The unknotting number, hard unknot diagrams, and Reinforcement Learning

Taylor Applebaum, Sam Blackwell, Alex Davies, Thomas Edlich, András Juhász, Marc Lackenby, Nenad Tomašev, and Daniel Zheng

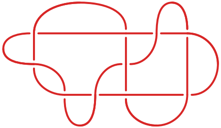
University of Oxford

22 May 2024

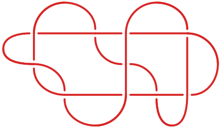
The unknotting number

- The unknotting number $u(\mathcal{D})$ of a diagram $\mathcal{D}$ of a knot K is the minimal number of crossing changes required to obtain a diagram of the unknot
- $u(\mathcal{D}) \leq c(\mathcal{D})/2$
- $u(K) := \min\{u(\mathcal{D}) : \mathcal{D} \text{ a diagram of } K\}$
- 6 out of 165 prime knots K with $c(K) \leq 10$ and 660 out of 2978 prime knots K with $c(K) \leq 12$ have unknown $u(K)$
- In comparison, smooth 4-genus $g_4(K)$ is known for $c(K) \leq 12$

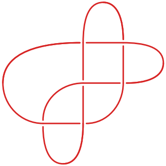
The knot 10_8



(a) The knot 10_8



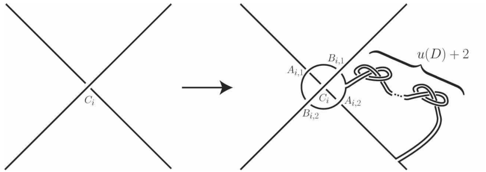
(b) Changing middle crossing



(c) After simplifying

- Only 25 knots in KnotInfo where $u(K)$ is known and $u(\mathcal{D}) > u(K)$
- 10_8 has a unique minimal diagram $\mathcal{D}$ with $u(\mathcal{D}) = 3$, but $u(10_8) = 2$
- If we change the middle crossing of 10_8, the resulting knot 6_2 has $u(6_2) = 1$, which can be seen after simplifying and changing the middle crossing
- By applying random Reidemeister moves, easy to find a diagram $\mathcal{D}'$ of 10_8 with $u(\mathcal{D}') = 2$

$u(K)$ versus $u(\mathcal{D})$

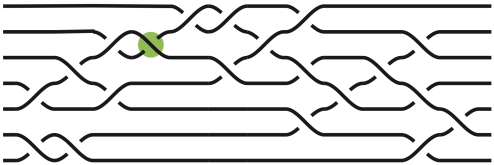


- Taniyama: given K and n, there is a diagram $\mathcal{D}$ of K with $u(\mathcal{D}) \geq n$
- Conjecture (Bernhard–Jablan): $\forall K$ has a minimal crossing number diagram $\mathcal{D}$ and a crossing c such that changing c gives a knot K' with

$$u(K') = u(K) - 1$$

- Brittenham and Hermiller: At least one of 13n3370, 12n288, 12n491, and 12n501 violates the conjecture

13n3370



- 13n3370 is the closure of the above 20-crossing braid
- Changing the green crossing gives 11n21 that has $u(11n21) = 1$
- So $u(13n3370) \leq 2$, but hard to find a diagram $\mathcal{D}$ with $u(\mathcal{D}) = 2$ using random Reidemeister moves

Computing the unknotting number

- Computing $u(\mathcal{D})$ is exponential in $c(\mathcal{D})$
- No algorithm known to compute $u(K)$
- Can often get a good upper bound on $u(K)$ by simplifying, changing a crossing such that the crossing number is minimal after simplifying and repeating
- $g_4(K) \leq u(K)$, so $\frac{|\sigma(K)|}{2}$, $\frac{|s(K)|}{2}$, $|\tau(K)|$, $|\nu(\pm K)|$ give computable lower bounds
- We know $u(K)$ if upper and lower bounds agree; e.g.,

$$u(T_{p,q}) = \frac{(p-1)(q-1)}{2}$$

Additivity of the unknotting number

- Conjecture: $u(K\#K') = u(K) + u(K')$
- Scharlemann: $u(K\#K') \geq 2$ if K, $K' \neq U$
- Alishahi–Eftekhari: $u(K\#T_{p,q}) \geq p - 1$ if $p < q$
- If, for example, $\text{sgn}(\sigma(K)) = \text{sgn}(\sigma(K'))$ and $u(K) = \frac{|\sigma(K)|}{2}$ and $u(K') = \frac{|\sigma(K')|}{2}$, then $u(K\#K') = u(K) + u(K')$
- Unknown whether $u(T_{2,3}\# - T_{2,5}) = u(T_{2,3}) + u(T_{2,5})$
- Strong conjecture: in every collection of unknotting crossing arcs for $K\#K'$, there is one that can be isotoped into K or K'

Figure: A crossing arc and the corresponding crossing change

Machine Learning

- Supervised Learning (SL): Given labelled data, learn a function that predicts the label, while minimising the error
- Can be classification (spam filter) or regression (predicting house prices, linear regression)
- Artificial Neural Network (ANN): A composition of affine maps and non-linearities. Trained using Stochastic Gradient Descent.
- Reinforcement Learning (RL): An agent learns to perform actions to maximise a reward (chess, Go, self-driving car, robot)
- Can be phrased as a Markov decision problem

Markov decision problems

- Markov decision problem: tuple (S, A, P_a, R_a), where
  - S: set of states
    (e.g., a knot diagram $\mathcal{D}$)
  - A_s: set of actions available from $s \in S$
    (changing a crossing)
  - $P_a(s, s')$: the probability that $a \in A_s$ leads to $s' \in S$
    (0 or 1 for a crossing change)
  - $R_a(s, s')$: immediate reward after transitioning from s to s' via action a
    (1 (or 0) if s' is a diagram of U and 0 (or -1) otherwise)

- Policy π: potentially probabilistic mapping from S to A

- Objective: Choose π to maximise the state value function

$$V^\pi(s) := E\left(\sum_{t=0}^{\infty} \gamma^t R_{\pi(s_t)}(s_t, s_{t+1})\right),$$

where $s_0 = s$, $s_{t+1} \sim P_{\pi(s_t)}(s_t, s_{t+1})$, and $\gamma \in [0, 1]$ discount factor

Q-learning

- Goal: Learn state-action value $Q(s, a)$, which is the expected reward if action a is taken in state s
- At time t, agent selects action a_t, observes reward r_t, and enters state s_{t+1}
- Initialise Q and update via Bellman equation:

$$Q^{\text{new}}(s_t, a_t) := Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a \in A_{s_{t+1}}} Q(s_{t+1}, a) - Q(s_t, a_t) \right),$$

  where $\alpha \in (0, 1]$ learning rate (step size)

- Selecting an action: exploration vs. exploitation
- ε-greedy policy: with probability ε, choose random action, with probability $1 - \varepsilon$, perform action a_t with maximal $Q(s_t, a_t)$
- Deep Q-learning: ANN $f \colon \mathbb{R}^S \to \mathbb{R}^A$, where $f(e_s) \cdot e_a = Q(s, a)$ for $s \in S$ and $a \in A_s$. Weights updated via Bellman equation

Importance Weighted Actor-Learner (IMPALA)

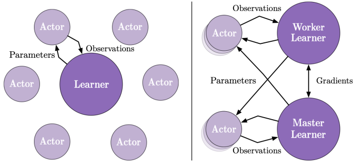


Figure 1. **Left: Single Learner.** Each *actor* generates trajectories and sends them via a queue to the *learner*. Before starting the next trajectory, *actor* retrieves the latest policy parameters from *learner*. **Right: Multiple Synchronous Learners.** Policy parameters are distributed across multiple *learners* that work synchronously.

- IMPALA [Espeholt et. al]: Distributed agent for parallelisation
- Learns policy π and value function V^{π} via stochastic gradient ascent
- Set of actors repeatedly generate trajectories of experience
- One or more learners use experience to learn π
- Policy of actors lags behind learner's

Supervised Learning and unknotting

- We trained a Random Forest classifier and an ANN (SL) to predict $u(\mathcal{D})$ from Alexander, Jones, writhe, and longitudinal translation
  (10k random diagram of 3–25 crossings, 80% accuracy, baseline 50%)
- In some diagrams, every crossing is in an unknotting set, in others, only small percentage
- We trained an SL agent to predict whether a crossing is in an unknotting set
  (100k random diagrams of 11–30 crossings, 85% accuracy, baseline 50%)

Reinforcement Learning and unknotting

- Goal: train an RL agent that performs crossing changes in a fixed diagram $\mathcal{D}$ to unknot it, giving an upper bound on $u(\mathcal{D})$
- Mostly used IMPALA agent
- Can determine $u(\mathcal{D})$ even when $c(\mathcal{D}) \approx 200$, when brute-forcing is not possible
- Representation: Knot invariants of diagram and all diagrams obtained by changing one crossing (diagrams hard to feed into ANN)

Features

- Using Alexander and Jones polynomial (coefficients, evaluations incl. derivatives, min and max degree), we got almost the same accuracy as with all features
- Other invariants either failed to compute for significant percentage of knots ($\geq 20\%$) or slow to compute for 100-crossing knots (HFK)
- Sum of absolute values of coefficients of Δ_K improves performance
- Jones + Alexander > Alexander only (esp. when forcing inter-component crossing changes for connected sums)
- V_K conjectured to detect unknot, Δ_K does not (algebraic unknotting number)
- One step lookahead: Agent computes Alexander/Jones polynomial of all knots obtained by changing one crossing as features
- Jones polynomial seems to contain yet unobserved unknotting information

Additivity of u and RL

- Obtained $u(K)$ for 31k random knots (10-60 crossings) and 26k QP knots (10-50 crossings) with $|\sigma(K)| \gg 0$, where upper bounds from RL agent and lower bounds from HFK coincide
- Searched for potential counterexamples to the additivity of u by overlaying summands with known u and, in some cases (100k), performing random Reidemeister moves (stochasticity vs. learning to find unknotting crossing arcs)
- Summands either from KnotInfo with known u, torus knots, or from the above 57k knots (random + QP)
- Limit of our RL $\approx$ 200 crossings, so too much mixing was not always feasible
- Have not found a counterexample to $u(K \# K') = u(K) + u(K')$

Strong conjecture

- Strong conjecture: in every collection of unknotting arcs for $K\#K'$, there is one that can be isotoped into K or K'
- Inter-component crossing change: results in a knot that is not a connected sum; e.g., not hyperbolic
- Agent performed several inter-component crossing changes
- Counterexamples to the strong conjecture by undoing all in-component crossing changes

A counterexample to the strong conjecture

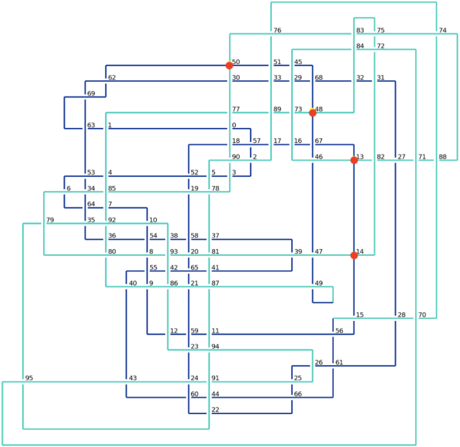
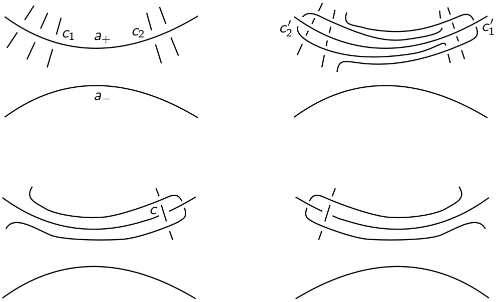


Figure: Unknotted by 13, 14, 48, 50

Strong counterexample

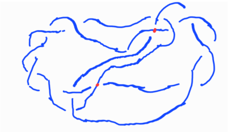


Theorem

Suppose that the prime knots K_1 and K_2 in S^3 are not 2-bridge. Suppose that, for $i \in \{1, 2\}$, there is a set of $u(K_i)$ crossing changes to K_i taking it to the unknot, with the property that changing any one of these crossings does not produce the connected sum of K_i and a non-trivial knot. Furthermore, assume that $u(K_1) > 1$ or $u(K_2) > 1$. Then there is a diagram of $K_1 \# K_2$ and a set C of unknotting crossings of size $u(K_1) + u(K_2)$ such that changing any crossing in C results in a prime knot.

New unknotting numbers assuming additivity of u

- If we assume u is additive and consider knots appearing along minimal unknotting trajectories of connected sums, we obtain the unknotting number of 43 knots K with $c(K) \leq 12$ that were unknown
- In all these examples, $u(K)$ was equal to the KnotInfo upper bound
- 39 of these knots K have a crossing change in their KnotInfo diagram $\mathcal{D}$ that results in a connected sum $K_0 \# K_1$ with $u(\mathcal{D}) = u(K_0) + u(K_1) - 1$
- We have found by hand a diagram for 12a981 where two crossing changes yield a diagram $\mathcal{D}$ of $T_{2,7} \# - T_{2,5}$ with $u(\mathcal{D}) = u(T_{2,7}) + u(T_{2,5}) - 2$
- Remaining 3 knots: 12a898, 12a917, 12a999



12a981

Hard unknot diagrams

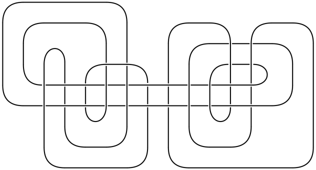
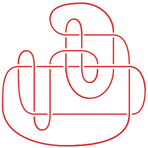


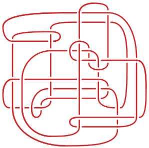
Figure 8. A 28-crossing diagram D_{28} of the unknot requiring three extra crossings.

- We say that a diagram of the unknot is hard if, in any sequence of Reidemeister moves to the trivial diagram, the crossing number has to first increase before it decreases
- 11 hard unknot diagrams and 2 special infinite families from the literature
  [Burton, Chang, Löffler, Mesmay, Maria, Schleimer, Sedgwick, Spreer. Hard diagrams of the unknot., *Exp. Math.*, 2023]
- Tried to construct using setter/solver

Hard unknot diagrams



(a) simplify('level') hard



(b) simplify('global') hard

- While running the unknotting agent, we have found $\approx$ 5.9M unknot diagrams that SnapPy could not simplify using simplify('level') (random R3 moves + R1 and R2)
- Out of 5.9M (between 9 and 75 crossings), verified that $\geq$ 2.46M are hard and not related by R3 moves
- 382 diagrams survive even 10^6 subsequent simplify('global') attempts: also picks up a strand and puts it elsewhere (pass move) to reduce $c(K)$
- Potential counterexamples to unknotting algorithm candidates

Thank you for your attention...

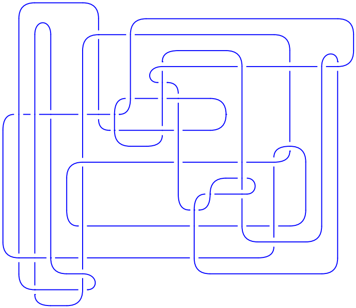


Figure: A 42-crossing hard unknot diagram with 6225 R3-equivalent diagrams that we have not been able to simplify by calling SnapPy's 'global' heuristic 100 times.