Quantum Complexity and Quantum Supremacy

Melody Hsu
mh4133
mh4133@columbia.edu

Julia Martin
jm5251
jm5251@columbia.edu

Rockwell Weiner
rjw2164
r.weiner@columbia.edu

1 Introduction

The question of quantum supremacy has been the focus of significant research in recent decades. It is widely believed, but not unconditionally proven, that quantum computers are able to efficiently solve many problems that are intractable using classical architecture. A natural question that arises from the quantum supremacy conjecture pertains to exactly what aspects of quantum computation lead to its purported power. Increasingly, research is being conducted not only on what these aspects are, but how much of that power is needed - and what problems, if any, can be solved given a limited amount of that power that could not be solved otherwise. Conclusive results in this area could have dramatic implications in numerous fields of theoretical computer science, including but not limited to complexity theory and cryptography.

This motivates the study of several complexity classes which have gained some attention as potential areas where answers to these questions may be hidden. We will give a brief overview of the current state of affairs in the world of quantum complexity, current known or believed relations between classical and quantum complexity classes, and implications and effects of unproven results. Finally, we propose some problems which may be fruitful areas of further investigation, as well as posit our own hypotheses to these questions.

This paper explores a handful of candidate paradigms which may be fruitful in demonstrating quantum advantage. As laid out by Kahanamoku-Meyer et al. [2021], three of the most popular candidates for this are passing interactive protocols, solving difficult deterministic problems, and sampling with entanglement. We discuss in some depth examples of all three candidates, along with the power of their results, and their feasibility of implementation. We begin with several promising interactive protocols, with the main focus on IQP. Next, we cover a paradigm for solving deterministic problems, DQC1. Finally, we close with an overview of the more physics-focused Boson sampling.

2 The Story So Far

The existence of classical problems for which there are no efficient *quantum* algorithms (it is widely believed, for example, that quantum computers *cannot* efficiently solve NP-complete problems) suggests that quantum computing is not so powerful that it subsumes all of classical complexity theory. Furthermore, it suggests that there is still interesting work to be done in the interstitial space in between the polynomial hierarchy and BQP - suggesting that a deeper understanding of problems that reside here may be able to shed further light on what separates BQP and PH. In this section, we give a brief overview of some of the foundational work in this space, up to the current state of affairs.

2.1 Non-Universal Bases and Quantum Computation

Non-universal bases are a natural starting point to investigate the question of quantum supremacy - if we start from a universal quantum basis and incrementally strip away elements in that basis, at what point can we no longer see improvement over the computing power of a classical (universal) basis? Work has been done manipulating the universal set of Clifford and T gates. By limiting the number

This is a course project for *COMS E6998 Frontiers of Quantum Complexity and Cryptography* at Columbia University.

of T gates available to us, are we still able to solve the full range of problems that a universal Clifford + T gate set is able to solve? What is the threshold number of T gates required? To what degree is a limited quantum circuit simulable by a classical circuit? While we do not tackle the question of universal gate sets directly, we explore other restricted models of quantum computation that approach related ideas from different angles, with the shared goal of finding evidence for quantum supremacy.

2.2 An (Brief) Overview of Classical and Quantum Computational Complexity

This section will be short, as we are assuming that the reader already has a solid understanding of the basics of both classical and quantum computational complexity. For a more in-depth description of the limits of quantum computing, see Aaronson [2008]. Rather than going over what they are, we will instead go over what quantum computation could achieve in the context of classical complexity. Namely, consider NP problems, for which a given solution is verifiable in polynomial time but that solution cannot be found efficiently. Most computer scientists believe that $NP \neq P$, although it has yet to be formally proven as a theorem, so the way that we circumnavigate that and attempt to solve NP problems in polynomial time is to try to broaden what we consider a computer, which is where quantum mechanics comes into play. Quantum mechanics allows you to store vast amounts of information on small amounts of particles, and if they are measured correctly, you can extract some of that information. Researchers are still narrowing down exactly what problems quantum computers can solve (known as the BQP complexity class), but it is known that they can efficiently solve those in P and some in NP (such as factoring, thanks to Peter Shor). However, contrary to myth or popular believe, they are not known to be able to efficiently solve NP-complete problems; at least, at the moment no efficient quantum algorithm has been found for an NP-complete problem (although there is also no proof that one does not exist). However, BQP could also contain some problems that are believed to be beyond NP (in $PSPACE$), but that is also not known for sure. It is known that BQP cannot extend beyond $PSPACE$.

So why do we continue to try to build quantum computers? Even though they face many similar limitations to our classical computers, they could provide extraordinary insight into the field of quantum mechanics itself. Apart from being able to simulate quantum physical phenomenon (with a universal QC), the process of building them would involve more indepth research into some of the murkier areas of physics. If scientists were to fail, and it were discovered that a universal quantum computer is truly impossible to build, it would likely turn the world of physics upside down, and it would require a completely new paradigm through which to view the universe.

2.3 Google Sycamore

Although this survey focuses primarily on theoretical constructions of quantum advantage, it behooves us to describe the state-of-the-art experimental evidence provided by Arute et al. [2019]. The authors built and tested the "Sycamore" processor, consisting of 53 functional qubits.

The chip was used to sample a pseudo-random quantum circuit, a task which seems to be exponentially difficult for a classical computer. From the experiments performed, it appears that this processor was indeed able to sample correctly from this circuit. However, the caveat is that in order to check whether these samples truly matched the expectation, one would need $O(2^{53})$ operations, which is of course infeasible even on current supercomputers. Instead, the researchers verified the results for smaller circuits, and extrapolated that the success holds even for the larger number of qubits.

As the authors point out, even if this experiment does not definitely prove a quantum advantage, it is at the very least a key step in demonstrating that quantum computation is possible, and circuits of meaningful size can be implemented, albeit with limited gate sets. Additionally, the fact that even our most powerful classical computers are already at their limits when validating the results of these experiments underscore the value of constructing a simpler proof-of-advantage procedure.

3 Interactive Proofs of Quantumness

3.1 IQP

The class of quantum computation known as "Instantaneous Quantum Polytime" (IQP) was first described by Shepherd and Bremner [2009]. This is a limited class of quantum computation, which

only allows those operations which are temporally unstructured, i.e., only qubit operations which commute are allowed. The original paper proposed a specific problem, computable under IQP, which was conjectured to be unfeasible for a classical computer to compute. However, Kahanamoku-Meyer [2019] eventually provided a polynomial time algorithm for this problem, effectively spelling the end of IQP.

The motivation behind the original paper was to define a problem which is difficult for a classical computer to solve efficiently, but which a quantum computer could handle quickly. Specifically, the authors hoped to find a problem which was solvable by a minimal quantum computer, one which could be developed relatively soon, and with fewer engineering challenges. Thus, IQP is a class restricted to operations which commute, since this gives the significant advantage that all of the operations could be described via a single Hamiltonian, acting on all qubits. The primary benefit of this "instantaneous" property is that it requires much shorter coherence times for qubits, as they theoretically would only need to pass through a single gate.

Furthermore, the authors lay out the format of a challenge, where Bob, who has access to a quantum (IQP) computer, wishes to convince skeptical Alice that he actually has a quantum advantage. In this case, Alice can construct a sort of one-way function to which she has the key, and where finding the key classically is unfeasible. If Bob is able to find this key relatively quickly, Alice should believe that he truly has a quantum machine.

To this end, the paper sets out a specific interactive protocol by which Bob can prove he has an IQP oracle. First, Alice chooses a code, from which she then constructs a matrix meeting certain properties. She additionally obfuscates this matrix by appending rows orthogonal to the vector defined by her original code (and shuffles the rows randomly). Bob then receives this matrix, and is tasked with interpreting it as a program which he can run on his IQP machine. He then shares with Alice the samples generated by several runs of this program, and she can check whether these samples are orthogonal to her original vector with the correct probability. If so, she should believe that he has a quantum computer.

For example, when Alice constructs the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

, Bob is expected to interpret it as a sort of "program", essentially a Hamiltonian of the form $X_1 + X_1X_2 + X_2X_3 + X_1X_3X_4 + X_2X_4 + X_3 + X_4$, multiplied by a constant. Bob then sends back the results of a series of independent trials of applying this matrix and measuring the outcome.

The paper goes into some detail as to why the authors expect this problem to be difficult for a classical machine to solve. However, they of course are not able to prove this difficulty unconditionally, and in that spirit, posted a public challenge problem with a $25 reward. Surprisingly enough, this reward was actually claimed (although at the time of this paper, not yet received) by Kahanamoku-Meyer, who demonstrated an algorithm for not just convincing a skeptical Alice, but also for finding exactly the secret vector which she selects.

Kahanamoku-Meyer also discusses several possible modifications to the original algorithm, which could potentially make the problem once again difficult for a classical prover. However, these each have issues of their own, and none are currently known to be serious candidates. While IQP is certainly an interesting class to analyze, at this moment it seems that it has little to no use as a test of quantum advantage.

3.2 Trapdoor Claw-Free Functions

One promising test of quantum advantage is described in Kahanamoku-Meyer et al. [2021], by the same researcher who classically solved the IQP challenge. This test also uses an interactive proof system, but relies primarily on the cryptographic hardness of reversing a one-way function, specifically two-to-one trapdoor claw-free functions (TCFs).

A function $f(x)$ is claw-free if it is hard to find any pair (x_0, x_1) such that $f(x_0) = f(x_1)$. In other words, it is difficult to find a collision. And the requirement that it has a trapdoor means that there should exist some t such that given t and y it is easy to find both x_0 and x_1 where $f(x_0) = f(x_1) = y$.

The entire protocol is visualized in Figure 1, for a generic TCF. The construction works similarly for any TCF, with Decisional Diffie-Helman and the Rabin function being popular choices. However, the mathematical analysis focuses specifically on the Rabin function, $x^2 \mod N$ where N is the product of two large primes. This is chosen because of the relatively small number of gates required to implement it for a quantum computer, and its simpler analysis.

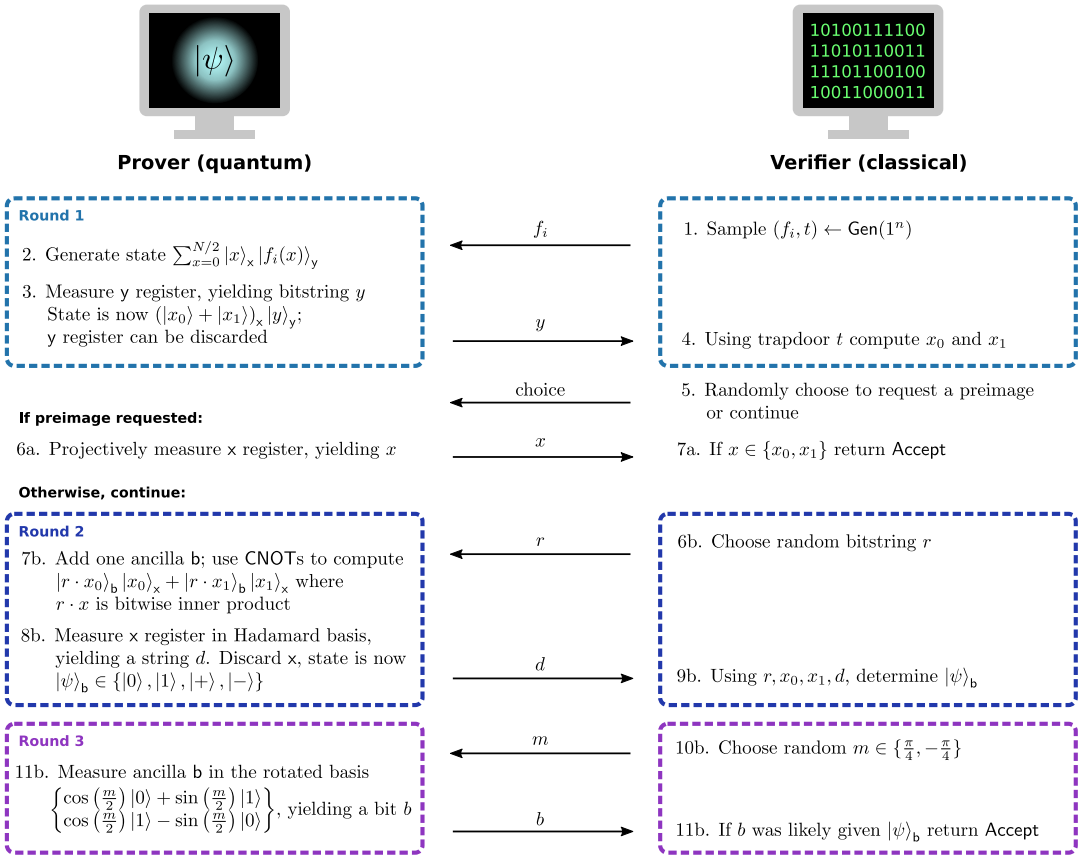

Figure 1: Diagram of the interactive proof protocol for a generic trapdoor claw-free function. Pulled from Kahanamoku-Meyer et al. [2021].

The authors show unconditionally that a quantum prover can pass this test with probability either 100% or 85%, depending on the verifier's choice in step 5. Meanwhile, under the assumption that we truly have a TCF, any classical prover can succeed with probabilities at most 100% or 75%, again depending on the verifier's choice in step 5. It is no coincidence that these values are identical to those found in the CHSH game, as one possible path the interactive proof may follow essentially mimics this well-known game.

Finally, the paper describes some details of handling noise in the quantum circuit, and how a lack of fidelity can be handled by an increase in the runtime. Additionally, the authors describe some of the more feasible implementations of a circuit for the Rabin function, which are hopefully candidates for implementation *relatively* soon. The conclusion is that with our current technology we would need a circuit of about 1000 qubits in order to successfully pass this test for a meaningfully large input. However, this is certainly closer to an upper bound than a lower.

4 DQC1

4.1 DQC1 and the "One Clean Qubit" Problem

The question of just how much "quantum-ness" is enough to effect pronounced reductions on the hardness of computing difficult problems is central to the work investigating near-quantum computation. Initially proposed by Knill and Laflamme [1998], the "One Clean Qubit" problem is defined as follows: Suppose that rather than having access to some number of pure states, our computer has access to only one clean qubit and an arbitrary number of qubits in the maximally mixed state. What problems can we solve that we couldn't solve before - if any? And similarly, what problems are we still unable to solve?

Knill and Laflamme propose the one clean qubit framework to formalize the notion of a computational model that exists somewhere in between classical and quantum computing. The model, called *deterministic quantum computation with one quantum bit* (DQC1), stands in contrast to *deterministic computation with pure states* (DQCp) - which is to say, standard quantum computation.

Computation using the DQC1 model proceeds as follows: Starting with an initial state over n qubits of the form $|\psi\rangle = |0\rangle\langle 0| \otimes (\frac{I}{2})^{\otimes(n-1)}$, we apply some polynomially-sized operation (i.e., some sequence of unitaries consisting of the tensor products of Pauli matrices I, X, Y, Z). After applying the unitaries, we measure a single output qubit in the computational basis. This measurement constitutes the outcome of some computation in a DQC1 algorithm.

A representative example of a generalized DQC1 circuit diagram is depicted below. The circuit receives as input n qubits; one qubit in state $|0\rangle$, and the remaining $n-1$ states in the maximally mixed state. The first qubit is the one that is measured after applying some polynomially-sized system of unitaries.

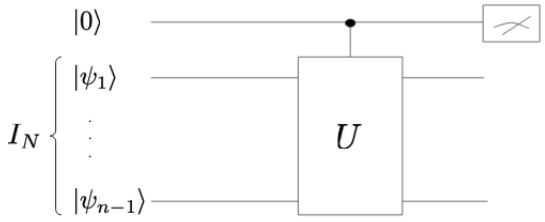

Figure 2: A DQC1 circuit.

4.2 The Power of One Clean Qubit

DQC1 directly addresses the question of "how much quantumness is needed to exhibit quantum supremacy" - i.e., can giving an algorithm access to restricted quantum machinery offer some observable improvement in performance? As it turns out, DQC1 can solve nontrivial, fully quantum problems efficiently, proving that it is indeed a fully quantum model of computation, despite the fact that it does not harness the power of full quantumness. This is in some ways unexpected - the presumed advantage that quantum algorithms have over classical algorithms is the ability to leverage quantum entanglement, which is not present in a DQC1 system (Datta et al. [2008]). However, while there is limited entanglement between the pure-state qubit and the mixed-state qubits, there are nonclassical correlations present in a typical instantiation of a DQC1 circuit. Datta shows that these nonclassical correlations can be measured through a quantity that he defines as *quantum discord*, defined $\mathcal{D}(S, M) = \mathcal{I}(S : M) - \mathcal{J}(S : M)$, where S, M are two systems, and $\mathcal{I}(S : M) = H(S) + H(M) - H(S, M)$ is the mutual information between S and M and $\mathcal{J}(S : M) = H(S) - \widetilde{H}(S|M)$ is the measurement-independent mutual information. The quantum discord value for S the pure-state qubit and M the mixed-state qubits indicates the presence of nonclassical correlations in the overall DQC1 circuit. It is hypothesized that these nonclassical correlations would be the source of DQC1's advantage, if any, over classical computation.

We now give a few examples of problems exhibiting this advantage. The problem of adding Hamiltonians is currently not known to be solvable on classical machinery, but is reasonably efficient

in DQC1. DQC1 has been shown to be able to efficiently estimate the coefficients of a Pauli operator expansion of a unitary U; using these coefficients, we can create unitary embeddings of two-bit Hamiltonians and then perform addition over those embeddings. We give an overview of the procedure below.

Claim: There is a polynomial-time algorithm for two-bit Hamiltonian addition that is in DQC1.

Proof. We first show that DQC1 can compute a related problem, the estimation of coefficients for the Pauli expansion operator of a unitary U.

Define the Pauli expansion operator of U to be $U = \sum_b \alpha_b P_b$, where $P_b \in \{I, X, Y, Z\}$ and $\alpha_b = \frac{1}{2^n}\text{Tr}(P_b U)$, with $\alpha_b \in \mathbb{C}$. We will show how to estimate α_b, which allows us to fully characterize U. Define V to be the "conditional U operator" which maps $|0\rangle|b\rangle$ to $|0\rangle U|b\rangle$ and leaves 1-states alone (i.e., $|1\rangle|b\rangle$ is mapped to $|1\rangle|b\rangle$). Such an operator can be constructed using a series of two-bit XOR gates, which will take overall linear resources (Barenco et al. [1995]). The following two computations will thus give us the values for the real and imaginary parts of α_b:

$$\frac{1}{2^{n+1}}\text{Tr}(X^{(1)}VX^{(1)}P_bV^\dagger)) = \frac{1}{2}(\text{Tr}(UP_b) + \text{Tr}(P_bU^\dagger)) = \text{Re}(\alpha_b)$$
$$\frac{1}{2^{n+1}}\text{Tr}(Y^{(1)}VX^{(1)}P_bV^\dagger) = \frac{i}{2}(\text{Tr}(UP_b) - \text{Tr}(P_bU^\dagger)) = -\text{Im}(\alpha_b)$$

Thus, it takes only two computations to obtain the real and imaginary parts of the coefficients of U.

Now we have the ingredients necessary to calculate the sum of arbitrary two-bit Hamiltonians. Any two-bit Hamiltonian can be represented in Hilbert space using a unitary embedding; the resulting summation can thus be approximated as the summation over complex conjugates of P_b as described above. □

As there is currently no known classical algorithm to compute the sum of two-bit Hamiltonians, the ability of a DQC1 algorithm to solve such a problem is evidence for DQC1's supremacy over classical computing. DQC1 algorithms have also been shown to solve other problems which are currently intractable on classical computers. Among these problems is approximation of the evaluation of the Jones polynomial at several points, an important equation in fields as diverse as knot theory, statistical mechanics and quantum field theory (Passante et al. [2009]). Exact evaluation of the Jones polynomial at most points is #P-hard; and evaluating the Jones polynomial at some specific points has been shown to be BQP-complete.

The following is the DQC1 algorithm to compute the fifth root of unity of the Jones polynomial, given by Shor and Jordan [2007].

Claim: Approximating the fifth root of unity of the Jones polynomial is DQC1-complete.

Proof. The following is a sketch of the algorithm for approximating the fifth root of unity of the Jones polynomial. (Briefly, a Jones polynomial is a knot polynomial that assigns a Laurent polynomial in the variable $t^{\frac{1}{2}}$ with integer coefficients. In our construction, we define $t = A^{-4}$, where $A = e^{-\frac{i3\pi}{5}}$.)

The construction of a DQC1 circuit can be thought of as acting on b maximally mixed qubits, plus some constant $\mathcal{O}(1)$ number of "clean" qubits. In terms of the computational basis, we can equivalently think of the first b qubits being in a uniform probabilistic mixture of the 2^b classical bitstring states. Before we proceed further, we must lay some groundwork that will help us think about the Jones polynomial (keeping in mind the applications of the Jones polynomial to knot theory).

Let P_n be the set of all strings of length n of the following form: a string in P_n is composed of consecutive p characters, interspersed with $*$ characters, such that no two $*$ are adjacent. Such a set P_n will have f_{n+2} elements, where f_i is the i^{th} Fibonacci number. We can map each string of this form into a 0-1 bitstring by taking $* = 0$ and $p = 1$; then, a length-n string $s = s_n s_{n-1} \dots s_1, s_i \in \{0, 1\}$ could be represented by $z(s) = \sum_{i=1}^{n} s_i f_{i+1}$. Call this representation the Zeckendorf representation of string $s \in P_n$. (Note that most of the 2^b bitstring states from the setup of the DQC1 circuit will correspond to strings in P_n.)

Now, consider a braid B_n composed of n strands. For the purposes of this proof, a physical braid is an accurate analogue to the braid as a mathematical construct; it is defined as some number of strands arranged in a linear fashion such that they can be indexed and such that they occupy some fixed "lane." These braids may cross over to other "lanes," as a car might cross lanes on a highway; thus, for a crossing to exist, it must come into contact with a strand in a different lane. We can represent such a braid with the Fibonacci representation described above, as well as its corresponding Zeckendorf representation $z(s)$. A crossing in the braid on strands i and $i-1$ corresponds to a linear transformation that changes the value of i only in the n-bit string.

Figure 3: Braid linear combinations (source: Shor and Jordan [2007])

At a high level, we claim (without proof) that because any elementary crossing between two strands in the braid corresponds to a linear combination of two other primitive non-crossed string configurations, we can compute this linear combination using the p and $*$ string representation, upon which all other crossings will build. This crossing can be represented by $(p\hat{*}p) = c(p*p) + d(ppp)$ for c and d defined below and where $p*p$ and ppp are two primitive crossing types and $(p\hat{*}p)$ is the resulting crossover (see figure for a depiction of this crossing). The following set of rules will allow us to compute the matrix representation of these crossings:

$$
\begin{aligned}
(*\hat{p}p) &= a(*pp) \\
(*\hat{p}*) &= b(*p*) \\
(p\hat{*}p) &= c(p*p) + d(ppp) \\
(p\hat{p}*) &= a(pp*) \\
(p\hat{p}*) &= d(p*p) + e(ppp)
\end{aligned}
$$

where

$$
\begin{aligned}
a &= -A^4 \\
b &= A^8 \\
c &= A^8\tau^2 - A^4\tau \\
d &= A^8\tau^{\frac{3}{2}} + A^4\tau^{\frac{3}{2}} \\
e &= A^8\tau - A^4\tau^2 \\
\tau &= \frac{2}{1+\sqrt{5}}
\end{aligned}
$$

Now, we relate this set of linear combinations to computing the Jones polynomial. We construct a quantum circuit which extracts the $(i-1)^{\text{th}}$, i^{th}, and $(i+1)^{\text{th}}$ bits from the bitstring and performs the linear operations on these bits as prescribed by the rules above, so that the bitstring now contains those modified bit values. We construct a quantum circuit to compute each crossing in the braid. Finally, we multiply these results together and perform DQC1 trace estimation on the product; this yields an approximation to the Jones polynomial. Note that the construction of circuits and application of linear transformations can all be done in polynomial time, and DQC1 trace estimation is also a polynomial-time operation; thus, this algorithm for computing the Jones polynomial is also polynomial-time overall.

The full proof of DQC1-hardness is omitted, but the idea is to reduce from the problem of estimating the trace of a quantum circuit by specifying an encoding from the set of p- and $*$-strings starting with $*$ and containing no consecutive $*$ characters. We construct a braid using the Fibonacci representation

described previously, and whose Fibonacci representation will encode the corresponding unitary transformation on the encoded bits. The goal of the reduction is to show equivalence between the Jones polynomial representation of the trace closure of this braid and the trace of the encoded quantum circuit. □

4.3 Limitations of DQC1

DQC1 is not a universal model of quantum computation, and as such, it is limited in its ability to solve quantum problems. It is difficult for a DQC1 algorithm to be able to distinguish between a pair of unitaries U and U' that are constructed to act differently on a quantum system conditional on the values of bits 2 through n.

Claim: Given an oracle which will apply some unitary to a quantum system, it is hard for a DQC1 algorithm to distinguish between U and U' such that U', conditional on the values of qubits 2 through n, will flip the value of the first qubit (relative to U).

Proof. Our task will be to estimate the coefficients of U applied to the system, namely $\text{Tr}\langle \mathbf{0}|U^\dagger Z^{(1)} U|\mathbf{0}\rangle$. We begin by preparing a pseudo-pure state from the Pauli matrix Z. Define the operator T_n which maps $|b_0\mathbf{0}\rangle$ to $|(b_0+1)\mathbf{0}\rangle$ and $|b_0 b\rangle$ to $|b_0 b\rangle$ for all $b \neq 0$. We can obtain the coefficient α of $Z^{(0)}Z^{(1)}$ by applying U to bits 1 through n at an intensity of $\frac{\alpha}{2^{n+1}}$, suggesting an exponential decay in the ability of our algorithm to "detect" α with each successive application of the unitary U.

If we perform k computations where we construct a quantum network and call the oracle to apply U, our expected measurement will be $v(U) = \frac{1}{2^m}\text{Tr}(V_r U \dots U V_0 Z^{(1)} V_0^\dagger U^\dagger \dots U^\dagger V_r^\dagger Z^{(1)})$ where r is the number of calls to the oracle, and V_i are the quantum network operators, each having m bits. Composing U with the operator T defined earlier gives us exactly the U' discussed previously such that it flips the first bit conditional on bits 2 through n. The expected measurement on U' is given by $v(U') = \frac{1}{2^m}(\text{Tr}(V_r(-2P)UV_{r-1}U' \dots) + \text{Tr}(V_r U V_{r-1} U' \dots))$.

From these definitions of $v(U)$ and $v(U')$, it can be shown that $|v(U') - v(U)| \leq \frac{4r}{2^n}$. In other words, because the expected measurement of U and U' applied are within an exponentially small distance apart from each other, it will take a DQC1 algorithm an exponential number of queries to distinguish between U and U'; this is a problem that is easy for a quantum algorithm with access to n pure state qubits. Thus, it is clear that DQC1 is limited in scope relative to full quantum algorithms. □

4.4 Key takeaways

What can this tell us? First of all, it is a positive result that DQC1 cannot solve everything that a fully quantum computer can solve. The separation between DQC1 and, say, BQP, seems to suggest quantum supremacy - it is not sufficient to have access to a restricted quantum computer to solve some problems.

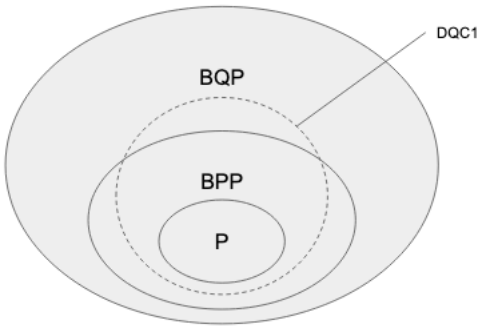

Figure 4: DQC1's hypothesized relations to selected classes

Against the backdrop of the greater complexity universe, we have that P $\subseteq$ DQC1 $\subseteq$ BQP; it is hypothesized that neither DQC1 nor BPP are fully contained in the other. This hypothesis of course

hinges on the belief that BPP $\neq$ BQP; if BPP = BQP, then DQC1 would be equal to both. It is widely believed, but not proven, that P = BPP; this relationship does not necessarily have implications on the relation between BPP and BQP, but it would suggest that BPP $\subseteq$ DQC1, which would be an impressive result.

In fact, there is still some debate about whether DQC1 is powerful enough to simulate all possible *classical* problems (Aaronson et al. [2016]). This may not necessarily suggest that DQC1 is less powerful than P, but it could have interesting implications about the likelihood of obtaining polynomial-time classical algorithms for quantum problems currently believed to be intractable on classical architecture. This would be an interesting area to explore more - given that P $\nsubseteq$ DQC1, what is the usefulness of DQC1 in proving quantum supremacy? Is it then easier to simulate a DQC1 circuit with a classical circuit?

5 Boson Sampling

5.1 Overview

When we think about actually building a quantum computer, there are a few questions that immediately come to mind. For example, what are the physical resource requirements to build such a device, and how much do we need to advance our technology to do this? How long before we could actually build a device which could outperform an existing quantum computer? In 2000, it was shown by Knill, Laflamme, and Milburn that linear optics with photon detection and single-photon sources is sufficient for universal quantum computation (Knill et al. [2000]). However, it would require billions of optical elements and millions of single-photon sources, and thus we are decades away from a universal optical quantum computer. This begs the question: is there an easier way? In 2011, Scott Aaronson and Alex Arkhipov found that there is: Boson Sampling (Aaronson and Arkhipov [2010]). This non-universal method of linear optical quantum computing (LOQC) is easier to construct than a full universal QC as it uses far fewer physical resources. In fact, it may only require 10's of photons and hundreds of optical elements, so it is massively simplified. In one of his blog posts, Scott Aaronson said that the goal of Boson Sampling is to "bridge the gap between what complexity theorists believe is true about quantum mechanics (that it is exponentially-hard to simulate on a classical computer) and what experimentalists can currently demonstrate" (Aaronson [2010]).

Essentially, the idea is to create something that is more practical to construct than a universal quantum computer, but can still solve a computational program that is thought to be intractable for classical computers. On a high level, boson sampling is, of course, a sampling problem, rather than a decision problem, and the solution to this sampling problem is a set of samples drawn from the probability distribution of the possible outcome states of many bosons which were scattered by a linear interferometer. There are several different kinds of boson sampling (Scattershot, Gaussian, etc), but we focus here on the photonic version which is the easiest to construct and scale. In this section, we will see what exactly Boson Sampling is, how it works, why it is hard, and what it can do for us in the future, primarily citing from the work of Aaronson and Arkhipov. As we will see, this problem exhibits a great intermediate step between classical and quantum computing, as well as a great avenue through which we can exhibit a proof of quantum supremacy and an example of quantum advantage. The main difficulty of solving this problem with a classical computer is the dependence on the permanent in the statistics of single-photon measurements. Computationally, devices that can solve boson sampling are hard to simulate classically, but again they are not capable of universal quantum computing. However, researchers continue to search for ways that this method can be utilized in combination with other components to build full universal devices, and it is considered a promising way to prove that quantum computing is certainly more powerful than its classical counterpart.

5.2 Setup

Let's start by discussing the setup of the photonic implementation of boson sampling. Basically, it only requires three things (Gard et al. [2015]).

1. Reliable single-photon sources: most commonly experimentalists use parametric down-conversion crystals
2. A linear interferometer: possibly a laser-written integrated interferometer, or a beam splitter, amongst other options

3. Highly efficient single photon-counting detectors: most likely those that are found on current-biased superconducting nanowires

Take a moment to note here that a boson sampling implementation does not require resources that other universal models of quantum computation do, such as ancillas or entangling operations. That's why it's easy to see how much more practical this is to implement.

So, say that we have some linear optical circuit with *N* modes and *M* indistinguishable single photons (with *N* > *M*). We also have a linear interferometer which is characterized by a unitary matrix *U*, which induces a unitary evolution $\Phi_M(U)$ on the *M*-photon states. Here we note that the Hilbert space of this system is given by a binomial coefficient, and grows exponentially as the size of the system grows. This is due to how photons interact at the beam splitters. The interferometer is injected with an input state of single photons, shown below, where each s_i is the number of photons injected into the i-th mode. Since $N > M$, there will be some single photon number states and some empty (vacuum) states.

$$|\phi_{in}\rangle = |s_1, s_2, ..., s_N\rangle, \text{ with } \sum_{i=1}^{N} s_i = M$$

The output state, after the unitary evolution through the passive linear optics network, is then

$$|\phi_{out}\rangle = \Phi_M(U)|s_1, s_2, ..., s_N\rangle$$

It can be shown (the full calculation is available in the Aaronson and Arkhipov paper) that the probability of detecting some number t_i of photons at the *i*-th output mode is given as

$$p(t_1, ..., t_N) = |\langle t_1, ..., t_N|\phi_{out}\rangle|^2 = \frac{|Perm(U_{S,T})|^2}{t_1!t_2!...t_N!s_1!s_2!...s_N!}$$

Where $Perm(U_{S,T})$ is the permanent of the matrix $U_{S,T}$, which is a matrix obtained from our original unitary U by repeating its i-th column s_i times and its *j*-th row t_j times (think of *S* as denoting the mode assignment list associated with the input state, and *T* denoting the mode assignment list associated with the output state). The goal is to use photon detection to sample the photon-number statistics at the output through repeated trials, with a distribution as shown above. This setup, as we can see, only uses photons, not qubits, and it also does not use qubit gates. A simple diagram of the model can be seen here

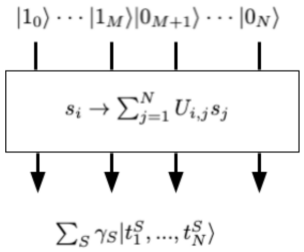

Figure 5: A Diagram of the Boson Sampling Model.

The above figure clearly shows the single photon states in M modes and the vacuum states in the remaining $N - M$ nodes. It's clear to see how they propagate through the linear network which implements a unitary map on the photons, and it outputs a superposition of all of the different ways the photons could have arrived at the output. The output represents all possible configurations, where each *S* is a unique one. The γ_S seen here represents the amplitude associated with a specific configuration, which is proportional to the permanent of some matrix built from the unitary matrix which characterizes the single-boson evolution, so the probability distribution of a specific configuration is roughly $P_S \propto |\gamma_S|^2$. Since we want to build up what this P_S looks like, this is where the photon detectors come in. However, it turns out that we do not need photon-number resolving detection to do this. Aaronson and Arkhipov showed how the number of vacuum modes scales on the order of the square number of single-photon modes, so in a large system with many photons, most of the modes are vacuum modes, so statistically we are almost guaranteed that we will never have more than one

photon in a given mode. Hence we can use bucket detectors to see whether or not photons exist in a mode.

To end this groundwork section, let's do a quick example of what solving the boson sampling problem looks like. Say that we have $M = 2$ photons and $N = 4$ modes, and we inject them such that the input state is $|1, 1, 0, 0\rangle$, so there is a single photon in the first two input modes. We can ignore output states with more than one photon in each mode, as that can be shown to happen with negligible probability, so there are $\binom{4}{2} = 6$ possible output two photon states:

$$(1, 1, 0, 0), (1, 0, 1, 0), (1, 0, 0, 1), (0, 1, 1, 0), (0, 1, 0, 1), (0, 0, 1, 1)$$

Let σ_i denote the *i*-th output state (so $\sigma_2 = (1, 0, 1, 0)$). A possible solution to this boson sampling problem could be a series of outcomes like $\sigma_1, \sigma_3, \sigma_2, \sigma_2, \sigma_6$

5.3 How Complex Is It?

Let's take a moment here to discuss why boson sampling is so hard. The complexity of performing this problem on a classical computer is a crux of Aaronson and Arkhipov's paper, and though we will not go over all 100 pages of their proof, it is important to bring in the following key points which make classically simulating this problem nearly impossible:

1. There are an exponential number of terms in the output superposition, so we cannot calculate the full state vector on a classical computer
2. The state is highly entangled, so we cannot sample each mode independently to work out the statistics
3. The amplitude of each term is related to a problem that is known to be $\#P$-hard (calculating the permanent), so they cannot be calculated in polynomial time

By far the most important point here is number 3. Calculating the permanent of a matrix is known to be $\#P$-hard, as is calculating it even to within some multiplicative error. Along these lines, if an algorithm could efficiently simulate boson sampling classically, the probability distribution would have been approximated in the BPP^{NP} complexity class, implying that $P^{\#P} = BPP^{NP}$. Thus Aaronson and Arkhipov showed these two facts, in conjunction with Toda's Theorem, would have very severe consequences, namely a collapse of the polynomial hierarchy to the third level, so boson sampling is classically inefficient.

What this means is that we can put a range on how many photons is the ideal number to work with in order to prove quantum supremacy. Essentially, we want to work with enough photons so that the Extended Church-Turing Thesis becomes less tenable as we continue to add photons (i.e. we doubt the truthfulness of it more and more with more photons); however, we don't want so many that a classical computer could not even efficiently verify that the probability distribution holds. Boson sampling does not have any known witness that a classical computer can efficiently verify, so for a large number of photons, even if a device is correctly solving boson sampling there might be no feasible way to prove this without presupposing the truth of the physical laws being tested. So for experimental purposes, Aaronson and Arkhipov state that the ideal number of photons, M is that "for which a classical computer has some difficulty computing an $M \times M$ permanent, but can nevertheless do so in order to confirm the results. We estimate this range as $10 \leq M \leq 50$" Aaronson and Arkhipov [2010]. There are some independent error models for which the hardness of boson sampling breaks down as it is scaled to large systems, so while some researchers believe that the experimental demonstration of boson-sampling would provide evidence against the Extended Church-Turing thesis, it cannot as of yet be disproven in the setting of a physically realistic error model.

5.4 A Note on the Permanent

It's important to remember that boson sampling does *not* calculate the permanent of an a-priori given matrix. While Boson Sampling does provide the ability to efficiently sample from a distribution, that ability does not translate into the ability to efficiently estimate the underlying probabilities. It is true that on *M* photons and $N >> M$ modes, the output of the sampler is an $M \times M$ submatrix which corresponds to the positions of the *M* photons in various modes, we are not controlling the submatrix

of input/output modes which is samples, so all we really can say for sure is that said submatrix has a large permanent. It is feasible to calculate if there are very few possible outcomes, but once you get exponentially many possible outcomes the task becomes too computationally intractable, and the possible output states increases exponentially with the number of photons you work with. As the output is a superposition with an exponential number of terms, each amplitude is proportional to a different matrix permanent, and in order to calculate said permanent you would need to know a particular probability with a high degree of precision. In theory, you could estimate the permanents to any desired degree of accuracy if you allow for arbitrarily many samples from the output and count the fraction of the times the *M* input photons come out in some *M* different ports; however, Boson Sampling is fundamentally about the efficiency of sampling, so this approach is antithetical to the ultimate goal of the problem.

It is a common misconception that boson sampling solves a $\#P$-complete problem, which computer scientists do not believe is possible for quantum computers (at least efficiently). It is, however, a sampling problem which is a function of a set of $\#P$-complete problems. Thinking of it as a solution to the permanent is like an archer shooting an arrow into a wall and then drawing a bulls-eye around where the arrow landed, except here you shoot photons through a bunch of beam-splitters and phase-shifters and then you reverse engineer the permanent of the corresponding submatrix based on the output. Essentially, you are creating your own perfect solution from the random result.

5.5 Real Implementations and Looking Forward

As we can see, this is a problem that has promising possibilities for obtaining an unequivocal quantum computational advantage, as it is hard to classically simulate but can be solved naturally by dedicated photonic quantum hardware. This has excited many experimentalists, and has lead to the implementation of progressively larger and more advanced devices in the years since Aaronson and Arkhipov's paper first came out. Very shortly after the paper came out, a few groups implemented functioning constructions using three photons with five and six modes, building up to nine modes shortly thereafter (Carolan et al. [2014]). These initial experiments mainly represented proofs-of-principal constructions of a functioning boson sampling device, and ultimately signified the first steps towards experimental demonstrations of quantum supremacy via this problem. Below, we show a table with relevant details from some of the main photonic boson sampling experiments which have been reported to date, with m detected photons and n optical nodes (Daniel J. Brod [2019]). Also note that the acronyms SPDC and SPAD stand for spontaneous parametric down conversion and silicon single-photon avalanche photodiodes respectively, SBS is scattershot boson sampling, and GBS is Gaussian boson sampling.

Experiment	m	n	Source	Unitary Transformation	Detector
Broome et al	3	6	SPDC	Fiber Splitters	SPAD
Spring et al	3	6	SPDC	Integrated Optics (UV Laser Written)	SPAD
Tillmann et al	3	5	SPDC	Integrated Optics (fs Laser Written)	SPAD
Spagnolo et al	3	9	SPDC	Integrated Optics (fs Laser Written)	SPAD
Carolan et al	4	21	SPDC	Integrated Optics (continuous coupling, SiN)	SPAD
Wang et al	5	9	Quantum Dot	Assembled Micro-Optics	SPAD
Zhong et al	5	12	SPDC and SBS	Integrated Optics and Polarization	SNSPD

While these experiments (and others not listed), are diverse in their individual details, they all share the main outline of generating several indistinguishable photons, undergoing a unitary transformation within a guided-wave architecture, and being detected by a photon detector. For the sake of brevity in this paper, we will not go into the specifics of how to produce more than two identical photons or how to implement the unitary via guided-wave architecture, although helpful information on those topics is found in Daniel J. Brod [2019]. Instead, we will focus on the limitations, which are shared by all of the above experiments, and are necessary to overcome if we hope to advance the field. First of all, in using the SPDC process, emission of multiple pairs is exponentially less efficient than the generation of a single pair, so this scheme does not easily allow scaling up the number of input photons beyond the few that we see currently implemented. One way that experimentalists have thought to deal with this is via scattershot boson sampling, a variant of the main problem in which *k* SPDC sources (with $k > m$) are connected to different input ports of the linear interferometer, and thus the expected

number of single photons per SPDC pulse is scaled up by a factor $\binom{k}{m}$, which is an exponentially higher generation rate than a fixed input version of the problem as m increases. On another hand, researchers could use quantum dot sources which generate little trains of indistinguishable single photons (with high efficiency).

Another important problem that must be solved is the relevant insertion loss of the circuit which implements the linear interferometer (which is generally on the order of 30% for a resonable device with 10-20 modes) (Daniel J. Brod [2019]). This is an important problem because the success rate of an experiment here scales as the losses to the power of m, so as we increase our number of photons, we will experience greater loss. One such solution, proposed by Wang et al. [2018], uses interferometers that operate in free space and consist of micro-optical assemblies. These experiments are conducive to triangular and rectangular geometries for interferometric layouts, and they allow arbitrary linear-optical transformations which have high optimal transmission (> 99%) and phase-stability. The downside of this approach is that it is unclear whether or not they could realize truly Haar-random linear-optical transformations. Alternately, experiments could try to use temporal modes rather than spatial ones, in which the modes are a train of time bins separated by a time interval τ. This scheme could be conducive to beam splitter networks allowing for arbitrary interferometers.

Efficiently verifying boson sampling is important and possibly out of reach due to how complex it is for classical computing. However, if we are to use boson sampling to prove quantum advantage, there must be a certification that the device is actually performing the process correctly, rather than just making up data or drawing it from other distributions (those which could be classically sampled in an efficient way but resemble the correct one). With only a few photons, verification is fairly attainable as the distribution can be evaluated classically; however, that approach cannot be scaled. One thought for how to approach verifying a system with more photons is to mock up distributions which would have plausible error models for the device operations; then those could could be applied to the models for testing mock data.

If we can rectify the issues above, we will be much closer to achieving a regime of quantum advantage. Several studies have focused on individual aspects of the problem, not excluding defining a limit on what classical computers can simulate. Similarly, technological advances have, over the course of the last decade, created opportunity for experimentalists to physically construct more advanced and larger systems, and new tools are being developed to make the boson sampling problem easier to solve.

6 Discussion & Conclusion

We have studied in depth a handful of different schemes for proving quantum supremacy, all of which use a restricted version of full quantum computation. This of course is no coincidence, as given the current infeasibility of building a universal quantum computer, our best bet in the near-term is to focus on simpler models.

Experimentally demonstrating this advantage has widely varying requirements across the paradigms we have shown, ranging from thousands of qubits for TCF, to only dozens for Boson sampling. Additionally, the operations and interactions among qubits in the schemes also is quite diverse. Given that this is only a small sample of the current work in this field, it seems likely that a fruitful physical experiment may occur in the near future.

On the other hand, we have different theoretical assumptions underlying the demonstrations of quantum supremacy in each of these models. For instance, the work of Kahanamoku-Meyer and others shows that trapdoor claw-free functions are sufficient, while Boson sampling requires only a separation of the polynomial hierarchy at the third level. This is quite encouraging, since it means that if only one of many possible conjectures is proven correct, we would have a definitive proof of some additional quantum power.

References

S. Aaronson. The limits of quantum computers. *Scientific American, Inc.*, March 2008. URL `https://www.cs.virginia.edu/~robins/The_Limits_of_Quantum_Computers.pdf`.

S. Aaronson. The computational complexity of linear optics, 2010. URL https://scottaaronson.blog/?p=473.

S. Aaronson and A. Arkhipov. The computational complexity of linear optics, 2010. URL https://arxiv.org/abs/1011.3245.

S. Aaronson, A. Bouland, G. Kuperberg, and S. Mehraban. The computational complexity of ball permutations, 2016. URL https://arxiv.org/abs/1610.06646.

F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574 (7779):505–510, 2019. doi: 10.1038/s41586-019-1666-5. URL https://doi.org/10.1038/s41586-019-1666-5.

A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995. doi: 10.1103/PhysRevA.52.3457. URL https://link.aps.org/doi/10.1103/PhysRevA.52.3457.

J. Carolan, J. D. A. Meinecke, P. J. Shadbolt, N. J. Russell, N. Ismail, K. Wörhoff, T. Rudolph, M. G. Thompson, J. L. O'Brien, J. C. F. Matthews, and A. Laing. On the experimental verification of quantum complexity in linear optics. *Nature Photonics*, 8(8):621–626, jul 2014. doi: 10.1038/nphoton.2014.152. URL https://doi.org/10.1038%2Fnphoton.2014.152.

A. C. R. O. N. S. F. S. Daniel J. Brod, Ernesto F. Galvão. Photonic implementation of boson sampling: a review. *Advanced Photonics*, May 2019. doi: 10.1117/1.AP.1.3.034001. URL https://doi.org/10.1117/1.AP.1.3.034001.

A. Datta, A. Shaji, and C. M. Caves. Quantum discord and the power of one qubit. *Physical Review Letters*, 100(5), feb 2008. doi: 10.1103/physrevlett.100.050502. URL https://doi.org/10.1103%2Fphysrevlett.100.050502.

B. T. Gard, K. R. Motes, J. P. Olson, P. P. Rohde, and J. P. Dowling. An introduction to boson-sampling. In *From Atomic to Mesoscale*, pages 167–192. WORLD SCIENTIFIC, jun 2015. doi: 10.1142/9789814678704_0008. URL https://doi.org/10.1142%2F9789814678704_0008.

G. D. Kahanamoku-Meyer. Forging quantum data: classically defeating an iqp-based quantum test, 2019. URL https://arxiv.org/abs/1912.05547.

G. D. Kahanamoku-Meyer, S. Choi, U. V. Vazirani, and N. Y. Yao. Classically-verifiable quantum advantage from a computational bell test, 2021. URL https://arxiv.org/abs/2104.00687.

E. Knill and R. Laflamme. Power of one bit of quantum information. *Physical Review Letters*, 81(25):5672–5675, dec 1998. doi: 10.1103/physrevlett.81.5672. URL https://doi.org/10.1103%2Fphysrevlett.81.5672.

E. Knill, R. Laflamme, and G. Milburn. Efficient linear optics quantum computation, 2000. URL https://arxiv.org/abs/quant-ph/0006088.

G. Passante, O. Moussa, C. A. Ryan, and R. Laflamme. Experimental approximation of the jones polynomial with one quantum bit. *Physical Review Letters*, 103(25), dec 2009. doi: 10.1103/physrevlett.103.250501. URL https://doi.org/10.1103%2Fphysrevlett.103.250501.

D. Shepherd and M. J. Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1413–1439, feb 2009. doi: 10.1098/rspa.2008.0443. URL https://doi.org/10.1098%2Frspa.2008.0443.

P. W. Shor and S. P. Jordan. Estimating jones polynomials is a complete problem for one clean qubit. 2007. doi: 10.48550/ARXIV.0707.2831. URL https://arxiv.org/abs/0707.2831.

H. Wang, W. Li, X. Jiang, Y.-M. He, Y.-H. Li, X. Ding, M.-C. Chen, J. Qin, C.-Z. Peng, C. Schneider, M. Kamp, W.-J. Zhang, H. Li, L.-X. You, Z. Wang, J. Dowling, S. Höfling, C.-Y. Lu, and J.-W. Pan. Toward scalable boson sampling with photon loss. *Physical Review Letters*, 120(23), jun 2018. doi: 10.1103/physrevlett.120.230502. URL https://doi.org/10.1103%2Fphysrevlett.120.230502.