The Jones polynomial: quantum algorithms and applications in quantum complexity theory

Pawel Wocjan* and Jon Yard†

Institute for Quantum Information
California Institute of Technology
Pasadena, CA 91125, USA

May 11th, 2006

Abstract

We analyze relationships between the Jones polynomial and quantum computation. First, we present two polynomial-time quantum algorithms which give additive approximations of the Jones polynomial, in the sense of Bordewich, Freedman, Lovász and Welsh, of any link obtained from a certain general family of closures of braids, evaluated at any primitive root of unity. This family encompasses the well-known plat and trace closures, generalizing results recently obtained by Aharonov, Jones and Landau. We base our algorithms on a local qubit implementation of the unitary Jones-Wenzl representations of the braid group which makes the underlying representation theory apparent, allowing us to provide an algorithm for approximating the HOMFLYPT two-variable polynomial of the trace closure of a braid at certain pairs of values as well. Next, we provide a self-contained proof that any quantum computation can be replaced by an additive approximation of the Jones polynomial, evaluated at almost any primitive root of unity. This theorem was originally proved by Freedman, Larsen and Wang in the context of topological quantum computation, and the necessary notion of approximation was later provided by Bordewich et al. Our proof is simpler as it uses a more natural encoding of two-qubit unitaries into the rectangular representation of the eight-strand braid group. We then give QCMA-complete and PSPACE-complete problems which are based on braids. Finally, we conclude with direct proofs that evaluating the Jones polynomial of the plat closure at most primitive roots of unity is a #P-hard problem, while learning its most significant bit is PP-hard, without taking the usual route through the Tutte polynomial and graph coloring.

1 Introduction

There is evidence that a computer which could manipulate quantum mechanical degrees of freedom would be more powerful than a classical computer. Since the discovery by Shor that a quantum computer could efficiently factor composite integers, a task which is believed to be hard on conventional computers, much effort has been expended toward understanding the capabilities and limitations of quantum computers. In this paper, we investigate ways in which the Jones polynomial invariant of knots and links contributes to this understanding. Witten discovered [39] that the Jones polynomial could be understood via tools from topological quantum field theories. Freedman, Kitaev, Larsen and Wang [11] established its connection to computer science in the

*wocjan@cs.caltech.edu
†jtyard@cs.caltech.edu

context of topological quantum computation. In a topological quantum computer, the trajectories of particles which are restricted to a plane are braided in order to manipulate the internal state of the computer. They showed that such computers can be simulated on conventional quantum computers, and also that when the underlying physics of a computer is described by a suitable topological quantum field theory, that it is possible to achieve universal quantum computation.

In [10], it was shown that topological quantum computers could be efficiently simulated by computers based on the standard quantum circuit model, implying the existence of an efficient algorithm for approximating the Jones polynomial of certain links obtained from braids. More recently, Aharonov, Jones and Landau [2] gave explicit quantum algorithms for approximating the Jones polynomial of either the trace or plat closure of a braid on a quantum computer. Our first contribution in this paper is an explicit local unitary implementation of the so-called Jones-Wenzl unitary representations of the braid group. We then generalize the main results of [2], presenting two different quantum algorithms for approximating the Jones polynomial of links obtained via a general class of closures of braids which encompasses the plat and trace closures treated in the earlier paper. Our implementation of the Jones-Wenzl representations is similar to that of [2], except that we make the underlying representation theory apparent, allowing us to give an algorithm for approximating the HOMFLYPT two-variable polynomial of the trace closure of a braid, evaluated at certain pairs of points corresponding to unitary Jones-Wenzl representations. In [7], the notion of an additive approximation of a function with respect to a normalization was introduced in order to make precise the kind of approximate evaluation of the Jones polynomial which captures the power of quantum computation. In the sense of that paper, our quantum algorithms obtain additive approximations of Jones and HOMFLYPT polynomial evaluations in polynomial-time.

In [12, 13], a converse to the above simulation results was proved. More specifically, it was shown there that quantum circuits can be simulated by braids, in such a way that the output probabilities of the quantum circuit are functions of the Jones polynomial of the plat closure of some braid of comparable length to that of the original circuit. Our second contribution is a simpler proof of their main result. In fact, this work grew from our attempts to understand the connections between these works and those of [2].

Finally, we give applications of the Jones polynomial to quantum complexity theory. We begin by restating a result that was proved in [7], showing that a machine which obtains an additive approximation of the Jones polynomial of the plat closure of a braid is equally as powerful as a quantum computer. We then introduce two new problems, **Increase Jones Plat** and **Approximate Concatenated Jones Plat**. The first asks if a given braid can be conjugated by another braid from a given class such that the Jones polynomial of its plat closure is nearly maximal. We prove that this problem is complete for the complexity class QCMA, a certain quantum analog of NP. The latter problem asks if a given braid, after being concatenated with itself exponentially many times, has a large plat closure. We show that this problem is PSPACE-complete. Finally, we give self-contained proofs that learning n of the most significant bits of the Jones polynomial of the plat closure of an n-strand braid is a #P-hard problem, while learning its most significant bit is PP-hard. This constitutes a simpler, quantum-based proof of the known result that computing the Jones polynomial is #P-hard. The original proof of this (see e.g. [38]) relates the Jones polynomial to the Tutte polynomial of a signed graph to the chromatic number of that graph to the #P-complete problem #SAT. Our proof, however, shows that learning a linear number of the highest order bits of the Jones polynomial of the plat closure of a braid is #P-hard.

The paper is organized as follows. In the next section, we review elements of the theory of links, braids and the Jones polynomial. There, we review some different ways of turning a braid into a link, while introducing a new type of closure which we call a generalized closure. In Section 3, we describe the connection between the Jones polynomial and the representation theory of the braid group, reviewing the necessary representation theory required to understand the results in this paper. Section 4 begins by recalling the notion of an additive approximation which was introduced

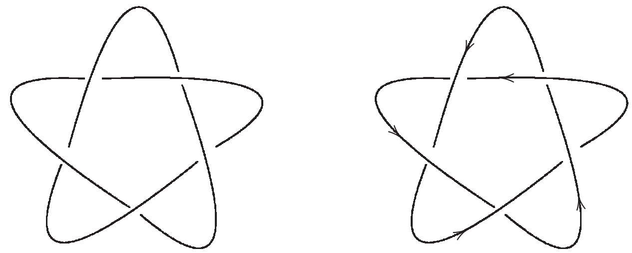


Figure 2.1: Cinquefoil represented by its link projection (left) and an oriented cinquefoil (right)

in [7], followed by a brief overview of the standard quantum circuit model. Then, we provide our local qubit model for the Jones-Wenzl representations, after which we present our algorithms for approximating the Jones and HOMFLYPT polynomials. In Section 5, we provide a self-contained proof of the converse of the results in the previous section – that local quantum circuits can be simulated by braids. Finally, in Section 6, we review notions of classical and quantum complexity theory, after which we present our complexity-theoretic results. We conclude with a discussion in Section 7.

2 Braids, links and the Jones polynomial

This section provides a short introduction to links, braids and the Jones polynomial. We refer the reader to, for example, [38] for more details. A *knot* is a closed, nonintersecting curve in $\mathbb{R}^3$. More generally, a *link* is an embedding of some finite number of closed curves in $\mathbb{R}^3$. Formally, a link is a closed 1-dimensional submanifold of $\mathbb{R}^3$. Links are identified up to *isotopy*, which means that two links which are related by some bijection $f : \mathbb{R}^3 \to \mathbb{R}^3$ of the ambient space with itself for which f and f^{-1} are continuous are considered to be *equivalent*. An *oriented link* is a link in which every component is assigned an orientation. If $\overrightarrow{L}$ is an oriented link, we will denote the unoriented link resulting from forgetting the orientation as L.

A central problem in knot theory is to determine, given descriptions of two links, whether or not they are equivalent. In order to solve such a problem, one must decide on how the links are to be described. One such way is in terms of a *link projection*. Informally, this is a two-dimensional diagram which uniquely specifies the link up to isotopy. Rather than give a formal definition, we provide an example of a projection of a link, with and without orientation, in Figure 2.1, referring the reader to textbooks such as [25, 27, 28, 38] for a more rigorous presentation. The key property of any link projection is that at each crossing, it keeps track of which string goes above or below the other. Usually, this information is conveyed by leaving a gap in the string of the undercrossing. A classic result in knot theory states that two unoriented links are equivalent if and only if the link projection of one can be transformed into that of the other by a finite sequence of *Reidemeister moves* (see e.g. any of the texts listed above), which are tabulated in Figure 2.2.

The Jones polynomial

Another way of (partially) distinguishing links involves assigning an *invariant* to each link so that links which are equivalent up to isotopy have the same invariant. In this paper, we will consider one such invariant, the *Jones polynomial*, which is an invariant of oriented links. A particularly simple definition of this invariant can be given in terms of the *Kauffman bracket* [21], which assigns to each unoriented link L a polynomial in variables A and A^{-1}, written $\langle L \rangle$. The Kauffman bracket can be computed inductively from a link projection as follows. At each step, one replaces a given link

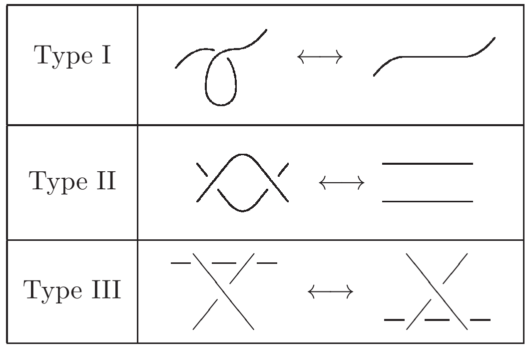


Figure 2.2: Reidemeister moves of type I, II and III.

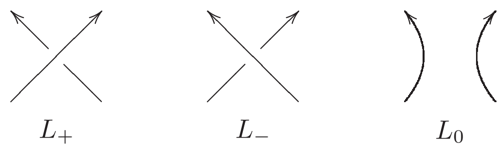
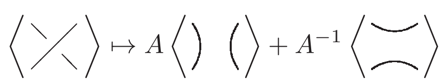


Figure 2.3: L_+, L_-, L_0 denote three oriented links that differ in a small region, as symbolized by the above diagrams, but are otherwise the same. We say that the crossing corresponding to L_+ is a positive crossing, and the one corresponding to L_- is a negative crossing.

projection with a formal linear combination of link projections with one fewer crossing, according to the substitution rule



(2.1)

The base case replaces m copies of the unknot with $(-A^2-A^{-2})^{m-1}$, where the quantity $-A^2-A^{-2}$ is usually referred to as the *loop constant* in the literature. The Kauffman bracket can be used to define the Jones polynomial, which is an invariant of *oriented* links, as follows. Given an oriented link $\overrightarrow{L}$, its *writhe* $w(\overrightarrow{L})$ is defined as the sum over components of the number of positive crossings L_+ minus the number of negative crossings L_- (see Figure 2.3) encountered while travelling along the assigned direction on each oriented component. The Jones polynomial $J_{\overrightarrow{L}}(q)$ of $\overrightarrow{L}$ is then defined as

$$J_{\overrightarrow{L}}(q) = (-A)^{-3w(\overrightarrow{L})} \, \langle L \rangle \Big|_{A=q^{-1/4}} . \tag{2.2}$$

On occasion, we write $J\big(\overrightarrow{L}, q\big) \equiv J_{\overrightarrow{L}}(q)$. It can be shown (see e.g. [38]) that the Jones polynomial satisfies the following *skein relation*

$$q^{-1} J_{L_+}(q) - q J_{L_-}(q) = (q^{1/2} - q^{-1/2}) J_{L_0}(q) \, , \tag{2.3}$$

which serves as an alternative definition of $J_{\overrightarrow{L}}(q)$. Historically, these definitions of the Jones polynomial were not the first to appear in the literature. Instead, its original definition was based on a description of a link as a certain closure of a braid, as we will describe in the next section. But first, we mention a two-variable generalization of the Jones polynomial which has become known as the HOMFLYPT [16, 31] polynomial $H_{\overrightarrow{L}}(t, x)$ of an oriented link $\overrightarrow{L}$. It satisfies the more general skein relation

$$t^{-1} H_{L_+}(t, x) - t H_{L_-}(t, x) = x H_{L_0}(t, x) \, . \tag{2.4}$$

The Jones polynomial is recovered as $J_{\overrightarrow{L}}(q) = H_{\overrightarrow{L}}(q, q^{1/2} - q^{-1/2})$.

The braid group and links as closures of braids

The braid group B_n on n strands is generated by $\{\sigma_1, \sigma_2, \ldots, \sigma_{n-1}\}$ which satisfy

$$\sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j, \qquad |i-j| = 1 \tag{2.5}$$

$$\sigma_i \sigma_j = \sigma_j \sigma_i \qquad |i-j| \geq 2. \tag{2.6}$$

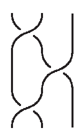
The reader should picture n hanging pieces of string, and interpret each generator σ_i as a counterclockwise exchange of the strands in positions i and $i+1$, with inverses of the generators corresponding to clockwise twists. For instance, the braids $\sigma_1, \sigma_2^{-1} \in B_3$ are given by

$\sigma_1 =$  and $\sigma_2^{-1} =$  .

The group product then corresponds to vertical concatenation of braids, so that

$\sigma_2^{-1}\sigma_1 =$  .

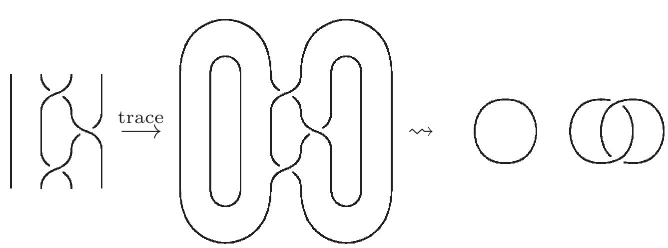
The relation (2.5) is known as the Yang-Baxter equation, which is diagrammatically expressed as

 =  .

For each n, B_n is isomorphic to the subgroup of B_{n+1} generated by $\{\sigma_1, \ldots, \sigma_{n-1}\}$, which consists of braids leaving the last strand alone. We denote the associated inclusion map as $\iota \colon B_n \to B_{n+1}$, which acts by adding an extra strand to the right of any given braid. Given a braid $b \in B_n$, there are many possible ways of *closing* the braid b to obtain a link. In the following we will recall three possible ways, the trace, the plat and the generalized closure.

Trace closure

One way to turn a braid into a link takes, for each $1 \leq i \leq n$, the i'th strand at the top of the braid and glues it to the i'th strand at the bottom. The resulting link $\widehat{b}$ is called the *trace closure* of b. For instance, the trace closure of the braid $b = \sigma_2 \sigma_3^{-1} \sigma_2$ is isotopic to the union of the unknot and the Hopf link, as



It is possible to show that any link can be represented as the trace closure of some braid. Conventionally, one considers the trace closure of a braid to yield an oriented link, since an orientation can be unambiguously defined to travel upward along the braid. It is readily verified that with this convention every generator σ_i gives rise to a positive crossing, while its inverse σ_i^{-1} to a negative

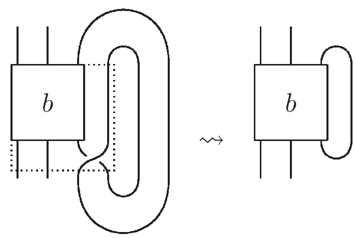


Figure 2.4: Trace closure after the second Markov move on some $b \in B_3$. The dotted box on the left represents the braid $\sigma_3 \iota(b) \in B_4$. We have omitted the strands which close the two leftmost strands for visibility.

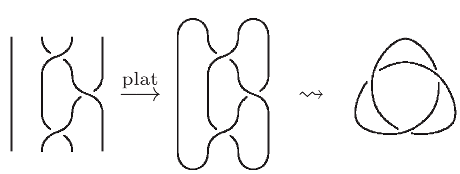
crossing in the sense of Figure 2.3. For computational purposes, however, it is sufficient to consider trace closures to be unoriented as changing orientation only affects the value of the writhe, which is easily computable in linear time in the number of braid generators.

A theorem of Markov demonstrates that, given braids b and b', their trace closures $\widehat{b}$ and $\widehat{b'}$ are isotopic if and only if b and b' are related by a finite sequence of *Markov moves*, which we now review. If $b \in B_n$, then the first Markov move consists of conjugating b by any braid $x \in B_n$, taking $b \mapsto xbx^{-1}$. The second Markov move adds another strand to b which is twisted either clockwise or counterclockwise with the n'th strand at the bottom of b, acting as $b \mapsto \sigma_n^{\pm}\iota(b)$. It is clear that the trace closures of braids related by the first Markov move are isotopic, since the top braid x is untied by the bottom braid x^{-1}. A picture more clearly illustrates the invariance of the trace closure under the second Markov move, as shown in Figure 2.4.

The original definition of the Jones polynomial was for oriented links $\widehat{b}$ obtained as the trace closure of some braid b. Specifically, $J_{\widehat{b}}(q)$ was defined as a weighted sum of traces of the images of b under certain irreducible representations of the braid group which are parametrized by the variable q. We will give this formula explicitly in Section 3.2 after we describe the above mentioned representations.

Plat closure

Another way of turning a braid on an even number $2n$ of strands into a link is known as the *plat closure*. Such a procedure starts with a braid $b \in B_{2n}$ and connects adjacent pairs of strands at the top and at the bottom. For example, the plat closure of $b = \sigma_2\sigma_3^{-1}\sigma_2 \in B_4$ is isotopic to the trefoil knot, as



Just as Markov's theorem states that two braids b and b' have isotopic trace closures if and only if we can go from b to b' by applying a sequence of Markov moves, there is similar characterization for the plat closure. We will call the corresponding moves the *Birman moves* [5], defined as follows.

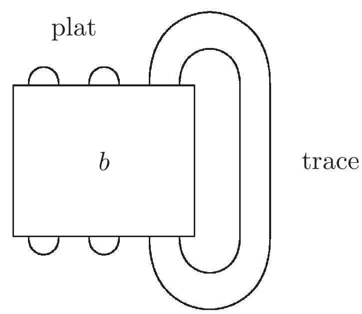
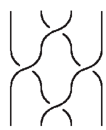
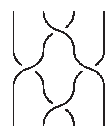


Figure 2.5: Generalized closure $\chi^{2,2}_{1,1}(b)$ of $b \in B_6$: the four leftmost strands are closed as in the plat closure and the remaining two as in the trace closure.

Define $x, y \in B_4$ as

$$x = \sigma_2\sigma_3\sigma_1\sigma_2 =$$  , $$y = \sigma_2\sigma_3\sigma_1^{-1}\sigma_2^{-1} =$$  .

For each n, the *Birman group* C_{2n} is the subgroup of B_{2n} generated by $\{x_i, y_i\}_{i=1}^{n-1}$ and $\{z_j\}_{j=1}^{n}$, where x_i is x acting on the strands $2i-1, 2i, 2i+1$ and $2i+2$, y_i is defined similarly, and $z_i = \sigma_{2i-1}$. The *first Birman move* takes $b \in B_{2n}$ to ubv where $u, v \in C_{2n}$. The *second Birman move* takes b to $\sigma_{2n}^{\pm 1}\iota \cdot \iota(b) \in B_{2n+2}$, where two strands are added to the right of b before multiplication by σ_{2n}.

Generalized closure

We now introduce a more general way to close a braid which encompasses both the trace and plat closures. Given a braid $b \in B_n$, a *generalized closure* of b is specified by the following data: two braids $x, y \in B_n$ and nonnegative integers p and r which satisfy $2p + r = n$. The generalized closure $\chi^{p,r}_{x,y}(b)$ of b is obtained by first forming the braid xby, performing the plat closure on the $2p$ leftmost strands at the top and bottom, then connecting the remaining r strands at the top with the corresponding t strands at the bottom as in the trace closure. This process is depicted in Figure 2.5. It is clear that this generalizes the trace and plat closures, as the former can be written $\chi^{0,n}_{1,1}(b)$, while the latter, if n is even, is just $\chi^{n/2,0}_{1,1}(b)$, where 1 denotes the identity element of the braid group B_n. In Section 3.2, we give a representation-theoretic formula for the Jones polynomial of such a closure of a braid at primitive roots of unity, while in Section 4, we present an algorithm which approximates the Jones polynomial of links obtained by such closures of a braid.

3 Representation theory and the Jones polynomial

We now revisit the connection between braids and the Jones polynomial from a representation-theoretic perspective. This allows us to present the original definition of the Jones polynomial of the trace closure of a braid, along with formulae for the plat and generalized closures as well. To this end, we summarize the representation theory of the braid group that we will need, following [37]. The relevant representations are inherited from representations of a certain family of Hecke algebras $H_n(q)$ which generalize the group algebra $\mathbb{C}S_n$ of the symmetric group S_n.

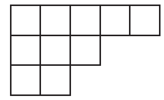


Figure 3.1: Young diagram for the partition $\lambda = [5, 3, 2]$.

3.1 Braid group representations

The Hecke algebra

It is well-known that the group algebra $\mathbb{C}S_n$ of the symmetric group S_n on $\{1, 2, \ldots, n\}$ objects has a presentation in terms of generators $\{1, s_1, s_2, \ldots, s_{n-1}\}$, where s_i is the involution which swaps i and $i+1$, which satisfy the relations

$$\begin{aligned} s_i^2 &= 1 && (3.1)\\ s_i s_j s_i &= s_j s_i s_j, \quad |i-j| = 1 && (3.2)\\ s_i s_j &= s_j s_i, \quad |i-j| > 1. && (3.3) \end{aligned}$$

For every $q \in \mathbb{C}^\times$, the *Hecke algebra* $H_n(q)$ of type A_{n-1} is generated by 1 and $\{g_1, \ldots, g_{n-1}\}$ satisfying the relations

$$\begin{aligned} g_i^2 &= g_i(q-1) + q && (3.4)\\ g_i g_j g_i &= g_j g_i g_j \qquad |i-j| = 1 && (3.5)\\ g_i g_j &= g_j g_i \qquad |i-j| > 1. && (3.6) \end{aligned}$$

Note that this is a deformation of the group algebra $\mathbb{C}S_n$ of the symmetric group S_n, which is obtained when $q = 1$. While it is known that $H_n(q) \simeq \mathbb{C}S_n$ whenever q is not a root of unity, we will rather be interested in the cases when q is a primitive root of unity, where the representation theory is slightly more subtle than that of $\mathbb{C}S_n$. One may represent the braid group B_n inside the Hecke algebra $H_n(q)$ by simply mapping each generator σ_i of B_n to the generator g_i of $H_n(q)$. Indeed, the representations of B_n we will use in this paper are induced by representations of $H_n(q)$ on suitable finite-dimensional Hilbert spaces. As with the braid groups, for each n, $H_n(q)$ is isomorphic to the subalgebra of $H_{n+1}(q)$ generated by 1 and $\{g_1, \ldots, g_{n-1}\}$.. We use the same notation for the inclusion maps as with the braid groups, writing $\iota\colon H_n(q) \to H_{n+1}(q)$. One important distinction is that, writing 1_n for the identity in $H_n(q)$, we have $\iota(1_n) = 1_{n+1}$. We will therefore often omit the subscript for the identity element of each $H_n(q)$ whenever there is no cause for confusion.

Young diagrams and tableaux

Like the unitary irreps of S_n, the unitary irreps of B_n are labelled by *partitions* $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ satisfying $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ and $\sum_i \lambda_i = n$. Throughout, we consider all partitions obtained by adding or deleting trailing zeros to be equivalent. We will identify any such partition with its *Young diagram* as pictured in Figure 3.1 for the partition $\lambda = [5, 3, 2]$. Note that the diagram has one row for each *part* of the partition, and that the i'th row contains λ_i boxes. We allow for the empty partition and diagram, which we denote $\emptyset$. Let Λ_n be the collection of n-box Young diagrams. Given two diagrams $\lambda \in \Lambda_n$ and $\mu \in \Lambda_m$, their *sum* $\lambda + \mu \in \Lambda_{n+m}$ is always well-defined, having parts equal to $(\lambda + \mu)_i = \lambda_i + \mu_i$. When it is well-defined as a partition, we will similarly define the *difference* $\lambda - \mu \in \Lambda_{n-m}$ to have parts $(\lambda - \mu)_i = \lambda_i - \mu_i$.

Define the sets of diagrams $\Lambda_{m,n} \equiv \biguplus_{i=m}^{n} \Lambda_i$ and abbreviate $\Lambda \equiv \Lambda_{0,\infty}$. We may consider elements of Λ to be the nodes of an infinite directed graph, which we will call the *Young graph*, which contains an edge from a diagram λ' to a diagram λ whenever a single box can be added to λ'

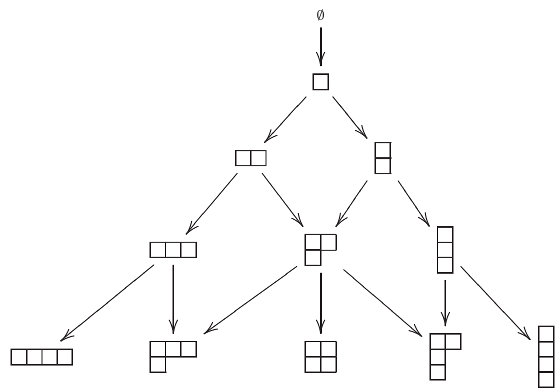


Figure 3.2: The Young graph $\Lambda_{0,4}$.

to obtain λ. We remark that we depart slightly from common terminology, where this graph, with each arrow reversed, is called a *Bratteli diagram*. With a slight abuse of notation, we write Λ for both the set of diagrams and for the corresponding Young graph, making a similar identification between subsets of Λ such as $\Lambda_{0,n}$ and the associated subgraph. In Figure 3.2, we give a picture of $\Lambda_{0,4}$.

By a *numbering* of an n-box diagram, we mean any assignment of the numbers $\{1, 2, \ldots, n\}$ to the boxes of the diagram, where each number appears only once. For a given diagram $\lambda \in \Lambda_n$, we write T_λ for the set of *standard tableaux* with shape λ, corresponding to numberings which are strictly increasing along each row and column. We often call a standard tableau just a *tableau*. We denote the set of all n-box tableaux as $T_n = \biguplus_{\lambda \in \Lambda_n} T_\lambda$ and the set of all tableaux as $T = \biguplus_{n=1}^{\infty} T_n$. We identify members of T_n with length-n paths starting at $\emptyset$ in $\Lambda_{0,n}$, pictured in Figure 3.2, with elements of T_λ correspond to paths which end at the diagram λ.

In this paper, we shall rather require certain restricted classes of Young diagrams and tableaux. To each pair (k, ℓ) of integers satisfying $\ell > k > 0$ corresponds a class of irreducible unitary representations of B_n. These representations are labelled by the n-box (k, ℓ)-*Young diagrams*, defined as

$$\Lambda_n^{(k,\ell)} = \{\lambda \in \Lambda_n : \lambda_{k+1} = 0, \lambda_1 - \lambda_k \leq \ell - k\}.$$

Note that these are the diagrams with at most k rows for which the difference between the numbers of boxes in the first and k'th rows is at most $\ell - k$. We will often just say that such a λ is an (k, ℓ)-diagram. For a given diagram λ, we refer to $\lambda_1 - \lambda_k$ as the *level* of λ, so that $\ell - k$ is the maximum level of all diagrams in $\Lambda_n^{(k,\ell)}$. As before, we make the similar abbreviations $\Lambda_{m,n}^{(k,\ell)}$ and $\Lambda^{(k,\ell)}$. We also identify any of these sets of diagrams with the appropriate subgraph of Λ. We refer to such a graph as a (k, ℓ)-*Young graph*. In Figure 3.3, we show the graph $\Lambda_{0,4}^{(2,5)}$. For a given $\lambda \in \Lambda_n^{(k,\ell)}$, define the (k, ℓ)-*tableaux* $T_\lambda^{(k,\ell)} \subset T_\lambda$ to be those standard tableaux of shape λ for which successively deleting the largest numbered boxes yields, at each step, another (k, ℓ)-diagram. These tableaux can be identified with paths in $\Lambda_{0,n}^{(k,\ell)}$ going from $\emptyset$ to λ. We also define $T_n^{(k,\ell)}$ and $T^{(k,\ell)}$ in a similar manner as above. In the next subsection, we will see that the $(2, \ell)$-diagrams and tableaux are what is needed to give representation-theoretic expressions for the Jones polynomials of closures of braids. Central to many of the results in this paper is the following easily verifiable observation about a subset of these diagrams and tableaux.

Remark. *Suppose that $\lambda \in \Lambda_n^{(2,\ell)}$ and $m \geq 0$. Then $[m, m] + \lambda \in \Lambda_{2m+n}^{(2,\ell)}$. If $t \in T_{[m,m]}^{(2,\ell)}$ and $t' \in T_\lambda^{(2,\ell)}$, define tt' to be the numbering of shape $[m, m] + \lambda$ obtained by adding $2m$ to each box*

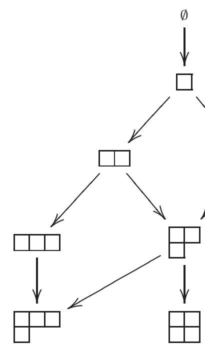


Figure 3.3: The Young graph $\Lambda_{0,4}^{(2,5)}$. Comparison with Figure 3.2 reveals that this is the subgraph of $\Lambda_{0,4}$ obtained by removing vertices whose diagrams either have more than two rows, or level greater than $3 = 5 - 2$.

of t', then placing t' to the right of t. Then $tt' \in T^{(2,\ell)}_{[m,m]+\lambda}$. Note that such a statement will not always hold if $[m,m]$ is replaced by an arbitrary $(2,\ell)$-diagram.

The symmetric group S_n has an obvious action on the numberings of any n-box diagram, obtained by permuting the numbers within the boxes. This induces an action on (k,ℓ)-tableaux $S_n\colon T_n^{(k,\ell)} \to T_n^{(k,\ell)}$ as follows. Since S_n is generated by local transpositions $\{s_1, s_2, \ldots, s_{n-1}\}$, where each s_i acts on $\{1,\ldots,n\}$ by only swapping i and $i+1$, it suffices to specify the action of each generator s_i on $T_n^{(k,\ell)}$. Let $t \in T_n^{(k,\ell)}$. If swapping the numbers i and $i+1$ in t results in another (k,ℓ)-tableau t', we define $s_i(t) = t'$ and $s_i(t') = t$, otherwise setting $s_i(t) = t$. In order that s_i act nontrivially on a (k,ℓ)-tableau t, it is necessary and sufficient that i and $i+1$ *not* be in the same row or column of t. In such a case, we may think of it as a deformation of the path in $\Lambda_{0,n}^{(k,\ell)}$ corresponding to t which changes only the i-box diagram passed in the original path. Since no permutation can change the shape of any tableau, it is clear that S_n acts transitively on T_n with orbits labeled by diagrams in $\Lambda_n^{(k,\ell)}$. Finally, note that each generator induces a partition of $T_n^{(k,\ell)}$ into sets of numberings (or paths) of size at most two, corresponding to the orbits under the action of the abelian subgroup $\{1, s_i\} \subset S_n$. In the next subsection, we will use the actions of the s_i's on (k,ℓ)-tableaux to define the unitary representations of B_n which we need in this paper.

Unitary Jones-Wenzl representations of B_n

The representations of B_n we will introduce are parametrized by integers $\ell > k > 0$, where ℓ specifies a primitive ℓ'th root of unity which we will denote $q = e^{2\pi i/\ell}$. The irreducible components of the representations we will describe reduce to the usual irreps of S_n when $k = n$ and $\ell \to \infty$, so that $q \to 1$. The matrix elements of these representations will be defined in terms of constants called *quantum integers*[1]. Define for each integer d and each nonnegative integer ℓ, the constant

$$[d]_\ell = \left.\frac{q^{d/2} - q^{-d/2}}{q^{1/2} - q^{-1/2}}\right|_{q=e^{2\pi i/\ell}} = \frac{\sin(\pi d/\ell)}{\sin(\pi/\ell)}.$$

[1]Our definition here departs from the usual convention [37, 12] which defines, for arbitrary q, $[d]_q = \frac{q^{d/2}-q^{-d/2}}{q^{1/2}-q^{-1/2}}$. This expression, however, evaluates to ours when q is a primitive ℓ'th root of unity, which is the case of interest to us. We warn that other authors (e.g. [19]) use a different convention for the quantum integer and q.

A value which will occur frequently is

$$[2]_\ell = q^{1/2} + q^{-1/2}\Big|_{q=e^{2\pi i/\ell}} = 2\cos(\pi/\ell). \tag{3.7}$$

For the rest of this section, we fix $\ell > k > 0$ and set $q = e^{2\pi i/\ell}$, describing a corresponding representation of $H_n(q)$ which induces a unitary representation of B_n. This representation is more easily described by a change of variables which leads to an equivalent presentation of $H_n(q)$. Rewriting the quadratic relation (3.4) as $(g_i + 1)(g_i - q) = 0$, we may define idempotents $e_i = (q - g_i)/(1 + q)$ (meaning that $e_i^2 = e_i$) for which

$$g_i = q(1 - e_i) - e_i = q - (1 + q)e_i. \tag{3.8}$$

Using these idempotents, the relations (3.4–3.6) may be expressed as

$$\begin{aligned} e_i^2 &= e_i && && (3.9)\\ e_ie_je_i - \tau e_j &= e_je_ie_j - \tau e_i, && |i-j| = 1 && (3.10)\\ e_ie_j &= e_je_i, && |i-j| > 1, && (3.11) \end{aligned}$$

where we follow the usual convention in setting $\tau \equiv [2]_\ell^{-2}$. For each integer d, define

$$a_\ell(d) = \frac{[d+1]_\ell}{[2]_\ell[d]_\ell}, \tag{3.12}$$

noting that $a_\ell(d) + a_\ell(-d) = 1$ and $a_\ell(1) = 1$. Given any n-box tableau t and an integer $1 \le i < n$, we respectively write $c_i(t)$ and $r_i(t)$ for the *column* and *row* which contain the number i, defining

$$d_i(t) \equiv c_i(t) - c_{i+1}(t) - (r_i(t) - r_{i+1}(t)). \tag{3.13}$$

Note that $d_i(t)$ gives the total number of leftward and downward steps it takes to "walk" from i to $i+1$ on t, where moving right or up counts negatively. We abbreviate $a_\ell(i,t) \equiv a_\ell\big(d_i(t)\big)$.

We now define a representation $\pi_n^{(k,\ell)}$ of $H_n(q)$ on a vector space $V_n^{(k,\ell)}$ which has an orthonormal basis labelled by n-box (k,ℓ) tableaux, i.e.

$$V_n^{(k,\ell)} = \mathrm{span}\big\{|t\rangle : t \in T_n^{(k,\ell)}\big\}.$$

For each i, $\pi_n^{(k,\ell)}(e_i)$ is a projection which is block diagonal in the $T_n^{(k,\ell)}$ basis with blocks of size either one or two. The size two blocks correspond to distinct pairs of tableaux t and $s_i(t)$ which are contained in $T_n^{(k,\ell)}$. Letting $V_{i,t} = \mathrm{span}\big\{|t\rangle, |s_i(t)\rangle\big\}$, the restriction of $\pi_n^{(k,\ell)}(e_i)$ to $V_{i,t}$ is the rank one projection onto $\sqrt{a_\ell(i,t)}|t\rangle + \sqrt{1 - a_\ell(i,t)}|s_i(t)\rangle$, written

$$\pi_n^{(k,\ell)}(e_i)\big|_{V_{i,t}} = \begin{pmatrix} a_\ell(i,t) & \sqrt{a_\ell(i,t)(1 - a_\ell(i,t))} \\ \sqrt{a_\ell(i,t)(1 - a_\ell(i,t))} & 1 - a_\ell(i,t) \end{pmatrix}. \tag{3.14}$$

For the remaining $t \in T_n^{(k,\ell)}$, set $\pi_n^{(k,\ell)}(e_i)|t\rangle = |t\rangle$ if i and $i+1$ are in the same column of t; otherwise setting $\pi_n^{(k,\ell)}(e_i)|t\rangle = 0$. This defines the action of $\pi_n^{(k,\ell)}(e_i)$ on all of $V_n^{(k,\ell)}$. Using (3.8), this representation maps each g_i to the unitary matrix

$$\pi_n^{(k,\ell)}(g_i) \quad = \quad q1_{V_n^{(k,\ell)}} - (1+q)\pi_n^{(k,\ell)}(e_i). \tag{3.15}$$

Because permutations cannot change the shapes of tableaux, it follows that this representation is reducible. In fact, its irreducible components are labelled by the (k,ℓ)-diagrams, so that

$$\pi_n^{(k,\ell)} = \bigoplus_{\lambda \in \Lambda_n^{(k,\ell)}} \pi_\lambda^{(k,\ell)},$$

where each $\pi_\lambda^{(k,\ell)}$ acts nondegenerately only on the subspace $V_\lambda^{(k,\ell)} = \{|t\rangle : t \in T_\lambda^{(k,\ell)}\}$, and

$$V_n^{(k,\ell)} = \bigoplus_{\lambda \in \Lambda_n^{(k,\ell)}} V_\lambda^{(k,\ell)}.$$

Let us briefly focus again on the case $k = 2$. Recall that as we remarked earlier in this section, for each pair $(t, t') \in T_{[m,m]}^{(2,\ell)} \times T_\lambda^{(2,\ell)}$ of tableaux, their concatenation satisfies $tt' \in T_{[m,m]+\lambda}^{(2,\ell)}$. In such a case, we will identify $V_{[m,m]}^{(2,\ell)} \otimes V_\lambda^{(2,\ell)}$ with the subspace of $V_{[m,m]+\lambda}^{(2,\ell)}$ spanned by the basis states $\{|tt'\rangle, (t,t') \in T_{[m,m]}^{(2,\ell)} \times T_\lambda^{(2,\ell)}\}$ via the identification $|t\rangle|t'\rangle \mapsto |tt'\rangle$. We remark that a similar identification might not be possible if $[m, m]$ is replaced by a non-rectangular diagram.

3.2 Representation-theoretic formulae for the Jones polynomial

We will now illustrate how to use the above representation theory to write expressions for the Jones polynomial of links obtained as closures of braids. First, we will show how positive, normalized Markov traces on the Hecke algebras give rise to expressions for the Jones polynomial and the HOMFLYPT polynomials of the trace closure of a braid, evaluated at particular points. After this, we introduce the Temperley-Lieb algebra, which we use to derive a formula for the Jones polynomial of our generalized closure of a braid, evaluated at primitive roots of unity. Finally, we illustrate how this formula specializes to the plat closure.

Derivation of the Jones polynomial of trace closures from the Markov trace

Using the above unitary representations, we will introduce the theory of Markov traces on Hecke algebras which leads to a formula for the Jones polynomial of the trace closure of a braid. In fact, we will show a bit more, taking a route through a two-variable generalization of the Jones polynomial which is known as the HOMFLYPT polynomial. We direct the reader who is not interested in these details to the formula (3.24). Our presentation is primarily based on Section 6 of [19] and Section 3 of [37], while being influenced by [4] as well. Throughout this section, we fix a primitive root of unity $q = e^{2\pi i/\ell}$ for some integer $\ell \geq 3$. It is a theorem of Ocneanu [16] that for every $1 \leq k < \ell$, if we set $\eta_{k\ell} \equiv a_\ell(-k)$, there is, for each n, a linear function $\mathrm{tr}_{k\ell} \colon H_n(q) \to \mathbb{C}$ uniquely determined by

1. $\mathrm{tr}_{k\ell}(1_n) = 1$ for $1_n \in H_n(q)$
2. $\mathrm{tr}_{k\ell}(wv) = \mathrm{tr}_{k\ell}(vw)$ for $w, v \in H_n(q)$
3. $\mathrm{tr}_{k\ell}(\iota(w)e_n) = \eta_{k\ell}\, \mathrm{tr}_{k\ell}(w)$ for all $w \in H_n(q)$ for all n.

It can be shown [19] that $\eta_{k\ell} = \mathrm{tr}_{k\ell}(e_i)$ for every $1 \leq i < n$. Such a function is known as a *positive Markov trace* on $H_\infty(q)$. For $w \in H_n(q)$, this trace has the following representation-theoretic formula

$$\mathrm{tr}_{k\ell}(w) = \sum_{\lambda \in \Lambda_n^{(k,\ell)}} s_\lambda^{(k,\ell)}\, \mathrm{Tr}\, \pi_\lambda^{(k,\ell)}(w). \tag{3.16}$$

Here, the *Markov weights* $s_\lambda^{(k,\ell)}$ are Schur functions given explicitly by

$$s_\lambda^{(k,\ell)} = \frac{1}{[k]_\ell^n} \prod_{(i,j)\in\lambda} \frac{[j-i+k]_\ell}{[h(i,j)]_\ell} \tag{3.17}$$

for each $\lambda \in \Lambda_n^{(k,\ell)}$ and $h(i,j)$ denotes the *hook length* of the box with row-column coordinates (i,j) in λ. For each $\lambda \in \Lambda_n^{(k,\ell)}$, if we write $d_\lambda^{(k,\ell)}$ for the dimension of $V_\lambda^{(k,\ell)}$, we may evaluate (3.16) at the identity $1_n \in H_n(q)$ to obtain the following formula relating the Markov weights and the dimensions of the irreps:

$$\sum_{\lambda \in \Lambda_n^{(k,\ell)}} s_\lambda^{(k,\ell)} d_\lambda^{(k,\ell)} = \mathrm{tr}_{k\ell}(1_n) = 1. \tag{3.18}$$

To see how such a positive Markov trace can be used to define an invariant of links given as trace closures of braids, recall that Markov's theorem states that two braids $b \in B_n$, $b' \in B_{n'}$ have isotopic trace closures if and only if they are connected by a finite sequence of *Markov moves* of

- type I: $b \mapsto \gamma b \gamma^{-1}$, where $b, \gamma \in B_n$
- type II: $b \mapsto \sigma_n^{\pm 1}\iota(b)$ if $b \in B_n$.

We remind the reader that for each n, the map $\iota\colon B_n \to B_{n+1}$ adds an extra strand to the right of a given braid. Since we represent B_n inside $H_n(q)$ by identifying each braid with its image under the mapping of generators $\sigma_i^{\pm} \mapsto g_i^{\pm}$, we extend this trace to B_n. While it is clear that this Markov trace is invariant under the first Markov move, it not invariant under the second move, because multiplication of $w \in H_n(q)$ by either g_n or g_n^{-1} affects the trace nontrivially. Our goal will then be to define, for each n, a normalized version $\overline{\mathrm{tr}}_{k\ell}\colon H_n(q) \to \mathbb{C}$ of the Markov trace which *is* invariant under the second Markov move. For this, we will require that

$$\overline{\mathrm{tr}}_{k\ell}(\sigma_n^{\pm 1}\iota(b)) = \overline{\mathrm{tr}}_{k\ell}(b) \tag{3.19}$$

for each $b \in B_n$. We begin by normalizing the g_i's so g_i and g_i^{-1} have the same trace. Letting $\theta_{k\ell} \in \mathbb{C}$ be such that $\mathrm{tr}_{k\ell}(\theta_{k\ell} g_i) = \mathrm{tr}_{k\ell}((\theta_{k\ell} g_i)^{-1})$ and solving for $\theta_{k\ell}^2$, one obtains $\theta_{k\ell}^2 = \mathrm{tr}_{k\ell}\, g_i^{-1} / \mathrm{tr}_{k\ell}\, g_i$. It is conventional to choose the negative square root, giving

$$\theta_{k\ell} = -\sqrt{\frac{\mathrm{tr}_{k\ell}\, g_i^{-1}}{\mathrm{tr}_{k\ell}\, g_i}} = -\sqrt{\frac{q^{-1} - (1+q^{-1})\eta_{k\ell}}{q - (1+q)\eta_{k\ell}}}. \tag{3.20}$$

Observe that since q is a root of unity and $\eta_{k\ell}$ is real, it follows that $\theta_{k\ell}$ is just a complex phase. After a little algebra, we may express the Markov trace of both normalized generators as

$$\nu_{k\ell} \equiv \mathrm{tr}_{k\ell}(\theta_{k\ell} g_i) = \mathrm{tr}_{k\ell}\left((\theta_{k\ell} g_i)^{-1}\right) = -\sqrt{(\mathrm{tr}_{k\ell}\, g_i)(\mathrm{tr}_{k\ell}\, g_i^{-1})} = -\sqrt{1 + [2]_\ell^2(\eta_{k\ell}^2 - \eta_{k\ell})}\,. \tag{3.21}$$

Therefore, if we instead represent B_n inside $H_n(q)$ via the identification of rescaled generators $\sigma_i^{\pm} \mapsto (\theta g_i)^{\pm 1}$, we obtain, for each n, a trace $\mathrm{tr}'_{k\ell}\colon H_n(q) \to \mathbb{C}$ which acts on each $b \in B_n$ as

$$\mathrm{tr}'_{k\ell}(b) = \theta_{k\ell}^{e(b)}\, \mathrm{tr}_{k\ell}(b).$$

Here, $e(b)$ is the sum of the exponents of the generators in the expression of b. To see that we are almost at (3.19), notice that $\mathrm{tr}'_{k\ell}\left(\sigma_n^{\pm 1}\iota(b)\right) = \nu_{k\ell}\, \mathrm{tr}'_{k\ell}(b)$. It is then immediate that we may define the *positive, normalized Markov trace* to act as

$$\overline{\mathrm{tr}}_{k\ell}(b) = \left(\frac{1}{\nu_{k\ell}}\right)^{n-1} \mathrm{tr}'_{k\ell}(b)$$

for each $b \in B_n$, or rather

$$\overline{\mathrm{tr}}_{k\ell}(b) = \left(\frac{1}{\nu_{k\ell}}\right)^{n-1} \theta_{k\ell}^{e(b)}\, \mathrm{tr}_{k\ell}(b). \tag{3.22}$$

This formula not only defines an invariant for links obtained by taking the trace closures of braids. It is also equal to the HOMFLYPT polynomial $H_{\hat{b}}(t, x)$ of the trace closure of b evaluated at the points $t = e^{2\pi i/\ell}$ and $x = \theta_{k\ell} e^{\pi i/\ell}$.

In this paper, we are mainly interested in the Jones polynomial, which is the specialization of the HOMFLYPT corresponding to $k = 2$. In this case, we find that

$$\eta_{2\ell} = a_\ell(-2) = \frac{[-2+1]_\ell}{[2]_\ell[-2]_\ell} = \frac{1}{[2]_\ell^2}.$$

Plugging this value into (3.21), (3.20) and (3.17), we obtain

$$\nu_{2\ell} = -\frac{1}{[2]_\ell}, \quad \theta_{2\ell} = -q^{-3/2} \quad \text{and} \quad s_\lambda^{(2,\ell)} = \frac{[\lambda_1 - \lambda_2 + 1]_\ell}{[2]_\ell^n}. \tag{3.23}$$

For the convenience of the reader, we have collected the calculations which lead to the above expressions in Section 9.2. Plugging these values into (3.22) leads to the following expression for the evaluation at $q = e^{2\pi i/\ell}$ of the Jones polynomial of the trace closure of a braid $b \in B_n$:

$$\begin{aligned} J_{\widehat{b}}(e^{2\pi i/\ell}) &= \overline{\mathrm{tr}}_{2\ell}(b) \\ &= (-[2]_\ell)^{n-1}\left(-e^{-3\pi i/2\ell}\right)^{e(b)} \sum_{\lambda\in\Lambda_n^{(2,\ell)}} \frac{[\lambda_1-\lambda_2+1]_\ell}{[2]_\ell^n} \operatorname{Tr}\pi_\lambda^{(2,\ell)}(b) \\ &= \frac{(-1)^{n-e(b)-1}\, e^{-3\pi i e(b)/2\ell}}{[2]_\ell} \sum_{\lambda\in\Lambda_n^{(2,\ell)}} [\lambda_1-\lambda_2+1]_\ell \operatorname{Tr}\pi_\lambda^{(2,\ell)}(b)\,. \end{aligned} \tag{3.24}$$

Because the numerator of the coefficient in front is just a complex phase, the absolute value of the Jones polynomial is

$$\left|J_{\widehat{b}}(e^{2\pi i/\ell})\right| = \left|\overline{\mathrm{tr}}_{2\ell}(b)\right| = \frac{1}{[2]_\ell}\left|\sum_{\lambda\in\Lambda_n^{(2,\ell)}} [\lambda_1-\lambda_2+1]_\ell \operatorname{Tr}\pi_\lambda^{(2,\ell)}(b)\right|. \tag{3.25}$$

In the rest of this section, we will obtain representation-theoretic formulae for the Jones polynomial of the generalized closure, and more specifically, of the plat closure of a braid. But first, we need to briefly describe the connection between the Jones polynomial and the Temperley-Lieb algebra.

The Temperley-Lieb algebra

As we have described in the previous subsection, the representations corresponding to the case $k = 2$ are what is relevant for the Jones polynomial. While this restriction may seem a bit arbitrary, it is actually because the discovery of the Jones polynomial was not made in the context of Hecke algebras. Rather, the original definition of the Jones polynomial involved the Temperley-Lieb algebra [35], which Jones had discovered independently in earlier work [17] on subfactors of von Neumann algebras. Setting $\tau = [2]_\ell^{-2}$ and $q = e^{2\pi i/\ell}$, the Temperley-Lieb algebra $TL_n(\tau)$ is equal to the quotient of $H_n(q)$ by the relation $e_ie_je_i = \tau e_i$, and is thus the algebra generated by 1 and $\{e_1, \ldots, e_{n-1}\}$ satisfying the relations

$$e_i^2 = e_i \tag{3.26}$$

$$e_ie_je_i = \tau e_i \qquad |i-j| = 1 \tag{3.27}$$

$$e_ie_j = e_je_i \qquad |i-j| > 1\,. \tag{3.28}$$

It is well-known (see e.g. [19]) that the relation $e_ie_je_i = \tau e_i$ is satisfied only in representations of the Hecke algebra with at most two rows; the irreducible representations of $TL_n(\tau)$ therefore are exactly given by those of $H_n(q)$ corresponding to two-row diagrams. While the representation theory of

$TL_n(\tau)$ is therefore a restricted version of that of $H_n(q)$, the former algebra is more expressive in the following sense. Recall that our representations of B_n were induced by representations of $H_n(q)$ via the mapping of generators $\sigma_i^{\pm 1} \mapsto g_i^{\pm 1}$. For obtaining the Jones polynomial, we could have done the same with $TL_n(\tau)$, with $g_i^{\pm 1} = q^{\pm 1} - (1 + q^{\pm 1})e_i$. In fact, it is possible to present $TL_n(\tau)$ in terms of the generators g_i as well, although the relations become somewhat complicated and unenlightening.

The power of the Temperley-Lieb algebra can be viewed from the perspective of the *diagram monoid* K_n, which was defined by Kauffman [21] as follows. For each $1 \le i \le n-1$, define the "cup-cap" diagram

$$\omega_i = \Big| \cdots \Big| \;\; \begin{matrix} \smile \\ \frown \end{matrix} \;\; \Big| \cdots \Big| ,$$

where the cup-caps act on strands i and $i+1$. Multiplication of cup-caps is performed pictorially, amounting to stacking the generators ω_i in the same way as with the generators of B_n. Then, K_n is generated by the cup-cap generators $\{\omega_1, \ldots, \omega_{n-1}\}$, together with an extra closed-loop diagram δ, which satisfy the relations

$$\omega_i^2 = \delta\omega_i \tag{3.29}$$
$$\omega_i\omega_j\omega_i = \omega_i \qquad |i-j| = 1 \tag{3.30}$$
$$\omega_i\omega_j = \omega_j\omega_i \qquad |i-j| > 1\,. \tag{3.31}$$

These relations are perhaps best understood in terms of pictures, for which we refer the reader to Figure 8 of [21]. While these relations are similar to those used above to define the Temperley-Lieb algebra, a more transparent connection is obtained by reformulating (3.26)–(3.28) in terms of the scaled projections $E_i \equiv [2]_\ell e_i$, in which case the corresponding relations

$$E_i^2 = [2]_\ell E_i \tag{3.32}$$
$$E_iE_jE_i = E_i \qquad |i-j| = 1 \tag{3.33}$$
$$E_iE_j = E_jE_i \qquad |i-j| > 1 \tag{3.34}$$

are formally identical to (3.29)–(3.31). The diagram monoid K_n is then represented inside $TL_n(\tau)$ via the map taking $\omega_i \mapsto E_i$ and $\delta \mapsto [2]_\ell$. In fact, this map leads to an isomorphism between the algebra $\mathbb{C}K_n$ of formal linear combinations of elements from K_n and $TL_n(\tau)$. As with B_n, we use this map to obtain representations of K_n from those of $TL_n(\tau)$ by setting

$$\pi_\lambda^{(2,\ell)}(\omega_i) = \pi_\lambda^{(2,\ell)}(E_i) = [2]_\ell \pi_\lambda^{(2,\ell)}(e_i).$$

For every even number $2n$, define the rectangular tableau $t_{2n} \in T_{[n,n]}$ by

$$t_{2n} = \begin{array}{|c|c|c|}\hline 1 & 3 & 5 \\ \hline 2 & 4 & 6 \\ \hline \end{array} \cdots \begin{array}{|c|}\hline {\scriptstyle 2n-1} \\ \hline 2n \\ \hline \end{array} . \tag{3.35}$$

For convenience we will write the projection onto the corresponding vector $|t_{2n}\rangle$ as $\varphi_{2n} \equiv |t_{2n}\rangle\langle t_{2n}|$. Consider the following lemma.

Lemma 3.1. *Let p and n be positive integers satisfying $2p \le n$. If $\lambda = [\lambda_1, \lambda_2] \in \Lambda_n^{(2,\ell)}$, then*

$$\pi_\lambda^{(2,\ell)}(\omega_1\omega_3\cdots\omega_{2p-1}) = \begin{cases} [2]_\ell^p\, \varphi_{2p} \otimes 1_{\lambda-[p,p]} & \textit{if } \lambda_2 \ge p \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. We abbreviate $e_i = \pi_n^{(2,\ell)}(e_i)$. It suffices to prove this for the case where $2p = n$, as the general case follows directly. It will thus be enough to check that $e_{2i-1}|t_{2p}\rangle = |t_{2p}\rangle$ for each $1 \leq i \leq p$, while also showing that for any $t \in T_{[p,p]}^{(2,\ell)}$ with $t \neq t_{2p}$, there exists some $1 \leq i \leq p$ for which $e_i|t\rangle = 0$. Recalling that $e_j|t\rangle = |t\rangle$ iff j and $j+1$ are in the same column, while $e_j|t\rangle = 0$ iff j and $j+1$ are in the same row, this follows because for every i, the boxes in t_{2p} containing $2i-1$ and $2i$ are never in the same column, while for any other t, there is necessarily an i for which this is true. ∎

It is possible to combine B_n and K_n into a unified structure which was called a *braid monoid* by Kauffman [21]. As this braid monoid contains diagrams consisting of both cup-caps and twists, its generators are the union of those of B_n and K_n. However, there are considerably more relations that these generators must satisfy (see e.g. [6]) in addition to those of the braid group and diagram monoid. For our purposes, however, we will not need the full braid monoid. Instead, we shall only need to consider such generalized braids of the form bw, where $b \in B_n$ and $w \in K_n$ is a word in the ω_i's, in which case $\pi_n^{(2,\ell)}(bw) = \pi_n^{(2,\ell)}(b)\pi_n^{(2,\ell)}(w)$.

Jones polynomial of the generalized and plat closures

For any braid $b \in B_n$, we give a formula for the absolute value of the Jones polynomial of the generalized closure of b when $x = y = 1_n \in B_n$. Such a closure then depends only the parameters p and r which satisfy $2p + r = n$. We obtain this formula by computing the normalized Markov trace of the braid b with p cup-caps at the tops of the leftmost $2p$ strands. The formula is

$$\begin{aligned}
\left|J\big(\chi_{1,1}^{p,r}(b), e^{2\pi i/\ell}\big)\right| &= \left|\overline{\mathrm{tr}}_{2\ell}(b\omega_1\omega_3\cdots\omega_{2p-1})\right| \\
&= \frac{1}{[2]_\ell}\left|\sum_{\lambda\in\Lambda_n^{(2,\ell)}} [\lambda_1-\lambda_2+1]_\ell \,\mathrm{Tr}\left(\pi_\lambda^{(2,\ell)}(b)\pi_\lambda^{(2,\ell)}(\omega_1\omega_3\cdots\omega_{2p-1})\right)\right| \\
&= [2]_\ell^{p-1}\left|\sum_{\lambda\in\Lambda_n^{(2,\ell)}:\lambda_2\geq p} [\lambda_1-\lambda_2+1]_\ell \,\mathrm{Tr}\left(\pi_\lambda^{(2,\ell)}(b)(\varphi_{2p}\otimes 1_{\lambda-[p,p]})\right)\right| \\
&= [2]_\ell^{p-1}\left|\sum_{\mu\in\Lambda_r^{(2,\ell)}} [\mu_1-\mu_2+1]_\ell \,\mathrm{Tr}\left(\pi_{[p,p]+\mu}^{(2,\ell)}(b)(\varphi_{2p}\otimes 1_{\mu})\right)\right|, \qquad (3.36)
\end{aligned}$$

In the second line, we have used the formula (3.25) for the absolute value of the normalized Markov trace. The third line is by Lemma 3.1, while the last line is straightforward.

We remark here that this expression yields an immediate upper bound on the absolute value of the Jones polynomial of a link obtained as a generalized closure of a braid. By unitarity, it is clear that the matrix traces above are maximized when the formula is evaluated at the identity braid $1_n \in B_n$. In this case, we obtain

$$\begin{aligned}
\left|J\big(\chi_{1,1}^{p,r}(1_n), e^{2\pi i/\ell}\big)\right| &= [2]_\ell^{p-1}\sum_{\mu\in\Lambda_r^{(2,\ell)}} [\mu_1-\mu_2+1]_\ell \, d_\mu^{(2,\ell)} \\
&= [2]_\ell^{p-1}\sum_{\mu\in\Lambda_r^{(2,\ell)}} [2]_\ell^r \, s_\mu^{(2,\ell)}\, d_\mu^{(2,\ell)} \\
&= [2]_\ell^{p+r-1}, \qquad (3.37)
\end{aligned}$$

where we have used (3.18) for the last step. By specializing (3.36) to the case where $r = 0$ and

$p = n$, we obtain the following formula for the Jones polynomial of the plat closure of $b \in B_{2n}$:

$$\begin{aligned} \left| J(\widetilde{b}, e^{i2\pi/\ell}) \right| &= [2]_\ell^{n-1} \left| \mathrm{Tr}\left(\pi_{[n,n]}^{(2,\ell)}(b)\, \varphi_{2n} \right) \right| \\ &= [2]_\ell^{n-1} \left| \langle t_{2n} | \pi_{[n,n]}^{(2,\ell)}(b) | t_{2n} \rangle \right| . \end{aligned} \tag{3.38}$$

4 An algorithm for approximating the Jones polynomial of generalized closures of braids

In this section, we give a polynomial-time quantum algorithm which approximates the absolute value of the Jones polynomial of a link obtained as a generalized closure of a braid. The notion of approximation which we will require was recently formalized in [7]; we review that material here. Let $\mathcal{X}$ be a set of problem instances, and suppose we are given a nonnegative function $f\colon \mathcal{X} \to \mathbb{R}$ which is potentially difficult to evaluate exactly. The main idea of [7] is to approximate the function f with respect to some positive normalization function $g\colon \mathcal{X} \to \mathbb{R}$. Departing slightly from the definition given in [7], an *additive approximation* for the normalized function f/g associates a random variable $Z(x)$ to any problem instance $x \in \mathcal{X}$ and $\delta > 0$ satisfying

$$\Pr\left\{ \left| \frac{f(x)}{g(x)} - Z(x) \right| \leq \delta \right\} \geq 3/4.$$

In addition, it is required that such an approximation be achieved in time which is polynomial in the size of the problem instance and in $1/\delta$.

As we saw in (3.37), the maximal absolute value of the Jones polynomial of links obtained by any closure of an n-strand braid is obtained by closing the identity braid. In particular, we saw that the generalized closure with $2p + r = n$ (with identity braids $x = y = 1_n$ at the top and bottom) of the identity braid 1_n is isotopic to $p + r$ unknots, yielding a Jones polynomial of $[2]_\ell^{p+r-1}$. We will use this normalization for our additive approximations, so that the absolute value of the resulting normalized Jones polynomial always lies between 0 and 1, regardless of the size of the braid, number of strands, or particular primitive root of unity. Specifically, we wish to solve the following problem.

Problem 4.1 (Approximate Jones Closure). Given is a braid $b \in B_n$ of length m, two braids $x, y \in B_n$ of length $O\big(\mathrm{poly}(m)\big)$, positive integers ℓ, p and r which satisfy $2p + r = n$, and $\delta > 0$. The task is to sample from a random variable Z which is an additive approximation of the absolute value of the Jones polynomial of the generalized closure, evaluated at $e^{2\pi i/\ell}$, in the sense that

$$\Pr\left\{ \left| \frac{1}{[2]_\ell^{p+r-1}} \left| J\big(\chi_{x,y}^{p,r}(b), e^{2\pi i/\ell}\big) \right| - Z \right| \leq \delta \right\} \geq 3/4.$$

The following theorem will be proved in Section 4.3.

Theorem 4.1. *There is a quantum algorithm which solves* **Approximate Jones Closure** *in* $O(\mathrm{poly}(m, 1/\delta))$ *time.*

Before we describe our quantum algorithm, we give a brief overview of the standard quantum circuit model.

4.1 Standard quantum circuit model

According to quantum mechanics, the state space of a quantum computer has the structure of a Hilbert space $\mathcal{H}$. Denoting by $U(\mathcal{H})$ the group of unitary matrices acting on $\mathcal{H}$, a *quantum gate* is any $U \in U(\mathcal{H})$. By the *standard quantum circuit model*, we shall mean quantum computers whose state space decomposes into a finite number of localized two-dimensional subsystems, or qubits, so that $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$. We fix a preferred orthonormal basis for the Hilbert space of each qubit, which we call the *computational basis*:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The state space of $\mathcal{H}$ is thus spanned by tensor products of these basis vectors, which we write as $\{|x^n\rangle : x^n \in \{0,1\}^n\}$, where

$$|x^n\rangle \equiv |x_1\rangle|x_2\rangle \cdots |x_n\rangle \equiv |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle.$$

We will abbreviate the set of all gates on n qubits as $U(2^n) \equiv U\big((\mathbb{C}^2)^{\otimes n}\big)$. For this paper, we will make the somewhat arbitrary assumption that the qubits are arranged on a line, so that each qubit has at most two neighbors. In this case, it is well-known that an arbitrary unitary $U \in U(2^n)$ can be performed on the qubits by a sequence of (possibly exponentially many in n) *local unitaries* from $U(4)$ which act only on adjacent pairs of qubits. By a *quantum circuit*, we will mean a sequence of local two-qubit gates. Whenever we say that a given unitary matrix U is a quantum circuit of length m, we will mean that it can be written as $U = U_m U_{m-1} \cdots U_1$, where each U_i is a two-qubit local unitary in $U(4)$ which we identify with its embedding into $U(2^n)$ via $U_i \mapsto 1^{\otimes i-1} \otimes U_i \otimes 1^{\otimes n-i-1}$. In the standard quantum circuit model, it is understood that the qubits can be kept from interacting amongst themselves and with the environment. In addition, we assume that it is possible to:

1. prepare each qubit in either of the basis states $|0\rangle$ or $|1\rangle$
2. perform two-qubit local gates
3. measure each qubit in the basis $\{|0\rangle, |1\rangle\}$.

Such a procedure results in a probabilistic outcome. According to quantum mechanics, if the qubits are prepared in the state $|x^n\rangle$ and transformed according to $U \in U(2^n)$, the probability that a computational basis measurement results in the bit string y^n is

$$\Pr\{\text{measure } y^n | \text{ prepared } x^n\} = |\langle y^n|U|x^n\rangle|^2.$$

As $U(4)$ contains uncountably many gates, an infinite amount of classical information would generally be required to specify an arbitrary unitary on just a single pair of qubits. As we will see, it is enough to be able to approximate these gates using a *finite* set of gates. A set of gates in $U(\mathcal{H})$ is said to be *universal* if it generates a dense subset of $U(\mathcal{H})$. Given unitaries U and U' and some $0 \le \epsilon \le 1$, we say that U *ϵ-approximates* U' if $\|U - U'\|_\infty \le \epsilon$, where for a square matrix M, $\|M\|_\infty$ denotes the *sup-norm* of M, which equals the largest singular value of M. The following lemma is proved in the appendix.

Lemma 4.2. *Let U and U' be $d \times d$ unitaries for which U' ϵ-approximate U. Then, for every pair of normalized pure states $|\phi\rangle, |\psi\rangle \in \mathbb{C}^d$, we have*

$$\big| |\langle \phi|U|\psi\rangle|^2 - |\langle \phi|U'|\psi\rangle|^2 \big| \le \epsilon,$$

In particular, this implies that the measurement probabilities of an ϵ-approximation of a quantum circuit differ by at most ϵ from those of the desired circuit. The following theorem assures that, given a finite universal set $\mathcal{G}_2$ of unitaries on two qubits, any unitary in $U(4)$ can be ϵ-approximated efficiently.

Solovay-Kitaev Theorem [33, 24, 8]. *Let $\mathcal{G} \subset U(d)$ be a finite universal set of unitaries on a Hilbert space of constant dimension d which are closed under inverses, i.e. $\mathcal{G}^{-1} = \mathcal{G}$. Then any unitary $U \in U(d)$ can be ϵ-approximated by a sequence of $O\big(\operatorname{poly}\log(1/\epsilon)\big)$ gates in $\mathcal{G}$. Moreover, such a sequence can be found by a classical computer in $O\big(\operatorname{poly}\log(1/\epsilon)\big)$ time.*

Fixing any finite, universal set of two-qubit unitaries $\mathcal{G}_2 \subset U(4)$, let $\mathcal{G}_n \subset U(2^n)$ be the set of gates obtained by letting $\mathcal{G}_2$ act on all pairs of neighboring qubits. Given a length m circuit $U_m U_{m-1} \cdots U_1$ consisting of gates from $\mathcal{G}_n$ and any *fixed* desired degree of accuracy ϵ, the Solovay-Kitaev Theorem further implies that for each i, there is an ϵ/m approximation U_i' to U_i which is obtained by a sequence of $O\big(\operatorname{poly}\log(m)\big)$ gates in $\mathcal{G}$, and also that such a sequence can be found in $O\big(\operatorname{poly}\log(m)\big)$ time (since ϵ does not grow with m, it does not figure into the complexity estimate). Observe that for any unitaries U and U', the operator norm is *stable*, meaning that $\|U - U'\|_\infty = \|U \otimes 1_d - U' \otimes 1_d\|_\infty$ for every finite dimension d. Therefore, the same approximations hold for the embeddings of the local gates into $U(2^n)$. By the following lemma, which is proved in the appendix, this means that there is an ϵ-approximation to the desired circuit which uses $O\big(m \operatorname{poly}\log(m)\big)$ gates from $\mathcal{G}_n$.

Lemma 4.3. *Let $\{U_i, U_i'\}_{i=1}^m$ be unitaries of the same size. Then*

$$\|U_m \cdots U_2 U_1 - U_m' \cdots U_2' U_1'\|_\infty \leq \sum_{i=1}^m \|U_i - U_i'\|_\infty.$$

Therefore, working with finite universal sets of gates is just as good as working with all possible two-qubit unitaries, provided that one can live with bounded errors and polynomially more gates. In Section 6, we define two classes of quantum computational problems which are insensitive to these limitations.

4.2 Local qubit model for Jones-Wenzl representations

We will now show, for any integers $\ell > k > 1$, how to implement the corresponding Jones-Wenzl representations $\pi_n^{(k,\ell)}$ defined in (3.15) on a quantum computer. The basic idea is to embed these representations into a tensor product space (c.f. Section 5.2 of [32]). For simplicity, we only explicitly present the case $k = 2$, describing the generalization to $k > 2$ at the end of this section. Our methods are similar in spirit to those of [2]. However, we give an explicit formulation in terms of poly-local unitaries which makes the underlying representation theory apparent. In addition, the straightforward extension to all k obtainable by our methods yields an efficient algorithm for approximating the HOMFLYPT two-variable polynomial at certain pairs of points, as we outline in Section 4.3. After describing an encoding of the basis states of the representation into a system of n qubits, we show how to implement the images $\pi_n^{(2,\ell)}(\sigma_i)$ of the braid group generators σ_i using $O\big(n\log(n)\big)$ ancillary qubits.

Given a tableau $t \in T_n^{(2,\ell)}$ and an integer $1 \leq i \leq n$, recall that we write $r_i(t)$ and $c_i(t)$ for the row and column which respectively contain the number i. Notice that these numbers satisfy $r_i(t) \in \{1, 2\}$ and $1 \leq c_i(t) \leq i \leq n$ for each i. We abbreviate $r(t) = r_1(t) r_2(t) \cdots r_n(t) \in \{1, 2\}^n$ and define $c(t) \in \{1, \ldots, n\}^n$ similarly. Our first observation is that $r(t)$ uniquely specifies t. We may thus assign to each t a computational basis state of n qubits with Hilbert spaces $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n$, where each Hilbert space $\mathcal{R}_i \simeq \mathbb{C}^2$ is spanned by computational basis states $\{|1\rangle^{\mathcal{R}_i}, |2\rangle^{\mathcal{R}_i}\}$. Together, the qubits have a combined Hilbert space which we write $\mathcal{R}^n = (\mathbb{C}^2)^{\otimes n}$, where we write tensor products of Hilbert spaces as $\mathcal{R}_i^j \equiv \mathcal{R}_i \mathcal{R}_{i+1} \cdots \mathcal{R}_j \equiv \bigotimes_{k=i}^j \mathcal{R}_i$ and abbreviate $\mathcal{R}_1^j \equiv \mathcal{R}^j$. This yields an embedding $|t\rangle \mapsto |r(t)\rangle^{\mathcal{R}^n}$ of $V_n^{(2,\ell)}$ into $\mathcal{R}^n$ whose image we call the *computational subspace*. We will now show that after the addition of $O\big(n\log(n)\big)$ ancillary qubits prepared in a computational basis state, it is possible to perfectly simulate each $\pi_n^{(2,\ell)}(\sigma_i)$ on the computational subspace of $\mathcal{R}^n$.

It will be convenient for us to add ancillary systems of dimension growing linearly with n to the standard circuit model. This clearly adds no additional computational power because an arbitrary transformation of $O(\log(n))$ qubits can be generated by $O(\mathrm{poly}(n))$ local gates. We may thus think of and treat such slowly growing subsystems as local systems in their own right.

We now introduce, for each $1 \leq i \leq n$, a register with Hilbert space $\mathcal{D}_i$, which will store the value $d_i(t)$ defined in (3.13) for each tableau. Because $|d_i(t)| \leq n$, we let $\mathcal{D}_i$ be spanned by basis states $\{|{-n}\rangle^{\mathcal{D}_i}, |{-n}{+}1\rangle^{\mathcal{D}_i}, \ldots, |n\rangle^{\mathcal{D}_i}\}$. We may now define an isometry $W\colon \mathcal{R}^n \to \mathcal{R}^n\mathcal{D}^n$ which coherently computes these values for a given encoding of a tableau as

$$W|r(t)\rangle^{\mathcal{R}^n} = |r(t)\rangle^{\mathcal{R}^n}|d(t)\rangle^{\mathcal{D}^n}. \tag{4.1}$$

Below, we will show that this isometry can be implemented efficiently. For each $1 \leq i < n$, define the local unitary U_i' acting on $\mathcal{R}_i\mathcal{R}_{i+1}\mathcal{D}_i$ as

$$U_i'|r_i\rangle|r_{i+1}\rangle|d\rangle \quad = \quad \alpha_d|r_i\rangle|r_{i+1}\rangle|d\rangle + \beta_d|r_{i+1}\rangle|r_i\rangle|{-d}\rangle,$$

where $\alpha_d = q - (1+q)a_\ell(d)$ and $\beta_d = -(1+q)\sqrt{a_\ell(d)(1-a_\ell(d))}$. Define the unitaries $U_i = W^{-1}U_i'W$ for $1 \leq i < n$. By construction, the map $\sigma_i \mapsto U_i$ defines a (degenerate) unitary representation $\rho_n^{(2,\ell)}$ of B_n which is isomorphic to $\pi_n^{(2,\ell)}$. Suppose now that we are given a braid $b = \sigma_{i_m}^{x_m}\cdots\sigma_{i_2}^{x_2}\sigma_{i_1}^{x_1} \in B_n$. The corresponding unitary

$$\begin{aligned} \sigma_{i_m}^{x_m}\cdots\sigma_{i_2}^{x_2}\sigma_{i_1}^{x_1} \quad &\mapsto \quad U_{i_m}^{x_m}\cdots U_{i_2}^{x_2}U_{i_1}^{x_1} \\ &= \quad W^{-1}U_{i_m}'^{x_m}\cdots U_{i_2}'^{x_2}U_{i_1}'^{x_1}W \end{aligned}$$

therefore evolves the computational subspace of $\mathcal{R}^n$ according to the following commutative diagram

$$\begin{tikzcd} V_n^{(2,\ell)} \arrow[r, "\pi_n^{(2,n)}(b)"] \arrow[d] & V_n^{(2,\ell)} \\ \mathcal{R}^n \arrow[r, "\rho_n^{(2,\ell)}(b)"'] & \mathcal{R}^n. \arrow[u] \end{tikzcd}$$

Now we will show that the isometry W in (4.1) can be efficiently implemented. For this, we introduce n systems with joint Hilbert space $\mathcal{C}^n$, where each $\mathcal{C}_i$ is spanned by computational basis states $\{|0\rangle^{\mathcal{C}_i}, \ldots, |n\rangle^{\mathcal{C}_i}\}$. The purpose of these systems is to store the column numbers of each i. In addition, we introduce two registers $\mathcal{L}_1$ and $\mathcal{L}_2$ with bases $\{|0\rangle, \ldots, |n\rangle\}$ which will store the *shape* of the tableau. Given a qubit encoding $|r(t)\rangle^{\mathcal{R}^n}$ of a tableau t, we define the isometry $W'\colon \mathcal{R}^n \to \mathcal{R}^n\mathcal{C}^n\mathcal{L}^2$ as

$$W'|r(t)\rangle^{\mathcal{R}^n} = |r(t)\rangle^{\mathcal{R}^n}|c(t)\rangle^{\mathcal{C}^n}|\lambda_1(t)\rangle^{\mathcal{L}_1}|\lambda_2(t)\rangle^{\mathcal{L}_2},$$

where $\lambda_j(t)$ denotes the number of boxes in the j'th row of t. This isometry can be efficiently implemented by first preparing the additional systems in the state $|00\cdots0\rangle^{\mathcal{C}^n}|00\rangle^{\mathcal{L}^2}$, after which the composition of local gates $W_n'\cdots W_2'W_1'$ is applied, where each W_i' acts on $\mathcal{R}_i\mathcal{C}_i\mathcal{L}^2$ as

$$W_i'|r\rangle^{\mathcal{R}_i}|0\rangle^{\mathcal{C}_i}|\lambda_1\lambda_2\rangle^{\mathcal{L}^2} = \begin{cases} |r\rangle^{\mathcal{R}_i}|\lambda_1{+}1\rangle^{\mathcal{C}_i}|\lambda_1{+}1\rangle^{\mathcal{L}_1}|\lambda_2\rangle^{\mathcal{L}_2} & r = 1 \\ \\ |r\rangle^{\mathcal{R}_i}|\lambda_2{+}1\rangle^{\mathcal{C}_i}|\lambda_1\rangle^{\mathcal{L}_1}|\lambda_2{+}1\rangle^{\mathcal{L}_2} & r = 2. \end{cases}$$

For each $1 \leq i < n$, we then define local unitaries W_i which act on $\mathcal{R}_i\mathcal{R}_{i+1}\mathcal{C}_i\mathcal{C}_{i+1}\mathcal{D}_i$ as

$$W_i|r_i\rangle|r_{i+1}\rangle|c_i\rangle|c_{i+1}\rangle|0\rangle = |r_i\rangle|r_{i+1}\rangle|c_i\rangle|c_{i+1}\rangle|d_i\rangle,$$

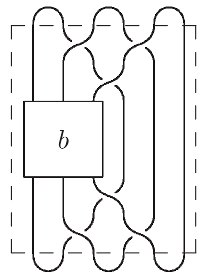


Figure 4.1: The trace closure of $b \in B_3$ is isotopic to the plat closure of the modified braid $(\sigma_3^5\sigma_2^3)^{-1}b\sigma_3^5\sigma_2^3 = \sigma_2^{-1}\sigma_4^{-1}\sigma_3^{-1}b\sigma_3\sigma_4\sigma_2$. Note that we identify b with its image $\iota \circ \iota \circ \iota(b) \in B_6$.

where $d_i = c_{i+1} - c_i - (r_{i+1} - r_i)$. Finally, observe that we may write $W = W_{n-1} \cdots W_2 W_1 W'$, implying that the $\mathcal{D}^n$ registers can be prepared efficiently.

To adapt the above construction to the $k > 2$ case, it is sufficient to increase the dimension of each $\mathcal{R}_i$ to n, while using n $\mathcal{L}_i$ registers to store the shape of an arbitrary n-box (k, ℓ)-diagram. The matrices W and U_i' generalize in a straightforward manner. It is clear that regardless of the values of k and ℓ, the resulting U_i matrices consist of $O\big(\mathrm{poly}(n)\big)$ local gates.

4.3 The algorithms

In this section, we begin with two different algorithms for sampling from an additive approximation of the Jones polynomial of the generalized closure of a braid. Our first algorithm uses the fact that any generalized closure of a braid can be presented as the plat closure of a related braid. Our second resembles that given in [2], although our more general implementation of the Jones-Wenzl representations of the braid group allows us to give an adaptation which approximates the HOMFLYPT polynomial of the trace closure of a braid as well.

Given is a braid $b \in B_n$, two nonnegative integers p, r for which $2p + r = n$ and a positive integer ℓ. We will lose no generality in assuming that the top and bottom braids which define the generalized closure are $x, y = 1_n$. Since the largest absolute value of the Jones polynomial is obtained when the closure yields $p+r$ unknots, we will focus on estimating the normalized quantity

$$\frac{1}{[2]_\ell^{p+r-1}} \left| J\big(\xi_{1,1}^{p,r}(b), e^{2\pi i/\ell}\big) \right|.$$

Algorithm I

Our first algorithm relies on the following proposition, which shows that the generalized closure $\xi^{p,r}(b)$ can be written as the plat closure of a related braid on $2p + 2r$ strands.

Proposition 4.4. *Let $b \in B_{2p+r}$. Then $\xi_{1,1}^{p,r}(b) = \widetilde{c^{-1}bc}$, where $c = \sigma_{2p+r}^{2p+2r-1} \cdots \sigma_{2p+3}^{2p+5}\sigma_{2p+2}^{2p+3}$ and for $i < j$, we write $\sigma_i^j = \sigma_i\sigma_{i+1}\cdots\sigma_{j-2}\sigma_{j-1}$ for the braid where strand j is moved above its neighboring strands on the left and is inserted in position i.*

While we invite the reader to verify this, the main idea is contained in Figure 4.1, where we illustrate how the trace closure can be obtained from the plat closure. Note that since the braid c in the proposition consists of $\sum_{i=1}^{t-1} i = O(t^2)$ generators, the modified braid $b' \equiv c^{-1}bc$ is at most polynomially longer than b itself. Setting $p' = p + r$, observe that since $b' \in B_{2p'}$, we have the following chain of equalities:

$$\begin{aligned}\frac{1}{[2]_\ell^{p+r-1}}\left|J\big(\xi_{1,1}^{p,r}(b), e^{2\pi i/\ell}\big)\right| &= \frac{1}{[2]_\ell^{p'-1}}\left|J\big(b', e^{2\pi i/\ell}\big)\right| \\ &= \left|\langle t_{2p'}|\pi^{(2,\ell)}_{[p',p']}(b')|t_{2p'}\rangle\right| \\ &= \left|{}^{\mathcal{R}^{2p'}}\!\langle 0101\cdots 01|\rho^{(2,\ell)}(b')|0101\cdots 01\rangle^{\mathcal{R}^{2p'}}\right|\end{aligned}$$

Here, the second line uses (3.38), while the last uses the construction of Section 4.2 to express the desired quantity in a form suitable for a quantum computer. In order to sample from a binary random variable Z with the above expectation, one first prepares the state $|0101\cdots 01\rangle^{\mathcal{R}^{2p'}}$, then evolves it according to $\rho^{(2,\ell)}(b')$ using the local qubit implementation described in Section 4.2. Next, each qubit is measured in the computational basis; if the result is the string $0101\cdots 01$ one sets $Z=1$, otherwise setting $Z=0$. This procedure is then repeated a suitable number of times, after which the average of the obtained samples is output.

Algorithm II

This algorithm makes use of the following representation-theoretic formula for the absolute value of the Jones polynomial:

$$\begin{aligned}\frac{1}{[2]_\ell^{p+r-1}}\left|J\big(\xi_{1,1}^{p,r}(b), e^{2\pi i/\ell}\big)\right| &= \frac{[2]_\ell^{p-1}}{[2]_\ell^{p+r-1}}\left|\sum_{\lambda\in\Lambda_r^{(2,\ell)}}[\lambda_1-\lambda_2+1]_\ell \operatorname{Tr}\left(\pi^{(2,\ell)}_{[p,p]+\lambda}(b)(\varphi_{2p}\otimes 1_\lambda)\right)\right| \\ &= \frac{1}{[2]_\ell^{r}}\left|\sum_{\lambda\in\Lambda_r^{(2,\ell)}}[\lambda_1-\lambda_2+1]_\ell \operatorname{Tr}\left(\pi^{(2,\ell)}_{[p,p]+\lambda}(b)(\varphi_{2p}\otimes 1_\lambda)\right)\right|. \qquad (4.2)\end{aligned}$$

We will require the following lemma.

Lemma 4.5 (Sampling Lemma). *Let U be a quantum circuit of length $O\big(\mathrm{poly}(n)\big)$ acting on n qubits, and let $|\psi\rangle$ be a pure state of the n qubits which can be prepared in time $O\big(\mathrm{poly}(n)\big)$. It is then possible to sample from random variables $X, Y \in \{\pm 1\}$ for which*

$$\mathbb{E}[X+iY] = \langle\psi|U|\psi\rangle$$

in $O\big(\mathrm{poly}(n)\big)$ time.

Proof of Lemma 4.5. Introduce an extra system labeled $\mathcal{C}$ which holds a single control qubit and denote the Hilbert space of the qubits acted on by U by $\mathcal{A}$. Because U consists of $O\big(\mathrm{poly}(n)\big)$ gates, then so does the controlled unitary $V: \mathcal{AC}\to\mathcal{AC}$, defined by $V|\psi\rangle|0\rangle = |\psi\rangle|0\rangle$ and $|\psi\rangle|1\rangle = (U|\psi\rangle)|1\rangle$. Initialize the qubit in the state $\frac{1}{\sqrt{2}}\big(|0\rangle+|1\rangle\big)^{\mathcal{C}}$, and prepare the state $|\psi\rangle^{\mathcal{A}}$. Applying the controlled unitary places everything into the state

$$|\Psi\rangle^{\mathcal{AC}} = \frac{1}{\sqrt{2}}\big(|\psi\rangle|0\rangle + (U|\psi\rangle)|1\rangle\big).$$

Writing $\psi = |\psi\rangle\langle\psi|$, the reduced density matrix $\tau^{\mathcal{C}}$ of the extra qubit is then equal to

$$\begin{aligned}
\tau^{\mathcal{C}} &= \mathrm{Tr}_{\mathcal{A}}\, |\Psi\rangle\langle\Psi|^{\mathcal{CA}} \\
&= \frac{1}{2}\,\mathrm{Tr}_{\mathcal{A}} \begin{pmatrix} \psi & \psi U^\dagger \\ U\psi & U\psi U^\dagger \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} 1 & \langle\psi|U^\dagger|\psi\rangle \\ \langle\psi|U|\psi\rangle & 1 \end{pmatrix} \\
&= \frac{1}{2}\Big(1_2 + \sigma_X \mathrm{Re}\langle\psi|U|\psi\rangle + \sigma_Y \mathrm{Im}\langle\psi|U|\psi\rangle\Big).
\end{aligned}$$

Since the ± 1 eigenstates of σ_X are $|\pm\rangle = \frac{1}{2}\big(|0\rangle \pm |1\rangle\big)$, measuring σ_X on the qubit yields a classical random variable $X \in \{+1, -1\}$ for which $\mathbb{E}\, X = \mathrm{Re}\langle\psi|U|\psi\rangle$ as required. Similarly, by running the algorithm a second time, we can sample from Y, which satisfies $\mathbb{E}\, Y = \mathrm{Im}\langle\psi|U|\psi\rangle$. ■

We begin by remarking that according to (3.18), the numbers $\Big\{P_\lambda = s_\lambda^{(2,\ell)} d_\lambda^{(2,\ell)} : \lambda \in \Lambda_r^{(2,\ell)}\Big\}$ sum to unity and thus define probabilities. The algorithm proceeds as follows:

1. Select an irrep $\lambda \in \Lambda_r^{(2,\ell)}$ at random with probability $P_\lambda \equiv s_\lambda^{(2,\ell)} d_\lambda^{(2,\ell)}$.
2. Select a path t in the Young graph $\Lambda_\lambda^{(2,\ell)}$ uniformly at random.
3. Using the construction in the proof of Lemma 4.5, sample from random variables $X, Y \in \{\pm 1\}$ with conditional expectations satisfying
$$\mathbb{E}[X + iY|\lambda, t] = \langle t_{2p}|\langle t|\pi_{[p,p]+\lambda}^{(2,\ell)}(b)|t_{2p}\rangle|t\rangle.$$
4. Repeat the procedure, averaging the absolute values of the results of step 3 until the desired precision is attained.

The (nonconditional) expected value of $X + iY$ at step 3 is

$$\begin{aligned}
\mathbb{E}[X+iY] &= \sum_{\lambda \in \Lambda_r^{(2,\ell)}} P_\lambda \frac{1}{d_\lambda^{(2,\ell)}} \sum_{t \in T_\lambda^{(2,\ell)}} E[X + iY|\lambda, t] \\
&= \sum_{\lambda \in \Lambda_r^{(2,\ell)}} s_\lambda^{(2,\ell)} \sum_{t \in T_\lambda^{(2,\ell)}} \langle t_{2m}|\langle t|\pi_{[p,p]+\lambda}^{(2,\ell)}(b)|t_{2m}\rangle|t\rangle \\
&= \sum_{\lambda \in \Lambda_r^{(2,\ell)}} s_\lambda^{(2,\ell)}\, \mathrm{Tr}\left[\left(\sum_{t \in T_\lambda^{(2,\ell)}} \varphi_{2p} \otimes |t\rangle\langle t|\right) \pi_{[p,p]+\lambda}^{(2,\ell)}(b)\right] \\
&= \sum_{\lambda \in \Lambda_r^{(2,\ell)}} s_\lambda^{(2,\ell)}\, \mathrm{Tr}\Big[(\varphi_{2p} \otimes 1_\lambda)\pi_{[p,p]+\lambda}^{(2,\ell)}(b)\Big] \\
&= \frac{1}{[2]_\ell^r} \sum_{\lambda \in \Lambda_r^{(2,\ell)}} [\lambda_1 - \lambda_2 + 1]_\ell\, \mathrm{Tr}\Big[(\varphi_{2p} \otimes 1_\lambda)\pi_{[p,p]+\lambda}^{(2,\ell)}(b)\Big].
\end{aligned}$$

In particular, $\mathbb{E}\Big[|X + iY|\Big]$ is precisely equal to the value (4.2) that we wish to estimate. We are now in a position to prove that the above algorithms solve **Approximate Jones Plat** in $O\big(\mathrm{poly}(m, 1/\delta)\big)$ time. As the proof for the first algorithm only requires a trivial application of the following version of the Chernoff bound, we will only prove that the second algorithm works as required.

Lemma 4.6 (Chernoff Bound). *Let* $\{Z_1, \ldots, Z_M\}$ *be random variables satisfying* $0 \le Z_j$, $\mathbb{E}\, Z_j = \mu$, *and* $\mathbb{E}\, Z_j^2 \le \kappa$. *Then*

$$\Pr\left\{\left|\frac{1}{M}\sum_{j=1}^{M} Z_j - \mu\right| > \delta\right\} \le 2\exp\left(-\frac{M\delta^2}{4\kappa}\right).$$

Proof of Theorem 4.1. First, note that there are only a polynomial number of weights, so they can be sampled from efficiently. Second, we remark that the paths can be chosen efficiently; this was shown in [2], although we mention below, in the context of approximating the HOMFLYPT polynomial, why this is true in the more general case. Suppose the above algorithm is run M times, obtaining random variables $\{X_j, Y_j\}_{j=1}^{M}$. Setting $Z_j = |X_j + iY_j|$ is is clear that $\mathbb{E}\, Z_j^2 \le \mathbb{E}\, X_j^2 + \mathbb{E}\, Y_j^2 = 2$. Thus, for any $\delta > 0$, the probability that the empirical mean $\frac{1}{M}\sum Z_j$ deviates from the desired value of the normalized Jones polynomial by more than δ is smaller than $2\exp(-M\delta^2/8)$, which is smaller than the desired error of 1/4 provided that the number of samples satisfies $M > 8\ln(2)/\delta^2$. Since each sample is obtained in $O\big(\mathrm{poly}(m)\big)$ time, the desired additive approximation is obtained in $O\big(\mathrm{poly}(m)/\delta^2\big) = O\big(\mathrm{poly}(m, 1/\delta)\big)$ time as required. Note that the running time is independent of ℓ, ■

Approximating the HOMFLYPT of the trace closure

A simple modification of the second algorithm above suffices to approximate the absolute value of the HOMFLYPT two-variable polynomial of the trace closure of a braid, evaluated at certain pairs of points. The restriction here to the absolute value is unnecessary; we focus on it for simplicity. We will focus on estimating the following normalized version of (3.22), for $b \in B_n$:

$$\begin{aligned}(-\nu_{k\ell})^{n-1}\left|H_{\widehat{b}}(e^{2\pi i/\ell}, \theta_{k\ell}e^{\pi i/\ell})\right| &= (-\nu_{k\ell})^{n-1}\left|\overline{\mathrm{tr}}_{k,\ell}(b)\right| \\ &= \sum_{\lambda \in \Lambda_n^{(k,\ell)}} s_\lambda^{(k,\ell)} \,\mathrm{Tr}\, \pi_\lambda^{(k,\ell)}(b).\end{aligned}$$

To sample from a ± 1 random variable with the above expectation, the same procedure as used to sample from the Jones polynomial is utilized. We will restrict our attention to the case where k is a fixed constant which does not grow with the number of crossings m or the number of strands n. By (3.18), the numbers $\left\{P_\lambda = s_\lambda^{(k,\ell)}\, d_\lambda^{(k,\ell)} : \lambda \in \Lambda_n^{(k,\ell)}\right\}$ are probabilities, so the algorithm proceeds as before. As with the algorithm for the Jones polynomial, the number of weights grows polynomially with n, so choosing a $\lambda \in \Lambda_n^{(k,\ell)}$ with probability P_λ can be done efficiently because those probabilities can be precomputed and sampled from by a standard procedure. Perhaps the only extra point needing explanation is the possibility of choosing a path $t \in T_\lambda^{(k,\ell)}$ uniformly at random. This can be done efficiently because the Young graph $\Lambda_{0,n}^{(k,\ell)}$ is a layered graph, with a polynomial number of diagrams $\le \big(\min\{n, \ell - k\} + 1\big)^k$ in each layer. Therefore, the number of paths from $\emptyset$ to a given node in the diagram can be computed in advance. To choose a path ending in a particular diagram $\lambda \in \Lambda_n^{(k,\ell)}$, the trick is to move in reverse; starting at λ, choose the next diagram $\lambda' \in \Lambda_{n-1}^{(k,\ell)}$ with a probability proportional to the number of paths from $\emptyset$ to λ', divided by the total number of paths from $\emptyset$ ending in the $n-1$'st layer.

5 Simulating local quantum circuits with braids

In this section, we show how to "compile" any quantum circuit on n qubits consisting of m two-qubit gates into a description of a braid on $4n$ strands with $\mathrm{poly}(m)$ crossings in such a way that

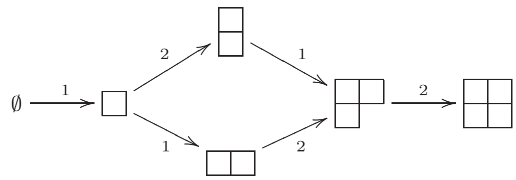


Figure 5.1: Qubit path basis for W_1. The upper path corresponds to the basis state $|t_0\rangle$, the lower path to $|t_1\rangle$.

the output of the circuit is encoded into an evaluation of the Jones polynomial of the plat closure of the braid at any primitive ℓ'th root of unity with $\ell \geq 5, \ell \neq 6$. While this was first proved in [12], our proof is simpler as it only uses the representation of B_8 corresponding to the rectangular diagram $\boxplus\boxplus$. After we introduce our encoding of circuits into braids, we will prove the following theorem.

Theorem 5.1. *Let $U = U_m U_{m-1} \cdots U_1$ be a quantum circuit of length m acting on n qubits and let $\epsilon = \Omega\big(2^{-\mathrm{poly}(n)}\big)$. Then there is a braid $b \in B_{4n}$ with $O\big(\mathrm{poly}(m)\big)$ crossings such that U is ϵ-approximated by $\pi^{(2,\ell)}_{[2n,2n]}(b)$. In this case, the following inequality holds:*

$$\left| |\langle 00\cdots 0|U|00\cdots 0\rangle|^2 - \frac{1}{[2]_\ell^{2n-1}} \left| J(\widetilde{b}, e^{2\pi i/l}) \right|^2 \right| \leq \epsilon \,. \tag{5.1}$$

Moreover, such a braid can be found from a description of the circuit in $O\big(\mathrm{poly}(m)\big)$ time on a classical computer.

Throughout this section, we fix an $\ell \geq 5, \ell \neq 6$ and for each diagram λ, we abbreviate $\pi_\lambda \equiv \pi^{(2,\ell)}_\lambda$ and $V_\lambda = V^{(2,\ell)}_\lambda$. We begin by describing how to encode a single logical qubit into $V_{\boxplus}$. This is a two-dimensional representation of B_4 with orthonormal basis

$$|t_0\rangle \equiv \left| \begin{smallmatrix} 1 & 3 \\ 2 & 4 \end{smallmatrix} \right\rangle \qquad\qquad |t_1\rangle \equiv \left| \begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right\rangle.$$

These basis states may also be viewed as paths in Young's lattice, as pictured in Figure 5.1. Let t and t' be $(2,\ell)$-tableaux of respective shapes $[j,j]$ and $[k,k]$. We will denote by tt' the tableau of shape $[j+k, j+k]$ obtained by adding $2j$ to each box in t' then placing it to the right of t. Central to our proof is that tt' is again a $(2,\ell)$-tableaux, as we remarked in Section 3.1. We encode n qubits into a subspace $W^{(n)}$ of the irreducible representation $V_{[2n,2n]}$ of B_{4n} which is spanned by the 2^n computational basis states $\big\{|t_{x_1} t_{x_2} \cdots t_{x_n}\rangle : (x_1, x_2, \ldots, x_n) \in \{0,1\}^n\big\}$. The subspace $W^{(n)}$ decomposes as a tensor product $\bigotimes_{i=1}^n W_i$, where for each $1 \leq i \leq n$, we identify W_i with $V_{\boxplus}$, so that it is spanned by $|t_0\rangle$ and $|t_1\rangle$. In this subspace of $V_{[2n,2n]}$, each consecutive group of four strands is associated to a single qubit in the following way. For each $1 \leq j < k \leq n-1$, define the subset $\Sigma_j^k = \{\sigma_j, \sigma_{j+1}, \ldots, \sigma_k\}$ of generators of B_n and write $B_j^k \subseteq B_n$ for the subgroup generated by Σ_j^{k-1}. For each $1 \leq i \leq n-2$, the image of the 8 strand braid group $B^{4(i+1)}_{4(i-1)+1}$ under the representation $\pi_{[2n,2n]}$, when restricted to $W^{(n)}$, acts nontrivially only on the subsystem $W_i \otimes W_{i+1}$. In fact, it rapidly approximates any $U \in SU(W_i \otimes W_{i+1})$. To see this, it suffices to look at the case $i = 1$. Note that $W_1 \otimes W_2 \subset V_{\boxplus\boxplus}$. The embedding $V_{\boxplus} \otimes V_{\boxplus} \hookrightarrow V_{\boxplus\boxplus}$ looks like

$$
\begin{aligned}
|t_0\rangle|t_0\rangle &\mapsto \left|\begin{array}{|c|c|c|c|}\hline 1&3&5&7\\\hline 2&4&6&8\\\hline\end{array}\right\rangle\\
|t_0\rangle|t_1\rangle &\mapsto \left|\begin{array}{|c|c|c|c|}\hline 1&3&5&6\\\hline 2&4&7&8\\\hline\end{array}\right\rangle\\
|t_1\rangle|t_0\rangle &\mapsto \left|\begin{array}{|c|c|c|c|}\hline 1&2&5&7\\\hline 3&4&6&8\\\hline\end{array}\right\rangle\\
|t_1\rangle|t_1\rangle &\mapsto \left|\begin{array}{|c|c|c|c|}\hline 1&2&5&6\\\hline 3&4&7&8\\\hline\end{array}\right\rangle.
\end{aligned}
$$

While this proves that $\dim V_{\yng(4,4)} \geq 4$, we mention that its dimension is 13 when $\ell = 5$ and is 14 when $\ell > 6$. So it is clear that $SU(W_1 \otimes W_2) \subset SU(V_{\yng(4,4)})$. Consider now the following proposition.

Proposition 5.2 (Freedman, Larsen, Wang [13]). *Let $\ell \geq 5, \ell \neq 6$. The image of B_8 under $\pi^{(2,\ell)}_{\yng(4,4)}$ is dense in $SU\Big(V^{(2,\ell)}_{\yng(4,4)}\Big)$.*

Now we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. Set $\epsilon' = \epsilon/m = \Omega\big(2^{-m}/m\big)$. Together with the Solovay-Kitaev Theorem, Proposition 5.2 implies that for each $U' \in SU(W_1 \otimes W_2)$, there is a braid $b \in B_8$ of length $O\big(\mathrm{poly}\log(1/\epsilon')\big) = O\big(\mathrm{poly}(m)\big)$ for which $\pi_{\yng(4,4)}(b)$ ϵ'-approximates U', where we consider U' to be embedded in $SU(V_{\yng(4,4)})$. Note that while some unitaries $\pi_{\yng(4,4)}(\sigma_i)$ will *not* be contained in $SU(W_1 \otimes W_2)$, the construction above ensures that $\pi_{\yng(4,4)}(b)$ can be made arbitrarily close to any $U' \in SU(W_1 \otimes W_2)$.

Because of the four-periodicity of our encoding, the same arguments apply for approximating gates on any other adjacent pair of qubits. Now, let $U = U_m U_{m-1} \cdots U_1$ as in the statement of the theorem. Without loss of generality, we may assume that each U_i has unit determinant. By Lemma 4.3, we may conclude that there is a braid $b \in B_{4n}$ of length $O\big(\mathrm{poly}(m)\big)$ for which $\pi_{[2n,2n]}(b)$ ϵ-approximates the circuit U, and also that such a braid can be determined on a classical computer in $O\big(\mathrm{poly}(m)\big)$ time. The rest of the theorem now follows by Lemma 4.2 and the expression (3.38) for the absolute value of the Jones polynomial of the plat closure; the normalization constant $1/[2]^{2n-1}$ is so because the braid has $4n$ strands. ∎

6 Complexity-theoretic applications of the Jones polynomial

Complexity theory characterizes the asymptotic consumption of resources, such as time and space, required to obtain the solutions of certain classes of problems on a computer. Writing $\{0,1\}^*$ for the set of all binary strings of finite length, a *decision problem* asks for the evaluation of some function $f\colon \{0,1\}^* \to \{0,1\}$ on an arbitrary input. Such a function determines a *language* $L \subseteq \{0,1\}^*$ defined as $L = f^{-1}(1)$, so equivalently, the task is to determine, for each bit string x^n, whether $x^n \in L$. Examples of languages include the set of all binary strings of even parity and the set of prime numbers, written in binary. A *complexity class* is a collection of computational problems, usually defined in terms of the resources required to determine membership of arbitrarily long bit strings in each language contained in that class. For instance, any class of languages is a complexity class. Two well-known complexity classes are P and NP. The former is defined to contain every language $L \subseteq \{0,1\}^*$ for which there is a uniform family $\big\{C_n\colon \{0,1\}^n \to \{0,1\}\big\}$ of polynomial-size classical circuits (see e.g. [29, 30] for a definition of a classical circuit) for which $C_n(x^n) = 1$ when $x^n \in L$, while $C_n(x^n) = 0$ otherwise. By a *uniform family*, it is meant that there should be a classical algorithm which, given the number n as input, produces a description of the circuit C_n in $O(\mathrm{poly}(n))$-time. We mention that this definition of a uniform family of circuits extends to quantum circuits in the obvious way. NP, on the other hand, contains those languages L for which there is a uniform family of classical circuits $\{C_n \colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}\}$, where the number m of *guess bits* grows polynomially with n, so that for each $x^n \in L$, there exists a *witness string*, or *certificate* y^m, so that $C_n(x^n, y^m) = 1$ if $x^n \in L$, while if $x^n \notin L$, $C_n(x^n, y^m) = 0$ for every possible certificate y^m. Below, we will recall the definitions of certain quantum generalizations of these classes, showing how they may be characterized using the Jones polynomial.

The complexity class BQP is defined to contain those decision problems whose solution can be determined with bounded error in polynomial time in the *standard quantum circuit model*. Formally, a language L is in BQP if there exists is a uniform family of quantum circuits $\{U_n\}$ of polynomial length such that if $x^n \in L$, a measurement of the first qubit of $U_n|x^n 00\cdots 0\rangle$ in the computational basis yields 1 with probability $\geq 3/4$ while if $x^n \notin L$, that measurement outcome will be 0 with probability $\geq 3/4$. By an elementary application of the Chernoff bound, the success probability $3/4$ can be amplified to be exponentially close to 1 by running the algorithm polynomially many times in n and utilizing a majority voting procedure. This definition of BQP is thus insensitive to the exact value taken by the success probability, provided that it is bounded above $1/2$ by a constant which is *independent* of the input string x^n. By an "in place amplification" result from [26], such a majority voting procedure can be implemented using a modification U_n' of U_n which acts only on polynomially many more qubits, while having only polynomially many more gates, in such a way that the amplified output is encoded into the probabilities associated to measuring just a single qubit. In addition, using a trick described in Figure 3 of [9], it is possible to modify a sufficiently amplified circuit U_n', obtaining a new circuit U_n'' which computes $U_n'|x^n 00\cdots 0\rangle$, copies the output qubit to yet another qubit using a controlled-not gate, then uncomputes U_n', in such a way that the squared matrix element $|\langle x^n 00\cdots 0|U_n''|x^n 00\cdots 0\rangle|^2$ is arbitrarily close to the probability of obtaining a 1 when measuring the output qubit of $U_n'|x^n 00\cdots 0\rangle$. Furthermore, the class BQP remains unchanged if the particular universal gate set chosen for the model merely contains a universal set of gates for each pair of qubits. This follows by the Solovay-Kitaev Theorem, together with the remarks immediately following its statement above. The following theorem essentially appears in [7]; we prove it in this paper for the sake of completeness.

Theorem 6.0. *Let A be a random oracle which solves* **Approximate Jones Closure** *with $\delta = 1/8$ for the plat closure and some constant $\ell \geq 5, \ell \neq 6$. Then $\mathrm{P}^A = \mathrm{BQP}$. Loosely speaking, this means that* **Approximate Jones Closure** *is "BQP-complete."*

Proof. To begin, it is an immediate consequence of Theorem 4.1 that $\mathrm{P}^A \subseteq \mathrm{BQP}$. To see that $\mathrm{BQP} \subseteq \mathrm{P}^A$, let $L \in \mathrm{BQP}$, and let U be n'th quantum circuit in the uniform family of circuits for L. By definition, U is of length $m = O\big(\mathrm{poly}(n)\big)$ and acts on $n' = O\big(\mathrm{poly}(n)\big)$ qubits. For each x^n, define the circuit $W_{x^n} = \bigotimes_{i=1}^n X^{x_i}$, which acts as the identity on the ancillary qubits. Because

$$\langle 00\cdots 0|W_{x^n} U W_{x^n}|00\cdots 0\rangle = \langle x^n 00\cdots 0|U|x^n 00\cdots 0\rangle,$$

it follows by Theorem 5.1 that for any constant $\epsilon > 0$, there is a braid b of length $O\big(\mathrm{poly}(m)\big) = O\big(\mathrm{poly}(n)\big)$ on $4n'$ strands which satisfies

$$\left| \frac{1}{[2]_\ell^{2n'-1}} \left| J(\widetilde{b}, e^{2\pi i/\ell}) \right|^2 - |\langle x^n 00\cdots 0|U|x^n 00\cdots 0\rangle|^2 \right| \quad \leq \quad \epsilon.$$

Moreover, there is an $O\big(\mathrm{poly}(n)\big)$-time classical algorithm for determining such a braid from a description of $W_{x^n} U W_{x^n}$. By invoking the oracle A, we obtain a random variable Z satisfying

$$\Pr\left\{ \left| Z - \frac{1}{[2]_\ell^{2n'-1}} \left| J(\widetilde{b}, e^{2\pi i/\ell}) \right| \right| < 1/8 \right\} > 3/4.$$

Since $|a^2 - b^2| \leq 2|a-b|$ for $0 \leq a, b \leq 1$, the last two estimates can be combined via the triangle inequality to show that $\left|Z^2 - |\langle x^n 00\cdots 0|U|x^n 00\cdots 0\rangle|^2\right| < 1/4 + \epsilon$ with probability at least 3/4. Choosing $\epsilon = 1/8$ is sufficient to complete the proof. ∎

The complexity class QCMA is a certain quantum analog of NP, consisting of those languages L for which a computationally unbounded oracle (generally personified as Merlin) can, with high probability, efficiently convince a computationally bounded verifier (Arthur) that each $x^n \in L$ is actually contained in L. Cheating is also discouraged, in that if $x^n \notin L$, the probability that Merlin will convince Arthur otherwise should be small. Formally, we say that a language L is in QCMA if there is a uniform family of quantum circuits $\{U_n\}$ of polynomial length such that for each length n string $x^n \in L$, there is a length m binary string y^m for which $P(x^n, y^m) \equiv |\langle x^n y^m 00\cdots 0|U_n|x^n y^m 00\cdots 0\rangle|^2 \geq 3/4$, while if $x^n \notin L$, we have $P(x^n, y^m) \leq 1/4$ for every such y^m. We remark that the usual definition of QCMA (see e.g. [3] for the original definition) is phrased in terms of measurement probabilities of a single output qubit; by the remarks following the definition above of BQP, the definition we give here is equivalent. For the same reasons as with BQP, the exact value 3/4 of the success probability is not important; what is important is that the square of the matrix element is *strictly* bounded away from 1/2.

Given a complexity class C, a problem is said to be C-*hard* if it is at least as hard as any other problem in the class C, in the sense that access to an oracle which immediately computes the solution of that problem allows the solution of any problem in the class C in polynomial-time. If that problem is also contained in the class C, we say that it is C-*complete*. We will show that the following problem is QCMA-complete.

Problem 6.1 (Increase Jones Plat). Given a braid $b \in B_{2n}$ with $O\big(\mathrm{poly}(n)\big)$ crossings, a fixed integer $\ell \geq 5$, $\ell \neq 6$, and a class of braids $\mathcal{C}_{2n} \subset B_{2n}$ for which membership can be decided in $O\big(\mathrm{poly}(n)\big)$ time on a classical computer, decide, with the promise that only these two cases can occur, whether there exists another braid $c \in \mathcal{C}_{2n}$ for which the absolute value of the normalized Jones polynomial of the plat closure

$$\frac{1}{[2]_\ell^{n-1}} \left| J(\widetilde{cbc^{-1}}, e^{2\pi i/\ell}) \right|^2 \geq 3/4$$

or if it is $\leq 1/4$ for all braids $c \in \mathcal{C}_{2n}$.

Theorem 6.1. Increase Jones Plat *is* QCMA-*complete.*

Proof. To begin, we demonstrate that **Increase Jones Plat** is in QCMA. Given a braid $b \in B_{2n}$ of length $O\big(\mathrm{poly}(n)\big)$, Merlin begins by sending Arthur the classical description of the "witness" braid c. Next, Arthur checks to see if $c \in \mathcal{C}_{2n}$, which can be done in polynomial-time by definition. Because of the promise, Arthur only needs to learn the value of

$$\frac{1}{[2]^{n-1}} \left| J(\widetilde{c^{-1}bc}, e^{2\pi i/\ell}) \right|^2$$

with an accuracy of $< 1/2$ with probability $\leq 3/4$. For this, it is sufficient for him to run **Approximate Jones Closure** on a quantum computer with $\delta = 1/4$, which takes $O\big(\mathrm{poly}(n)\big)$-time.

To see that this problem is QCMA-hard, and is thus QCMA-complete, suppose that $L \in$ QCMA. Let U be the n'th circuit from the uniform family of quantum verifiers for L, and suppose that U acts on n' qubits while accepting witnesses of size m. Letting $\epsilon > 0$ be a constant to be determined at the end of the proof, Arthur precomputes a set of four braids $\mathcal{C}_8 \subset B_8$ which ϵ/m-approximate the circuits $\big\{X^{y_1} \otimes X^{y_2} : y_1, y_2 \in \{0,1\}\big\}$ (obviously, there is nothing to compute when $y_1 = y_2 = 0$). This can be done in polynomial-time. Define $\mathcal{C}_{4n'} \subset B_{4n'}$ to consist of braids which act as the identity everywhere except for the strands $4n+1, \ldots 4n+4m$, where only a single braid from $\mathcal{C}_8$ can act on each group of eight strands $8i+1, \ldots, 8i+8$ within those strands. If we define V_{y^m} in an analogous manner to W_{x^n} in the proof of Theorem 6.0, where it acts only by flipping the witness qubits, it follows by Lemma 4.3 that for each braid c in $\mathcal{C}_{4n'}$, the corresponding unitary $\pi^{(2,\ell)}_{[2n',2n']}(c)$ ϵ-approximates V_{y^m} for one and only one y^m (this is because $\|V_{y^m} - V_{z^m}\|_\infty = 2$ whenever $y^m \neq z^m$). Arthur then tells x^n to Merlin, who sends him a witness braid c from $\mathcal{C}_{4n'}$. Define W_{x^n} as in the proof of Theorem 6.0, except that it acts as the identity on the witness and ancillary qubits. By the first half of Theorem 5.1, Arthur may compute a braid b for which the circuit $W_{x^n} U W_{x^n}$ is ϵ-approximated by $\pi^{(2,\ell)}_{[2n',2n']}(b)$. Therefore, we obtain by Lemma 4.3 that $\pi^{(2,\ell)}_{[2n',2n']}(c^{-1}bc)$ is a 3ϵ-approximation of $W_{x^n} V_{y^m} U W_{x^n} V_{y^m}$. Since

$$\langle 00\cdots 0|W_{x^n} V_{y^n} U W_{x^n} V_{y^n}|00\cdots 0\rangle = \langle x^n y^m 00\cdots 0|U|x^n y^m 00\cdots 0\rangle,$$

we may apply the rest of Theorem 5.1 to obtain that

$$\left| \left|\langle x^n y^m 00\cdots 0|U|x^n y^m 00\cdots 0\rangle\right|^2 - \frac{1}{[2]^{2n'-1}} \left| J(\widetilde{c^{-1}bc}, e^{2\pi i/\ell})\right|^2 \right| \leq 3\epsilon.$$

In order to complete the proof, notice that because of the promise, it is sufficient to choose $\epsilon = 1/8$. ∎

A rather large complexity class is PSPACE, which contains all languages which can be decided using only polynomial space. In [40], PSPACE is characterized as the class of languages which can be decided by applying the same polynomial-size circuit (possibly exponentially) many times. Formally, it is shown there that if $L \in$ PSPACE, then there exists a uniform family of polynomial-size quantum circuits $\{U_n\}$, together with a sequence of polynomial-time computable natural numbers e_n, for which, if $f\colon \{0,1\}^* \to \{0,1\}$ is such that $f^{-1}(1) = L$, then

$$\langle x^n 100\cdots 0|U_n^{e_n}|x^n 100\cdots 0\rangle = f(x^n).$$

Using this characterization, we will show that the following problem is PSPACE-complete.

Problem 6.2 (Approximate Concatenated Jones Plat). Given a braid $b \in B_{2n}$ with $O\big(\mathrm{poly}(n)\big)$ crossings, a constant ℓ satisfying $\ell \geq 5$, $\ell \neq 6$, and a positive integer e which is describable by $O\big(\mathrm{poly}(n)\big)$ bits, decide, with the promise that only these two cases can occur, whether the squared absolute value of the normalized Jones polynomial of the plat closure of the concatenated braid b^e satisfies

$$\frac{1}{[2]_\ell^{n-1}} \left| J(\widetilde{b^e}, e^{2\pi i/\ell})\right|^2 \geq 3/4$$

or if it is $\leq 1/4$.

Theorem 6.2. Approximate Concatenated Jones Plat *is* PSPACE*-complete.*

Proof. To see that this problem is PSPACE-hard, choose any language $L \in$ PSPACE, let U be the n'th quantum circuit from its associated uniform family and let $e = O\big(2^{\mathrm{poly}(n)}\big)$ be the corresponding exponent. Suppose that U acts on n' qubits. For a given x^n, we will show how to use an oracle for **Approximate Concatenated Jones Plat** to determine whether or not $x^n \in L$. Define the circuit W_{x^n} as in the proof of Theorem 6.0, except that it also flips the $n+1$st qubit. By choosing $\epsilon = \epsilon'/e = \Omega(2^{-\mathrm{poly}(n)})$ for a constant ϵ' to be determined later, we may obtain, via Theorem 5.1, a braid b with only $O\big(\mathrm{poly}(n)\big)$ crossings such that $\pi^{(2,\ell)}_{[2n',2n']}(b)$ ϵ-approximates $W_{x^n} U W_{x^n}$. Note that

$$f(x^n) = \langle x^n 100\cdots 0|(W_{x^n} U W_{x^n})^e|x^n 100\cdots 0\rangle = \langle 00\cdots 0|W_{x^n} U^e W_{x^n}|00\cdots 0\rangle.$$

Because $\epsilon'/e = \epsilon$, it follows by Lemma 4.3 that $\pi^{(2,\ell)}_{[2n',2n']}(b^e)$ ϵ'-approximates the circuit $W_{x^n} U^e W^{x^n}$. Choosing any $\epsilon' < 1/2$ is sufficient to complete the proof of PSPACE-hardness. All that is left to do is to check that this problem is contained in PSPACE as well. First, it is clear that **Approximate Concatenated Jones Plat** is contained in BQPSPACE, as our local qubit implementation of Section 4.2 requires only polynomial space. On the other hand, Watrous has shown [36] that PSPACE = BQPSPACE, completing the proof. ∎

Another class of computational tasks is comprised of *counting problems*, which ask, for a given function $f\colon \{0,1\}^* \to \{0,1\}$, "how many $x^n \in \{0,1\}^n$ satisfy $f(x^n) = 1$?" The class #P contains those counting problems arising from a function f for which $f^{-1}(1) \in$ P. We refer the reader seeking more detail to [29] for a more rigorous definition.

Theorem 6.3. *Given a braid $b \in B_{2n}$ with $O\big(\mathrm{poly}(n)\big)$ crossings and a fixed integer $\ell \geq 5$, $\ell \neq 6$, computing the n most significant bits of the absolute value of the normalized Jones polynomial of the plat closure of b*

$$\frac{1}{[2]_\ell^{n-1}} \left| J(\widetilde{b}, e^{2\pi i/\ell}) \right|$$

is #P-hard.

Proof. Suppose $f\colon \{0,1\}^n \to \{0,1\}$ is computable in polynomial-time on a classical computer. It is well-known (see e.g. [30]), that there is a quantum circuit U, acting on $n' = O\big(\mathrm{poly}(n)\big)$ qubits, which computes f reversibly and exactly, namely

$$U|x^n\rangle|y\rangle|00\cdots0\rangle = U|x^n\rangle|y \oplus \overline{f(x^n)}\rangle|00\cdots0\rangle.$$

The state

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x^n \in \{0,1\}^n} |x^n\rangle|1\rangle|00\cdots0\rangle$$

can clearly be prepared efficiently. Observe that

$$\langle\psi|U|\psi\rangle \quad = \quad \frac{1}{2^n}\sum_{x^n}\langle 1|f(x^n)\rangle = \frac{N_1}{2^n},$$

where $N_1 = |f^{-1}(1)|$ is the solution to the #P problem of counting the solutions to f. Let W be the unitary circuit which acts with Hadamard gates on the first n qubits, flips the next qubit, while leaving the rest alone, and set $\epsilon = 4^{-n-2}$. Clearly $W|00\cdots0\rangle = |\psi\rangle$. Since $|a-b| \leq \sqrt{|a^2-b^2|}$ for $0 \leq a, b \leq 1$, it follows by Theorem 5.1 that, given a description of the circuit $W^{-1}UW$, we may, in polynomial-time, compute a braid b for which

$$\left| |\langle 00\cdots0|W^{-1}UW|00\cdots0\rangle| - \frac{1}{[2]^{2n'-1}} \left| J(\widetilde{b}, e^{2\pi i/\ell}) \right| \right| \leq \sqrt{\epsilon} = 2^{-n-1}.$$

In particular, this means that the n most significant bits of the two terms in the above absolute value agree. Therefore, an oracle which gives the $4n'$ most significant bits of that evaluation of the Jones polynomial can be used to obtain all n bits of the number N_1. ∎

In fact, n bits is more than enough bits for #P-hardness. Learning just $\Omega(n^\delta)$ of the most significant bits, for any constant $\delta > 0$, is #P-hard as well. This can be seen by padding with a sufficiently large, but polynomial, number of extra qubits. On the other hand, we will see that learning just the highest order bit of the Jones polynomial is PP-hard.

The complexity class PP, introduced in [15], contains those languages recognized by some uniform family of polynomial-size probabilistic classical circuits with probability greater than 1/2. Equivalently, L is in PP if there is a nondeterministic polynomial-time machine which accepts $x^n \in L$ on more than half of its paths.

Theorem 6.4. *Given a braid $b \in B_{2n}$ with $O\big(\mathrm{poly}(n)\big)$ crossings and a fixed positive integer $\ell \geq 5$, $\ell \neq 6$, computing the most significant bit of the absolute value of the normalized Jones polynomial of the plat closure of b*

$$\frac{1}{[2]_\ell^{n-1}} \left| J(\widetilde{b}, e^{2\pi i/\ell}) \right|$$

is PP*-hard.*

Proof. The proof follows by exactly the same manner as in the proof of #P-hardness. The only difference is that the most significant bit of the Jones polynomial is sufficient for deciding if the number is solutions is greater than 2^{n-1}. ∎

7 Conclusions

We gave an implementation of the Jones-Wenzl unitary representations of the braid group at a primitive ℓ'th root of unity in a local qubit architecture. We constructed the image of each generator using a number of local two-qubit unitaries which was polynomial in the number n of strands, independent of how ℓ grows with n. We then used this model to give algorithms for obtaining additive approximations of the Jones and HOMFLYPT polynomials on a quantum computer. These algorithms run in time which is polynomial in the number of crossings and the inverse of the desired accuracy, independent of how fast ℓ grows with these parameters. Our first algorithm used a reduction based on the fact that the generalized closure of a braid is isotopic to the plat closure of a related braid, while the second was motived by techniques introduced in [2].

Other authors have approached the problem of implementing the Jones representations on a quantum computer. In [22], a representation of the three-strand braid group was given in a qubit architecture. Another approach toward implementing the Jones representations is contained in [34]; however, the final conclusions of that paper rested on an incorrect assumption that unitary matrix elements could be measured precisely in a single time step. Finally, a recent article [14] drew on a connection between topological quantum field theories and spin networks in order to outline a quantum algorithm for approximating the colored Jones polynomial; those authors left as an open question whether their algorithm could be efficiently implemented on a local quantum computer. On a classical computer, if the number of strands is constant (i.e. not included in the complexity estimate), then the Jones polynomial of the trace closure of a braid with m crossings can be computed exactly by simply multiplying representation matrices of the braid group, requiring time which is polynomial in m. However, accounting for the number of strands, the matrices will be exponentially large in the number n of strands. On a quantum computer, however, multiplication of exponentially large matrices is not a problem, provided that they are sufficiently "sparse," or rather, that they arise via local unitary transformations of the state space of the quantum computer. On the other hand, there is a sense in which such large unitary matrices are not explicitly observable; an exponential number of queries seem to be required in order to learn the trace of the matrix representing the unitary evolution of a quantum circuit. Nonetheless, the additive approximation of the Jones polynomial we achieved with a quantum algorithm is unobtainable on a classical computer, unless of course BPP $\neq$ BQP.

On the other side of the fence, we have simplified the proof from [12, 13] which shows how to simulate the standard quantum circuit model with braids. By using four strands per qubit, rather than the three of the original proof, we simplified the representation theory dramatically. The four-periodicity of our encoding means that shifting the eight-strand braid which represents a given local unitary on two qubits by a multiple of four strands realizes the same gate on another pair of qubits. The original result of [12] requires consideration of two different cases, depending on whether the first qubit is located at an even or an odd site. Our encoding applies to schemes for topological quantum computation based on the braiding of nonabelian anyons with label $\frac{1}{2}$

in $SU(2)_{\ell-2}$ Chern-Simons field theories with $\ell \geq 5$, $\ell \neq 6$. On the other hand, our local qubit implementation of the Jones-Wenzl representations allows efficient simulation of the braiding of particles with label $\frac{1}{2}$ in $SU(k)_{\ell-k}$ Chern-Simons theories. While slightly less general than the results of [10], as the only mapping class group our methods apply to is the braid group, we have tried to make our presentation accessible to readers who are unfamiliar with the intricacies of topological quantum field theories.

8 Acknowledgments

We would like to thank Scott Aaronson, Stephen Bigelow, Andrew Childs, Matthias Christandl, John Preskill, Robert Raussendorff, Joost Slingerland and Zhenghan Wang for helpful discussions. We also thank Michael Freedman for bringing the paper [7] to our attention. Some of our diagrams are based on examples from Aaron Lauda's XY-pic tutorial. JY is thankful for support from the National Science Foundation under grant PHY-0456720 through the Institute for Quantum Information at the California Institute of Technology. PW would also like to thank Jörg Bühler for discussions that helped to initiate this project. He is also grateful for support by the National Security Agency under Army Research Office contract number W911NF-05-1-0294.

9 Appendix

9.1 Proofs of Lemmas 4.2 and 4.3

These proofs rely on a dual characterization of the sup-norm in terms of the *trace norm* $\|M\|_1$ of a square matrix M. While the trace norm of M is defined to be the sum of the singular values of M, an equivalent definition is $\|M\|_1 = \max_V \left| \operatorname{Tr} MV \right|$, where the maximization is over all unitaries of the same size as M. The sup-norm is then $\|M\|_\infty = \max\{\|MX\|_1 : \|X\|_1 = 1\}$.

Proof of Lemma 4.2. We begin by bounding

$$\begin{aligned} \left|\langle\phi|U|\psi\rangle\right|^2 &= \left|\langle\phi|U'|\psi\rangle + \langle\phi|(U-U')|\psi\rangle\right|^2 \\ &\leq \left|\langle\phi|U'|\psi\rangle\right|^2 + \left|\langle\phi|(U-U')|\psi\rangle\right|^2. \end{aligned}$$

Moving the first term on the last line to the left-hand side, we continue bounding

$$\begin{aligned} \left|\langle\phi|U|\psi\rangle\right|^2 - \left|\langle\phi|U'|\psi\rangle\right|^2 &\leq \left|\langle\phi|(U-U')|\psi\rangle\right|^2 \\ &= \left|\operatorname{Tr}(U-U')|\psi\rangle\langle\phi|\right|^2 \\ &\leq \max_V \left|\operatorname{Tr}(U-U')|\psi\rangle\langle\phi|V\right|^2 \\ &= \left\|(U-U')|\psi\rangle\langle\phi|\right\|_1^2 \\ &\leq \max_{\|X\|_1=1} \left\|(U-U')X\right\|_1^2 \\ &= \left\|U-U'\right\|_\infty^2 \leq \epsilon^2 \leq \epsilon. \end{aligned}$$

A similar argument shows that $\left|\langle\phi|U'|\psi\rangle\right|^2 - \left|\langle\phi|U|\psi\rangle\right|^2 \leq \epsilon$. ■

Proof of Lemma 4.3. By the dual characterization of the sup-norm, there exists a square matrix X with $\|X\|_1 = 1$ for which

$$\|U_m \cdots U_2U_1 - U'_m \cdots U'_2U'_1\|_\infty = \|(U_m \cdots U_2U_1 - U'_m \cdots U'_2U'_1)X\|_1.$$

Letting $W = U_m \cdots U_3 U_2$ and $W' = U'_m \cdots U'_3 U'_2$, we may rewrite and bound the above as

$$\begin{aligned} \|(WU_1 - W'U'_1)X\|_1 &= \|(W - W')U_1 X + W'(U_1 - U'_1)X\|_1 \\ &\leq \|(W - W')U_1 X\|_1 + \|W'(U_1 - U'_1)X\|_1 \\ &\leq \|W - W'\|_\infty + \|U_1 - U'_1\|_\infty. \end{aligned}$$

The second line is by the triangle inequality, while the third uses the unitary invariance of the trace norm together with the dual characterization of the sup-norm. The result follows by induction. ∎

9.2 Auxiliary calculations from Section 3.2

In this subsection, we will provide derivations of the expressions in (3.23). Plugging $\eta_{2\ell} = [2]_\ell^{-2} = \frac{1}{2+q+q^{-1}}$ into the square of the expression (3.20) and abbreviating $[2] = [2]_\ell$, we obtain

$$\begin{aligned} \theta_{2\ell}^2 &= \frac{q^{-1} - (1+q^{-1})[2]^{-2}}{q - (1+q)[2]^{-2}} \\ &= \frac{q^{-1}(2+q+q^{-1}) - 1 - q^{-1}}{q(2+q+q^1) - 1 - q} \\ &= \frac{q^{-1} + q^{-2}}{q + q^2} \\ &= \frac{q^{-3/2}[2]}{q^{3/2}[2]} \\ &= q^{-3}, \end{aligned}$$

while the square of the expression (3.21) evaluates to

$$\begin{aligned} \nu_{2\ell}^2 &= (\mathrm{tr}_{2\ell}\, g_i)(\mathrm{tr}_{2\ell}\, g_i^{-1}) \\ &= 1 - \frac{q(1+q^{-1})}{[2]^2} - \frac{q^{-1}(1+q)}{[2]^2} + \frac{(1+q)(1-q^{-1})}{[2]^4} \\ &= 1 - \frac{1+q}{[2]^2} - \frac{1+q^{-1}}{[2]^2} + \frac{2+q+q^{-1}}{[2]^4} \\ &= 1 - \frac{[2]^2}{[2]^2} + \frac{[2]^2}{[2]^4} \\ &= \frac{1}{[2]^2}. \end{aligned}$$

Taking the appropriate square roots yields the values listed in (3.23).

To evaluate the Markov weights $s^{(2,\ell)}_{[\lambda_1,\lambda_2]}$ when $\lambda_1 + \lambda_2 = n$, we proceed in two steps. By Lemma 3.5(b) of [37], if λ' is the $(2,\ell)$-diagram obtained by adding r columns of two boxes to the left of another $(2,\ell)$-diagram λ, it follows that $s^{(2,\ell)}_{\lambda'} = [2]_\ell^{-2r} s^{(2,\ell)}_{\lambda}$. A direct computation from the hook length formula (3.17) gives the Markov weight for a single row diagram $[m]$ as

$$s^{(2,\ell)}_{[m]} = \frac{1}{[2]_\ell^m} \frac{[2]_\ell [3]_\ell \cdots [m+1]}{[1]_\ell [2]_\ell \cdots [m]_\ell} = \frac{[m+1]_\ell}{[2]_\ell^m}.$$

Since $\lambda = [\lambda_1, \lambda_2]$ can be obtained by adding λ_2 columns of two boxes to the left of the row diagram $[\lambda_1 - \lambda_2]$, it follows that

$$s^{(2,\ell)}_{\lambda} = \frac{s^{(2,\ell)}_{[\lambda_1-\lambda_2]}}{[2]_\ell^{2\lambda_2}} = \frac{[\lambda_1 - \lambda_2 + 1]_\ell}{[2]_\ell^{\lambda_1+\lambda_2}} = \frac{[\lambda_1 - \lambda_2 + 1]_\ell}{[2]_\ell^n}$$

as required.

References

[1] L. Adleman, J. Demarrias, M. A. Huang, "Quantum computability," *SIAM Journal on Computing,* vol. 5, pp. 1524–1540, 1997.

[2] D. Aharonov, V. F. R. Jones, Z. Landau, "A Polynomial Quantum Algorithm for Approximating the Jones Polynomial," `quant-ph/0511096`.

[3] D. Aharonov, T. Naveh, "Quantum NP – a survey," `quant-ph/0210077`.

[4] S. Bigelow, "Braid groups and Iwahori-Hecke algebras," `math.GT/0505064`.

[5] J. Birman, *Braids, Links and Mapping Class Groups,* Princeton University Press, 1974.

[6] J. Birman, H. Wenzl, "Braids, link polynomials and a new algebra," *Transactions of the American Mathematical Society,* vol. 313, no. 1, 1989.

[7] M. Bordewich, M. Freedman, L. Lovász, D. Welsh, "Approximate counting and quantum computation," to appear in *Combinatorics, Probability and Computing*, 2006.

[8] C. M. Dawson, M. A. Nielsen, "The Solovay-Kitaev algorithm," `quant-ph/0505030`.

[9] C. M. Dawson, H. L. Haselgrove, A. P. Hines, D. Mortimer, M. A. Nielsen, T. J. Osborne, "Quantum computing and polynomial equations over the finite field $\mathbb{Z}_2$," *Quantum Information and Computation*, vol. 5 (2), pp. 102–112, 2005, `quant-ph/0408129`.

[10] M. H. Freedman, A. Y. Kitaev, M. J. Larsen, "Simulation of topological quantum field theories with quantum computers," *Communications in Mathematical Physics,* vol. 227, no. 3, pp. 605–622, 2002, `quant-ph/0001071`.

[11] M. H. Freedman, A. Y. Kitaev, M. J. Larsen, Z. Wang, "Topological quantum computation," *Bulletin of the American Mathematical Society*, vol. 40, no. 1, pp. 31–38, 2003.

[12] M. H. Freedman, M. J. Larsen, Z. Wang, "A modular functor which is universal for quantum computation," *Communications in Mathematical Physics,* vol. 227, no. 3, pp. 587–603, 2002, `quant-ph/0001108`.

[13] M. H. Freedman, M. J. Larsen, Z. Wang, "The two-eigenvalue problem and density of Jones representation of braid groups," *Communications in Mathematical Physics,* vol. 228, no. 1, pp. 177–199, 2002, `math.GT/0103200`.

[14] S. Garnerone, A. Marzuoli, M. Rasetti, "Quantum automata, braid group and link polynomials," `quant-ph/0601169`.

[15] J. Gill, "Computational complexity of probabilistic Turing machines," *SIAM Journal on Computing,* vol. 6, pp. 675–695, 1977.

[16] P. Freyd, D. Yetter, J. Hoste, W. Lickorish, K. Millet, A. Ocneanu, "A new polynomial invariant of knots and links," *Bulletin of the American Mathematical Society*, vol. 12, pp. 239–246, 1985.

[17] V. F. R. Jones, "Index for subfactors," *Inventiones Mathematicae*, vol. 72, pp. 1–25, 1983.

[18] V. F. R. Jones, "A polynomial invariant for links from Von Neumann algebras," *Bulletin of the American Mathematical Society,*

[19] V. F. R. Jones, "Hecke algebra representations of braid groups and link polynomials," *Annals of Mathematics*, vol. 126, pp. 335–388, 1987.

[20] V. F. R. Jones, “The Jones polynomial,” *preprint*, 2005.

[21] L. H. Kauffman, “State models and the Jones polynomial,” *Topology*, vol. 3, no. 26, 395–407, 1987.

[22] L. H. Kauffman, “Quantum computing and the Jones polynomial,” `math.QA/0105255`.

[23] S. J. Lomonaco and L. H. Kauffman, “Quantum computing and the Jones polynomial,” `quant-ph/0605004`.

[24] A. Y. Kitaev, “Quantum computations: algorithms and error correction,” *Russian Mathematical Surveys,* vol. 52(6), pp. 1191–1249, 1997.

[25] W. B. R. Lickorish, *An Introduction to Knot Theory*, Springer-Verlag New York, Inc., 1997.

[26] C. Marriott, J. Watrous, “Quantum Arthur-Merlin Games”, *Computational Complexity*, vol. 14 (2), pp. 122–152, 2005, `cs.CC/0506068`.

[27] K. Murasugi, *Knot Theory and its Applications*, Birkhäuser Boston, 1997.

[28] T. Ohtsuki, *Quantum Invariants: A Study of Knots, 3-Manifolds, and Their Sets*, World Scientific Publishing Co. Pte. Ltd., 2002.

[29] C. H, Papadimitriou, *Computational Complexity*, Addison-Wesley Publishing Company, Inc., 1994.

[30] J. Preskill, unpublished lecture notes.

[31] J. H. Przytycki, P. Traczyk, “Invariants of links of Conway type,” *Kobe Journal of Mathematics*, vol. 4, no. 2, pp. 115–139, 1988.

[32] J.K. Slingerland and F.A. Bais, ”Quantum groups and nonabelian braiding in quantum Hall systems”, *Nucl.Phys.*, vol. B612, pp. 229-290, 2001.

[33] Solovay, unpublished.

[34] V. Subramaniam, P. Ramadevi, “Quantum computation of Jones’ polynomials,” `quant-ph/0210095`.

[35] H. Temperley, E. Lieb, “Relations between the ‘percolation’ and ‘colouring’ problem and other graph-theoretical problems associated with regular planar lattices: Some exact results for the ‘percolation’ problem,” *Proceedings of the Royal Society of London A*, vol. 322(1549), pp. 251-280, 1971.

[36] J. Watrous, “Relationships between quantum and classical space-bounded complexity classes,” *Proc. Thirteenth Annual IEEE Conference on Computational Complexity,* pp. 210–227, 1998.

[37] H. Wenzl, “Hecke algebras of type A_n and factors,” *Inventiones Mathematicae*, vol. 92, pp. 349–383, 1988.

[38] D. J. A. Welsh, *Complexity: Knots, Colourings and Counting,* London Mathematical Society, Lecture Notes Series 189, Cambridge University Press, 1993.

[39] E. Witten, “Quantum field theory and the Jones polynomial,” *Communications in Mathematical Physics,* vol. 121, pp. 351-399, 1989.

[40] P. Wocjan, D. Janzing, T. Decker, T. Beth, “Measuring 4-local n-qubit observables could probabilistically solve PSPACE”, `quant-ph/0308011`.