On the Complexity of Random Quantum Computations and the Jones Polynomial

Ryan L. Mann[1,*] and Michael J. Bremner[1,2]

[1] *Centre for Quantum Software and Information, Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia*

[2] *Centre for Quantum Computation and Communication Technology, Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia*

There is a natural relationship between Jones polynomials and quantum computation. We use this relationship to show that the complexity of evaluating relative-error approximations of Jones polynomials can be used to bound the classical complexity of approximately simulating random quantum computations. We prove that random quantum computations cannot be classically simulated up to a constant total variation distance, under the assumption that (1) the Polynomial Hierarchy does not collapse and (2) the average-case complexity of relative-error approximations of the Jones polynomial matches the worst-case complexity over a constant fraction of random links. Our results provide a straightforward relationship between the approximation of Jones polynomials and the complexity of random quantum computations.

I. INTRODUCTION

The complexity of quantum computation is completely determined by the complexity of quantum circuit amplitudes. These amplitudes can encode the solution to computationally hard problems, such as Jones polynomials [1], Tutte polynomials [2], and matrix permanents [3, 4]. Unfortunately, quantum mechanics does not provide us with a method for directly measuring these amplitudes or their corresponding probabilities. We must instead infer approximations to them via repeated computations.

There is often a significant difference between the complexity of an exact evaluation of a function and an approximation to it. For example, in the case of the ferromagnetic Ising model, an exact evaluation of its partition function is #P-hard. However, a relative-error approximation can be achieved with a classical computer in polynomial time [5]. Another interesting example is the Jones polynomial. Exactly computing the Jones polynomial is #P-hard [6]. However, unlike the ferromagnetic Ising model, the Jones polynomial retains this complexity for relative-error approximations [7]. It is known that, for the same class of Jones polynomials, computing additive-error approximations is BQP-hard [8]. Therefore, it seems unlikely that quantum computers can produce relative-error approximations of Jones polynomials in polynomial time.

We show that the complexity of evaluating relative-error approximations of Jones polynomials can be used to bound the classical complexity of approximately simulating random quantum computations. Under the assumption that (1) the Polynomial Hierarchy (PH) does not collapse [9] and (2) the average-case complexity of relative-error approximations of the Jones polynomial matches the worst-case complexity over a constant fraction of random links (Conjecture 1), we prove that random quantum computations cannot be classically simulated up to a constant total variation distance (Theorem 10). This argument follows as a natural extension to those given for Instantaneous Quantum Polynomial-time (IQP) circuits [10, 11] and for other classes of random quantum circuits [12], when combined with results on approximate designs [13, 14]. Our results provide a straightforward relationship between the approximation of Jones polynomials and the complexity of random quantum computations.

Many quantum circuit classes can be associated with functions that are #P-hard to evaluate up to a relative error. This feature has been used to construct arguments in favour of a separation between the power of classical and quantum computation (for a review on this topic see Ref. [15] and Ref. [16]). While we do not believe that quantum computers can exactly evaluate such functions, they play a vital role in defining the complexity of sampling from the output probability distribution of quantum circuits. Terhal and DiVincenzo [17] first used this feature to bound the capability of classical computers to simulate constant-depth quantum computations. This was later extended to the problem of sampling from linear optical networks [18] and IQP circuits [10].

Aaronson and Arkhipov [18] proved an important relationship between the complexity of approximate sampling and the average-case complexity of relative-error approximations to counting problems. They showed that the complexity of evaluating relative-error approximations to matrix permanents can be used to bound the classical complexity of sampling from random linear optical networks up to a constant total variation distance — a notion of approximation that is realistic for quantum computation. They conjecture that (1) the average-case complexity of the permanent of Gaussian matrices is #P-hard and (2) the permanent of Gaussian matrices satisfies a certain anti-concentration bound. Assuming

* mail@ryanmann.org; http://www.ryanmann.org

that these conjectures are true, they show that the existence of an efficient classical algorithm which can approximately sample from these networks would imply the collapse of the Polynomial Hierarchy [18]. A similar result was proven for IQP circuits [11] — extending this argument to the quantum circuit model under a different average-case complexity conjecture, where the equivalent anti-concentration conjecture could be proven.

These sampling problems are not just a good candidate for proving a separation between classical and quantum computation, but also for providing experimental benchmarks [12, 16]. This has motivated the study of many other sampling problems. Each of these conjecture the equivalence of the average-case and worst-case complexity of relative-error approximations of a given function. These include: (1) the permanent of Gaussian matrices [18], (2) the gap of degree-three polynomials over $\mathbb{F}_2$ [11, 19], (3) output probabilities of conjugated Clifford circuits [20], and (4) complex-temperature Ising model partition functions over dense [11], sparse [21], and three-dimensional models [12, 22, 23].

These average-case complexity conjectures are each associated with a class of quantum circuits. These quantum circuits are not thought to be universal for quantum computation, with the exception of the three-dimensional Ising model case, but nonetheless become universal under post-selection. Understanding the distinctions between these conjectures is essential for understanding the relationship between these classes of quantum circuits. However, resolving such conjectures would require non-relativising techniques [24]. We therefore expect this to be a hard open problem.

We consider the problem of sampling from random quantum computations that are distributed according to an approximate unitary $(t \geq 2)$-design. We observe that these approximate unitary designs produce output probability distributions that satisfy an anti-concentration bound. This bound is used to prove that if there exists an efficient classical algorithm which can sample from these distributions up to a constant total variation distance, then Stockmeyer's Counting Theorem (Theorem 12) can be used to produce relative-error approximations to a constant fraction of their output probabilities (Theorem 2). This same observation has been used to establish arguments for the complexity of random quantum circuits [12, 23] and conjugated Clifford circuits [20].

We define a natural model of random links via the braid group. A random braid is generated by applying generators of the braid group uniformly at random. A random link is then the plat closure of a random braid. We show that the output probability amplitudes of random quantum computations are proportional to the Jones polynomial of a random link. Furthermore, we show that in the k^{th} path model representation with $k = 5$ or $k \geq 7$, random braids on $2n$ strands of length $\Omega[n(n + \log(1/\epsilon))]$ form an ϵ-approximate unitary 2-design (Corollary 9). This leads us to conjecture that it is #P-hard to approximate the Jones polynomial, up to a relative error, on at least a constant fraction of random links (Conjecture 1). This provides a natural conjecture for bounding the classical complexity of simulating random quantum computations.

This paper is structured as follows. In Section II, we provide an introduction to random quantum computations and approximate unitary designs. We then state our result on the classical simulation of random quantum computations. In Section III, we briefly introduce the theory of knots, braids, and the Jones polynomial. We review the relationship between Jones polynomials and quantum computing in Section IV. In Section V, we relate the complexity of random quantum computations to the complexity of approximating the Jones polynomial of random links. Finally, we conclude in Section VI with some remarks and open problems.

II. RANDOM QUANTUM COMPUTATIONS

A *random quantum computation* is the action of (1) preparing an initial state, (2) applying a randomly chosen unitary matrix, and (3) measuring in the computational basis. This is equivalent to sampling from a probability distribution $\mathcal{D}_U$, where U is a randomly chosen unitary matrix.

Definition 1 ($\mathcal{D}_U$)**.** For a $d \times d$ unitary matrix U, we define $\mathcal{D}_U$ to be the probability distribution over integers $x \in [d]$, given by

$$\mathbf{Pr}[x] := |\langle x| U |0\rangle|^2 .$$

It is natural to consider unitary matrices drawn from the uniform distribution. The uniform distribution over the unitary group $\mathrm{U}(d)$ is defined by the *Haar measure*, which is the unique translation-invariant measure on the group. Unfortunately, random unitary matrices drawn from the Haar measure cannot be implemented efficiently by a quantum computer as they typically require an exponential number of gates [25].

For our purposes, it is important that the random quantum computations can be implemented efficiently. We achieve this by weakening the requirement that the unitary matrices are drawn from the Haar measure. Instead, we require only that the unitary matrices are drawn from a distribution that is close to the Haar measure.

A *unitary t-design* is a distribution over a finite set of unitary matrices which imitates the properties of the Haar measure up to the t^{th} moment. For convenience, let $\mathrm{Hom}_{(t,t)}(\mathrm{U}(d))$ be the set of polynomials homogeneous of degree t in the matrix elements of U and homogeneous of degree t in the matrix elements of U^*.

Definition 2 (Unitary t-design [26])**.** A distribution $\mathcal{D} = \{p_i, U_i\}$ over unitary matrices in dimension d is a unitary t-design if, for any polynomial

$f \in \mathrm{Hom}_{(t,t)}(\mathrm{U}(d))$,

$$\sum_{U_i \in \mathcal{D}} p_i f(U_i) = \int_{\mathrm{U}(d)} f(U) dU.$$

Definition 3 (ϵ-approximate unitary t-design)**.** A distribution $\mathcal{D} = \{p_i, U_i\}$ over unitary matrices in dimension d is an ϵ-approximate unitary t-design if, for any polynomial $f \in \mathrm{Hom}_{(t,t)}(\mathrm{U}(d))$,

$$(1-\epsilon)\int_{\mathrm{U}(d)} f(U)dU \leq \sum_{U_i \in \mathcal{D}} p_i f(U_i) \leq (1+\epsilon)\int_{\mathrm{U}(d)} f(U)dU.$$

Brandao, Harrow, and Horodecki [14] showed that G-local random quantum circuits acting on n qudits composed of polynomially many gates form an approximate unitary poly(n)-design. Here, $G = \{g_i\}_{i=1}^m$ is a universal set of gates containing inverses with each $g_i \in \mathrm{U}(d^2)$ composed of algebraic entries.

Definition 4 (G-local random quantum circuit)**.** At each time step, two indices, i and j, are chosen uniformly at random from $[m]$ and $[n-1]$, respectively. The gate g_i is then applied to the two neighbouring qudits j and $j+1$.

Theorem 1 (Brandao, Harrow, and Horodecki [14])**.** *Fix $d \geq 2$. Let $G = \{g_i\}_{i=1}^m$ be a universal set gates containing inverses with each $g_i \in \mathrm{U}(d^2)$ composed of algebraic entries. There exists a constant $\lambda = \lambda(G) > 0$ such that G-local random quantum circuits of length*

$$\lambda n \left\lceil \log_d(4t) \right\rceil^2 t^5 t^{3.1/\log(d)} \left[nt \log(d) + \log(1/\epsilon) \right]$$

form an ϵ-approximate unitary t-design.

We shall, therefore, restrict our attention to random quantum computations where the unitary matrices are drawn from an ϵ-approximate unitary $(t \geq 2)$-design. We are interested in a classical simulation of random quantum computations, for which we have the following result:

Theorem 2. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 2)$-design and let $\mathcal{D}_U$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any U, samples from a probability distribution $\mathcal{D}'$, such that $||\mathcal{D}' - \mathcal{D}_U||_1 \leq \mu$. Then, for any γ such that $0 < \gamma < 1 - \epsilon$, there is an $\mathrm{FBPP}^{\mathrm{NP}^C}$ algorithm which approximates $|\langle 0| U |0\rangle|^2$ up to a relative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of matrices.*

We prove Theorem 2 and several supporting lemmas in Appendix A. Theorem 2 tells us that, if there exists an efficient classical algorithm which can approximately sample from any random quantum computation, then, there is an $\mathrm{FBPP}^{\mathrm{NP}}$ algorithm which can approximate $|\langle 0| U |0\rangle|^2$ up to a relative error for a fraction of matrices U. Suppose that this algorithm solves a #P-hard problem, then, by Toda's Theorem [27], the Polynomial Hierarchy collapses to its third level.

Theorem 3 (Toda [27])**.**

$$\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}.$$

In Section V, we show that $|\langle 0| U |0\rangle|^2$ is proportional to the Jones polynomial of a random link, which is known to be #P-hard to approximate up to a relative error in the worst case [7]. We conjecture that this remains true in the average case.

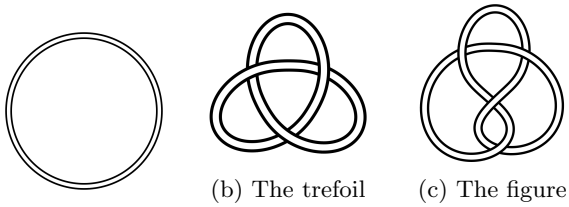
III. KNOTS, BRAIDS, AND THE JONES POLYNOMIAL

We now briefly introduce the theory of knots, braids, and the Jones polynomial.

Definition 5 (Knot)**.** A knot K is subset of points in $\mathbb{R}^3$ that is homeomorphic to a circle.

Informally, a knot is a tangled strand of string with the open ends closed to form a loop. Much like the everyday knots that we use when we tie our shoelaces, ties, and so on — mathematical knots are exactly that, except that the open ends are fused together.

The most simple knot you can think of is the *unknot*, also called the *trivial knot*, which is a closed loop without a knot (Fig. 1a). Other examples of knots include the *trefoil knot* (Fig. 1b), and the *figure eight knot* (Fig. 1c).



(a) The unknot. (b) The trefoil knot. (c) The figure eight knot.

FIG. 1: Examples of basic knots.

We have seen how a knot is an embedding of a circle in $\mathbb{R}^3$. We can now generalise this idea by considering an embedding of multiple circles in $\mathbb{R}^3$.

Definition 6 (Link)**.** A link L is a finite disjoint union of knots $L = \bigcup_i K_i$. Each knot K_i in the union is called a *component* of the link.

Definition 7 (Oriented link)**.** An oriented link is a link in which each component is assigned an orientation.

We can now see that a knot is a link of only one component. The generalisation of the unknot to a link on n components is called the *unlink*, which is a collection of n unknots that are not interlinked. An example of a

slightly more interesting link is the *Borromean rings* link (Fig. 2), which has the property that removing any single component of the link gives the two component unlink.

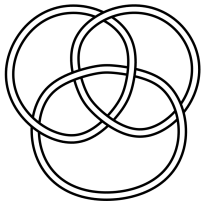


FIG. 2: The Borromean rings link.

A important problem in knot theory is the *link recognition problem* — given two links are they the same? To answer this, we must first ask, what does it mean for two links to be the same?

Definition 8 (Link equivalence)**.** Two links L_1 and L_2 are said to be equivalent if there exists a orientation-preserving homeomorphism $f : \mathbb{R}^3 \to \mathbb{R}^3$ so that $f(L_1) = L_2$.

Essentially, two links are equivalent if they can be deformed into one another. We can prove that two links are equivalent by producing a set of instructions that will deform one link into the other. However, proving that two links are not equivalent is much more difficult, as we would need to prove that no set of instructions exist.

Link invariants are an important concept in knot theory as they allow us to study the link recognition problem.

Definition 9 (Link invariant)**.** A link invariant is a function from the set of links to some other set, such that the output of the function depends only on the equivalence class of the link.

Definition 10 (Jones polynomial [28])**.** The Jones polynomial $V_L(\omega)$ is a link invariant, which assigns to each oriented link a Laurent polynomial in the variable $\omega^{1/2}$.

The Jones polynomial is characterised by the *skein relation* and the normalisation that the Jones polynomial of the unknot $V_{\bigcirc}(\omega) = 1$.

Definition 11 (Skein relation)**.** Given three links L_-, L_0, and L_+ that are identical, except for a local region where they differ according to Fig. 3, then the following skein relation holds

$$(\omega^{1/2} - \omega^{-1/2})V_{L_0}(\omega) = \omega^{-1}V_{L_+}(\omega) - \omega V_{L_-}(\omega).$$

The skein relation is sufficient for a recursive computation of the Jones polynomial of a link. It follows that the Jones polynomial of a link can be computed in time exponential in the number of crossings. A classic result of Jaeger, Vertigan, and Welsh [6] states that exactly computing the Jones polynomial $V_L(\omega)$ of a link is #P-hard except when ω is one of a few special points. Bordewich et al. [29] showed that it is BQP-hard to approximate the Jones polynomial up to an additive error. Kuperberg [7] proved that it remains #P-hard to approximate the Jones polynomial up to a relative error.

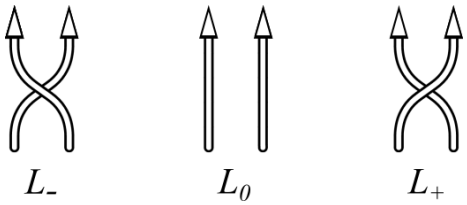


FIG. 3: Diagrams for the skein relation.

Theorem 4 (Jaeger, Vertigan, and Welsh [6])**.** *Evaluating the Jones polynomial $V_L(\omega)$ of a link is* #P-hard *except when $\omega = \pm\exp(2\pi i/k)$ with $k \in \{1, 2, 3, 4, 6\}$ when it can be evaluated in polynomial time.*

We now introduce the theory of braids, which provides us with a convenient way to represent any link.

Definition 12 (Braid)**.** Let

$$A = \{(x, 0, 0) \mid x \in \mathbb{Z}^+, x \le n\},$$
$$B = \{(x, 0, 1) \mid x \in \mathbb{Z}^+, x \le n\}.$$

Then, an n-strand braid is a collection of non-intersecting smooth paths in $\mathbb{R}^3$ connecting the points in A to the points in B.

Informally, a braid is a collection of strands of string that may cross over and under each other, and must always move from left to right. An example of a braid is given in Fig. 4.

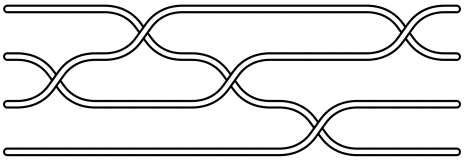


FIG. 4: An example of a braid on 4 strands.

The set of all braids on n strands form an infinite group B_n, generated by the $n-1$ generators $\{\sigma_i\}$ and their inverses $\{\sigma_i^{-1}\}$. The generator σ_i crosses the i^{th} strand over the $(i+1)^{\text{th}}$ strand and its inverse σ_i^{-1} crosses the i^{th} strand under the $(i+1)^{\text{th}}$ strand.

Definition 13 (Braid group)**.** The braid group on n strands B_n is the group given by the Artin presentation

$$\left\langle \{\sigma_i\}_{i=1}^{n} \middle| \begin{array}{ll} \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} & \text{for } 1 \le i \le n-2 \\ \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for } |i-j| \ge 2 \end{array} \right\rangle.$$

Each braid can be described by a *braid word*.

Definition 14 (Braid word)**.** A braid word is word on the set of generators $\{\sigma_i\}$ and their inverses $\{\sigma_i^{-1}\}$. The *length* of a braid word is the number of characters in the word.

We can connect the endpoints of any braid in a number of ways to form a link. For a braid with an even number of strands a natural way to do this is by the *plat closure*.

Definition 15 (Plat closure)**.** The plat closure of a $2n$-strand braid $b \in B_{2n}$ is the link formed by connecting pairs of adjacent strands on the left and the right of the braid. The link that is formed by the plat closure of the braid is often denoted b^{pl}.

Alexander [30] showed that we can generate all possible links this way. We can, therefore, describe any link as the closure of a braid given by its braid word.

Theorem 5 (Alexander [30])**.** *Every link can be represented by the closure of some braid.*

IV. THE JONES POLYNOMIAL AND QUANTUM COMPUTING

Freedman, Kitaev, and Wang [31] established a quantum algorithm for additively approximating the Jones polynomial at any principle root of unity in polynomial time. This algorithm was later formalised by Aharonov, Jones, and Landau [1]. Freedman, Larsen, and Wang [32] proved that when $\omega = \exp(2\pi i/k)$ is a *principle non-lattice root of unity*, i.e. $k = 5$ or $k \geq 7$, the problem of additively approximating the Jones polynomial is universal for quantum computation. Aharonov and Arad [8] extended this result to values of k that grow polynomially with the number of strands and crossings.

Theorem 6 (Aharonov and Arad [8])**.** *Let ω be a principle non-lattice root of unity, and let $b \in B_{2n}$ be a braid. Then, the problem of additively approximating the Jones polynomial $V_{b^{pl}}(\omega)$ to within the same accuracy as the Aharonov-Jones-Landau algorithm* [1] *is* BQP-hard.

The Aharonov-Jones-Landau algorithm is based on the *path model representation of the braid group* [28, 33], which is unitary when $\omega = \exp(2\pi i/k)$ is a principle root of unity. For an integer k, the k^{th} path model representation of the braid group B_{2n} is defined on the vector space spanned by walks of length $2n$, on a $k-1$ vertex path graph G_k, which start and finish on the first vertex.

To calculate the dimension of this vector space it is sufficient to count the number of walks of length $2n$ on the graph G_k. From a combinatorial perspective, the walks on the graph G_k can be seen as *Dyck paths* of length $2n$, which never go above a height $k-2$. It is well known that the number of Dyck paths of length $2n$ is the n^{th} *Catalan number*, which provides an upperbound for the dimension of the vector space.

Definition 16 (Catalan number)**.** The n^{th} Catalan number is defined by

$$C_n := \frac{1}{(n+1)}\binom{2n}{n}.$$

Claim 7. *For $n \geq 1$,*

$$C_n < 4^n.$$

Proof. The claim follows directly from Stirling's approximation for factorials. ∎

In this representation, each braid $b \in B_{2n}$ is mapped to a unitary matrix $\rho_k(b)$ composed of algebraic entries. These unitary matrices have the property that the expectation value $\langle 0|\, \rho_k(b)\, |0\rangle$ is proportional, up to an efficiently computable factor, to the Jones polynomial $V_{b^{pl}}(\omega)$ of the plat closure of b. Aharonov, Jones, and Landau [1] showed that such representations can be implemented efficiently on a quantum computer.

In their construction, the unitary representation of each generator $\rho_k(\sigma_i^{\pm})$ of the braid group B_{2n} acts on a subspace of the Hilbert space of qudits. The Solovay-Kitaev theorem [34] guarantees that these unitary matrices can be implemented efficiently. An entire braid $b \in B_{2n}$ is implemented efficiently by applying the corresponding unitary matrix of each generator in the order of the braid word of b.

V. RANDOM QUANTUM COMPUTATIONS AND RANDOM LINKS

We now relate random quantum computations and the Jones polynomial of random links. We define a *random link* to be the plat closure of a *random braid*.

Definition 17 (Random braid)**.** A random braid on $2n$ strands is generated by uniformly at random choosing generators from the set $\{\sigma_i^{\pm}\}_{i=1}^{2n-1}$.

Definition 18 (Random link)**.** A random link is generated by the plat closure of a random braid.

In the k^{th} path model representation the generators of the braid group $\{\sigma_i^{\pm}\}$ are mapped to unitary matrices $\{\rho_k(\sigma_i^{\pm})\}$. In this representation, a random braid is equivalent to a product of random matrices chosen uniformly at random from the set $\{\rho_k(\sigma_i^{\pm})\}$. Since each $\rho_k(\sigma_i^{\pm})$ acts on a subspace of the Hilbert space of qudits, a random braid is equivalent to a G-local random quantum circuit, with the number of strands proportional to the number of qudits. When $k = 5$ or $k \geq 7$ these gates are universal for quantum computation.

Theorem 8. *In the k^{th} path model representation with $k = 5$ or $k \geq 7$, there exists a constant $\lambda > 0$, such that random braids on $2n$ strands of length*

$$\lambda n \left\lceil \log_2(4t) \right\rceil^2 t^5 t^{3.1/\log(2)} \left[t \log\left(C_n\right) + \log(1/\epsilon) \right],$$

form an ϵ-approximate unitary t-design.

Proof. The proof follows from combining Theorem 1 with the fact that the dimension of the vector space in the path model representation is bounded from above by the n^{th} Catalan number and that the local dimension is bounded from below by 2. ■

Corollary 9. *In the k^{th} path model representation with $k = 5$ or $k \geq 7$, there exists a constant $\lambda > 0$, such that random braids on $2n$ strands of length*

$$\lambda n \left[n + \log(1/\epsilon)\right],$$

form an ϵ-approximate unitary 2-design.

Proof. The proof follows from setting $t = 2$ in Theorem 8 and from the upperbound for the n^{th} Catalan number found in Claim 7. ■

We now relate the classical simulation of random quantum computations and the complexity of approximating the Jones polynomial of random links.

Theorem 10. *Fix $0 < \epsilon < 1$. Let $k = 5$ or $k \geq 7$ be an integer, and $\omega = \exp(2\pi i/k)$ its corresponding root of unity. Let $b \in B_{2n}$ be a random braid on $2n$ strands of length $\Omega\left[n(n + \log(1/\epsilon))\right]$. Let $\rho_k(b)$ be the k^{th} path model representation of b, and let $\mathcal{D}_{\rho_k(b)}$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any b, samples from a probability distribution $\mathcal{D}'$, such that $\left|\left|\mathcal{D}' - \mathcal{D}_{\rho(b)}\right|\right|_1 \leq \mu$ and assume that Conjecture 1 holds. Then, there is a BPP^{NP} algorithm for solving any problem in $\text{P}^{\#\text{P}}$ and by Toda's Theorem the Polynomial Hierarchy collapses to its third level.*

Proof. The proof follows from combining Theorem 2, Corollary 9, and Toda's Theorem (Theorem 3). ■

Conjecture 1. *In the notation of Theorem 10. For some $0 < \gamma < 1 - \epsilon$, it is #P-hard to approximate the Jones polynomial $V_{b^{pl}}(\omega)$ up to a relative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of random braids.*

Conjecture 1 is based on the average-case complexity of relative-error approximations of Jones polynomials. It is known that it is #P-hard to approximate the Jones polynomial up to a relative error in the worst case [7]. Therefore, Conjecture 1 states that this worst-case hardness result can be extended to an average-case hardness result.

Assuming that Conjecture 1 holds and the Polynomial Hierarchy does not collapse, Theorem 10 tells us that there is no efficient classical algorithm which can sample from any random quantum computation. This implies that random quantum computations can not be efficiently simulated by a classical computer.

It is worth noting that the 5^{th} path model representation is equivalent to the *Fibonacci representation of the braid group* [35]. Therefore, our results extend to the random braiding of Fibonacci anyons.

VI. CONCLUSION & OUTLOOK

We have provided strong evidence that simulating random quantum computations is intractable for classical computers. Specifically, we have shown that if Conjecture 1 holds and the Polynomial Hierarchy does not collapse, then there is no efficient classical algorithm which can approximately sample from the output probability distribution of random quantum computations.

There are a number of natural problems that remain to be solved. The most obvious of which is to resolve Conjecture 1. Unfortunately, we are unaware of any proof techniques which are capable of extending the worst-case hardness result to an average-case hardness result. Moreover, the results of Aaronson and Chen [24] imply that any proof of this conjecture would require non-relativising techniques.

Another natural problem is whether Corollary 9 can be strengthened to random braids of a shorter length. In Theorem 10, the length of a random braid is determined by the requirement that in the path model representation it is distributed according to an ϵ-approximate unitary 2-design. Therefore, any improvement to this bound yields a stronger version of Theorem 10. It is an open problem whether this bound can be improved.

It would also be interesting to adapt our results to other functions, such as Tutte polynomials [2], Turaev-Viro invariants [36], and matrix permanents [4]. These functions are all known to be #P-hard to compute in the worst case and BQP-hard to approximate up to an additive error.

ACKNOWLEDGEMENTS

We thank Scott Aaronson, Sergio Boixo, Adam Bouland, Gavin Brennen, Aram Harrow, Saeed Mehraban, Ashley Montanaro, Peter Rohde, and Marco Tomamichel for helpful discussions. MJB acknowledges support from the Australian Research Council via the Future Fellowship scheme (grant FT110101044) and as a member of the ARC Centre of Excellence for Quantum Computation and Communication Technology (CQC2T), project number CE170100012.

Appendix A: Proof of Theorem 2

We now prove Theorem 2, which is restated below for convenience. Our proof requires several lemmas which we prove in the remainder of the section.

Theorem 2. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 2)$-design and let $\mathcal{D}_U$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any U, samples from a probability distribution $\mathcal{D}'$, such that $||\mathcal{D}' - \mathcal{D}_U||_1 \leq \mu$. Then, for any*

γ such that $0 < \gamma < 1 - \epsilon$, there is an $\mathrm{FBPP}^{\mathrm{NP}^C}$ algorithm which approximates $|\langle 0| U |0\rangle|^2$ up to a relative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of matrices.

Proof. Lemma 11 tells us that, for any $0 < \delta < 1$, there is an $\mathrm{FBPP}^{\mathrm{NP}^C}$ algorithm, which approximates $|\langle x| U |0\rangle|^2$, up to an additive error

$$O\left[(1+o(1))\frac{\mu(1+\epsilon)}{\delta d} + \frac{|\langle x| U |0\rangle|^2}{\mathrm{poly}(n)}\right],$$

with probability at least $1-\delta$ over the choice of U. Combining this with Lemma 13 and setting $\delta = \frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$, it follows that there is an $\mathrm{FBPP}^{\mathrm{NP}^C}$ algorithm, which approximates $|\langle 0| U |0\rangle|^2$ up to a relative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of matrices U. ∎

We now prove Lemma 11, which relates the simulation of random quantum computations to approximating individual output probabilities. Our proof closely follows that of Lemma 4 from Ref. [11].

Lemma 11. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 1)$-design and let $\mathcal{D}_U$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any U, samples from a probability distribution $\mathcal{D}'$, such that $||\mathcal{D}' - \mathcal{D}_U||_1 \leq \mu$. Then, for any δ such that $0 < \delta < 1$, there is an $\mathrm{FBPP}^{\mathrm{NP}^C}$ algorithm, which approximates $|\langle 0| U |0\rangle|^2$, up to an additive error*

$$O\left[(1+o(1))\frac{\mu(1+\epsilon)}{\delta d} + \frac{|\langle 0| U |0\rangle|^2}{\mathrm{poly}(n)}\right],$$

with probability at least $1-\delta$ over the choice of U.

Proof. Define

$$Q_U := |\langle 0| U |0\rangle|^2, \quad T_U := \Pr[C \text{ outputs } 0 \text{ on input } U].$$

For any U, we can use Stockmeyer's Counting Theorem (Theorem 12) to obtain a relative-error approximation to T_U in $\mathrm{FBPP}^{\mathrm{NP}^C}$,

$$|T_U - T'_U| \leq \frac{T_U}{\mathrm{poly}(n)}.$$

Then,

$$\begin{aligned}
|Q_U - T_U\prime| &\leq |Q_U - T_U| + |T_U - T'_U| \\
&\leq |Q_U - T_U| + \frac{T_U}{\mathrm{poly}(n)} \\
&\leq |Q_U - T_U| + \frac{(Q_U + |Q_U - T_U|)}{\mathrm{poly}(n)} \\
&= |Q_U - T_U|\left(1 + \frac{1}{\mathrm{poly}(n)}\right) + \frac{Q_U}{\mathrm{poly}(n)}.
\end{aligned}$$

As C approximates $\mathcal{D}_U$ up to an l_1 error μ, it follows from Markov's inequality and the approximate design condition (Lemma 15) that, for any $0 < \delta < 1$,

$$\Pr_U\left[|Q_U - T_U| \geq \frac{\mu(1+\epsilon)}{\delta d}\right] \leq \delta.$$

Therefore,

$$|Q_U - T'_U| \leq \frac{\mu(1+\epsilon)}{\delta d}\left(1 + \frac{1}{\mathrm{poly}(n)}\right) + \frac{Q_U}{\mathrm{poly}(n)},$$

with probability at least $1-\delta$ over the choice of U. ∎

The proof of Lemma 11 requires a classic result of Stockmeyer [37], which allows us to approximately count in the Polynomial Hierarchy.

Theorem 12 (Stockmeyer's Counting Theorem [37])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a function, and let $F = \sum_{x \in \{0,1\}^n} f(x)$. Then there exists an $\mathrm{FBPP}^{\mathrm{NP}^f}$ algorithm, which outputs a value α, such that*

$$|\alpha - F| < \Omega\left[\frac{F}{\mathrm{poly}(n)}\right].$$

We now prove that unitary matrices distributed according to an ϵ-approximate unitary $(t \geq 2)$-design satisfy the anti-concentration bounds claimed in Theorem 2. This was proven independently by Hangleiter et al. [23].

Lemma 13. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 2)$-design, then, for any unit vectors $|\alpha\rangle$, $|\beta\rangle$ and a constant $0 \leq \gamma \leq 1 - \epsilon$, the following holds*

$$\Pr_U\left[|\langle \alpha| U |\beta\rangle|^2 > \frac{\gamma}{d}\right] \geq \frac{(1-\epsilon-\gamma)^2}{2(1+\epsilon)}.$$

Proof. The Paley-Zygmund inequality (Lemma 14) tells us that

$$\Pr_Z\left[Z > \frac{\gamma}{d}\right] \geq \left(1 - \frac{\gamma}{d\mathbb{E}[Z]}\right)^2 \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]},$$

for any $0 \leq \gamma \leq d\mathbb{E}[Z]$. Setting $Z = |\langle \alpha| U |\beta\rangle|^2$, it follows from the approximate design condition (Lemma 15), that

$$\begin{aligned}
\Pr_U\left[Z > \frac{\gamma}{d}\right] &\geq \frac{1}{2}\left(1 - \frac{\gamma}{(1-\epsilon)}\right)^2 \frac{(1-\epsilon)^2}{(1+\epsilon)} \frac{(d+1)}{d} \\
&\geq \frac{1}{2}\left(1 - \frac{\gamma}{1-\epsilon}\right)^2 \frac{(1-\epsilon)^2}{1+\epsilon} \\
&= \frac{(1-\epsilon-\gamma)^2}{2(1+\epsilon)},
\end{aligned}$$

for any $0 \leq \gamma \leq 1 - \epsilon$. ∎

The proof of Lemma 13 combines the Paley-Zygmund inequality and the approximate design condition. The Paley-Zygmund inequality bounds the probability that a non-negative random variable is small in terms of its first and second moment.

Lemma 14 (Paley-Zygmund inequality)**.** *If $Z \geq 0$ is a random variable with finite variance, and if $0 \leq \theta \leq 1$, then*

$$\mathop{\mathbf{Pr}}_{Z}[Z > \theta \mathbb{E}[Z]] \geq (1-\theta)^2 \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}.$$

We are interested in bounding the probability that the random variable $Z = |\langle\alpha| U |\beta\rangle|^2$ is small. In the case of an exact unitary $(t \geq 2)$-design the first and second moments of Z match those of the Haar measure. For an ϵ-approximate $(t \geq 2)$-design the approximate design condition bounds the distance of the first and second moments of Z from those of the Haar measure.

Lemma 15 (Approximate design condition [38])**.** *If U is a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary t-design, then, for any unit vectors $|\alpha\rangle$, $|\beta\rangle$ and an integer $k \leq t$,*

$$\frac{(1-\epsilon)}{\binom{k+d-1}{d-1}} \leq \mathbb{E}\left[|\langle\alpha| U |\beta\rangle|^{2k}\right] \leq \frac{(1+\epsilon)}{\binom{k+d-1}{d-1}}.$$

[1] D. Aharonov, V. Jones, and Z. Landau, Algorithmica **55**, 395 (2009).
[2] D. Aharonov, I. Arad, E. Eban, and Z. Landau, arXiv:quant-ph/0702008 (2007).
[3] S. Scheel, arXiv:quant-ph/0406127 (2004).
[4] T. Rudolph, Physical Review A **80**, 054302 (2009).
[5] M. Jerrum and A. Sinclair, SIAM Journal on computing **22**, 1087 (1993).
[6] F. Jaeger, D. L. Vertigan, and D. J. Welsh, in *Mathematical Proceedings of the Cambridge Philosophical Society* (Cambridge Univ Press, 1990), vol. 108, pp. 35–53.
[7] G. Kuperberg, arXiv:0908.0512 (2009).
[8] D. Aharonov and I. Arad, New Journal of Physics **13**, 035019 (2011).
[9] C. H. Papadimitriou, *Computational complexity* (John Wiley and Sons Ltd., 2003).
[10] M. J. Bremner, R. Jozsa, and D. J. Shepherd, in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (The Royal Society, 2010), p. rspa20100301.
[11] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Physical Review Letters **117**, 080501 (2016).
[12] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, J. M. Martinis, and H. Neven, arXiv:1608.00263 (2016).
[13] A. W. Harrow and R. A. Low, Communications in Mathematical Physics **291**, 257 (2009).
[14] F. G. Brandão, A. W. Harrow, and M. Horodecki, Communications in Mathematical Physics **346**, 397 (2016).
[15] A. Lund, M. J. Bremner, and T. Ralph, NPJ Quantum Information **3**, 1 (2017).
[16] A. W. Harrow and A. Montanaro, Nature **549**, 203 (2017).
[17] B. M. Terhal and D. P. DiVincenzo, Quantum Information & Computation **4**, 134 (2004).
[18] S. Aaronson and A. Arkhipov, in *Proceedings of the forty-third annual ACM symposium on Theory of computing* (ACM, 2011), pp. 333–342.
[19] J. Miller, S. Sanders, and A. Miyake, arXiv preprint arXiv:1703.11002 (2017).
[20] A. Bouland, J. F. Fitzsimons, and D. E. Koh, arXiv preprint arXiv:1709.01805 (2017).
[21] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Quantum **1**, 8 (2017).
[22] X. Gao, S.-T. Wang, and L.-M. Duan, Physical Review Letters **118**, 040502 (2017).
[23] D. Hangleiter, J. Bermejo-Vega, M. Schwarz, and J. Eisert, arXiv:1706.03786 (2017).
[24] S. Aaronson and L. Chen, arXiv preprint arXiv:1612.05903 (2016).
[25] E. Knill, arXiv:quant-ph/9508006 (1995).
[26] A. Roy and A. J. Scott, Designs, codes and cryptography **53**, 13 (2009).
[27] S. Toda, SIAM Journal on Computing **20**, 865 (1991).
[28] V. F. Jones, Bulletin of the American Mathematical Society **12**, 103 (1985).
[29] M. Bordewich, M. Freedman, L. Lovász, and D. Welsh, Combinatorics, Probability and Computing **14**, 737 (2005).
[30] J. W. Alexander, Proceedings of the National Academy of Sciences **9**, 93 (1923).
[31] M. H. Freedman, A. Kitaev, and Z. Wang, Communications in Mathematical Physics **227**, 587 (2002).
[32] M. H. Freedman, M. Larsen, and Z. Wang, Communications in Mathematical Physics **227**, 605 (2002).
[33] V. F. Jones, Geometric methods in operator algebras (Kyoto, 1983) **123**, 242 (1983).
[34] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and quantum computation*, vol. 47 (American Mathematical Society Providence, 2002).
[35] P. W. Shor and S. P. Jordan, Quantum Information & Computation **8**, 681 (2008).
[36] G. Alagic, S. P. Jordan, R. König, and B. W. Reichardt, Physical Review A **82**, 040302 (2010).
[37] L. Stockmeyer, SIAM Journal on Computing **14**, 849 (1985).
[38] F. G. Brandão and M. Horodecki, Quantum Information and Computation **13**, 901 (2013).