Parameterized Complexity of Quantum Knot Invariants

Clément Maria
INRIA Sophia Antipolis-Méditerranée, France

Abstract

We give a general fixed parameter tractable algorithm to compute quantum invariants of links presented by planar diagrams, whose complexity is singly exponential in the carving-width (or the tree-width) of the diagram.

In particular, we get a $O(N^{\frac{3}{2}\,\mathrm{cw}}\mathrm{poly}(n)) \in N^{O(\sqrt{n})}$ time algorithm to compute any Reshetikhin-Turaev invariant – derived from a simple Lie algebra $\mathfrak{g}$ – of a link presented by a planar diagram with n crossings and carving-width cw, and whose components are coloured with $\mathfrak{g}$-modules of dimension at most N. For example, this includes the N^{th}-coloured Jones polynomial.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Theory of computation → Computational geometry; Mathematics of computing → Graphs and surfaces

Keywords and phrases computational knot theory, parameterized complexity, quantum invariants

Digital Object Identifier 10.4230/LIPIcs.SoCG.2021.53

Related Version *Full Version*: `https://arxiv.org/abs/1910.00477`

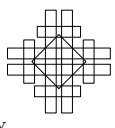
Funding *Clément Maria*: This work has been partially supported by the ANR project ANR-20-CE48-0007-01 (AlgoKnot).

1 Introduction

In geometric topology, testing the topological equivalence of knots (up to isotopy) is a fundamental yet remarkably difficult algorithmic problem.

A main approach is to compare knots by properties depending on their topological types, called *invariants*. Starting with the introduction by Jones [15] in the 1980s of a new polynomial invariant of knots, we have witnessed the birth of a new domain of low dimensional topology called *quantum topology*. From the study of quantum groups [6, 14] in algebra, topologists have designed new families of topological invariants for knots, links, and 3-manifolds, such as the Reshetikhin-Turaev invariants [25]. In practice, these *quantum invariants* have shown outstanding discriminative properties for non-equivalent knots and links, *e.g.*, in the composition of knot census databases [4] that are fundamental to the mathematical work in geometric topology. The infinite families of quantum invariants have even been conjectured to be complete, *i.e.*, that no non-equivalent knot may have all identical quantum invariants. They are also at the heart of deep mathematical conjectures in the field [7, 8, 16, 24].

Consequently, in order to effectively distinguish between knots or verify experimentally mathematical conjectures, efficient algorithms to compute quantum invariants are of strong interest. However, even the simplest quantum invariants, such as the Jones polynomial [13], are #P-hard to compute. A successful approach towards practical implementations has been the introduction of *parameterized complexity* to low dimensional topology. Independently, computing the Jones polynomial [19] and the HOMFLYPT polynomial [3] have been shown to admit fixed parameter tractable algorithms in the tree-width of the input link diagrams. Note that similar techniques have been applied to quantum invariants of 3-manifolds, such



as the Turaev-Viro-Barrett-Westbury invariants of triangulated 3-manifolds [5, 30], where other types of parameters have also been considered [22]. These algorithms led to significant speed-ups in practice, with applications in experimental mathematics [4, 21].

Contribution. We introduce an algorithm to compute quantum invariants derived from ribbon categories [25, 29], taking into account the carving-width of the input link diagram.

▶ **Theorem 1.** *Fix a strict ribbon category $\mathcal{C}$ of $\mathbb{Z}[q]$-modules, and free modules $V_1, \dots, V_m \in \mathcal{C}$ of dimension bounded by N. The problem:*

> Quantum invariant at $\mathcal{C}, V_1, \dots, V_m$:
> **Input**: *m-components link L, presented by a diagram $D(L)$,*
> **Output**: *quantum invariant $J^{\mathcal{C}}_L(V_1, \dots, V_m)$*

can be solved in $O(\text{poly}(n)N^{\frac{3}{2}\,\text{cw}}) \in N^{O(\sqrt{n})}$ machine operations, with $O(N^{\text{cw}} + n)$ memory words, where n and cw *are respectively the number of crossings and the carving-width of the diagram $D(L)$.*

In particular, this implies that, up to some easily computable re-normalisation, computing any fixed Reshetikhin-Turaev invariant derived from a simple Lie algebra $\mathfrak{g}$ is *fixed parameter tractable* (complexity class FPT) in the carving-width (and also the tree-width) of the input link diagram.

To compare with the state-of-the-art, the only known parameterized and sub-exponential time algorithms for quantum knot invariants are the Jones polynomial [19] ($\mathfrak{g} = \mathfrak{sl}(2, \mathbb{C})$ and $N = 2$) and the HOMFLYPT polynomial [3] (which interpolates the invariants for $\mathfrak{g} = \mathfrak{sl}(m, \mathbb{C})$, for all m). So,

1. Theorem 1 generalises these results (fixed parameter tractable algorithm and sub-exponential time algorithm) to *all the infinitely many* dimensions $N \geq 2$ – the so called *coloured* Jones polynomials – quantum invariants, and in general to all coloured invariants derived from other simple Lie algebras $\mathfrak{g}$.
2. It also generalises Burton's result on the parameterized complexity of the HOMFLYPT polynomial, to offer a parameterized solution to the *coloured* HOMFLYPT polynomials by interpolation; see [23].
3. More generally, for knot theorists, the algorithm of Theorem 1 is a low exponent ($\frac{3}{2}$ cw) singly exponential algorithm for quantum invariants[1]. In practice, and considering past experience in 3-manifolds, this work offers a practical algorithm in order to verify fundamental mathematical conjectures [7, 8, 16, 24] experimentally, and further tools to distinguish between non-equivalent knots in the constitution of knot censuses [4] thanks to the outstanding discriminative power of quantum invariants.
4. Finally, for computer scientists, the Jones polynomial has been studied in detail as it has a very rich complexity theory: it is $\#P$-hard [13] but admits an FPT and sub-exponential time algorithm [19] ; it is also one of the few known natural BQP-complete problems in quantum computing [1]. This article completes the computational complexity picture on quantum invariants by showing that very large families of them admit an FPT and sub-exponential algorithm.

[1] Note that previous algorithms [19] are expressed in terms of tree-width, which is proportional but not equal to the carving-width, in consequence exponents are not directly comparable ; see Theorem 4.

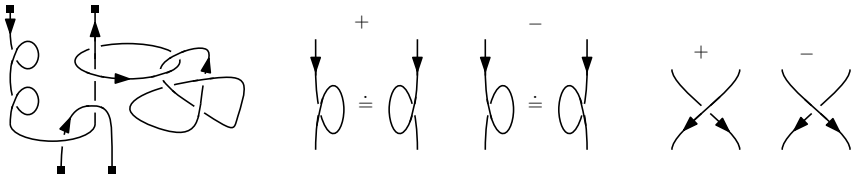


Figure 1 Left: Diagram of a 4-components oriented tangle; leftmost component has framing +2. Right: Positive/negative twists and crossings. The $\doteq$ symbol is an equivalence of diagrams.

The key approach of this work is the use of the abstract construction of quantum invariants via *ribbon categories*, in order to state the complexity result in full generality. This way, we abstract ourselves from the specificity of each singular invariant to give a general algorithm.

In Section 2 we recall the general definition of quantum invariants derived from ribbon categories, and notions of parameterized complexity. In Section 3 we give the general dynamic programming approach for the parameterized algorithm, relying on a decomposition of the knot diagram. In Section 4 we detail the topological and algebraic operations necessary to implement the dynamic programming procedure ; the connection between topology and algebra relies on graphical calculus. Finally, in Section 5, we implement the algebraic operations with matrix multiplications, and analyse their complexity and the complexity of the whole algorithm. In conclusion, we prove that all arithmetic operations have polynomial time complexity and prove the main Theorem 1 in Section 6. Finally, in [23], we implement the algorithm and verify mathematical conjectures on formally intractable quantum invariants.

2 Background

We introduce the necessary notions from knot theory and parameterized complexity.

Tangles and their diagrams. A *tangle* is a piecewise linear embedding of a collection of arcs and circles into $\mathbb{R}^2 \times [0,1]$, such that the arcs' endpoints belong to the top or bottom boundaries $\mathbb{R}^2 \times \{0\}$ and $\mathbb{R}^2 \times \{1\}$. A tangle intersecting i times $\mathbb{R}^2 \times \{0\}$ and j times $\mathbb{R}^2 \times \{1\}$ is an (i,j)-tangle.

A *link* is a tangle whose connected components are all closed curves (a $(0,0)$-tangle), and a *knot* is a 1-component link. An *orientation* on a tangle is an orientation of each tangle component. Two tangles are equivalent *iff* they differ by an ambient isotopy of $\mathbb{R}^2 \times [0,1]$ maintaining the boundary fixed.

A tangle *diagram* is a projection of the tangle into the plane, induced by a projection of $\mathbb{R}^2 \times [0,1]$ into $\mathbb{R} \times [0,1]$, sending $\mathbb{R}^2 \times \{0\}$ and $\mathbb{R}^2 \times \{1\}$ to $\mathbb{R} \times \{0\}$ and $\mathbb{R} \times \{1\}$ respectively. In a tangle diagram, the only multiple points are *crossings*, at which one section of the tangle crosses under or over another one transversally. We consider diagrams of $(0,0)$-tangles (*i.e.*, link diagrams) as drawn on the sphere S^2. Component orientations are pictured with arrow heads, and a $k \in \mathbb{Z}$ *framing* is pictured by k *positive twists* if $k > 0$, and k *negative twists* is $k < 0$. See Figure 1. We refer to [17] for more details on knot theory.

Graphical calculus on coloured tangles. We work in the category $\mathcal{C}$ of free finitely generated $\mathcal{R}$-modules, for a commutative ring $\mathcal{R}$, with their usual tensor product $\otimes$ and dual V^* for every object V. Additionally, some fixed special objects and morphisms are distinguished and play an important role:

(a) a finite set of objects $\{V_i : i \in I\} \subset \mathcal{C}$, called *colours*,
(b) for every pair of colours V_i, V_j, an isomorphism $c_{V_i,V_j} : V_i \otimes V_j \to V_j \otimes V_i$ and its inverse $c^{-1}_{V_i,V_j}$, called *braidings*,
(c) for every colour V_i, two morphisms $b_{V_i} : \mathbb{1} \to V_i \otimes V_i^*$ and $d_{V_i} : V_i^* \otimes V_i \to \mathbb{1}$, called respectively *co-evaluation* and *evaluation*, and
(d) for every colour V_i, an isomorphism $\theta_{V_i} : V_i \to V_i$ and its inverse $\theta^{-1}_{V_i}$, called a *twist*.

We call the ring $\mathcal{R}$, seen as a module over itself, the *unit module*, and denote it by $\mathbb{1}$. Note that the set of morphisms $\mathrm{Hom}_{\mathcal{C}}(\mathbb{1}, \mathbb{1})$ has the structure of a commutative ring, isomorphic to $\mathcal{R}$, *i.e.*, every module morphism $\mathbb{1} \to \mathbb{1}$ is a multiplication by a scalar $\tau \in \mathcal{R}$. By convention, the tensor product of zero objects is equal to $\mathcal{R}$.

Such category of modules, with *braidings*, *twists*, *evaluations* and *co-evaluations*, is a *strict ribbon category* if the objects and morphisms satisfy a certain set of additional equations. The theory of strict ribbon category is outside the scope of this article, and we only use the fact that strict ribbon categories define topological invariants of links (Theorem 2 below). We refer the interested reader to [29] for the general theory.

Fix a strict ribbon category $\mathcal{C}$. A *colouring* of a link L, with m ordered components $L_1, \ldots, L_m$, is an assignment of a colour $V_i \in \mathcal{C}$, $1 \leq i \leq m$, to every component L_i of L.

A link diagram is in *standard position* if it can be decomposed into the following pieces: vertical strands, positive and negative crossings, positive and negative right twists, and caps and cups. See Figure 3 for a Hopf link in standard position. Any link diagram can be isotoped into standard position without modifying the diagram combinatorially [29].

Figure 2 presents the conversion from oriented coloured tangles to $\mathcal{C}$-morphisms, called *graphical calculus*, and Figure 3 gives a full example on the Hopf link. Specifically, given a coloured link diagram $D(L)$ in standard position, the graphical calculus turns the diagram into a morphism, following a set of conversion rules (from link diagram to morphism) and equivalence relations, pictured graphically in Figure 2, and described below:

Figure 2a (i)&(ii) A morphism $f : U \to V$ in $\mathcal{C}$ is represented graphically by a box, aligned with x- and $y-$axis, called a *coupon*, with an incoming vertical V-coloured strand (top) and an outgoing vertical U-coloured strand (bottom). The identity morphism id_V is equivalent to a vertical downward-oriented V-coloured strand. Note that morphisms, and more generally entire diagrams, are read *bottom-to-top*.

Figure 2a (iii) reversing a component orientation changes a colour V to its dual V^*. See also Figure 2b for equivalent representations of a morphism changing strands orientations.

Figure 2c (i)&(ii) two parallel strands coloured U (left) and V (right) respectively are equivalent to a single strand coloured $U \otimes V$. Two side-by-side coupons for morphisms h_1 (left) and h_2 (right) are equivalent to a single coupon for morphism $h_1 \otimes h_2$.

Figure 2c (iii) a coupon for morphism g on top of a coupon for morphism f is equivalent to their composition $g \circ f$,

Figure 2d (i)&(ii) a positive crossing is equivalent to a braiding morphism, a negative crossing is equivalent to the inverse of the braiding morphism. See also Figure 2d (iii) for a realisation of $c^{-1} \circ c = \mathrm{id}$ as a Reidemeister move.

Figure 2e (i)&(ii) positive and negative twists are equivalent to the twist morphism and its inverse respectively,

Figure 2f (i)&(ii) caps and cups are equivalent to evaluation and co-evaluation morphisms respectively.

Figure 2g graphical and algebraic definition of the dual morphism $f^*\colon V^* \to U^*$ of a morphism $f\colon U \to V$.

Figure 2h Sliding coupons along strands does not change the corresponding morphism.

The morphisms are applied to the objects colouring the entering and leaving strands. Note that, by convention, morphisms are composed bottom-up. Additionally, we can reverse the orientation of strands by dualising their colours if convenient. Figure 3 gives the morphism associated to the Hopf link coloured with objects U and V.

Consequently, for a category $\mathcal{C}$, graphical calculus associates to any coloured link, seen as a $(0,0)$-tangle, a morphism $\mathbb{1} \to \mathbb{1}$. More generally, it associates to a coloured (i,j)-tangle a morphism $U_1 \otimes \ldots \otimes U_i \to V_1 \otimes \ldots \otimes V_j$, for the $\mathcal{C}$-objects U_k and V_ℓ colouring the bottom and top endpoints of the tangle respectively.

If the ordered components of a link L are coloured $V_1, \ldots, V_m$, this morphism is written:

$$J_L^{\mathcal{C}}(V_1, \ldots, V_m) \in \mathrm{Hom}_{\mathcal{C}}(\mathbb{1}, \mathbb{1}).$$

Strict ribbon categories produce topological invariants, called *quantum invariants*:

▶ **Theorem 2** ([25, 29]). *Let L be an m-components link, and D_1 and D_2 two arbitrary diagrams of L in S^2. Let $V_1, \ldots, V_m \in \mathcal{C}$ be a colouring of the components of L, and $J_{D_1}^{\mathcal{C}}(V_1, \ldots, V_m)$ and $J_{D_2}^{\mathcal{C}}(V_1, \ldots, V_m)$ the two quantities obtained by graphical calculus on D_1 and D_2 respectively. Then:*

$$J_{D_1}^{\mathcal{C}}(V_1, \ldots, V_m) = J_{D_2}^{\mathcal{C}}(V_1, \ldots, V_m) = J_L^{\mathcal{C}}(V_1, \ldots, V_m).$$

Using $\mathrm{Hom}(\mathbb{1}, \mathbb{1}) \cong \mathcal{R}$, we identify the invariant $J_L^{\mathcal{C}}(V_1, \ldots, V_k)$ to a scalar in $\mathcal{R}$.

Graph parameters. The *carving-width*, also known as *congestion*, is a graph parameter introduced by Seymour and Thomas [28].

▶ **Definition 3.** *Let $\mathcal{G} = (X, E)$ be a graph on n vertices, with loops and multiple edges. Let $\mathcal{T}$ be an unrooted binary tree, with all internal nodes of degree 3, and with n leaves.*

An embedding *ϕ of $\mathcal{G}$ into $\mathcal{T}$ is a bijective mapping between the nodes X of $\mathcal{G}$ and the leaves of $\mathcal{T}$. Every edge e of $\mathcal{T}$ induces a partition of the vertices of $\mathcal{G}$ into two sets, $X = X_e^{(1)} \sqcup X_e^{(2)}$, inherited from the partition of $\mathcal{T} \setminus e$ into two trees. Let $\mathrm{w}(e)$ denote the number of edges in $\mathcal{G}$ between $X_e^{(1)}$ and $X_e^{(2)}$, called the weight of e. See Figure 4.*

The width *of an embedding $(\mathcal{T}, \phi)$ is the maximal weight of a tree edge:* $\max_{e \text{ edge of } \mathcal{T}} \mathrm{w}(e)$.

The carving-width $\mathrm{cw}(\mathcal{G})$ *of a graph $\mathcal{G}$ is the minimal width over all its embeddings into binary trees. The* carving-width $\mathrm{cw}(D(L))$ of a link diagram $D(L)$ *is the carving-width of the 4-valent planar graph it realises (we use the same notation).*

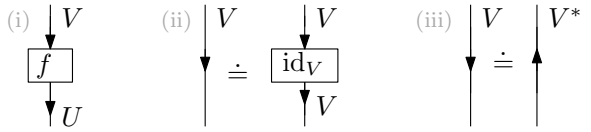
The carving-width $\mathrm{cw}(\mathcal{G})$ of a graph $\mathcal{G}$ is closely related to its *tree-width* [26] $\mathrm{tw}(\mathcal{G})$, which plays a major role in combinatorial algorithms.

▶ **Theorem 4** (Theorem 1 of [2]). *Let $\mathcal{G}$ be a graph of maximal degree δ. Then,*

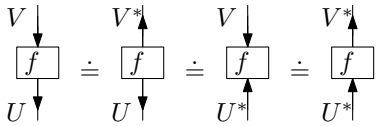
$$\frac{2}{3}(\mathrm{tw}(\mathcal{G}) + 1) \leq \mathrm{cw}(\mathcal{G}) \leq \delta(\mathrm{tw}(\mathcal{G}) + 1). \qquad \textit{For tangle diagrams } \delta \leq 4.$$

Carving-width has several advantages over tree-width, and has been successfully used in low dimensional topology [11, 12, 20, 27].

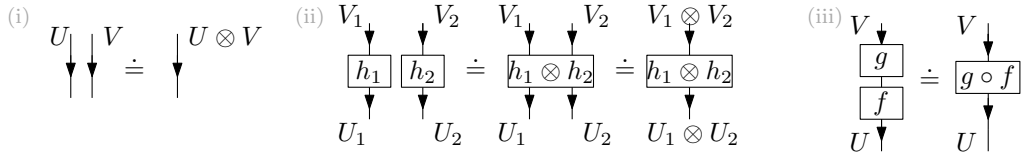
First, optimal tree embeddings of planar graphs can be realised topologically, as stated below. A tree embedding $(\mathcal{T}, \phi)$ of $\mathcal{G}$ is *bond* if the two vertex sets $X_e^{(1)}$ and $X_e^{(2)}$ from the cut associated to an edge e of $\mathcal{T}$ induce connected sub-graphs in $\mathcal{G}$.



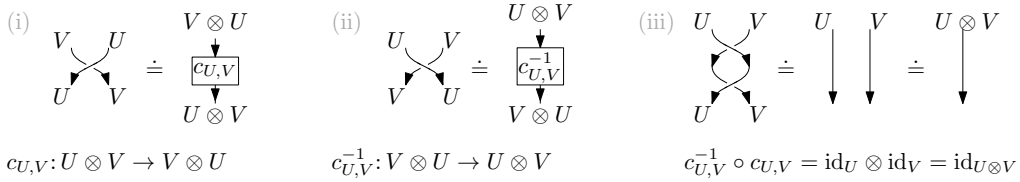
(a) (i) Graphical representation of morphism $f\colon U \to V$ via stands and coupon. (ii) Equivalence strand and identity. (iii) Dualisation and strand orientation.



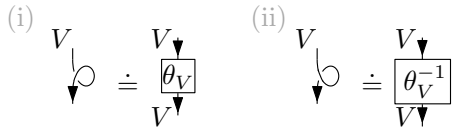
(b) Four equivalent representations of morphism $f\colon U \to V$ with all possible strands orientations.



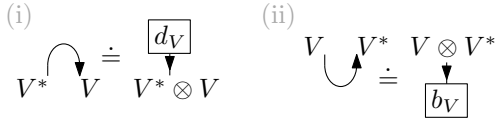
(c) (i) & (ii) Side-by-side coloured strands and morphisms is equivalent to tensor product of colours and morphisms. (iii) Morphisms on top of one another is equivalent to composition.



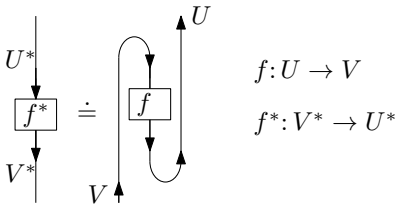
(d) (i) Positive and (ii) negative crossings are equivalent to the braiding morphism and its inverse, respectively. As illustration, the equation $c^{-1} \circ c = \mathrm{id}$ is equivalent to a Reidemeister move II.



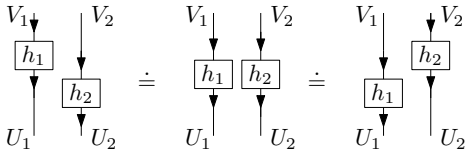
(e) (i) Positive and (ii) negative twists (framing) are equivalent to the twist morphism and its inverse, respectively.



(f) (i) Local maxima (caps) and (ii) local minima (cups) are equivalent to evaluation and coevaluation morphisms.



(g) Definition of the dual f^* of a morphism f, as $f^* := (d_V \otimes \mathrm{id}_{U^*}) \circ (\mathrm{id}_{V^*} \otimes f \otimes \mathrm{id}_{U^*}) \circ (\mathrm{id}_{V^*} \otimes b_U)$.



(h) Invariance by sliding coupons along strands, *i.e.*, $(h_1 \otimes \mathrm{id}_{U_2}) \circ (\mathrm{id}_{U_1} \otimes h_2) = h_1 \otimes h_2 = (\mathrm{id}_{U_1} \otimes h_2) \circ (h_1 \otimes \mathrm{id}_{U_2})$.

Figure 2 Formal rules and some equations for graphical calculus.

▶ **Theorem 5** ([28, Theorem 5.1]). *Let $\mathcal{G}$ be 2-connected with at least two nodes. If $\mathcal{G}$ has carving-width* cw *then there exists a bond tree embedding of $\mathcal{G}$ of width* cw.

This theorem applies to link diagrams [27]. We interpret a bond tree embedding of a planar graph (on the sphere S^2) as a collection of disjoint Jordan curves $\lambda_e \subset S^2$, one for each edge e of $\mathcal{T}$, realising the cut $X_e^{(1)} \sqcup X_e^{(2)}$.

$$[(d_U \otimes d_{V^*}) \circ (\mathrm{id}_{U^*} \otimes c_{U,V} \otimes \mathrm{id}_{V^*}) \circ (\mathrm{id}_{U^*} \otimes c_{V,U} \otimes \mathrm{id}_{V^*}) \circ (b_{U^*} \otimes b_V)] : \mathbb{1} \to \mathbb{1}$$

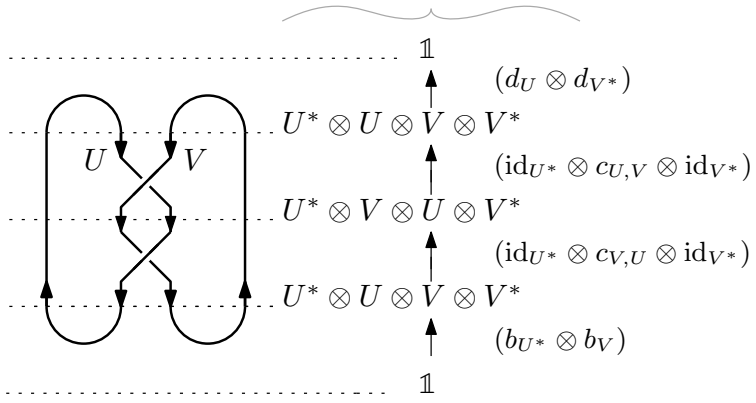


Figure 3 Application of graphical calculus to the Hopf link coloured by objects U and V from a strict ribbon category, leading to a $\mathbb{1} \to \mathbb{1}$ morphism by composition. Composition is bottom-to-top, the expression of morphisms is given on the right, and correspond to the two cups (co-evaluations), then the crossings (braidings), then the two caps (evaluations).

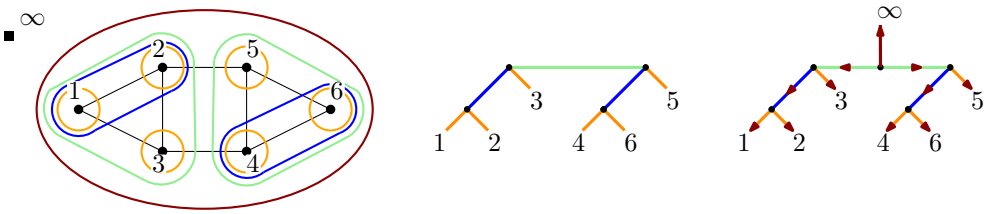


Figure 4 Left/middle: 6-vertex graph on S^2 with width 3 bond tree embedding (tree and Jordan curves). Note that the two green curves represent the same cut on S^2. Right: rooting of the tree by adding a node at ∞.

For planar graphs, a bond tree embedding of minimal width can be computed in polynomial time [10, 28]. No such result is known for tree-width. By the planar separator theorem [18], the carving-width of a planar graph with n-vertices is in $O(\sqrt{n})$.

3 Fixed parameter tractable algorithm via graphical calculus

Let $\mathcal{C}$ be a strict ribbon category, and let L be an oriented link with m components $L_1, \ldots, L_m$. Let $D(L)$ be an oriented link diagram of L, where each link component L_i is coloured by an object V_i from the category $\mathcal{C}$, such that graphical calculus gives an isotopy invariant of L associated to its colouring, as described in Theorem 2.

It follows from the definition of graphical calculus, and particularly equality Figure 2c (i)&(ii), that the quantum invariant of a separable link $L \cup L'$ is the product of the invariants of L and L', such that they can be computed separately. W.l.o.g. we assume that the diagram $D(L)$ is connected as a graph, and has at least 2 crossings, not all twists. If any of the two last requirements are not met, which can be checked in linear time, the diagram represents a possibly framed trivial knot, for which quantum invariants can be deduced directly. Consequently, all crossings of diagrams have degree four, with at most one self-edge ; see Figure 5. These properties are used in the complexity analysis Section 5.2.

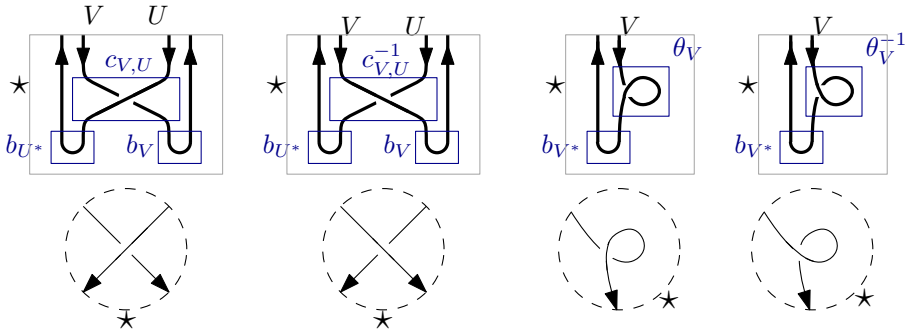


Figure 5 Morphisms associated to neighbourhoods of crossings, *i.e.*, morphisms at leaves of a tree embedding. From left to right, the corresponding equations are given by Eq. (1), (2), (3), and (4) of Section 3.3. The starred point is selected such that only these four morphisms are encountered (*i.e.*, between the two arrow heads for crossings, and on the side of the loop for twists).

3.1 Tree embedding of link diagrams

Let $(\mathcal{T}, \phi)$ be a bond tree embedding of the planar graph of $D(L)$ in S^2. We add a fake, disconnected, *node at infinity* to the graph $D(L)$, and we add a corresponding new leaf to the tree embedding $\mathcal{T}$ by subdividing an arbitrary tree edge, and connecting a leaf to it. Topologically, the Jordan curve in a neighbourhood of the node at infinity encircles the entire graph $D(L)$. We root the tree $\mathcal{T}$ to the new inner node to which the new leaf is attached ; see Figure 4. All edges of $\mathcal{T}$ have now a *parent* and *child* endpoint.

Let e be an edge of $\mathcal{T}$ with child node z, and Z the set of crossings mapped to the leaves of the subtree $\mathcal{T}_z$ rooted at z. According to Theorem 5, there exists a Jordan curve λ_e separating Z from the rest of the diagram. All tangle diagrams are on the sphere. By convention, we draw the tangle inside the Jordan curve when we represent it on the plane.

To an edge e corresponds a $(0, \mathrm{w}(e))$-tangle T, spanned by the crossings Z and contained inside the Jordan curve λ_e. We mark an arbitrary but fixed *starred point* on λ_e, and order the endpoints of the tangle T counter-clockwise starting at that starred point. We get a $(0, \mathrm{w}(e))$-tangle by isotopically sliding all tangle endpoints to the top boundary, such that the first endpoint in the *starred point ordering* is rightmost on the top boundary. See Figure 5 for examples of $(0, \mathrm{w}(e))$-tangles at the tree leaves, and Figure 6 (left) for endpoints ordered by starred point ordering.

In the process of the algorithm below, starred points are assigned on the fly. They are used exclusively to deterministically order tangles' endpoints and process the isotopy described above ; they induce a planar isotopy of the tangle and consequently affect the computation, but lead to the same output thanks to the isotopy invariance Theorem 2.

3.2 Tree traversal algorithm

Let $D(L)$ be coloured by objects of the category $\mathcal{C}$. To every edge e of weight $\mathrm{w}(e)$ in $\mathcal{T}$, graphical calculus assigns a $\mathcal{C}$-morphism $f_e \colon \mathbb{1} \to V_1 \otimes \ldots \otimes V_{\mathrm{w}(e)}$ to the associated tangle, where $V_1, \ldots, V_{\mathrm{w}(e)}$ are the colours of the strands intersecting the Jordan curve λ_e.

The morphism associated to the half-edge at the root is a $\mathbb{1} \to \mathbb{1}$ morphism, because the corresponding Jordan curve does not intersect the link diagram. This morphism gives the invariant $J^{\mathcal{C}}_L \in \mathcal{R}$ of Theorem 2. All edge morphisms are computed recursively following a depth first traversal of $\mathcal{T}$. We describe the base morphisms assigned to the edges whose child node is a leaf, and we describe an algorithm for inner edges in the next section.

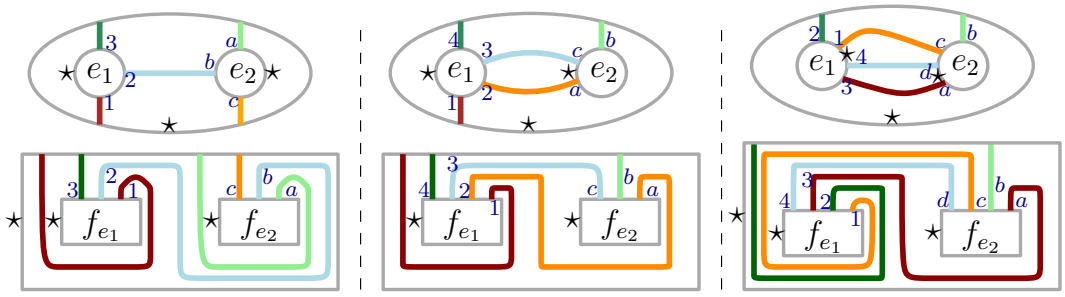


Figure 6 Merging two sub-trees. Top: Planar embeddings of the diagram with Jordan curves λ_{e_1}, λ_{e_2} (inner circles) and λ_e (outer circle), depending on the relative position of the starred points for λ_{e_1} and λ_{e_2}. The bold lines connecting the Jordan curves represent multiple parallel strands connecting the corresponding tangles. Bottom: Coupons for f_{e_1}, f_{e_2} and f_e (outer coupon) obtained after plane isotopy. The starred point for λ_e is selected so as to restrict to these three cases.

3.3 Morphisms at the leaves

The tree leaves are in bijection with tangles made by small neighbourhoods around single crossings. All crossings being of degree 4, and up to reorientation of the strands which algebraically consists of dualising colours, we get four base morphisms: one positive or negative crossing, or one positive or negative twist (with a self-edge),

$$(\mathrm{id}_{U^*} \otimes c_{V,U} \otimes \mathrm{id}_{V^*}) \circ (b_{U^*} \otimes b_V) \quad (1) \qquad (\mathrm{id}_{U^*} \otimes c_{V,U}^{-1} \otimes \mathrm{id}_{V^*}) \circ (b_{U^*} \otimes b_V) \quad (2)$$

$$(\mathrm{id}_{V^*} \otimes \theta_V) \circ b_{V^*} \quad (3) \qquad (\mathrm{id}_{V^*} \otimes \theta_V^{-1}) \circ b_{V^*} \quad (4)$$

They correspond graphically to the diagrams in Figure 5 (nodes of degree 4 are crossings, nodes of degree 2 are twists).

3.4 Merging morphisms at tree nodes

Every inner node x of $\mathcal{T}$ is the parent node of two edges e_1 and e_2, and the child of an edge e. Given the morphisms f_{e_1} and f_{e_2} for edges e_1 and e_2 respectively, we construct the morphism f_e for edge e. This is the fundamental operation of the dynamic programming algorithm ; the end of Section 3 and Section 4 are dedicated to the design of an algorithm for this operation.

First, note that the starred point ordering of the strands intersecting λ_{e_1} and λ_{e_2} leads to three configurations when representing morphisms f_{e_1} and f_{e_2} with coupons ; see Figure 6 where thick lines represent sets of parallel tangle strands. By hypothesis, morphisms on tree edges have domain $\mathbb{1}$. The coupons for f_{e_1}, f_{e_2}, and f_e (the outer coupon) are obtained by a plane isotopy forcing the strands to intersect coupons on their top side, and putting starred points on the coupons' left sides. More specifically, the isotopies are implemented by rotating annulus neighbourhoods of the Jordan curves (grey circles in Figure 6) in order to position the starred point on the left, then straightening the strands.

The starred point of the outer coupon f_e is selected so as to restrict to the three configurations of Figure 6.

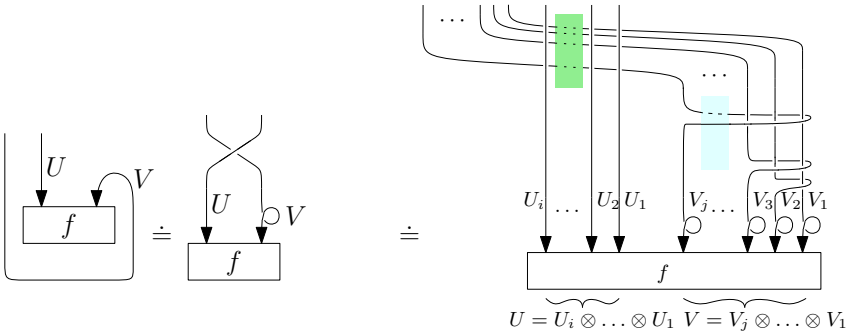


Figure 7 Sliding of a $V = V_j \otimes \ldots \otimes V_1$-coloured strand under an f-coupon by underlying knot isotopy. The operation composes f with a consecutive sequence of j twists θ_{V_ℓ}, of $j(j-1)$ crossings $c^{\pm}_{V_i,V_j}$, and ij crossings $c^{\pm}_{U_k,V_\ell}$.

4 Factorisation of morphisms at tree nodes

Given the morphisms f_{e_1} and f_{e_2} in Figure 6, we describe graphically a factorisation scheme to obtain the morphism f_e.

4.1 Sliding and canonical form

The *canonical form* for morphisms to be merged is depicted in the top left corner of Figure 8. It consists of two side-by-side morphisms g_1 and g_2, bridged by parallel strands coloured $U_1, \ldots U_k$. All other strands go up vertically.

Given morphisms f_{e_1} and f_{e_2} in Figure 6, we obtain a canonical form by sliding strands, wrapping clockwise around the coupons, under the coupons. For example, in the top right case of Figure 6, we slide strand 1 under the f_{e_1}-coupon, and strands a and b under the f_{e_2}-coupons.

The details of the operation are depicted in Figure 7, where the V-strand wraps clockwise around the f-coupon, and f is a $\mathbb{1} \to U \otimes V$ morphism. Sliding the V-strand under the coupon by tangle isotopy produces a positive twist θ_V and a positive crossing $c_{V,U}$.

More specifically, decomposing further in Figure 7, let $U = U_i \otimes \ldots \otimes U_1$ be the tensor product of the colours of i parallel strands, and $V = V_j \otimes \ldots \otimes V_1$ the tensor product of j parallel strands wrapping clockwise around the f coupon. As depicted in the figure, sliding the j strands under f induces:

- a twist θ_{V_ℓ} on each of the V_ℓ-coloured strands, $1 \leq \ell \leq j$,
- a sequence of $j(j-1)$ positive and negative crossings of type $c^{\pm}_{V_\ell,V_k}$, followed by
- a sequence of ij positive crossings of type c_{V_ℓ,U_k}.

We obtain the morphisms g_1, g_2 of the canonical form (Figure 8) by factorising the morphisms f_{e_1} and f_{e_2} with these sequences of twists and crossings, after the sliding operation.

4.2 Factorisation of the canonical form

Figure 8 pictures two factorisation schemes for side-by-side morphisms g_1 and g_2 in canonical form, bridged by k parallel strands coloured $U_1, \ldots, U_k$. Denote by cw the carving-width of the link diagram, and assume the tree embedding $(\mathcal{T}, \phi)$ has width cw.

We distinguish between two cases, depending on the value of k, which is necessary to obtain the factor $\frac{3}{2}$ in the exponent of the complexity analysis of the main Theorem 1:

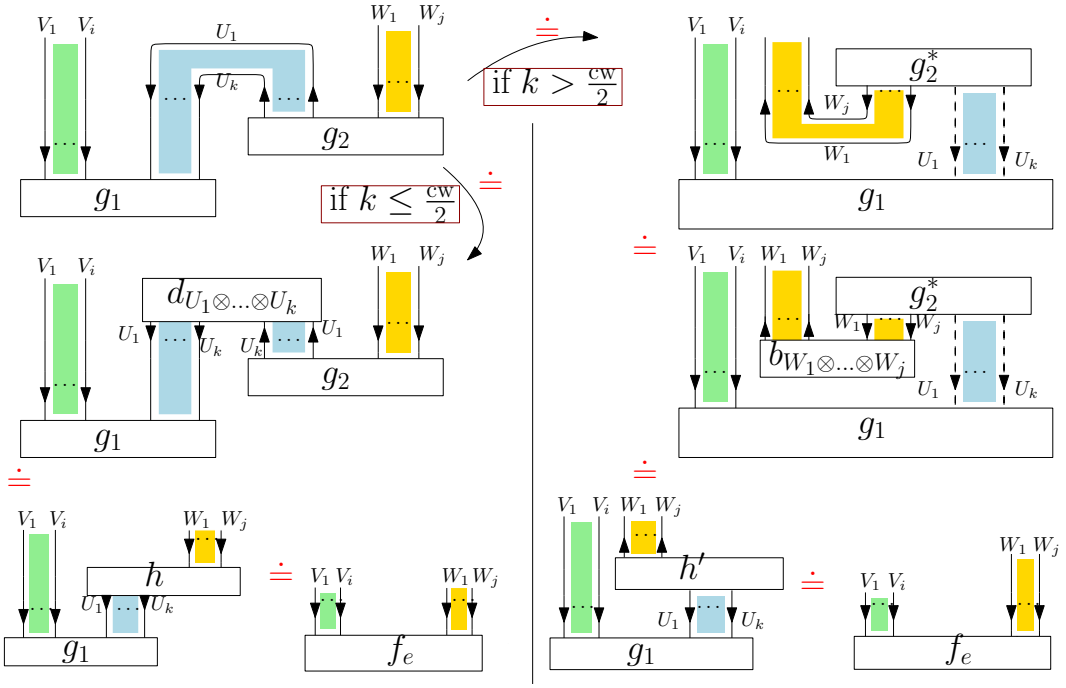


Figure 8 Merging of two coupons in a canonical form (top left) along k strands coloured $U_1, \ldots, U_k$. The factorisation scheme differs whether $k \leq \mathrm{cw}/2$ (left column) or $k > \mathrm{cw}/2$ (right column). The top right equivalence comes from the equality in Figure 9.

Small bridge. For k at most half the carving-width (Figure 8, Left), we consider first the morphism $d_{U_1\otimes\ldots\otimes U_k}$ induced by the composition of the evaluation morphisms d_{U_ℓ}, $\ell = k \ldots 1$. More precisely, the morphism $d_{U_1\otimes\ldots\otimes U_k} \colon U_1 \otimes \ldots \otimes U_k \otimes U_k^* \otimes \ldots \otimes U_1^* \to \mathbb{1}$, is obtained by composing the evaluation morphisms from bottom up:

$$\begin{aligned} d_{U_1\otimes\ldots\otimes U_k} \quad &: \quad U_1 \otimes \ldots \otimes U_k \otimes U_k^* \otimes \ldots \otimes U_1^* \to \mathbb{1}, \\ &= \quad \prod_{\ell=k}^{1} \left(\mathrm{id}_{U_1\otimes\ldots\otimes U_{\ell-1}} \otimes d_{U_\ell^*} \otimes \mathrm{id}_{U_{\ell-1}^*\otimes\ldots\otimes U_1^*} \right), \end{aligned} \tag{5}$$

where $\ell = k$ is the rightmost term of the composition.

The (partial) composition of $d_{U_1\otimes\ldots\otimes U_k}$ with g_2 through $U_k^* \otimes \ldots \otimes U_1^*$ gives the morphism:

$$h \colon U_1 \otimes \ldots \otimes U_k \to W_1 \otimes \ldots \otimes W_j, \qquad h := \left(d_{U_1\otimes\ldots\otimes U_k} \otimes \mathrm{id}_{W_1\otimes\ldots\otimes W_j}\right) \circ \left(\mathrm{id}_{U_1\otimes\ldots\otimes U_k} \otimes g_2\right). \tag{6}$$

Finally, the morphism f_e obtained from the merging of f_{e_1} and f_{e_2} is given by the (partial) composition of g_1 and h, through $U_1 \otimes \ldots \otimes U_k$. Precisely,

$$f_e \colon \mathbb{1} \to V_1 \otimes \ldots \otimes V_i \otimes W_1 \otimes \ldots \otimes W_j, \qquad f_e := (\mathrm{id}_{V_1\otimes\ldots\otimes V_i} \otimes h) \circ g_1. \tag{7}$$

By construction, these operations give the morphism f_e induced by graphical calculus on the coloured tangle associated to the subtree of $\mathcal{T}$ rooted at the child node of edge e.

Large bridge. The case k strictly larger than half the carving-width starts by flipping upside-down coupon g_2. This operation is depicted in Figure 9. Starting with a morphism g, it consists of a planar isotopy to produce g^*, the dual morphism to g. In the case where the category $\mathcal{C}$ satisfies the hypothesis of Theorem 2, Figure 9, depicting an isotopy, proves the equality: $(d_U \otimes \mathrm{id}_V) \circ (\mathrm{id}_{U^*} \otimes g) = (\mathrm{id}_V \otimes g^*) \circ (b_V \otimes \mathrm{id}_{U^*})$.

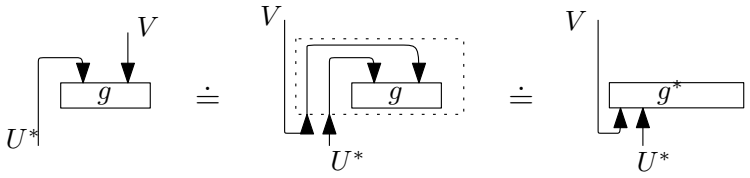


Figure 9 Planar isotopy, then factorisation with g^*, the dual morphism to g.

Applied to the canonical form on g_1 and g_2 (Figure 8, Top) the operation gives the composition of morphisms, involving g_1 and g_2^*, in the top right corner of Figure 8.

The following compositions are similar to the case of a small bridge. Morphism $b_{W_1\otimes\ldots\otimes W_j}$ describes the composition of the co-evaluation morphisms for $W_1,\ldots,W_j$, *i.e.*,

$$\begin{aligned} b_{W_1\otimes\ldots\otimes W_j} &: \mathbb{1} \to W_1^*\otimes\ldots\otimes W_j^*\otimes W_j\otimes\ldots\otimes W_1, \\ &= \prod_{\ell=1}^{j}\left(\mathrm{id}_{W_1^*\otimes\ldots\otimes W_{\ell-1}^*}\otimes b_{V_\ell}\otimes \mathrm{id}_{W_{\ell-1}\otimes\ldots\otimes W_1}\right), \end{aligned} \tag{8}$$

where $\ell = 1$ is the rightmost term of the composition.

The morphism h' is obtained by (partial) composition of $b_{W_1\otimes\ldots\otimes W_j}$ and g_2^*: $h' = \left(\mathrm{id}_{W_1^*\otimes\ldots\otimes W_j^*}\otimes g_2^*\right)\circ\left(b_{W_1\otimes\ldots\otimes W_j}\otimes \mathrm{id}_{U_1\otimes\ldots\otimes U_k}\right)$, and f_e is obtained by (partial) composition of g_1 and h': $f_e = (\mathrm{id}_{V_1\otimes\ldots\otimes V_i}\otimes h')\circ g_1$.

Correctness. The algorithm consists uniquely of:

- ambient isotopies, when ordering tangle endpoints (Figure 6), sliding strands under coupons (tree leaves Figure 5, and inner tree nodes Figure 7), and flipping coupons upside down (dual morphism, Figure 9), and,
- factorisations of morphisms in a specific order (Figure 8).

By virtue of Theorem 2, ambient isotopies do not affect the output of the algorithm (even though it changes computation), as the algorithm computes a topological invariant of links. By properties of ribbon categories, and more specifically the equivalence relations pictured in Figure 2, the order of factorisation of morphisms does not change the output. In conclusion, the high-level algorithm presented in Sections 3 and 4 outputs the topological invariant $J_L^{\mathcal{C}}(V_1,\ldots,V_m)$ for a $(V_1,\ldots,V_m)$-coloured link diagram L.

5 Algebraic implementation and complexity

For the implementation of the algorithm, we assume that the objects in the category $\mathcal{C}$ are finite dimensional free $\mathcal{R}$-modules, for a commutative ring with unity $\mathcal{R}$ and usual tensor product. Denote the dimension of every link component colour V_i by $N_i := \dim V_i$, and let $N := \max_i\{\dim V_i\}$. Fixing a basis for every V_i, all morphisms in $\mathcal{C}$ – in particular the distinguished braiding, evaluation and co-evaluation, and twist morphisms – are represented by matrices with $\mathcal{R}$ coefficients.

This model is general, and contains all quantum invariants derived from simple Lie algebras. We describe seven elementary matrix operations in Section 5.1, and use them in Section 5.2 as building blocks to implement and analyse the complexity of the algorithm for computing quantum invariants.

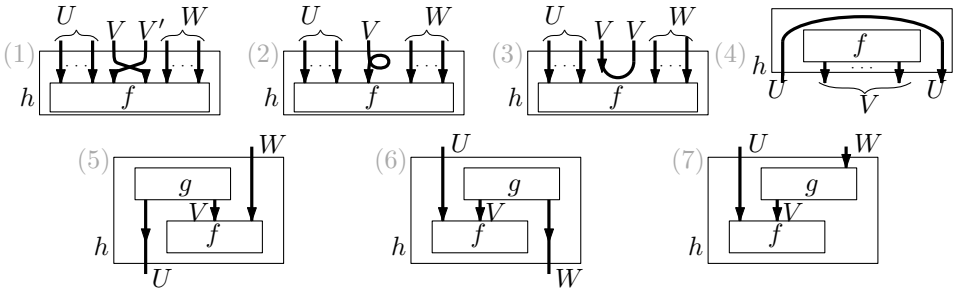


Figure 10 Graphical representation of the seven elementary compositions of morphisms.

5.1 Elementary compositions

We consider the seven elementary compositions of morphisms depicted in Figure 10. They respectively represent the composition with (1) a single braiding, (2) a single twist, (3) a single co-evaluation, (4) a single evaluation. Cases (5), (6), and (7) represent three types of partial compositions of the morphisms f and g. We describe algorithms to perform these compositions on matrices, then use them in Section 5.2 to implement the FPT algorithm described in Sections 3 and 4.

▶ **Lemma 6.** *Consider the elementary morphism compositions in Figure 10 (1), (2), (3), and (4). Let U, V, V', W be finite dimensional free $\mathcal{R}$-modules, with $\dim U = a$, $\dim V = b$, $\dim V' = b'$, and $\dim W = c$. Then, given the matrices for morphisms f, $\theta_V^{\pm}$, $c_{V,V'}^{\pm}$, b_V, and d_U, we can compute the matrix for morphism h in:*

- *$O(a(bb')^2c)$ arithmetic operations in $\mathcal{R}$ for (1),*
- *$O(abc)$ arithmetic operations for (2),*
- *$O(ab^2c)$ arithmetic operations for (3), and*
- *$O(a^2b)$ arithmetic operations for (4).*

The memory complexity of the operation does not exceed the size of the output, which is a row or column vector h containing scalars from $\mathcal{R}$.

Proof.

Figure 10 (1), (2), and (3). All three cases consist of the matrix-vector product $h = (\mathrm{id}_U \otimes M \otimes \mathrm{id}_W) \cdot f$, where M is respectively the $(bb' \times bb')$-matrix $c_{V,V'}^{\pm}$, the $(b \times b)$-matrix $\theta_V^{\pm}$, and the $(b^2 \times 1)$-matrix b_V. Consider M to be an $(m \times m')$-matrix, with coefficients $(M_{i,j})_{1 \le i \le m, 1 \le j \le m'}$. Matrix $(\mathrm{id}_U \otimes M \otimes \mathrm{id}_W)$ has at most m' non-zero coefficients per row. Restricting to these non-zero coefficients, we get the i^{th} entry of h:

$$h_{i,1} = \sum_{k=1\ldots m'} M_{\beta+1,k} \cdot f_{\alpha cm' + \gamma + (k-1)c, 1},$$

where i is uniquely written as $i = \alpha \cdot cm + \beta \cdot c + \gamma$, with $0 \le \alpha \le a-1$, $0 \le \beta \le m-1$, and $1 \le \gamma \le c$. Computing h requires $O(m'|f|)$ arithmetic operations in $\mathcal{R}$, where $|f|$ is the length of vector f, storing $O(|f|)$ scalars from $\mathcal{R}$. Note that matrix $(\mathrm{id}_U \otimes M \otimes \mathrm{id}_W)$ does not need to be explicitly constructed, and only matrix M suffices.

Figure 10 (4) is treated similarly; see [23] for an explicit formula. ◀

▶ **Lemma 7.** *Consider the elementary morphism compositions in Figure 10 (5), (6), and (7). Let U, V, W be finite dimensional free $\mathcal{R}$-modules, with $\dim U = a$, $\dim V = b$, and $\dim W = c$. Then, given the matrices for morphisms f and g, we can compute the matrix for morphism h in $O(abc)$ arithmetic operations in $\mathcal{R}$, and memory complexity $O(ab + bc + ac)$ times the size of a scalar in $\mathcal{R}$.*

Proof. In the same spirit of Lemma 6, the proof exploits the sparsity of tensoring with the identity. Explicit formulae can be found in [23]. ◀

5.2 Implementation and complexity analysis

We implement the algorithm described in Sections 3 and 4 using the elementary compositions of Figure 10. Let N be an upper bound on the dimensions of the different modules U_i, V_j, W_k, over all i, j, k, colouring the components of the link.

Leaf morphisms. The leaf morphisms described in Equations (1)-(4) and Figure 5 are implemented using elementary compositions (1) and (2). By Lemma 6, the complexity is at most $O(N^6)$ arithmetic operations in $\mathcal{R}$.

Sliding under a coupon. The sliding operation as presented in Figure 7 composes a morphism f with a sequence of twist and braiding morphisms. More precisely, let h denote the entire morphism in Figure 7. Starting from the $(O(N^{i+j}) \times 1)$ matrix f, it is computed iteratively applying j times elementary composition (2) for the twists, then $j(j-1)$ times elementary composition (1) for the braidings between V_i- and V_j-strands, and finally ij times elementary composition (1) for the braidings between V_i- and U_j-strands.

During the computation, we maintain a vector of size $(1 \times O(N^{i+j}))$. Applying Lemma 6, the sliding operation runs in $O(j(i+j)N^{i+j+2})$ arithmetic operations in $\mathcal{R}$, storing $O(N^{i+j})$ scalars from $\mathcal{R}$. In the algorithm, $i + j \leq \mathrm{cw}$, the carving-width of the link diagram. Consequently, we get $O(\mathrm{cw}^2\, N^{\mathrm{cw}+2})$ operations, with memory $O(N^{\mathrm{cw}})$.

Construction of evaluations and co-evaluations. The morphism $d_{U_1 \otimes \ldots \otimes U_k}$ appearing in Figure 8 is the result of k elementary compositions of type (4). The morphisms maintained during the computation are of size $(1 \times O(N^{2k}))$. Applying Lemma 6, the computation takes a total of $O(kN^{2k})$ arithmetic operations in $\mathcal{R}$, storing $O(N^{2k})$ scalars from $\mathcal{R}$. The case $b_{W_1 \otimes \ldots \otimes W_j}$ is similar. In the algorithm, k (or j) is at most $\mathrm{cw}/2$. Consequently, the complexity is $O(\mathrm{cw}\, N^{\mathrm{cw}})$ arithmetic operations, storing $O(N^{\mathrm{cw}})$ scalars.

Composition of morphisms. Finally, the compositions of morphisms described in Figure 8 are implemented with a constant number of elementary compositions (5), (6), and (7). Considering Lemma 7, the product abc of dimensions never exceeds $N^{\frac{3}{2}\,\mathrm{cw}}$. Indeed, by property of the tree embedding, we have $i + k, j + k, i + j \leq \mathrm{cw}$ in Figure 8 ; the dichotomy with $k \overset{?}{\leq} \mathrm{cw}/2$ further ensures than no more than $3/2\,\mathrm{cw}$ are involved in the elementary compositions. Consequently, the compositions of Figure 8 are implemented using $O(N^{\frac{3}{2}\,\mathrm{cw}})$ arithmetic operations in $\mathcal{R}$, storing $O(N^{\mathrm{cw}})$ scalars from $\mathcal{R}$.

Overall complexity. In conclusion, we sum up the different steps of the algorithm and its implementation. Let D be a coloured link diagram with n crossings and carving width cw, where the dimension of each colouring module is at most N. The algorithm first computes an optimal tree embedding in $O(\mathrm{poly}(n))$ operations. The tree has size n and width cw. W.l.o.g., we assume the diagram has at least one crossing that is not a twist, and consequently $\mathrm{cw} \geq 4$, the maximal degree of the graph. Considering $\mathrm{cw} \in O(\sqrt{n})$ and $4 \leq \mathrm{cw}+2 \leq \frac{3}{2}\,\mathrm{cw}$, the quantum invariant associated to the colouring is computed in:

$$O(nN^{\frac{3}{2}\,\mathrm{cw}} + n^2 N^{\mathrm{cw}+2}) \text{ arithmetic operations in } \mathcal{R},$$

storing: $O(n)$ words for the diagram, plus $O(N^{\mathrm{cw}})$ scalars from $\mathcal{R}$ for the matrices.

Note that this complexity is expressed in *algebraic operations in* $\mathcal{R}$, when the main Theorem 1 is expressed in *machine operations*. We bound the arithmetic complexity of operations in $\mathcal{R}$ in the following Section 6.

6 Bounding arithmetic complexity and proof of the main theorem

Working with matrices with $\mathcal{R}$-coefficients, for a ring $\mathcal{R}$, allows the algorithm to be applied in great generality. We focus on the case $\mathcal{R} = \mathbb{Z}[q]$, which is sufficient for all $J_L^{\mathfrak{g}}$ invariants of Theorem 1. See [29, Chapter 6] for an explicit construction of the $J_L^{\mathfrak{g}}$.

▶ Remark 8. Note that quantum invariants are usually defined in the category of $\mathbb{Z}[q, q^{-1}]$-modules. Multiplying the braiding, twist, and (co)evaluation matrices by q^a for a large enough, and re-normalising the output, allows us to restrict to the case of $\mathbb{Z}[q]$-modules.

Processing matrix multiplications with coefficients in $\mathbb{Z}[q]$, both degrees of polynomials as well as values of coefficients may blow-up during intermediate computation. Specifically, both arithmetic operations in $\mathcal{R}$ and bit size of $\mathcal{R}$-elements may become *exponential in* n.

In the long version of the article [23], we describe a solution based on Lagrange interpolation and the Chinese remainder theorem to reconstruct polynomial $J_L^{\mathfrak{g}}$ from evaluations, using only integers of bit size $O(\log n)$. Specifically, we prove that, for a fixed category $\mathcal{C}$ of $\mathbb{Z}[q]$-modules:

▶ **Proposition 9.** *The invariant $J_L^{\mathcal{C}}(V_1, \dots, V_m) \in \mathbb{Z}[q]$ of an m-component link L, of size n and coloured with $V_1, \dots, V_m \in \mathcal{C}$ of dimension at most N, can be computed by:*

- *running $O(n^2\sqrt{n})$ times the FPT algorithm described in this article, using matrices with coefficients in $\mathbb{Z}/p\mathbb{Z}$, for various primes $p \in O(n\sqrt{n})$,*
- *and an extra cost of $O(n^4)$ for large integers reconstruction (Chinese remainder theorem) and polynomial reconstruction (Lagrange interpolation).*

The complexity analysis of Section 5.2 and Theorem 9 allows us to conclude the proof of the main Theorem 1. Considering arithmetic operations in $\mathbb{Z}/p\mathbb{Z}$, $p \in O(n\sqrt{n})$, constant time, the algorithm has complexity $O(n^3\sqrt{n}N^{\frac{3}{2}\,\mathrm{cw}} + n^4\sqrt{n}N^{\mathrm{cw}+2})$ *machine operations*.

Conclusion. We have introduced a general sub-exponential, and fixed parameter tractable algorithm to compute any quantum knot invariants derived from a ribbon category, which include infinite families of meaningful knot invariants such as coloured Jones polynomials.

We mention two open research directions:

- This new algorithm offers a new tool to verify important mathematical conjectures [7, 8, 16, 24] on knots that where until now intractable computationally. As a proof of concept, we have implemented the algorithm for coloured Jones polynomials and verified experimentally some behaviours related to the *volume conjecture* and the *slope conjecture* in [23].
- No lower bound is known for the computation of quantum invariants. However, progress [9] on the complexity of *planar Tutte polynomials* under the *exponential time hypothesis* (ETH) may lead to a lower bound of $2^{\Omega(\sqrt{n})}$ for the Jones polynomial, which would imply that our FPT algorithm is optimal up to a constant in the exponent, under ETH.

References

1 Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the jones polynomial. *Algorithmica*, 55:395–421, 2009.

2 Daniel Bienstock. On embedding graphs in trees. *Journal of Combinatorial Theory, Ser. B*, 49(1):103–136, 1990. doi:10.1016/0095-8956(90)90066-9.

3 Benjamin A. Burton. The HOMFLY-PT polynomial is fixed-parameter tractable. In *34th International Symposium on Computational Geometry, SoCG 2018*, pages 18:1–18:14, 2018. doi:10.4230/LIPIcs.SoCG.2018.18.

4 Benjamin A. Burton. The next 350 million knots. In *36th International Symposium on Computational Geometry, SoCG 2020*, pages 25:1–25:17, 2020.

5 Benjamin A. Burton, Clément Maria, and Jonathan Spreer. Algorithms and complexity for Turaev-Viro invariants. *Journal of Applied and Computational Topology*, 2(1-2):33–53, 2018. doi:10.1007/s41468-018-0016-2.

6 Vladimir G. Drinfeld. Quantum groups. In *Proceedings International Congress of Mathematicians*, pages 798–820, 1986.

7 Stavros Garoufalidis. On the characteristic and deformation varieties of a knot. *Geometry & Topology Monographs*, 7:291–309, 2004.

8 Stavros Garoufalidis. The Jones slopes of a knot. *Quantum Topology*, 2(1):43–69, 2011.

9 Leslie Ann Goldberg and Mark Jerrum. Inapproximability of the Tutte polynomial of a planar graph. *Computational Complexity*, 21:605–642, 2012.

10 Qian-Ping Gu and Hisao Tamaki. Optimal branch-decomposition of planar graphs in O(n3) time. *ACM Transaction on Algorithms*, 4(3):30:1–30:13, 2008.

11 Kristóf Huszár and Jonathan Spreer. 3-manifold triangulations with small treewidth. In *35th International Symposium on Computational Geometry, SoCG 2019*, pages 44:1–44:20, 2019. doi:10.4230/LIPIcs.SoCG.2019.44.

12 Kristóf Huszár, Jonathan Spreer, and Uli Wagner. On the treewidth of triangulated 3-manifolds. *Journal of Computational Geometry*, 2(10):70–98, 2019.

13 François Jaeger, Dirk L. Vertigan, and Dominic J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108(1):35–53, 1990.

14 Michio Jimbo. A q-difference analogue of U(g) and the Yang-Baxter equation. *Letters in Mathematical Physics*, 10(1):63–69, 1985.

15 Vaughan F. R. Jones. A polynomial invariant for knots via von Neumann algebras. *Bulletin of the American Mathematical Society*, 12:103–111, 1985.

16 Rinat M. Kashaev. The hyperbolic volume of knots from the quantum dilogarithm. *Letters in Mathematical Physics*, 39(3):269–275, 1997.

17 W. B. Raymond Lickorish. *An Introduction to Knot Theory*. Graduate Texts in Mathematics. Springer-Verlag New York, 1997.

18 Richard J. Lipton and Robert E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

19 Janos A. Makowsky and Julián P. Mariño. The parameterized complexity of knot polynomials. *Journal of Computer and System Sciences*, 67(4):742–756, 2003.

20 Clément Maria and Jessica S. Purcell. Treewidth, crushing, and hyperbolic volume. *Algebraic & Geometric Topology*, 19:2625–2652, 2019.

21 Clément Maria and Owen Rouillé. Computation of large r asymptotics of 3-manifold quantum invariants. In *Proceedings of the 23rd Meeting on Algorithm Engineering and Experiments, ALENEX 2021*. SIAM, 2021.

22 Clément Maria and Jonathan Spreer. A polynomial-time algorithm to compute Turaev-Viro invariants TV4,q of 3-manifolds with bounded first Betti number. *Found. Comput. Math.*, 20(5):1013–1034, 2020. doi:10.1007/s10208-019-09438-8.

23 Clément Maria. Parameterized complexity of quantum knot invariants, 2019. arXiv:1910.00477.

24 Hitoshi Murakami and Jun Murakami. The colored Jones polynomials and the simplicial volume of a knot. *Acta Mathematica*, 186(1):85–104, 2001.

25 Nicolai Y. Reshetikhin and Vladimir G. Turaev. Ribbon graphs and their invariants derived from quantum groups. *Communications in Mathematical Physics*, 127(1):1–26, 1990.

26 Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

27 Saul Schleimer, Arnaud de Mesmay, Jessica Purcell, and Eric Sedgwick. On the tree-width of knot diagrams. *Journal of Computational Geometry*, 10(1):164–180, 2019.

28 Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

29 Vladimir G. Turaev. *Quantum Invariants of Knots and 3-Manifolds*. de Gruyter Studies in Mathematics. Walter de Gruyter & Co., Berlin, revised edition, 2010.

30 Vladimir G. Turaev and Oleg Y. Viro. State sum invariants of 3-manifolds and quantum $6j$-symbols. *Topology*, 31(4):865–902, 1992.