Approximating the Turaev-Viro Invariant of Mapping Tori is Complete for One Clean Qubit

Stephen P. Jordan[1] and Gorjan Alagic[2(✉)]

[1] Institute for Quantum Information, California Institute of Technology,
Pasadena, US
`sjordan@caltech.edu`

[2] Institute for Quantum Computing, University of Waterloo, Waterloo, Canada
`galagic@iqc.ca`

Abstract. The Turaev-Viro invariants are scalar topological invariants of three-dimensional manifolds. Here we show that the problem of estimating the Fibonacci version of the Turaev-Viro invariant of a mapping torus is a complete problem for the one clean qubit complexity class (DQC1). This complements a previous result showing that estimating the Turaev-Viro invariant for arbitrary manifolds presented as Heegaard splittings is a complete problem for the standard quantum computation model (BQP). We also discuss a beautiful analogy between these results and previously known results on the computational complexity of approximating the Jones Polynomial.

1 Introduction

Classifying the power of quantum computers is a fundamental problem in quantum information science. The computational power of a general-purpose quantum computer is identified with the complexity class BQP (bounded-error quantum polynomial time). The famous problems of factoring and discrete logarithm, for instance, are in BQP. An essential ingredient of BQP computation is the ability to initialize a large number of qubits into a specific pure state. In some proposed physical implementations, however, this appears to be an extremely difficult task. In 1998, Knill and Laflamme proposed that exponential speedups over classical computers could still be possible, even if one can only initialize a single qubit into a pure state, with the rest of the qubits in the maximally mixed state [17]. The complexity class thus defined is called DQC1 (deterministic quantum computation with one clean qubit), or simply "the one clean qubit class." This class contains several problems for which no efficient classical algorithms are known. The most basic of these is the problem of estimating the trace of a unitary operator. In fact, trace estimation is DQC1-complete: not only is it in DQC1, but any other problem in DQC1 can be reduced to it.

Finding natural BQP-complete and DQC1-complete problems is essential to our understanding of the computational power afforded by quantum computers. Remarkably, BQP-complete problems can be found in areas of mathematics

without *a priori* close connection to quantum computation. In particular, approximating the Jones polynomial, a famous invariant of links, is a BQP-complete problem [1,2,12–14,29]. The input is an element of the braid group, and the output is an estimate of the Jones polynomial of the so-called *plat closure* of the braid. Estimating the Jones polynomial of the so-called *trace closure* of the braid is DQC1-complete [16,25].

Recent work [3,15] showed that (the decision version of) approximating certain invariants of 3-manifolds is a BQP-complete problem. In this formulation, the input is a so-called *Heegaard splitting* of a 3-manifold, specified as an element of the mapping class group. The output is an estimate of the Turaev-Viro invariant of the input manifold. In this article we show that approximating the Turaev-Viro invariant of a 3-manifold specified as a *mapping torus* is a complete problem for the one clean qubit class. In Sect. 5, we use the language of Topological Quantum Field Theories (or TQFTs) to explain the mathematical underpinnings of the relationship between approximating the Jones polynomial of the plat and trace closures, and approximating the Turaev-Viro invariant of Heegaard splittings and mapping tori.

We assume only a basic understanding of topology and quantum computation. Needed concepts in manifold invariants and one clean qubit computation are explained in Sect. 2. Our exposition focuses on the Witten-Reshetikhin-Turaev (or WRT) invariant. This is only a matter of convenience, as it is known that the Turaev-Viro invariant is equal to the absolute square of the WRT invariant [23,26–28].

2   Background

2.1   Two-Manifolds and Three-Manifolds

We begin by setting down a few basic definitions from low-dimensional topology. Recall that an *n-manifold* is a topological space[1] whose every point has a neighborhood that looks like (i.e., is homeomorphic to) an open subset of $\mathbb{R}^n$. Simple examples of one-dimensional manifolds include the line $\mathbb{R}$ and the circle S^1. Simple examples of two-dimensional manifolds include the the plane $\mathbb{R}^2$, the sphere S^2, and the torus $\Sigma_1 = S^1 \times S^1$, which we can visualize as the surface of a donut. More generally, the surface of a donut with g holes is also a two-manifold, which we call the surface of genus g and denote by Σ_g. The genus is a complete invariant of surfaces[2]: homeomorphic surfaces have the same number of handles (invariance), and non-homeomorphic surfaces have a different number of handles (completeness).

The simplest example of a 3-manifold is $\mathbb{R}^3$ itself. A nontrivial example is found by taking the product of Σ_1 with a third circle; the result is the three-dimensional torus $T^3 = S^1 \times S^1 \times S^1$. Given a surface Σ_g, the *cylinder* $\Sigma_g \times [0, 1]$

---

[1] More precisely, a second-countable Hausdorff space.

[2] In this work, we implicitly assume that all surfaces are closed, compact, connected and orientable.
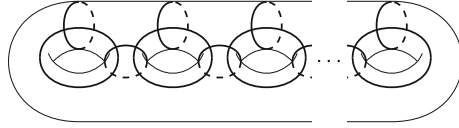
Fig. 1. A Dehn twist is a 2π rotation about a closed curve. The Dehn twists along the $3g - 1$ curves illustrated here constitute a standard set of generators for the mapping class group of the genus g surface.

is a 3-manifold whose boundary consists of two copies of Σ_g (specifically, the bottom $\Sigma_g \times \{0\}$ and the top $\Sigma_g \times \{1\}$.) We can turn the cylinder into a 3-manifold without boundary by choosing a homeomorphism $f : \Sigma_g \to \Sigma_g$ and gluing each point on the top to its image under f on the bottom. The result is the *mapping torus* of f:

$$T_{g,f} = \frac{\Sigma_g \times [0,1]}{(x,1) \sim (f(x),0)} \ .$$

For example, choosing $g = 1$ and f to be the identity map, we see that $T_{1,\mathbb{1}} = T^3$. A useful example of a nontrivial self-homeomorphism of Σ_g is the so-called Dehn twist. To visualize a Dehn twist, imagine cutting the handle of Σ_1 to get a tube, performing a 2π twist on one end of the tube, and then gluing the handle back together. In general, a Dehn twist can be performed around any noncontractible closed curve.

The (homeomorphism class of) the mapping torus $T_{g,f}$ depends only on the isotopy class of f. The orientation-preserving self-homeomorphisms of Σ_g form a group under composition. This group, taken modulo isotopy, is called the mapping class group of Σ_g, and is denoted MCG(g). MCG(g) is generated by the Dehn twists about the $3g - 1$ canonical curves shown in Fig. 1. Any mapping torus $T_{g,f}$ is thus described by a word in the Dehn twist generators of MCG(g).

2.2   The Witten-Reshetikhin-Turaev Invariants

Recall that the genus is an invariant of surfaces because it assigns the same number to homeomorphic surfaces. One can also define invariants of 3-manifolds, although none are as simple and powerful as the genus. In the 1990s, Witten, Reshetikhin, and Turaev discovered a family of 3-manifold invariants arising from their work in Topological Quantum Field Theory. While these invariants can be defined for arbitrary 3-manifolds, we only concern ourselves with the special case of mapping tori, where the definitions are relatively straightforward. Specifically, the Witten-Reshetikhin-Turaev (WRT) invariant of a mapping torus $T_{g,f}$ is equal to the trace of f in a certain projective representation of the mapping class group MCG(g). Note that the WRT function is only a topological invariant up to a phase (see [3]). In general, the WRT invariant is parametrized by a quantum group, such as $SU(N)_k$ or $SO(N)_k$. Although some of our results apply more
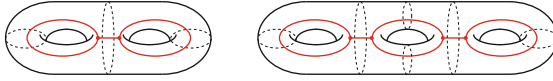
Fig. 2. The dashed lines indicate a set of cuts that decomposes the surface into two three-punctured-spheres ("pants"). Dual to this is a trivalent graph called the "spine," in red. The genus two and genus three cases are shown here.
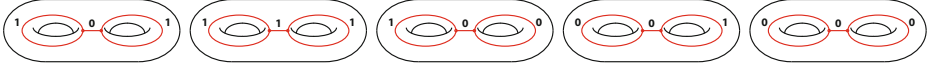

Fig. 3. The Fibonacci model's fusion rules allow five labelings of the standard spine of the genus two surface. This means that the WRT representation of MCG(2) is five-dimensional.

generally, we focus on the case of $SO(3)_3$, sometimes called the Fibonacci model. In this case, the description of the representation is particularly simple, and can be understood with no background in quantum groups.

The Fibonacci representation is defined as follows. Any genus g surface (for $g > 1$) can be cut into three-punctured spheres, resulting in a so-called pants decomposition. Dual to such a decomposition is a trivalent graph on the surface, called a spine. As illustrated in Fig. 2, the spine has one vertex for every pant in the decomposition. Whenever two pants meet at a puncture, the spine has an edge between the corresponding vertices. While a surface admits many spines (and corresponding pants decompositions), we call the one shown in Fig. 2 the *standard spine*. We label the edges of the standard spine by so-called anyon types, with fusion rules enforced at each vertex. For the Fibonacci model, there are only two anyon types: 0 and 1, and only one fusion rule: no vertex can have exactly two edges labeled 0 incident on it. The case $g = 2$ is pictured in Fig. 3. The formal span (over $\mathbb{C}$) of all such labelings associates a finite-dimensional vector space to the surface. Different spines yield different bases for this same space. We can move between these spines (and the corresponding bases) by means of two "moves," the F-move:

$$\left(\text{graphic} \right) = \sum_n F^{ijm}_{kln} \left(\text{graphic} \right)$$

and the S-move:

$$\left(\text{graphic} \right) = \sum_k S^i_{jk} \left(\text{graphic} \right)$$

For the Fibonacci model F_{abc}^{def} is as follows

$$\left.\right\rangle\!\!\!\underset{0}{\times}\!\!\!\left\langle\right. = \frac{2}{1+\sqrt{5}}\ \left.\right\rangle\!\!\!\underset{0}{\times}\!\!\!\left\langle\right. + \sqrt{\frac{2}{1+\sqrt{5}}}\ \left.\right\rangle\!\!\!\underset{1}{\times}\!\!\!\left\langle\right.$$

$$\left.\right\rangle\!\!\!\underset{1}{\times}\!\!\!\left\langle\right. = \sqrt{\frac{2}{1+\sqrt{5}}}\ \left.\right\rangle\!\!\!\underset{0}{\times}\!\!\!\left\langle\right. + \frac{-2}{1+\sqrt{5}}\ \left.\right\rangle\!\!\!\underset{1}{\times}\!\!\!\left\langle\right.$$

with all other values equal to zero or one as dictated by the fusion rules. As one can calculate using the prescription described in [3], S_{jk}^i is given in the Fibonacci model by

$$DS_{00}^0 = 1$$

$$DS_{10}^0 = DS_{01}^0 = \frac{1+\sqrt{5}}{2}$$

$$DS_{11}^0 = 1 + \frac{1+\sqrt{5}}{2}e^{i4\pi/5}$$

$$DS_{11}^1 = \sqrt{\frac{1+\sqrt{5}}{2}}\left(1 - e^{i4\pi/5}\right)$$

with $D = \sqrt{1 + \left(\frac{1+\sqrt{5}}{2}\right)^2}$ and all other values of S_{jk}^i equal to zero by the fusion rules.

The space described above is the underlying vector space for the Fibonacci representation of MCG(g). We define this representation in the basis corresponding to the standard spine. Since the mapping class group is finitely-generated, it suffices to describe the images of the Dehn twist generators. Any such generator is a 2π twist along some canonical curve c from Fig. 1. It is not hard to check that, by applying at most one F-move and one S-move, the standard spine can be adjusted so that c is a cut in the corresponding pants decomposition. In this basis, the Dehn twist about c induces a diagonal linear transformation. To each labeling of the spine corresponds a basis vector, and this basis vector obtains a phase determined by the label on the edge of the spine that intersects c. In the Fibonacci model, edges labeled 0 obtain a phase of 1, and edges labeled 1 obtain a phase of $e^{i3\pi/5}$. In the standard spine basis, the matrix corresponding to the Dehn twist about c is thus simply a product of at most five matrices: at most two of the moves pictured above, followed by a diagonal matrix, followed by the inverse moves to return to the original basis. The WRT invariant of the mapping torus $T_{g,f}$ is now simply the trace of the Fibonacci representation, evaluated at f.

2.3   One Clean Qubit

In some proposed implementations of quantum computers, such as nuclear magnetic resonance (NMR) the most difficult task is initializing qubits into a pure

state. In 1998, Knill and Laflamme proposed that exponential speedups over classical computation might be possible without pure state initialization. To mathematically investigate this possibility, they introduced the one clean qubit model [17]. In this model, one is given an initial state ρ with n qubits in the maximally mixed state, and one qubit in the pure state $|0\rangle$.

$$\rho = |0\rangle \langle 0| \otimes \frac{\mathbb{1}}{2^n}$$

One then applies any quantum circuit of $\mathrm{poly}(n)$ gates to this state, and measures the first qubit in the computational basis. Computational problems are solved by performing polynomially many such experiments, each starting with the initial state ρ, and recording the output statistics. The class of decision problems solvable with bounded probability of error using this procedure is called DQC1.

DQC1 contains several computational problems not known to be solvable in polynomial time on classical computers. Most fundamentally, given a description of a quantum circuit of T gates implementing the unitary transformation U on n qubits, a one clean qubit computer can estimate the normalized trace $\frac{\mathrm{Tr}U}{2^n}$ to within $\pm\epsilon$ in time $O(T/\epsilon^2)$ by means of the circuit shown in Fig. 4. Furthermore, this problem of estimating the trace of a quantum circuit is DQC1-hard [17,24,25]. Efficient one clean qubit algorithms have been discovered for estimating certain quadratically signed weight enumerators [18] and estimating certain Jones [25] and HOMFLY [16] polynomials. A version of the Jones polynomial problem is DQC1-complete [25], and has been demonstrated experimentally with NMR [20,22]. A certain problem of approximating partition functions for quantum systems is also DQC1-hard [6].

In many ways, it is surprising that one clean qubit computers can do any nontrivial computations at all. If all $n+1$ qubits were maximally mixed, the resulting state would be invariant under all unitaries. Furthermore, DQC1 computations involve very little entanglement [7–11,19]. Ambainis *et al.* give an impossibility proof against a certain natural approach to simulating standard quantum computers using one clean qubit computers, and on the other hand show that one clean qubit computers can efficiently simulate classical logarithmic depth (NC1) computations [4].

The DQC1 complexity class is robust against a variety of modifications to the computational model. The class of computational problems solvable in polynomial time with up to logarithmically many clean qubits is the same as
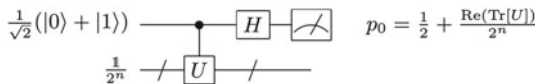

Fig. 4. By repeating this one clean qubit computation, and recording the fraction of 0 outcomes, one estimates the real part of $\mathrm{Tr}[U]/2^n$. Similarly, by initializing the clean qubit to $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$, one obtains $p_0 = \frac{1}{2} + \frac{\mathrm{Im}(\mathrm{Tr}[U])}{2^n}$.

that solvable in polynomial time with one clean qubit [25]. If the clean qubit is not pure, but has $1/\text{poly}(n)$ polarization, the set of efficiently solvable problems also remains DQC1 [17]. As shown in Appendix A, the one clean qudit model on d-dimensional qudits is equivalent in power to the one clean qubit model, for any constant d.

3   Algorithm

In this section we construct an efficient one clean qubit algorithm for approximating the Fibonacci WRT invariant of a mapping torus. Generalizing to other tensor categories such as $\text{SU}(N)_k$ and $\text{SO}(N)_k$ is straightforward. The main idea of the algorithm is, given a word w in the Dehn twist generators of $\text{MCG}(g)$, to find a quantum circuit of $\text{poly}(w, g)$ gates on $\text{poly}(g)$ qubits whose trace is equal to the WRT invariant of the 3-manifold $T_{g,w}$. This trace can then be approximated by means of the circuit in Fig. 4. For this purpose, we encode the allowed labelings of a spine of Σ_g into qubits, and then construct a quantum circuit implementing the Fibonacci representation of $\text{MCG}(g)$ on this encoding. The most obvious encoding would be to directly assign one qubit to store the particle type for each edge of the spine. However, a one clean qubit computer yields the normalized trace over all 2^n bitstrings, of which only an exponentially small fraction represent valid spine labelings in this encoding.

We instead construct a many-to-one map

$$\varphi : \{0, 1\}^{\beta(3g-3)} \to \{\text{valid labelings}\}$$

with $\beta = O(\log|g|)$ such that the preimage of each spine-labeling consists of approximately the same number of bitstrings. That is, $|\varphi^{-1}(x)|$ is approximately independent of x. Thus, the normalized trace of the Fibonacci representation of $w \in \text{MCG}(g)$ acting on the φ-encoded labelings of the spine of Σ_g is approximately equal to $\text{WRT}(T_{g,w})$. We construct φ following a method introduced in [16]. We assign a register of $\beta = O(\log|g|)$ qubits to each edge of the spine. The bitstring contained in register i is interpreted as an integer $0 \le x_i \le 2^\beta - 1$. We then assign a threshold T_i so that $x_i \le T_i$ encodes a zero label on edge i, and $x_i > T_i$ encodes a one label. By carefully choosing the thresholds $T_1, \ldots, T_{3g-3}$ we ensure that $|\varphi^{-1}(x)|$ is approximately independent of x.

Number the edges of the spine from one to $3g-3$, left to right and top to bottom, as illustrated in Fig. 5. Let $s_1, \ldots, s_{3g-3} \in \{0, 1\}^{3g-3}$ be the labels of these edges. The uniform probability distribution over all fusion-consistent labelings
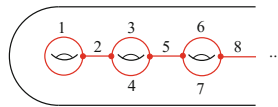

Fig. 5. We number the edges of the standard spine from left to right, with ambiguities resolved by ordering from top to bottom.

of the spine induces a probability distribution $p_g(s_1 \ldots, s_{3g-3})$ over $\{0,1\}^{3g-3}$, with zero probability for strings that violate fusion rules, and uniform probability for the rest. For the genus-g standard spine, we define $p_g(s_i|s_1, \ldots, s_{i-1})$ to be the conditional probability that label i takes the value s_i given that labels 1 through $i-1$ take the values $s_1, \ldots, s_{i-1}$. For a register representing a label s_i we choose the threshold dependent on the values of $s_1, \ldots, s_{i-1}$ according to

$$T_i(g; s_1, \ldots, s_{i-1}) = \left\lceil 2^\beta p_g(0|s_1, \ldots, s_{i-1}) \right\rceil. \tag{1}$$

One can see that this choice ensures that a uniformly selected assignment of bitstrings to the registers yields a uniform distribution over fusion-consistent labelings, up to the errors induced by rounding. Hence, $|\varphi^{-1}(x)|$ is approximately independent of x. More precisely, let

$$\tilde{p}_g(0|s_1, \ldots, s_{i-1}) = T_i(g; s_1, \ldots, s_{i-1})/2^\beta$$
$$\tilde{p}_g(1|s_1, \ldots, s_{i-1}) = 1 - \tilde{p}_g(0|s_1, \ldots, s_{i-1})$$

Thus,

$$\begin{aligned}
|\varphi^{-1}(s_1, \ldots, s_{3g-3})| =& 2^{\beta(3g-3)}\tilde{p}_g(s_{3g-3}|s_1, \ldots, s_{3g-4}) \times \\
& \times \ \tilde{p}_g(s_{3g-4}|s_1, \ldots, s_{3g-5}) \times \ldots \times p(s_1) \\
=& 2^{\beta(3g-3)} \left(p_g(s_{3g-3}|s_1, \ldots, s_{3g-4}) \pm O(2^{-\beta}) \right) \times \\
& \times \ldots \times \left(p_g(s_1) \pm O(2^{-\beta}) \right) \\
=& p_g(s_1, \ldots, s_{3g-3}) \pm O(g2^{-\beta}).
\end{aligned}$$

Thus it suffices to choose $\beta = O(\log g)$. Furthermore, by the locality of the fusion rules, $p_g(s_i|s_1, \ldots, s_{i-1})$ is always independent of $s_1, \ldots, s_{i-3}$. We may thus write

$$\begin{aligned}
p_g(s_i|s_1, \ldots, s_{i-1}) &= p_g(s_i, s_{i-1}, s_{i-2}; i) \\
T_i(g; s_1, \ldots, s_{i-1}) &= T_i(g; s_i, s_{i-1}, s_{i-2}).
\end{aligned} \tag{2}$$

As illustrated in Fig. 6, the Fibonacci representation of a Dehn twist from the standard generating set is a unitary transformation acting on at most five spine labels. Because the encoding φ is many-to-one, the unitary transformation on these spine labels does not uniquely define a unitary operation on the bitstrings encoding them. We say that a pair of spine-labelings is *connected* if the Fibonacci representation of a Dehn twist from the standard set of generators has a nonzero matrix element between them. By choosing a bijection $b_{x,y}$ between the encodings of each pair of connected spin-labelings we define a unitary transformation on the encodings: if the matrix element between labeling x and y is $\rho_{x,y}$ then,

$$U_{i,j} = \begin{cases} \rho_{x,y} & \text{if } \varphi(i) = x, \varphi(j) = y, \text{and } b_{x,y}(i) = j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

is a corresponding unitary representation on the encodings. Our choice of bijections does not matter. We may for concreteness match bitstrings by lexicographic
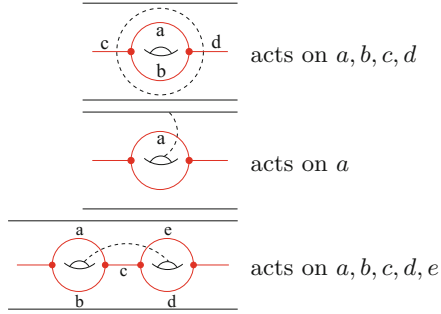
Fig. 6. The Fibonacci representation of a Dehn twist (shown as a dashed line) from the standard generating set is a unitary transformation acting on at most five spine labels.

ordering. One can verify that U is a direct sum of many copies of the Fibonacci representation ρ. (The rounding involved in (1) introduces a minor technical complication, whose resolution may be found in [16].)

For any of the standard Dehn twist generators, $U_{i,j}$ acts on at most 5β qubits, which encode the spine-labels on which ρ acts. The matrix elements by which U acts on these qubits depends on the corresponding thresholds. By (2), these depend on at most two additional registers of qubits, which encode the two spine labels to the left of those being acted upon. Thus, for any of the standard Dehn twist generators, $U_{i,j}$ is a controlled unitary acting on at most 5β target qubits and 2β control qubits. Recalling that $\beta = O(\log|g|)$, we can apply the standard construction from Section 4.5 of [21] to implement this unitary transformation with poly($|g|$) quantum gates, provided each matrix element of $U_{i,j}$ can be computed efficiently. By (3), one sees that the only potentially difficult part of computing the matrix elements of 3 is the computation of the thresholds. An efficient classical algorithm for this task is given in Appendix C.

4   Hardness

In this section we prove that the problem of estimating the normalized WRT Fibonacci invariant of a mapping torus, given by a polynomial-length word in the standard Dehn twist generators of the mapping class group, to within $\pm\epsilon$ is DQC1-hard for $\epsilon < 1/3900$. Generalizing our hardness proof beyond the Fibonacci model seems less straightforward than generalizing our algorithm. However, we consider it likely to be possible. Extending hardness to larger values of ϵ we leave as an open problem. To prove hardness, we reduce from the problem of estimating the absolute value of the normalized trace of a quantum circuit. A proof of the hardness of absolute trace estimation is given in Appendix B. We thus require an efficient procedure that, given a description of a quantum circuit for implementing a unitary U, outputs a description of a mapping torus (i.e., a word in the Dehn twist generators) whose WRT invariant

is close to the trace of U. It turns out to be convenient to suppose that U is a quantum circuit acting on a collection of 5-dimensional qudits ("qupents"). As shown in Appendix A, this makes no difference: the one-clean-qubit model is equivalent to the one-clean-qupent model.

Let U be a quantum circuit of G gates acting on n qupents arranged in a line. Without loss of generality, we may assume that each gate acts either on a single qupent or a pair of neighboring qupents. To prove hardness, we first define a many-to-one encoding $\psi : S_{3n} \to \{0,1,2,3,4\}^n$, where S_{3n} is the set of fusion-consistent labelings of the standard spine of the surface of genus $3n$. We divide the genus-$3n$ surface into n segments, each having three handles. The number of fusion-consistent labelings for a genus-three segment with two punctures depends on the labels on the incoming and outgoing edges, as shown below.

In all cases, the number of fusion-consistent labelings is a multiple of five. Thus, in every case a qupent can be encoded in the space of labelings, together with a "gauge" qudit, whose value we ignore, which has dimension 3, 4, or 7, depending on the labels of the incoming edges. Thus the size $|\psi^{-1}(z)|$ of the preimage of any $z \in \{0,1,2,3,4\}^n$ is exactly independent of z. Given any unitary U acting on n qupents, there corresponds a unitary acting on the span of S_{3n} which acts as U on the encoded qupents, and as the identity on the gauge qudits. We call this the ψ-encoding of U.

As shown in [14], the Fibonacci representation of the mapping class group of the genus $g > 1$ surface is dense in the corresponding unitary group, modulo phase. Thus, given any unitary operation on n qupents, we can find a sequence of Dehn twists which approximates its ψ-encoding arbitrarily closely. The trace of the ψ-encoding is thus equal to the trace of the original quantum circuit, up to a phase. The remaining question is whether this reduction can be done efficiently.

Cutting the genus-$3n$ surface into n equal segments yields $n - 2$ genus-3 doubly-punctured surfaces, and two genus-3 singly-punctured surfaces, as shown below.

One can pants-decompose a punctured surface, thereby associating the surface to a spine. The spine has one "external" edge for each puncture, which attaches to the rest of the spine at only one vertex. Upon labeling the spine, we can associate the label of any external edge with the corresponding puncture. The Fibonacci representation may then be extended to the label-preserving mapping

class group of the punctured surface. This group includes all the standard Dehn twists, together with braiding of punctures with the other punctures of the same label. In the Fibonacci representation, braiding of zero-labeled punctures has no effect, thus a zero-labeled puncture is equivalent to the absence of a puncture.

Theorem 6.2 of [14] states that for any fixed labels on the punctures, the Fibonacci representation of the label-preserving mapping class group of the r-punctured genus-g surface is dense in the corresponding unitary group modulo phase, provided $g + r > 1$. Thus, given any one-qupent gate, the Solovay-Kitaev theorem [21] efficiently yields a sequence of Dehn twists and braid moves on the corresponding genus-3 singly-punctured or doubly-punctured surface, whose Fibonacci representation approximates the ψ-encoded gate arbitrarily closely. Similarly, one efficiently approximates two-qupent gates by moves on genus-6 surfaces with one or two punctures.

We must modify the above construction so as not to use any braiding of punctures. On the leftmost or rightmost qupents there is no problem; the corresponding surfaces have only one puncture, and therefore Theorem 6.2 implies density without using any braiding moves. Similarly, on any of the central surfaces, Theorem 6.2 implies density without using any braiding moves if at least one of the punctures has a zero label. We can ensure this prior to the application of any given gate by adapting the "inchworm" technique from [25], as described in Appendix D. In this method, we bring a pair of zero labels adjacent to the target segment, then implement the desired gate there, and carry the zeros to the segment where the next gate is to be implemented. At the end, we return these zeroes to their original location among the leftmost six handles. As discussed in Appendix D, the inchworm construction entails some overhead in ϵ, which gives rise to the value 1/3900.

In the above construction, we need density on two-punctured segments in which one puncture is guaranteed to be labeled zero, and the other puncture has unknown label. Theorem 6.2 of [14] implies density separately in the subspace in which the other label is zero and in which the other label is one. Because these subspaces have different dimension (20 and 15, respectively) we may apply the decoupling Lemma from [1], which shows that a sequence of Dehn twists can be found to approximate arbitrary pairs of independent unitaries on these two subspaces, as desired.

5   Analogy with Jones Polynomials

In this paper we have shown that estimating the Turaev-Viro invariant of a mapping torus in the Fibonacci model is DQC1-complete. In [3], it was shown that estimating the Turaev-Viro invariant of a general 3-manifold presented as a Heegaard splitting is BQP-complete. Similarly, estimating the Jones polynomial of the trace closure of a braid is DQC1-complete [16,25], while estimating the Jones polynomial of the plat closure of a braid is BQP-complete [1,2,12,13,29]. This suggests a relationship between trace closures and mapping tori on one hand, and between plat closures and Heegaard splittings on the other. Indeed,
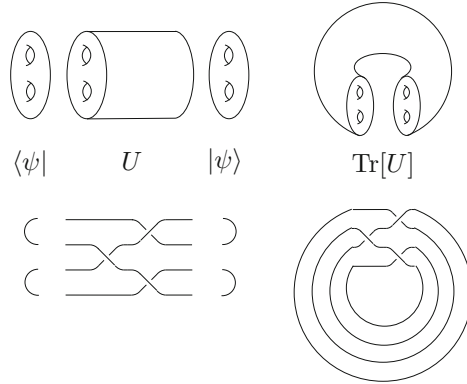
Fig. 7. The problems of estimating the Jones polynomial of the plat closure of a braid and the Turaev-Viro invariant of a Heegaard splitting (left) are BQP-complete. The problems of estimating the Jones polynomial of the trace closure of a braid and the Turaev-Viro invariant of a mapping torus (right) are DQC1-complete. These situations are fundamentally analogous, as discussed in Sect. 5. We stress that the manifold figures are illustrations of the topological ideas behind this analogy, and are not correct two-dimensional projections of the manifolds themselves. In particular, after gluing, the two manifolds shown do not in reality have any boundaries.

such a relationship can be understood in the framework of axiomatic topological quantum field theory, and suggests further generalizations to, for instance, topological invariants of higher dimensional manifolds.

A topological quantum field theory can be axiomatized as a functor T from the category of cobordisms between n-manifolds to the category of linear transformations between vector spaces [5,28]. That is, to each n-manifold the TQFT associates a vector space, and to any $(n + 1)$-manifold whose boundary is the union of two disjoint n-manifolds the TQFT associates a linear transformation between the two associated vector spaces. The functorial property means that gluing together two cobordisms and then applying T yields the same linear transformation that is obtained by applying T to each of the two cobordisms and then composing the resulting linear transformations; see Fig. 8. A TQFT maps the empty n-manifold to the base field, which for the examples we consider is always $\mathbb{C}$. Hence, for M a manifold whose boundary ∂M has a single connected component, $T(M)$ is a map either from $\mathbb{C}$ to the vector space $T(\partial M)$, that is, a vector in $T(\partial M)$, or a map from $T(\partial M)$ to $\mathbb{C}$, that is, a dual vector. The choice between these two possibilities is determined by the orientation of the cobordism.

Recall that the genus-g handlebody is the 3-manifold whose boundary is the genus-g surface Σ_g. For example, the genus-1 handlebody is simply the solid donut. After assigning an orientation, we may think of a handlebody as a cobordism from the empty manifold to Σ_g, or as a cobordism from Σ_g to the empty manifold. Hence, in the TQFT framework, genus-g handlebodies are associated to vectors or dual vectors. We denote these as $|\psi_g\rangle$ and $\langle\psi_g|$, respectively. These
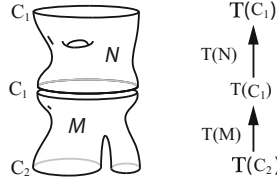
Fig. 8. M can be viewed as a two-manifold with two boundaries: a circle and a pair of circles. The TQFT associates a Hilbert space $T(C_2)$ to the pair of circles, a Hilbert space $T(C_1)$ to the circle, and a linear transformation $T(M) : T(C_2) \rightarrow T(C_1)$ to M. Similarly, $T(N)$ is a linear transformation from $T(C_1)$ to itself. If we glue together M and N along the circle as shown, we obtain a manifold MN with boundaries C_2 and C_1. The corresponding linear transformation is $T(NM) = T(N) \circ T(M)$.

vectors live in the Hilbert space which the TQFT associates to Σ_g. In the case of the Fibonacci model, this is precisely the vector space defined in Sect. 2.2.

In the Fibonacci model, a cobordism from a surface to itself is mapped to a unitary linear transformation U on the associated Hilbert space[3]. If the surface is Σ_g, then we may "cap" the cobordism with handlebodies on both ends. The resulting 3-manifold has no boundary, and thus corresponds to a linear map from $\mathbb{C}$ to itself, i.e., a complex number. In this case, this number is the matrix element $\langle \psi_g | U | \psi_g \rangle$, as illustrated in Fig. 7. The problem of estimating a matrix element of the unitary transformation induced by a quantum circuit is BQP-complete, and this fact underlies the BQP-completeness proof for the Turaev-Viro invariant of Heegaard splittings in [3]. Instead of "capping" the two ends of the cobordism with handlebodies, we could have simply glued the two ends together, resulting in a mapping torus. This is again a 3-manifold without boundary, which thus also corresponds to a complex number. In a TQFT, gluing the two ends of a cobordism corresponds to contracting the two indices of the linear transformation. In other words, instead of a single matrix entry, we now obtain the trace of U. Finding the trace of the unitary transformation induced by a quantum circuit is DQC1-complete, and this fact underlies the DQC1-completeness proof for the Turaev-Viro invariant of mapping tori given in this paper.

The situation regarding Jones polynomials is directly analogous. A TQFT gives us a unitary representation of the braid group. Gluing the two ends of a braid together (i.e., taking the trace closure), as illustrated on the righthand side of Fig. 7, corresponds to taking the trace of the unitary and yields a DQC1-complete problem. Caps correspond to vectors and dual vectors depending on orientation, hence capping a braid (taking the plat closure, as illustrated on the lefthand side of Fig. 7) yields a matrix element of the associated unitary transformation, and corresponds to a BQP-complete problem. The analogy can

---

[3] We may think of the cobordism as describing a sort of spacetime evolution, while the unitary transformation describes the corresponding quantum time evolution. Indeed, this was one of the central motivating ideas behind the development of TQFTs.

be tightened further by noting that the braid group B_n is simply the mapping class group of the surface of genus zero and $n + 1$ punctures (that is, the n-punctured disk), whereas in the case of 3-manifold invariants we consider the mapping class group of the genus-g surface with no punctures. On the other hand, it is worth bearing in mind that the notion of equivalence captured by the Jones polynomial is ambient isotopy, in contrast to the Turaev-Viro and WRT invariants, which capture homeomorphism.

The analogy presented here naturally suggests an extension of BQP-completeness and DQC1-completeness results to n-manifold invariants arising from TQFTs at higher n. More generally, one could attempt to isolate a property of pairs, consisting of a group G and one of its representations U, such that estimating matrix entries of U is BQP-complete while estimating the trace of U is DQC1-complete. Perhaps one could find a general theorem encompassing many such results. We leave this as an open problem.

A   Equivalence Between One Clean Qudit Models

Given a quantum circuit on a-dimensional qudits we wish to construct a quantum circuit on b-dimensional qudits that has the same trace. If $b = ca$ for some integer c then this is easy. We just consider each b-dimensional qudit to be an a-dimensional qudit plus a c-dimensional "gauge" qudit that we ignore. Similarly, if $b^d = ca$ for some integers d, c then we can treat d-tuples of b-dimensional qubits as an a-dimensional qudit plus a c-dimensional gauge qudit. For these encodings, the encoded circuit is easy to construct gate by gate. Given a gate acting on n a-dimensional qudits, we can write down a unitary acting on dn b-dimensional qudits equal to the original gate tensored with the c-dimensional identity on the gauge system. This dn-dimensional gate can be exactly decomposed into a product of $O(b^{2dn})$ 2-qudit gates using the standard construction from Section 4.5 of [21]. Because d and n are constants, this is sufficiently efficient. The normalized trace of the encoded circuit is exactly equal to the normalized trace of the original circuit.

The harder case is when there do not exist integers c and d such that $b^d = ca$. In this case we find $c, d \in \mathbb{Z}$ such that $b^d \simeq ca$. Specifically, suppose we achieve

$$\frac{ca}{b^d} = 1 - \delta \tag{4}$$

for some $\delta \ll 1$. Then we can encode one a-dimensional qudit plus a c-dimensional gauge qudit into d b-dimensional qudits with a few (namely δb^d) noncoding states left over. We can define our encoded gates to act as the identity on these noncoding states. If we make sure the noncoding states are a small fraction of all

b^{dn} states, the normalized trace of the encoded circuit will approximately match the normalized trace of the original circuit.

Let U_a be the original unitary acting on n a-dimensional qudits and let U_b be the unitary acting on dn b-dimensional qudits, in which we encode U_a as described above. Then, U_b acts on b^{dn} states, of which $(ca)^n$ encode states of the original circuit,

$$\frac{\mathrm{Tr}[U_b]}{b^{dn}} = \frac{c^n \mathrm{Tr}[U_a] + (b^{dn} - (ca)^n)}{b^{dn}}.$$

The magnitude of the discrepancy Δ between the normalized traces of U_b and U_a is thus

$$\Delta = \left| \frac{c^n \mathrm{Tr}[U_a] + (b^{dn} - (ca)^n)}{b^{dn}} - \frac{\mathrm{Tr}[U_a]}{a^n} \right|$$

$$= \left| \left(\left(\frac{ca}{b^d} \right)^n - 1 \right) \frac{\mathrm{Tr}[U_a]}{a^n} + 1 - \left(\frac{ca}{b^d} \right)^n \right|$$

$$\leq \left| \left(\frac{ca}{b^d} \right)^n - 1 \right| \cdot \left| \frac{\mathrm{Tr}[U_a]}{a^n} \right| + \left| 1 - \left(\frac{ca}{b^d} \right)^n \right|$$

$$\leq \left| \left(\frac{ca}{b^d} \right)^n - 1 \right| + \left| 1 - \left(\frac{ca}{b^d} \right)^n \right|$$

$$= 2 \left| (1 - \delta)^n - 1 \right|.$$

Thus if

$$\delta = \frac{\epsilon}{n} \tag{5}$$

we have, for small ϵ,

$$\lim_{n \to \infty} \Delta = 2 \left| e^{-\epsilon} - 1 \right| \simeq 2\epsilon. \tag{6}$$

Comparing (4), (5), (6), we see that in the limit of large n and small ϵ, in order to achieve error upper bounded by Δ it suffices to obtain

$$\frac{b^d - ca}{b^d} \leq \frac{\Delta}{2n}.$$

For given b, d, a there always exists an integer c such that $b^d - c \leq a$. So we just need to choose d sufficiently large that

$$\frac{a}{b^d} \leq \frac{\Delta}{2n}.$$

Equivalently,

$$d \geq \log_b \left(\frac{2na}{\Delta} \right).$$

A k-qudit gate from U_a thus gets encoded as a dk-qudit gate in U_b. This encoded gate acts on a b^{dk}-dimensional space. We have just shown that it suffices to choose $d = \log_b \left(\frac{2na}{\Delta} \right)$. Thus the encoded k-qudit gate acts on a $\left(\frac{2na}{\Delta} \right)^k$-dimensional space. Using the construction from section 4.5 of [21], we

can implement an arbitrary D-dimensional unitary exactly with $O(D^2)$ 2-qudit gates. Thus each k-qudit gate in U_a gets encoded by $O\left(\left(\frac{2na}{\Delta}\right)^{2k}\right)$ elementary gates in U_b. By gate universality, we can assume $k \leq 2$, so our encoding has an overhead quartic in n and $1/\Delta$. This is perhaps not very efficient, but is nevertheless polynomial, and thus suffices to prove the equivalence of DQC1 defined with qudits of any constant dimension.

B   Estimating the Absolute Trace is DQC1-Hard

In this section we slightly adapt the proof from [24] to show that estimating the absolute value of the trace of a quantum circuit to within $\pm 1/24$ is a DQC1-complete problem. Consider an arbitrary DQC1 computation. We start with the state $|0\rangle \langle 0| \otimes \frac{\mathbb{1}}{2^n}$, apply an arbitrary quantum circuit U, and then measure the first qubit in the $|0\rangle, |1\rangle$ basis. Changing the initial state of the pure qubit, or changing the measurement basis does not add generality, as these changes can be subsumed into U. The probability of measurement outcome $|0\rangle$ is

$$p_0 = \mathrm{Tr}\left[(|0\rangle \langle 0| \otimes \mathbb{1})U(|0\rangle \langle 0| \otimes \mathbb{1}/2^n)U^\dagger\right]. \tag{7}$$

Let U' be the unitary implemented by the following quantum circuit on $n+2$ qubits.
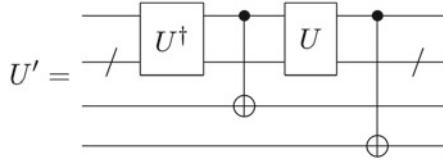
Thus, $p_0 = 2\frac{\mathrm{Tr}U'}{2^{n+2}}$, as one can see by writing out the trace as a sum over diagonal matrix elements in the computational basis. Because p_0 is real it is also true that $p_0 = 2\frac{|\mathrm{Tr}U'|}{2^{n+2}}$. Hence estimating the absolute value of the normalized trace of quantum circuits to suffices to predict the outcome of any DQC1 experiment.

As is standard in the complexity theory of probabilistic computation, "yes" instances of DQC1 are defined to have acceptance probability $2/3$ and "no" instances are defined to have acceptance probability $1/3$. Thus, deciding DQC1 is equivalent to estimating the normalized trace of a quantum circuit to within $\pm 1/6$. The reduction here has a factor of four overhead in normalization, thus estimating the absolute trace to within $\pm 1/24$ is DQC1-complete.

C   Efficiently Computing Thresholds

Consider the standard spine of the genus-g surface, numbered as in Fig. 5. Suppose edges 1 through i have been labeled in a fusion-consistent manner with anyon types $s_1, \ldots, s_i$. We wish to compute how many completions there are to

this partial labelling. That is, we wish to compute the number of fusion-consistent strings of $3g - 3$ labels, whose first i labels are given by $s_1, \ldots, s_i$.

Denote the horizontal edges of the standard spine from right to left by $e_1, e_2, \ldots$, as shown below.
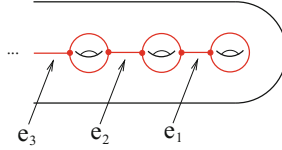
Let $Z_b^{(k)}$ be the number of completions in which the rightmost labeled edge is e_k and has label $b \in \{0, 1\}$. One sees that $Z_0^{(1)} = 2$, and $Z_1^{(1)} = 1$, by the following enumeration of fusion-consistent diagrams.
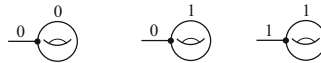
Furthermore, we have the recurrence relations $Z_0^{(n+1)} = 2Z_0^{(n)} + Z_1^{(n)}$ and $Z_1^{(n+1)} = 3Z_1^{(n-1)} + Z_0^{(n-1)}$, by the following enumeration of fusion-consistent diagrams.
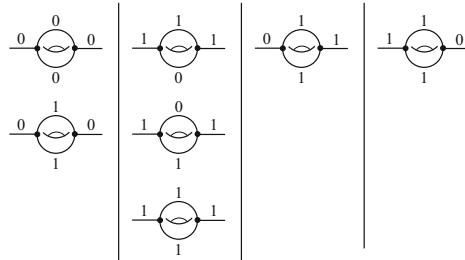

Solving these recurrence relations yields

$$\begin{bmatrix} Z_0^{(n)} \\ Z_1^{(n)} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}^{n-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

The other two cases—completions starting with an upper curved edge, or a lower curved edge—can be solved similarly. The nth power of a matrix may be computed using $O(\log n)$ operations, thus calculating the number of completions for any i in $O(\log g)$ steps. The corresponding thresholds are then immediately obtained by taking ratios of these.

D   Inchworm

Suppose the spine-labeling contains a segment of the following form.
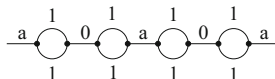
(8)

Here a can be 1 or 0. We call this configuration the inchworm. We may regard
the right instance of as its head, and the left instance as its tail. We
next show a sequence of two reversible operations by which we can move the
inchworm one handle rightward. In the first step the head moves one handle to
the right, leaving the tail in place, and in the second step, the tail catches up,
hence the name "inchworm."
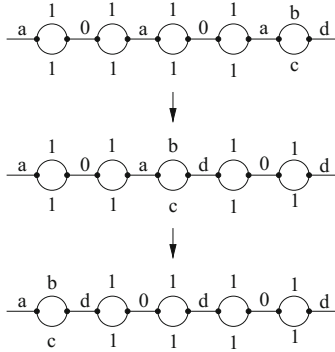
Examination of the above diagram shows that if the fusion rules are obeyed
in the initial configuration, they are also obeyed in the intermediate and final
configurations. Furthermore, both steps are reversible (*i.e.* information preserv-
ing). Thus, they may be written as permutation matrices acting on the space
of allowed configurations, and are therefore unitary. The first unitary transfor-
mation can be implemented by local Dehn twists, because the zero in the tail
of the inchworm implies density of the Fibonacci representation on the segment
to the right of it. The second unitary transformation can be implemented by
local Dehn twists because the zero in the head of the inchworm implies density
on the segment to the left of it. (In both steps, we are applying density to the
twice-punctured genus-4 surface with one puncture labeled zero. There are 75
labelings in which the other puncture is labeled one and 50 labelings in which
the other puncture is labeled zero. Thus, the decoupling lemma of [1] implies
density *jointly* on these two subspaces.) Repeating this process and its reverse,
we may bring the inchworm to any location within the spine.

   To use the inchworm construction, we need to ensure that a segment of the
form (8) exists in the first place. We may do this by implementing a reversible
operation on the leftmost six handles, so that if the configuration (8) is absent,
the matrix is strictly off-diagonal, and does not contribute to the trace. Specifi-
cally, we consider the leftmost two handles to be an ancilla system, and the next
four handles to be the starting location of the inchworm. If these four handles
do not take the form (8) we cyclically permute the (five) basis states of the
ancilla system. Because this is done on the leftmost six handles, the segment
is only singly-punctured, and thus Theorem 6.2 of [14] implies density without
braiding.

The noncontributing labelings decrease the normalized WRT by a constant factor, which correspondingly necessitates decreasing the precision parameter ϵ by the same factor. More precisely, in the Fibonacci model, there are 325 fusion-consistent labelings for the spine of the genus-four doubly-punctured surface. Among these, there are two inchworm configurations ($a = 0$ and $a = 1$). Compounding this 2/325 normalization cost with the precision $\epsilon = 1/24$ obtained in Appendix B for DQC1-hardness of absolute trace, we find that estimating the normalized WRT invariant to within $\pm 1/3900$ is DQC1-hard.

As an aside, we note that the inchworm construction here is simpler than that in [25], in the following sense. The inchworm construction of [25] involved reversible operations on logarithmically large regions. Although the density theorems imply that arbitrary reversible operations can be implemented on these regions, they do not imply that the decomposition into local moves is efficient. Rather this had to be explicitly proven in Appendix D of [25]. In contrast the inchworm construction here involves reversible operations only on $O(1)$ handles, thus no question of efficiency arises.

References

1. Aharonov, D., Arad, I.: The BQP-hardness of approximating the Jones polynomial. New J. Phys. **13**, 035019 (2011). arXiv:quant-ph/0605181
2. Aharonov, D., Jones, V., Landau, Z.: A polynomial quantum algorithm for approximating the Jones polynomial. In: STOC 06 (2006). arXiv:quant-ph/0511096
3. Alagic, G., Jordan, S., König, R., Reichardt, B.: Approximating turaev-Viro 3-manifold invariants is universal for quantum computation. Phys. Rev. A **82**, 040302(R) (2010). arXiv:1003.0923
4. Ambainis, A., Schulman, L.J., Vazirani, U.: Computing with highly mixed states. J. Assoc. Comput. Mach. **53**(3), 507–531 (2006). A preliminary version appears in 2000 and is available at arXiv:quant-ph/0003136
5. Atiyah, Michael: Topological quantum field theories. Publ. Mathématiques de l'IHÉS **68**, 175–186 (1988)
6. Brandão, F.: Entanglement theory and the quantum simulation of many-body physics. Ph.D thesis, Imperial College London (2008). arXiv:0810.0026
7. Datta, A.: Studies on the role of entanglement in mixed-state quantum computation. Ph.D thesis, University of New Mexico (2008). arXiv:0807.4490
8. Datta, A., Flammia, S.T., Caves, C.M.: Entanglement and the power of one qubit. Phys. Rev. A **72**, 042316 (2005). arXiv:quant-ph/0505213
9. Datta, A., Gharibian, S.: Signatures of non-classicality in mixed-state quantum computation. Phys. Rev. A **79**, 042325 (2009). arXiv:0811.4003
10. Datta, A., Shaji, A., Caves, C.M.: Quantum discord and the power of one qubit. Phys. Rev. Lett. **100**, 050502 (2008). arXiv:0709.0548
11. Datta, A., Vidal, G.: On the role of entanglement and correlations in mixed-state quantum computation. Phys. Rev. A **75**, 042310 (2007). arXiv:quant-ph/0611157
12. Freedman, M., Kitaev, A., Wang, Z.: Simulation of topological field theories by quantum computers. Commun. Math. Phys. **227**, 587–603 (2002). arXiv:quant-ph/0001071
13. Freedman, M., Larsen, M., Wang, Z.: A modular functor which is universal for quantum computation. Commun. Math. Phys. **227**, 605 (2002). arXiv:quant-ph/0001108

14. Freedman, Michael H., Larsen, Michael J., Wang, Zhenghan: The two-eigenvalue problem and density of Jones representation of braid groups. Commun. Math. Phys. **228**, 177–199 (2002)
15. Garnerone, S., Marzuoli, A., Rasetti, M.: Efficient quantum processing of three-manifold topological invariants. Adv. Theor. Math. Phys. **13**(6), 1601–1652 (2009). arXiv:quant-ph/0703037
16. Jordan, S.P., Wocjan, P.: Estimating Jones and HOMFLY polynomials with one clean qubit. Quantum Inf. Comput. **9**, 264–289 (2009)
17. Knill, E., Laflamme, R.: Power of one bit of quantum information. Phys. Rev. Lett. **81**(25), 5672–5675 (1998). arXiv:quant-ph/9802037
18. Knill, E., Laflamme, R.: Quantum computation and quadratically signed weight enumerators. Inf. Process. Lett. **79**(4), 173–179 (2001). arXiv:quant-ph/9909094
19. Luo, Shunlong: Using measurement-induced disturbance to correlations as classical or quantum. Physi. Rev. A **77**, 022301 (2008)
20. Marx, R., Fahmy, A., Kauffman, L., Lomonaco, S., Spörl, A., Pomplun, N., Myers, J., Glaser, S.J.: NMR quantum calculations of the Jones polynomial (2009). arxiv:0909.1080
21. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
22. Passante, G., Moussa, O., Ryan, C.A., Laflamme, R.: Experimental approximation of the Jones polynomial with DQC1. Phys. Rev. Lett. **103**, 250501 (2009)
23. Roberts, Justin D.: Skein theory and Turaev-Viro invariants. Topology **34**, 771–787 (1995)
24. Shepherd, D.: Computation with unitaries and one pure qubit (2006). arXiv:quant-ph/0608132
25. Shor, P.W., Jordan, S.P.: Estimating Jones polynomials is complete for one clean qubit. Quantum Inf. Comput. **8**(8/9), 681–714 (2008)
26. Turaev, V.G.: Topology of shadows. preprint (1991)
27. Turaev, V.G.: Quantum Invariants of Knots and 3-manifolds. de Gruyter Studies in Mathematics, vol. 18. de Gruyter, New York (1994)
28. Walker, K.: On Witten's 3-manifold invariants. http://canyon23.net/math/1991TQFTNotes.pdf (1991)
29. Wocjan, P., Yard, J.: The Jones polynomial: quantum algorithms and applications in quantum complexity theory. Quantum Inf. Comput. **8**, 147–180 (2008). arXiv:quant-ph/0603069