantum Supremacy Experiments. *ArXiv e-prints*, Dec. 2016, arXiv:1612.05903.

[2] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(5):052328, Nov. 2004, arXiv:quant-ph/0406196.

[3] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven. Characterizing Quantum Supremacy in Near-Term Devices. *ArXiv e-prints*, July 2016, arXiv:1608.00263.

[4] R. Koenig and J. A. Smolin. How to efficiently select an arbitrary Clifford group element. *Journal of Mathematical Physics*, 55(12):122202, Dec. 2014, arXiv:1406.2170.

[5] matthewkhoury96. matthewkhoury96/random_quantum_circuits: Software to Simulate Pseudo-Random Quantum Circuits, Aug. 2017. doi:10.5281/zenodo.846467.

[6] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.