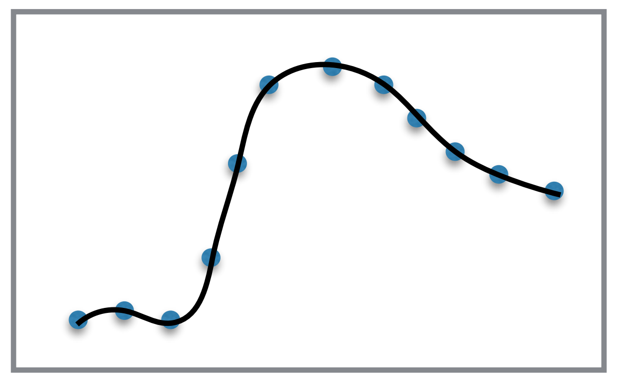
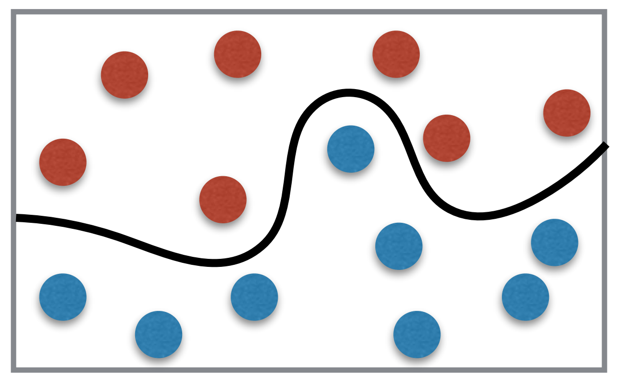
Lecture 3
RL applications to Knot Theory

FABIAN RUEHLE (NORTHEASTERN UNIVERSITY & IAIFI)

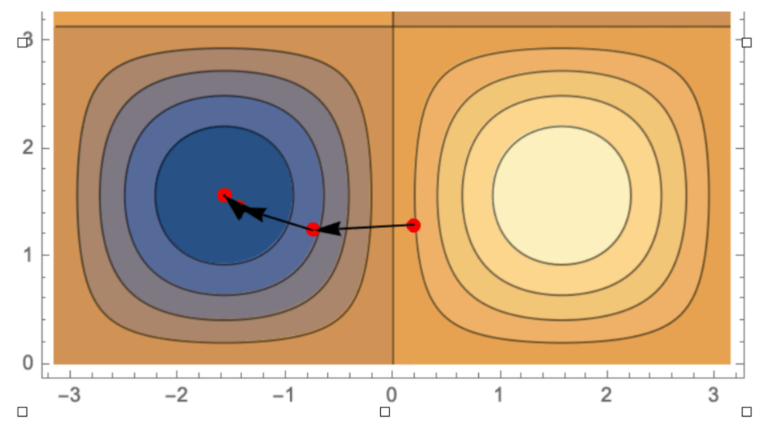
ML in Maths and Physics 2023
University of Oxford

17-21 July 2023

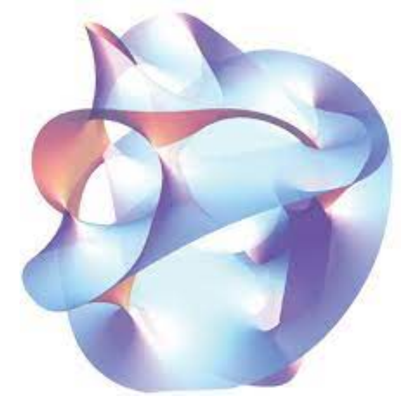
Overview

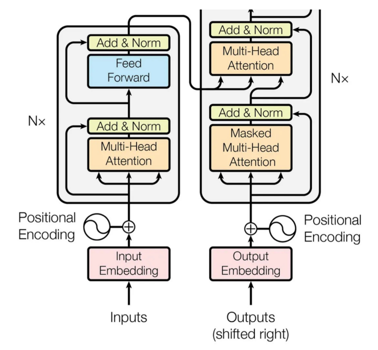
- Classification / regression:
  - Continuous inputs
  - Labels (truth values) available
  - Not interested in the NN function, only its output

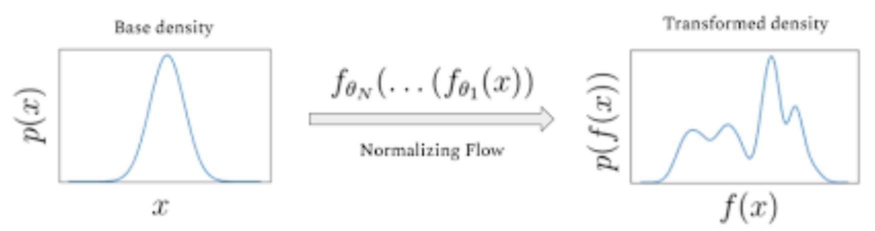


- (Continuous) optimization
  - Continuous input/output
  - Can find minimum with SGD

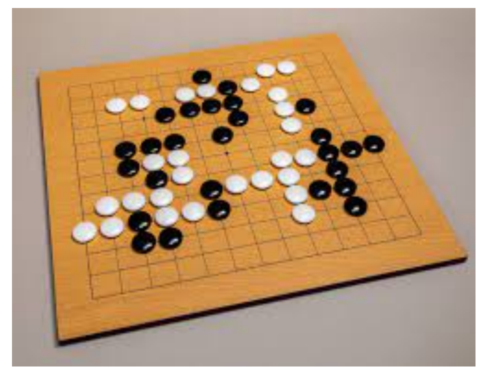


- Semi-supervised:
  - Continuous inputs
  - Labels (truth values) not available
  - Interested in the NN function itself





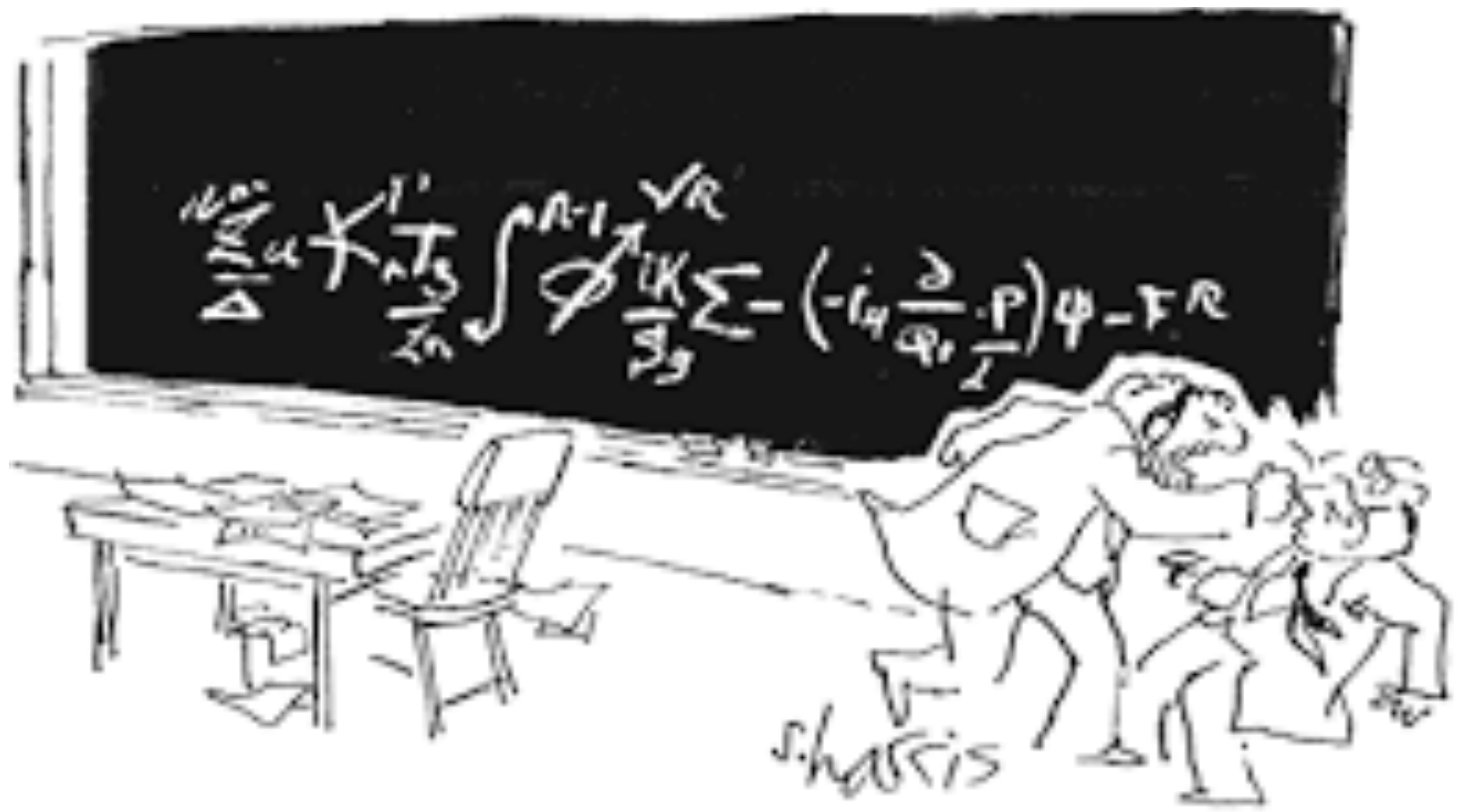
- Generative Models
  - Discrete or continuous inputs/outputs
  - Labels not available
  - Not interested in the NN function, only its output



- Discrete optimization
  - Discrete inputs (no SGD possible)
  - Labels not available
  - Not interested in the NN function, only its output

Discrete Math as RL problems

- Want provable results
- Setup RL algorithm that solves the "yes"-instances of the decision problem: "Does object O have property X"
  - The algorithm should halt on the "yes" instances and produce a truth certificate
  - The algorithm can run forever on the "no" instances

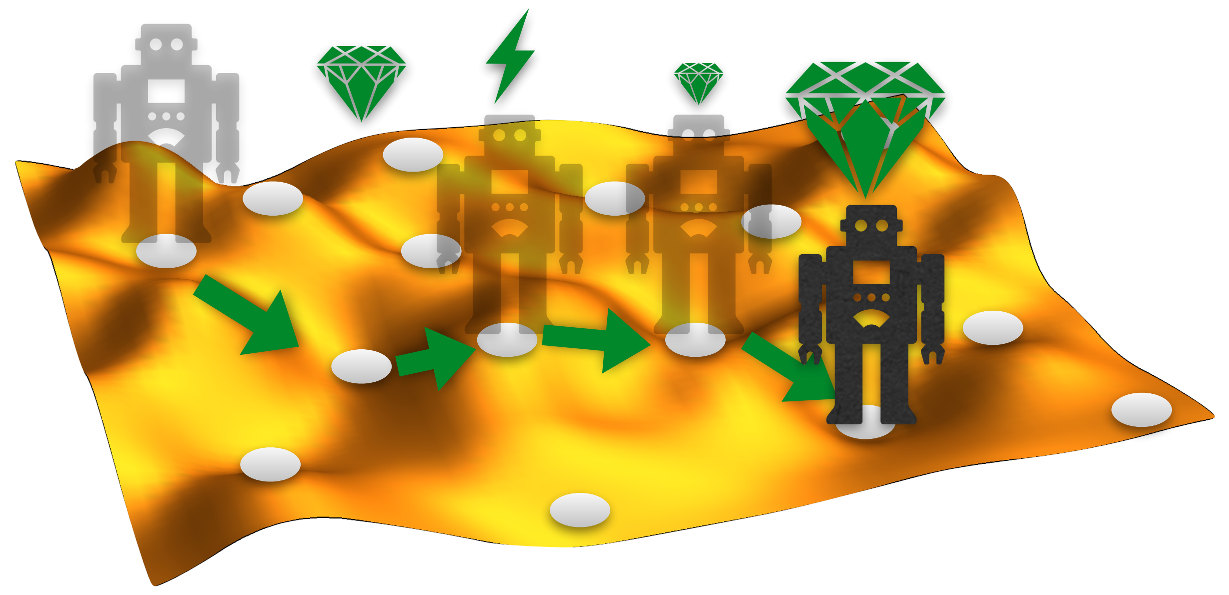


"You want proof? I'll give you proof!"

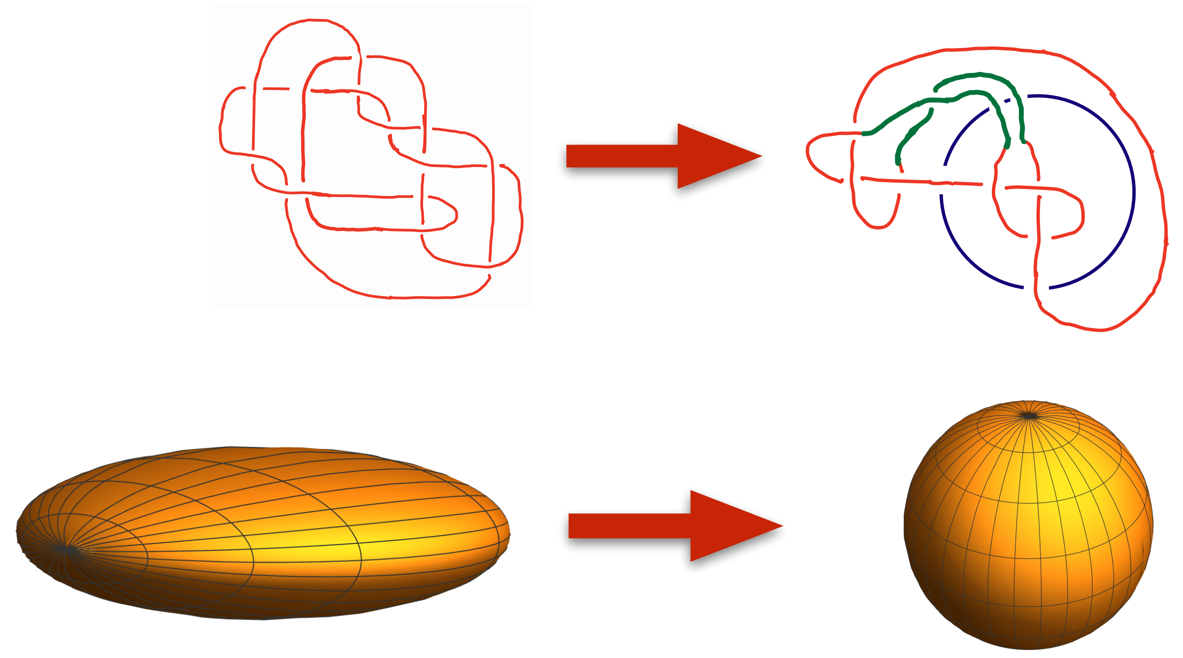
[cartoon by Sidney Harris]

Outline

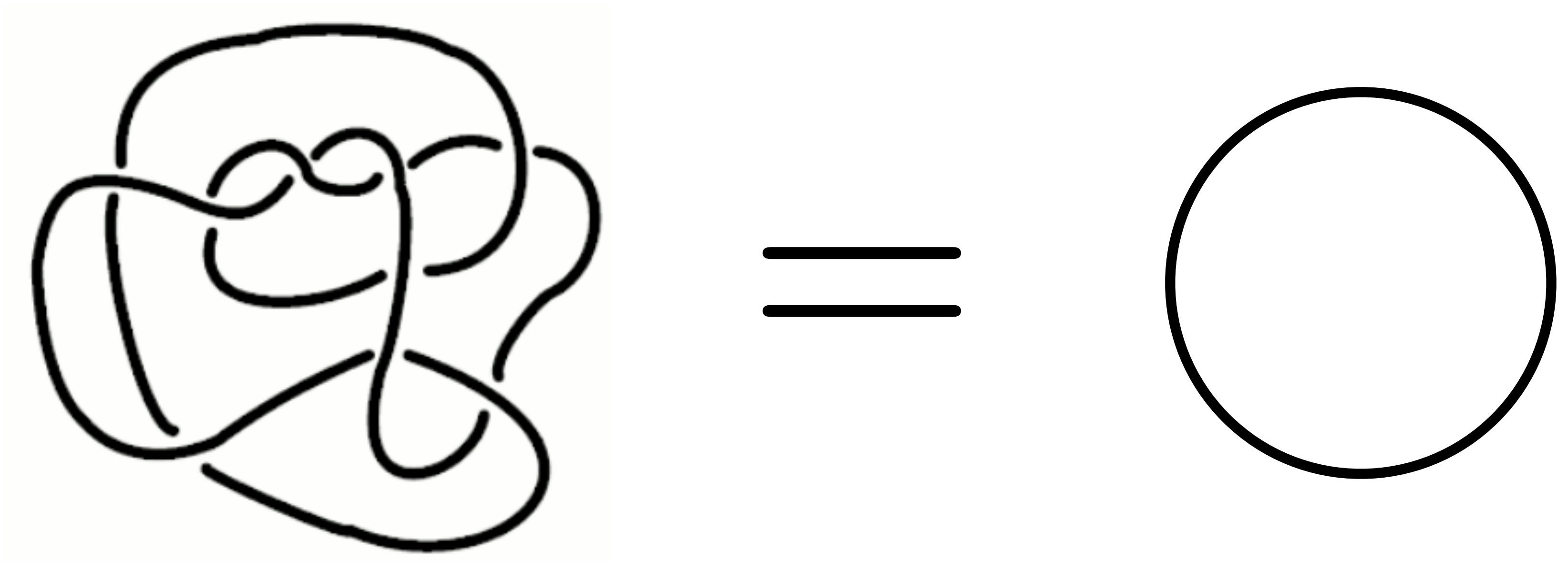
1 Knots as RL problems



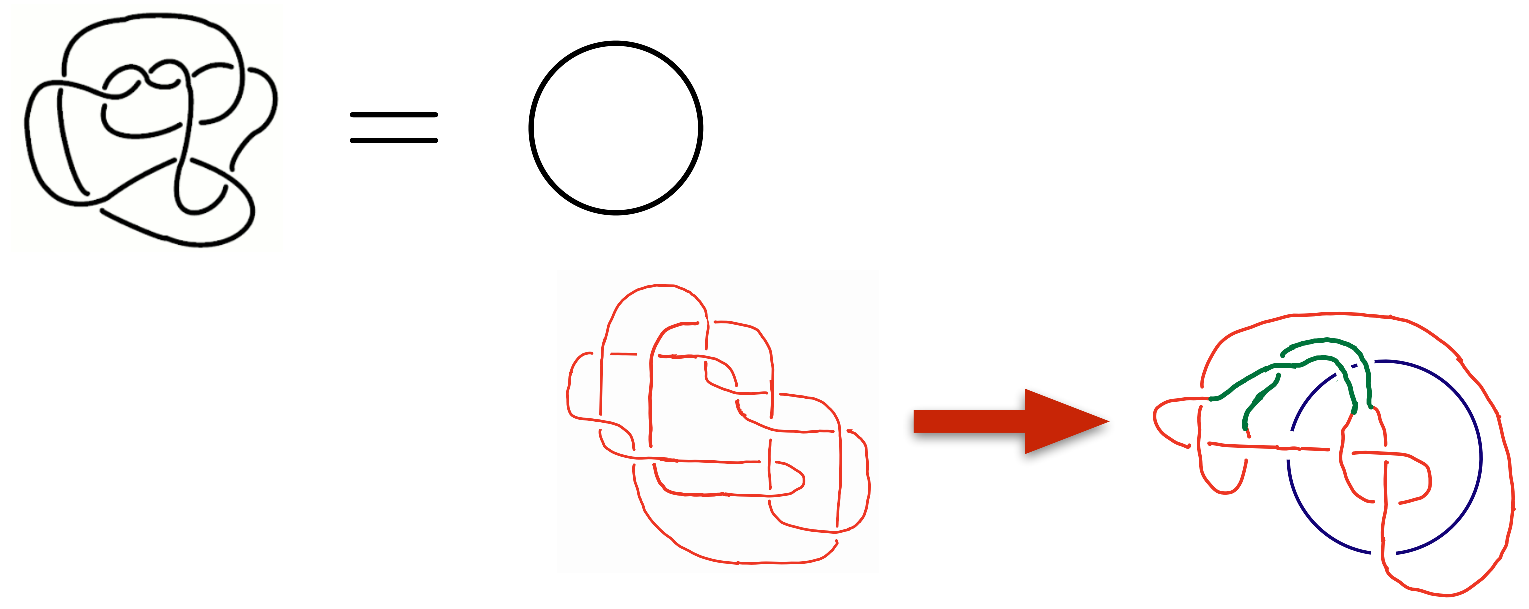
3 Ribbon knots and SPC4



2 The Unknot Decision Problem



4 Recap



Collaborators

Sergei Gukov

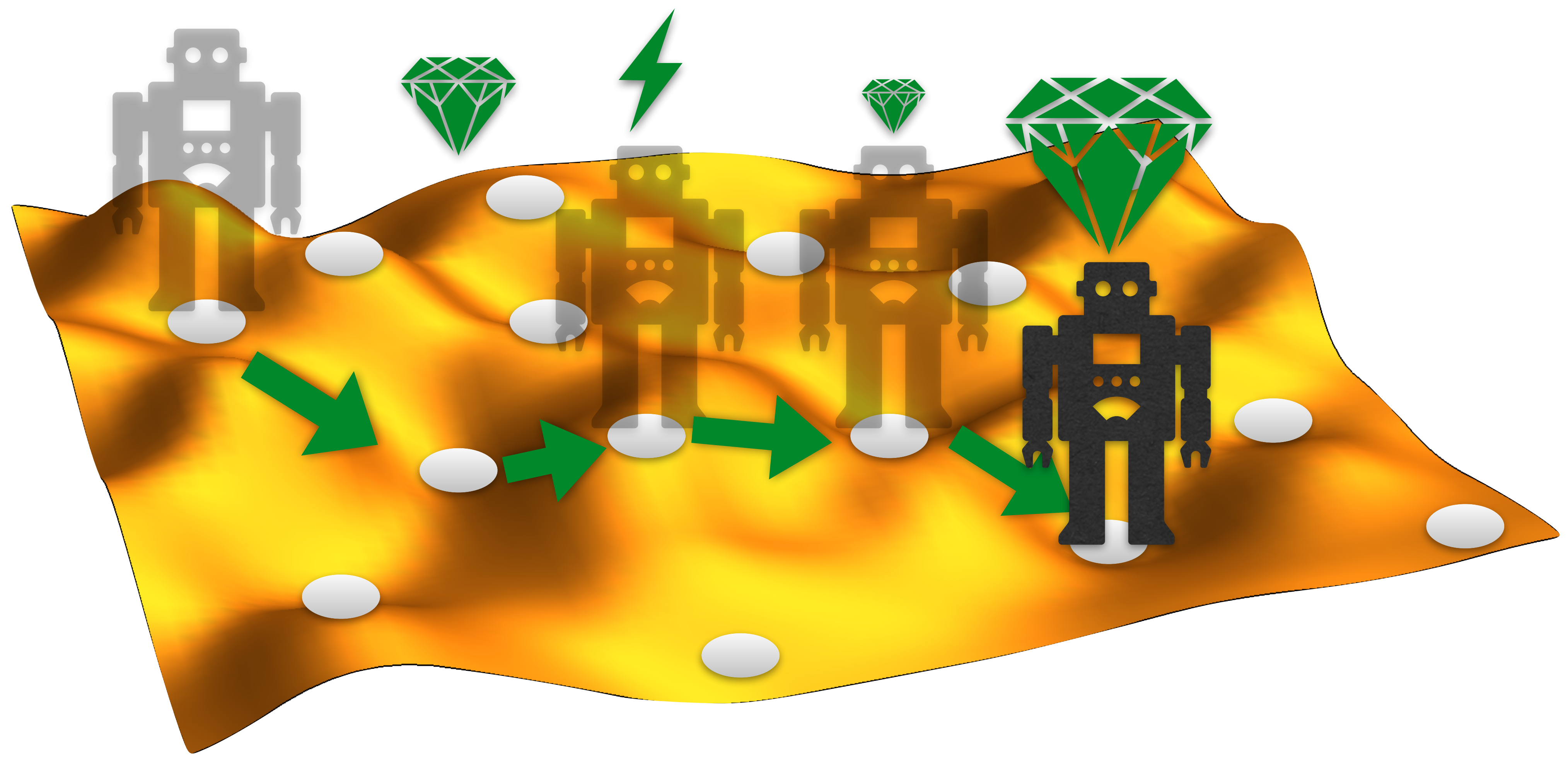
Jim Halverson

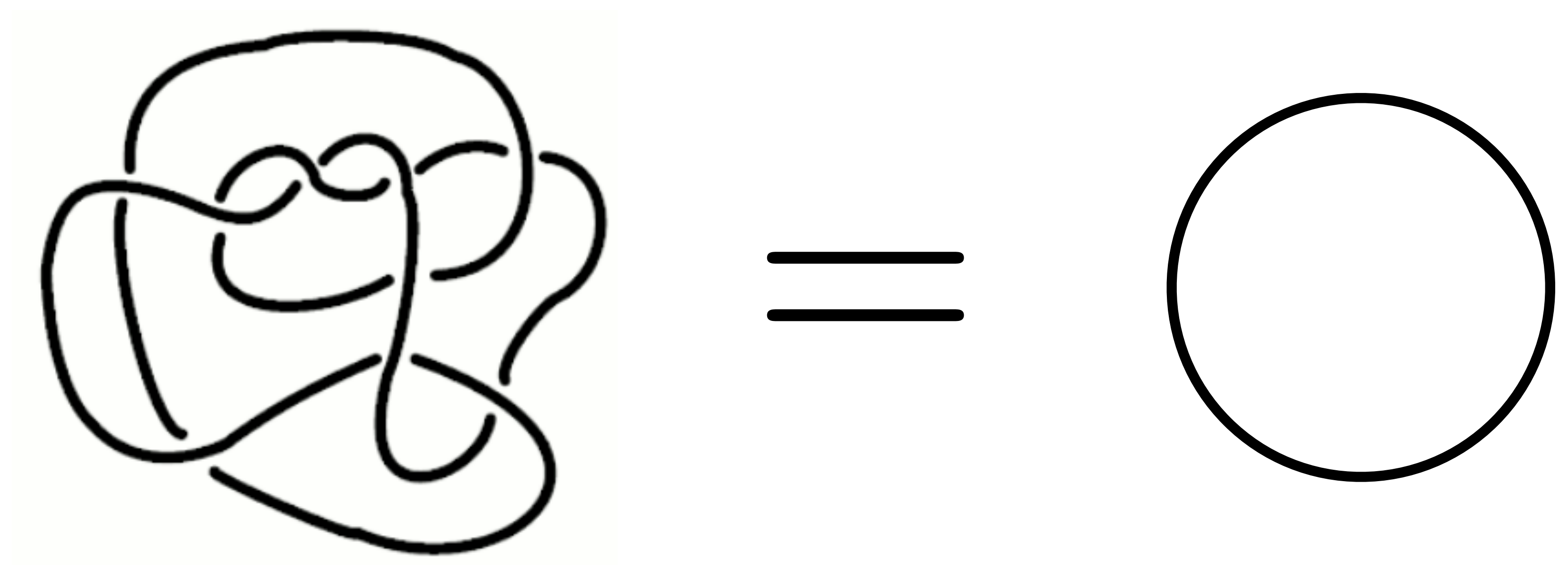
Ciprian Manolescu

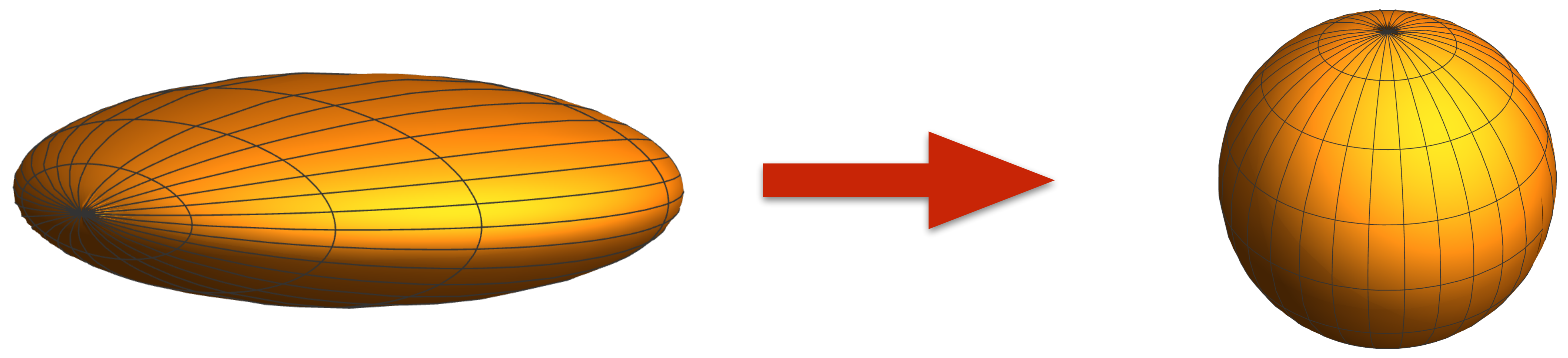
Piotr Sulkowski

[Gukov, Halverson, FR, Sulkowski: *Learning to Unknot*, arXiv: 2010.16263]

[Gukov, Halverson, Manolescu, FR: *Searching for ribbons with machine learning*, arXiv: 2304.09304]



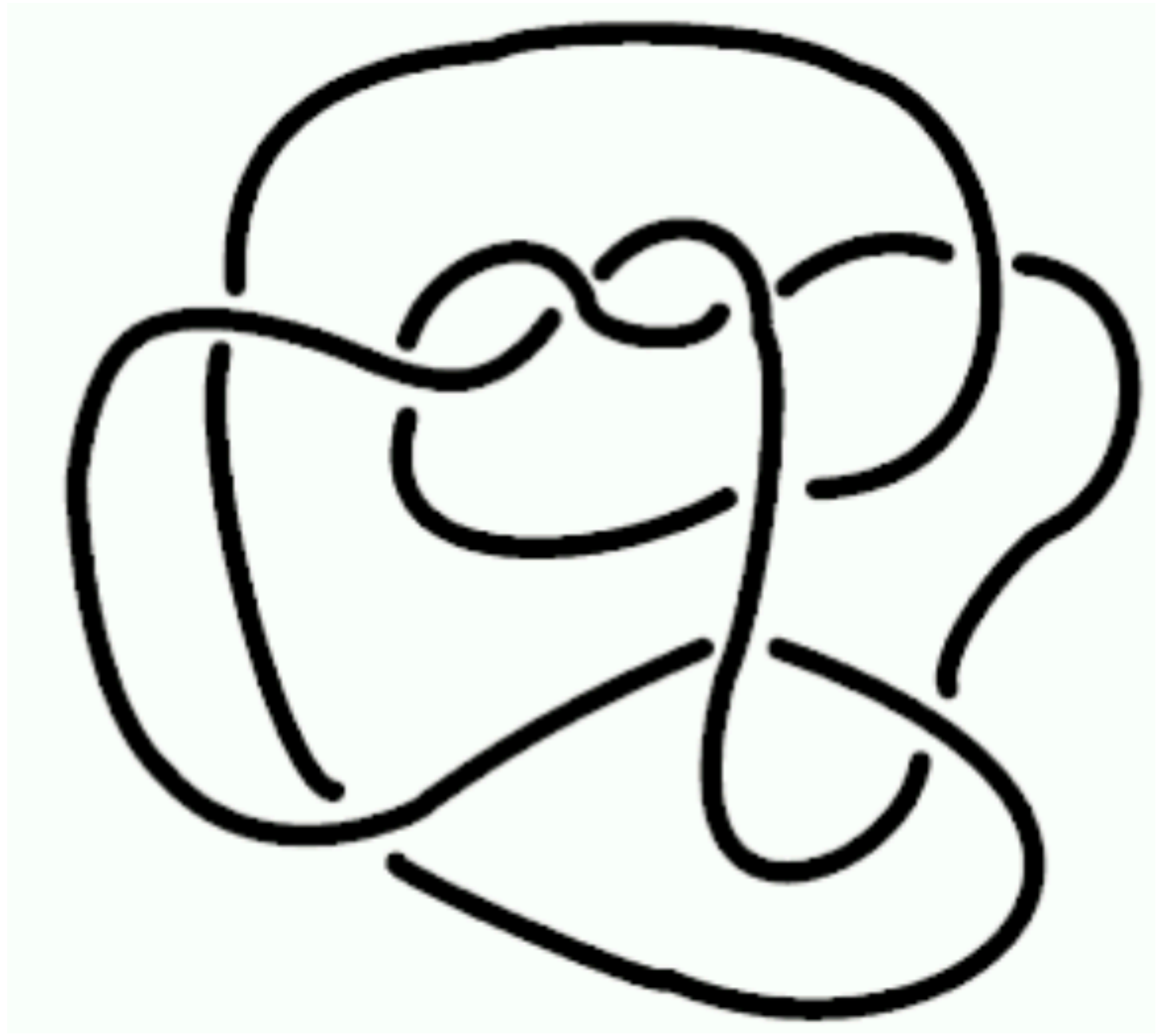




Knots as RL problems

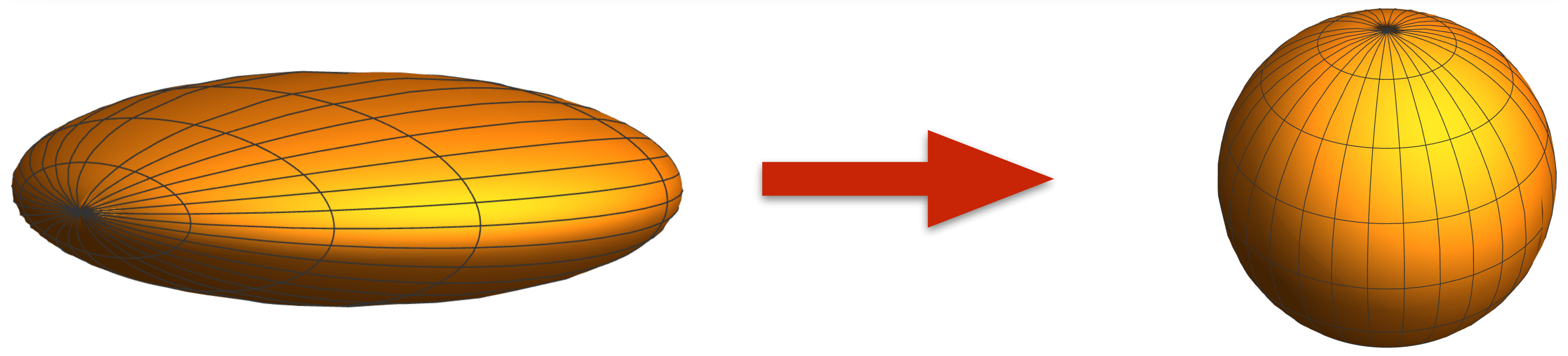
Unknot Decision Problem

Unknot decision problem
Is a given knot a representation of the unknot



[image: wikipedia]

Smooth Poincare Conjecture 4D
Can anything that looks like a sphere be cont. deformed into a sphere in 4D?



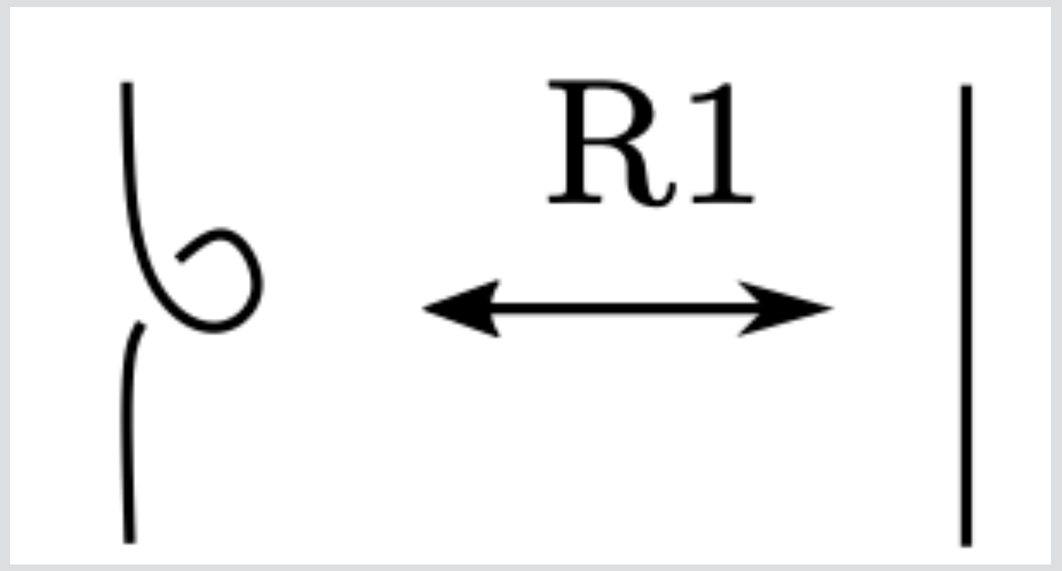
How is this an RL problem?
Looks like a cont problem

What if all knots are unknots?
Do non-trivial knots even exist?

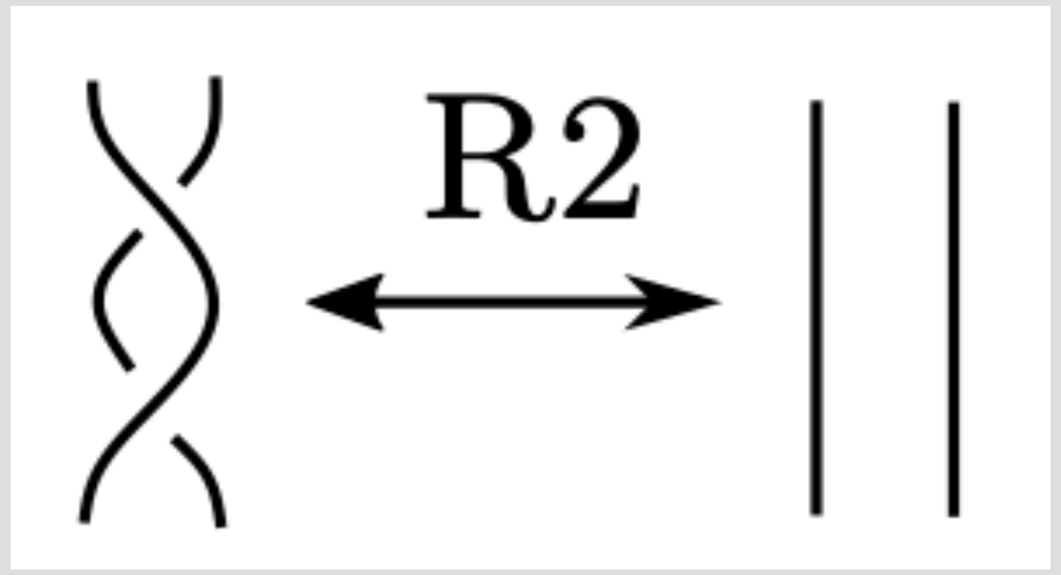
Reidemeister Moves

- Reidemeister proved that non-trivial knots exist by proving that all equivalent knots can be related to one another by a sequence of three moves

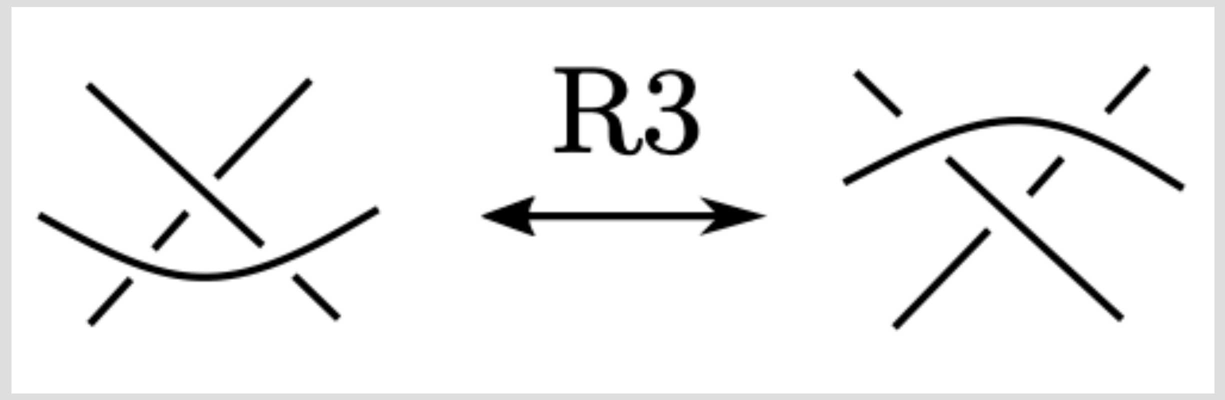
Twist/Untwist

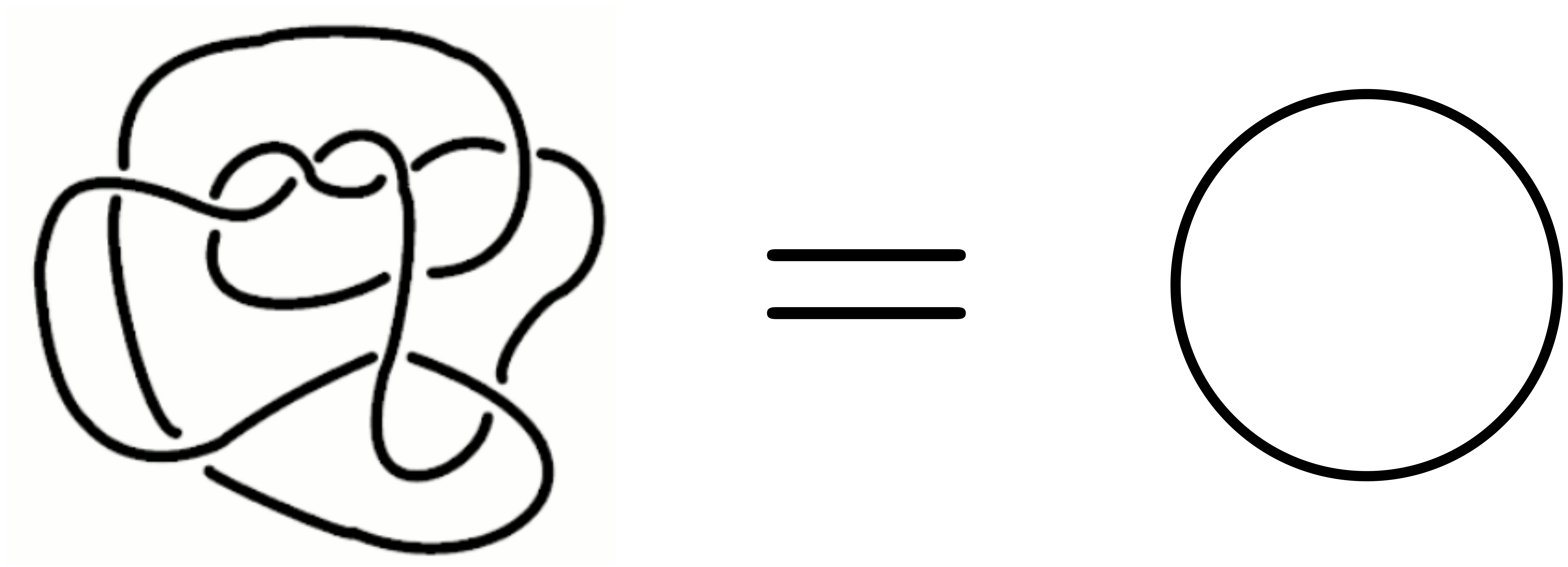


Poke/Unpoke



Slide



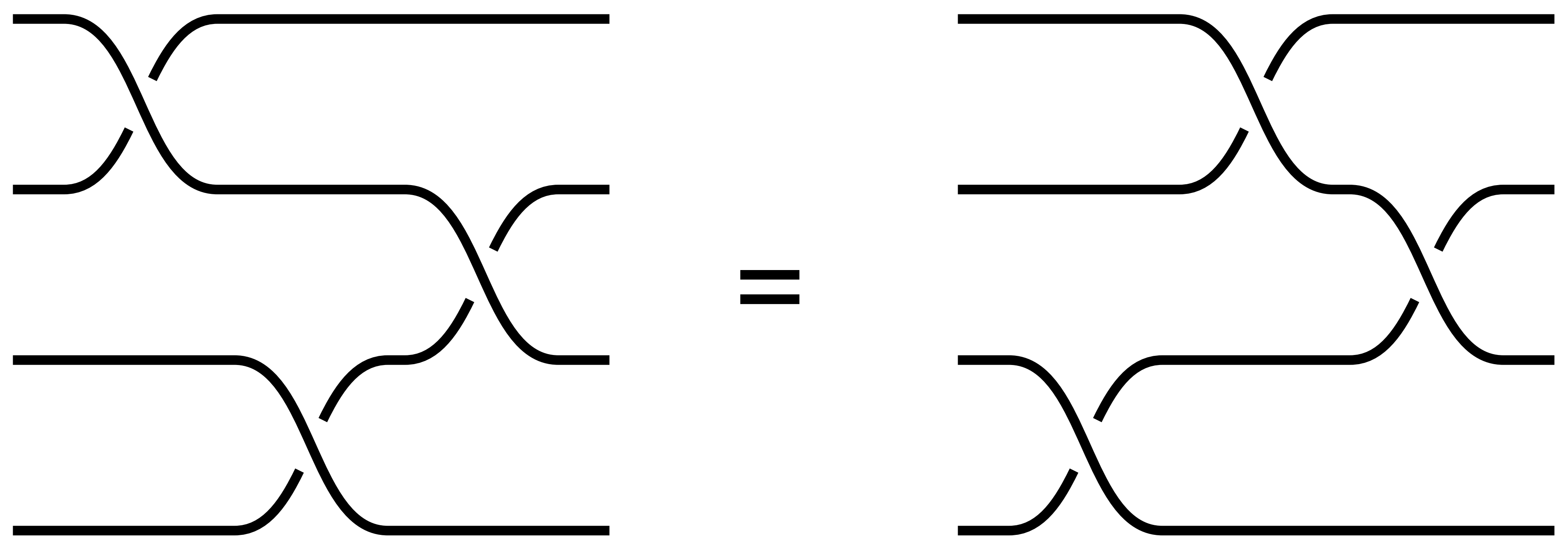


Unknot Decision Problem

Braids and Knots

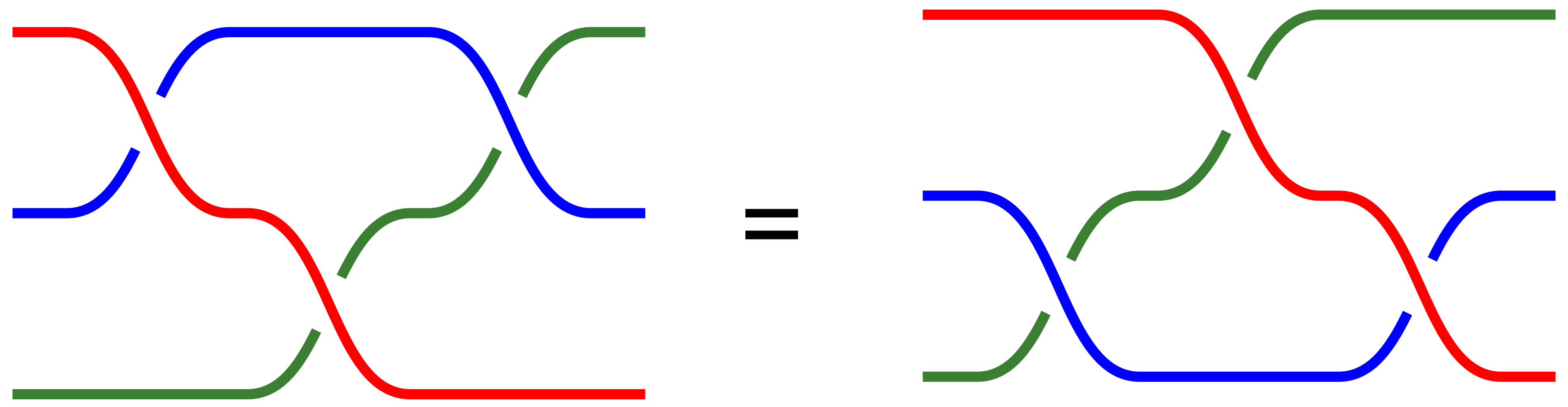
Definition (Braid group B_n): Non-abelian group with n generators σ_i

- Braid relation 1:



$\sigma_1 \ \sigma_3 \ \sigma_2$ = $\sigma_3 \ \sigma_1 \ \sigma_2$

- Braid relation 2:



$\sigma_1 \ \sigma_2 \ \sigma_1$ = $\sigma_2 \ \sigma_1 \ \sigma_2$

- Some braid generators commute:

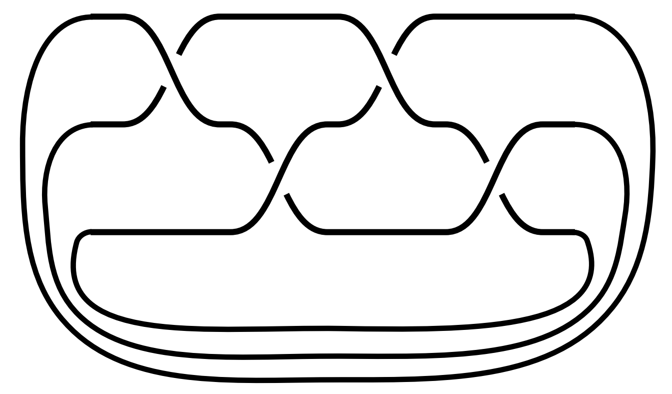
$$\sigma_i\sigma_j = \sigma_j\sigma_i \quad \text{if} \quad |i-j| > 1$$

- Can rearrange the braid:

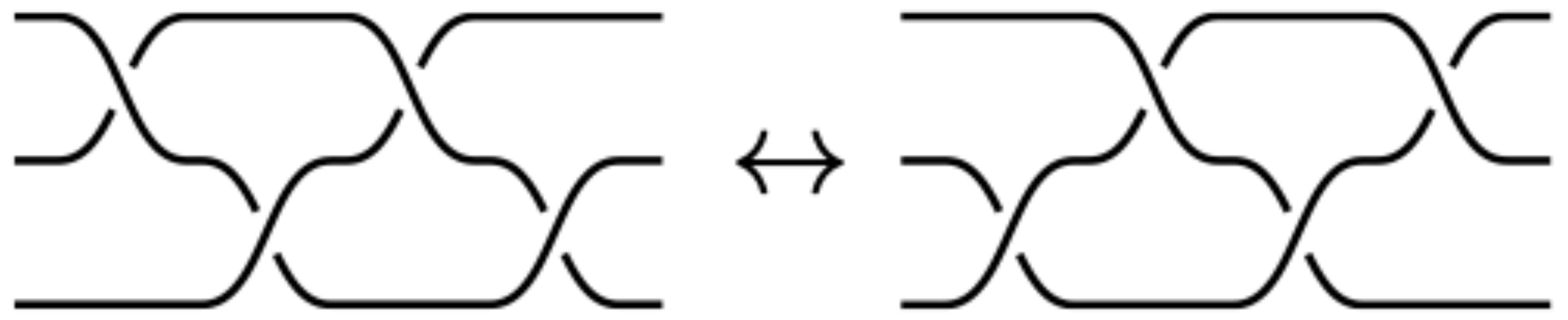
$$\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}$$

Moves on knots

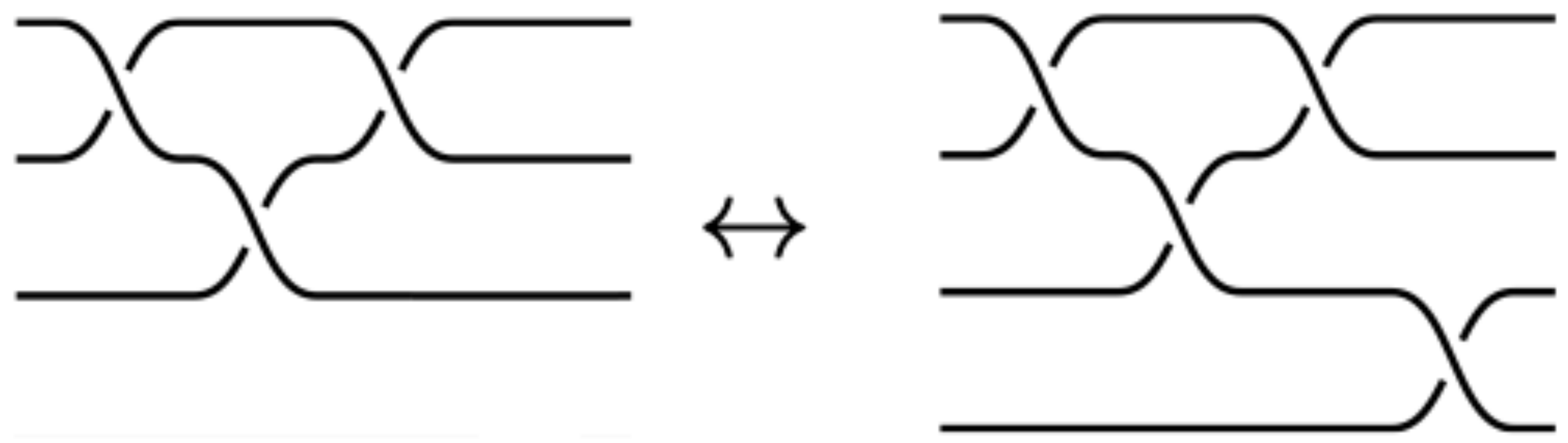
- Every knot can be described as the closure of a braid



- On braids, Reidemeister moves translate to 2 Markov moves plus braid



(a) Move 1 (conj.): $\sigma_1\sigma_2^{-1}\sigma_1\sigma_2^{-1} \leftrightarrow \sigma_2^{-1}\sigma_1\sigma_2^{-1}\sigma_1$



(b) Move 2 (stabilization):

- Simplifying a knot then becomes a shortest word problem in group theory
- The word problem is in general undecidable, but it is decidable for braids
- For knots, the complexity is at least as hard as the unknot decision problem

Find the unknot with RL

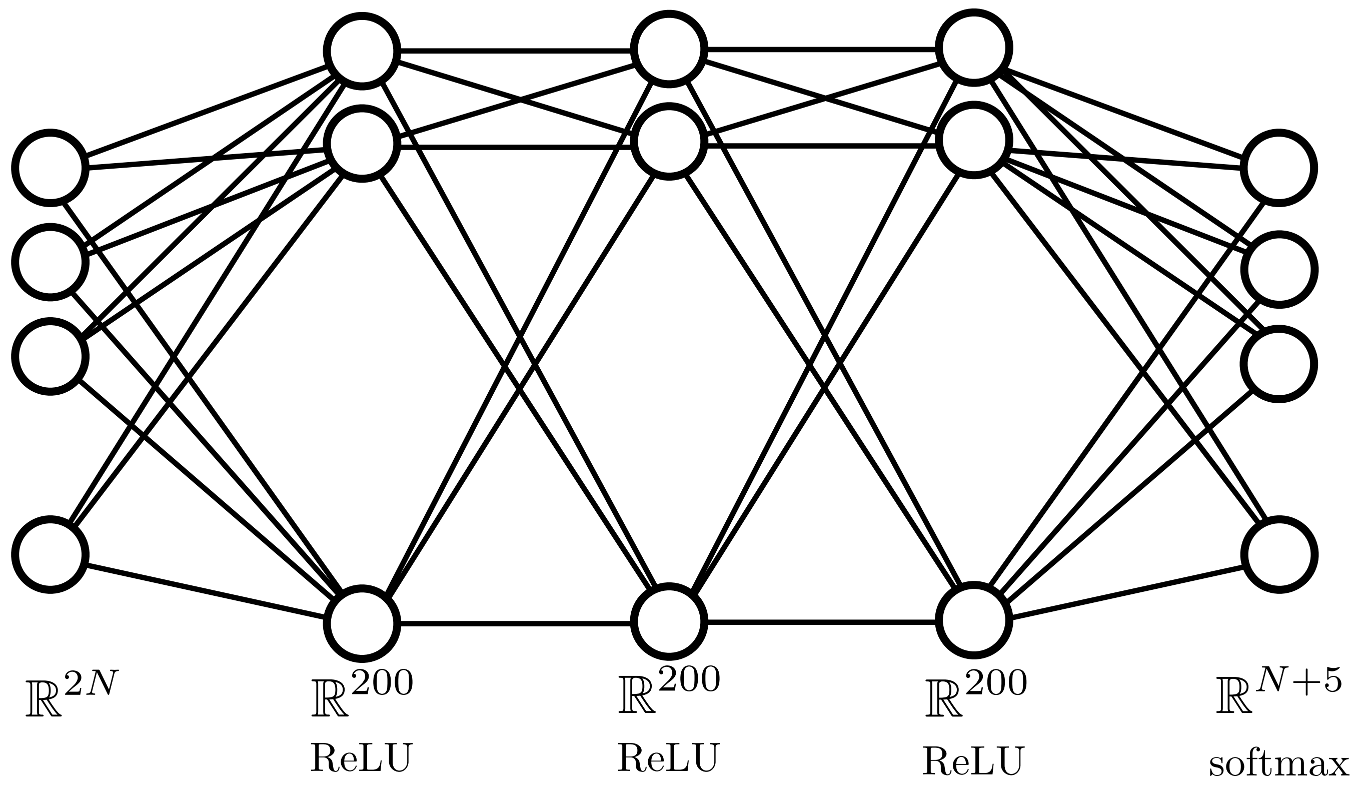
- **Actions:** By Markov's theorem, two closed braids that represent the same link can be transformed into each other using [Markov `36]
  - Braid relations ($\sigma_i \sigma_j = \sigma_j \sigma_i$ if $|i-j|>1$, $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$)
  - Markov moves (Conjugation, Stabilization)
  - In practice, the number of actions grows with N = length of braid word
    - ✦ Need to specify the position where to apply braid relation 1 $\Rightarrow$ N moves
    - ✦ Need to specify the position where to apply braid relation 2 $\Rightarrow$ N moves
    - ✦ Need to specify the element with which to conjugate $\Rightarrow$ N moves
  - Conjugation can and stabilization does change the length of the braid word $\Rightarrow$ The length of the input to the NN varies $\Rightarrow$ zero-pad braids to length $2N$

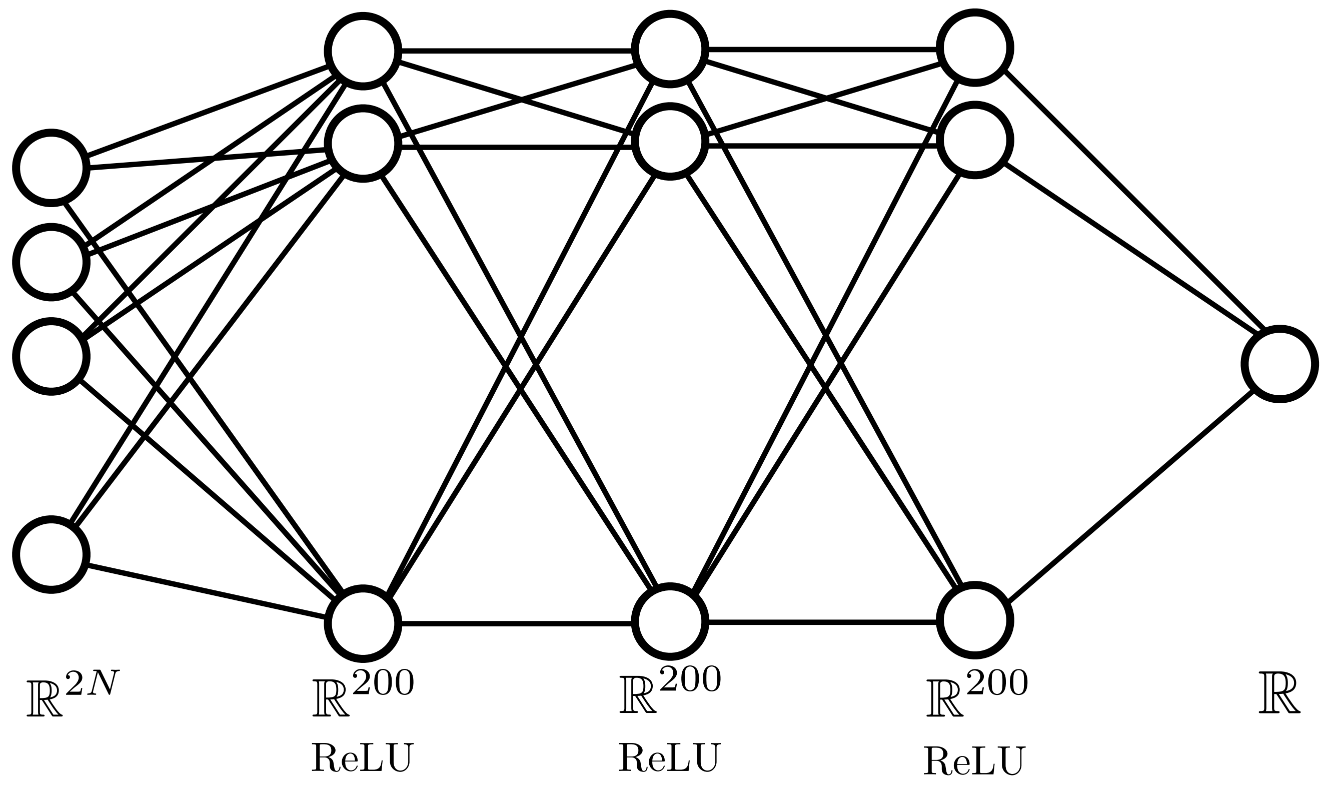
Find the unknot with RL

- **Better set of actions:** use high-level actions composed of
  - Conjugation (only conjugate with elements at the end of the braid word) $\Rightarrow$ 2 moves (shift generators left/right)
  - Stabilization (only stabilize/destabilize on the right) $\Rightarrow$ 1 move
  - Apply braid relations at first possible occurrence and shift this occurrence to the end via conjugation
  - Bundle destabilization + removing inverses in the braid into one action
- **States:** zero-padded braid words of length $2N$

Find the unknot with RL

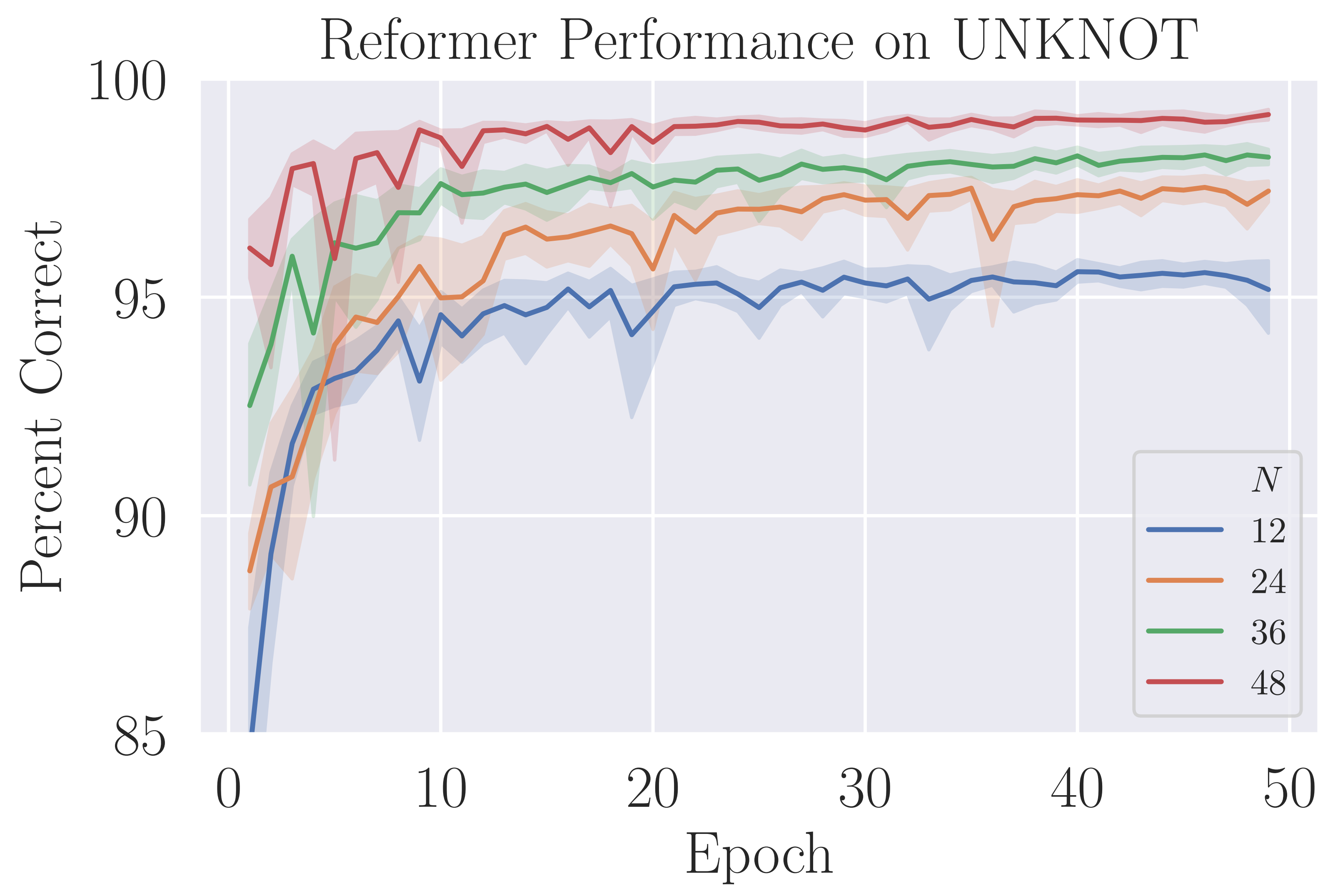
- **Reward:** Want to get to the shortest braid word as quickly as possible. “Reward” agent with negative braid length (i.e. agent is only punished)
- **Policy and state value function:** Use 2 neural nets with architectures

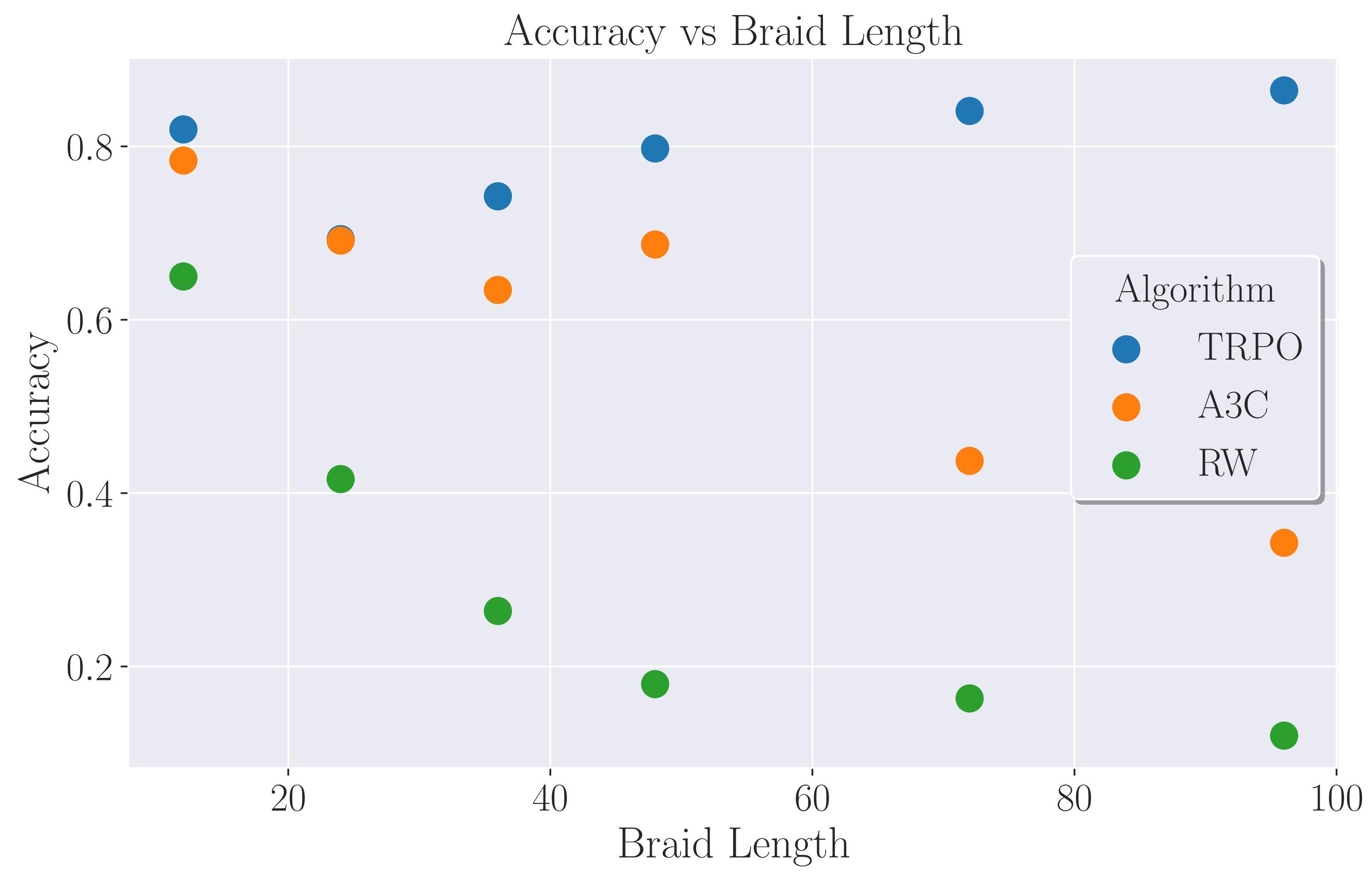


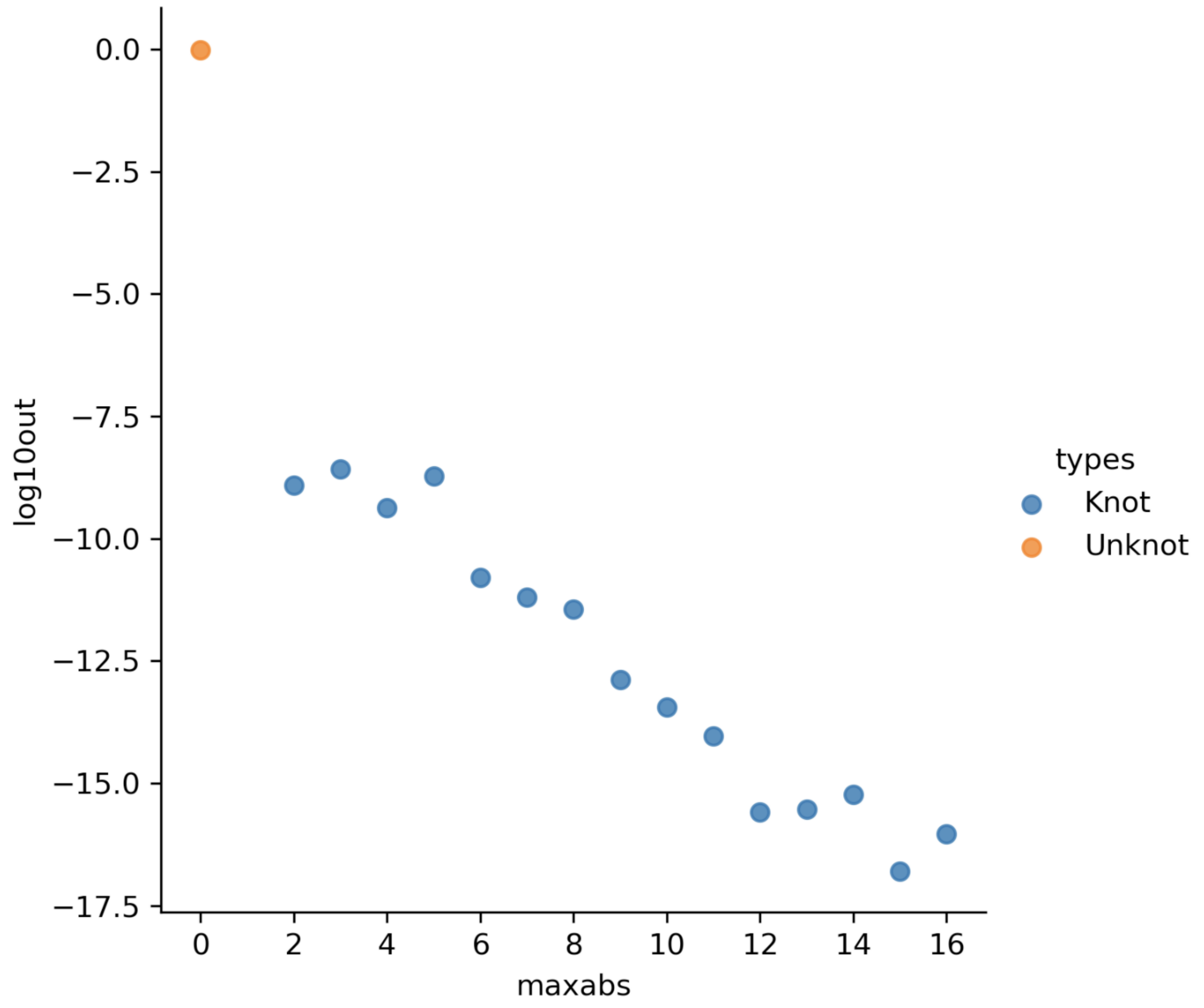


- **Terminal state:**
  - End if the braid word is empty (in this case the knot is the unknot)
  - End after 300 actions (if the braid cannot be simplified further; the agent will perform idle actions that do not change the braid length, e.g. shift left/right)

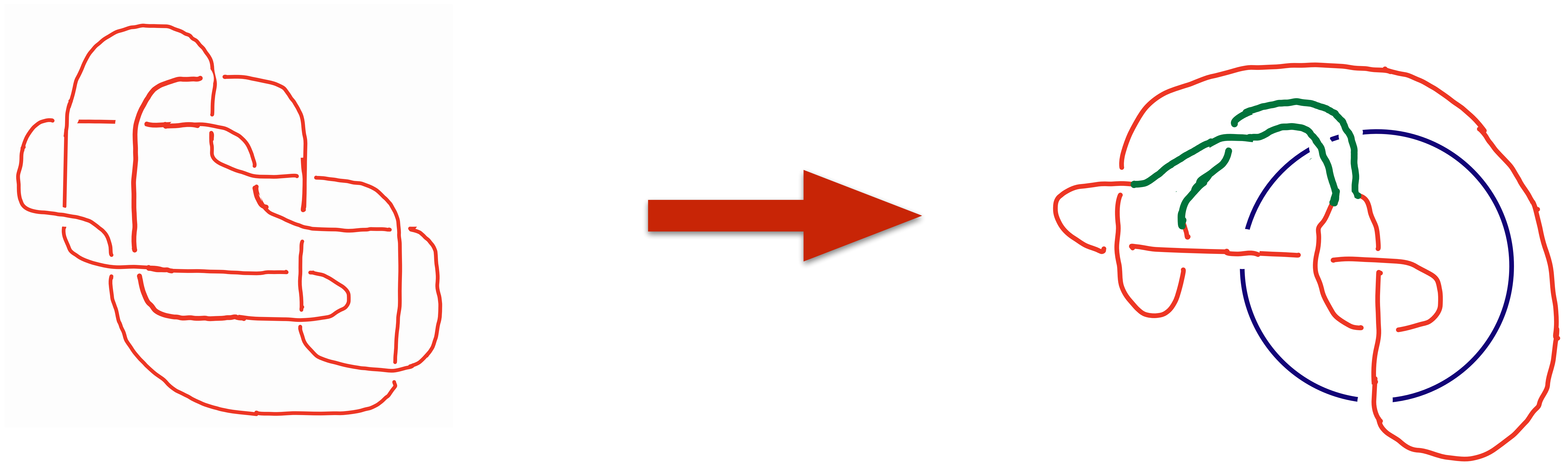
Find the unknot with RL

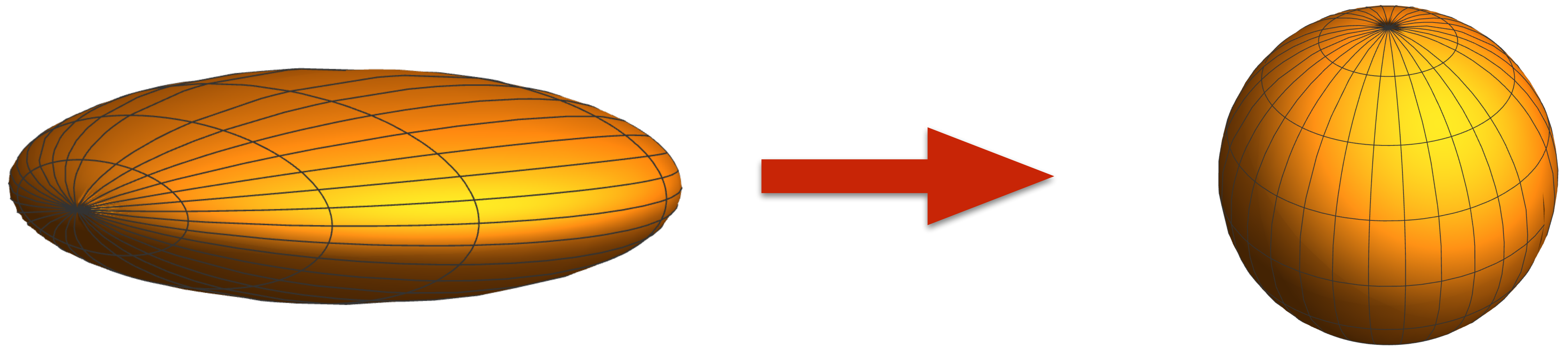






[Gukov, Halverson, FR, Sulkowski `20]





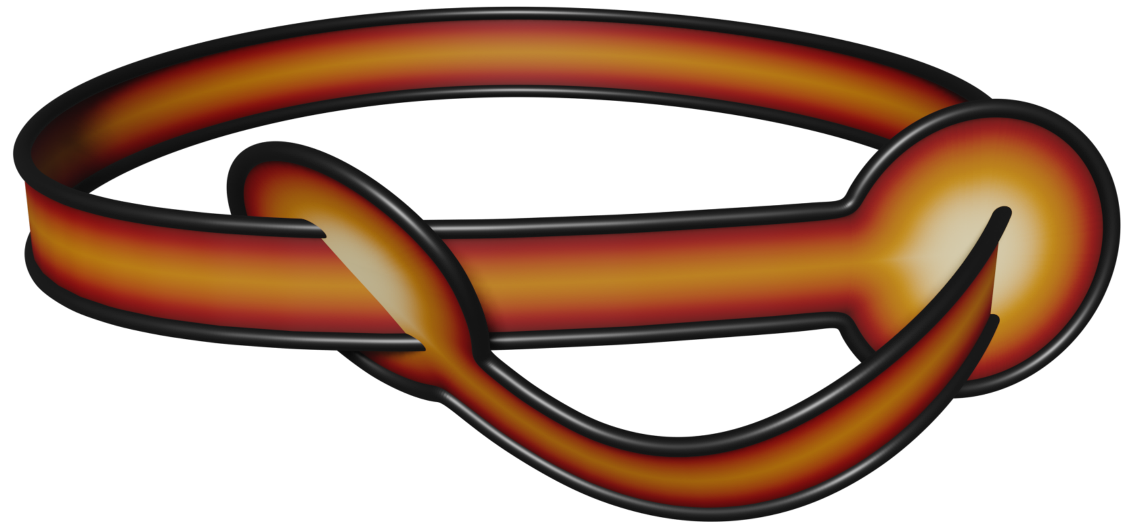
Ribbon Knots and SPC 4

Sliceness and Ribbonness

- We are interested in the slice decision problem

(Smooth / topological) slice decision problem
Does a knot $K \subset S^3$ bound a (smooth/locally flat) disk in the 4-ball

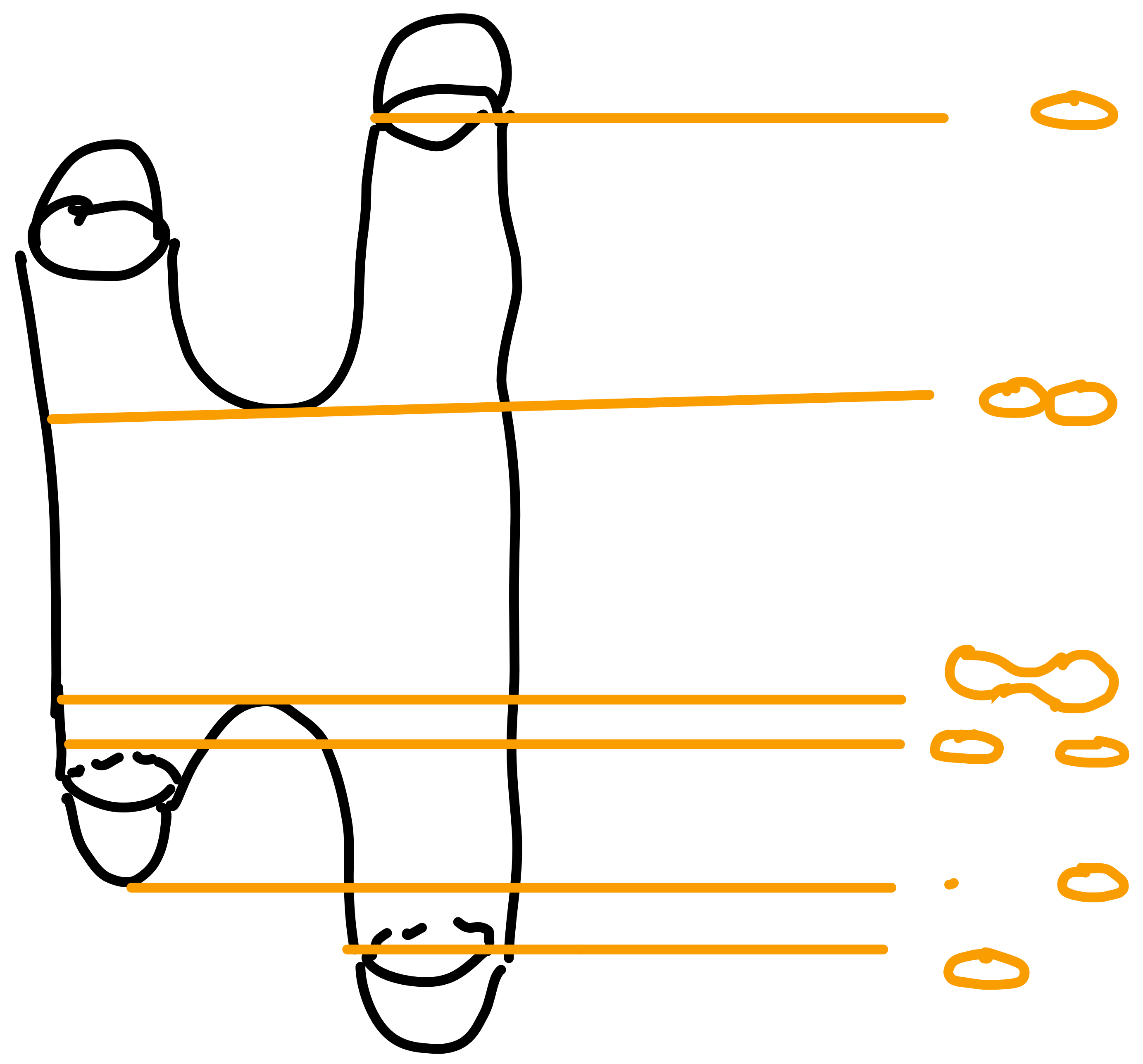
- This is related to longstanding open conjectures:
  - Slice-Ribbon-Conjecture: Is every (smooth) slice knot a ribbon knot?
  - SPC4: Is every 4-sphere diffeomorphic to the standard 4-sphere?
- If a knot is slice, it is concordant to the unknot



Rephrasing the problem for RL

- How do we relate the question of whether a knot bounds a disk in a 4-ball to RL?

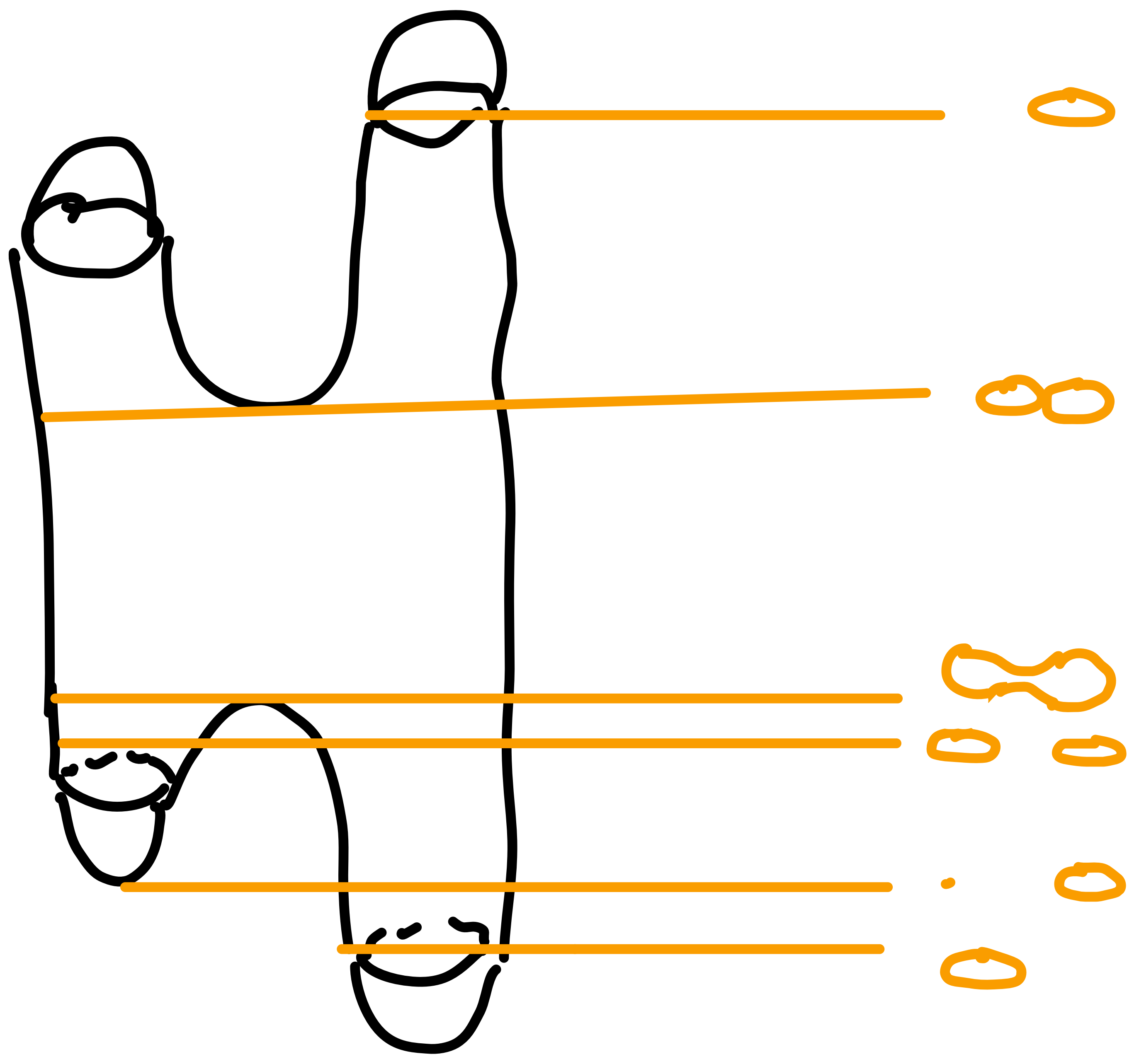
Construct link concordance (by adding bands) until one obtains the unknot

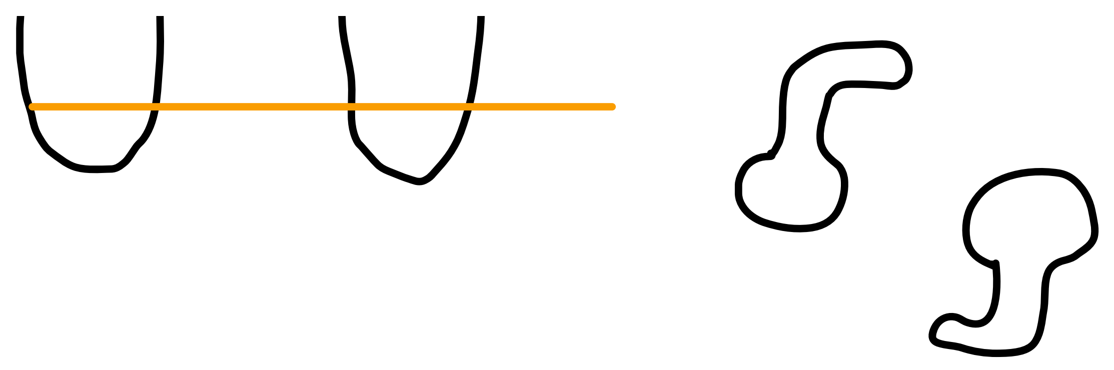


Rephrasing the problem for RL

- How do we relate the question of whether a knot bounds a disk in a 4-ball to RL?

Construct link concordance (by adding bands) until one obtains the unknot

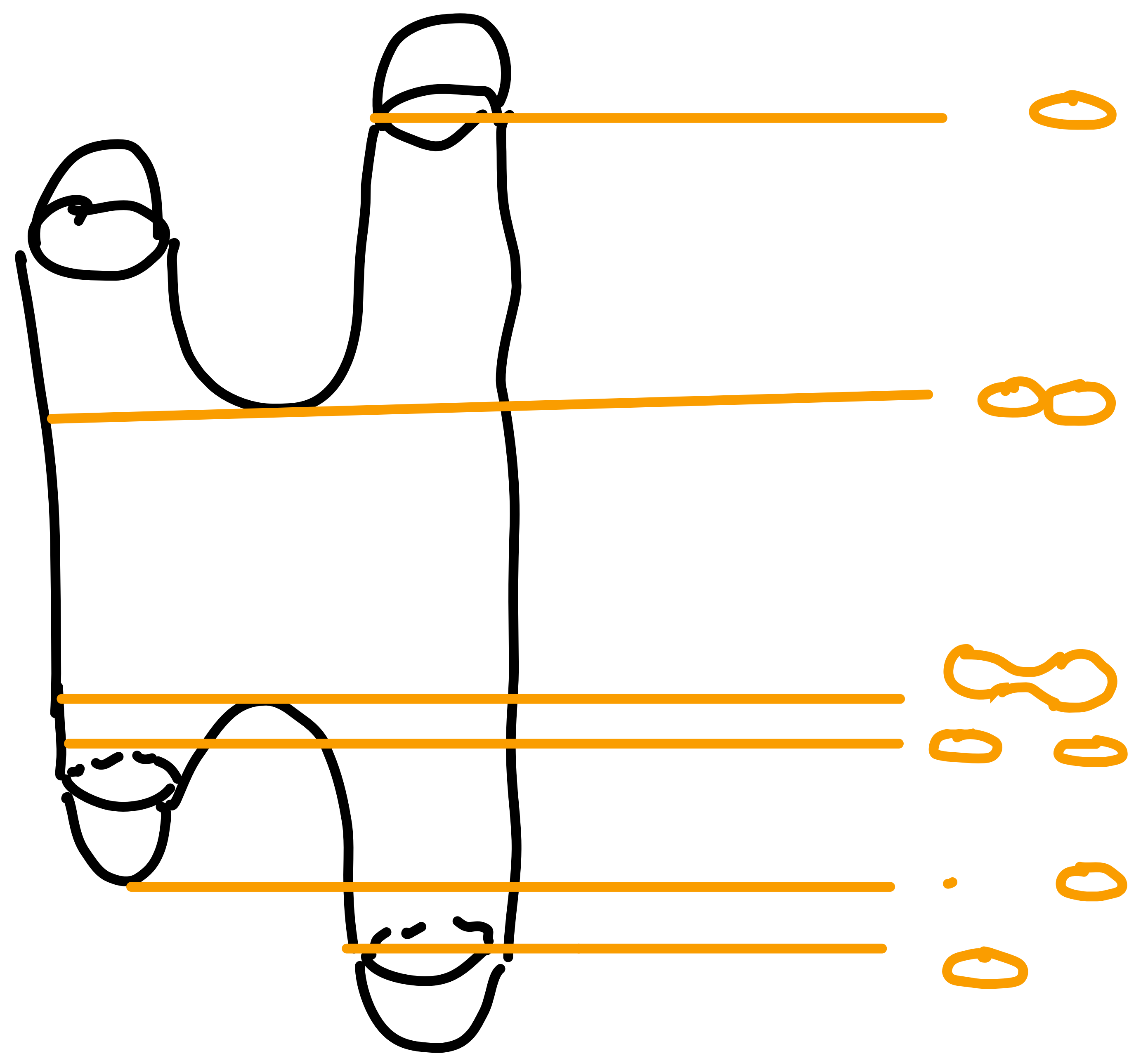


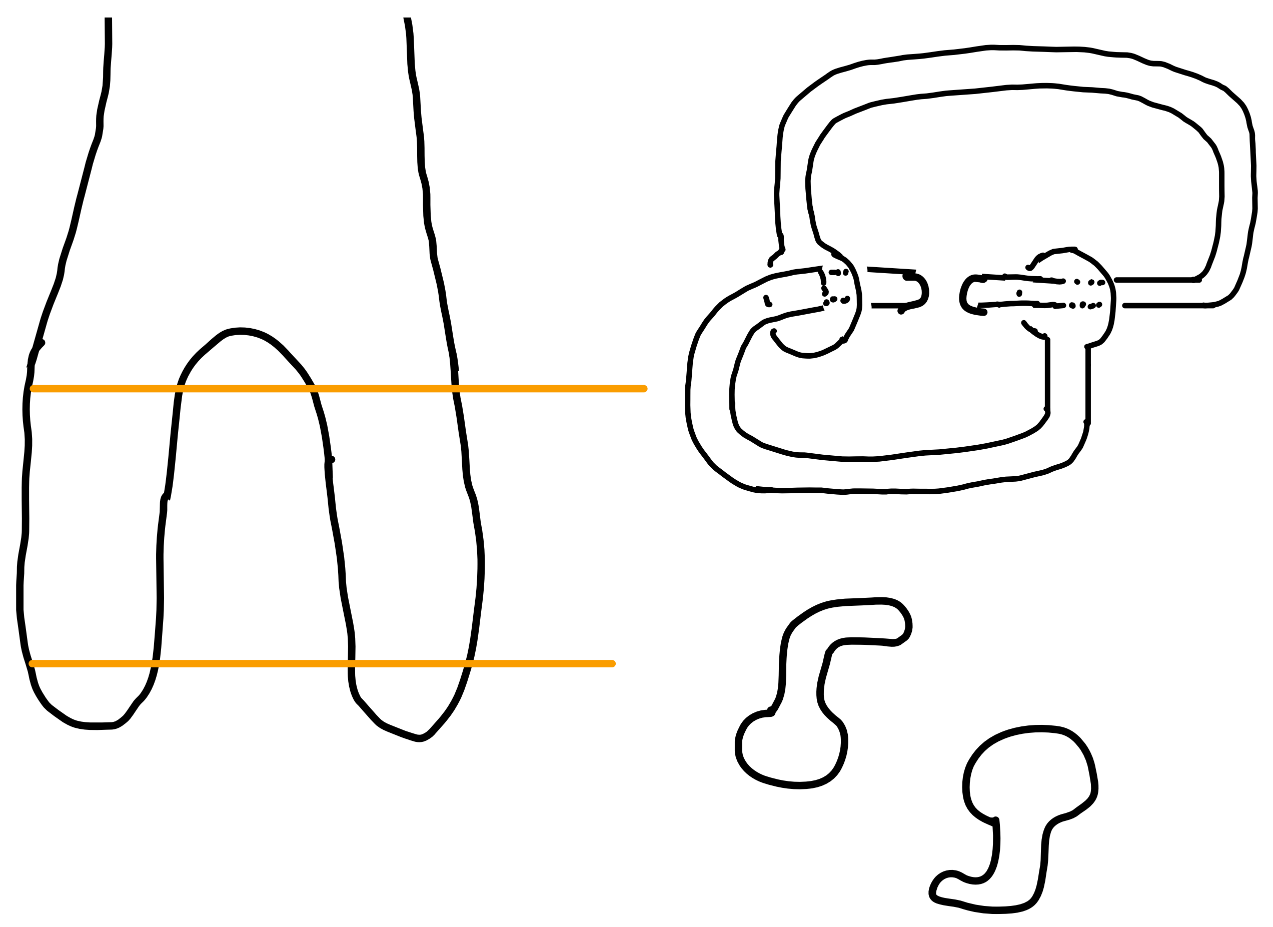


Rephrasing the problem for RL

- How do we relate the question of whether a knot bounds a disk in a 4-ball to RL?

Construct link concordance (by adding bands) until one obtains the unknot

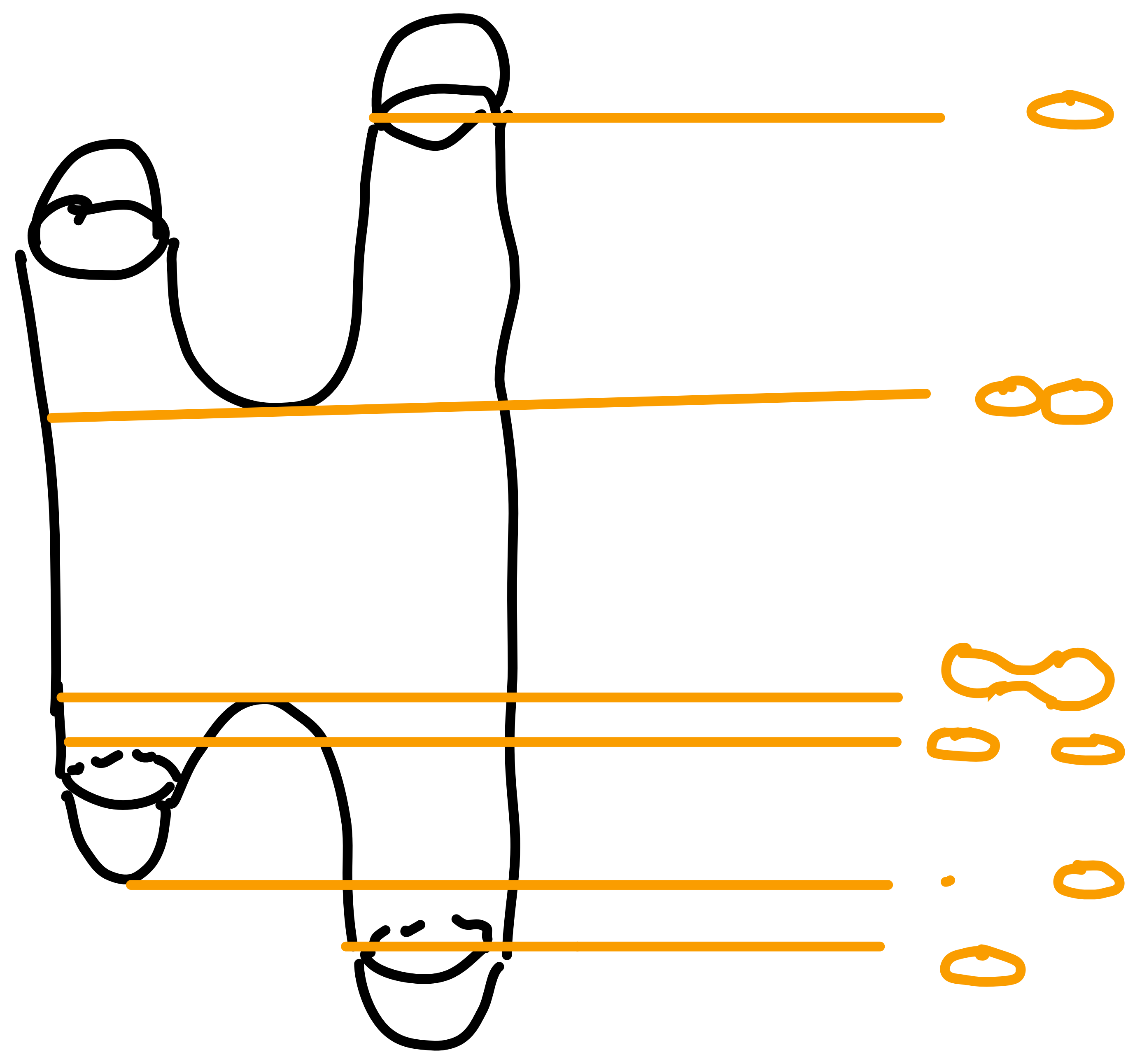


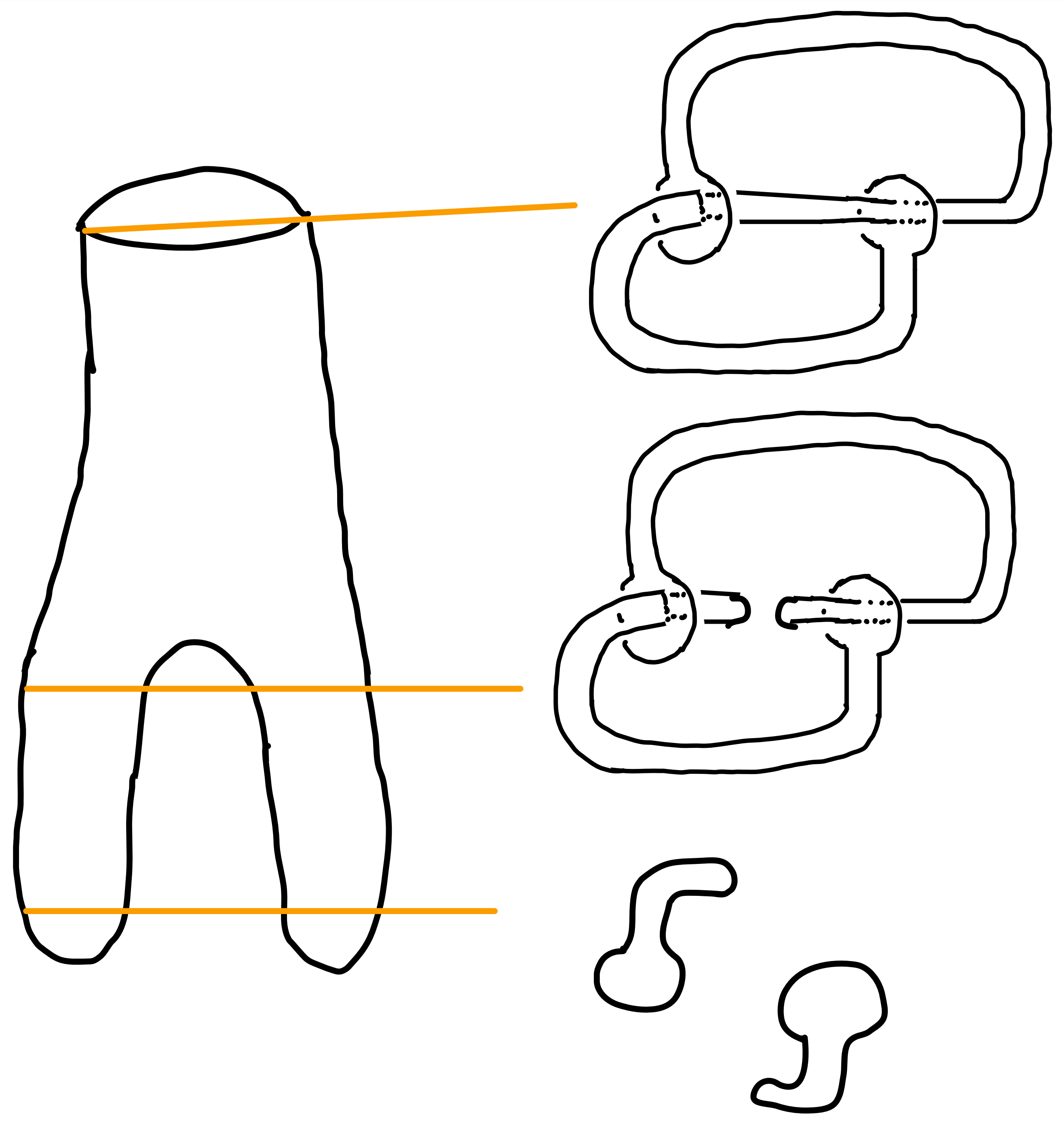


Rephrasing the problem for RL

- How do we relate the question of whether a knot bounds a disk in a 4-ball to RL?

Construct link concordance (by adding bands) until one obtains the unknot

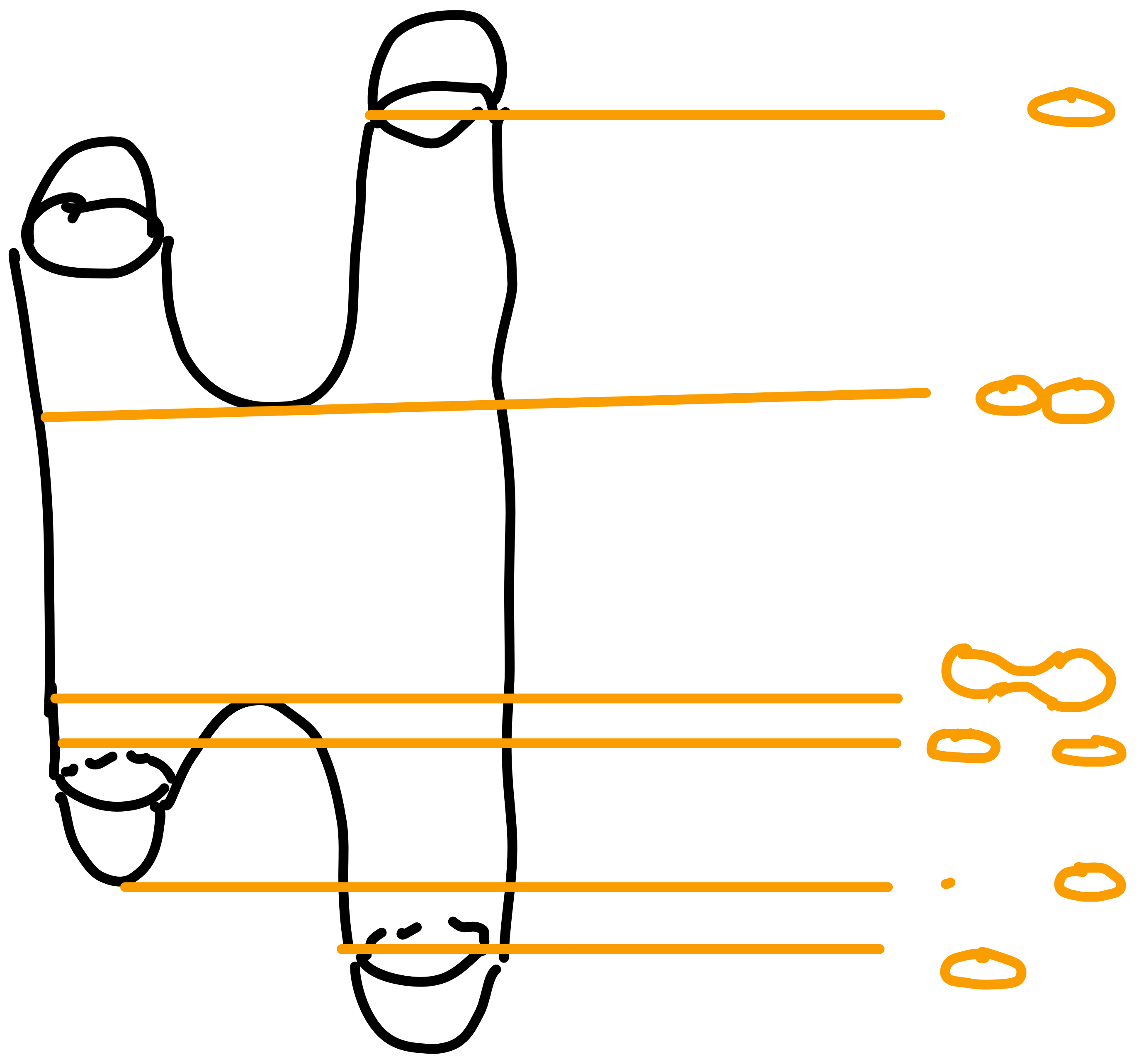


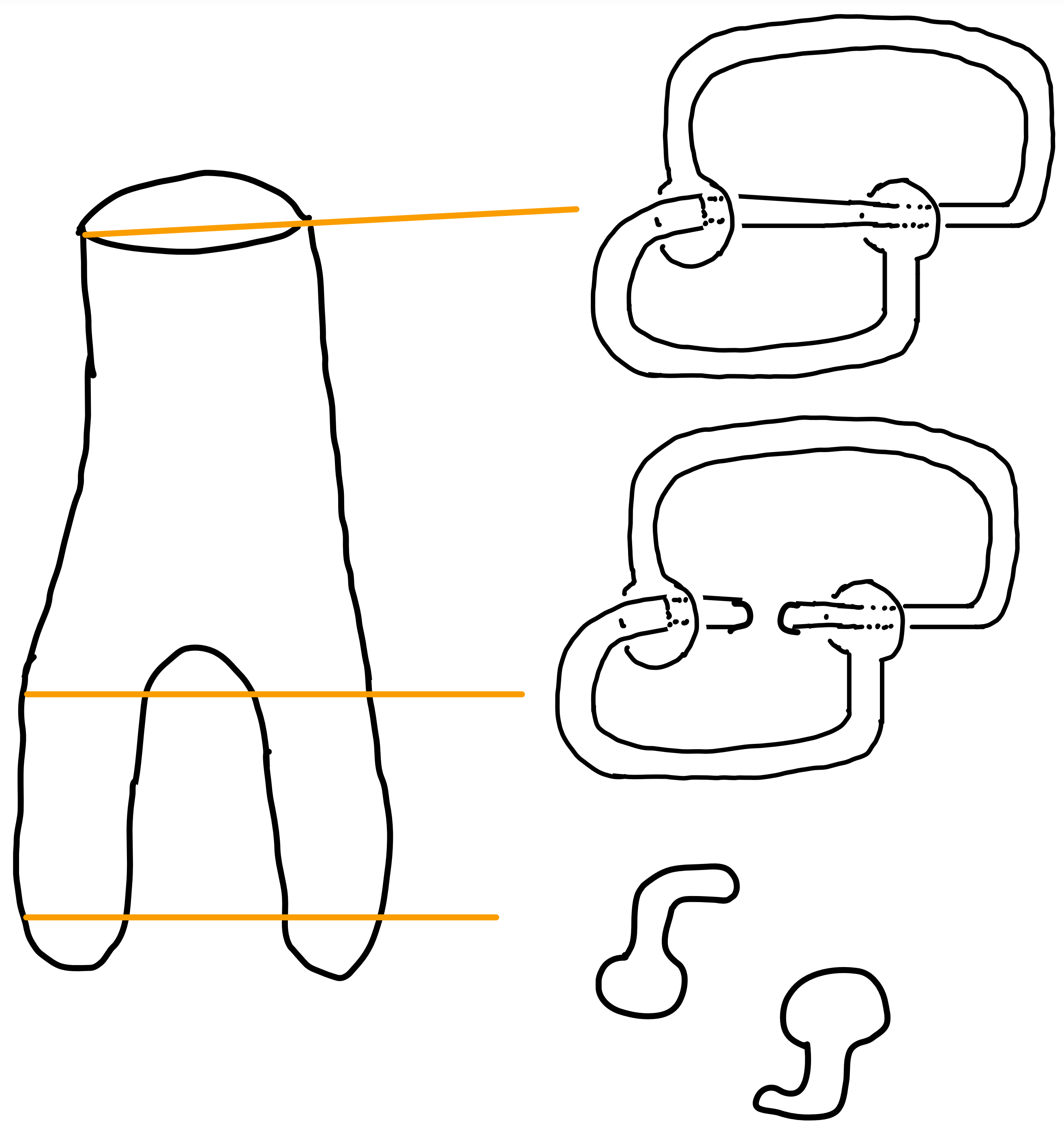


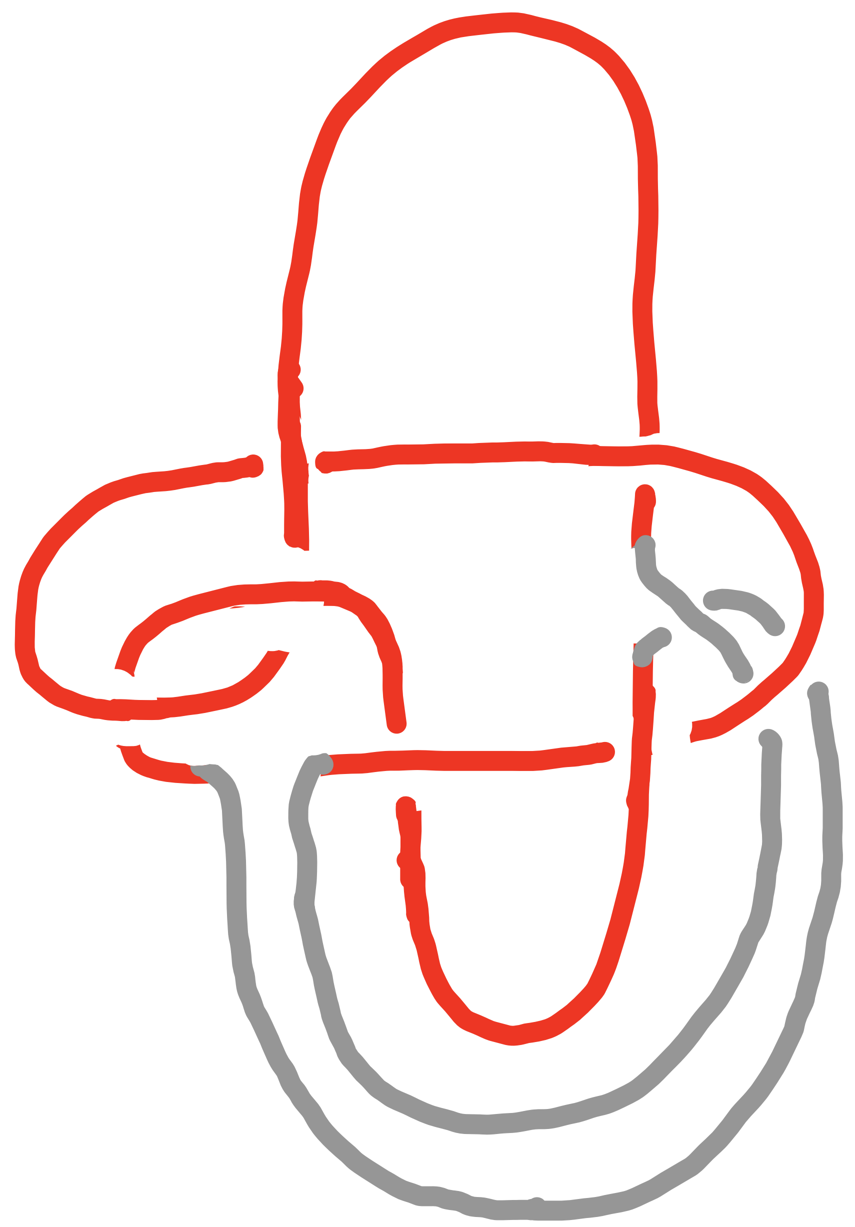
Rephrasing the problem for RL

- How do we relate the question of whether a knot bounds a disk in a 4-ball to RL?

Construct link concordance (by adding bands) until one obtains the unknot

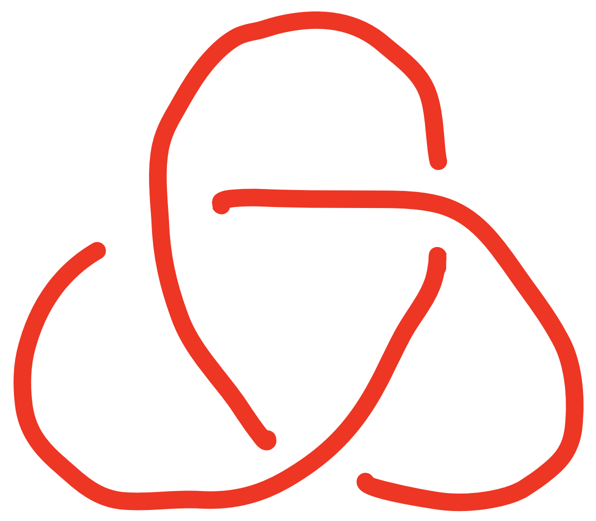




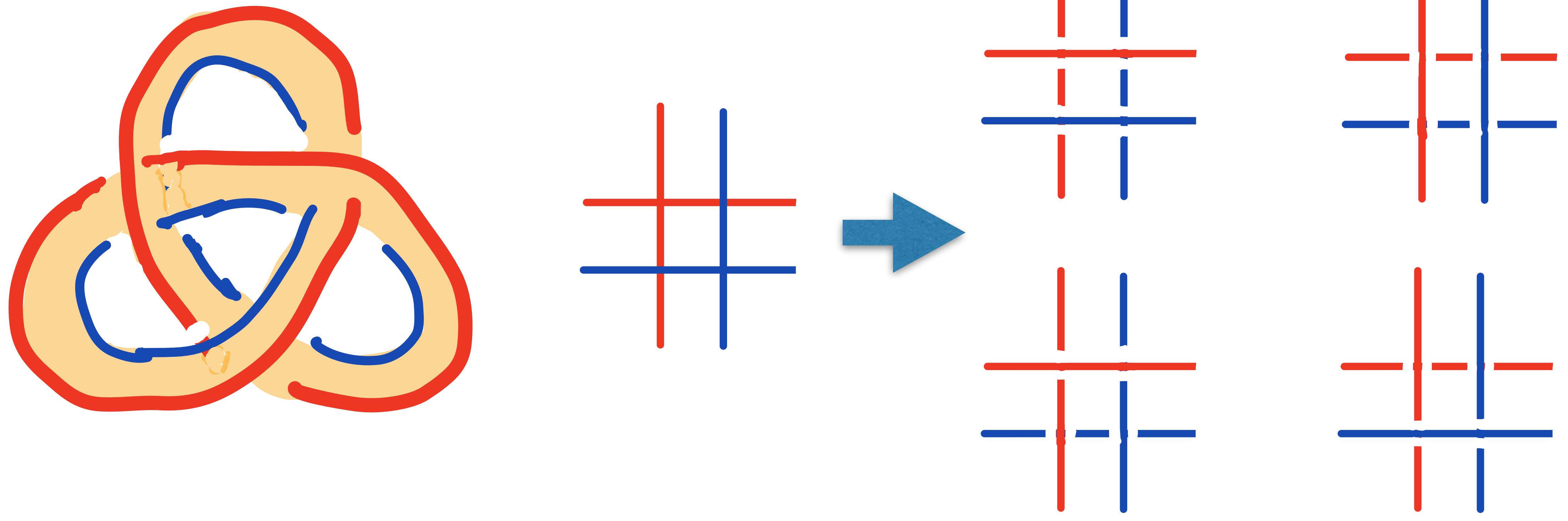


Generate training set (algorithm 1)

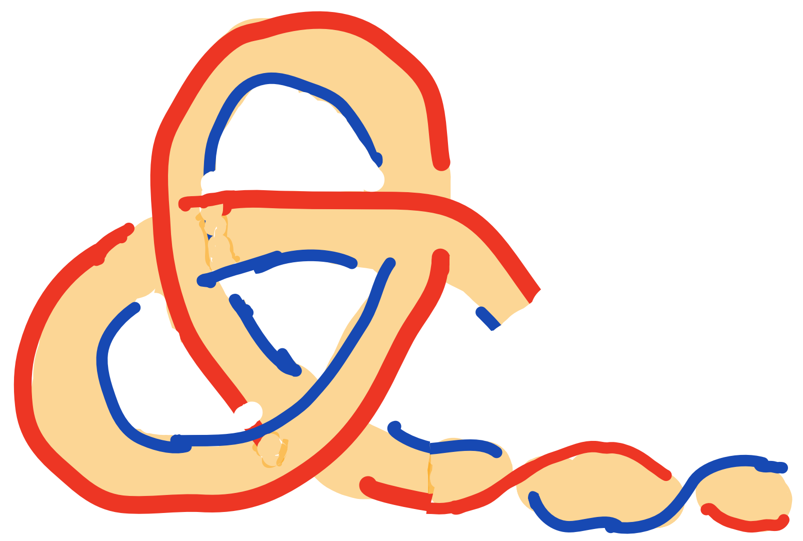
- Generate a random (complicated) knot



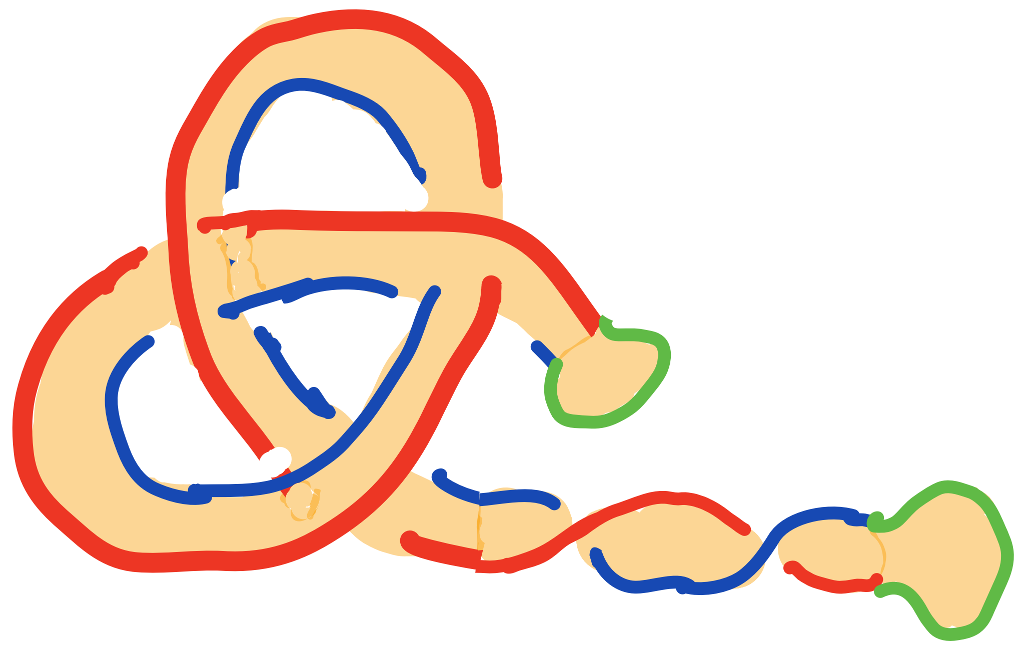
- "Double" the knot to turn it into a band, choosing random band intersections



- Insert a few twists

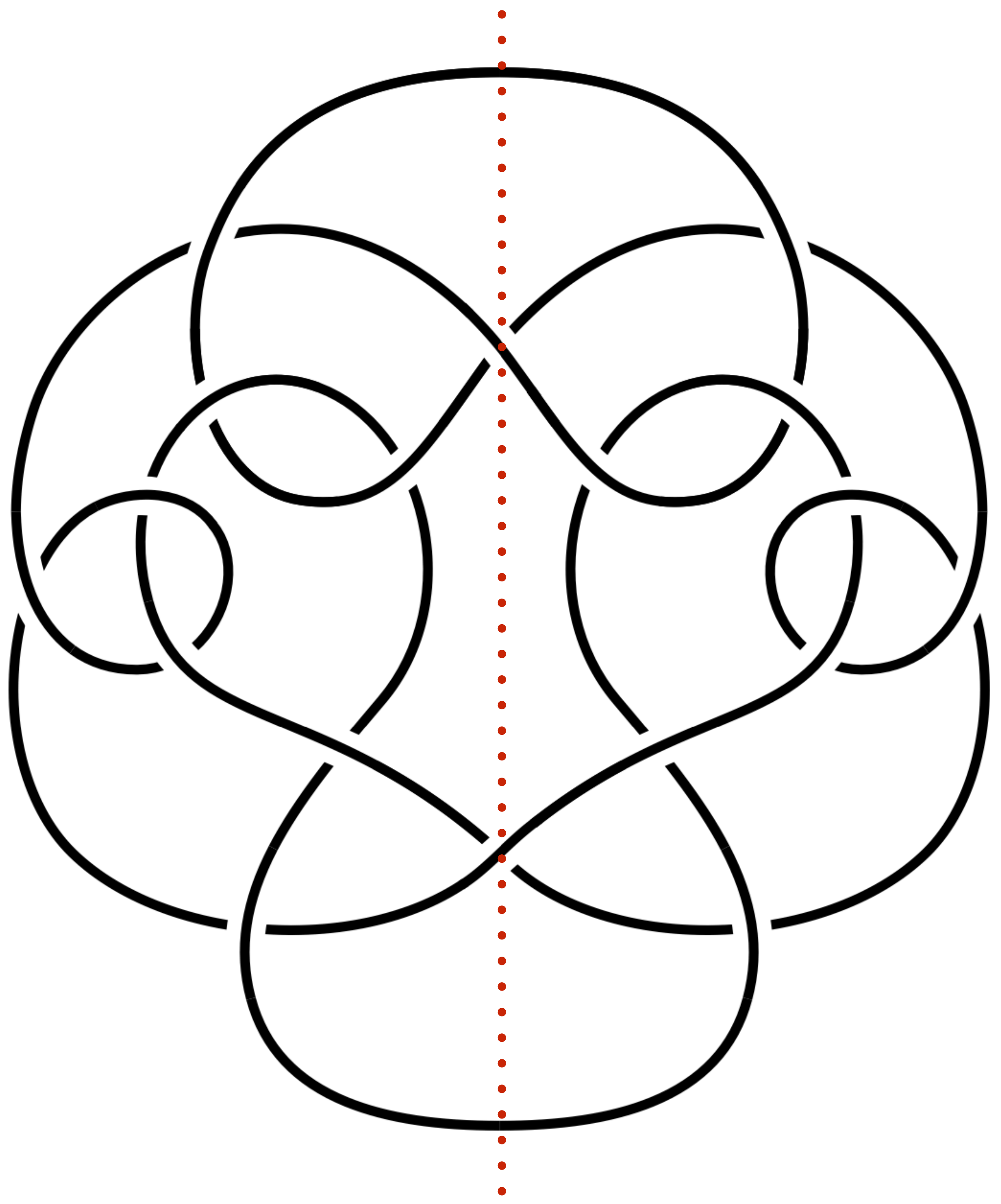


- Cut the band open and put it in between two knots (we choose unknots)



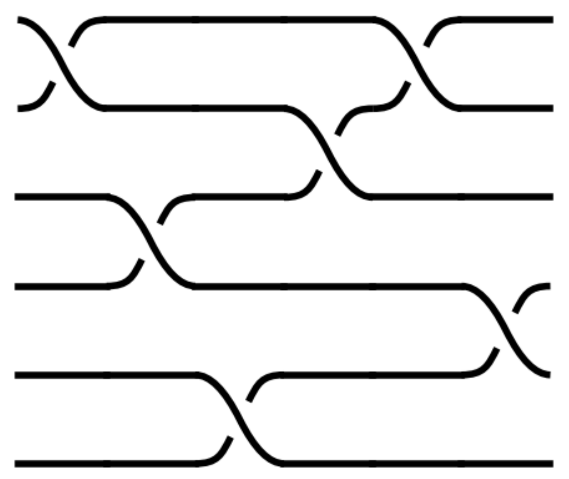
Generate training set (algorithm 2)

- Symmetric knots are ribbon

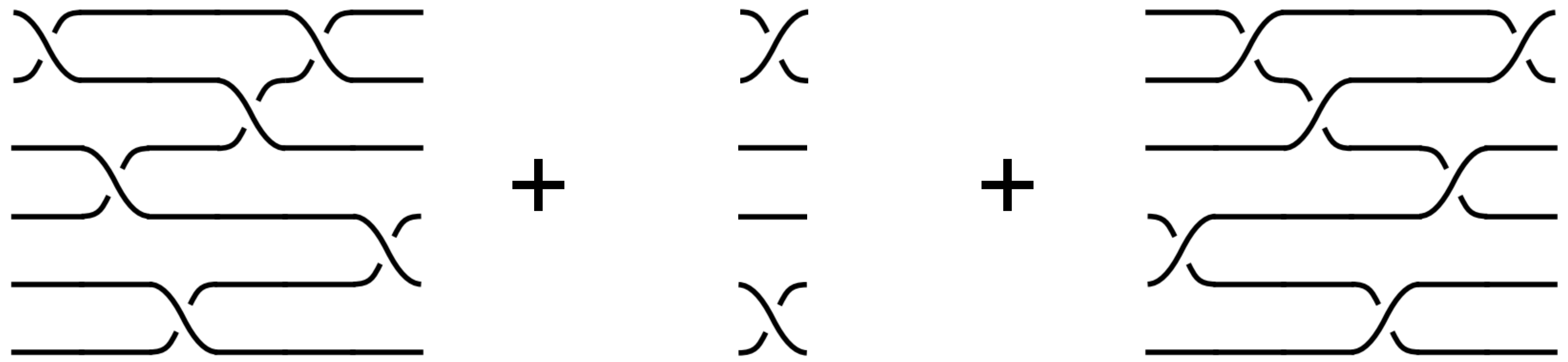


[image: Seelinger `13]

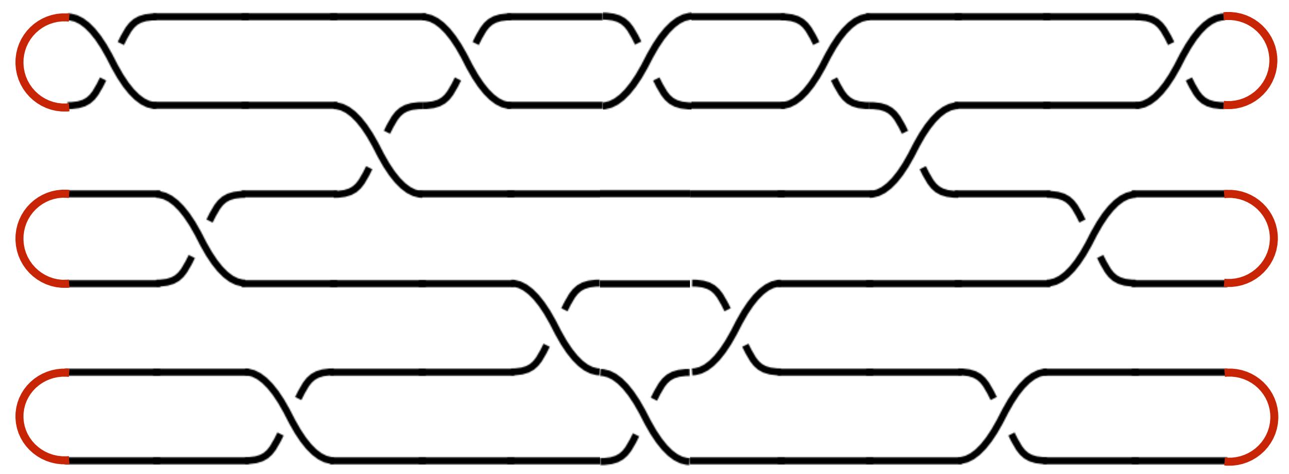
- Generate random braid w/ even number of generators



- Reflect & insert crossings in the middle



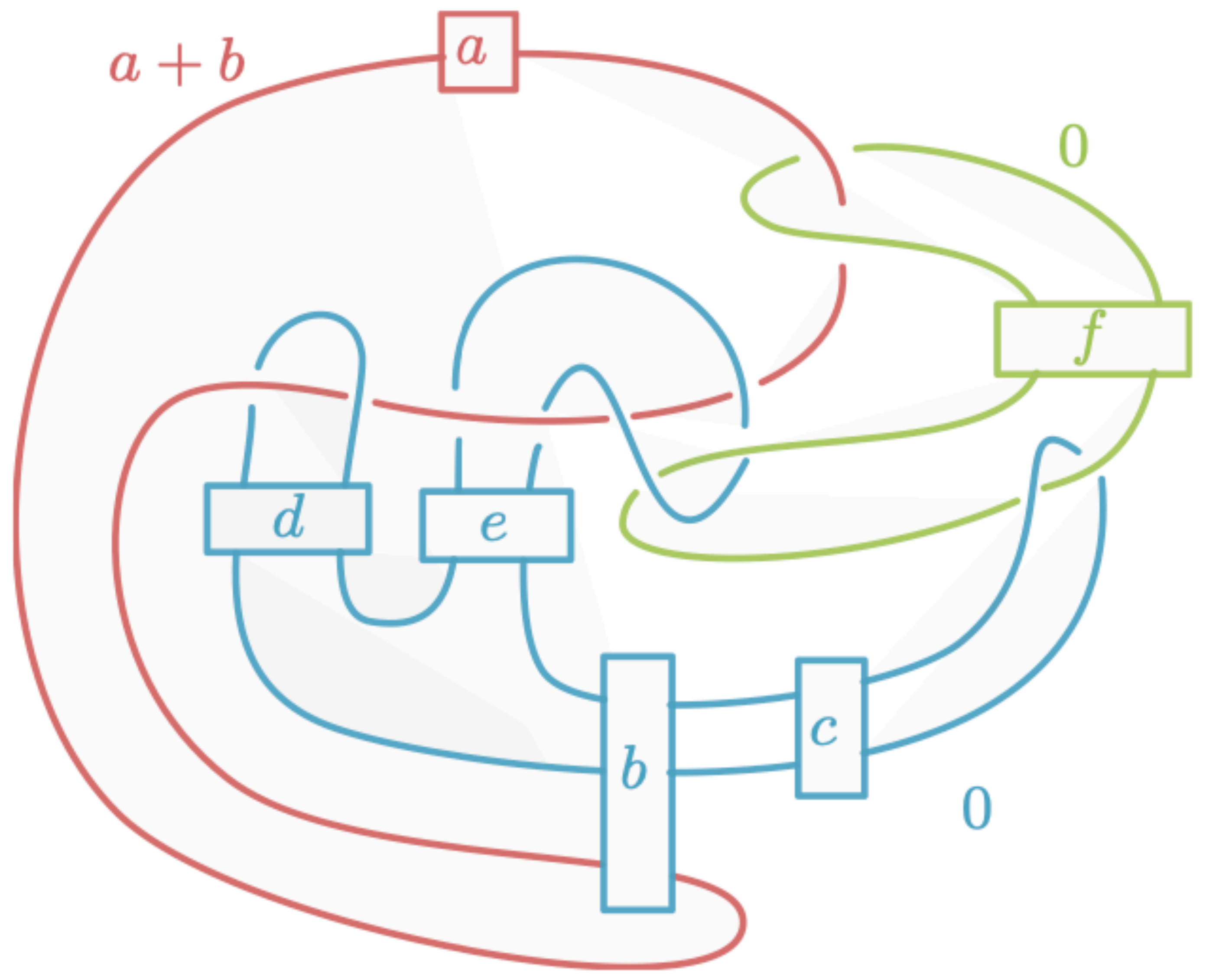
- Identify ends pair-wise



Databases of interesting knots

- Manolescu-Piccirillo constructed a family of 3375 links

  $a, c, e \in [-2, 2], \quad b, d, e \in [-1, 1]$ [Manolescu,Piccirillo `21]
- Out of them, 1051 have signature 0 and their Alexander poly factorizes to satisfy the Fox-Minor criterion [Fox,Milnor `66]
- 892 G- and 893 B-type links are found to have fusion number 1 within 5 min (unknot after adding 1 band)
- In 17 cases, B was found to be ribbon but not K, or vice versa (potential SPC4 counterexamples)
  - Trying these knots longer left 3 potential counter-examples,
  - Ciprian and Lisa ruled out using other techniques by hand
- 10 cases have no known slice obstructions (checked concordance invariants signature, Rasmussen, Tau, HKL [Herald, Kirk, Livingston `10]) but neither the B nor the K link were shown to be ribbon with our techniques (we are running a yet more focused search)

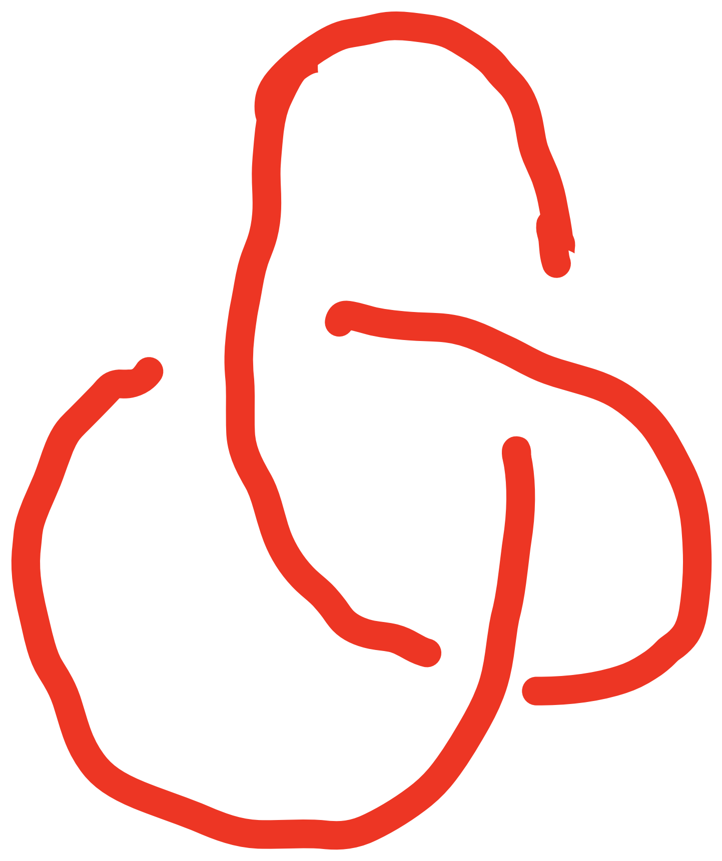


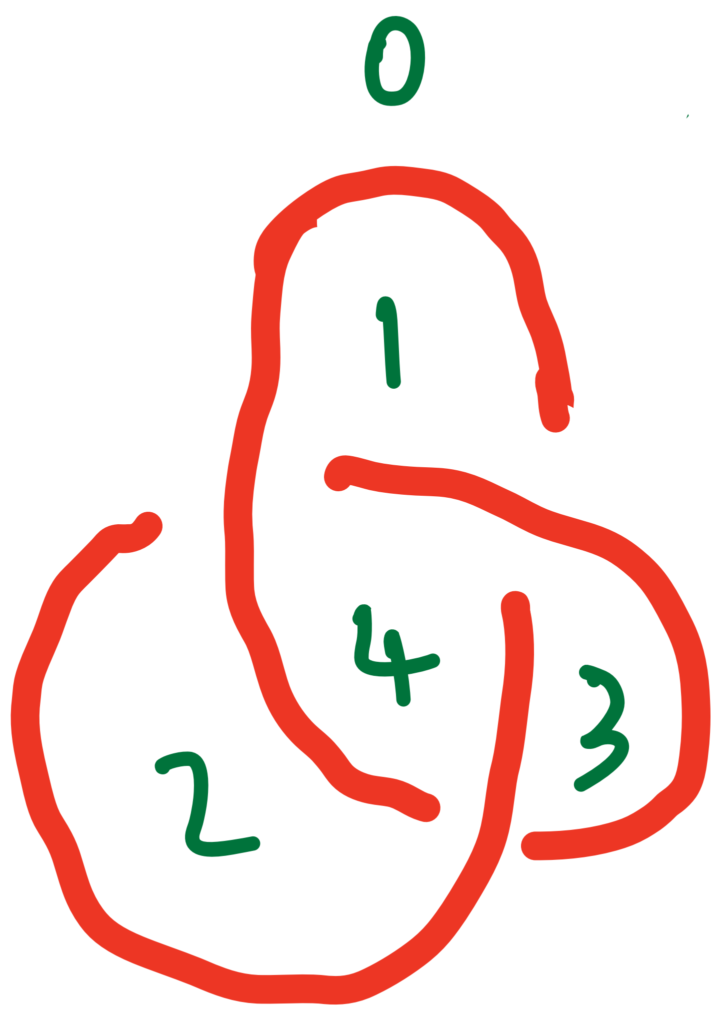
Databases of interesting knots

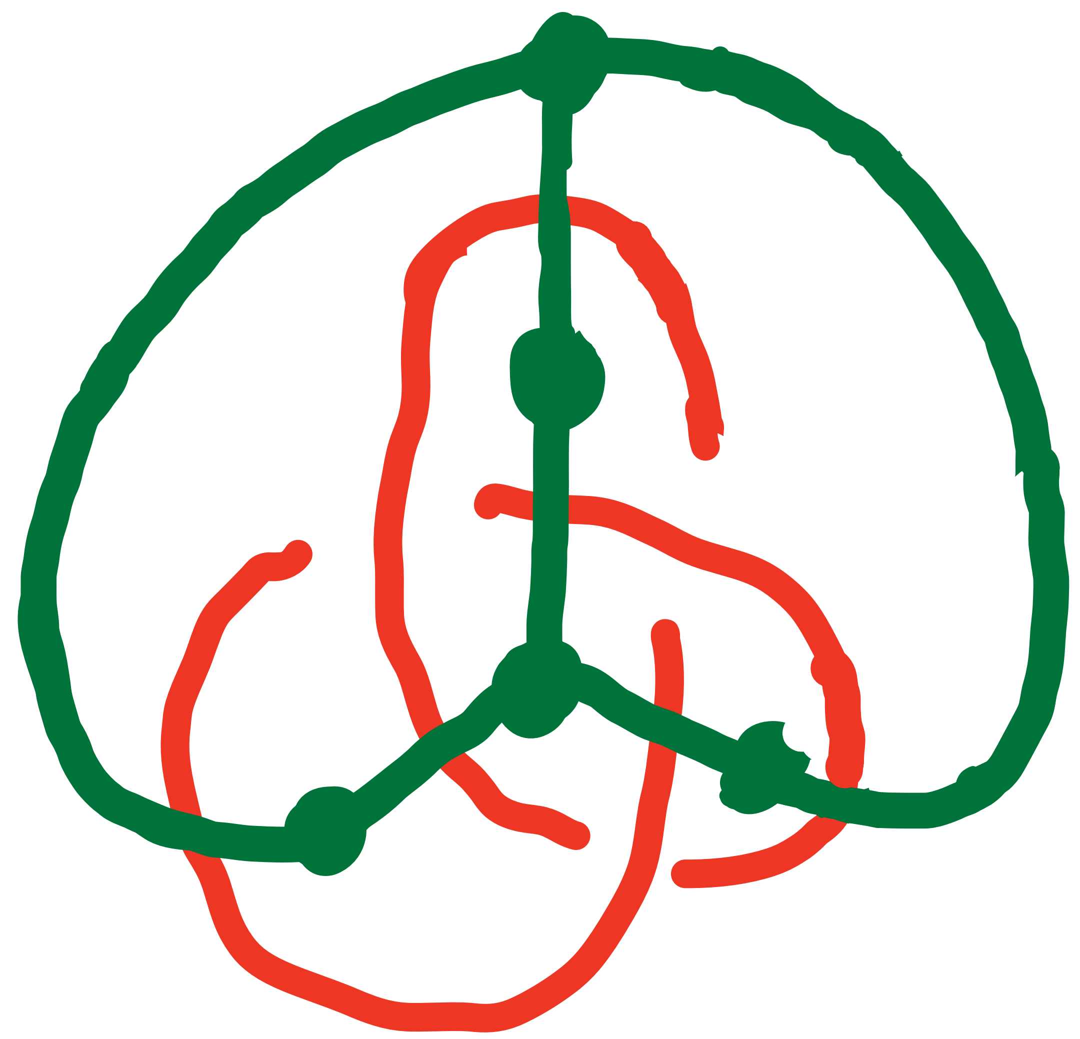
- Knots have been classified up to O(15) crossings or so. We can run the program on all of them and check. Up to 14 crossings, there are 27 “oddities”, i.e., knots with no known slice obstruction that are not known to be slice or ribbon. We found bands for 5 of them, proving they are ribbon. **[SnappPy Database]**
- Gompf-Scharlemann-Thompson proposed a family of knots that could serve as a potential counterexample to the slice-ribbon conjecture. We could not find bands for these examples **[Gompf, Scharlemann, Thompson `13]**
- Owens-Swenton proposed an algorithm for finding ribbon disks in alternating knots. They provide a database of alternating ribbon knots up to 20 crossings. The status of some remains unresolved (with a bounty of $10 per knot to resolve their property). **[Ownes, Swenton `21]**

Reinforcement Learning for Bands

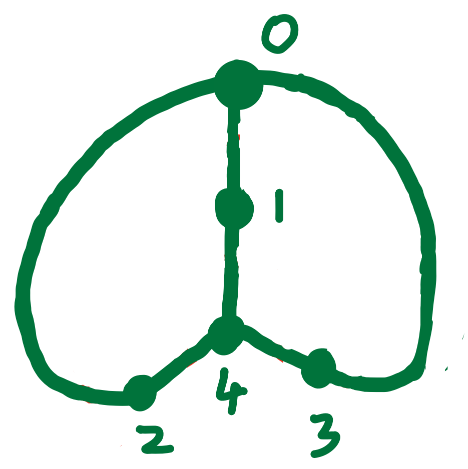
- **States:** a representation of the knot + band (we choose a matrix representation based on dual graphs)



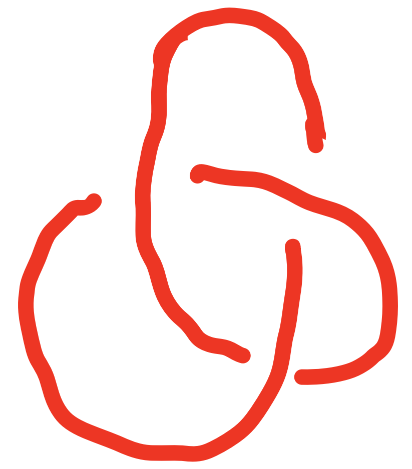




States

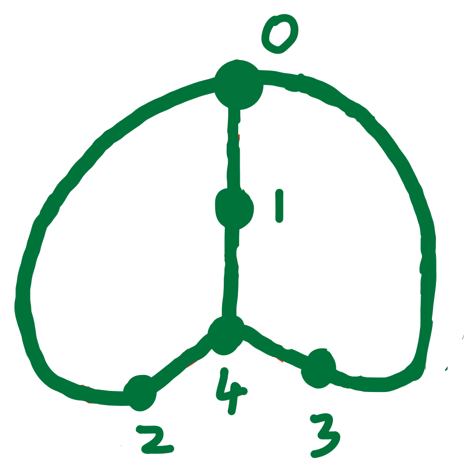


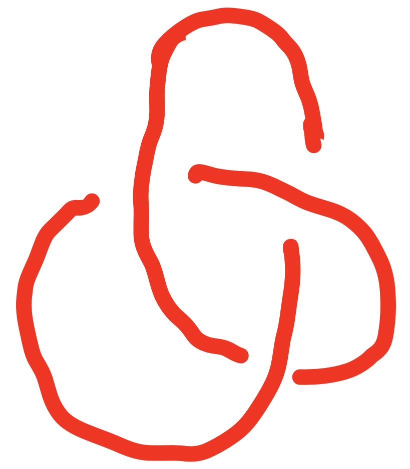
$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix}$$



- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings

States



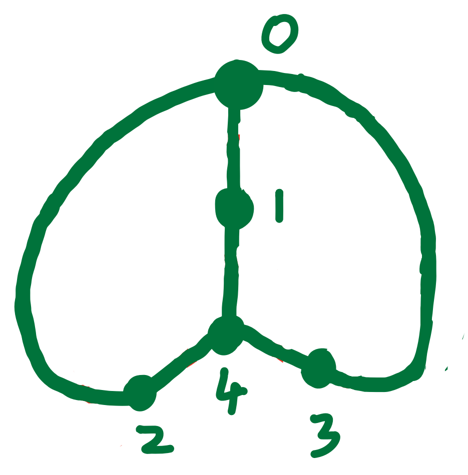


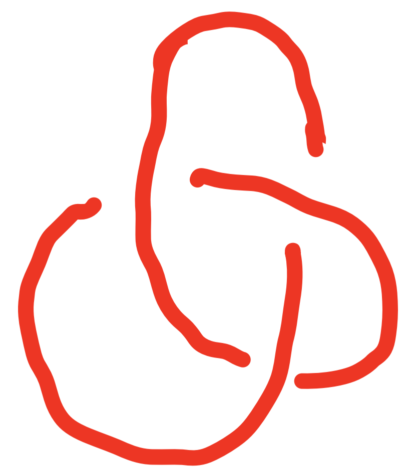
$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to

States



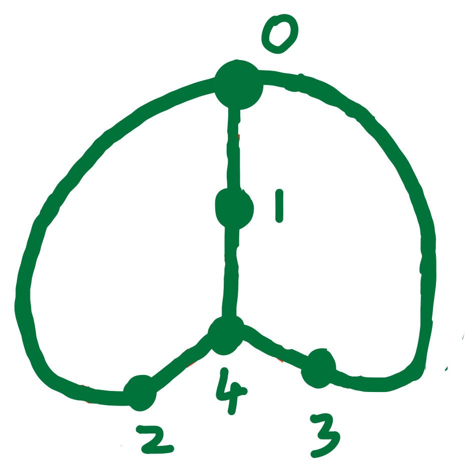


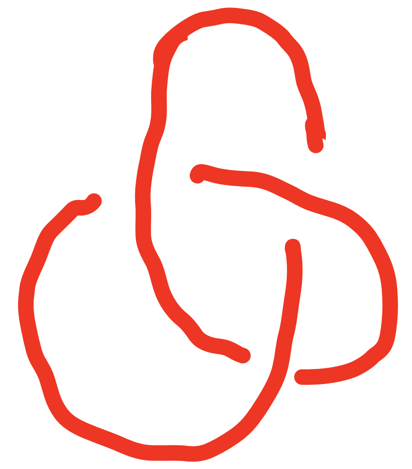
$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to
  - Band: The path of the band, expressed as numbered faces in the adjacency matrix; signs indicate over/under

States



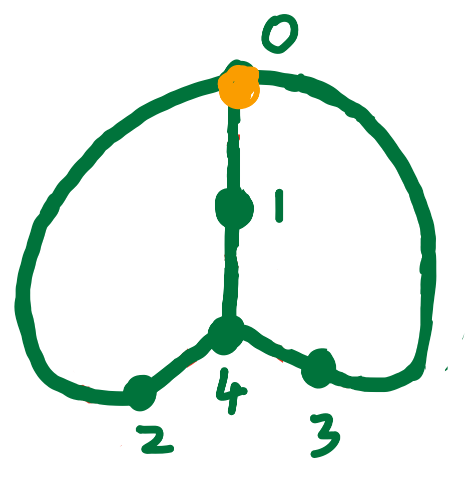


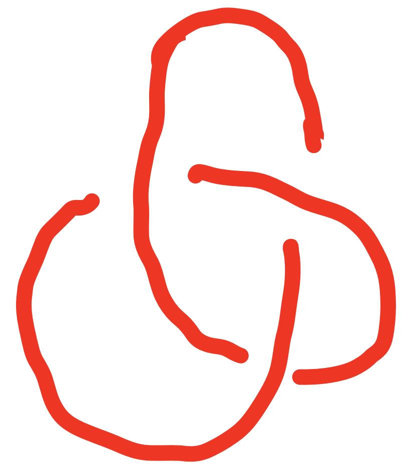
$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to
  - Band: The path of the band, expressed as numbered faces in the adjacency matrix; signs indicate over/under
  - Twists: Indicates the number (and direction) of band twists

States



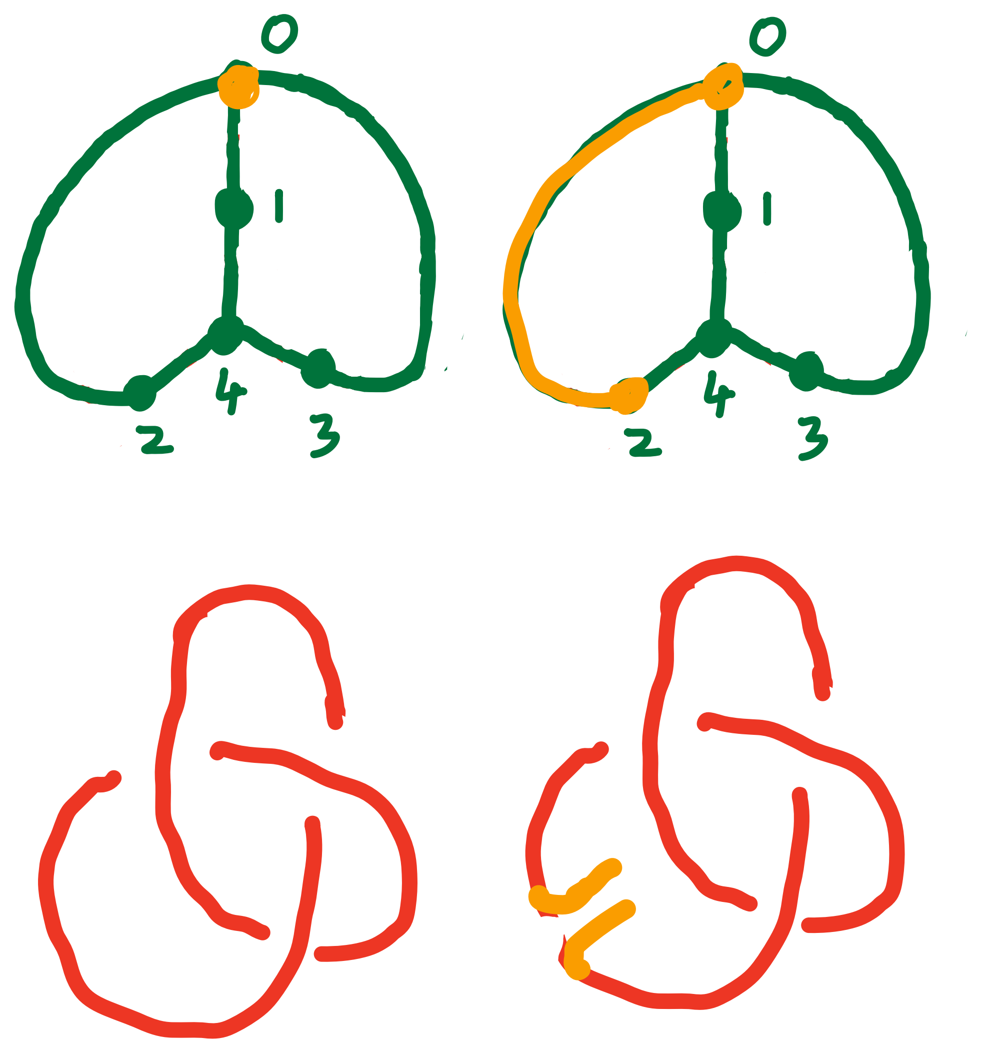


$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix} \begin{pmatrix} \bigcirc\!\!\!\!1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to
  - Band: The path of the band, expressed as numbered faces in the adjacency matrix; signs indicate over/under
  - Twists: Indicates the number (and direction) of band twists

States

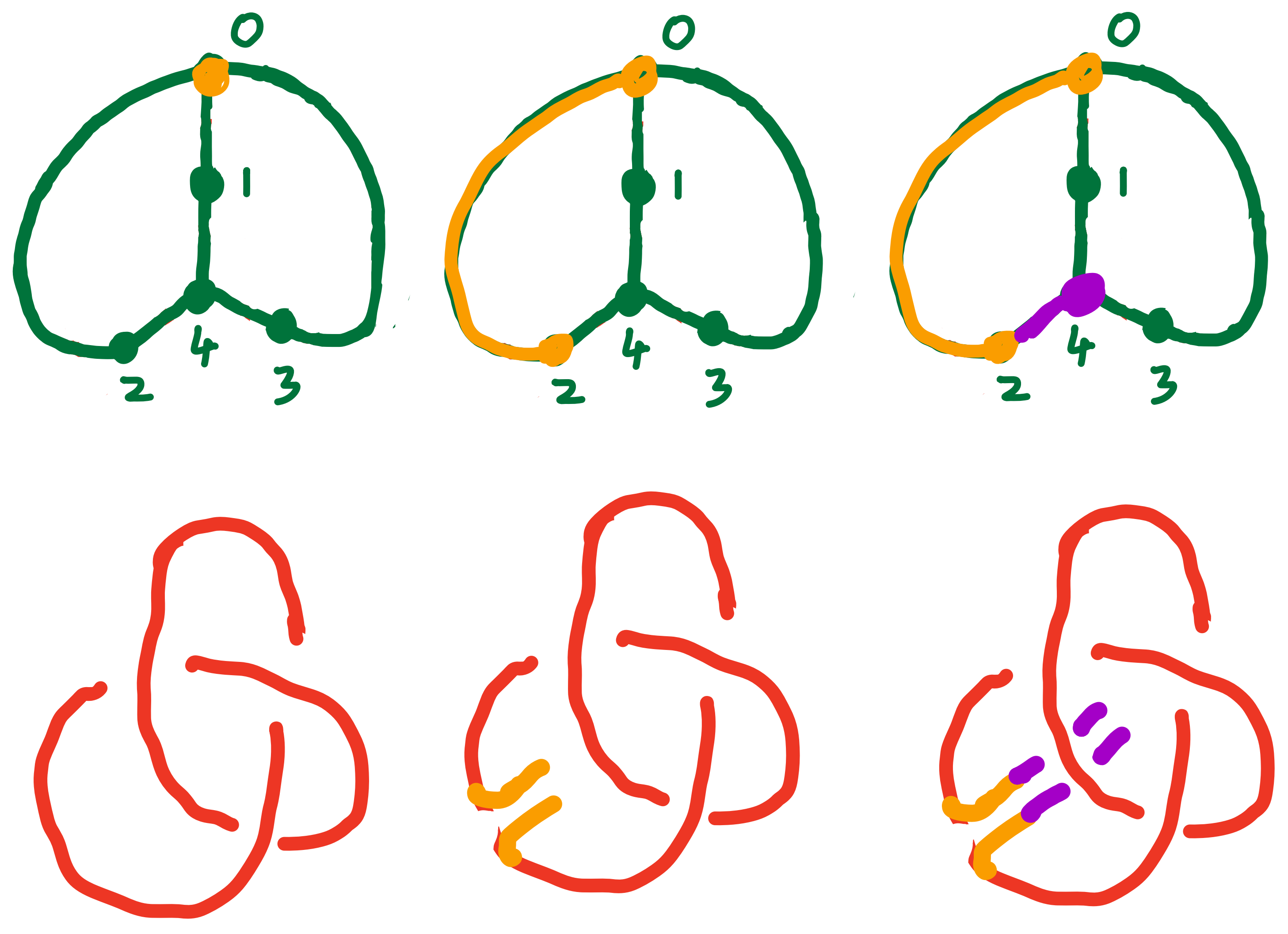


$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to
  - Band: The path of the band, expressed as numbered faces in the adjacency matrix; signs indicate over/under
  - Twists: Indicates the number (and direction) of band twists

States

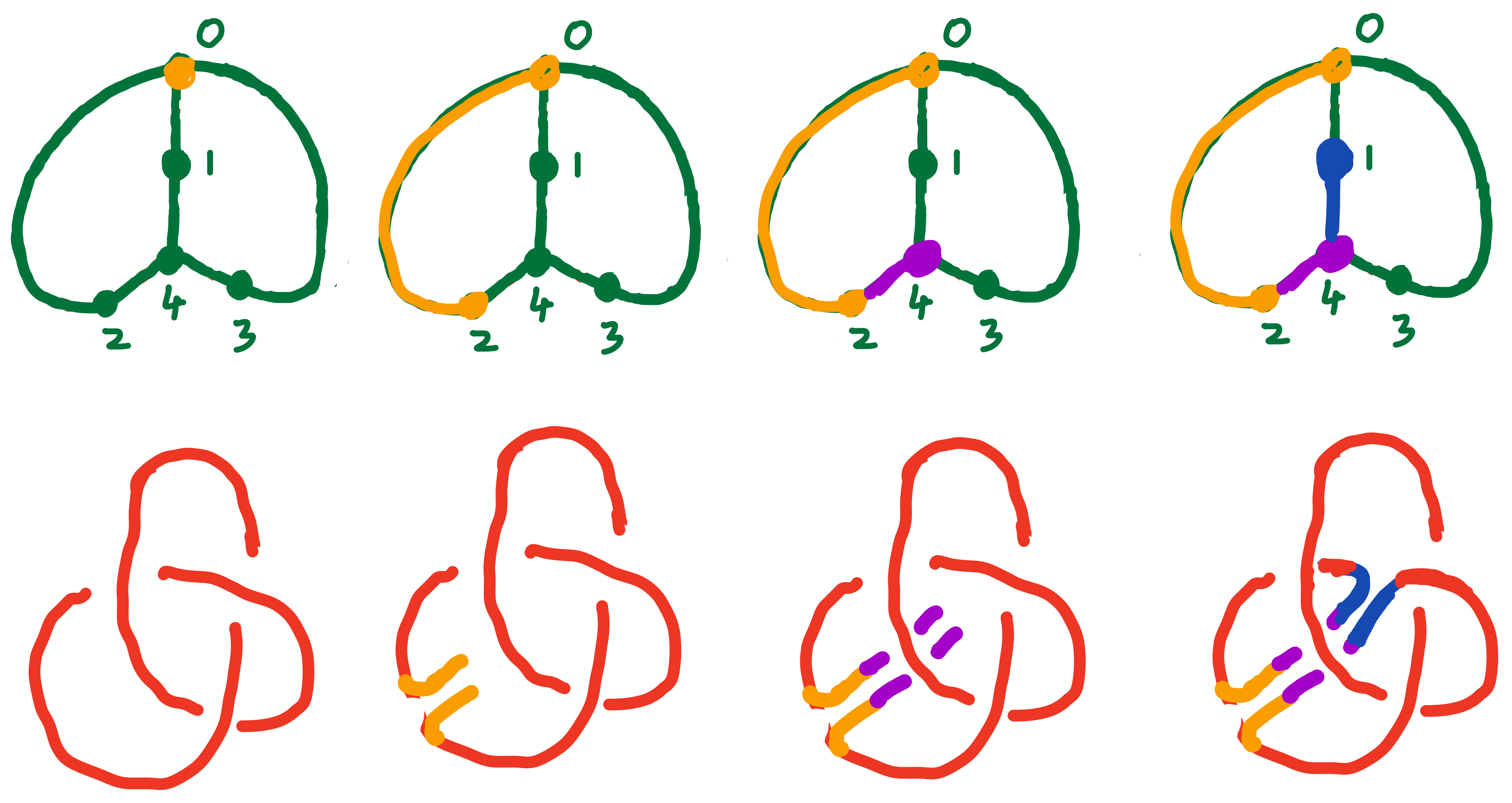


$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \boxed{-1} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{-1} & 0 & \boxed{1} \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to
  - Band: The path of the band, expressed as numbered faces in the adjacency matrix; signs indicate over/under
  - Twists: Indicates the number (and direction) of band twists

States



$$\begin{pmatrix} 0 & 1 & -2 & 3 & 0 \\ -1 & 0 & 0 & 0 & -3 \\ 2 & 0 & 0 & 0 & 2 \\ -3 & 0 & 0 & 0 & -1 \\ 0 & 3 & -2 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

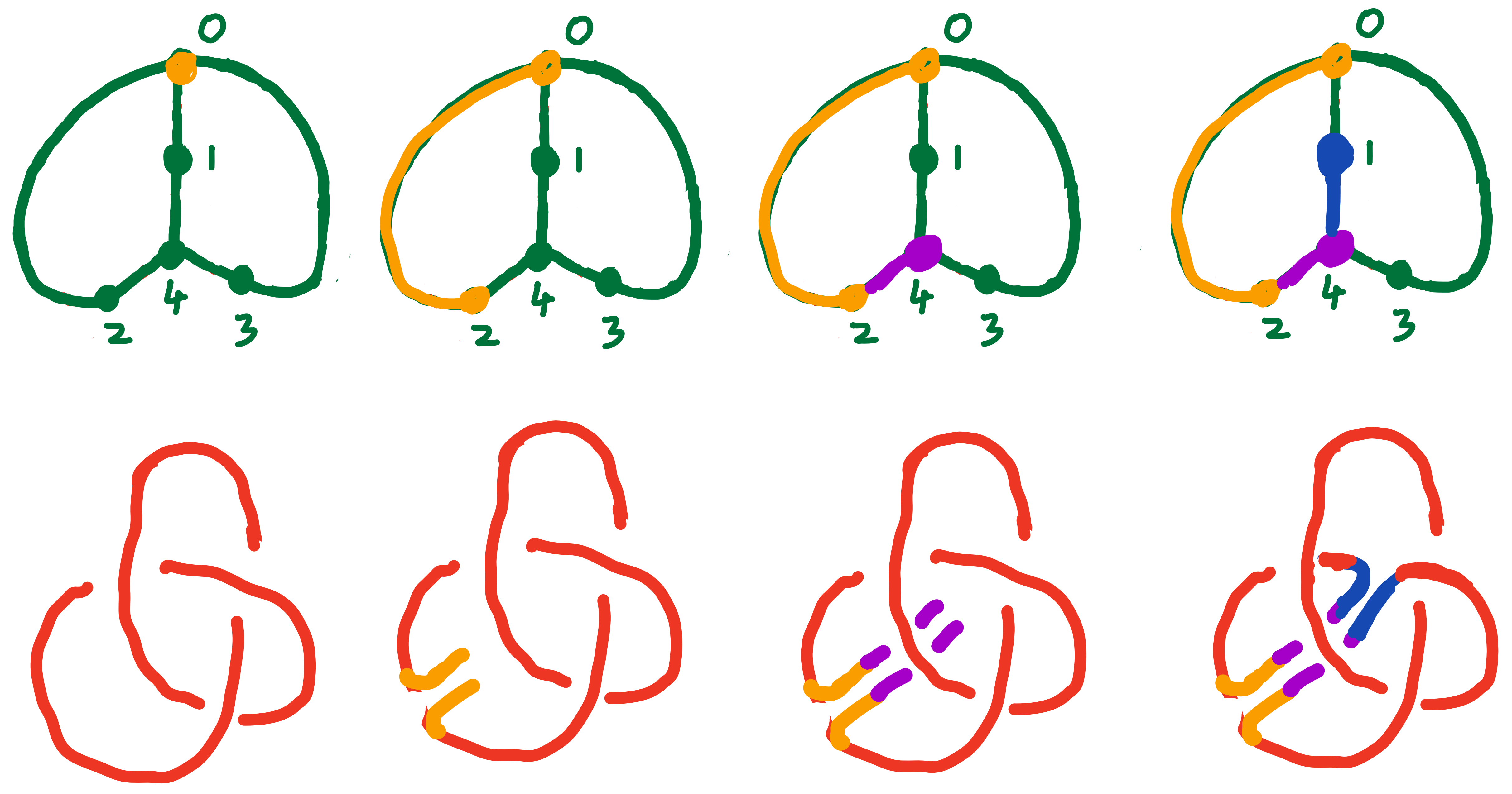
$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- States: four $(N+2) \times (N+2)$ matrices
  - Graph: Gives the dual graph of the link (encoded as a signed adjacency matrix that encodes over- and under-crossings
  - Components: Labels each vertex according to the link component int belongs to
  - Band: The path of the band, expressed as numbered faces in the adjacency matrix; signs indicate over/under
  - Twists: Indicates the number (and direction) of band twists

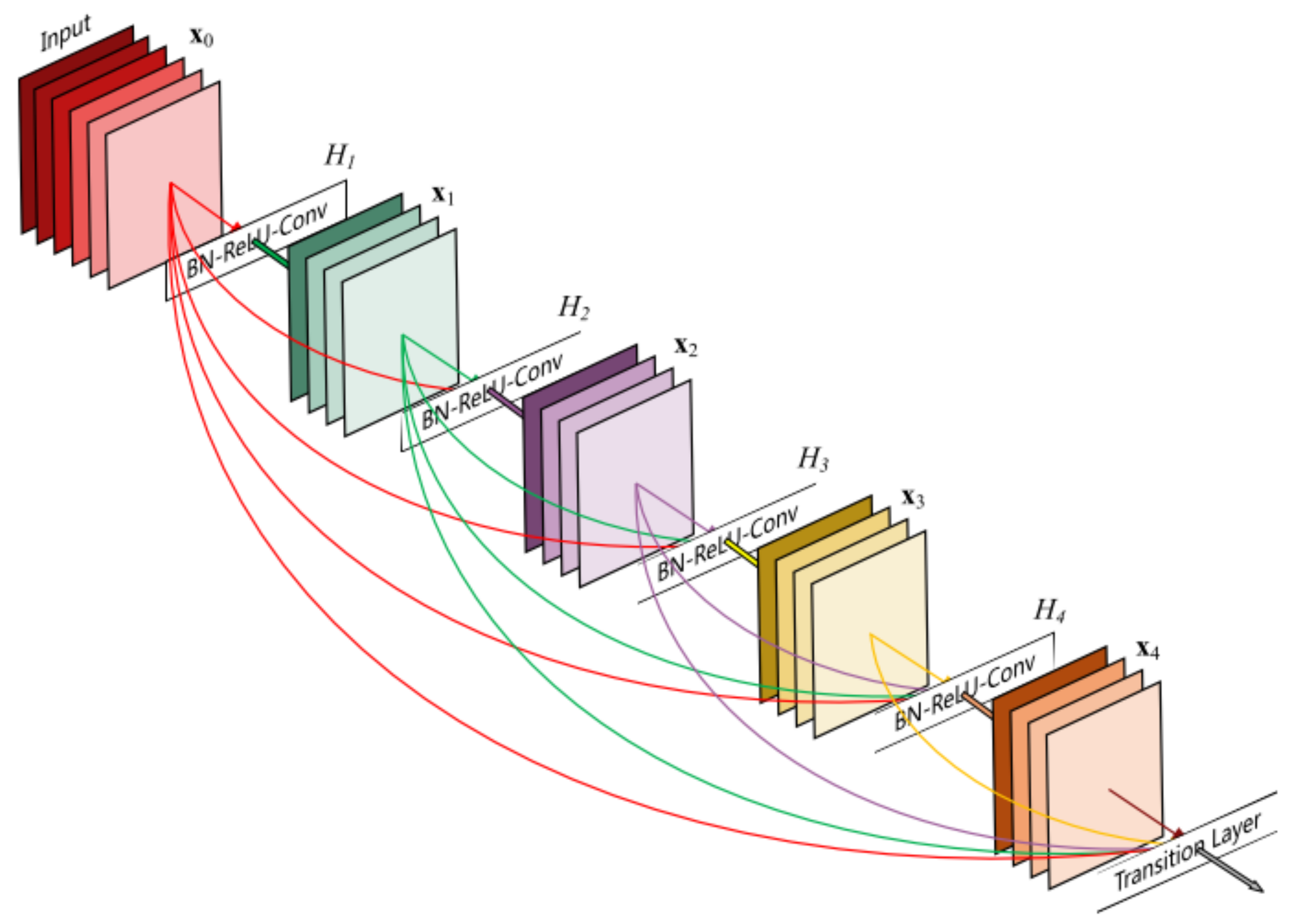
Reinforcement Learning for Bands

- **States:** a representation of the knot + band (we choose a matrix representation based on dual graphs)
- **Actions:** where to start the band + where to move it over/under + twists + where to end it
    - Mask illegal actions
    - Force to attach if self-avoiding walk ends; if attach impossible (since vertex belongs to different component) $\Rightarrow$ terminal state
    - Forbid twisting indefinitely and twisting back and forth
    - Set max number of actions before reset

Actions



- Actions:
- Essentially a self-avoiding random walk
- Predicted by a Neural Network

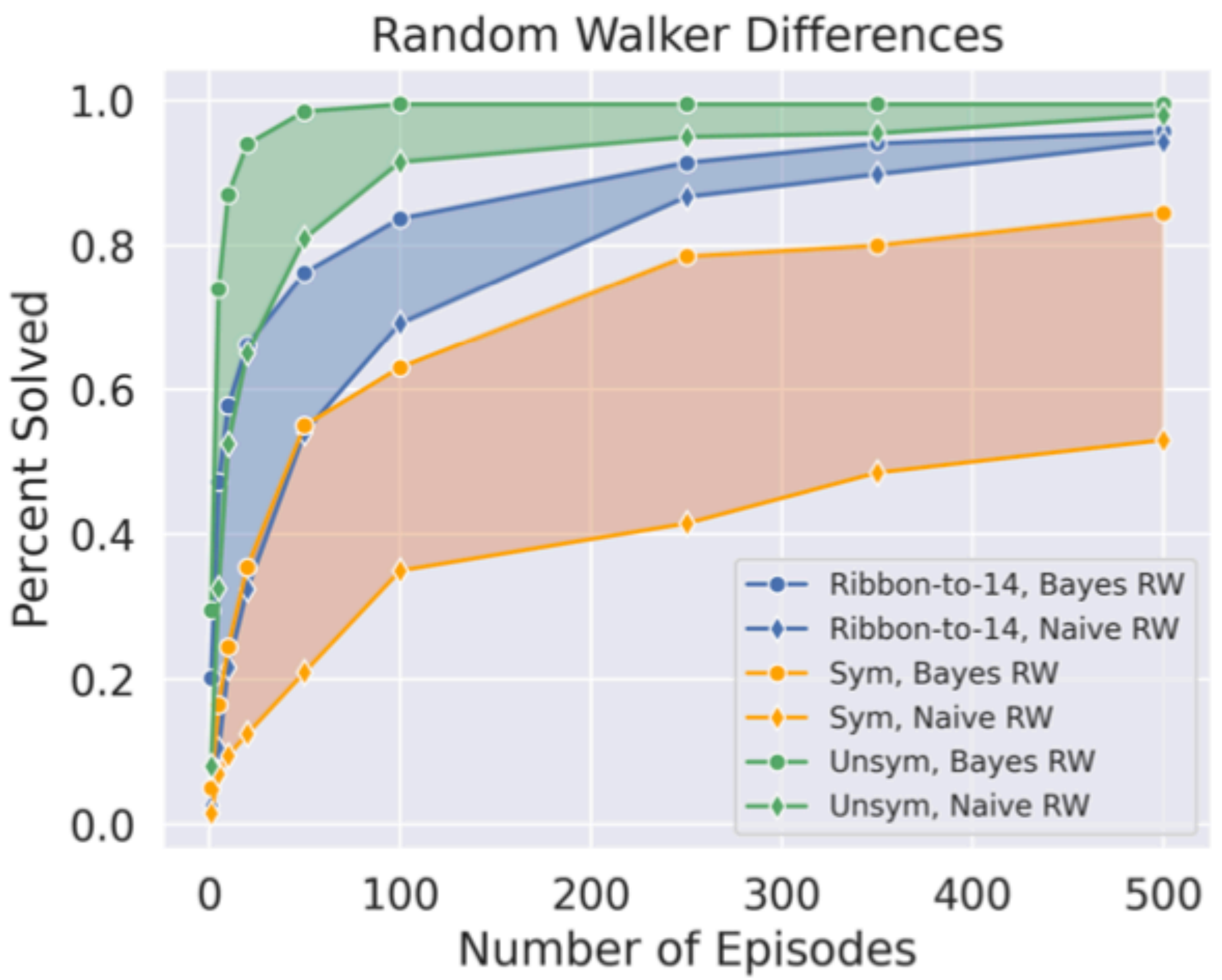


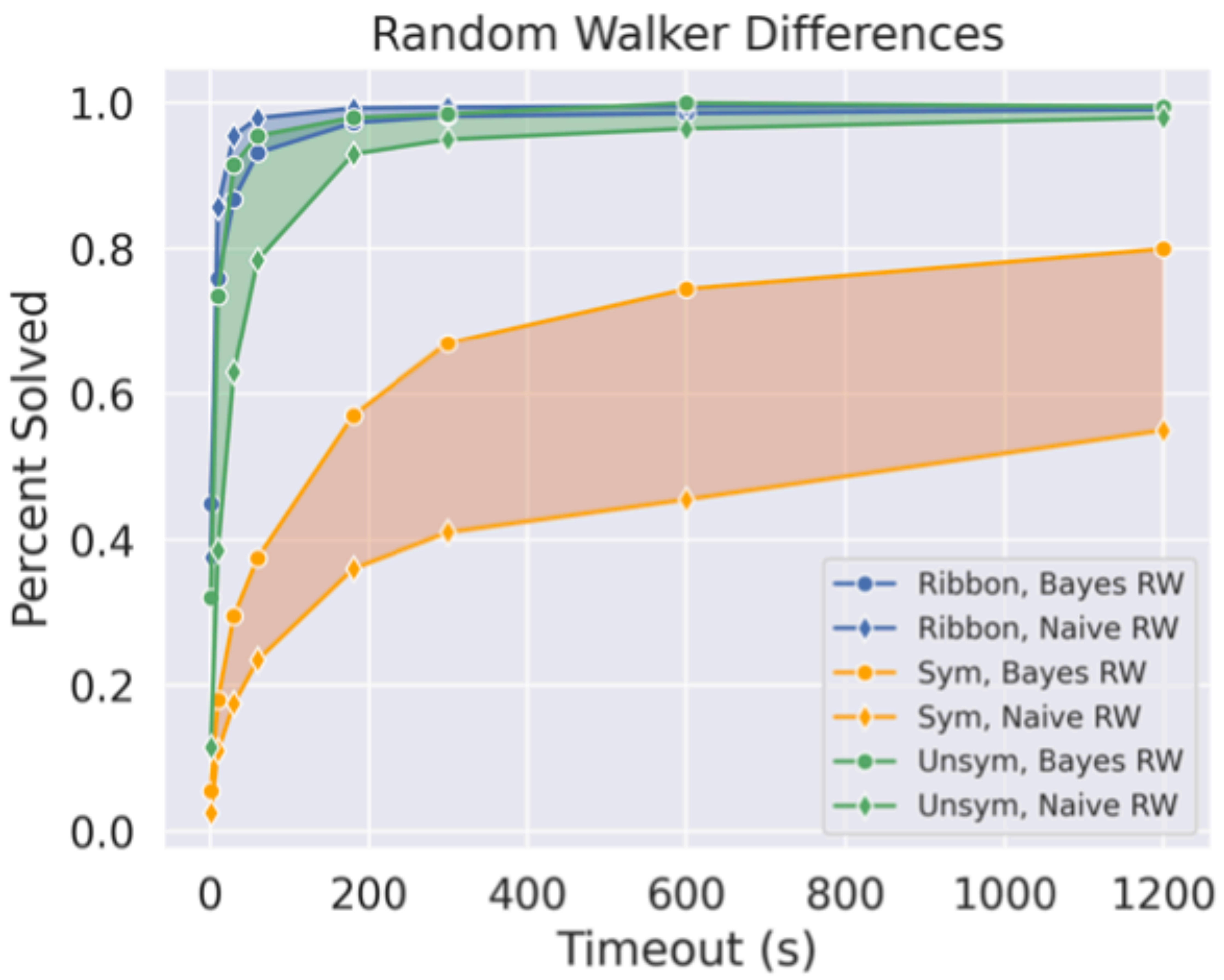
[image taken from stack exchange]

Reinforcement Learning for Bands

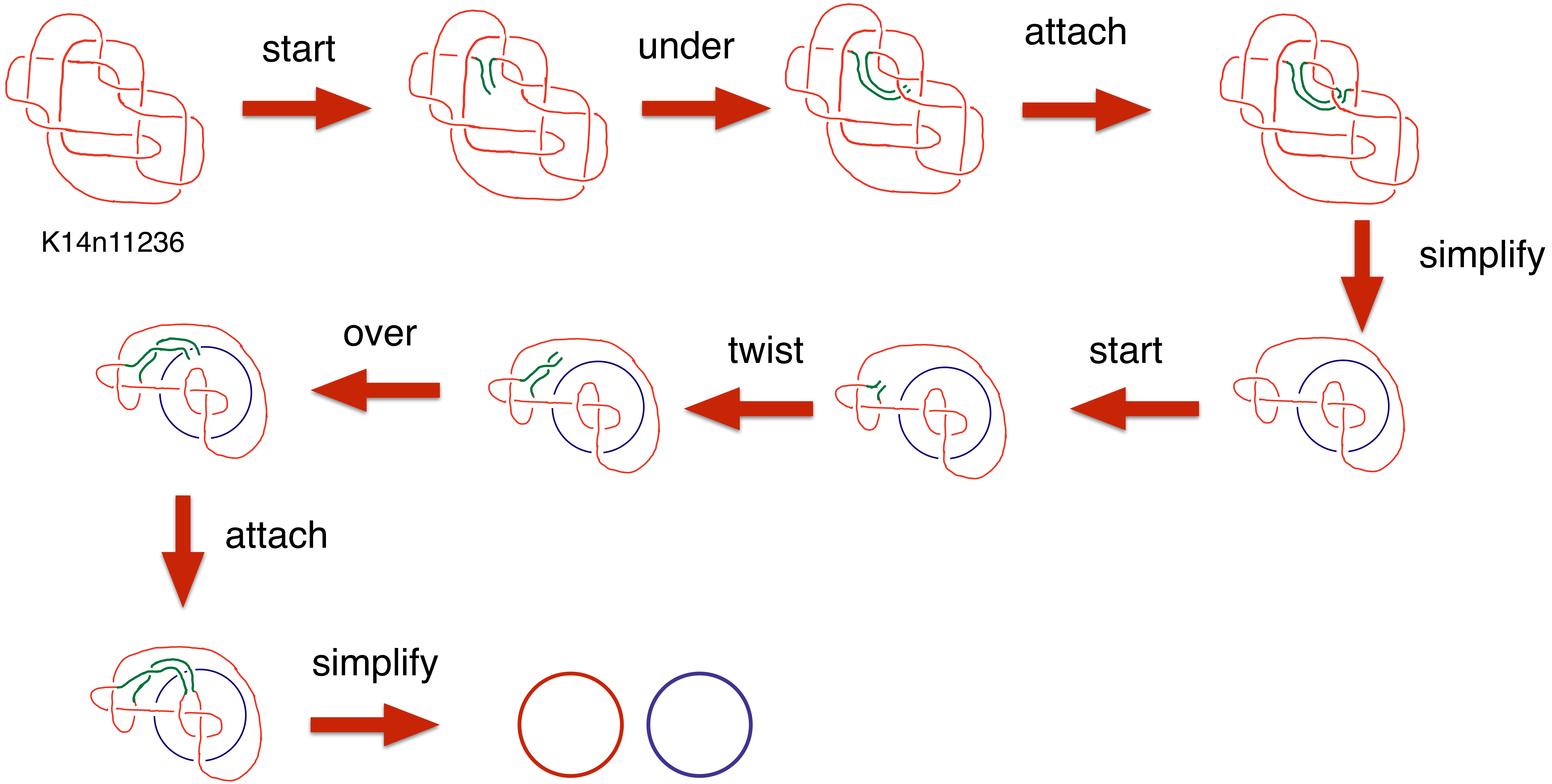
- **States:** a representation of the knot + band (we choose a matrix representation based on dual graphs)
- **Actions:** where to start the band + where to move it over/under + twists + where to end it
- **Terminal states:** Unknot (or unlinked unknots) or stuck
- **Reward:**
    - (Optional): After attaching the band, the number of crossings of the old minus new link: $R = \mathrm{N_X}(L_{\mathrm{old}}) - \mathrm{N_X}(L_{\mathrm{new}})$
    - Final reward when the knot is shown to be ribbon

Ribbon Knots Performance

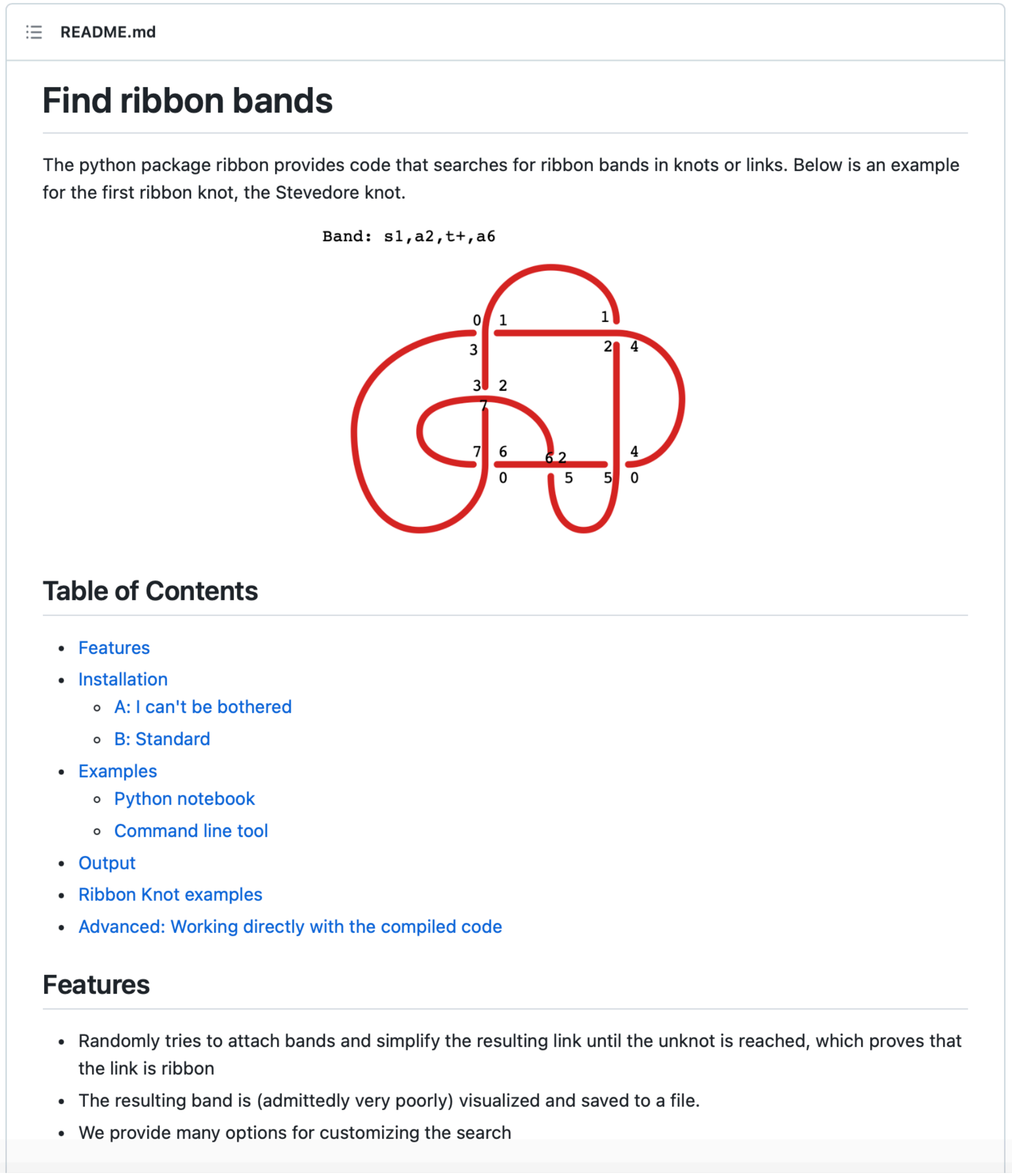




Ribbon Knots



Try it yourself…



README.md

Find ribbon bands

The python package ribbon provides code that searches for ribbon bands in knots or links. Below is an example for the first ribbon knot, the Stevedore knot.

```
Band: s1,a2,t+,a6
```

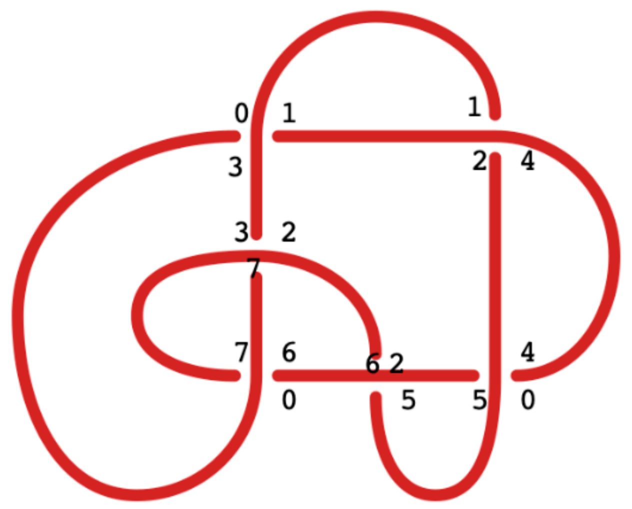


Table of Contents

Features

- Randomly tries to attach bands and simplify the resulting link until the unknot is reached, which proves that the link is ribbon
- The resulting band is (admittedly very poorly) visualized and saved to a file.
- We provide many options for customizing the search

https://github.com/ruehlef/ribbon

Multiple Postdoc Positions at Northeastern

- Groups of Halverson and Ruehle
- Deadline December 1st

Multiple Postdoc Positions at IAIFI

- More info:
  https://iaifi.org/fellows
- Job ad / application:
  https://academicjobsonline.org/ajo/jobs/25055

- Work at MIT with scientists from Harvard, MIT, Northeastern, Tufts
- 3 years
- Deadline October 10

Recap

1 Knots as RL problems

- Need provable results
- Can break up manipulations in set of (Reidemeister) moves

2 The Unknot Decision Problem

- TRPO achieves consistency good performance over many crossings

3 Ribbon knots and SPC4

- Bayes RW performs very well
- Can rule out essentially all proposed counter-examples to SPC4
- Proved ribbonness for new knots