Computing Surface Cobordism Maps in Khovanov Homology

Mentor: Mary Stelow

Mohamed Wacyl Meddour and Veronika Moroz

Summer 2025

1 Introduction and Background

1.1 Khovanov Homology

Originally proposed by Mikhail Khovanov [14] in the early 2000s, Khovanov homology assigns a chain complex to a projection of an oriented link L so that the chain complex's homology is an invariant up to homotopy equivalence.

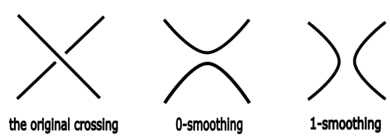
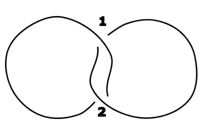
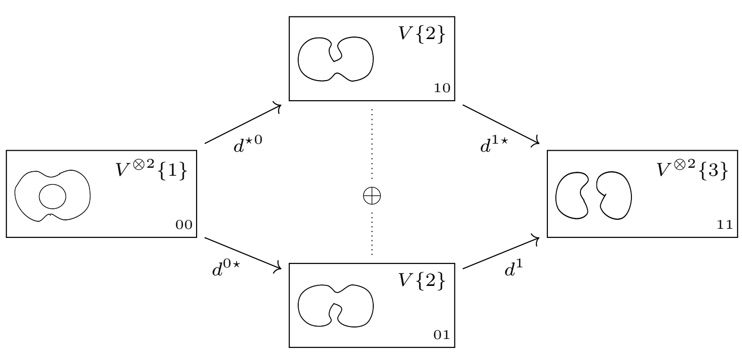


Figure 1: We can resolve each crossing with a 0-smoothing (middle) or a 1-smoothing (right).

To build this chain complex, we resolve each crossing of L using either a *0-smoothing* or a *1-smoothing*, as depicted in Figure 1. Then, we form a hypercube of resolutions, where each edge replaces one 0-smoothing with a 1-smoothing.

An example of such a hypercube for the Hopf link is pictured below, with the Hopf link itself in the top-left corner.





$$C^0 \xrightarrow[d^0 = \sum_{|\xi|=0} (-1)^{\xi} d^{\xi}]{} C^1 \xrightarrow[d^1 = \sum_{|\xi|=1} (-1)^{\xi} d^{\xi}]{} C^2 \tag{1}$$

If we number the crossings on L, we can summarize how each crossing is smoothed through a binary string called a *state*, where the the kth digit is a 0 or a 1, depending on if L has a 0-resolution or a 1-resolution at the kth crossing. To build the ith cochain group C^i, we will assign a vector space to each smoothing and take the direct sum of the smoothings at the vertices k edges away from the vertex of all 0-smoothings. We label each loop in a smoothing with a v_+ or a v_-, where v_+ and v_- are the generators of a graded vector space V. The grading of v_+ is 1 and the grading of v_- is -1. Thus, the chain group C^n is a direct sum of graded vector spaces of the form $V^{\otimes k}$, where k is the number of loops in a given smoothing.

Definition 1.1. For a graded vector space W with homogenous components W_m, the graded dimension of W is the power series $q \dim W := \sum_m q^m \dim W_m$.

Definition 1.2. The shift $\{l\}$ of a graded vector space W has graded components $W\{l\}_m = W_{m-l}$, so $q \dim W\{l\} = q^l W$.

Definition 1.3. The height shift $[s]$ of a chain complex C is the chain complex $C[s]$ with rth chain group $C[s]^r = C^{r-s}$.

Using these definitions, we shift our indices of the chain complex using the operations $[-n_-]\{n_+ - 2n_-\}$, where n_- is the number of negative crossings and n_+ is the number of positive crossings. We will later refer to the shift in homological and quantum degree as a normalization.

Now that we have created the vector spaces, the next step is to create the differentials. We define $d^n : C^n \to C^{n+1}$ to be the map from a labeled smoothing σ in C^n to the direct sum of each picture made by replacing one of σ's 0-smoothings with a 1-smoothing. We denote this as $d^n = \sum (-1)^\alpha d^{\cdots * \cdots}$, where the * represents the digit being changed from 0 to 1. To make (C^n, d^n) a chain complex, we need to make every square in the cube of resolutions anticommute. So, we insert signs by multiplying each component of the differential by $(-1)^\alpha$, where α is the number of 1s that go before * in the superscript of the differential.

Each edge on the hypercube will either *merge* two loops or *split* one loop by changing a 0-smoothing to a 1-smoothing. For algebraic reasons, the merge map, m, and the split map, Δ, are defined on v_+ and v_- as follows:

$$m(v_+ \otimes v_+) = v_+; \quad m(v_+ \otimes v_-) = m(v_- \otimes v_+) = v_-; \quad m(v_- \otimes v_-) = 0,$$
$$\Delta(v_+) = v_+ \otimes v_- + v_- \otimes v_+, \quad \Delta(v_-) = v_- \otimes v_-.$$

The resulting cohomology on these chain complexes is called the *unreduced Khovanov Homology.* One can check that the cohomology of these chain complexes is invariant under Reidemeister moves. The information about the Khovanov homology with field coefficients of a link can be represented by the *Khovanov polynomial*, where each term is of the form ct^aq^b. Here, a is the homological degree, b is the quantum degree (qdegree), and c is the dimension of the associated homology. We denote the chain complex associated to a link projection of L by $\mathrm{CKh}(L)$. The (invariant) homology of that complex is denoted by $\mathrm{Kh}(L)$.

1.2 Khovanov Surface Maps

One advantage of Khovanov homology over the Jones polynomial is that it has an inherent functorial property induced by cobordisms. Given two links L_0 and L_1 embedded in $\mathbb{S}^3$, and given any orientable surface $\Sigma \subseteq \mathbb{B}^4$ for which $\partial\Sigma = -L_0 \sqcup L_1$, there is an induced map:

$$\mathrm{Kh}(\Sigma) : \mathrm{Kh}(L_0) \to \mathrm{Kh}(L_1)$$

from the Khovanov homology of one link to the other. These maps are compatible with composition of two surface cobordisms. In other words, given links L_0, L_1, L_2 and surfaces Σ_0, Σ_1 with $\partial\Sigma_0 = -L_0 \sqcup L_1$ and $\partial\Sigma_1 = -L_1 \sqcup L_2$, the following diagram commutes:

$$\begin{array}{ccc} \mathrm{Kh}(L_0) & \xrightarrow{\mathrm{Kh}(\Sigma_0 \cup \Sigma_1)} & \mathrm{Kh}(L_2) \\ & {\scriptstyle \mathrm{Kh}(\Sigma_0)} \searrow \quad \nearrow {\scriptstyle \mathrm{Kh}(\Sigma_1)} & \\ & \mathrm{Kh}(L_1) & \end{array}$$

More formally and in the context we care about in this paper, this map corresponds to the composition of the functor in Theorem 4 and the TQFT in Proposition 7.2 in [4]. In this section, we aim to detail how these maps are defined.

By Morse theory [12], any surface cobordism Σ can be decomposed into a *movie presentation*, i.e. a sequence of moves that one can perform successively starting from L_0 and modifying the link until arriving to L_1.

Definition 1.4. Let L_0, L_1 be two links in the 3-sphere and let Σ be an orientable surface in the 4-ball bounding L_0 and L_1. The *movie presentation* of Σ is a sequence of moves of the following type that describe our surface cobordism:

- A Reidemeister move
- A saddle move: merging or splitting two circles
- A birth or death move: an index zero or two critical point which creates or destroys a copy of the unknot, respectively

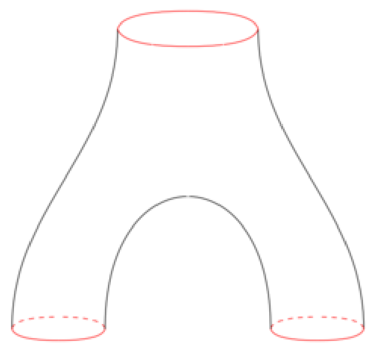


Figure 2: The movie presentation of a saddle move that splits an unknot into two unknots

All such surfaces Σ can be decomposed as above [3]. We make use of the movie presentation of a surface to describe how the cobordism maps behave. We already have the necessary maps for all the moves except birth and death.

Definition 1.5. Given a link L, let $L' = L \sqcup \text{unknot}$. We define the *birth map* to be the chain map $\epsilon : \mathrm{CKh}(L) \to \mathrm{CKh}(L')$ defined by labeling the additional circle in any resolution with a v_+, written locally as $\epsilon : 1 \mapsto v_+$. We define the *death map* as the chain map $\eta : \mathrm{CKh}(L') \to \mathrm{CKh}(L)$ defined by killing any resolution with a v_+ in the additional circle and removing the circle if it is labeled as a v_-, locally given by

$$\eta : \begin{cases} v_- \mapsto 1 \\ v_+ \mapsto 0 \end{cases} .$$

We are now ready define the map induced from a cobordism of links.

Definition 1.6. Let Σ be a surface bounding two links L_0 and L_1, and assume that Σ is described as a movie presentation of links:

$$L_0 = D_0 \to D_1 \to D_2 \ldots \to D_n = L_1.$$

Then, we define the map $\mathrm{CKh}(\Sigma) : \mathrm{CKh}(L_0) \to \mathrm{CKh}(L_1)$ by

$$\mathrm{CKh}(\Sigma) = f_0 \circ f_1 \circ \ldots f_{n-1}$$

where each $f_i : \mathrm{CKh}(D_i) \to \mathrm{CKh}(D_{i+1})$ is defined by

$$f_i = \begin{cases} m \text{ or } \Delta & \text{if } D_i \to D_{i+1} \text{ is a merge or split move} \\ \epsilon & \text{if } D_i \to D_{i+1} \text{ is a birth move} \\ \eta & \text{if } D_i \to D_{i+1} \text{ is a death move} \\ R_j & \text{if } D_i \to D_{i+1} \text{ is the } j\text{'th Reidemeister move} \end{cases} .$$

Each R_i is described by the canonical homotopy equivalence induced from invariance of Khovanov homology under Reidemeister moves, and is formally described in 7.2. The map on homology induced by the chain maps above is the *surface cobordism map* $\mathrm{Kh}(\Sigma)$.

This procedure fully defines the chain maps induced by cobordisms. In practice, we will draw a movie presentation of a cobordism and compute each map step by step to deduce the image of an element through a cobordism movie map. Such maps lie at the heart of this report, in which we aim to verify attainability of certain elements via cobordism maps. A program based on SageMath is provided in 7.3 that implements movie cobordisms in Python.

Note that Khovanov surface maps have a well-defined grading: if Σ is a link cobordism, then $\mathrm{Kh}(\Sigma)$ raises quantum degree by $\chi(\Sigma)$. If we consider surfaces up to smooth isotopy rel boundary, then $\mathrm{Kh}(\Sigma)$ is well-defined up to sign. This is because smooth isotopies rel boundary in the 4-ball can be decomposed into a series of Carter-Saito moves that preserve $\mathrm{Kh}(\Sigma)$ up to sign [12].

Proposition 1.7. *Let Σ be a smooth surface in $I \times \mathbb{S}^3$ bounding two links L_0 and L_1. Then, $Kh(\Sigma)$ is a well-defined map between $Kh(L_0)$ and $Kh(L_1)$ with quantum degree $\chi(\Sigma)$. Furthermore, if Σ' is another surface smoothly isotopic to Σ rel boundary, then $Kh(\Sigma') = \pm Kh(\Sigma)$.*

1.3 Motivation

One of the most famous outstanding questions in low dimensional topology is the smooth Poincaré conjecture, which asks if every manifold which is homeomorphic to $\mathbb{S}^4$ is also diffeomorphic to $\mathbb{S}^4$.

Definition 1.8. We call two manifolds an *exotic pair* if they are homeomorphic but not diffeomorphic.

Because of the groundbreaking work of Donaldson [6] in the 1980s, we know that there are many other exotic four-manifolds, but it is still an open problem to develop methods to distinguish smooth manifolds with the simplest topology.

There are many variants of exotic behavior one can study. For instance, in the relative case, we can ask for a pair of *exotic surfaces* Σ, Σ' inside a four manifold X – by these we mean surfaces Σ, Σ' inside X that are topologically isotopic relative to their boundary, but not smoothly so.

Khovanov homology is an invariant of the smooth isotopy class of a surface in $I \times S^3$ relative to its boundary, so one approach to finding exotic surfaces in $I \times S^3$ or B^4 is to exhibit topologically isotopic surfaces inducing different maps on Khovanov homology. This approach has been deployed successfully [10]; [11].

Definition 1.9. The *relative Khovanov-Jacobsson class* for a knot L with movie diagram D is the equivalence class of the image of 1 under the cobordism $\Sigma : \emptyset \to L$ [18].

The motivating question of this research project is

Question 1.10. *When can elements of Khovanov homology be realized as Khovanov-Jacobsson classes?*

As part of this project, we wrote a program to calculate Khovanov-Jacobsson classes for all knots, and ultimately proved what form the Khovanov-Jacobsson class would take for the Seifert surface cobordism for any knot. We also examined special cases such as negative links, positive links, and twist knots.

From Bar-Natan's categorical point of view, Khovanov homology arises from a functor $\mathcal{F}$ from an additive tangle category to the category of $\mathbb{Z}$-modules [4]. Since the Khovanov chain complex of an empty link is trivial, morphisms from $\mathrm{Kh}(\emptyset)$ are determined by the image of 1, making our question equivalent to finding the morphisms $\mathrm{Kh}(\emptyset) \to \mathrm{Kh}(L)$ induced from morphisms in $\mathrm{Mor}(\emptyset, L)$ by $\mathcal{F}$.

2 Results

2.1 Negative and positive links

We start by discussing the behavior of Khovanov homology and Seifert surface cobordisms for negative and positive links, meaning links whose crossings are all positive or all negative. These links are easy to understand because the zeroth chain group Kh^0 is the direct sum of exactly one vector space, since there is only one vertex of the cube of resolutions at the appropriate height. The resolution corresponding to this vector space is the oriented resolution.

Definition 2.1. Let L be an oriented link. L is said to be *positive* (resp. *negative*) if there exists a projection of L for which all crossings are positive (resp. negative). Recall that the sign of a crossing is positive if one needs to rotate the understrand clockwise to match the overstrand, and negative otherwise.

We start by understanding the oriented resolution of any oriented link.

Definition 2.2. Take a smoothing and draw a dashed line between the two loops formed by the smoothing. This dashed line is called a *tracing*, and is pictured in Figure 3.

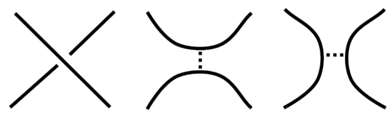


Figure 3: The 0-tracing (middle) and 1-tracing (right) of a crossing

Borrowing terminology from [7] we say that a tracing is *0-merging* if it is on a 0-smoothing between two loops, and changing the 0-smoothing to a 1-smoothing will merge those loops. Similarly, a tracing is *1-merging* if its drawn on a 1-smoothing and changing the 1-smoothing to a 0-smoothing merges the loops together.

Proposition 2.3. *Let L be an oriented link. Then, the oriented resolution of L as a diagram lies in* $CKh(L)^0$*. Furthermore, every tracing left by the resolutions is merging: that is, all of the 0-smoothings are 0-merging and all of the 1-smoothings are 1-merging.*

Proof. Let n_+ be the number of positive crossings and $n_- = n - n_+$ be the number of negative crossings of L. Based on the normalization from the $-n_-$ homological degree shift, for a link L's resolution to lie in $\mathrm{CKh}(L)^0$, it needs to have n_+ 0-resolutions and n_- 1-resolutions. Notice, however, that in the oriented resolution, every negative crossing must be 1-resolved and every positive crossing must be 0-resolved, so this follows. The second part follows because the resolution preserves orientation. □

We now turn to positive and negative links. With proper labeling, the oriented resolution gives a generating element for the Khovanov homology at that specific homological and quantum degree.

Proposition 2.4. *Let L be a negative (resp. positive) link. Let σ be the enhanced Kauffman state one gets by labeling the oriented resolution of L with only v_+s (resp. only v_-s). If q is the quantum degree of σ, then σ generates* $Kh(L)^{(0,q)}$.

Proof. We prove this for negative links first. Clearly, since the link is negative, all resolutions are 1-resolutions, and hence the diagram is of maximal degree, so $\dim \mathrm{Kh}(L)^{0,q} \leq 1$. We show it is exactly 1 by showing σ is nonzero in homology. To see this, it is enough to show σ is not in the image. From Proposition 2.3, all traces of σ are merging, so if it was in the image, it arises from a linear combination of split maps. However, any circle split induces at least one minus label, so degree reasons imply σ cannot be in the image, as desired.

A similar argument follows for positive links: all resolutions are 0-resolutions, σ lies in the kernel because merging any two minus signs gives 0, and $\dim \mathrm{Kh}(L)^{0,q} \leq 1$ because σ has the smallest possible degree. □

We relate the topics of negative and positive links to movie cobordisms by discussing Seifert surfaces and what we will describe as the Seifert surface cobordism.

Definition 2.5. Let L be a link embedded in $\mathbb{S}^3$. Any surface $\Sigma \subset S^3$ for which $\partial\Sigma = L$ is called a *Seifert surface* for the link L. The map on Khovanov homology for a Seifert surface is well-defined and independent of which order the births, twists, and merges are completed (see 7.1).

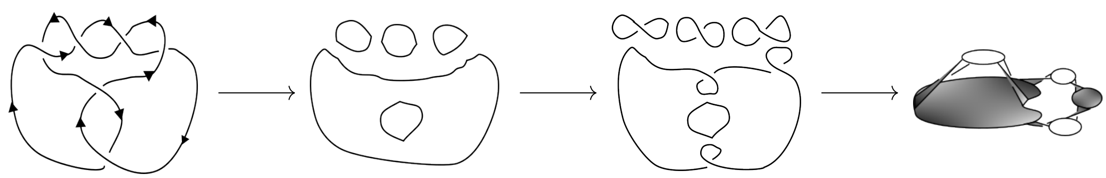
Seifert surfaces always exist and are not unique. Given a diagram of a link L, Seifert's algorithm describes how to produce an oriented surface bounding it [3]:

1. Draw the oriented resolution of our link L.
2. Create a disk in space for each circle in the oriented resolution of L, with adjacent disks at the same height, and with disks that were initially embedded in one another at different heights. These disks are called *Seifert circles*.
3. Add a twisted band connecting each pair of disks that are connected by crossings.

The validity of this procedure follows again from the fact that the oriented resolutions's tracings are all merging. Furthermore, one can check that the surface we get is orientable. We can immediately describe a movie presentation of the surface found by Seifert's algorithm, composed of the following steps:

1. Birth all Seifert circles.
2. For each pair of strands separated by the trace of a crossing, create a twist in one of the two strands matching the crossing's orientation.
3. For each twist created between two strands separated by a tracing, join the loop of the twist with the other strand to create the crossing.

Below is an example of Seifert's algorithm, applied to the 6_1 knot. The surface is orientable, so we color one side gray and the other white.



(2)

Since $S(L)$ is orientable, it induces a movie cobordism $\mathrm{Kh}(S(L)) : \mathrm{Kh}(\emptyset) \to \mathrm{Kh}(L)$ that is well-defined up to sign. In fact, one can check that despite the implicit choices in the above construction (such as when to birth/twist/saddle and which strand to saddle), all the surfaces we get are isotopic rel boundary, and hence the map $\mathrm{Kh}(S(L))$ is well-defined up to sign. This is what we will refer to as the *Seifert surface cobordism map*.

A simpler question than determining which elements of Khovanov homology of a link are Khovanov-Jacobsson classes is to instead ask about the image of 1 under a *Seifert* surface cobordism map, specifically, one built via Seifert's algorithm, for a link L. We can fully answer this question for negative links, as well as its dual for positive links.

Proposition 2.6. *Let L be a negative link and let $S(L)$ be the Seifert surface we get from Seifert's algorithm. Then, $Kh(S(L))$ maps $1 \in Kh(\emptyset)^{0,0}$ to $\pm\sigma$, where σ is the state defined in Proposition 2.4. Dually, if L is a positive link, then the image of the state σ through $Kh(S(L))^* : Kh(L) \to Kh(\emptyset)$ is 1.*

Proof. Because 1 lies in homological degree 0, it is forcibly mapped to an element in homological degree 0 in $\mathrm{Kh}(L)$. The only diagram in homological degree 0 is the oriented resolution, so we just have to check that all labels are positive.

All births will create circles with positive labels. Since the link L is negative, all twists will be left-handed (i.e. their 0-resolution will not have an additional circle). As such, the chain map will simply birth a circle corresponding to the loop with a positive label. Finally, all saddles will be done with positive labeled loops, as desired.

For the dual statement, notice that all saddles will consist of Δ maps (meaning they connect the same circle) and hence induce minus signs for the loop circles. Notice furthermore that all twists are right-handed (their 0-resolution has an additional circle), so that the chain map deletes that circle by applying η. In the end, we are left with one negative circle that gets killed to get 1, as desired. □

We now aim to answer the following question: for an arbitrary link L, what is the image of 1 under the Seifert surface cobordism $\mathrm{Kh}(S(L))$? We already know the answer if L is negative. In fact, it is enough to answer this question for positive knots to get a general answer. This is because the addition of negative (left handed) twists acts as an identity and can be ignored.

We first start with an easy claim describing a necessary condition for when the image is nonzero. This proposition will generalize soon enough.

Proposition 2.7. *Let L be a positive link. If two Seifert circles of L are connected by 3 or more crossings in L, then $Kh(S(L))$ maps 1 to 0.*

Proof. Since the chain map induced by a positive twist sends v_+ to $v_+ \otimes v_- - v_- \otimes v_+$ and sends v_- to $v_- \otimes v_-$, we compute:

$$\begin{aligned} v_+ \otimes v_+ &\mapsto (v_+ \otimes v_- - v_- \otimes v_+) \otimes v_+ \\ &\mapsto v_+ \otimes v_- - v_- \otimes v_+ \\ &\mapsto (v_+ \otimes v_- - v_- \otimes v_+) \otimes v_- - v_- \otimes v_- \otimes v_+ \\ &\mapsto -2(v_- \otimes v_-) \\ &\mapsto -2(v_- \otimes v_- \otimes v_-) \\ &\mapsto 0. \end{aligned}$$

The maps are (in order): twist, join, twist, join, twist, join, and the twists are positive. □

Definition 2.8. Let L be a positive link. Define $G(L)$ as the (multi)graph whose vertices are Seifert circles of L with an edge between Seifert circles for every crossing between them in L. Proposition 2.7 ensures that there are at most 2 edges between any two fixed vertices in $G(L)$.

Figure 4 shows an example of $G(L)$ for the positive 7_4 knot.

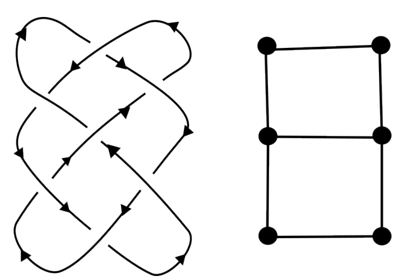


Figure 4: The 7_4 knot (left) and $G(7_4)$ (right)

We remark that $G(L)$ is 2-colorable: since $S(L)$ is oriented, the only saddle moves allowed are orientation-preserving saddle moves. This forces the orientation of Seifert circles to switch from clockwise to counter-clockwise at each connection, which serves as a 2-coloring for $G(L)$. Proposition 2.7 nicely generalizes in the terminology of Definition 2.8.

Proposition 2.9. *Let L be a positive link and let S be a subgraph of $G(L)$. If there are strictly more edges (counted with multiplicity) in S than vertices in S, then $Kh(S(L))$ maps 1 to 0. In other words, if a certain subset of k Seifert circles is joined by at least $k+1$ bands, then the Seifert surface cobordism is trivial.*

Proof. Let L' be the link one gets by considering the result of the Seifert surface cobordism map restricted to the subset S of Seifert circles. We will show $\mathrm{Kh}(S(L'))$ maps 1 to 0, which is enough by well-definiteness of the Seifert surface cobordism map (see 7.1).

Let $s = |S|$ be the number of Seifert circles and $c = |E(S)|$ be the number of bands connecting them. It is well-known that the Euler characteristic of the Seifert surface cobordism map on L' is given by $s - c$. However, the minimal degree element in the diagram attainable from the Seifert surface cobordism map of L' (the oriented resolution on L') has degree $-s + c$ ($-s$ from labeling all circles with a v_-, c from the normalization term since we have c positive crossings). But $-s + c > s - c$, so the result follows for degree reasons. □

One remark for the reader: if S is connected, it is enough to check this condition at the level of the connected component of $G(L)$ containing S. This is because expanding S by adjoining Seifert circles connected to S can only decrease $s - c$.

Finally, to fully answer the question for a positive link L, it is enough to answer it for distinct connected components of $G(L)$. So, from now on, we assume that $G(L)$ is connected. In this case, we call L a *positive connected link*. This is equivalent to L being a nonsplittable oriented positive link.

We are now ready to answer the question for positive connected links. We remark that for a positive connected link L with n Seifert circles, $G(L)$ must have at least $n-1$ edges.

The following theorem has already been proven in [18]. However, we conjectured and proved it independently before coming across [18], so we have included it in this report.

Theorem 2.10. *Let L be a positive connected link, and assume $G(L)$ has n vertices and m edges. Let σ be the (unlabeled) oriented resolution of L. Then (up to sign):*

1. *If $m > n$, $Kh(S(L))$ is trivial and maps 1 to 0.*

2. *If $m = n$, then $Kh(S(L))$ maps 1 to $2 \cdot \sigma_-$ where σ_- is the enhanced state one gets by labeling all Seifert circles with v_-s in the oriented resolution σ.*

3. *If $m = n - 1$, then $Kh(S(L))$ maps 1 to $\sum_{i=1}^{n} \epsilon_i \cdot \sigma_i$. Here, σ_i is the state one gets by labeling the ith circle of σ as a v_+ and all other circles as a v_-, and $\epsilon_i = (-1)^{d(c_1, c_i)}$, where c_1 is a fixed circle in $G(L)$ and c_i is the ith circle.*

Proof. We work case by case:

1. Follows from the previous proposition verbatim.

2. If $m = n$, then there must exist a cycle somewhere in $G(L)$ (we count a double edge as a 2-cycle). 2-colorability implies the cycle must be of even length, say $c_1, \ldots, c_{2k}$. Computing the cobordism by first birthing these circles then twisting the bands forming the cycle and saddling them shows that all $c_1, \ldots, c_{2k}$ must be labeled v_- and a factor of 2 appears. For any other circle linked with a c_i, it automatically picks up a v_- label, so we are done.

3. If $m = n-1$, then $G(L)$ is a tree and no double edge can appear. If t is the number of v_+ labels minus the number of v_- labels, since the map is graded by the Euler characteristic of the Seifert surface, one needs that $t + n - 1 = 1 \implies t = 2 - n$. But, positive and negative circles must add up to n, so adding the two equations gives that the number of positive labels is exactly 1, so the image is a linear combination of the σ_i states. Fixing a state c_1 and resolving from there, one sees by induction that the image is given, up to sign, by the 2-coloring of $G(L)$, as desired.

□

Remark 2.11. For positive connected links L with as many bands as Seifert circles, $\mathrm{Kh}(S(L))$ is trivial over $\mathbb{Z}/2\mathbb{Z}$. This showcases the sensitivity of Khovanov cobordism maps to the underlying ring.

The fact above is enough to compute the image of the Seifert surface cobordism map for an arbitrary positive link L. We can then extend this to any link L by first computing the induced map from negative twists (which will just give the enhanced state from the oriented resolution where all Seifert circles are labeled positively) and then doing the positive twists by following the rules of thumb above. Once one allows negative twists, the graph of the oriented resolution might not be connected even if the link is nonsplit. More formally:

Corollary 2.12. *Let L be a link and let $G(L)$ be the graph of Seifert circle for a projection of that link where two Seifert circles are connected if there exists a positive crossing joining them in the original projection. Let $S_1, \ldots, S_k$ be the connected components of the graph, $n_i = |S_i|$, and m_i be the number of edges counted with multiplicity in S_i. Then:*

- *If $m_i > n_i$ for some i, $Kh(S(L))(1) = 0$.*

- *Otherwise, without loss of generality let $S_1, \dots, S_l$ be the connected components for which $m_i = n_i - 1$ and number the Seifert circles in S_i as $\{c_i^1, \dots, c_i^{n_i}\}$. Let $I = \{(i_1, \dots, i_l) \in \mathbb{N}^l \mid 1 \le i_j \le n_j\}$ Then, up to an overall sign:*

$$Kh(S(L))(1) = 2^{k-l} \cdot \sum_{(i_1,\dots,i_l)\in I} (-1)^{d(c_1^1, c_1^{i_1}) \cdot d(c_2^1, c_2^{i_2}) \cdot \dots \cdot d(c_l^1, c_l^{i_l})} \cdot \sigma_{i_1,\dots,i_l}$$

where d is the usual distance function in our graph and $\sigma_{i_1,\dots,i_l}$ is the enhanced Kauffmann state of the oriented resolution in which all circles are labeled with a v_- except $c_j^{i_j}$ for $j = 1, \dots, l$.

One example of a negative link is the $T(2, k)$ torus link with negative crossings (mirror image), for which the proposition below follows from Theorem 2.10.

Proposition 2.13. *For $T(2, k)$ torus links with negative crossings, $1 \mapsto v_+ \otimes v_+$ under the Seifert surface cobordism map $Kh(S(L))$.*

In the next section, we use our result for the Seifert surface cobordism map of a general link to study the twist knots, for which we may understand exactly which elements in Kh^0 are generated by this image.

2.2 Twist Knots

Definition 2.14. A *twist knot* K_n is a knot made by making n Reidemeister 1 (R1) half-twists on a closed loop, and then merging the ends of the loop together.

One example of a twist knot is the 6_1 knot. It has four half-twists, so we denote it as K_4. A picture of K_4 can be found in Figure 5.

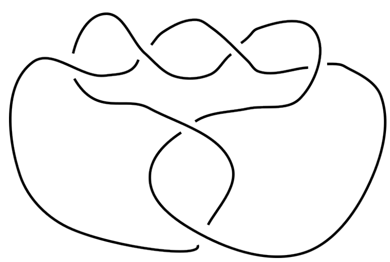


Figure 5: K_4

For twist knots, the number of Seifert circles $s = c - 1$ [5], where c is the number of crossings. Since the genus of a knot $g = \frac{1-s+c}{2}$, for twist knots, $g = \frac{1-c+1+c}{2} = 1$ [3].

The connection between the number of crossings of a twist knot and the number of Seifert circles makes it easier to understand the image of 1 under the Seifert surface cobordism for a twist knot: since K_n has $n+1$ Seifert circles, the image of 1 under the Seifert surface cobordism will be spanned by $n+1$ v_+s or v_-s tensored together.

We go into more detail about the homology of twist knots and the image of 1 under the Seifert surface cobordism for twist knots in the next section.

2.2.1 Odd Twist Knots

We define an *odd* twist knot to be a twist knot with an odd number of half-twists, which we can represent as K_{2n+1} for some whole number n.

It follows from Proposition 2.3 that the oriented resolution for K_{2n+1} is $2n + 2$ non-nested circles, as depicted in Figure 6.

Since these are negative links, Proposition 2.4 shows that the $\mathrm{Kh}^{(0,-1)}(K_{2n+1})$ is spanned by the oriented resolution of $\mathrm{Kh}^{(0,-1)}(K_{2n+1})$, where every Seifert circle is labeled with a v_+. The image of 1 under this Seifert surface cobordism spans $\mathrm{Kh}^{(0,-1)}$.

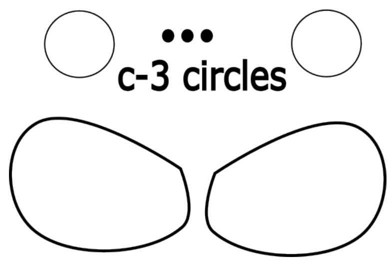


Figure 6: The oriented resolution for all odd twist knots

2.2.2 Even Twist Knots

We define an *even* twist knot to be a twist knot with an even number of half-twists, which we can represent as K_{2n} for some whole number n.

Proposition 2.15. *For K_{2n}, the oriented resolution consists of two nested circles and $2n-1$ non-nested circles, as depicted in Figure 7.*

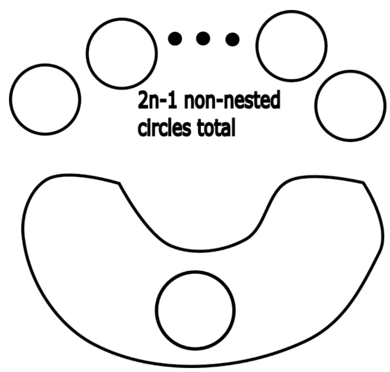


Figure 7: The oriented resolution for K_{2n}

Proof. We work inductively. The base case, K_2, is the 4_1 knot. Its oriented resolution is two nested circles and one non-nested circle, as shown in Figure 8.

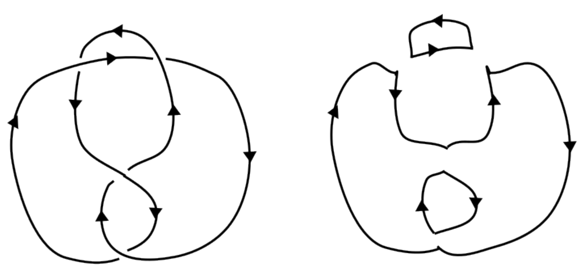
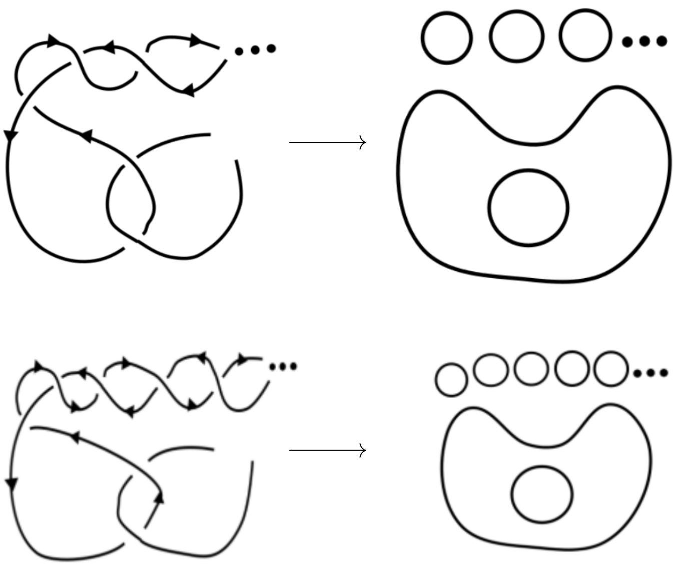


Figure 8: K_2 (left) and its oriented resolution (right)

For each m, $K_{2(m+1)} = K_{2m+2}$ just has two more non-nested circles than K_{2m}, as shown below.



(3)

So K_{2m+2} is of the same form as K_{2m}.

□

Next, we work to compute the homology for the even twist knots.

For any link L, recall that we can construct a graph $G(L)$, where each vertex is a Seifert circle and each edge represents a positive crossing, as described in Section 2.1.

Proposition 2.16. *For the oriented resolution, the labeling with v_-s on the two nested circles and v_+s on the non-nested circles (see Figure 9) is a nontrivial element in $Kh^{(0,-1)}$.*

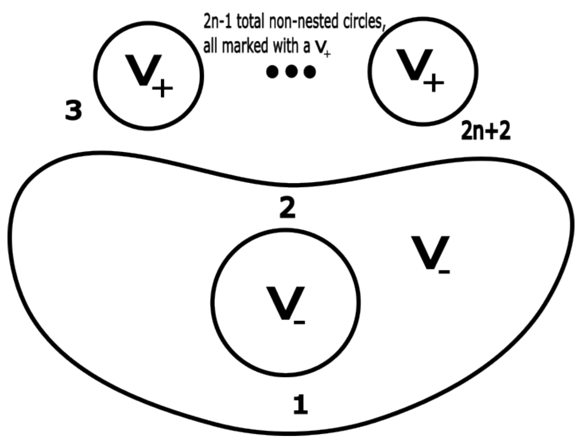


Figure 9: This labeling on the oriented resolution for the even twist knot generates $Kh^{(0,-1)}$

Proof. In order for something to be nontrivial in homological degree 0, it needs to be in the kernel of the next differential, d^0, but not in the boundary of d^{-1}.

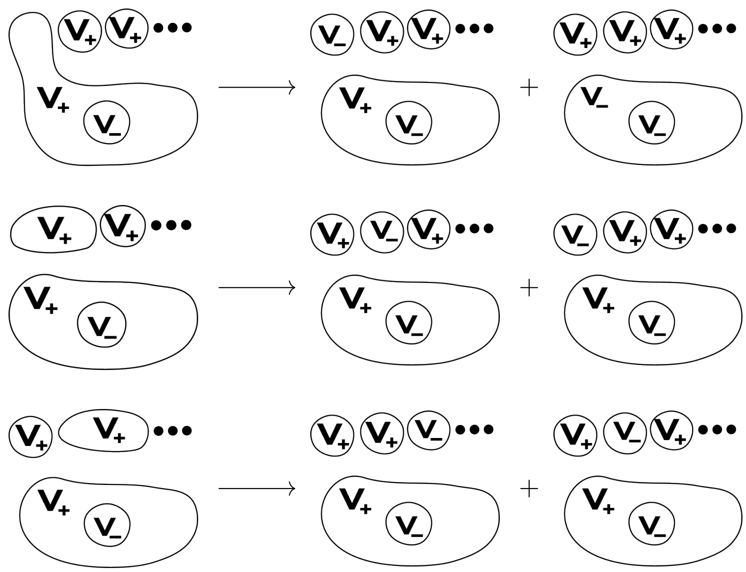
Looking at the oriented resolution, the only two 0-smoothings form the nested circles, because all of the half-twists form negative crossings. So, if we label the two nested circles with v_-, the diagram will certainly be in the kernel, because $m(v_- \otimes v_-) = 0$.

Without loss of generality, we create our state by numbering the crossings so that the bottommost crossing forming the nested circles is first, the other crossing forming the nested circles is second, and the rest of the crossings are numbered left to right. This is depicted in Figure 9 for convenience. Call this state σ.

Next, we want to show that this labeling of v_+s and v_-s is not the boundary. We proceed using a proof by contradiction: suppose that it is in the boundary. We construct a subspace A of C^0 generated by labelings of the oriented resolution A with qdegree -1 where the inner nested circle is labeled with a v_-. Note that this means that there are $2n$ elements in A, for the $2n$ circles where the second v_- can be placed. Let π_A be the projection of C^0 onto A, using its direct sum description.

Then by assumption, there exists x in C^{-1} so that $\pi_A \circ d^{-1}(x) = \sigma$.

Note that in order for $\pi_A \circ d^{-1}(x)$ to be nonzero, we must have that x is a labeled smoothing as shown below on the left, with it's image $\pi_A \circ d^{-1}$ on the right.



(4)

We create $2n$ vectors f_i, where f_i represents the element in the pre-image of d^{-1} where the ith 1-smoothing is replaced with a 0-smoothing (these are the vectors on the left of the diagram above). We also create $2n$ vectors g_i, where g_i represents the labeling on C^0 where the ith loop is labeled with a v_- and all the other loops are labeled with a v_+.

The matrix representing $\pi_A \circ d^{-1}$ has rows of the form $\pm(g_i + g_{(i+1)(\mod 2n)})$ because $\Delta(v_+) = v_+ \otimes v_- + v_- \otimes v_+$, and there is a sign on the differential. Because we are only looking to compute the rank of the matrix, we can multiply columns by -1. So, we can construct a matrix of the following form:

$$\begin{bmatrix} 1 & 0 & 0 & ... & 0 & 0 & 1 \\ 1 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & 1 & 1 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & 1 & ... & 0 \\ ... & ... & ... & ... & 1 & 1 & 0 \\ 0 & 0 & 0 & ... & 0 & 1 & 1 \end{bmatrix}$$

.

This matrix has rank $2n-1$, contradicting our initial assumption that the labeling in Figure 9 is in the boundary, because if it was in the boundary, then all g_i would be in the image, which is impossible. Therefore, the labeling in Figure 9 is nontrivial in $\mathrm{Kh}^{(0,-1)}$.

□

In fact, we found that the labeled smoothing in Figure 9 is a generator for $\mathrm{Kh}^{(0,-1)}$. In order to prove this, we will need to show a little bit of algebra first.

Proposition 2.17. *The Jones polynomial for an even twist knot with n twists is* $J(K_n) = \frac{t^3+t-t^{3-n}+t^{-n}}{t+1}$.

Proof. We proceed by induction. The Jones polynomial of the first even twist knot, $4_1 = K_2$, is $t^2 + t^{-2} - t - t^{-1} + t$ [2]. Algebra shows us that this is equal to $\frac{t^3+t-t^{3-2}+t^{-2}}{t+1}$.

Next, suppose that this is true for some even n. That is, $J(K_n) = \frac{t^3+t-t^{3-n}+t^{-n}}{t+1}$.

Now, we wish to show that it's also true for the next even twist knot, $n+2$. To do this, we apply the skein relation on one of the twists, as shown in Figure 10.

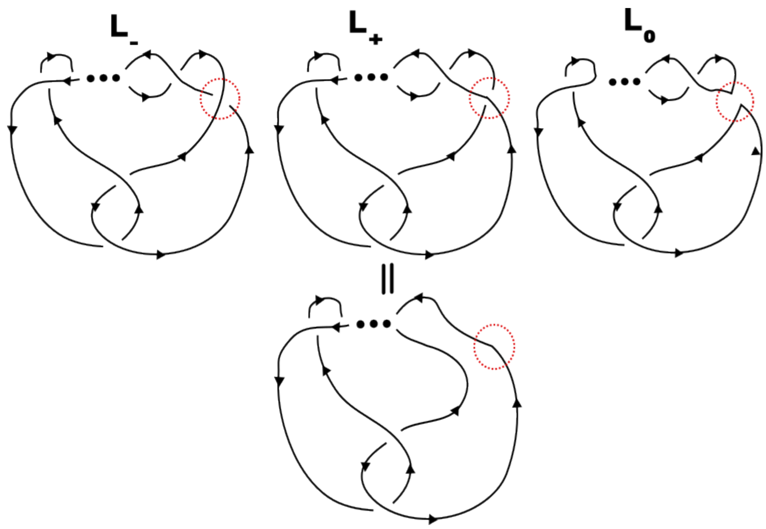


Figure 10: The skein relation for an even twist knot, applied to the twist crossings circled in red

For a Jones polynomial, the skein relation states that $t^{-1}J(L_+) - tJ(L_-) + (t^{\frac{-1}{2}} - t^{\frac{1}{2}})J(L_0) = 0$ [3]. Figure 10 depicts how L_- is K_{2n+2}, L_+ is K_{2n}, and L_0 is the Hopf link up to a lot of R1 moves.

In this case, for our L_0 Hopf link, the Jones polynomial is $-t^{\frac{5}{2}} - t^{\frac{1}{2}}$.

So, we have $t^{-1}J(K_{n+2}) - tJ(K_n) + (t^{\frac{-1}{2}} - t^{\frac{1}{2}})(-t^{\frac{5}{2}} - t^{\frac{1}{2}}) = 0$.

Bringing everything under a common denominator and solving for $J(K_{n+2})$, we get

$$J(K_{n+2}) = \frac{t^3 + t - t^{3-(n+2)} + t^{-(n+2)}}{t+1},$$

as desired.

□

Theorem 2.18. *For an even twist knot, the Khovanov polynomial over $\mathbb{Q}$ has coefficient 1 in front of the t^0q^{-1} term. $Kh^{(0,-1)}$ is generated by a labeled smoothing of the oriented resolution, and the Seifert surface cobordism maps one to twice this generator.*

Proof. It suffices to show that the coefficient of the $q^{-1}t^0$ term of $\mathrm{Kh}^{(0,-1)}$ is 1.

We want to utilize [9], which states that for an alternating knot,

$$\mathrm{Kh}(t,q) = \frac{q^\sigma}{qt^{\frac{1}{2}} + q^{-1}t^{-\frac{1}{2}}}(J(iqt^{\frac{1}{2}})(iqt^{\frac{1}{2}})^{-\sigma}(q^2t^{\frac{1}{2}} + q^{-2}t^{\frac{-1}{2}}) + t^{\frac{1}{2}} + t^{\frac{-1}{2}}),$$

where σ is the signature of the knot and i is the square root of -1.

First, we need to match the convention of [15] and [9] and rewrite the Jones polynomial in terms of q, where $q^2 = t$. So, we define $J'(q) = J(q^2)$. This gets us $J'(K_n) = \frac{q^6+q^2-q^{6-2n}+q^{-2n}}{q^2+1}$.

Evaluating $J'(iqt^{\frac{1}{2}})$, we get

$$\frac{-q^6t^3 - q^2t + q^{6-2n}t^{3-n} + q^{-2n}t^{-n}}{-q^2t+1}.$$

Since the even twist knots are oriented nonsplit alternating links, we calculate σ using Proposition 3.11 from [15]. That formula states that $\sigma(L) = o(D) - y(D) - 1$, where D is a reduced alternating diagram of the link L, $o(D)$ is the number of circles in the resolution of only 0s, and $y(D)$ is the number of right-handed crossings. For even twists knots, $o(D) = 3$ and $y(D) = 2$, so, $\sigma(L) = 0$.

Thus, the formula for the Khovanov polynomial of alternating links simplifies to

$$\mathrm{Kh}(t,q) = \frac{1}{qt^{\frac{1}{2}} + q^{-1}t^{-\frac{1}{2}}}\left(\frac{-q^6t^3 - q^2t + q^{6-2n}t^{3-n} + q^{-2n}t^{-n}}{-q^2t+1}(q^2t^{\frac{1}{2}} + q^{-2}t^{\frac{-1}{2}}) + t^{\frac{1}{2}} + t^{\frac{-1}{2}}\right).$$

Using difference of squares to get a nice denominator, we get

$$\frac{-q^9t^4 - q^5t^3 - q^5t^2 + q^{9-2n}t^{4-n} + q^{5-2n}t^{3-n} + q^{3-2n}t^{1-n} + q^{-1-2n}t^{-n} + q - q^3t^2 - q^3t}{1-q^4t^2}.$$

To deal with the denominator, we apply the power series expansion $\frac{1}{1-x} = \sum_{m=0}^{\infty} x^m$ to our expression and get

$$(-q^9t^4 - q^5t^3 - q^5t^2 + q^{9-2n}t^{4-n} + q^{5-2n}t^{3-n} + q^{3-2n}t^{1-n} + q^{-1-2n}t^{-n} + q - q^3t^2 - q^3t)\sum_{m=0}^{\infty} q^{4m}t^{2m}.$$

Now, we want to find the coefficient of the q^{-1} term. To do this, we notice that all of the terms that have constant, positive powers are not helpful, so we ignore them and focus on the terms with an expression of n in the exponent.

The first term is $q^{9-2n}t^{4-n}$. In order for $q^{9-2n}t^{4-n} \cdot q^{4m}t^{2m}$ to have t raised to the power of 0, we need $m = n-4$. Since m would be even, we could multiply the term by $q^{2n-8}t^{n-4}$ and get q. However, this is not the right qdegree, so it won't help us find a coefficient of q^{-1}.

The second term is $q^{5-2n}t^{3-n}$, so to get t^0 when we multiply by $q^{4m}t^{2m}$, we can try setting $m = \frac{n-3}{2}$. Since n is even, this is a non-integer power, which don't work in power series, so this won't help us find a coefficient of q^{-1}.

The third term is $q^{3-2n}t^{1-n}$, so we would need to multiple by $q^{2n-2}t^{n-1}$, but again, we cannot set $m = \frac{n-1}{2}$ because n is even.

Finally, the fourth term is $q^{-1-2n}t^{-n}$. In this case, we can set $m = \frac{n}{2}$, and we will get a product of q^{-1}, as desired. Therefore, the coefficient of $q^{-1}t^0$ is 1.

From here, it follows that the $q = -1$, $t = 0$ homology is generated by Figure 9, because the homology of $q = -1$, $t = 0$ is one-dimensional, and Figure 9 is in it. ☐

With this knowledge, we can now compare the image of 1 under the Seifert surface cobordism for the even twist knot with this generator.

Proposition 2.19. *Up to sign, the Seifert surface cobordism Σ for K_{2n} maps 1 to twice the generator of $Kh^{(0,-1)}(K_{2n})$.*

Proof. We refer to Figure 11, which depicts a condensed version of the movie diagram for $\Sigma(1)$.

To get from D_0 to D_1, we do $2n+1$ births, which map 1 to $1^{\otimes 2n+1}$. We note that we will be making both left-handed R1 moves to connect the non-nested circles and right-handed R1 moves to connect the nested circles. Since the chain map for a left-handed R1 is just a birth, we will not label the non-nested circles or the new edges made through the left-handed R1. On the other hand, we will label all of the edges involved in right-handed R1s.

Remark 2.20. In this proof, we will be writing out the steps for computing the image of the Seifert surface cobordism explicitly. However, one could also use the general procedure from Theorem 2.10.

So, we can also say that the map from D_0 to D_1 maps 1 to $v_+^1 \otimes v_+^2 \otimes v^{\otimes 2n-1}$.

To get from D_1 to D_2, we do a right-handed R1 on edge number 1. The chain map induces the expression $(v_+^1 \otimes v_-^3 - v_-^1 \otimes v_+^3) \otimes v_+^2 \otimes v^{\otimes 2n-1}$.

To get from D_2 to D_3, we do a right-handed R1 on edge 1 again. The chain map for a right-handed R1 move sends v_+ to $v_+ \otimes v_- - v_- \otimes v_+$ and v_- to $v_- \otimes v_-$, so we end up with $((v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_-^3 - v_-^1 \otimes v_+^3 \otimes v_-^4) \otimes v_+^2 \otimes v_+^{\otimes 2n-1}$.

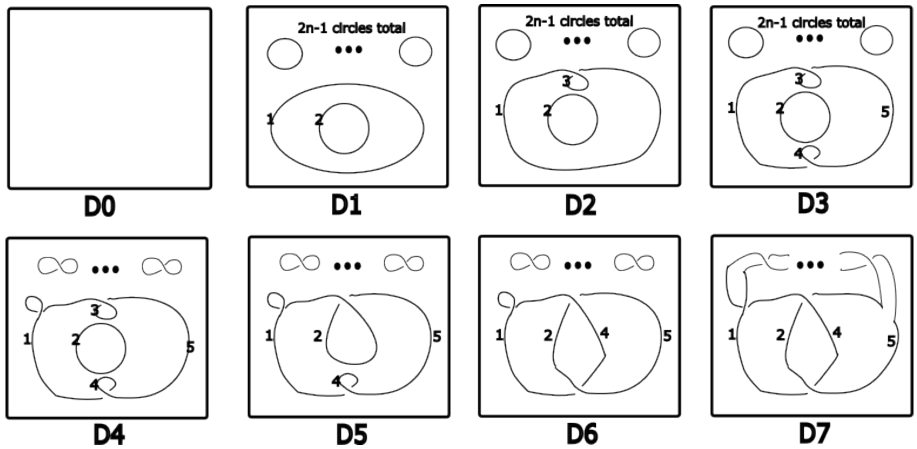


Figure 11: The movie diagram for the Seifert surface cobordism for K_{2n}

To get from D_3 to D_4, we do $2n-1$ left-handed R1s, which has the same chain map as $2n-1$ births. This gets us to $((v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_-^3 - v_-^1 \otimes v_+^3 \otimes v_-^4) \otimes v_+^2 \otimes v_+^{\otimes 4n-2}$.

To get from D_4 to D_5, we merge arc 2 and arc 3, getting us $((v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_-^2 - v_-^1 \otimes v_+^2 \otimes v_-^4) \otimes v_+^{4n-2}$.

To get from D_5 to D_6, we merge arcs 2 and arcs 4. Since the merge map on $v_- \otimes v_-$ goes to 0, the algebra simplifies to $-2v_-^1 \otimes v_-^2 \otimes v_+^{\otimes 4n-2}$.

Finally, to get from D_6 to D_7, we merge all of the left-handed R1s, getting $-2v_-^1 \otimes v_-^2 \otimes v_+^{\otimes 2n-1}$.

This diagram has two v_-s and $2n-1$ v_+s on the same edges as Figure 9, which by Proposition 2.16 is the generator of $\mathrm{Kh}^{(0,-1)}(K_{2n})$. $\square$

Remark 2.21. Over $\mathbb{Z}$, $\mathrm{Kh}^{(0,-1)}$ has the same rank as over $\mathbb{Q}$, with the same generator for its free summand. However, over $\mathbb{Z}$, the image of 1 under the Seifert surface cobordism for an even twist knot does not generate $\mathrm{Kh}^{(0,-1)}$, because the image is twice the generator.

3 Alternating Knots

We've already categorized for which knots the image of 1 under the Seifert Surface Cobordism is 0 on the chain level. We can go further and figure out when the Khovanov homology at the corresponding quantum degree and homological degree is zero, as well.

Theorem 3.1. *For an alternating knot projection K in which $G(K)$ is a connected graph with more edges than vertices, $Kh^{(0,\chi)} = 0$, where χ is the Euler characteristic of the Seifert surface.*

Proof. For an alternating knot, the places where Khovanov homology is nonzero are of the form (i,j), where i is the homological degree, j is the quantum degree, and $j = 2i - \sigma(K) \pm 1$ [15]. (Here, σ is the signature of the link, which we write as $s_0 - n_+ - 1$, with s_0 representing the number of circles in the resolution where every crossing is 0-smoothed and n_+ representing the number of positive crossings.)

This means that when $i = 0$, the minimum supported quantum degree is $-\sigma(K) - 1 = -s_0 + n_+$. So, it suffices to show that $\chi < -s_0 + n_+$.

Because $\chi = s - c$, where s is the number of Seifert circles and c is the number of crossings, we want to show that $s - c < -s_0 + n_+$. Rearranging this inequality, we get the equivalent inequality

$$s - n_+ < c - s_0$$

.

The rest of the proof focuses on proving that this inequality holds. Since $G(K)$ is connected and has more edges than vertices, $s < n_+$, so $s - n_+$ is negative. Thus, it suffices to show that $c - s_0 \geq 0$.

We begin by smoothing the entire knot K with 0-smoothings. Then, we go through each negative crossing and replace its 0-smoothing with a 1-smoothing, creating a link that looks like our knot, except for the fact that the knot's negative crossings have been replaced with 1-smoothing loops. This new link has s'_0 circles and c' crossings.

We claim that $s_0 - s'_0 \leq c - c'$. This is because for each negative crossing we remove, the number of circles can increase by 1 or decrease by 1, while the number of crossings always decreases by 1.

Now that we are left with a resolution of a positive knot, our graph of only zero smoothings is the oriented resolution, and still has $s < n_+$. This means that $s'_0 = s < n_+ = c'$.

Adding together the two inequalities $s_0 - s'_0 \leq c - c'$ and $s'_0 < c'$, we get $s_0 < c$, so $c - s_0 \geq 0$, as desired. □

In fact, this idea generalizes to any number of components of $G(K)$.

Theorem 3.2. *For an alternating knot projection K where $G(K)$ has a connected component with more edges than vertices, $Kh^{(0,\chi)} = 0$, where χ is the Euler characteristic of the Seifert surface.*

Proof. Suppose that we have some knot K where $G(K)$ contains $n + 1$ connected components, at least one of which has more edges than vertices. We number the components 0 through n, where 0th component has more edges than vertices.

Just like in the previous theorem, we want to show that $s - n_+ < c - s_0$ for K, where s is the number of Seifert circles, n_+ is the number of positive crossings, c is the number of total crossings, and s_0 is the number of circles in the all 0-smoothing resolution.

In other words, we want to show that $\sum_{i=0}^{n} s^{(i)} - n_+^{(i)} < c - s_0$. Since K is a knot, we split c into N_-, the number of negative crossings joining distinct connected components of $G(K)$, and $c^{(i)}$, the number of positive and negative crossings occurring in the region of K corresponding to the ith connected component.

We already know that $s^{(0)} - n_+^{(0)} < c^{(0)} - s_0^{(0)}$, so now we we just need to show that $\sum_{i=0}^{n} s^{(i)} - n_+^{(i)} \leq N_- + \sum_{i=0}^{n} c^{(i)} - s_0 + s_0^{(0)}$, or, equivalently, that

$$N_- + \sum_{i=1}^{n} c^{(i)} + s_0^{(0)} - s_0 - (\sum_{i=1}^{n} s^{(i)} - n_+^{(i)}) \geq 0$$

. The rest of this proof will focus on evaluating the expression on the left-hand side.

We are going to look at our total knot K with two different graphs.

The first one is $G(K)$, in which the vertices are Seifert circles and the edges are positive crossings.

The second one is $H(K)$, which we use to understand the difference between $s_0^{(0)}$ and s_0. We construct $H(K)$ by taking the Seifert circles corresponding to each connected component of $G(K)$ and re-smoothing all the negative crossings from 1-smoothings to 0-smoothings. This means that the negative crossings going between distinct connected components of $G(K)$ should not be 0-smoothed for $H(K)$, but the negative crossings within connected components of $G(K)$ should be 0-smoothed.

Here is an example knot, with $G(K)$ and $H(K)$ labeled.

We can use $H(K)$ to create a definitive expression for s_0. Let $s_{0_{involved}}^{(i)}$ denote the circles in the ith connected component of $G(K)$ that have at least one edge coming out of them in $H(K)$. Similarly, $s_{0_{uninvolved}}^{(i)}$ denotes the circles in the ith component of $G(K)$ that have no edges coming out of them in $H(K)$. (The subscript 'involved' marks the circles which will change when we 0-smooth the negative crossings represented by edges in $H(K)$, and the subscript 'uninvolved' marks the circles which won't change during this process.)

Now, look at each connected component of $H(K)$. When we 0-smooth each of the negative crossings represented by edges, the new number of circles we get will be equal to the number of faces in the planar graph for $H(K)$. Figure 13 contains an example that will make this visually clear.

By Euler's formula for planar graphs $v - e + f = 2$, where v is the number of vertices, e is the number of edges, and f is the number of faces), this means that the new number of circles is $f = 2 + e - v$, or, in our notation, $2 + N_- - \sum_{i=1}^{n} s_{0_{involved}}^{(i)}$.

Therefore, $s_0 = \sum_{i=0}^{n} s_{0_{uninvolved}}^{(i)} + 2 + N_- - \sum_{i=0}^{n} s_{0_{involved}}^{(i)}$.

This means that $s_0^{(0)} - s_0 = s_0^{(0)} + \sum_{i=0}^{n} s_{0_{involved}}^{(i)} - 2 - N_- - \sum_{i=0}^{n} s_{0_{uninvolved}}^{(i)}$.

Plugging this into the expression that we're trying to evaluate, we get

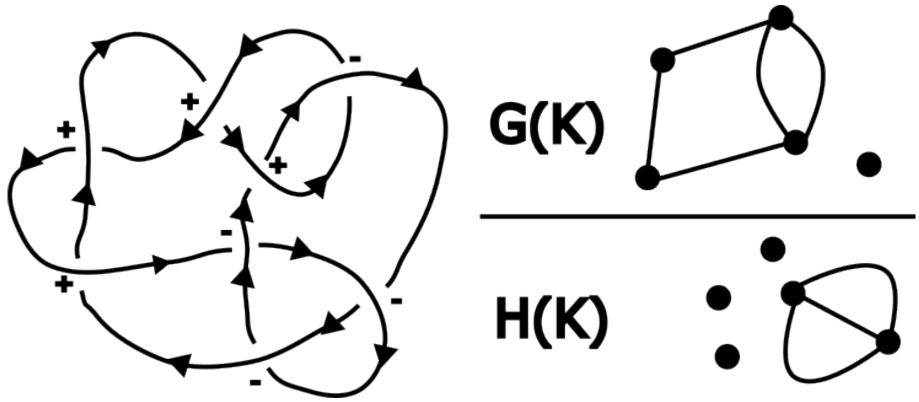


Figure 12: The 8_8 knot, along with its corresponding $G(K)$ and $H(K)$. In this case, none of the negative crossings occured between two circles within a connected component, so the vertices of $G(K)$ are the vertices of $H(K)$. Positive crossings are labeled with a + and negative crossings are labeled with a $-$.

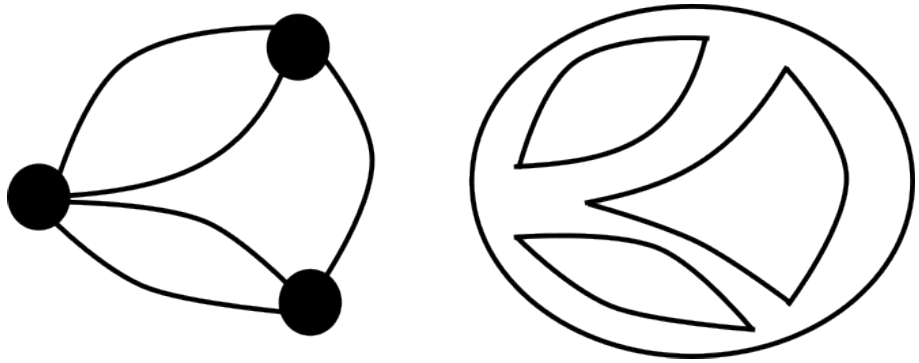


Figure 13: An example of a component of $H(K)$ and the corresponding all-0 smoothing.

$$
\begin{aligned}
& N_- + \sum_{i=1}^{n} c^{(i)} + s_0^{(0)} - s_0 - (\sum_{i=1}^{n} s^{(i)} - n_+^{(i)}) \\
&= N_- + \sum_{i=1}^{n} c^{(i)} + s_0^{(0)} + \sum_{i=0}^{n} s_{0_{involved}}^{(i)} - 2 - N_- - \sum_{i=0}^{n} s_{0_{uninvolved}}^{(i)} - (\sum_{i=1}^{n} s^{(i)} - n_+^{(i)}) \\
&= \sum_{i=1}^{n} c^{(i)} + 2s_{0_{involved}}^{(0)} + \sum_{i=1}^{n} s_{0_{involved}}^{(i)} - 2 - \sum_{i=1}^{n} s_{0_{uninvolved}}^{(i)} - (\sum_{i=1}^{n} s^{(i)} - n_+^{(i)}) \\
&= \sum_{i=1}^{n} (c_i - s_{0_{involved}}^{(i)} - s_{0_{uninvolved}}^{(i)}) + 2\sum_{i=0}^{n} s_{0_{involved}}^{(i)} - 2 - \sum_{i=1}^{n} s^{(i)} - n_+^{(i)}
\end{aligned}
$$

.

Now, we provide lower bounds of some of our expressions.

First we show that $\sum_{i=1}^{n} c_i - s_{0_{involved}}^{(i)} - s_{0_{uninvolved}}^{(i)} \geq -n$. Here's why: when you take each connected component of $G(K)$, since it is connected, we know that $c_i + 1 \geq s^{(i)}$ is true for the oriented resolution, because we know by connectedness that $n_+^{(i)}$ (the number of positive crossings in that component) has $n_+^{(i)} + 1 \geq s^{(i)}$. To get to the all-0 smoothing, we would replace the 1-smoothings corresponding to negative crossings with 0-smoothings. Each change from a 0-smoothing to a 1-smoothing can either increase the number of circles by 1 or decrease the number of circles by one. The number of changes is equal to the number of negative crossings in that component, so we can bound the change in the number of circles by the number of negative crossings, which we denote $n_-^{(i)}$. So, $n_+^{(i)} + n_-^{(i)} + 1 \geq s_0^{(i)}$.

Next, we show that $\sum_{i=0}^{n} s_{0_{involved}}^{(i)} \geq n + 1$. This is because there are $n + 1$ connected components of $G(K)$, and we need to join all of them together with negative crossings, because if they weren't joined together, K would be a link, not a knot. Therefore, we need to pick at least one circle from each of the $n+1$ components, so we need at least $n + 1$ circles.

Finally, because each component of $G(K)$ is connected, $n_+^{(i)} + 1 \geq s^{(i)}$, which means that $\sum_{i=0}^{n} s^{(i)} - n_+^{(i)} \leq n$.

This means that we can bound our entire expression by
$\geq -n + 2(n + 1) - 2 - n$
$= 0$. This completes the proof.

□

4 Computational Methods

4.1 Movie cobordism code

As explained in Section 1.2, the Khovanov chain maps induced by a surface cobordism between two links is algorithmic in the sense that it can be represented by finitely many movie moves. Consequently, we developed an algorithm to represent cobordisms and compute the chain maps they induce.

The code runs in Python 3.12 and is based on SageMath's implementation of Khovanov homology for knots and links [1]. The chain complexes are computed as free modules over the base ring, with a basis given by enhanced Kauffman states as described in 2.4 of [19]. Each map is constructed by relying on the local behavior of movie moves.

Oriented links are represented by a Sage object and described by their planar diagram notation, where each strand is labeled by a *negative* integer. A crossing is represented by 4 integers in order, representing the 4 strands forming the crossing, starting from the understrand going into the crossing and then circling around counterclockwise. Since the code relies on planar diagram notation, simple loops like copies of the unknot are represented by twisted loops instead (except the unknot who's planar diagram notation is []). Conventions are described more carefully in 7.3. The code is referenced in [17]. One main difficulty in implementing the code was explicitly describing the homotopy equivalences induced by the four possible $R3$ variants, detailed in the appendix. This required careful thought about sign conventions for the chain complexes and chain maps, and use of the cone of the saddle map one gets by resolving the middle crossing, as described in [4].

4.2 Ribbon Knots

Definition 4.1. A *ribbon knot* is a knot that can be represented as a cobordism with births, R1 moves, R2 moves, and no deaths. We call this cobordism a *ribbon cobordism*.

The ribbon disk gives an injective map on Khovanov homology, with the left inverse given by the cobordism in the opposite direction. Because of this, the image of the ribbon cobordism is the sum of labeled smoothings that all have a coefficient of 1 [16].

One example is the 6_1 ribbon knot, which is pictured in Figure 14. The image of 1 under the ribbon cobordism for this knot is too big to compute by hand: it's the direct sum of 32 different labelings. So, we use our code to do it.

Here are the commands we used to generate the movie diagram for this ribbon.

```
ribbon_6_1 = Movie(Link([]));
ribbon_6_1.birth();
ribbon_6_1.poke(-1, -3, -1, -1);
ribbon_6_1.poke(-5, -8, 1, 1);
ribbon_6_1.untwist(-2);
ribbon_6_1.untwist(-4);
```

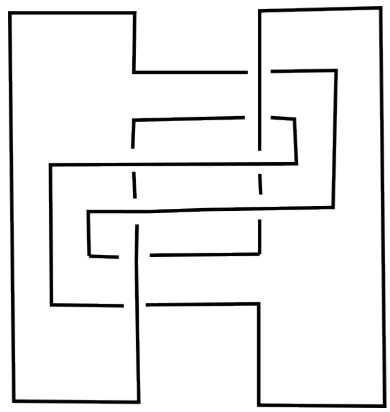


Figure 14: The 6_1 ribbon, as described in [13]

```
ribbon_6_1.poke(-9, -7, -1, 1);
ribbon_6_1.poke(-13, -7, 1, -1);
ribbon_6_1.saddle(-17, -12);
print(ribbon_6_1.push(0, vector([1])));
```

Figures 15 and 16 show some steps of the movie diagram computed by our code.

The print statement from our program gives us the planar diagram notation for the image of 1 under the 6_1 ribbon cobordism, up to sign.

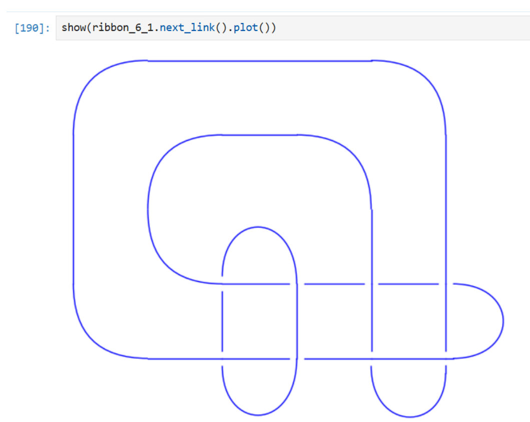


Figure 15: One of the intermediate movie diagram images for the 6_1 ribbon knot

We also mention that when updating this preprint, the authors found another implementation by Zsombor Fehér [8] of a similar computational tool to compute Khovanov homology maps induced by cobordisms.

5 Future Directions

This project mainly investigated Seifert surface cobordisms, as these were the easiest to understand computationally when working out examples by hand. Now that there is working code to compute cobordism maps, we would like to collect data about more complicated cobordisms, such as ribbon disks. We also hope to use our description of the Seifert surface map to further describe when elements in the Khovanov homology of alternating knots are Seifert surface realizable.

Question 5.1. *For dotted Seifert surface cobordisms, what does the image of 1 look like for positive and negative links?*

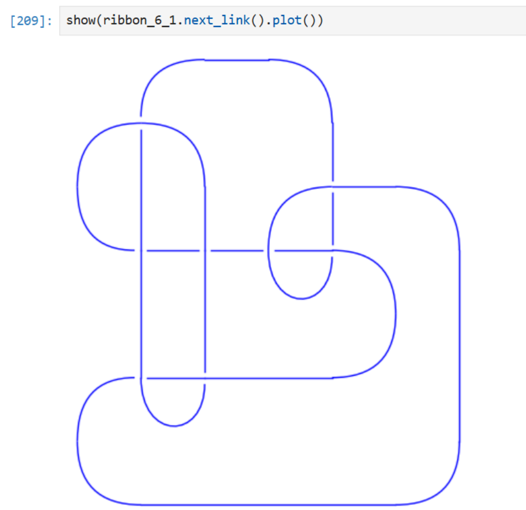


Figure 16: The 6_1 ribbon knot

Question 5.2. *The Seifert surface cobordism map can output twice a labeled smoothing. Is it possible to see other coefficients other than 2^p from more complicated surface cobordisms? This question is related to questions about non-$\mathbb{Z}/2^p\mathbb{Z}$ torsion in Khovanov homology.*

Question 5.3. *Is the zeroth homology of a cable of a negative link Seifert surface realizable?*

6 Acknowledgements

This paper was written as part of the SPUR+ program at MIT. The authors would like to thank their mentor, Mary Stelow, for her guidance and expertise. They'd also like to thank Roman Bezrukavnikov, Jonathan Bloom, and Yufei An for advising and organizing the SPUR program.

References

[1] SageMath Link documentation. Available at `https://doc.sagemath.org/html/en/reference/knots/sage/knots/link.html`.

[2] The Knot Atlas. Available at `https://katlas.org/wiki/Main_Page`.

[3] Colin Conrad Adams. *The knot book: an elementary introduction to the mathematical theory of knots.* American Mathematical Soc., 2004.

[4] Dror Bar-Natan. Khovanov's homology for tangles and cobordisms. *Geom. Topol. 9 (2005) 1443-1499*, 2005.

[5] Nairi Davidian and Winnie Gong. Knot theory: Genus. PRIMES Circle, Cambridge, MA, 2024.

[6] Simon K Donaldson. An application of gauge theory to four-dimensional topology. *Journal of Differential Geometry*, 18(2):279–315, 1983.

[7] Andrew Elliott. *State cycles, quasipositive modification, and constructing H-thick knots in Khovanov homology.* Rice University, 2010.

[8] Zsombor Fehér. A new computational tool for Khovanov cobordism maps, 2025.

[9] Stavros Garoufalidis. A conjecture on Khovanov's invariants. *Fundamenta Mathematicae*, 184:99–101, 2004.

[10] Kyle Hayden, Seungwon Kim, Maggie Miller, JungHwan Park, and Isaac Sundberg. Seifert surfaces in the 4-ball. *arXiv preprint arXiv:2205.15283*, 2022.

[11] Kyle Hayden and Isaac Sundberg. Khovanov homology and exotic surfaces in the 4-ball. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 2024(809):217–246, 2024.

[12] Magnus Jacobsson. An invariant of link cobordisms from Khovanov homology. *Algebr. Geom. Topol. 4 (2004) 1211-1251*, 2004.

[13] Akio Kawauchi. *Survey on knot theory*. Springer Science & Business Media, 1996.

[14] Mikhail Khovanov. A categorification of the Jones polynomial. 2000.

[15] Eun Soo Lee. An endomorphism of the Khovanov invariant. *Advances in Mathematics*, 197(2):554–586, 2005.

[16] Adam Simon Levine and Ian Zemke. Khovanov homology and ribbon concordance. *arXiv preprint arXiv:1903.01546*, 2019.

[17] Mohamed Wacyl Meddour and Veronika Moroz. Khovanov homologie movie on cobordisms. `https://github.com/Wacyl/Cobordism-induced-maps-on-Khovanov-homology-SPUR-2025-`, 2025.

[18] Isaac Sundberg and Jonah Swann. Relative Khovanov–Jacobsson classes. *Algebraic & Geometric Topology*, 22(8):3983–4008, 2023.

[19] Oleg Viro. Remarks on definition of Khovanov homology. *arXiv:math/0202199v1*, 2002.

7 Appendix

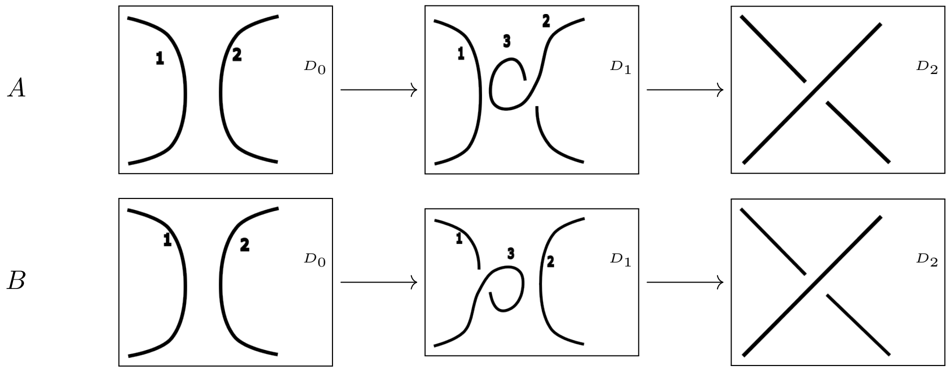
7.1 Well-definedness of Seifert surface cobordisms

In our description of the movie presentation of Seifert's algorithm, there were some internal choices of which strands to twist and in which order to do the births, twists, and merges.

Remark 7.1. In fact, all such descriptions will give smoothly isotopic Seifert surfaces, which can be shown via an appeal to the Carter-Saito movie moves or via explicit description of isotopies. To avoid introducing new terminology here, we instead simply show that these surfaces all produce the same maps on Khovanov homology up to sign. Note that we can explicitly see the problem of sign ambiguity for the Khovanov surface maps in this example.

Proposition 7.2. *Up to sign, the choices made in the description of the movie presentation of Seifert's cobordism do not affect the maps on Khovanov homology.*

Proof. The Seifert surface cobordism is composed of births, left-handed R1 moves, right-handed R1 moves, and merges. The chain map for a birth just adds a v_+, and the chain map for a left-handed R1 move is a birth. So, doing a left-handed R1 move and then a merge induces the identity on chain maps. Now, we just need to show that for right-handed R1 moves followed by merges, it doesn't matter which of the two circles will be twisted as long as the twist is merged with the other circle. In other words, we want to show that the chain map on the top and bottom of the figure below are the same, up to sign.



(5)

If both loops are initially marked with v_+, then for Diagram A, the first step is $v_+^1 \otimes v_+^2$, and the second step is $v_+^1 \otimes (v_+^2 \otimes v_-^3 - v_-^2 \otimes v_+^3)$. Merging loops 1 and 3, we get $v_+^2 \otimes v_-^1 - v_-^2 \otimes v_+^1$. For B, the first step is $v_+^1 \otimes v_+^2$, performing the right-handed R1 move gets us $(v_+^1 \otimes v_-^3 - v_-^1 \otimes v_+^3) \otimes v_+^2$, and merging loops 2 and 3 gets us $v_+^1 \otimes v_-^2 - v_-^1 \otimes v_+^2$, which is exactly -1 multiplied by the result from Diagram A.

If the left loop is initially marked with v_- and the right loop is initially marked with v_+, for Diagram A, the second step gets us $v_-^1 \otimes (v_+^2 \otimes v_-^3 - v_-^2 \otimes v_+^3)$, and merging loops 1 and 3 gets us $-v_-^1 \otimes v_-^2$. For Diagram B, the second step gets us $v_-^1 \otimes v_-^3 \otimes v_+^2$, and merging 2 and 3 gets us $v_-^1 \otimes v_-^2$, which is exactly -1 multiplied by the result from Diagram A.

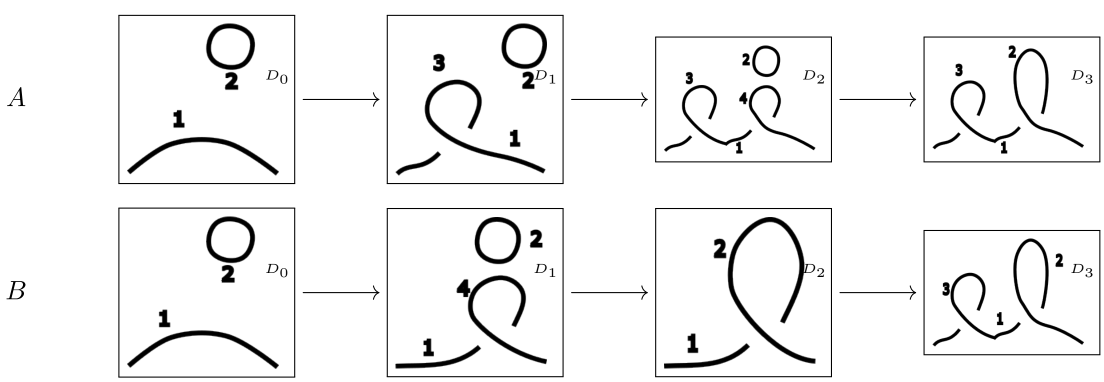
If the left loop is initially marked with v_+ and the right loop is initially marked with v_-, for Diagram A, the second step gets us $v_+^1 \otimes v_-^2 \otimes v_-^3$, and merging loops 1 and 3 gets us $v_-^1 \otimes v_-^2$. For Diagram B, the second step gets us $(v_+^1 \otimes v_-^3 - v_-^1 \otimes v_+^3) \otimes v_-^2$, and merging loops 3 and 2 gets us $-v_-^1 \otimes v_-^2$, which is exactly -1 times the result from Diagram A.

Finally, if both loops are initially marked with v_-, the second step for both Diagram A and B is $v_-^1 \otimes v_-^2 \otimes v_-^3$, and any merge will result in 0.

□

Proposition 7.3. *For any Seifert surface cobordism Σ, the image of $Kh(\Sigma)$ is independent of the order of births and merges in the movie presentation for Σ.*

Proof. Again, the chain map of the left-handed R1 move composed with the merge is the identity on chain maps, so it suffices to show that it doesn't matter what order we perform right-handed R1 moves and merges in. In other words, we want to show that Diagrams A and B below result in the same image. This is sufficient, because any situation where we do some merges before we finish twisting can be represented by a labeling of v_+s and v_-s on this diagram.



(6)

There are four possible cases to check, representing the four different ways we can label loops 1 and 2 with v_+s and v_-s.

The first case is when D_0 is $v_+^1 \otimes v_+^2$. For Diagram A, D_1 becomes $(v_+^1 \otimes v_-^3 - v_-^1 \otimes v_+^3) \otimes v_+^2$. Then, D_2 is $((v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_-^3 - v_-^1 \otimes v_-^4 \otimes v_+^3) \otimes v_+^2$. Finally, merging loops 2 and 4 gives us $((v_+^1 \otimes v_-^2 - v_-^1 \otimes v_+^2) \otimes v_-^3) - v_-^1 \otimes v_-^2 \otimes v_+^3$ for D_3. For Diagram B, D_1 becomes $(v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_+^2$. Merging loops 2 and 4, we get $v_+^1 \otimes v_-^2 - v_-^1 \otimes v_+^2$. A final right-handed R1 twist gets us D_3, which is $(v_+^1 \otimes v_-^3 - v_-^1 \otimes v_+^3) \otimes v_-^2 - v_-^1 \otimes v_+^2 \otimes v_-^3$. Expanding and refactoring this expression gives us the same answer as in Diagram A.

The second case is when loop 1 is labeled with a v_+ and loop 2 is labeled with a v_-. For Diagram A, D_1 becomes $(v_+^1 \otimes v_-^3 - v_-^1 \otimes v_+^3) \otimes v_-^2$. Then, D_2 is $((v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_-^3 - v_-^1 \otimes v_-^4 \otimes v_+^3) \otimes v_-^2$. Finally, merging loops 2 and 4 gives us D_3, which is $-v_-^1 \otimes v_-^2 \otimes v_-^3$. For Diagram B, D_1 becomes $(v_+^1 \otimes v_-^4 - v_-^1 \otimes v_+^4) \otimes v_-^2$. Merging loops 2 and 4 gets us $-v_-^1 \otimes v_-^2$. Finally, doing one more right-handed R1 twist gets us $-v_-^1 \otimes v_-^2 \otimes v_-^3$, just like in Diagram A.

The third case is when loop 1 is labeled with v_-, and loop 2 is labeled with v_+. For Diagram A, D_1 is $(v_-^1 \otimes v_-^3) \otimes v_-^2$, D_2 is $(-v_-^1 \otimes v_-^3 \otimes v_-^4) \otimes v_+^2$, and D_3 is $v_-^1 \otimes v_-^2 \otimes v_-^3$. For Diagram B, D_1 is $(v_-^1 \otimes v_-^4) \otimes v_+^2$, D_2 is $v_-^1 \otimes v_-^2$, and D_3 is $v_-^1 \otimes v_-^2 \otimes v_-^3$, just like in Diagram A.

The fourth and final case is when both loop 1 and loop 2 are labeled with v_-s. For Diagram A, D_1 is $v_-^1 \otimes v_-^3 \otimes v_-^2$, D_2 is $v_-^1 \otimes v_-^3 \otimes v_-^2 \otimes v_-^4$, and D_3 is 0. For Diagram B, D_1 is $v_-^1 \otimes v_- 4 \otimes v_-^2$, and D_2 is 0, so D_3 is 0, just like in Diagram A.

Thus, changing the order of the twists and merges does not change the image of the Seifert surface cobordism.

□

7.2 Homotopy equivalences

In this section, we explain the convention we chose for the homotopy equivalences under Reidemeister moves. Indeed, if L and L' are links that differ by one Reidemeister move, then from invariance of Khovanov homology, $\mathrm{Kh}(L) \simeq \mathrm{Kh}(L')$ by Theorem 1 in [4]. These are also the maps used internally by the code, whose conventions we will explain in the next section.

The homotopy equivalences presented in this section are from [11], with some changes provided to the tables of $R3$ moves.

Table 1: Chain maps induced by a R1. This table is reproduced from [11]

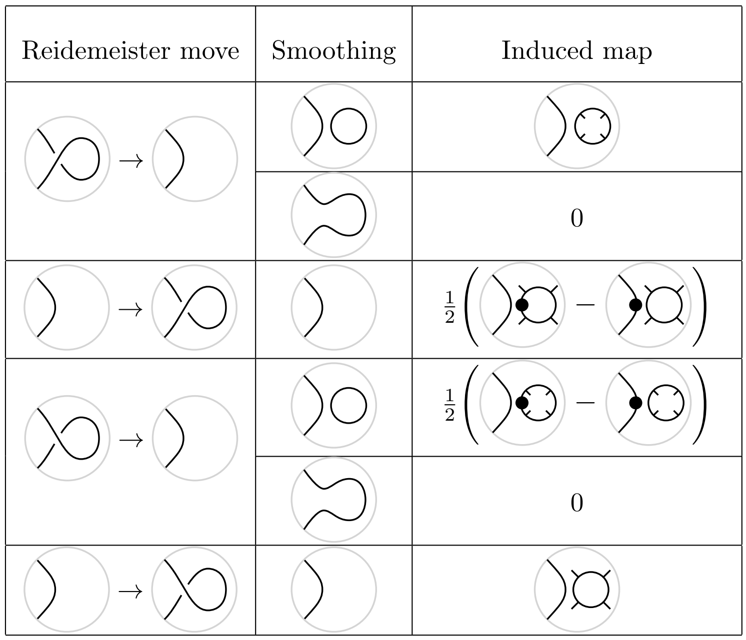


Table 2: Chain maps induced by an R2 [11]

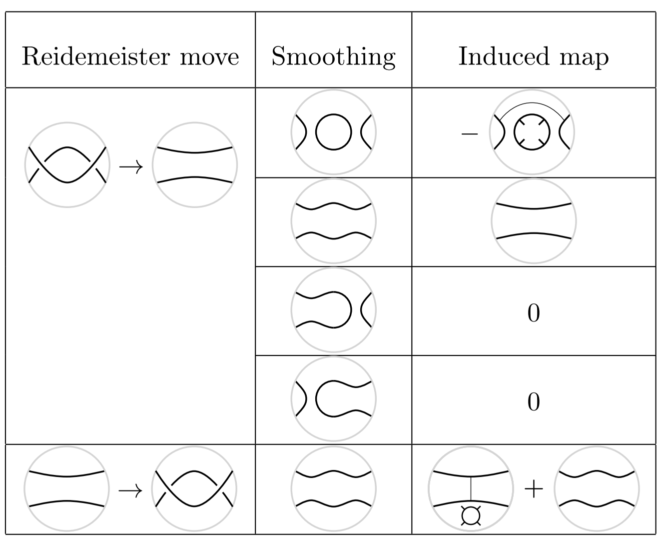


Table 3(a)

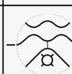
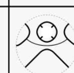
	000	100	010	001	110	101	011	111
000	I							
100			I	I				
010		I						
001								
110					$-I$			
101						$-I$	- 	
011								
111								

Table 3(b)

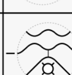
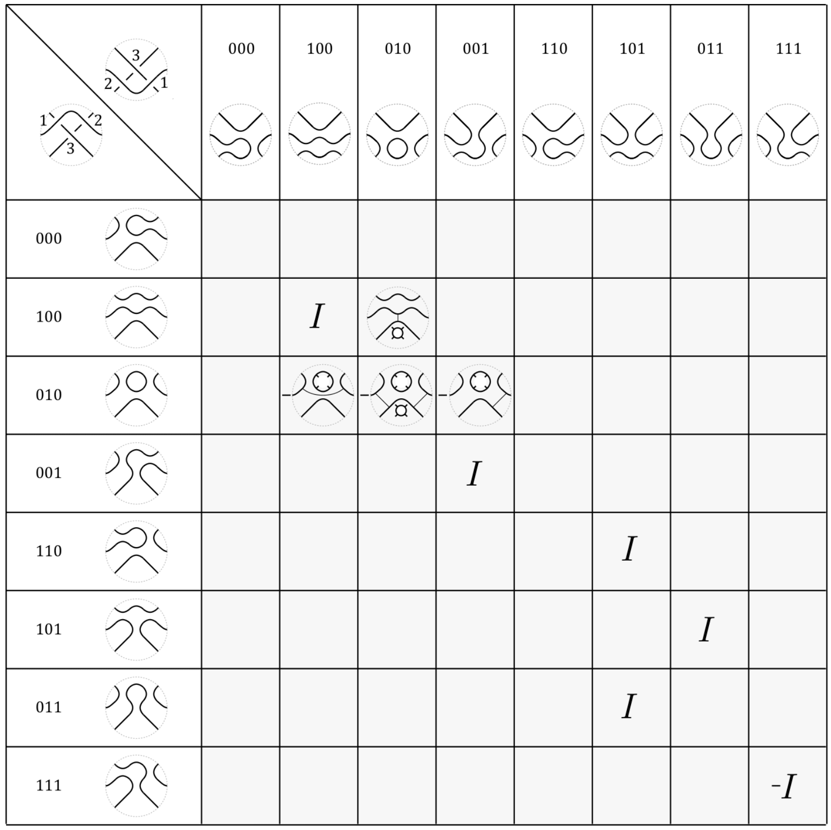
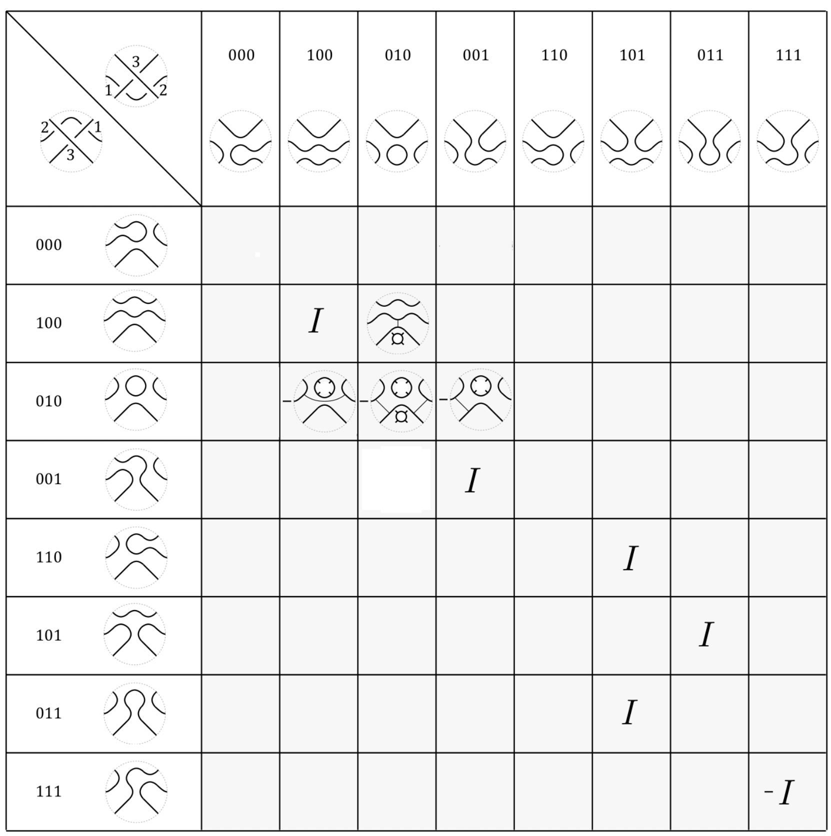
	000	100	010	001	110	101	011	111
000	I							
100			I	I				
010		I						
001								
110					$-I$			
101						$-I$	- 	
011								
111								

Table 3(c)



	000	100	010	001	110	101	011	111
000								
100		I						
010		–	–	–				
001				I				
110						I		
101							I	
011						I		
111								$-I$

Table 3(d)



	000	100	010	001	110	101	011	111
000								
100		I						
010		–	–	–				
001				I				
110						I		
101							I	
011						I		
111								$-I$

7.3 Code conventions

The code runs in Python and is based on SageMath's latest implementation at the time of release. Movies are computed degreewise. To start a movie diagram, one needs to create an instance of the Movie class as follows:

```
movie_diagram = Movie(link, starting_qdeg, ring)
```

The *link* argument is a SageMath Link object. The *starting_ qdeg* argument represents the starting qdegree for the link, 1 by default. The *ring* argument represents the underlying ring to compute Khovanov homology, by default $\mathbb{Q}$. **The link argument should be inputted via planar diagram notation where all strands are labeled with negative numbers.**

The movie works by allowing the user to input Reidemeister moves, requesting enough information each time to compute the next link's planar diagram notation as well as the induced chain map. It is suggested to track the planar diagram notation while constructing a movie to avoid potential errors.

Because the underlying program uses planar diagram notation, we take the convention that birthing a circle actually births a twisted circle instead, and that killing a circle requires killing a twisted circle.

We now explain the methods of the movie class as well as how they translate in terms of planar diagram notation changes. The program internally keeps track of the minimum strand number (a negative number) that appears in the planar diagram notation. Should it need more labels to perform a move, it will go down in negatives from the last minimum label. The relevant methods for the class are:

- `twist(self, strand, orientation = 1, strand_type = 1, print_pd=True):`

  This method creates a twist at the inputted strand as a label. If **orientation** is 1, then the 0-resolution of the loop crossing will have an additional circle, and if **orientation** is -1, it will not. In terms of labeling, the strand incoming to the loop crossing keeps the old *strand* label. The label of the loop is the next available label by our convention (1 lower than the minimum label in the diagram), and the strand outgoing is one less than that. Should we twist the unknot, the outgoing strand for the loop crossing will also keep the *strand* label. The *strand_ type* argument should be 1 if we desire the crossing to be an overcrossing and -1 otherwise.

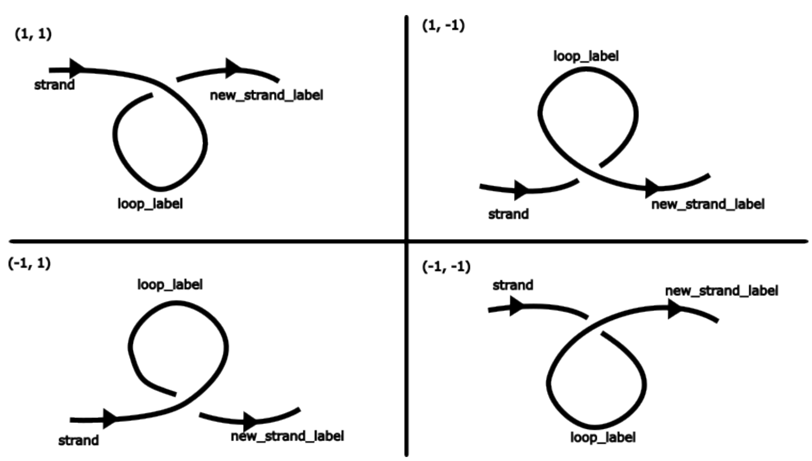


Figure 17: R1 Move Labeling Convention; tuple on upper left is (orientation,strand_type)

- `poke(self, strand1, strand2, parallel=-1, over=1, print_pd=True):`

  This method creates a "poke" (an $R2$ move). This assumes that the two strands are vertical and that the left strand is *strand1*, which goes up. To determine this, rotate the picture, and then figure out the remaining arguments. The *parallel* argument should be 1 if the strands have the same orientation and −1 otherwise. The *over* argument should be 1 if we want *strand1* to poke over *strand2* and −1 otherwise. The numbering of the strands goes as follows: outgoing parts of strand1 and strand2 remain the same. We then number down the middle of strand1 and the middle of strand2, and finally the last part of strand1 and the last part of strand2, in that order.

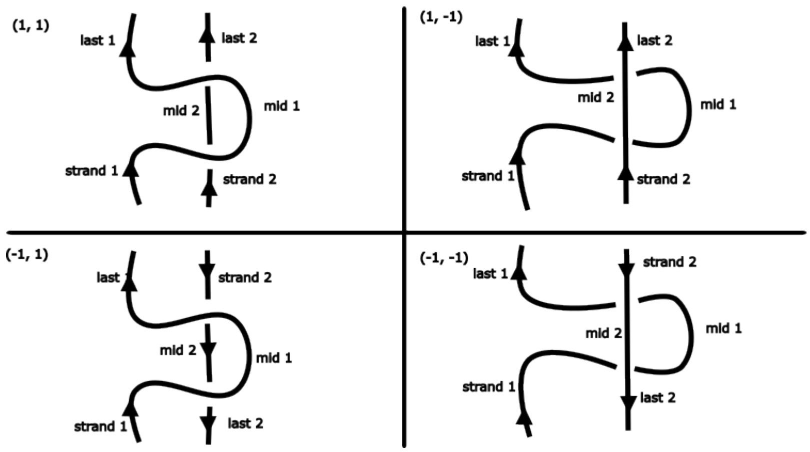
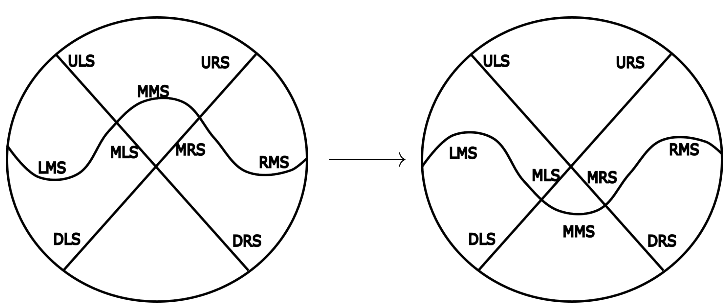


Figure 18: R2 Move Labeling Convention; tuple on upper left is (parallel,over)

- `slide(self, left_moving_strand, orientation_moving_strand, orientation_left_strand, orientation_right_strand, print_pd = True):`

  This method does an $R3$ move. It assumes that the strand we want to pass over the crossing is above the crossing, rotates the picture so that it is the case and then provides the remaining arguments. The *left_ moving_ strand* argument (LMS in the picture) is the label of the left part of the moving strand in the planar diagram notation of the last link. The argument *orientation_ moving_ strand* should be 1 if the moving strand goes from left to right, and −1 otherwise. The argument *orientation_ left_ strand* should be 1 if the bottom-left strand (DLS in the picture) goes up, and −1 otherwise. The argument *orientation_ right_ strand* should be 1 if the bottom-right strand (DRS in the picture) goes up, and −1 otherwise. The change in planar diagram notation is denoted naturally below, given by a local reflection around the crossing we pass over.



(7)

- `untwist(self, loop_label, print_pd = True):`

  This method is the untwist part of $R1$. The *loop_ label* argument is the planar diagram label of the loop in the twist. It removes the twist and the remaining strand's label is the one of the strand incoming to the twist.

- `unpoke(self, strand1, strand2, print_pd = True):`

  This method is the unpoke part of $R2$. The arguments *strand1* and *strand2* follow the same convention as in the poke method. The labeling keeps *strand1* and *strand2* as the labels of the two strands we get from unpoking, except if there is a collapse, in which case it keeps only the first strand.

- `saddle(self, strand1, strand2, print_pd=True):`

  This method represents a saddle move between two strands. It is required that *strand1* and *strand2* have opposite orientations since only orientation-preserving saddle moves are allowed. In the new picture, the labeling is done with the same labels (*strand1* and *strand2*) so that in both pictures, *strand1* is outgoing from the same crossing (same for *strand2*).

- `birth(self, print_pd = True):`

  This creates a twisted birth. Should the last link be the unknot, it will first twist the unknot and then birth a twisted circle. The twist we choose is always the one whose 0-resolution does not have an additional circle from its loop.

- `death(self, loop_label, print_pd=True):`

  This method kills a twisted circle. The argument *loop_ label* is the label of a loop in the twisted circle we want to kill. The method simply removes the crossing in the planar diagram notation.

To find the image of a vector v (as a SageMath object) in homological degree t of the original link, one can simply call the method

```
push(t,v)
```

This will output the image as a vector as well as the enhanced Kauffman states in the image whose coefficients are nonzero. For each such state, denoted *state*, we have that *state*[0] represents the resolutions of the crossings, *state*[1] represents the negative cirlces and *state*[2] the positive circles.

One final remark is that despite the limitation due to twisted births and deaths, the image under cobordisms of vectors will not be affected by these twists since the twist homotopy equivalence used is a strong deformation retraction in the sense of Bar-Natan [4]. It is hence enough to untwist them once the unknot component is nontrivially connected to some other part of the diagram.