

Towards a Complexity-Theoretic Dichotomy for TQFT Invariants

Nicolas Bridges   

Department of Mathematics, Purdue University, West Lafayette, IN, USA

Eric Samperton   

Departments of Mathematics and Computer Science, Purdue University, West Lafayette, IN, USA

Abstract

We show that for any fixed $(2+1)$ -dimensional TQFT over \mathbb{C} of either Turaev-Viro-Barrett-Westbury or Reshetikhin-Turaev type, the problem of (exactly) computing its invariants on closed 3-manifolds is either solvable in polynomial time, or else it is $\#P$ -hard to (exactly) contract certain tensors that are built from the TQFT's fusion category. Our proof is an application of a dichotomy result of Cai and Chen [J. ACM, 2017] concerning weighted constraint satisfaction problems over \mathbb{C} . We leave for future work the issue of reinterpreting the conditions of Cai and Chen that distinguish between the two cases (*i.e.* $\#P$ -hard tensor contractions vs. polynomial time invariants) in terms of fusion categories. We expect that with more effort, our reduction can be improved so that one gets a dichotomy directly for TQFTs' invariants of 3-manifolds rather than more general tensors built from TQFTs' fusion categories.

2012 ACM Subject Classification Mathematics of computing \rightarrow Geometric topology; Theory of computation \rightarrow Problems, reductions and completeness; Theory of computation \rightarrow Quantum complexity theory

Keywords and phrases Complexity, topological quantum field theory, dichotomy theorems, constraint satisfaction problems, tensor categories

Digital Object Identifier 10.4230/LIPIcs.TQC.2025.5

Related Version *Full Version*: <https://arxiv.org/abs/2503.02945>

Funding Both authors supported in part by NSF CCF grant 2330130.

1 Introduction

1.1 Main results

Quantum computation – especially *topological* quantum computation – motivates a number of complexity-theoretic questions concerning TQFT invariants of manifolds, particularly in dimensions 2 and 3. One of the most central is to classify “anyonic systems” according to whether or not they are powerful enough to (approximately) encode arbitrary quantum circuits over qubits. Anyons that are powerful in this way are important because (in theory) it should be possible to build fault tolerant quantum computers using them [12, 10]. We refer the reader to [23, 19] for a broad review of the mathematical side of these matters and Subsection 1.3 for more discussion. For now, we simply note that the Property F conjecture of Naidu and Rowell is currently the only concrete, published formulation of a proposed (partial) answer to this classification question that we know. The conjecture is surprisingly easy to formulate: the possible braidings of n copies of a simple anyon X in a unitary modular tensor category \mathcal{B} generate only finitely many unitaries for each n (and, hence, are not “braiding universal” for quantum computation) if and only if the square of the quantum dimension of X is an integer $d_X^2 \in \mathbb{Z}$ [17].



© Nicolas Bridges and Eric Samperton;
licensed under Creative Commons License CC-BY 4.0

20th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2025).

Editor: Bill Fefferman; Article No. 5; pp. 5:1–5:21



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this work, we will not attack the Property F conjecture directly. However, our main result has a similar spirit and shares the same motivations. See Subsection 1.3 for some discussion of these points.

► **Theorem 1.**

- (a) *Fix a spherical fusion category \mathcal{C} over \mathbb{C} , presented skeletally with all data given as algebraic numbers over \mathbb{Q} . Then $\#\text{CSP}(\mathcal{F}_{\mathcal{C}})$ —the problem of contracting tensor networks defined from \mathcal{C} —is either solvable in polynomial time or $\#\text{P}$ -hard. Moreover, if M is a closed, oriented, triangulated 3-manifold (treated as computational input), then either the problem of computing the Turaev-Viro-Barrett-Westbury invariant $|M|_{\mathcal{C}} \in \mathbb{C}$ is solvable in (classical) polynomial time or $\#\text{CSP}(\mathcal{F}_{\mathcal{C}})$ is $\#\text{P}$ -hard.*
- (b) *Fix a modular fusion category \mathcal{B} over \mathbb{C} , presented skeletally with all data given as algebraic numbers over \mathbb{Q} . Then $\#\text{CSP}(\mathcal{F}_{\mathcal{B}})$ —the problem of contracting tensor networks built from \mathcal{B} —is either solvable in polynomial time or $\#\text{P}$ -hard. Moreover, if M is a closed, oriented 3-manifold encoded via a surgery diagram (treated as computational input), then either the problem of computing the Reshetikhin-Turaev invariant $\tau_{\mathcal{B}}(M) \in \mathbb{C}$ is solvable in (classical) polynomial time or $\#\text{CSP}(\mathcal{F}_{\mathcal{B}})$ is $\#\text{P}$ -hard.*

Two routine points of clarification are due.

First, we note that all fusion and modular categories over \mathbb{C} admit finite skeletal presentations using algebraic numbers over \mathbb{Q} . This is because the defining equations for the skeletal data are all algebraic over \mathbb{Q} . In particular, since we are interested in how the complexity of $|M|_{\mathcal{C}}$ or $\tau_{\mathcal{B}}(M)$ depends on variable M for fixed \mathcal{C} or \mathcal{B} , there is no harm in assuming that \mathcal{C} and \mathcal{B} are encoded in this way.

Second, as usual in computational 3-manifold topology, to say that a problem whose input is a triangulated 3-manifold is solvable in polynomial time means that there exists an algorithm to solve the problem that runs in time polynomial in the size of the triangulation. For a 3-manifold presented via integral surgery on a link diagram in S^3 , the algorithm must run in time jointly polynomial in the crossing number of the link diagram, the number of components of the link, and the absolute values of the surgery coefficients.¹

We refer the reader to [7] for further elaboration of both of these matters.

We now explain briefly the meaning and importance of dichotomy theorems within complexity theory. Of course, it is an infamous open problem to show that $\text{P} \neq \text{NP}$ (the two complexity classes might be equal, but most experts do not expect this to be the case). To establish this inequality it is necessary and sufficient to show that there exists an NP -complete problem with no polynomial-time algorithm. Intriguingly, Ladner showed that if $\text{P} \neq \text{NP}$, then there exist problems in NP that are neither in P nor NP -complete [16]. These are usually referred to as *NP-intermediate*. In other words, an NP -intermediate problem is a problem in NP that is neither in P nor NP -hard. Intuitively, Ladner’s theorem shows that if one considers a family of decision problems, then it need not be the case that every problem in the family is either “easy” (that is, in P) or “hard” (that is, NP -hard)—there could be problems that have intermediate complexity. When a given family of problems has the property that none of the problems has intermediate complexity, then one says that the family satisfies a *dichotomy theorem*. The archetypical dichotomy theorem was established by Schaefer, who showed that Boolean satisfiability problems in generalized conjunctive form (where the clauses are taken from a finite set of constraints) satisfy a dichotomy theorem (with respect to the set of constraints) [21].

¹ In particular, we might understand the surgery coefficients as being expressed in unary, not binary. (If we used the latter, then we would not be able to build a triangulation from a surgery diagram in polynomial time.)

In the case of our Theorem 1, we interpret (2+1)-d TQFT invariants as generalized (*i.e.* \mathbb{C} -valued instead of \mathbb{N} -valued) counting problems parametrized by spherical fusion categories \mathcal{C} and modular fusion categories \mathcal{B} (the categories are analogs of the allowed local constraints in Schaefer’s dichotomy). Our results establish the dichotomy that either a function of the type $M \mapsto |M|_{\mathcal{C}} \in \mathbb{C}$ or $M \mapsto \tau_{\mathcal{B}}(M) \in \mathbb{C}$ is “easy” to compute (polynomial time) or else it is “hard” to contract certain tensors built from the category \mathcal{C} or \mathcal{B} ($\#P$ -hard). In this way, there are no (2+1)-d TQFTs whose tensors are of “intermediate” complexity. In fact, we conjecture that the same can be said directly of the TQFT’s *invariants of 3-manifolds per se*. Let us expound on these points now.

1.2 Mapping the dichotomy

Whether or not $M \mapsto |M|_{\mathcal{C}}$ or $M \mapsto \tau_{\mathcal{B}}(M)$ is computable in polynomial time depends on \mathcal{C} and \mathcal{B} . Having established Theorem 1 – which only asserts the *existence* of a dichotomy – it is natural to wonder where one should draw the line between easy and hard. Better yet, ideally, one would like to be able to prove that the dichotomy of Theorem 1 is *effective*, meaning, given \mathcal{C} or \mathcal{B} , there exists a polynomial-time algorithm to decide precisely when the category falls into the easy case (here “polynomial-time” means in the size of the skeletalization of \mathcal{C} or \mathcal{B}). Our proof of Theorem 1 relies on the main result of Cai and Chen’s work [2], which establishes a dichotomy theorem for a generalized type of “solution counting” to constraint satisfaction problems $\#CSP(\mathcal{F})$ with a fixed “ \mathbb{C} -weighted constraint family” \mathcal{F} . We carefully define $\#CSP(\mathcal{F})$ in Subsection 2.1, but here we note that not only do Cai and Chen prove that for every choice of constraint family \mathcal{F} , $\#CSP(\mathcal{F})$ is either $\#P$ -hard or computable in polynomial time – they also provide three necessary and sufficient conditions that *characterize precisely* which \mathcal{F} allow for polynomial time solutions to $\#CSP(\mathcal{F})$. These conditions are called “block orthogonality,” “Mal’tsev” and “Type Partition”. Our proof of Theorem 1 consists in converting a spherical fusion category \mathcal{C} or modular fusion category \mathcal{B} into an appropriate constraint family $\mathcal{F}_{\mathcal{C}}$ or $\mathcal{F}_{\mathcal{B}}$ such that computing $|M|_{\mathcal{C}}$ or $\tau_{\mathcal{B}}(M)$ is equivalent to computing an instance (depending on M) of a problem in $\#CSP(\mathcal{F}_{\mathcal{C}})$ or $\#CSP(\mathcal{F}_{\mathcal{B}})$, respectively. In particular, for the constraint families $\mathcal{F}_{\mathcal{C}}$ and $\mathcal{F}_{\mathcal{B}}$ we shall build, it should be possible to interpret the three conditions of Cai and Chen directly in terms of the categories \mathcal{C} and \mathcal{B} . It is beyond the scope of the present work to attempt to accomplish this. However, we believe this is an important problem, since it should shed light on variations of the Property F conjecture related to anyon classification, as we explain in the next subsection.

Let us now address the more important deficiency of Theorem 1, alluded to at the end of the previous subsection: it would be better to get an outright dichotomy for 3-manifold invariants, and not just general tensors derived from a fusion category. To this end, Theorem 1 can be understood as a first step towards proving the following more desirable result.

► Conjecture 2.

- (a) Fix a spherical fusion category \mathcal{C} over \mathbb{C} , presented skeletally with all data given as algebraic numbers over \mathbb{Q} . If M is a closed, oriented, triangulated 3-manifold (treated as computational input), then computing the Turaev-Viro-Barrett-Westbury invariant $|M|_{\mathcal{C}} \in \mathbb{C}$ is either solvable in (classical) polynomial time or is $\#P$ -hard.
- (b) Fix a modular fusion category \mathcal{B} over \mathbb{C} , presented skeletally with all data given as algebraic numbers over \mathbb{Q} . If M is a closed, oriented 3-manifold encoded via a surgery diagram (treated as computational input), then the problem of computing the Reshetikhin-Turaev invariant $\tau_{\mathcal{B}}(M) \in \mathbb{C}$ is either solvable in (classical) polynomial time or is $\#P$ -hard.

See Subsection 3.4 for some discussion of how we expect one might get started on proving this conjecture – in particular, the relevance of holant problems [3].

1.3 Implications for “anyon classification”

Theorem 1 and Conjecture 1 assert that for certain precise formulations of the problem of “anyon classification,” whatever the “type” is for a given modular fusion category \mathcal{B} , it can only be one of two things, with no “intermediate” cases. In order to explain this more carefully, we pause to note the many ways one can make the problem of anyon classification precise, and situate our result exactly in this milieu. Moving from the more “purely mathematical” to the more “applied” end of the spectrum, “anyon classification” could mean any of the following precise problems:

1. Algebraic classification of unitary modular fusion categories (MFCs) up to ribbon tensor equivalence.
 - Much of the literature on fusion categories can be considered as contributing to this problem.
 - Presumably one would be satisfied with a solution to this problem “modulo finite group theory.”
2. Algebraic classification of simple objects X in unitary MFCs \mathcal{B} according to whether or not the braid group representations $B_n \rightarrow U(\text{End}_{\mathcal{B}}(X^{\otimes n}))$ have finite image, dense image, or something else.
 - The Property F conjecture is of course directly related to this matter.
 - One can generalize this question to consider mapping class group representations of higher genus surfaces with different types of anyons on them.
3. Complexity-theoretic classification of MFCs according to how easy or hard it is to *exactly compute* their Reshetikhin-Turaev 3-manifold invariants (as algebraic numbers over \mathbb{Q}).
 - **Our Theorem 1 is situated here – almost!** We have established a dichotomy of the form either “3-manifold invariants easy” or “tensors in the category are hard to contract”. Our results represent a non-trivial step towards the desired dichotomy of Conjecture 2: “invariants easy” or “invariants hard”.
 - One can ask this question for *restricted classes of 3-manifolds* (such as “knots in S^3 ” or “links in S^3 ” or “integer homology spheres”), and the classification may change [7].
4. Complexity-theoretic classification of MFCs according to how easy or hard it is to “*approximate*” their Reshetikhin-Turaev 3-manifold invariants.
 - There are different types of approximations one might ask for. *A priori*, each type should be understood as giving a different version of this question.
 - “Exactly compute” is one way to “approximate.”
 - Pioneering works of Freedman, Kitaev, Larsen and Wang show that for certain approximation schemes, there exists unitary MFCs \mathcal{B} for which the ability to approximate their 3-manifold invariants is equivalent in power to BQP (bounded-error quantum polynomial time) [9, 10]. In particular, their work established the original paradigm for topological quantum computation via anyon braiding.
 - Kuperberg showed that results for one type of approximation can have important implications for other types of approximations [13]. In particular, the kinds of approximations that a quantum computer can efficiently make for Reshetikhin-Turaev invariants are (in general/worst case) not precise enough to do anything useful for distinguishing 3-manifolds even if their invariants are promised to be unequal by a large amount.

5. Complexity-theoretic classification of MFCs according to whether or not they support universal quantum computation with braiding *and adaptive anyonic charge measurements*.
 - Quantum computation using braidings and charge measurements of anyons in a fixed unitary MFC is often called “topological quantum computing with adaptive charge measurement.”
 - Our entirely subjective opinion is that this is the most important flavor of anyon classification, at least when considered from the perspective of the goal of actually building a universal, fault-tolerant quantum computer.
 - Even unitary MFCs whose anyons all have Property F (and, hence, are not universal via braiding alone) can be universal when braiding is supplemented with charge measurements, see e.g. [6].
 - While adaptive charge measurement is generally considered fault-tolerant for topological reasons, unlike the case of braiding-only topological quantum computing, the amplitudes with which one performs a quantum computation in this paradigm are not (normalizations of) Reshetikhin-Turaev invariants of 3-manifolds.

There are known relations between these different classification problems. For example, on one hand, if X is an anyon such that $B_n \rightarrow U(\text{End}_{\mathcal{B}}(X^{\otimes n}))$ is dense, then the Solovay-Kitaev theorem implies that \mathcal{B} supports universal topological quantum computation via braiding (without needing adaptive charge measurement). On the other hand, if X has Property F, then it is known that braiding with X is never powerful enough to encode all of BQP in its braidings. This latter point was the main motivation for the Property F conjecture in the first place, since one would like to rule out the “obviously” un-useful anyons easily.

To understand the potential usefulness of Theorem 1 or Conjecture 2, it is perhaps helpful to pull on the thread of these motivations for the Property F conjecture a bit more so that we can compare and contrast.

On one hand, there is no “unconditional” implication known between classification problems (2) and (4) above in either direction, except if we condition on properties in a way we have already mentioned, namely: if an anyon has Property F, then it is definitely not braiding universal, while if an anyon has dense braidings, then it is universal. This is not “unconditional” in the sense that as far as problem (2) is concerned, there is a third case that remains to be addressed: anyons with braidings that are neither dense nor have Property F. Do they even exist? If so, what are we to make of them? Are they universal or not? Maybe sometimes they are and sometimes they are not? Conversely, if an anyon is braiding universal, does it necessarily have dense braid group representations? These are interesting questions worth pursuing, but it could require quite a bit of effort to resolve each of them.

On the other hand, there is an “unconditional” connection between (3) and (4) in at least one direction: (3) is simply the special case of (4) where the type of “approximation” is chosen to be “exact computation.” So classification problem (3) might be understood as a warm-up to the version of problem (4) where the type of approximation is not “exact”, but is instead the kind of approximation relevant to topological quantum computing. (For the sake of space, we refrain from precisely defining this type of approximation here; see the intro discussions of [13] or [20].) The key technical issue that this perspective highlights is the following: even categories whose anyons all have property F (and thus are not braiding universal) can have $\#P$ -hard invariants [11, 14, 15]. Hence, more work needs to be done to properly understand the relationship between the BQP-universality of anyon braidings in a given modular fusion category \mathcal{B} and $\#P$ -hardness of (exactly) computing $\tau_{\mathcal{B}}(M)$ on 3-manifolds M . At the end of the day this is not so different from the situation between (2) and (4).

However, we conclude this discussion by noting that it is conceivable there exists a very tight connection between anyon classification problems (3) and (5) (while there is essentially no way to relate (2) and (5)). Indeed, one might reasonably guess that $\#P$ -hardness for the exact calculation of invariants implies that topological quantum computing with adaptive charge measurements is always sufficient to generate BQP-universal topological gates. This guess would be consistent with all known examples. We plan to explore these matters in future work.

1.4 Outline

Subsection 2.1 briefly reviews the definition of $\#CSP(\mathcal{F})$, as well as Cai and Chen's dichotomy theorem for these problems. Subsection 2.2 contains the proof of part (a) of Theorem 1. Subsection 2.3 contains some preliminary results about the graphical calculus in fusion categories needed for the proof of part (b). The proof of part (b) itself is relegated to Appendix A for the sake of space. Section 3 contains some further discussion.

2 Proof of Theorem 1

2.1 Cai and Chen's dichotomy for weighted CSPs

Before proving either part of our main theorem, we review the definition of $\#CSP(\mathcal{F})$, following [2]:

- We fix a finite set $D = \{1, \dots, d\}$ called the *domain* (which, by an abuse of notation, we will suppress from the notation $\#CSP(\mathcal{F})$).
- We fix a (\mathbb{C} -valued) *weighted constraint family* $\mathcal{F} = \{f_1, \dots, f_h\}$, where each f_i is a \mathbb{C} -valued function $f_i : D^{r_i} \rightarrow \mathbb{C}$ for some $r_i \geq 1$ called the *arity* of f_i . We assume all the values that the f_i assume are encoded as algebraic numbers over \mathbb{Q} .
- An *instance* I of $\#CSP(\mathcal{F})$ consists of a tuple $\mathbf{x} = (x_1, \dots, x_n)$ of variables over D (which will be suppressed in our notation) and a set I of tuples (f, i_1, \dots, i_r) in which f is an r -ary function from \mathcal{F} and $i_1, \dots, i_r \in \{1, \dots, n\}$ are indices of the variables in \mathbf{x} .
- The *output* of $\#CSP(\mathcal{F})$ on instance I is the algebraic number $Z(I) \in \mathbb{C}$ given by

$$Z(I) \stackrel{\text{def}}{=} \sum_{\mathbf{x} \in D^n} F_I(\mathbf{x}),$$

where

$$F_I(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{(f, i_1, \dots, i_r) \in I} f(x_{i_1}, \dots, x_{i_r}).$$

The main result of [2] is

► **Theorem 3** (Thm. 1, [2]). *Given any constraint set \mathcal{F} as above, $\#CSP(\mathcal{F})$ is either computable in polynomial time or $\#P$ -hard.*

2.2 Proof of Theorem 1(a)

Proof. Let \mathcal{C} be a spherical fusion category over \mathbb{C} . To prove part (a) of Theorem 1, it suffices – thanks to Theorem 3 – to build a domain $D_{\mathcal{C}}$ and weighted constraint set $\mathcal{F}_{\mathcal{C}}$ with the following property: there exists a polynomial time algorithm that converts a triangulated 3-manifold M into an instance I_M of $\#CSP(\mathcal{F}_{\mathcal{C}})$ such that

$$Z(I_M) = |M|_{\mathcal{C}}.$$

$$\phi(j_1, j_2, j_3; k) = \begin{array}{c} \text{---} k \text{---} \\ \downarrow j_1 \quad \downarrow j_2 \\ \text{---} k \text{---} \\ \uparrow j \end{array}$$

■ **Figure 1** A 3j+1k-symbol.

Readers already familiar with the state-sum formula for TVBW invariants will notice that the definition of $Z(I)$ is quite similar in spirit. The goal of the present proof is simply to make this similarity precise.

To this end, let us recall the state-sum formula for $|M|_{\mathcal{C}}$:

$$|M|_{\mathcal{C}} = \mathcal{D}^{-2|V_M|} \sum_{\substack{L: E_M \rightarrow \text{Irr}(\mathcal{C}) \\ F_L: F_M \rightarrow \mathcal{N} \text{ consistent w/ } L}} \frac{\prod_{e \in E_M} \dim(L(e))^2 \prod_{t \in T_M} |t^L|}{\prod_{f \in F_M} |f^L|}$$

where our notation is as follows:

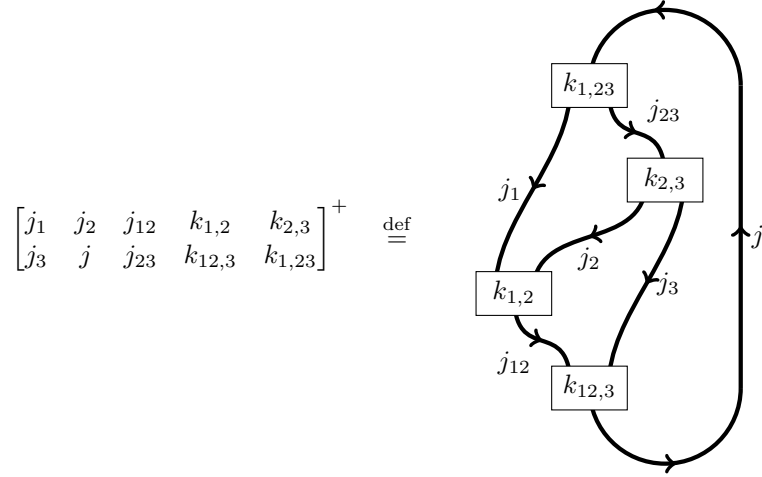
- V_M is the ordered list of vertices in the triangulation M and \mathcal{D} is the total quantum dimension of \mathcal{C} .
- E_M is the set of edges in the triangulation M and $\text{Irr}(\mathcal{C})$ is set of simple objects in the given skeletalization of \mathcal{C} .
- F_M is the set of faces in the triangulation M and \mathcal{N} is the set of labels of the trivalent Hom spaces

$$\text{Hom}(k, i \otimes j) = \text{span} \left\{ \begin{array}{c} i \quad j \\ \swarrow \quad \searrow \\ \boxed{\alpha} \\ \downarrow \\ k \end{array} \right\}_{\alpha=1, \dots, N_{ij}^k}$$

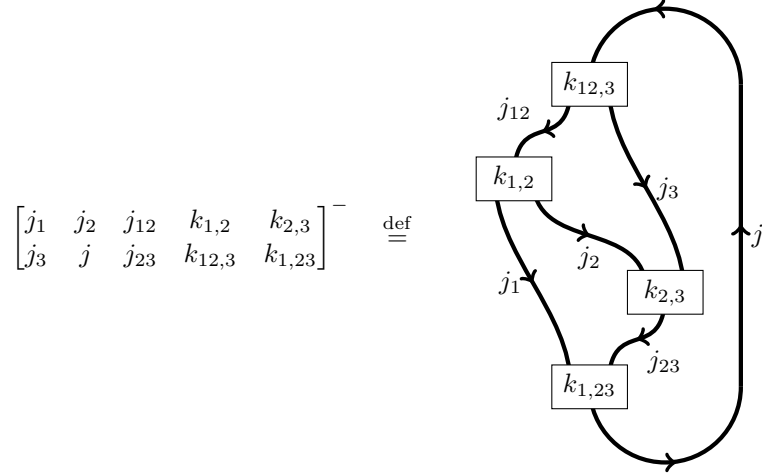
and $|f^L|$ is the 3j+1k-symbol obtained by evaluating the face f with a given labeling L of the edges and faces of M (See Figure 1).

- T_M is the set of tetrahedra in the triangulation M , and $|t^L|$ is the 6j+4k-symbol obtained by evaluating the tetrahedron t with a given labeling L of the edges and faces of M , where we take into account whether the orientation of t given by the orientation of M matches the induced orientation given by the ordering of the vertices. This will be made more precise below.

Since we assume \mathcal{C} is not necessarily multiplicity-free, then instead of 6j-symbols, we will be using so-called 6j+4k-symbols $\begin{bmatrix} j_1 & j_2 & j_3 & k_{1,2} & k_{2,3} \\ j & j_{12} & j_{23} & k_{12,3} & k_{1,23} \end{bmatrix}^{\pm}$, which are defined by the contraction of a specific colored graph.



This defines the “positive” $6j+4k$ -symbols. We also define a “negative” version of the $6j+4k$ -symbols. We will call them negative $6j+4k$ -symbols since they correspond to negatively-oriented tetrahedra with respect to the standard orientation on \mathbb{R}^3 :



Here, $j_1, j_2, j_3, j, j_{12}, j_{23}$ are simple objects, $k_{1,2} \in \{0, \dots, N_{j_1 j_2}^{j_{12}}\}$, $k_{2,3} \in \{0, \dots, N_{j_2 j_3}^{j_{23}}\}$, $k_{12,3} \in \{0, \dots, N_{j_{12} j_3}^j\}$, and $k_{1,23} \in \{0, \dots, N_{j_1 j_{23}}^j\}$.

We now have enough to identify our domain and weighted constraint set. Define

$$D_{\mathcal{C}} \stackrel{\text{def}}{=} \text{Irr}(\mathcal{C}) \sqcup \mathcal{N} \sqcup \{*\}.$$

Now extend the $6j+4k$ symbols to be 10-ary functions on our domain $D_{\mathcal{C}}$ in the “trivial” way:

$$\Delta^+(x_1, \dots, x_{10}) \stackrel{\text{def}}{=} \begin{cases} \begin{bmatrix} x_1 & x_2 & x_5 & x_7 & x_8 \\ x_3 & x_4 & x_6 & x_9 & x_{10} \end{bmatrix}^+ & \text{if } x_1, \dots, x_6 \in \text{Irr}(\mathcal{C}), x_7, \dots, x_{10} \in \mathcal{N}, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\Delta^-(x_1, \dots, x_{10}) \stackrel{\text{def}}{=} \begin{cases} \begin{bmatrix} x_1 & x_2 & x_5 & x_7 & x_8 \\ x_3 & x_4 & x_6 & x_9 & x_{10} \end{bmatrix}^- & \text{if } x_1, \dots, x_6 \in \text{Irr}(\mathcal{C}), x_7, \dots, x_{10} \in \mathcal{N}, \\ 0 & \text{otherwise.} \end{cases}$$

We similarly define 4-ary functions on our domain using the 3j+1k-symbols ϕ by taking

$$\Phi^{-1}(x_1, x_2, x_3, x_4) \stackrel{\text{def}}{=} \begin{cases} \phi(x_1, x_2, x_3; x_4)^{-1} & \text{if } x_1, x_2, x_3 \in \text{Irr}(\mathcal{C}), x_4 \in \mathcal{N}, \\ 0 & \text{otherwise.} \end{cases}$$

And we define 1-ary functions using the quantum dimensions of simple objects:

$$\text{d}^2(x) \stackrel{\text{def}}{=} \begin{cases} \dim(x)^2 & \text{if } x \in \text{Irr}(\mathcal{C}), \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we define a 1-ary function to encode the total quantum dimension of \mathcal{C} :

$$\mathcal{D}^{-2} \stackrel{\text{def}}{=} \mathcal{D}^{-2}(x) \stackrel{\text{def}}{=} \begin{cases} \left(\sum_{j \in \text{Irr}(\mathcal{C})} \dim(j)^2 \right)^{-1} & \text{if } x = *, \\ 0 & \text{otherwise.} \end{cases}$$

Using these functions, we define our weighted constraint family

$$\mathcal{F}_{\mathcal{C}} \stackrel{\text{def}}{=} \{\Delta^{\pm}, \Phi^{-1}, \text{d}^2, \mathcal{D}^{-2}\}.$$

Our next goal is to describe how to convert a triangulation M of an oriented manifold into an instance I_M of $\#\text{CSP}(\mathcal{F}_{\mathcal{C}})$.

The data of M is comprised of:

- An ordered list of vertices $\{v_1, \dots, v_a\}$.
- A list of oriented edges $\{e_1(v_{i_1}, v_{i_2}), \dots, e_b(v_{b_1}, v_{b_2})\}$ where $e_i(v_{i_1}, v_{i_2})$ means that e_i is an edge connecting v_{i_1} to v_{i_2} . (Note that these orientations are chosen arbitrarily.)
- A list of oriented faces $\{f_1(e_{i_1}, e_{i_2}, e_{i_3}), \dots, f_c(e_{c_1}, e_{c_2}, e_{c_3})\}$ where $f_i(e_{i_1}, e_{i_2}, e_{i_3})$ means that f_i is a face whose boundary consists of the edges e_{i_1} , e_{i_2} , and e_{i_3} . (Note that the orientations of the faces need to be consistent with the edge orientations in any way.)
- A list of tetrahedra $\{t_1, \dots, t_d\}$ where $t_i = t_i(f_{i_1}, \dots, f_{i_4})$ means that t_i is a tetrahedron with faces given by f_{i_1}, \dots, f_{i_4} .
- To encode the orientation of M , each tetrahedron t_i is endowed with a sign $+$ or $-$ to indicate the local orientation inside that tetrahedron.²

For a given triangulation M as described, define a tuple

$$\mathbf{x}_M \stackrel{\text{def}}{=} (x_1, \dots, x_a, y_1, \dots, y_b, z_1, \dots, z_c)$$

that has a variable for each vertex, edge and face in M . We now describe how to build the desired instance I_M of $\#\text{CSP}(\mathcal{F}_{\mathcal{C}})$. It will be clear from the construction that the mapping $M \mapsto I_M$ can be done in polynomial time in the size of M .

First we put the functions $\mathcal{D}^{-2}(x_1), \dots, \mathcal{D}^{-2}(x_a)$ and $\text{d}^2(y_1), \dots, \text{d}^2(y_b)$ in I_M for every vertex and edge of M . For each face $f_j(e_{j_1}, e_{j_2}, e_{j_3})$, we include $\Phi^{-1}(y_{j_1}, y_{j_2}, y_{j_3}, z_j)$. Finally, for each tetrahedron t_i , we include either

$$\Delta^+(y_{j_1}, \dots, y_{j_6}, z_{i_1}, \dots, z_{i_4}),$$

² These signs must assemble to give a $\{\pm\}$ -valued 0-cocycle on the dual cellulation. This condition could be easily checked, but for our purposes it is simply part of the data structure of M , and so this condition can be assumed to be met as a promise. This condition is not necessary for our proof (although it is necessary for the proof that $|M|_{\mathcal{C}}$ is an invariant of M).

or

$$\Delta^-(y_{j_1}, \dots, y_{j_6}, z_{i_1}, \dots, z_{i_4}),$$

where t_i has faces $f_{i_1}(e_{j_1}, e_{j_2}, e_{j_5})$, $f_{i_2}(e_{j_5}, e_{j_3}, e_{j_4})$, $f_{i_3}(e_{j_3}, e_{j_2}, e_{j_6})$, and $f_{i_4}(e_{j_6}, e_{j_1}, e_{j_4})$. To determine whether we should include Δ^+ or Δ^- for t_i , we check if the orientation of t_i given by the orientation of M matches the induced orientation by the ordering of the vertices; if they match, then we use Δ^+ , and otherwise we use Δ^- .

This I_M defines an instance of $\#\text{CSP}(\mathcal{F}_C)$ that computes the Turaev-Viro invariant for M . Indeed, plugging in the definitions of our constraint functions, we get

$$Z(I_M) = \sum_{\mathbf{x} \in D^{a+b+c}} \prod_{v=1}^a \mathcal{D}^{-2}(x_v) \prod_{e=1}^b d^2(y_e) \prod_{F_i=(E_{i_1}, E_{i_2}, E_{i_3})} \Phi^{-1}(y_{i_1}, y_{i_2}, y_{i_3}, z_i) \prod_{T_i=(F_{i_1}, \dots, F_{i_4})} \Delta^{\eta_i}(y_{j_1}, \dots, y_{j_6}, z_{i_1}, \dots, z_{i_4}))$$

where $\eta_i = +$ if the orientation of T_i given by the orientation of M matches the induced orientation by the ordering of the vertices, and $\eta_i = -$ otherwise. *A priori*, $Z(I_M)$ includes a sum over more types of labelings than the state-sum formula for $|M|_C$. However, because of how we have chosen to define the functions in the constraint family \mathcal{F}_C , all of these additional terms in the sum vanish. To see this, first note that when $x_1, \dots, x_v \neq *$, the entire term is 0. In particular, this means all non-zero terms have a common factor of the global quantum dimension to the $-a$ power, and hence we can pull it out as the normalizing factor. Similarly, when the edges are not labeled by elements of the domain D_C that are not simple objects of \mathcal{C} , or the faces are not labeled with multiplicities, the terms are zero. We have furthermore arranged so that when the labeling of the edges and faces is not admissible, the $6j+4k$ -symbol for that term vanishes. Therefore, the only surviving terms in the sum $Z(I_M)$ are the \mathbf{x} which define admissible labelings of the edges and faces of M . It is then straightforward to see $Z(I_M) = |M|_C$ recovers the Turaev-Viro invariant. ◀

2.3 Graphical calculus preliminaries

Before proving part (b) of Theorem 1, we establish a technical result about the graphical calculus in a spherical fusion category. The result is likely well-known to experts, but does not appear in the literature anywhere that we are aware of. We begin by reviewing what we need of closed trivalent graphs in S^2 .

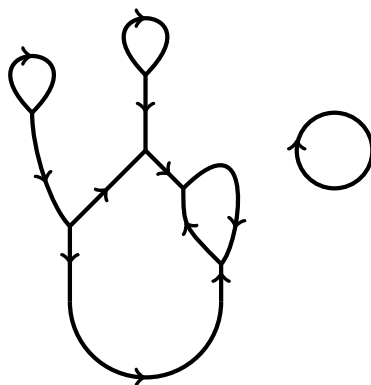
For our purposes, a (*closed*) *trivalent graph* Γ in \mathbb{R}^2 (or $S^2 = \mathbb{R}^2 \cup \{\infty\}$) has:

- A finite collection V of vertices v_1, v_2, \dots, v_n , where $v_i \in \mathbb{R} \times \{i\}$
- A collection E of directed edges e_1, e_2, \dots, e_k in S^2

subject to the conditions that

- Each edge e_i is either a loop disjoint from V or an arc connecting two (not necessarily distinct) vertices, with an interior that is disjoint from all vertices in V .
- If v is a vertex, then there is an open disk neighborhood $D(v)$ so that $D(v) \cap \Gamma$ has three arcs (coming from intersections of $D(v)$ with E , not necessarily distinct) emanating from v with one arc parallel to the vector $\langle 0, 1 \rangle$, one arc parallel to the vector $\langle 1, 1 \rangle$, and one arc parallel to the vector $\langle -1, 1 \rangle$.

Such a graph Γ is closed in the sense that there are no vertices that are involved in precisely one half-edge.



■ **Figure 2** A trivalent graph in \mathbb{R}^2 which exhibits all three types of edges.

We will also need to consider closed trivalent graphs which have crossings. We will call these *crossed (closed) trivalent graphs*. Crossed trivalent graphs are trivalent graphs where they are allowed to have finitely many double points in the interior of its edges. These double points must indicate which segment of the edge crosses “over” or “under” the other.

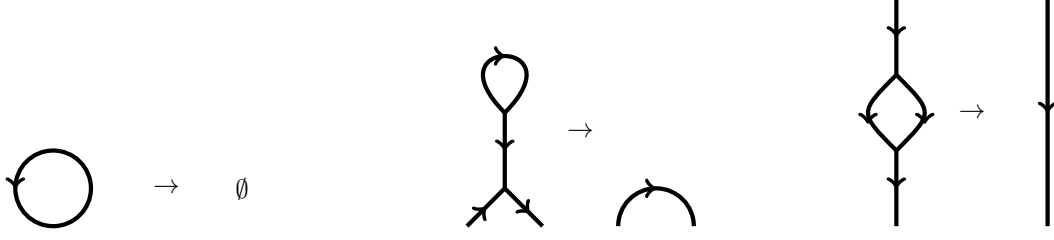
Recall that computing the Reshetikhin-Turaev invariant $\tau_{\mathcal{B}}(M)$ of a 3-manifold M presented by a surgery diagram involves a process where we interpret a coloring of the components of the diagram by simple objects of \mathcal{B} as an endomorphism of the tensor unit $\mathbf{1}$ of \mathcal{B} . Since $\mathbf{1}$ is itself simple, such a coloring gives rise to an endomorphism $\mathbf{1} \rightarrow \mathbf{1}$, which in turn can be identified with a complex number because $\text{End}(\mathbf{1}) = \mathbb{C}$. The invariant $\tau_{\mathcal{B}}(M)$ is then (roughly) the sum of all of these numbers over all choices of colorings of the surgery diagram of M . For any single coloring, the complex number associated to it can generally be understood as the result of a sequence of tensor contractions on a tensor induced by that coloring. Moreover, this sequence of tensor contractions can be represented diagrammatically, using a small number of standard diagrammatic operations that are determined from the data of the modular fusion category \mathcal{B} . We call these operations Circle Removal, Tadpole Trim, Bubble Pop, F -Move, and Vertex Spiral (aka “bending” moves); see Figures 3-5. We also allow ourselves to reverse edge orientations. To compute the complex number associated to a colored surgery link, one must identify a sequence of these operations that simplifies the diagram to the empty diagram. The desired number will then be a product of numbers determined from the operations in the sequence and the given coloring.

Our proof of Theorem 1(b) essentially revolves around two key observations.

First, a kind of uniformity: given a surgery description of M , there exists a *single sequence* of diagrammatic operations that can be used to evaluate *all* colorings of the surgery link to complex numbers. This uniformity is, more-or-less, what will make it possible to encode $M \mapsto \tau_{\mathcal{B}}(M)$ as an instance I_M of $\#\text{CSP}(\mathcal{F})$ for an appropriately chosen \mathcal{F} .

Second: such a uniform sequence of operations can be identified in polynomial time from the surgery description of M . This will imply that the reduction $M \mapsto I_M$ can be performed in polynomial time. We make this point more precise now.

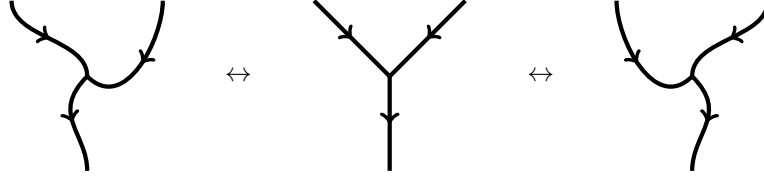
► **Lemma 4.** *If $\Gamma \subset S^2$ is a closed trivalent graph embedded in S^2 , then there is a polynomial time algorithm (in the size of the encoding of Γ) to construct a sequence of embedded graphs $\Gamma_0, \Gamma_1, \dots, \Gamma_l$ where $\Gamma_0 = \Gamma$, $\Gamma_l = \emptyset$ such that each Γ_{i+1} is related to Γ_i by one of the diagrammatic operations in Figures 3-5 or edge orientation reversals.*



■ **Figure 3** A circle removal, tadpole trim, and bubble pop, respectively.



■ **Figure 4** The F -moves.



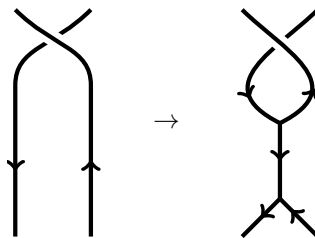
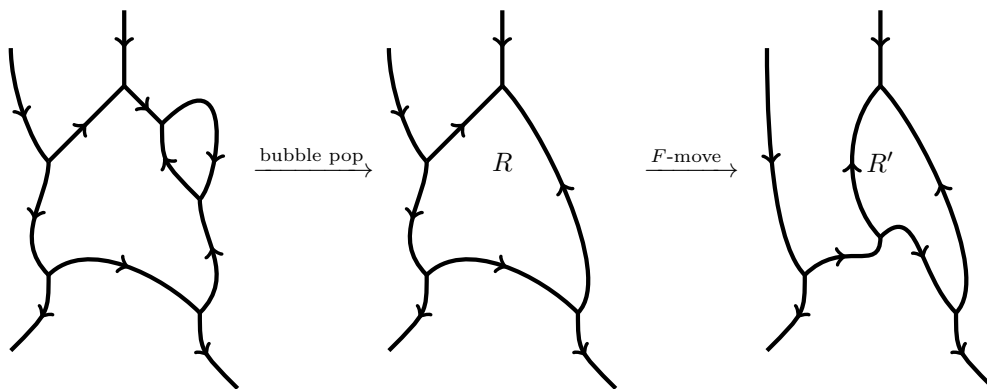
■ **Figure 5** Vertex spiral (aka “bending”).

Proof. A simple greedy algorithm suffices. We sketch the idea.

Begin by greedily choosing a complementary region R of Γ , *i.e.* a connected component R of $S^2 \setminus \Gamma$. Note that we may identify the boundary edges and vertices of R in polynomial time. Suppose R has k unique edges and l unique vertices on its boundary. Now simplify and update Γ according to the following cases.

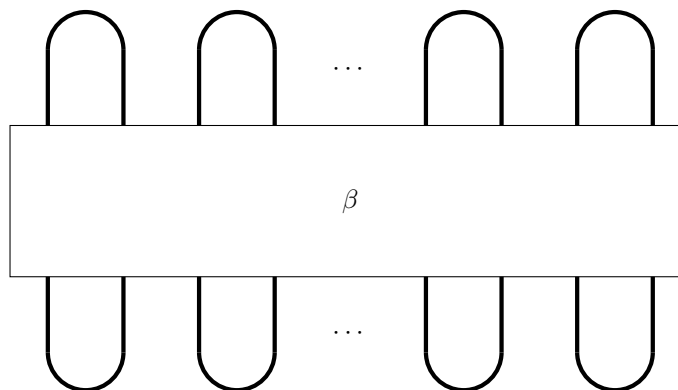
1. If $(k, l) = (0, 0)$, then $\Gamma = \emptyset$, and so we terminate.
2. If $(k, l) = (1, 0)$, then the boundary of R is a circle in Γ , which we remove as in Figure 3.
3. If $(k, l) = (1, 1)$, then the boundary of R is part of a tadpole, which we trim as in Figure 3, but possibly only after first applying an appropriate set of vertex spirals as in Figure 5 and edge orientation reversals.
4. If $(k, l) = (2, 2)$, then the boundary of R is part of a bubble, which we pop as in Figure 3, but possibly only after first applying an appropriate set of vertex spirals and edge orientation reversals.
5. Otherwise, $k = l > 2$. Greedily pick an edge e on the boundary of R . After perhaps first applying up to two vertex spirals and 5 edge orientation reversals, we can arrange so that around e , Γ looks like one of the four diagrams in Figure 4, with e the edge in the middle. Apply the available F -move around e . The complementary regions of the resulting graph are naturally in bijection with the regions of the previous graph (see Figure 6 for an example). Let R' be the region of the new graph associated with R . If R' has $k = l > 2$ edges on its boundary, then repeat what we just did, but with R' and the new graph, instead of R ; otherwise, R' has $k = l = 2$ edges, and we pop the bubble as in case (3).

Repeat this process of greedily picking a region R and proceeding as in the above cases. Each step of identifying an R and carrying through the appropriate case takes polynomial time, and, moreover, reduces the number of complementary regions of Γ by 1. Since there are at most a polynomial number of complementary regions to begin with, the entire procedure takes place in polynomial time. ◀



■ **Figure 7** Inserting trivalent vertices to resolve the identity. Following up with R -moves reduces us to planar diagrams.

Lemma 4 only involves *planar* trivalent graphs, while the proof of part (b) of Theorem 1 needs crossings. Indeed, when we compute Reshetikhin-Turaev invariants, a 3-manifold is encoded by a framed link diagram L (which can be assumed to be in plat position, as in Figure 8). Fortunately, for modular fusion categories we have “ R -moves” that – together with a “resolution of the identity” trick shown in Figure 7 – allow us to reduce to the planar case, and thereby use Lemma 4. The basic strategy is then not so different from the proof of part (a), but it requires that we introduce new variables for every crossing and then every step of the algorithm of the lemma. The details are found in Appendix A.



■ **Figure 8** A blackboard-framed link L in standard plat position is determined by a braid word $\beta \in B_{2k}$.

3 Discussion

3.1 Unitarity

We note that none of our results depend on the unitarity of the spherical fusion category \mathcal{C} or the modular fusion category \mathcal{B} . This is not surprising, since a choice of unitary structure is not necessary to define the TQFT invariants of closed 3-manifolds from \mathcal{C} or \mathcal{B} , and so, as far as exact calculation of invariants is concerned, such a choice will not affect any dichotomies. Nevertheless, a unitary structure is certainly needed in order to do topological quantum computation with the TQFT determined by \mathcal{C} or \mathcal{B} , since, for such applications, one needs the TQFT to be unitary. *A priori*, a specific choice of unitary structure might affect the BQP-universality of braiding (with or without adaptive measurement); however, *a posteriori*, this is not the case because Reutter showed that unitarizable fusion categories admit *unique* unitary structures [18] (we thank Milo Moses for reminding us of this reference).

3.2 TQFTs in other dimensions

We furthermore note that the same strategy we used for the proof of Theorem 1(a) should work more generally to prove that any fully-extended $(d+1)$ -dimensional TQFT in any dimension will satisfy a similar dichotomy involving its invariants of closed $(d+1)$ -dimensional manifolds, so long as the TQFT is defined using a state-sum formula based on finite combinatorial-algebraic data. In particular, similar dichotomies should be possible for $(3+1)$ -dimensional TQFTs based on spherical fusion 2-categories [8] or lattice gauge theories based on finite groups (sometimes called Dijkgraaf-Witten theories) in arbitrary dimension.

3.3 Alternative proof strategies

Building on the previous point, one might try to give an alternative proof of Theorem 1(b) by using the $(3+1)$ -dimensional Crane-Yetter TQFT based on the modular fusion category \mathcal{B} [5]. To put it more carefully, it is known that the Reshetikhin-Turaev invariant $\tau_{\mathcal{B}}(M)$ can be computed by choosing a triangulated 4-manifold Y whose boundary is $\partial Y = M$, and computing an appropriate state-sum invariant of Y (similar to the Turaev-Viro invariant of a triangulated 3-manifold) [4]. So one could try to prove a dichotomy for the $(2+1)$ -dimensional surgery-invariant case of Reshetikhin-Turaev in a simpler way by instead proving a dichotomy for the $(3+1)$ -dimensional triangulation-invariant case of Crane-Yetter. Accomplishing this requires using the fact that given a surgery diagram for a 3-manifold M , one can build a triangulated 4-manifold Y with $\partial Y = M$ in polynomial time (for example, cf. [1]).

In the opposite direction, it is known that for a spherical fusion category \mathcal{C} , $|M|_{\mathcal{C}} = \tau_{\mathcal{Z}(\mathcal{C})}(M)$, where $\mathcal{Z}(\mathcal{C})$ is the Drinfeld center of \mathcal{C} . If one were able to efficiently convert a triangulation of a 3-manifold M into a surgery presentation of the same manifold, then part (b) of Theorem 1 (or Conjecture 2) would immediately imply part (a) of the same.

3.4 Towards Conjecture 2

The current gap between Theorem 1 and Conjecture 2 is explained by a rather simple and undesirable property of our proof of the former: our reductions $M \mapsto I_M$ are not “surjective” from 3-manifold encodings to instances of $\#\text{CSP}(\mathcal{F}_{\mathcal{C}})$ or $\#\text{CSP}(\mathcal{F}_{\mathcal{B}})$. For example, it would be consistent with our results for there to exist a spherical fusion category \mathcal{C} such that $|M|_{\mathcal{C}}$ is computable in polynomial-time, and yet, the problem $\#\text{CSP}(\mathcal{F}_{\mathcal{C}})$ is still $\#\text{P}$ -hard (although we consider this unlikely).

To establish an outright dichotomy theorem for TQFT invariants via Cai and Chen’s Theorem 3, we would need to arrange our choices of \mathcal{F}_C and \mathcal{F}_B with more care so that every instance I of $\#\text{CSP}(\mathcal{F}_C)$ or $\#\text{CSP}(\mathcal{F}_B)$ that is not of the form I_M satisfies two properties: first, it can be identified in polynomial time as an instance that is not of the form I_M , and, second, $Z(I)$ can be computed in polynomial time. This seems difficult to arrange, as it is not clear how to choose the constraint families \mathcal{F}_C and \mathcal{F}_B so that instances can “self-report” as not being of the form I_M .

It is instructive to compare TQFT invariants with “holant problems” as defined in [3] and inspired by the “holographic reductions” of [22]. Holant problems are a kind of generalization of counting CSPs that impose more structure on the way in which the individual functions comprising an instance are “wired together.” Intuitively, an instance of $\#\text{CSP}(\mathcal{F})$ has no locality constraints on its variables, other than that the constraint functions $f \in \mathcal{F}$ have bounded arity (assuming \mathcal{F} is finite). An instance of a holant problem, on the other hand, has a set of variables that are determined by the *edges of a graph* with constraint functions assigned to the *vertices*. The Turaev-Viro-Barrett-Westbury invariant of closed 3-manifolds determined by a spherical fusion category \mathcal{C} can be seen as generalization of this idea, with variables assigned to both the edges *and* faces of a 3-dimensional triangulation, and constraints assigned to the tetrahedra. We expect it should be possible to formulate TVBW invariants of triangulated 3-manifolds directly as instances of holant problems using a similar construction as in the proof of Theorem 1(a). Of course, even if one could achieve this, such a reduction from TVBW invariants to holant problems would – *a priori* – suffer in the same way as our current reduction to $\#\text{CSP}(\mathcal{F}_C)$. Thus, it seems likely that proving Conjecture 2 will require substantially new ideas. Nevertheless, it appears that there could be much to gain by attempting to import what has been learned about holant problem dichotomies to TQFT invariants of 3-manifolds.

References

- 1 Rhuaidi Antonio Burke. Practical Software for Triangulating and Simplifying 4-Manifolds. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG 2024)*, volume 293 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2024.29.
- 2 Jin-Yi Cai and Xi Chen. Complexity of counting CSP with complex weights. *Journal of the ACM (JACM)*, 64(3):1–39, 2017.
- 3 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of holant problems. *SIAM Journal on Computing*, 40(4):1101–1132, 2011.
- 4 Louis Crane and Louis H Kauffman. Evaluating the crane-yetter invariant. *Quantum topology*, 3:131–8, 1993.
- 5 Louis Crane and David N Yetter. A categorical construction of 4d tqfts. *arXiv preprint hep-th/9301062*, 1993.
- 6 Shawn X Cui and Zhenghan Wang. Universal quantum computation with metaplectic anyons. *Journal of Mathematical Physics*, 56(3), 2015.
- 7 Colleen Delaney, Clément Maria, and Eric Samperton. An Algorithm for Tambara-Yamagami Quantum Invariants of 3-Manifolds, Parameterized by the First Betti Number. In Oswin Aichholzer and Haitao Wang, editors, *41st International Symposium on Computational Geometry (SoCG 2025)*, volume 332 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:15, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2025.38.
- 8 Christopher L Douglas and David J Reutter. Fusion 2-categories and a state-sum invariant for 4-manifolds. *arXiv preprint arXiv:1812.11933*, 2018.

- 9 Michael H Freedman, Alexei Kitaev, and Zhenghan Wang. Simulation of topological field theories by quantum computers. *Communications in Mathematical Physics*, 227:587–603, 2002.
- 10 Michael H Freedman, Michael Larsen, and Zhenghan Wang. A modular functor which is universal for quantum computation. *Communications in Mathematical Physics*, 227:605–622, 2002.
- 11 Robion Kirby and Paul Melvin. Local surgery formulas for quantum invariants and the Arf invariant. *Geometry & Topology Monographs*, 7, 2004.
- 12 Alexei Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, 2006.
- 13 Greg Kuperberg. How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015.
- 14 Greg Kuperberg and Eric Samperton. Computational complexity and 3-manifolds and zombies. *Geometry & Topology*, 22(6):3623–3670, 2018.
- 15 Greg Kuperberg and Eric Samperton. Coloring invariants of knots and links are often intractable. *Algebraic & Geometric Topology*, 21(3):1479–1510, 2021.
- 16 Richard E Ladner. On the structure of polynomial time reducibility. *Journal of the ACM (JACM)*, 22(1):155–171, 1975.
- 17 Deepak Naidu and Eric C Rowell. A finiteness property for braided fusion categories. *Algebras and representation theory*, 14:837–855, 2011.
- 18 David Reutter. Uniqueness of unitary structure for unitarizable fusion categories. *Comm. Math. Phys.*, 397(1):37–52, 2023. doi:10.1007/s00220-022-04425-7.
- 19 Eric Rowell and Zhenghan Wang. Mathematics of topological quantum computing. *Bulletin of the American Mathematical Society*, 55(2):183–238, 2018.
- 20 Eric Samperton. Topological quantum computation is hyperbolic. *Communications in Mathematical Physics*, 402(1):79–96, 2023.
- 21 Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978.
- 22 Leslie G Valiant. Holographic algorithms. *SIAM Journal on Computing*, 37(5):1565–1594, 2008.
- 23 Zhenghan Wang. *Topological quantum computation*, volume 112 of *CBMS Regional Conference Series in Mathematics*. Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 2010. doi:10.1090/cbms/112.

A **Proof of Theorem 1(b)**

Proof. Let \mathcal{B} be a modular fusion category over \mathbb{C} . As in our proof of part(a) of Theorem 1, it suffices to build a domain $D_{\mathcal{B}}$ and weighted constraint set $\mathcal{F}_{\mathcal{B}}$ for which there is a polynomial time algorithm to encode a surgery diagram for a 3-manifold M into an instance I_M of $\#CSP(\mathcal{F}_{\mathcal{B}})$ such that

$$Z(I_M) = \tau_{\mathcal{B}}(M)$$

Let us recall the formula for $\tau_{\mathcal{B}}(M)$:

$$\tau_{\mathcal{B}}(M) = p_-^{\frac{\sigma(L)-m-1}{2}} p_+^{\frac{-\sigma(L)-m-1}{2}} \sum_{\text{col}:\{1,\dots,m\} \rightarrow \text{Irr}(\mathcal{B})} \left(\prod_{j=1}^m \dim(\text{col}(j)) \right) |L^{\text{col}}|$$

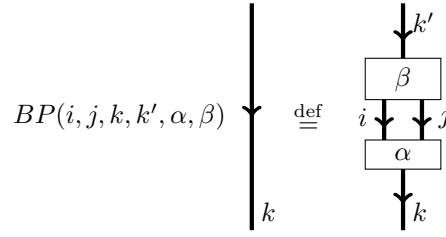
where our notation is as follows:

- L is the surgery link diagram that defines the 3-manifold M . L consists of m components (labeled $1, 2, \dots, m$) and, for convenience is endowed with the blackboard framing.³ We will also assume, for convenience, that L is in “standard plat position.” This means that all of the local minima of the diagram (which, recall, is a picture in the xy -plane) occur below all crossings, all local maxima of the diagram occur above all crossings, and the sets of cups and caps have no “nesting”.⁴ This means L is entirely determined by a braid word. See Figure 8.
- $\sigma(L)$ is the signature of L : the number of positive eigenvalues of the linking matrix minus the number of negative eigenvalues. (This can be computed in polynomial time from L .)
- $p_{\pm} = \sum_{i \in \text{Irr}(\mathcal{B})} \theta_i^{\pm} (\dim(i))^2$ are the Gauss sums of \mathcal{B} .
- $|L^{\text{col}}|$ is the evaluation of the colored ribbon graph defined by coloring the components of L by a function $\text{col} : \{1, \dots, m\} \rightarrow \text{Irr}(\mathcal{B})$.

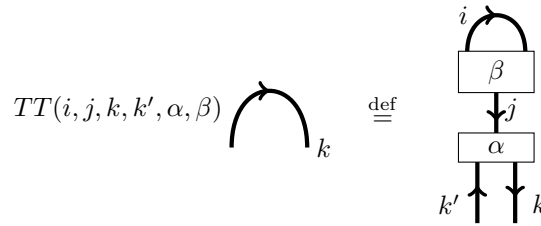
We now define

$$D_{\mathcal{B}} \stackrel{\text{def}}{=} \text{Irr}(\mathcal{B}) \sqcup \mathcal{N} \sqcup \{*\}$$

where \mathcal{N} is the set of labels of the trivalent Hom space as in the proof of Theorem 1(a). As hinted at in Lemma 4, we need \mathbb{C} -valued constraint functions on the domain $D_{\mathcal{B}}$ that implement bubble pops, tadpole removals, F -moves, *etc.* We define them as follows.



where $i, j, k, k' \in \text{Irr}(\mathcal{B})$, $\alpha \in \{1, \dots, N_{ij}^k\}$, and $\beta \in \{1, \dots, N_{ij}^{k'}\}$. These implement the bubble pop.



where $i, j, k, k' \in \text{Irr}(\mathcal{B})$, $\alpha \in \{1, \dots, N_{k'k}^j\}$, $\beta \in \{1, \dots, N_{i*}^j\}$. These implement tadpole trims. There is an upside-down version of the tadpole trim, which we denote \widetilde{TT} .

³ To justify this convenience, simply apply a Reidemeister move of type 1 to each the components of L so that the blackboard framing agrees with the desired integral surgery coefficients. This can be done in polynomial time because we encode the surgery coefficients in unary.

⁴ This convenience can be justified by the fact that any link diagram can be put in standard plat position in polynomial time by simply applying a sequence of Reidemeister 2 moves.

We define functions VS^{left} by requiring

$$\begin{array}{c} k \downarrow \\ \boxed{\alpha} \\ i \downarrow \end{array} \begin{array}{c} j \uparrow \\ \uparrow \end{array} = \sum_{\beta \in \{1, \dots, N_{kj}^i\}} VS^{\text{left}}(i, j, k, \alpha, \beta) \begin{array}{c} k \downarrow \\ \boxed{\beta} \\ i \downarrow \end{array}$$

where $i, j, k \in \text{Irr}(\mathcal{B})$ and $\alpha \in \{1, \dots, N_{ij}^k\}$. These implement the “left” vertex spin. We similarly define VS^{right} coefficients to implement the right vertex spin. There are upside-down versions of these vertex spirals. We denote them by $\widetilde{VS}^{\text{left}}$ and $\widetilde{VS}^{\text{right}}$, respectively.

We need the F -matrices, which have coefficients F^+ that satisfy the equation

$$\begin{array}{c} a \swarrow \\ \boxed{\alpha} \\ n \swarrow \\ \boxed{\beta} \\ d \downarrow \end{array} \begin{array}{c} b \swarrow \\ \swarrow \\ c \end{array} = \sum_{\substack{m \in \text{Irr}(\mathcal{B}) \\ \delta \in \{1, \dots, N_{bc}^m\} \\ \gamma \in \{1, \dots, N_{am}^d\}}} F^+(a, b, c, d, m, n, \alpha, \beta, \delta, \gamma) \begin{array}{c} a \swarrow \\ \swarrow \\ \boxed{\gamma} \\ d \downarrow \end{array} \begin{array}{c} b \swarrow \\ \boxed{\delta} \\ c \swarrow \\ \swarrow \end{array}$$

where $a, b, c, d, m \in \text{Irr}(\mathcal{B})$, $\alpha \in \{1, \dots, N_{ab}^n\}$, and $\beta \in \{1, \dots, N_{mc}^d\}$. The inverse F -matrix has coefficients that we denote by F^- . We also need to include the matrix coefficients implementing the upside-down version of this picture. We call these G^\pm , respectively.

We then extend the above to functions on our domain:

$$F^\pm(x_1, x_2, \dots, x_{10}) \stackrel{\text{def}}{=} \begin{cases} F^\pm(x_1, x_2, \dots, x_{10}) & \text{if } x_1, \dots, x_6 \in \text{Irr}(\mathcal{B}) \text{ and } x_7, \dots, x_{10} \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$G^\pm(x_1, x_2, \dots, x_{10}) \stackrel{\text{def}}{=} \begin{cases} G^\pm(x_1, x_2, \dots, x_{10}) & \text{if } x_1, \dots, x_6 \in \text{Irr}(\mathcal{B}) \text{ and } x_7, \dots, x_{10} \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$BP(x_1, x_2, \dots, x_6) \stackrel{\text{def}}{=} \begin{cases} BP(x_1, x_2, \dots, x_6) & \text{if } x_1, x_2, x_3, x_4 \in \text{Irr}(\mathcal{B}) \text{ and } x_5, x_6 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$TT(x_1, x_2, \dots, x_6) \stackrel{\text{def}}{=} \begin{cases} TT(x_1, x_2, \dots, x_6) & \text{if } x_1, x_2, x_3, x_4 \in \text{Irr}(\mathcal{B}) \text{ and } x_5, x_6 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$\widetilde{TT}(x_1, x_2, \dots, x_6) \stackrel{\text{def}}{=} \begin{cases} \widetilde{TT}(x_1, x_2, \dots, x_6) & \text{if } x_1, x_2, x_3, x_4 \in \text{Irr}(\mathcal{B}) \text{ and } x_5, x_6 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$VS^*(x_1, x_2, x_3, x_4, x_5) \stackrel{\text{def}}{=} \begin{cases} VS^*(x_1, x_2, x_3, x_4, x_5) & \text{if } x_1, x_2, x_3 \in \text{Irr}(\mathcal{B}) \text{ and } x_4, x_5 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

$$\widetilde{VS}^*(x_1, x_2, x_3, x_4, x_5) \stackrel{\text{def}}{=} \begin{cases} \widetilde{VS}^*(x_1, x_2, x_3, x_4, x_5) & \text{if } x_1, x_2, x_3 \in \text{Irr}(\mathcal{B}) \text{ and } x_4, x_5 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

for $*$ \in {left, right}.

We also need to implement braidings, which are described diagrammatically via R -moves. Recall the definition of the R -symbols: for $i, j, k \in \text{Irr}(\mathcal{B})$ and $\alpha \in \{1, \dots, N_{ji}^k\}$, they satisfy

$$\begin{array}{c} i \\ \diagdown \\ \diagup j \\ \alpha \\ \downarrow k \end{array} = \sum_{\beta \in \{1, \dots, N_{ij}^k\}} R^+(i, j, k, \alpha, \beta) \begin{array}{c} i \\ \downarrow \\ \downarrow j \\ \beta \\ \downarrow k \end{array}$$

The inverse R -symbol $R^-(i, j, k, \alpha, \beta)$ is similarly defined to describe the inverse braiding. We turn these R -symbols into 5-ary functions on the entire domain $D_{\mathcal{B}}$ in the same trivial way as before, namely

$$R^{\pm}(x_1, x_2, x_3, x_4, x_5) \stackrel{\text{def}}{=} \begin{cases} R^{\pm}(x_1, x_2, x_3, x_4, x_5) & \text{if } x_1, x_2, x_3 \in \text{Irr}(\mathcal{B}) \text{ and } x_4, x_5 \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases}$$

Finally, we define 1-ary dimension functions, 1-ary Gauss sum functions (and their inverses), 1-ary dual functions, and 2-ary Kronecker delta functions as follows (respectively):

$$\begin{aligned} d(x) &\stackrel{\text{def}}{=} \begin{cases} \dim(x) & \text{if } x \in \text{Irr}(\mathcal{B}) \\ 0 & \text{otherwise} \end{cases} \\ p_{\pm}^{1/2}(x) &\stackrel{\text{def}}{=} \begin{cases} \left(\sum_{j \in \text{Irr}(\mathcal{B})} \theta_j^{\pm 1} \dim(j)^2 \right)^{1/2} & \text{if } x = * \\ 0 & \text{otherwise} \end{cases} \\ p_{\pm}^{-1/2}(x) &\stackrel{\text{def}}{=} \begin{cases} (p_{\pm}(x))^{-1} & \text{if } x = * \\ 0 & \text{otherwise} \end{cases} \\ \delta(x_1, x_2) &\stackrel{\text{def}}{=} \begin{cases} \delta_{x_1, x_2} & \text{if } x_1, x_2 \in \text{Irr}(\mathcal{B}) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

With all of this, we define our weighted constraint family $\mathcal{F}_{\mathcal{B}}$ to be

$$\mathcal{F}_{\mathcal{B}} \stackrel{\text{def}}{=} \{F^{\pm}, G^{\pm}, BP, TT, \widetilde{TT}, VS^{\text{left}}, VS^{\text{right}}, \widetilde{VS}^{\text{left}}, \widetilde{VS}^{\text{right}}, R^{\pm}, d, p_{\pm}^{1/2}, p_{\pm}^{-1/2}, \delta\}.$$

We reiterate that all of this is computed independently of M , and can simply be considered as part of what it means to “have the data” of \mathcal{B} .

We conclude our proof by describing how to encode a surgery presentation of M into an instance I_M of $\#\text{CSP}(\mathcal{F}_{\mathcal{B}})$. In addition to the conveniences described above, we assume that the surgery link diagram L is oriented, embedded in $\mathbb{R} \times [-1, 2]$, and is given by plat closure of a braid word $b_1 b_2 \cdots b_n$ so that each crossing corresponding to b_i lies in $\mathbb{R} \times (\frac{i-1}{n}, \frac{i}{n})$ and the only maxima or minima lie in $\mathbb{R} \times ([-1, 0) \cup (1, 2])$.

In order to describe the variables that will be involved in the instance I_M we want, we first describe a polynomial time algorithm to replace L with a planar trivalent graph $\Gamma_L \subset S^2$:

1. At each crossing b_i , insert trivalent vertices to resolve the identity (see Figure 7) so that there is a vertex directly adjacent to the crossing, resulting in a crossed trivalent graph.
2. Perform an R -move for each crossing, resulting in a trivalent graph in $\mathbb{R}^2 \subset S^2$ (after potentially scaling so the vertices lie in $\mathbb{R} \times \mathbb{Z}$).

We can now identify the tuple of variables (valued in the domain $D_{\mathcal{B}}$) that will be involved in our desired instance I_M . Recall that m is the number of components of L , $\sigma(L)$ is the signature, and n is the number of crossings. Let $\Gamma_0 = \Gamma_L, \Gamma_1, \dots, \Gamma_l = \emptyset$ be the sequence of

graphs provided by Lemma 4, and define N_v to be the sum of the number of vertices in all of these graphs. Similarly, define N_e to be the sum of the number of edges in the graphs. Then define a tuple of variables associated to the instance I_M by

$$\mathbf{x}_M \stackrel{\text{def}}{=} (x_0, x_1, \dots, x_m, y_1, \dots, y_{|\sigma(L)|}, z_1, \dots, z_n, u_1, \dots, u_{N_v}, w_1, \dots, w_{N_e}).$$

To determine the functions involved in I_M , we simply need to keep appropriate account of the sequence of diagrammatic operations involved in taking L to Γ_L and then taking Γ_L to \emptyset . For each diagrammatic operation in the algorithms above, we will define a list of functions in $\mathcal{F}_{\mathcal{B}}$ which account for the contribution of those operations to $\tau_{\mathcal{B}}(M)$.

The first operations are those involved in step (1) of the process of taking L to Γ_L , and involve the insertion of “resolutions of the identity” at every crossing of L . To account for these operations, we define a set Init that is a list of Kronecker delta functions, each of which pairs up edges which used to belong to the same link component before the operation. For example, in Figure 7, the top-right edge of the upper vertex and the bottom-right edge of the lower vertex were a part of the same link component before the operation, so we introduce a Kronecker delta function between the associated variables. The middle edge is free to vary, as in the original algorithm there is a sum over this edge.

We then account for the operations in step (2) of the process of taking L to Γ_L , all of which are R -moves. Each operation happens locally on the diagram, so we define lists R_i for $1 \leq i \leq n$ that are given by $R^\pm(w_{i_1}, w_{i_2}, w_{i_3}, u_{i_4}, u_{i_5})$, where we use $+$ or $-$ depending on the strand that crosses over. The variables $w_{i_1}, w_{i_2}, w_{i_3}$ are the edges in the trivalent vertex in the relevant order, u_{i_4} is the labeling of the vertex before the R -move, and u_{i_5} is the labeling of the vertex after the operation.

The next operations are given using the algorithm of Lemma 4. The algorithm provides an ordered list of operations O_1, \dots, O_l , where upon completion of the final operation O_l , the graph Γ_l is empty. Each operation here is local, so we need only consider the local changes when defining our list. Consider operation O_i in this sequence:

- If O_i is a circle removal, define a list C_i which contains the relevant d function.
- If O_i is a tadpole trim, define a list T_i which contains the relevant TT or \widehat{TT} function.
- If O_i is a bubble pop, we define a list B_i which contains the relevant BP function.
- If O_i is a vertex spiral, we define a list V_i which contains the relevant VS^{left} or VS^{right} (or their upside-down versions).
- If O_i is an F -move, we define a list F_i which contains the relevant F^\pm or G^\pm functions.
- if O_i is an orientation reversal where w_{i_1} is the variable associated to the edge before the operation, and w_{i_2} is the variable associated to the edge after the operation, then define a list O_i which contains the Kronecker delta function $\delta(w_{i_1}^*, w_{i_2})$.

For each list, we append Kronecker delta functions δ on all edges or vertices which are held constant before and after performing the given operation. E.g. if the edge associated to w_{22} will be held constant after an F -move, and will then be associated with w_{100} in the trivalent graph associated to after the operation, then we introduce $\delta(w_{22}, w_{100})$ to the list F associated to the operation. If $\sigma(L) \geq 0$, we then define the set I_M by:

$$\begin{aligned} I_M = \{ & p_+^{-1/2}(x_0), p_-^{-1/2}(x_0), p_+^{-1/2}(x_1), \dots, p_+^{-1/2}(x_m), p_-^{-1/2}(x_1), \dots, p_-^{-1/2}(x_m), \\ & p_-^{1/2}(y_1), \dots, p_-^{1/2}(y_{|\sigma(L)|}), p_+^{-1/2}(y_1), \dots, p_+^{-1/2}(y_{|\sigma(L)|}), d(x_1), \dots, d(x_m) \} \\ & \cup \text{Init} \cup R_1 \cup \dots \cup R_n \cup \bigcup_{p_1, \dots, p_6} C_{p_1} \cup T_{p_2} \cup B_{p_3} \cup V_{p_4} \cup F_{p_5} \cup O_{p_6} \end{aligned}$$

where p_1, p_2, \dots, p_6 each run over all $1, \dots, l$ which were defined by the algorithm above. If $\sigma(L) < 0$, then replace the elements $p_-^{1/2}(y_1), \dots, p_-^{1/2}(y_{|\sigma(L)|})$ with the inverses $p_-^{-1/2}(y_1), \dots, p_-^{-1/2}(y_{|\sigma(L)|})$. By virtue of Lemma 4, the construction of I_M can be carried out in polynomial time in the size of M .

We now explain why this choice of instance I_M defines an output which computes the Reshetikhin-Turaev invariant $\tau_B(M)$. The $p_+^{\pm 1/2}$ and $p_-^{\pm 1/2}$ functions at the beginning implement the normalizing factor since these functions are constant. This guarantees they may be factored out of the sum as global factors. The d functions implement the product of dimensions we see in the sum.

Notice that if at any point a coloring is not admissible, the term in I_M is 0. It is thus clear that all circle removals, edge orientation reversals, bubble pops, and tadpole trims are correctly implemented, so we just need to check that the F -moves, R -moves, and vertex spirals are correct.

In the standard algorithm, when an F -move occurs, there are three additional variables introduced in the summation: one for the interior edge and two for each interior vertex. These additional variables are introduced here as well, since we are summing over all edges that occur throughout the algorithm, the new edge which is created in the F -move will contribute to the sum, while the rest are held constant due to the inclusion of the Kronecker delta functions.

Similarly, we see that R -moves and vertex spirals are correctly implemented. Note that also there are no extraneous variables in the sum since the algorithm guarantees that the graph will become empty, and our instance is completely determined by how the algorithm behaves. ◀