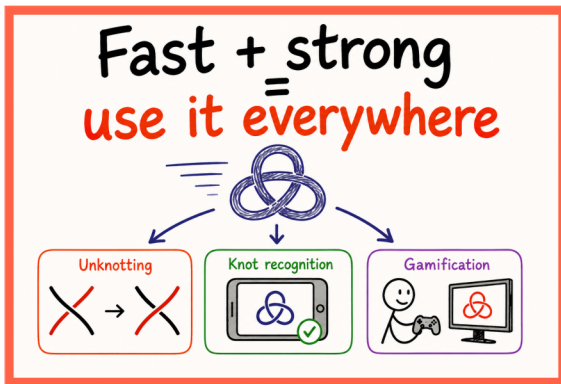


Applications of a fast and strong knot invariant

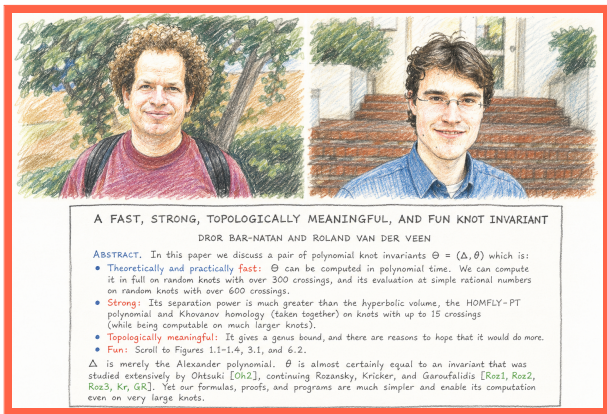
Or: Once an invariant is cheap, use it everywhere

It's also topologically meaningful and fun, but I wouldn't know that



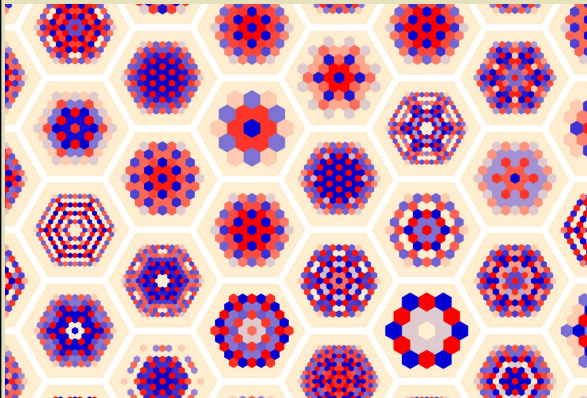
I report on work of many people (Bar-Natan–van der Veen, Brittenham, Hermiller, Applebaum et al., Dranowski, Jie, Kabkov, ...)

Why I like Θ



- ▶ Why I like Θ Fast + strong = use it everywhere
- ▶ Three examples Unknotting, knot picture recognition, gamification
- ▶ Moral Cheap + strong invariants become infrastructure

Fast + strong = apply it



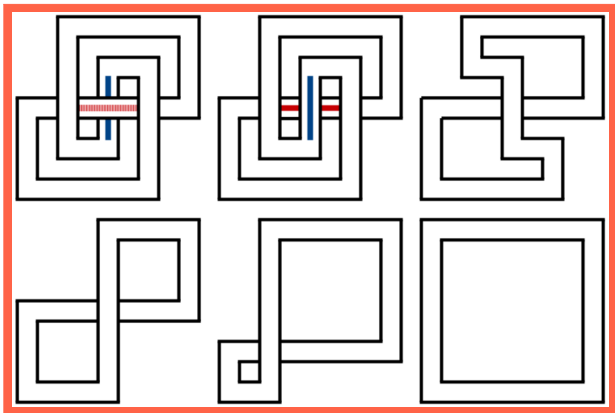
That is the whole talk:
once a good invariant is cheap enough,
it stops being a theorem only
and becomes a tool.

▶ Why I like

▶ Three ex

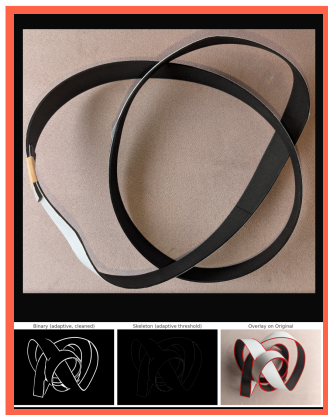
▶ Moral Cheap + strong invariants become infrastructure

Why I like Θ



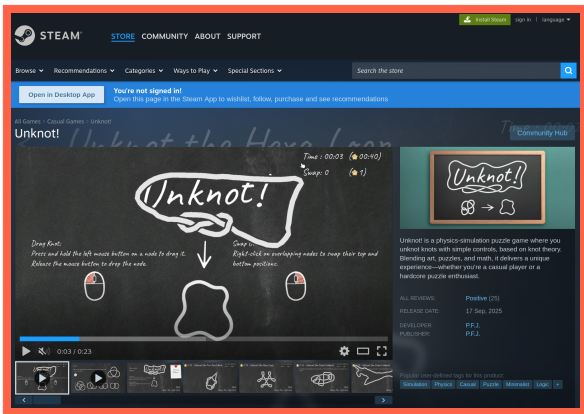
- ▶ Example 1 Unknotting (numbers): that old nasty problem
- ▶ Point The machine tries bad-looking routes
- ▶ Θ role Fast sanity check for the route

Why I like Θ



-
- ▶ **Example 2** Pictures: Identifying knots on your phone
 - ▶ **Point** ML is good with pixels but bad at topology
 - ▶ **Θ role** Cheap + strong invariants make guesses safer

Why I like Θ

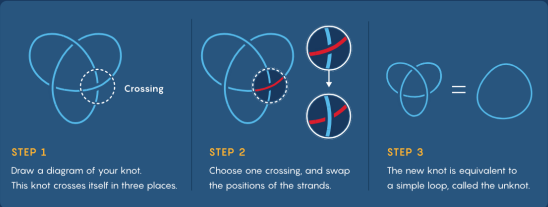


- ▶ **Example 3** Games: players try to unknot things
- ▶ **Point** In 2026, humans are still better than AI, but we need many
- ▶ **Θ role** Cheap + strong verification before belief

Unknotting is hard

What is an Unknotting Number?

To compare and measure the complexity of knots, mathematicians compute their unknotting numbers — the minimum number of steps it takes to transform a knot into a simple loop.



STEP 1
Draw a diagram of your knot.
This knot crosses itself in three places.

STEP 2
Choose one crossing, and swap the positions of the strands.

STEP 3
The new knot is equivalent to a simple loop, called the unknot.

The unknotting number of the original knot is 1, because only one crossing change is needed to transform it into the unknot. But more complicated knots require more crossing changes and therefore have larger unknotting numbers.

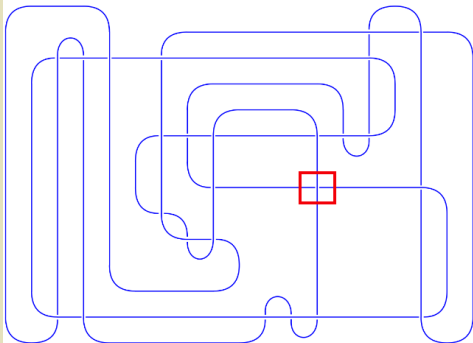
—
Mark Behar/Quanta Magazine

- ▶ u = minimal crossing flops needed to unknot
- ▶ **Dream** Find u on a small diagram (this is wrong!)
- ▶ **Problem** This may require making things worse

Unknotting is hard

Why hard?

An inflation of $4_1 \# 9_{10}$ where u is not additive



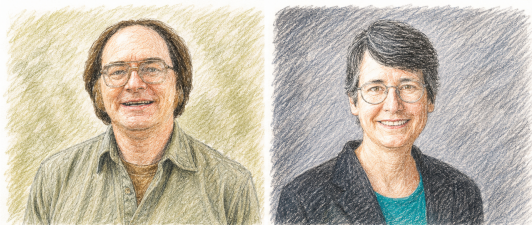
Human intuition is local;
unknotting often is not.
The right move may look wrong.

► u = minimal

► Dream Find

► Problem This may require making things worse

Unknotting is hard



UNKNOTTING NUMBER IS NOT ADDITIVE UNDER CONNECTED SUM

MARK BRITTENHAM AND SUSAN HERMILLER

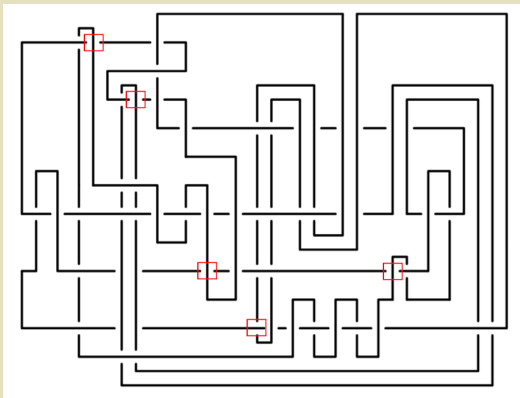
ABSTRACT. We give the first examples of a pair of knots K_1, K_2 in the 3-sphere for which their unknotting numbers satisfy $u(K_1 \# K_2) < u(K_1) + u(K_2)$. This answers question 1.69(B) from Kirby's problem list [24], in the negative.

Mathematics Subject Classification 2020: 57K10, 57K31

- ▶ Open for a long time Is u additive under connected sum?
- ▶ No! Brittenham–Hermiller: obvious diagrams can hide the route
- ▶ Key Same local-versus-global headache

Unknotting is hard

$7_1 \# \text{mir}(7_1)$



The minimal known diagram with $u = 5$ has 56 crossings.
Note the one funny place to swap crossings.
That is not a nice by-hand calculation.

► Open for

► No! Bri

► Key Same local versus global heuristics

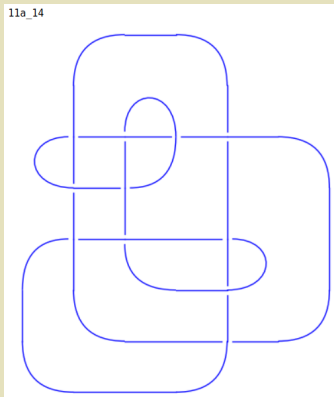
Unknotting is hard



- ▶ **Input** Start with known lower bound, unclear upper bound
- ▶ **Goal** Improve the upper bound by finding a better route
- ▶ **Key** Do not insist on the visible diagram

Why start from a list?

11a_14

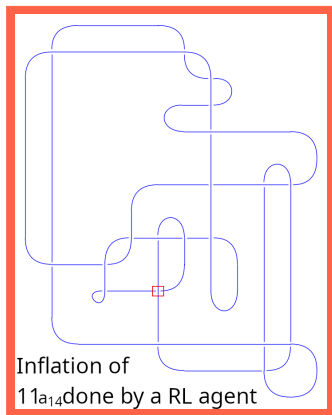


Target knots with no sharp upper bound.

Example: The knot $11a_{14}$ had $u \in [2, 3]$.

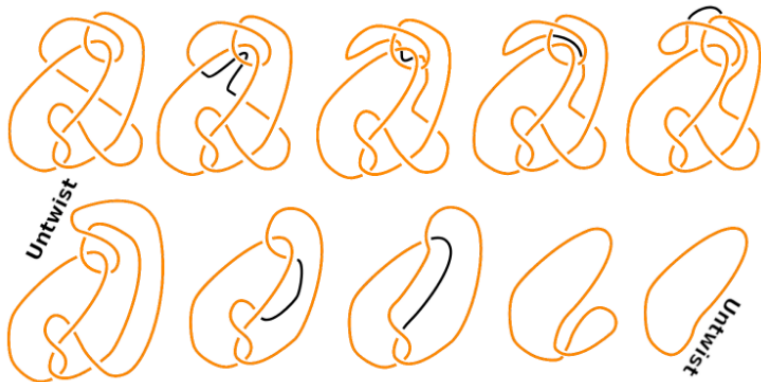
Every success gives new information (self-cleaning because we update the source list).

Unknotting is hard



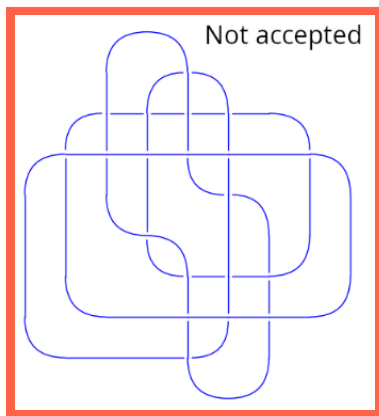
- ▶ **Step 1** Inflate the diagram, but not randomly
- ▶ **RL agent 1** Guess promising diagrams/crossings
- ▶ **Hope** A larger diagram may reveal the right flop

Why inflate first?



Sometimes the visible diagram is stuck.
The decisive crossing change
may not be visible yet.

Unknotting is hard



-
- ▶ **Step 2** Flop one chosen crossing
 - ▶ **Reality** Still too messy to read (too many crossings; exceeds our list)
 - ▶ **Step 3** Pass it to the unknotter (RL agent 2; give me a second)

Unknotting is hard

After the flop

10_1	$[-2, 8, 1, -1, 2, -2, \rightarrow]$	$[[2,11,3,12],[4,20, \rightarrow]$	1
10_2	$[1, 11, 1, -1, 2, -2, \rightarrow]$	$[[1,13,2,12],[3,14, \rightarrow]$	3
10_3	$[-4, 6, 1, -1, 2, -3, \rightarrow]$	$[[2,10,3,9],[4,17,5, \rightarrow]$	2
10_4	$[-5, 5, 1, -2, 3, -3, \rightarrow]$	$[[2,13,3,14],[4,11, \rightarrow]$	2
10_5	$[-9, 1, -1, 2, -3, 4, \rightarrow]$	$[[2,13,3,14],[4,17, \rightarrow]$	2
10_6	$[0, 10, 1, -1, 3, -4, \rightarrow]$	$[[2,11,3,12],[4,16, \rightarrow]$	$[2,3]$
10_7	$[-1, 9, 1, -2, 4, -5, \rightarrow]$	$[[1,12,2,13],[3,15, \rightarrow]$	1
10_8	$[-2, 8, 1, -1, 2, -3, \rightarrow]$	$[[1,12,2,13],[3,10, \rightarrow]$	2
10_9	$[-3, 7, 1, -2, 3, -4, \rightarrow]$	$[[1,13,2,12],[3,16, \rightarrow]$	1
10_10	$[-7, 3, -1, 2, -3, 5, \rightarrow]$	$[[1,11,2,10],[3,18, \rightarrow]$	1
10_11	$[-3, 7, 1, -1, 3, -5, \rightarrow]$	$[[2,15,3,16],[4,13, \rightarrow]$	$[2,3]$
10_12	$[-8, 2, -1, 2, -4, 6, \rightarrow]$	$[[2,11,3,12],[4,18, \rightarrow]$	2
10_13	$[-4, 6, 1, -2, 5, -7, \rightarrow]$	$[[2,17,3,18],[4,8,5, \rightarrow]$	2
10_14	$[0, 10, 1, -2, 4, -6, \rightarrow]$	$[[1,14,2,15],[3,17, \rightarrow]$	2
10_15	$[-6, 4, -1, 2, -4, 6, \rightarrow]$	$[[2,6,3,5],[4,16,5, \rightarrow]$	2
10_16	$[-3, 7, 1, -2, 4, -5, \rightarrow]$	$[[1,10,2,11],[3,15, \rightarrow]$	2
10_17	$[-5, 5, -1, 2, -3, 5, \rightarrow]$	$[[6,2,7,1],[12,4,13, \rightarrow]$	1

Even the correct crossing change usually leaves a messy diagram, and we need to make sure it appears on our list.

▶ Step 2 Flop

▶ Reality Still

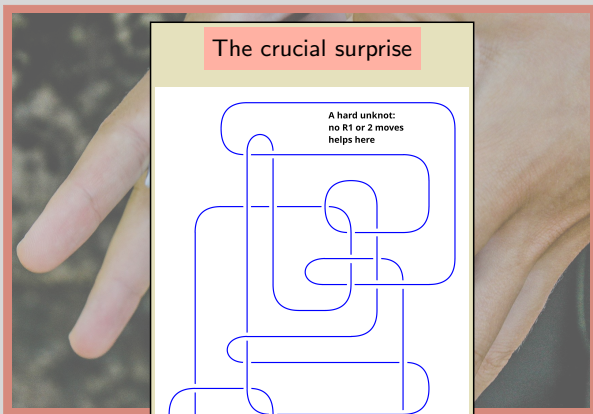
▶ Step 3 Pass it to the unknotter (RE agent 2, give me a second)

Unknotting is hard



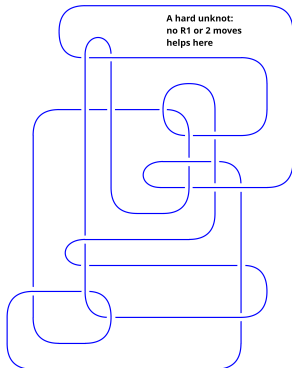
- ▶ **Unknotter** RL learns from rewards and penalties
- ▶ **Moves** Simplify, shuffle, sometimes add crossings
- ▶ **Subtlety** Good moves may look bad immediately

Unknotting is hard



The crucial surprise

A hard unknot:
no R1 or 2 moves
helps here



Greedy simplification can fail.
Adding crossings first can be
the winning move.

► Unknotter RL learns

► Moves Simplify, shuffle

► Subtlety Good moves may look bad immediately

From Applebaum et al.:

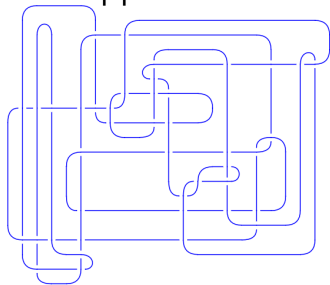


FIGURE 17. A 42-crossing hard unknot diagram with 6225 R3-equivalent diagrams that we have not been able to simplify by calling SnapPy's 'global' heuristic 10000 times.

- ▶ **Benchmarks** Hard unknots are the right stress test
- ▶ **Result** It finds nontrivial reduction sequences ($\approx 95\%$ of the time)
- ▶ **Message** Adding crossings first is the point

Unknotting is hard

Hard unknot punchline

run	resolved_count	resolved_pct	median_min_crossings	max_min_crossings	
0	1	367	95.324675	0.0	46
1	2	367	95.324675	0.0	38
2	3	365	94.805195	0.0	39
3	4	357	92.727273	0.0	46
4	5	357	92.727273	0.0	41
5	6	360	93.506494	0.0	46
6	7	367	95.324675	0.0	39
7	8	368	95.584416	0.0	46
8	9	365	94.805195	0.0	46
9	10	368	95.584416	0.0	40

Not just a toy
(though I like the toy).
It works on serious benchmarks.

Unknotting is hard



A FAST, STRONG, TOPOLOGICALLY MEANINGFUL, AND FUN KNOT INVARIANT

DROR BAR-NATAN AND ROLAND VAN DER VEEN

ABSTRACT. In this paper we discuss a pair of polynomial knot invariants $\Theta = (\Delta, \theta)$ which is:

- Theoretically and practically fast: Θ can be computed in polynomial time. We can compute it in full on random knots with over 300 crossings, and its evaluation at simple rational numbers on random knots with over 600 crossings.
- Strong: Its separation power is much greater than the hyperbolic volume, the HOMFLY-PT polynomial and Khovanov homology (taken together) on knots with up to 15 crossings (while being computable on much larger knots).
- Topologically meaningful: It gives a genus bound, and there are reasons to hope that it would do more.
- Fun: Scroll to Figures 1.1-1.4, 3.1, and 6.2.

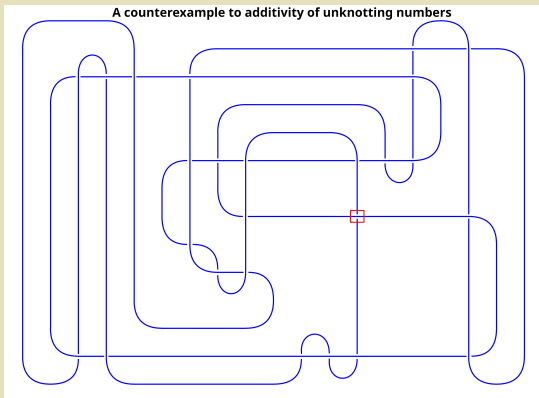
Δ is merely the Alexander polynomial. θ is almost certainly equal to an invariant that was studied extensively by Ohtsuki [Oht2], continuing Rozansky, Kricker, and Garoufalidis [Roz1, Roz2, Roz3, Kr, GR]. Yet our formulas, proofs, and programs are much simpler and enable its computation even on very large knots.

- ▶ **Θ** role Cheap + good identification of knots
- ▶ Cheap You need to run this 10k times
- ▶ Reliable Avoid duplicates

Unknotting is hard

Result

A counterexample to additivity of unknotting numbers

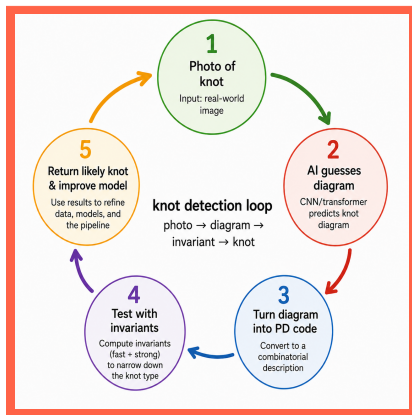


The pipeline improved unknotting numbers of $\approx 5k$ knots.
It rediscovered the Brittenham–Hermiller result.

Right now it looks for new counterexamples to $u(K\#L) = u(K) + u(L)$.

Reliable Avoid duplicates

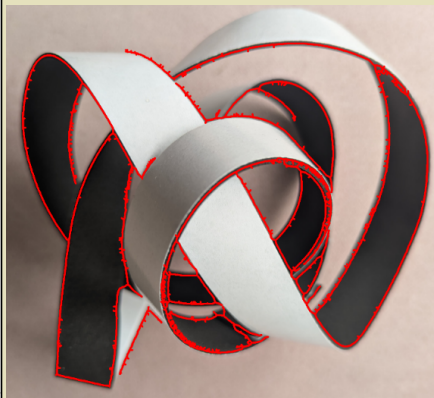
Knot picture recognition



- ▶ **Dream in progress** Take a photo and your phone recognizes the knot
- ▶ **First stage** Use CNNs/transformers for the pixels
- ▶ **Second stage** Reconstruct PD code, compute invariants

Knot picture recognition

Life and math



► Dream in progress

► First stage Use

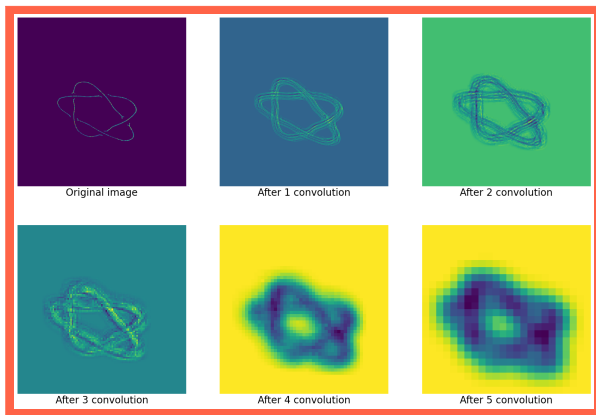
► Second stage

Easy: good pictures of small knots.
Difficult: bad pictures of small knots.
Very difficult: big knots.

Reconstruct PD code, compute invariants

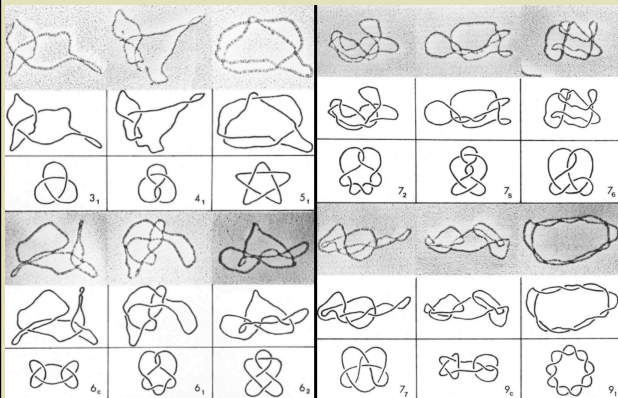
minimizes the knot

Knot picture recognition



- ▶ ML part Good at noisy visual input
- ▶ Topology part Good at actual knot distinctions
- ▶ Slogan ML guesses; invariants keep it honest

Dreamy goal




Currently DNA knots (EM images) are identified using human resources. Well, do we need to do that?

▶ ML part

▶ Topology part Good at actual knot distinctions

▶ Slogan ML guesses; invariants keep it honest

Knot picture recognition



A FAST, STRONG, TOPOLOGICALLY MEANINGFUL, AND FUN KNOT INVARIANT
DROR BAR-NATAN AND ROLAND VAN DER VEEN

ABSTRACT. In this paper we discuss a pair of polynomial knot invariants $\Theta = (\Delta, \theta)$ which is:


- **Theoretically and practically fast:** Θ can be computed in polynomial time. We can compute it in full on random knots with over 300 crossings, and its evaluation at simple rational numbers on random knots with over 600 crossings.
- **Strong:** Its separation power is much greater than the hyperbolic volume, the HOMFLY-PT polynomial and Khovanov homology (taken together) on knots with up to 15 crossings.
- **Topologically meaningful:** It gives a genus bound, and there are reasons to hope that it would do more.
- **Fun:** Scroll to Figures 1.1-1.4, 3.1, and 6.2.

Δ is merely the Alexander polynomial. θ is almost certainly equal to an invariant that was studied extensively by Ohtsuki [Oh2], continuing Rozansky, Kricker, and Garoufalidis [Roz1, Roz2, Roz3, Kr, GR]. Yet our formulas, proofs, and programs are much simpler and enable its computation even on very large knots.

- ▶ Θ role A cheap + strong invariant gives a fast + reliable filter
- ▶ Why cheap matters Attention span of users is limited
- ▶ Why strong matters Can push recognition to higher crossing range

Knot picture recognition

Where Θ fits



A FAST, STRONG KNOT INVARIANT

ABSTRACT. In this paper we study the complexity of computing the invariant Δ in full on rational numbers on real numbers on real numbers.

- **Theoretically** Δ is merely the Alexander polynomial studied extensively by mathematicians. Yet our formulas, provide a more efficient way to compute it.
- **Strong:** Its complexity is polynomial and its computation is topologically meaningful.
- **Fun:** Scroll to see the knot picture recognition.

... which is: an computer rational ...

... OMFY-PT ...

... it would do more.

... that was ... , Roz2, Roz3, Kr, GRJ]. very large knots.

► Θ role A cheap +

► Why cheap matters

► Why strong matters

This is literally designed to be on your phone, so no resources to waste.

... liable filter

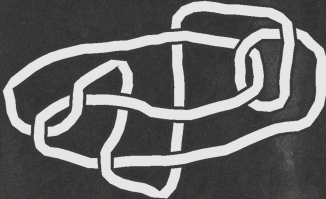
... d

Can push recognition to higher crossing range

🌟 1.15 - Unknot the Culprit Unknot

Time: 00:03 (🌟 01:00)
Swap: 0 (🌟 0)

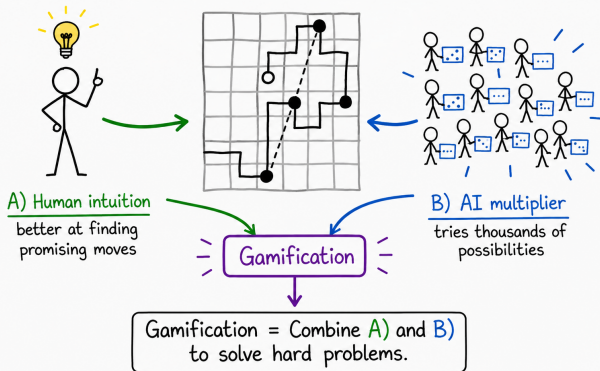
Ken Millett coined "the Culprit" to test algorithms—it famously fooled early programs into thinking it was knotted.



Back
Retry Flip Load Save

- ▶ **Unknot!** A physics puzzle game about unknotting
- ▶ **Opportunity** Players generate human unknotting attempts
- ▶ **Useful data?** We currently collect it, but we need to improve it

Why games?



A) Human intuition is still better than AI.

B) AI has the advantage of being like 1000 undergraduates.

Gamification = Use A) with the multiplier in B).

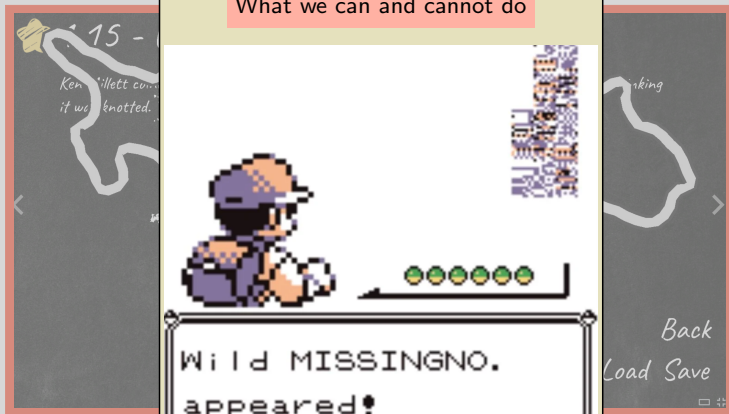
► Unkn

► Opportunity Players generate human unknotting attempts

► Useful data? We currently collect it, but we need to improve it

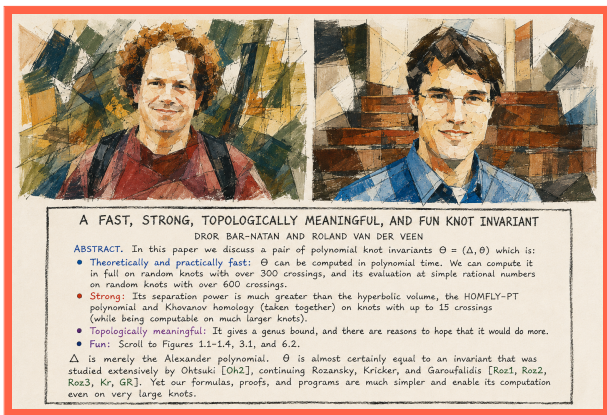


- ▶ **Plan** Collect attempted unknotting routes (semi-working)
- ▶ **Data** Which crossings do humans try, etc.
- ▶ **Hope** Human play may reveal new math



- ▶ **Plan** Collect at (g)
- ▶ **Data** Which cr
- ▶ **Hope** Human play may reveal new math

This is physics-based, but has glitches.
To make this foolproof
we have to add an invariant check.



- ▶ **Θ role** Cheap + reliable check that the knot did not change
- ▶ **Bug filter** Physics engines and input can lie
- ▶ **Fast** Don't let the player wait!

Still a long way to go



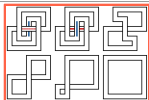
Well, still: This is supposed to be presented during the Tokyo Game Show 2026, so we hope for more players.

▶ **Θ role** Cheap

▶ **Bug filter** Ph

▶ **Fast** Don't let the player wait!

Why I like Θ



- ▶ **Example 1** Unknotting (numbers): that old nasty problem
- ▶ **Point** The machine tries bad-looking routes
- ▶ **Q rule** Fast sanity check for the route

Applications of a fast and strong knot invariant Part 1, string 1, slide 3 comparison July 2026 9 / 6

Why I like Θ



- ▶ **Example 2** Pictures: Identifying knots on your phone
- ▶ **Point** ML is good with pixels but bad at topology
- ▶ **Q rule** Cheap + strong invariants make games safer

Applications of a fast and strong knot invariant Part 1, string 1, slide 5 comparison July 2026 9 / 6

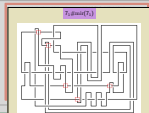
Why I like Θ



- ▶ **Example 3** Games: players try to unknot things
- ▶ **Point** In 2026, humans are still better than AI, but we need many
- ▶ **Q rule** Cheap + strong verification before belief

Applications of a fast and strong knot invariant Part 1, string 1, slide 6 comparison July 2026 9 / 6

Unknotting is hard



- ▶ **Open to** The minimal known diagram with $w = 5$ has 58 crossings. Note the one funny place to loop crossings. That is not a nice-by-hand calculation.
- ▶ **Not to**
- ▶ **Q rule**

Applications of a fast and strong knot invariant Part 1, string 1, slide 7 comparison July 2026 9 / 6

Unknotting is hard



- ▶ **Input** Start with known lower bound, unclear upper bound
- ▶ **Goal** Improve the upper bound by finding a better route
- ▶ **Key** Do not insist on the visible diagram

Applications of a fast and strong knot invariant Part 1, string 1, slide 8 comparison July 2026 9 / 6

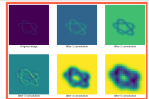
Unknotting is hard



- ▶ Sometimes the visible diagrams is stuck. The decisive crossing change may not be visible yet.
- ▶
- ▶

Applications of a fast and strong knot invariant Part 1, string 1, slide 9 comparison July 2026 9 / 6

Knot picture recognition



- ▶ **ML part** Good at noisy visual input
- ▶ **Topology part** Good at actual knot distinctions
- ▶ **Slogan** ML guesses, invariants keep it honest

Applications of a fast and strong knot invariant Part 1, string 1, slide 10 comparison July 2026 9 / 6

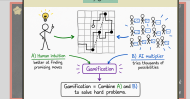
Knot picture recognition



- ▶ **Q rule** A cheap + strong invariant is a cheap + strong invariant
- ▶ **Why cheap matters** This is literally designed to be on your phone, so no resources to waste.
- ▶ **Why strong matters** Can push recognition to higher crossing range

Applications of a fast and strong knot invariant Part 1, string 1, slide 11 comparison July 2026 9 / 6

Gameficat

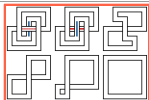


- ▶ **Unknotting** A) Human intuition is still better than AI. B) AI has the advantage of being like 1000 undergraduates. C) Comparison = Use A) with the multiplier in B).
- ▶ **Useful data?** We currently collect it, but we need to improve it

Applications of a fast and strong knot invariant Part 1, string 1, slide 12 comparison July 2026 9 / 6

There is still much to do...

Why I like Θ



- ▶ **Example 1** Unknotting (numbers): that old nasty problem
- ▶ **Point** The machine tries bad-looking routes
- ▶ **Q rule** Fast sanity check for the route

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

Why I like Θ



- ▶ **Example 2** Pictures: Identifying knots on your phone
- ▶ **Point** ML is good with pixels but bad at topology
- ▶ **Q rule** Cheap + strong invariants make guess safer

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

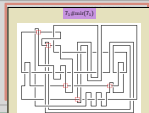
Why I like Θ



- ▶ **Example 3** Games: players try to unknot things
- ▶ **Point** In 2026, humans are still better than AI, but we need many
- ▶ **Q rule** Cheap + strong verification before belief

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

Unknotting is hard



- ▶ **Open to** The minimal known diagram with $w = 5$ has 58 crossings. Note the one funny place to loop crossings. That is not a nice-by-hand calculation.
- ▶ **Not to**
- ▶ **Key**

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

Unknotting is hard



- ▶ **Input** Start with known lower bound, unclear upper bound
- ▶ **Goal** Improve the upper bound by finding a better route
- ▶ **Key** Do not insist on the visible diagram

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

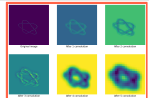
Unknotting is hard



- ▶ Sometimes the visible diagrams is stuck. The decisive crossing change may not be visible yet.

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

Knot picture recognition



- ▶ **ML part** Good at noisy visual input
- ▶ **Topology part** Good at actual knot distinctions
- ▶ **Slogan** ML guesses, invariants keep it honest

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

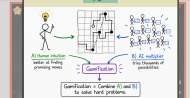
Knot picture recognition



- ▶ **Q rule** A cheap + strong invariant is a good resource to assist
- ▶ **Why cheap matters** Can push recognition to higher crossing range

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

Classification



- ▶ **Unknotting** Human intuition is still better than AI
- ▶ **Classification** AI has the advantage of being like 1000 undergraduates. Classification \Leftarrow Use AI with the multiplier in B).
- ▶ **Useful data?** We currently collect it, but we need to improve it

Applications of a fast and strong knot invariant Part 1, slide 7, see 5 comparisons July 2026 6 / 6

Thanks for your attention!