Fixed-parameter tractable computation of Reshetikhin–Turaev knot polynomials via tensor networks

Shana Y. Li

University of Illinois, Urbana-Champaign

Oct 2026

Table of Contents

1 Reshetikhin–Turaev functor

2 Tensor network

3 Implications and applications

Table of Contents

1 Reshetikhin–Turaev functor

2 Tensor network

3 Implications and applications

Category of oriented tangles:

- Objects: Signed dots along a line.

Category of oriented tangles:

- Objects: Signed dots along a line.
- Morphisms: Oriented tangles attached to two lines.

Category of oriented tangles:

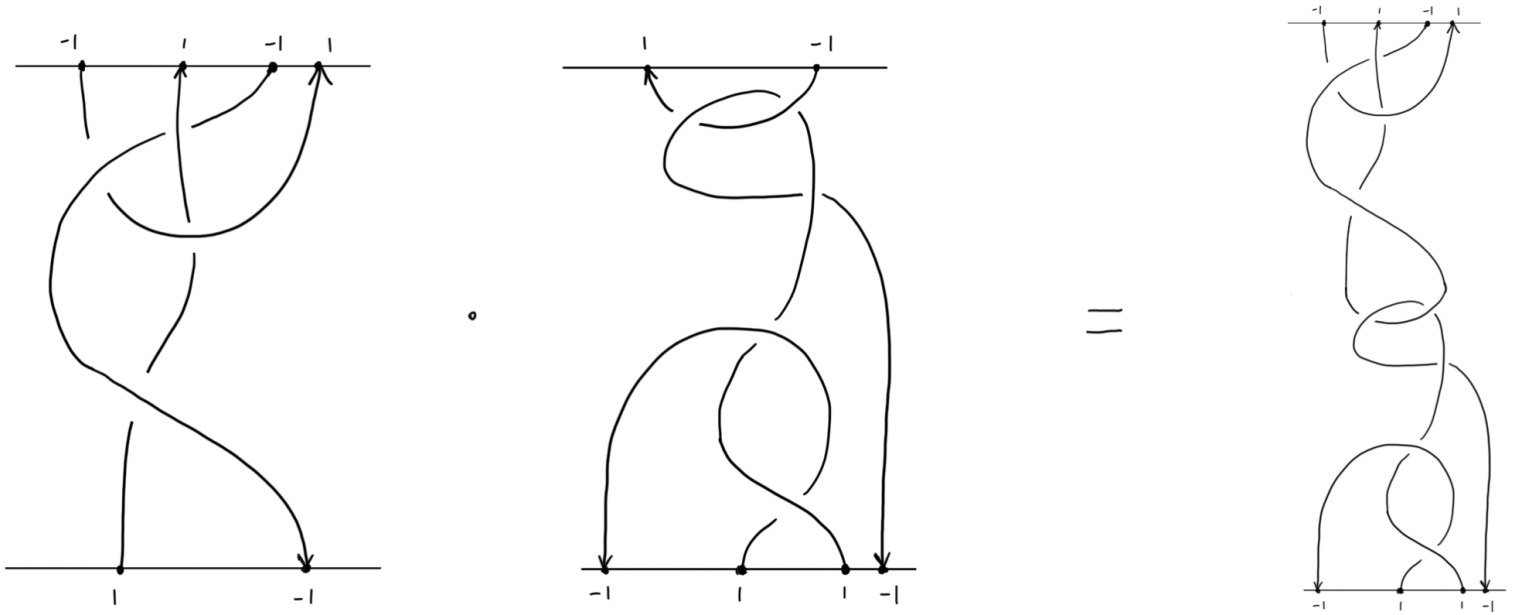
- Objects: Signed dots along a line.
- Morphisms: Oriented tangles attached to two lines.
- Composition: Gluing of oriented tangles.

Category of oriented tangles:

- Objects: Signed dots along a line.
- Morphisms: Oriented tangles attached to two lines.
- Composition: Gluing of oriented tangles.

Category of oriented tangles:

- Objects: Signed dots along a line.
- Morphisms: Oriented tangles attached to two lines.
- Composition: Gluing of oriented tangles.



Reshetikhin–Turaev functor: oriented tangles $\mapsto$ vector spaces

Reshetikhin–Turaev functor: oriented tangles $\mapsto$ vector spaces

Ingredients:

- A vector space V over a field $\mathbb{F}$.
- An R-matrix $R \in \mathrm{Aut}(V \otimes V)$.
- A ribbon element $h \in \mathrm{Aut}(V)$.

Reshetikhin–Turaev functor: oriented tangles $\mapsto$ vector spaces
Ingredients:

- A vector space V over a field $\mathbb{F}$.
- An R-matrix $R \in \mathrm{Aut}(V \otimes V)$.
- A ribbon element $h \in \mathrm{Aut}(V)$.

satisfying *some* conditions, e.g. the Yang–Baxter equation:

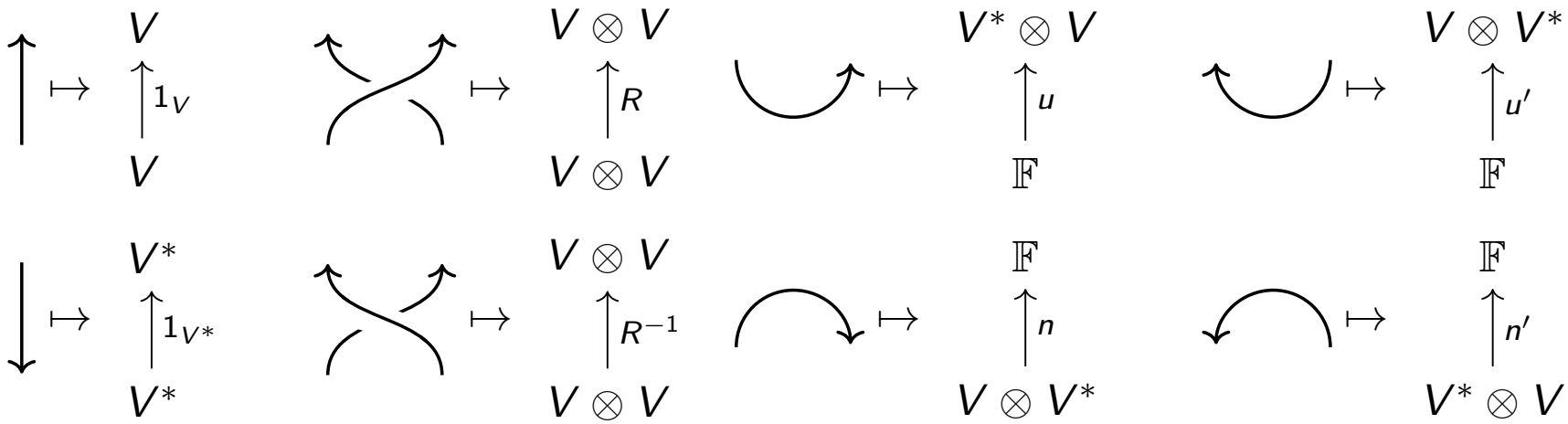
Reshetikhin–Turaev functor: oriented tangles $\mapsto$ vector spaces
Ingredients:

- A vector space V over a field $\mathbb{F}$.
- An R-matrix $R \in \mathrm{Aut}(V \otimes V)$.
- A ribbon element $h \in \mathrm{Aut}(V)$.

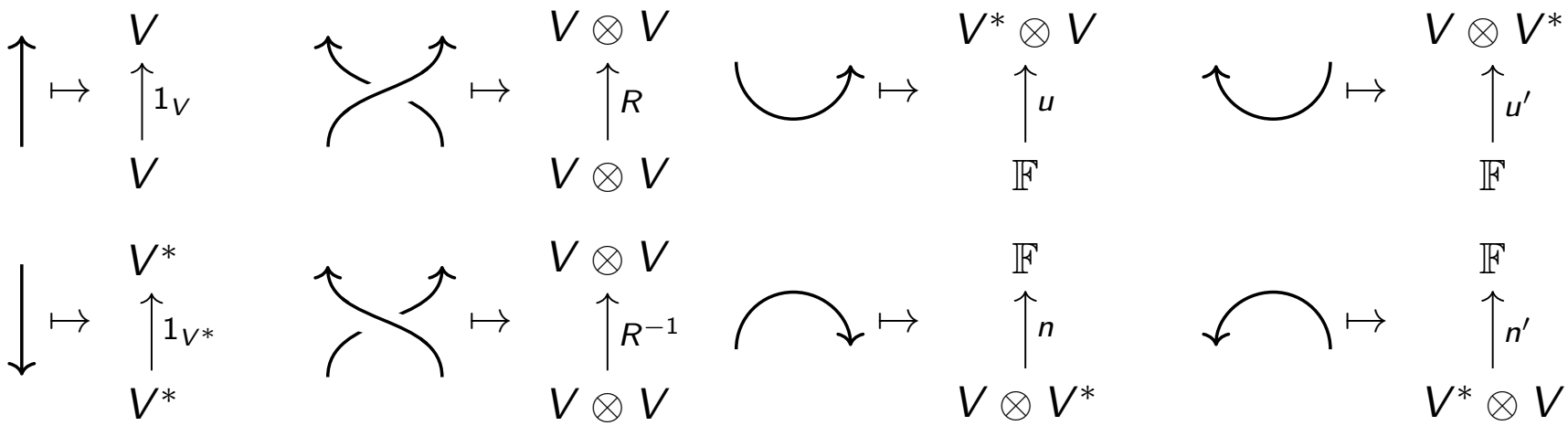
satisfying *some* conditions, e.g. the Yang–Baxter equation:

$$(I \otimes R) \circ (R \otimes I) \circ (I \otimes R) = (R \otimes I) \circ (I \otimes R) \circ (R \otimes I).$$

Reshetikhin–Turaev functor: oriented tangles $\mapsto$ vector spaces



Reshetikhin–Turaev functor: oriented tangles $\mapsto$ vector spaces

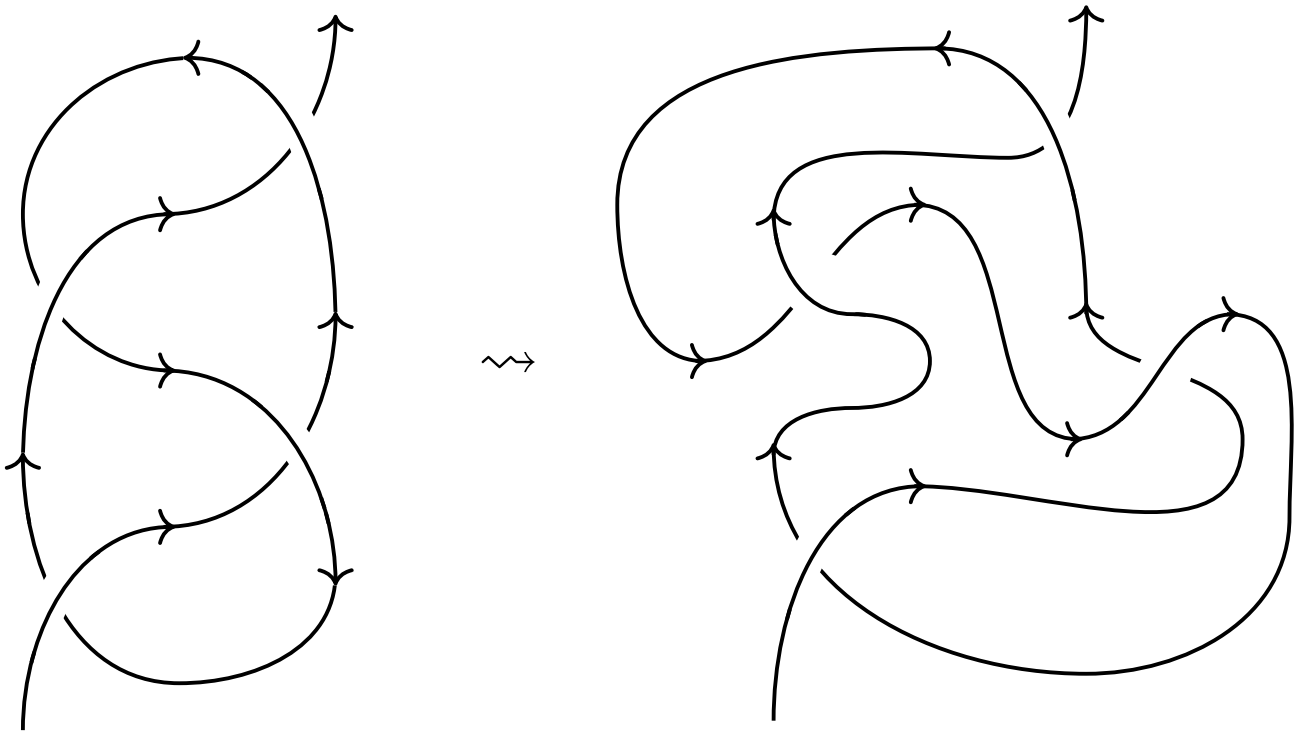


$$u(1) = \sum_i e_i^* \otimes h(e_i), \quad u'(1) = \sum_i e_i \otimes e_i^*,$$

$$n(x \otimes f) = f(h^{-1}(x)), \quad n'(f \otimes x) = f(x)$$

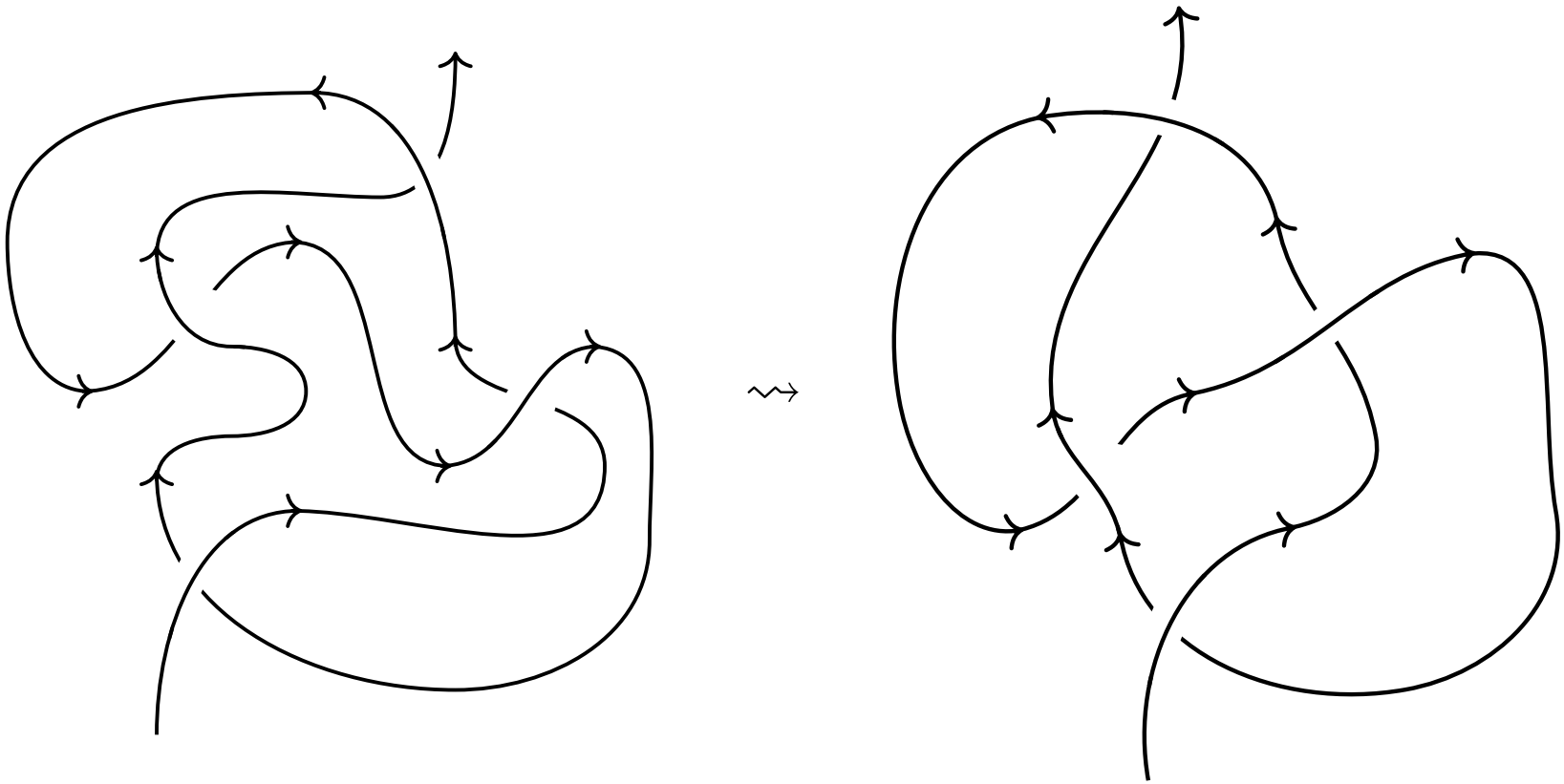
Reshetikhin–Turaev: oriented tangles $\mapsto$ vector spaces

Example: the 4_1 knot



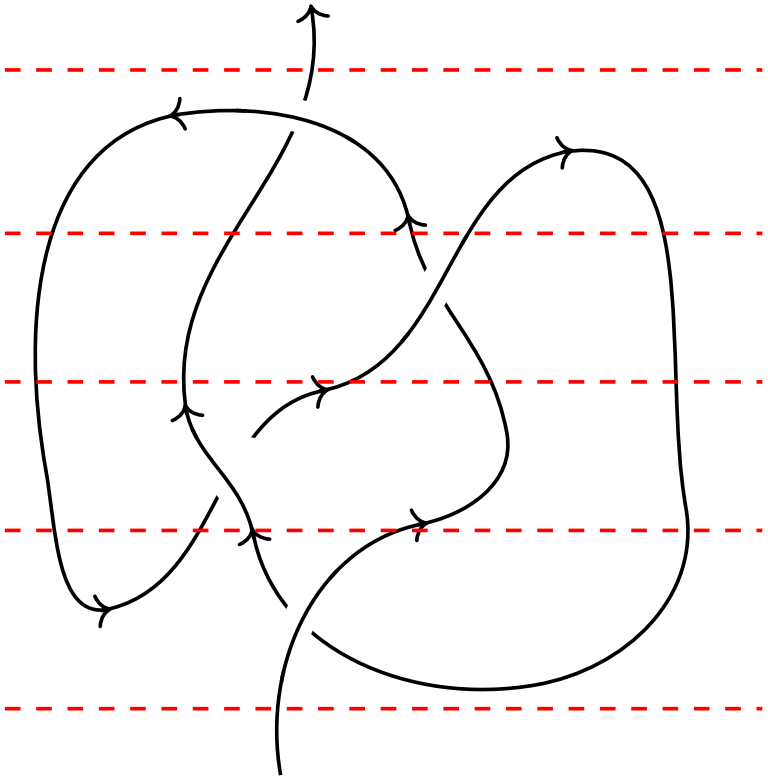
Reshetikhin–Turaev: oriented tangles $\mapsto$ vector spaces

Example: the 4_1 knot



Reshetikhin–Turaev: oriented tangles $\mapsto$ vector spaces

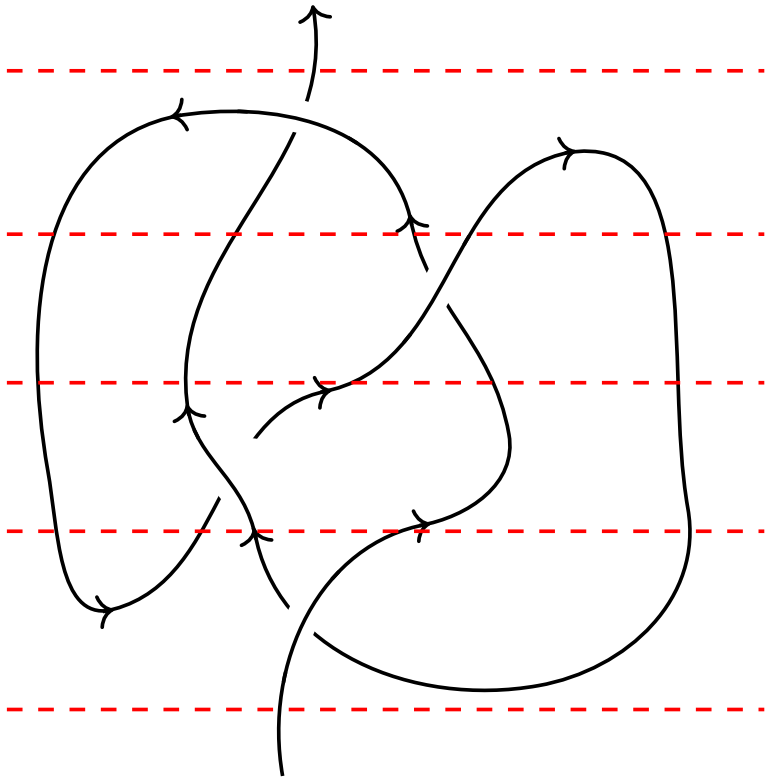
Example: the 4_1 knot



$\mapsto$

$$
\begin{array}{ll}
V & \\
\uparrow & \left(\left(n' \otimes 1_V\right) \circ \left(1_V \otimes R^{-1}\right)\right) \otimes n \\
V^{\otimes 5} & \\
\uparrow & 1_V \otimes 1_V \otimes R \otimes 1_V \\
V^{\otimes 5} & \\
\uparrow & 1_V \otimes R^{-1} \otimes 1_V \otimes 1_V \\
V^{\otimes 5} & \\
\uparrow & u \otimes \left((R \otimes 1_V) \circ \left(1_V \otimes u'\right)\right) \\
V &
\end{array}
$$

Example: the 4_1 knot



$\mapsto$

$$
\begin{array}{ll}
V & \\
\uparrow & \left(\left(n' \otimes 1_V\right) \circ \left(1_V \otimes R^{-1}\right)\right) \otimes n \\
V^{\otimes 5} & \\
\uparrow & 1_V \otimes 1_V \otimes R \otimes 1_V \\
V^{\otimes 5} & \\
\uparrow & 1_V \otimes R^{-1} \otimes 1_V \otimes 1_V \\
V^{\otimes 5} & \\
\uparrow & u \otimes \left(\left(R \otimes 1_V\right) \circ \left(1_V \otimes u'\right)\right) \\
V &
\end{array}
$$

Fact

The endomorphism on V we obtain is a scalar multiple of 1_V.
Plugging indeterminants in R and h gives polynomial invariants.

Fix a basis $\mathcal{B} := \{e_i\}$ of V, $R^{\pm 1} \in \mathrm{Aut}(V \otimes V)$ (and h) become matrices whose entries can be denoted by $\left(R^{\pm 1}\right)_{e_i \otimes e_j}^{e_k \otimes e_l}$ (and $h_{e_i}^{e_j}$).

Fix a basis $\mathcal{B} := \{e_i\}$ of V, $R^{\pm 1} \in \mathrm{Aut}(V \otimes V)$ (and h) become matrices whose entries can be denoted by $\left(R^{\pm 1}\right)_{e_i \otimes e_j}^{e_k \otimes e_l}$ (and $h_{e_i}^{e_j}$). By basic linear algebra, to compute the eigenvalue of the $\mathrm{End}(V)$-valued invariant is to evaluate a sum of the form

$$\sum_{\substack{a_1,\cdots,a_{2n-1} \in \mathcal{B} \\ a_0 = a_{2n} = e_1}} \underbrace{\left(R^{\pm 1}\right)_{a_0 \otimes a_1}^{a_2 \otimes a_3} \cdots \left(R^{\pm 1}\right)_{a_{2n-3} \otimes a_{2n-2}}^{a_{2n-1} \otimes a_{2n}}}_{\text{a product of length } n},$$

where n is the number of crossings of the knot.

Fix a basis $\mathcal{B} := \{e_i\}$ of V, $R^{\pm 1} \in \mathrm{Aut}(V \otimes V)$ (and h) become matrices whose entries can be denoted by $\left(R^{\pm 1}\right)_{e_i \otimes e_j}^{e_k \otimes e_l}$ (and $h_{e_i}^{e_j}$).
By basic linear algebra, to compute the eigenvalue of the $\mathrm{End}(V)$-valued invariant is to evaluate a sum of the form

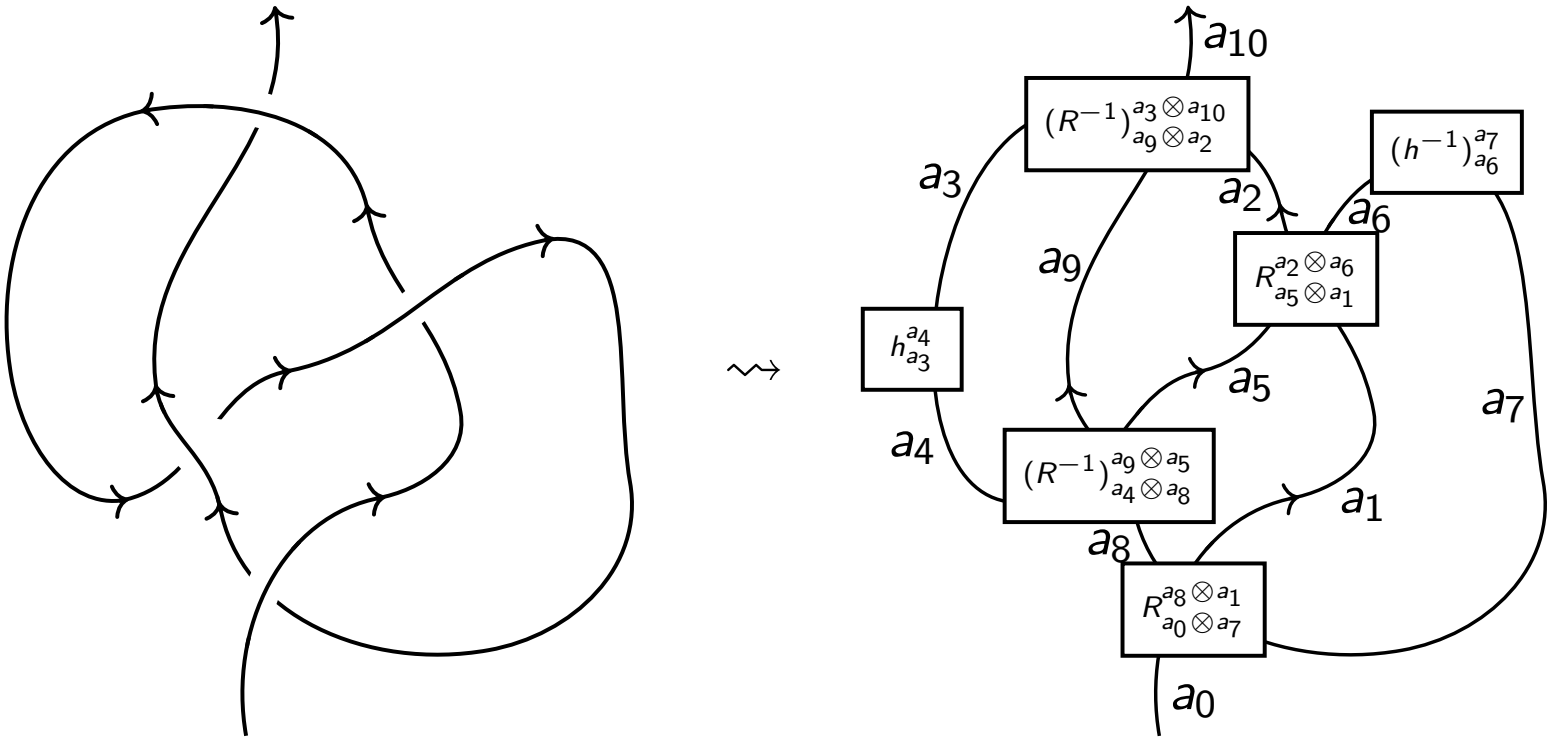
$$\sum_{\substack{a_1,\cdots,a_{2n-1} \in \mathcal{B} \\ a_0 = a_{2n} = e_1}} \underbrace{\left(R^{\pm 1}\right)_{a_0 \otimes a_1}^{a_2 \otimes a_3} \cdots \left(R^{\pm 1}\right)_{a_{2n-3} \otimes a_{2n-2}}^{a_{2n-1} \otimes a_{2n}}}_{\text{a product of length } n},$$

where n is the number of crossings of the knot.
Hence naively it requires

$$n \cdot (\dim V)^{2n-1}$$

multiplications and additions to compute the eigenvalue.

Example: the 4_1 knot again



Example: the 4_1 knot again

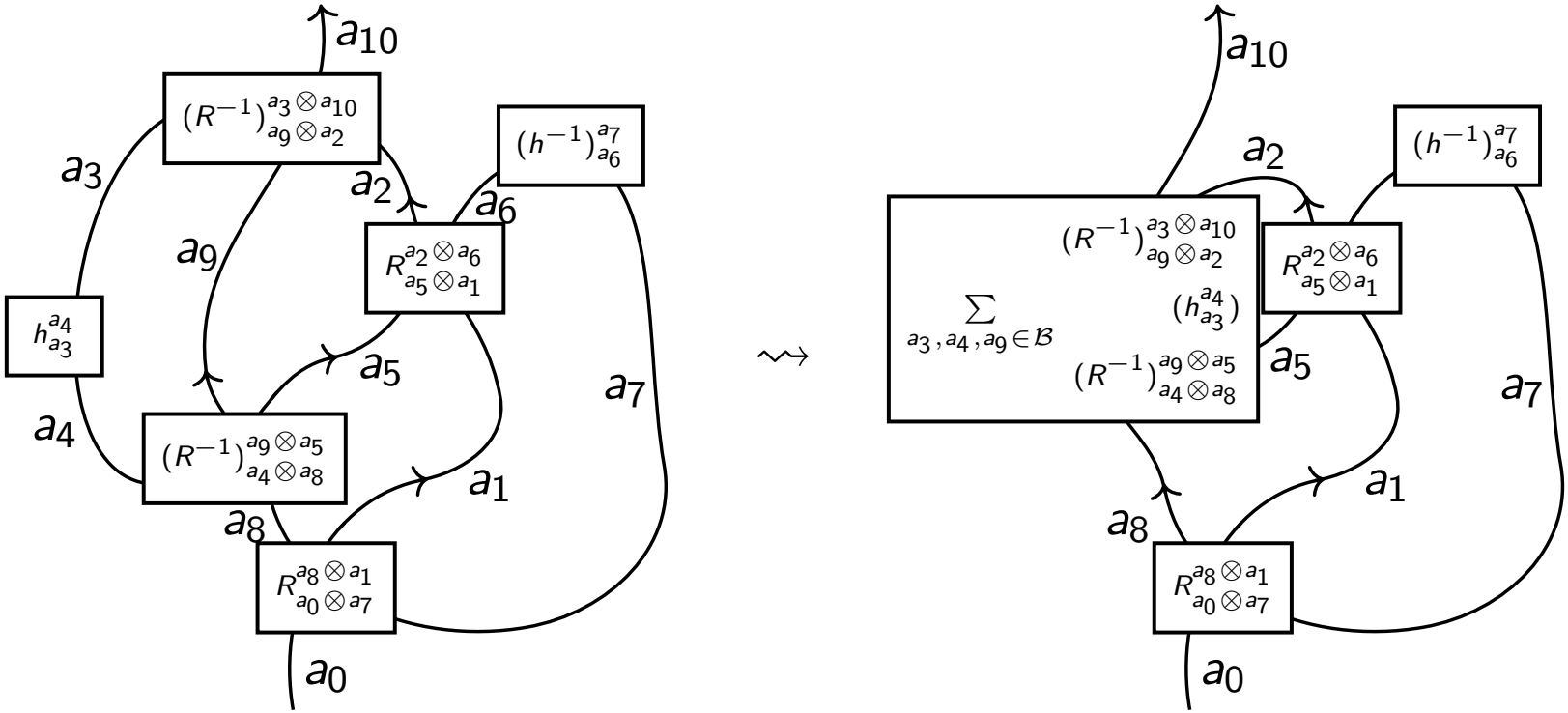


Table of Contents

1 Reshetikhin–Turaev functor

2 Tensor network

3 Implications and applications

Definition (Tensor)

An (equi-dimensional) n-tensor T over a ring $\mathcal{R}$ is a tuple $(T_{i_1,\dots,i_n})$ of elements of $\mathcal{R}$ with $i_j = 1, \dots, d$ for each j.

Definition (Tensor)

An (equi-dimensional) n-tensor T over a ring $\mathcal{R}$ is a tuple $(T_{i_1,\cdots,i_n})$ of elements of $\mathcal{R}$ with $i_j = 1, \ldots, d$ for each j.

Definition (Tensor contraction)

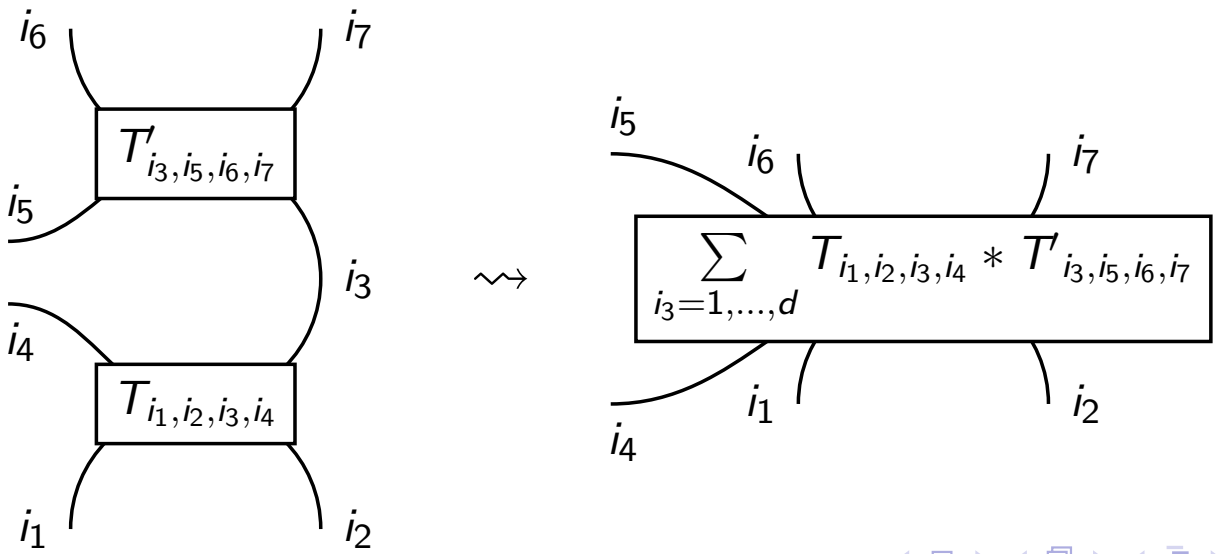
Given two tensors T and T', their contraction is a new tensor obtained by summing over one or more pairs of indices.

Definition (Tensor)

An (equi-dimensional) n-tensor T over a ring $\mathcal{R}$ is a tuple $(T_{i_1,\dots,i_n})$ of elements of $\mathcal{R}$ with $i_j = 1, \dots, d$ for each j.

Definition (Tensor contraction)

Given two tensors T and T', their contraction is a new tensor obtained by summing over one or more pairs of indices.



Computing a contraction of c tensors summing along s indices is

$$O(c \cdot d^s) \cdot d^{N-2s} = O(c \cdot d^{N-s}),$$

where N is the total number of indices of the c tensors.

Computing a contraction of c tensors summing along s indices is

$$O(c \cdot d^s) \cdot d^{N-2s} = O(c \cdot d^{N-s}),$$

where N is the total number of indices of the c tensors.

Definition (Contraction sequence)

A contraction sequence $\mathcal{S}$ on a tensor network Γ is a sequence of tensor contractions resulting in only one remaining tensor.

Computing a contraction of c tensors summing along s indices is

$$O(c \cdot d^s) \cdot d^{N-2s} = O(c \cdot d^{N-s}),$$

where N is the total number of indices of the c tensors.

Definition (Contraction sequence)

A contraction sequence $\mathcal{S}$ on a tensor network Γ is a sequence of tensor contractions resulting in only one remaining tensor.

Definition (Contraction-width)

Let $\mathfrak{S}(\Gamma)$ denote the set of all contraction sequences on Γ:

$$\underbrace{\min_{\mathcal{S}\in\mathfrak{S}(\Gamma)} \left(\overbrace{\max_{i=1,\ldots,|\mathcal{S}|} \underbrace{(N_i - s_i)}_{\text{local contraction-width of the } i\text{-th contraction}}}^{\text{contraction-width of } \mathcal{S}} \right)}_{\text{contraction-width of } \Gamma}$$

Theorem (L. 2026)

Computing a k-variable Reshetikhin–Turaev knot polynomial on an n-crossing diagram by contracting with contraction-width w is

$$O((\dim V)^w n^{k+2} (\log n)^3).$$

Theorem (L. 2026)

Computing a k-variable Reshetikhin–Turaev knot polynomial on an n-crossing diagram by contracting with contraction-width w is

$$O((\dim V)^w n^{k+2} (\log n)^3).$$

Lemma (Consequence of [Markov & Shi, 2008])

Let $w(\Gamma)$ and $tw(\Gamma)$ be the contraction-width and tree-width of Γ,

$$w(\Gamma) \leq \Delta(\Gamma) \cdot (tw(\Gamma) + 1) = O(\Delta(\Gamma) \cdot \sqrt{n}),$$

where $\Delta(\Gamma)$ is the maximum degree of Γ.

Theorem (L. 2026)

Computing a k-variable Reshetikhin–Turaev knot polynomial on an n-crossing diagram by contracting with contraction-width w is

$$O((\dim V)^w n^{k+2} (\log n)^3).$$

Lemma (Consequence of [Markov & Shi, 2008])

Let $w(\Gamma)$ and $tw(\Gamma)$ be the contraction-width and tree-width of Γ,

$$w(\Gamma) \leq \Delta(\Gamma) \cdot (tw(\Gamma) + 1) = O(\Delta(\Gamma) \cdot \sqrt{n}),$$

where $\Delta(\Gamma)$ is the maximum degree of Γ.

Corollary

Computing Reshetikhin–Turaev knot polynomials are $e^{O(\sqrt{n})}$.

Table of Contents

1 Reshetikhin–Turaev functor

2 Tensor network

3 Implications and applications

- There is no known fixed-parameter tractable algorithm for unknot recognition.

- There is no known fixed-parameter tractable algorithm for unknot recognition.
- There is no known algorithm that runs in sub-exponential time which computes the genus of a knot.

- There is no known fixed-parameter tractable algorithm for unknot recognition.
- There is no known algorithm that runs in sub-exponential time which computes the genus of a knot.

- There is no known fixed-parameter tractable algorithm for unknot recognition.
- There is no known algorithm that runs in sub-exponential time which computes the genus of a knot.

Theorem (Garoufalidis & L., 2026)

The t-degree of the 2-colored Links–Gould polynomial detects the genus for all 352,152,252 knots with ≤ 19 crossings.

- There is no known fixed-parameter tractable algorithm for unknot recognition.
- There is no known algorithm that runs in sub-exponential time which computes the genus of a knot.

Theorem (Garoufalidis & L., 2026)

The t-degree of the 2-colored Links–Gould polynomial detects the genus for all 352,152,252 knots with ≤ 19 crossings.

Conjecture

The 2-colored Links–Gould polynomial detects the genus.

- There is no known fixed-parameter tractable algorithm for unknot recognition.
- There is no known algorithm that runs in sub-exponential time which computes the genus of a knot.

Theorem (Garoufalidis & L., 2026)

The t-degree of the 2-colored Links–Gould polynomial detects the genus for all 352,152,252 knots with ≤ 19 crossings.

Conjecture

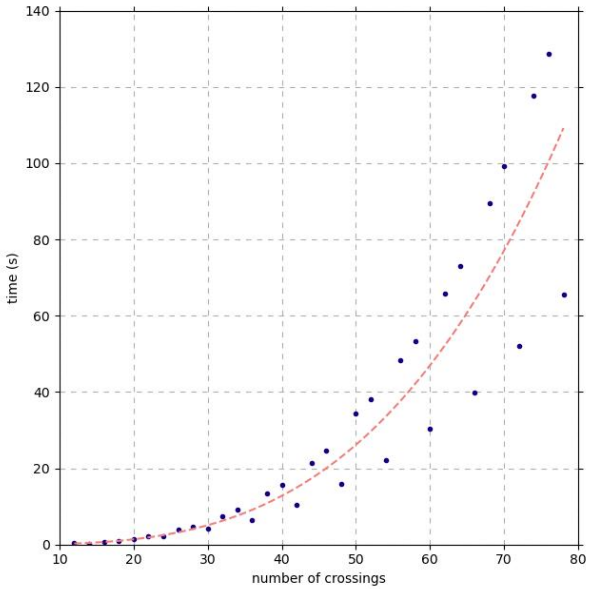
The 2-colored Links–Gould polynomial detects the genus.

Corollary

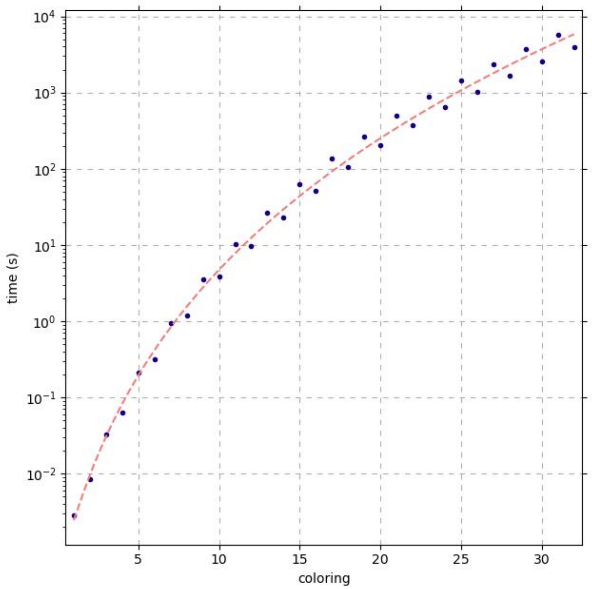
If the above conjecture is true, the computation of the 2-colored Links–Gould polynomial gives algorithms for the top two items.

The algorithm has been implemented in the in-development version of SnapPy, which computes any RT knot polynomial given its R-matrix and ribbon element.

The algorithm has been implemented in the in-development version of SnapPy, which computes any RT knot polynomial given its R-matrix and ribbon element.



The 2-colored Links–Gould polynomial for $(3, i)$-torus links as i ranges from 6 to 39.



The n-colored Jones polynomial for 11n34 as n ranges from 1 to 32. (plotted in log scale)

Comparison with other implementations on computing the Colored Jones polynomial:

Comparison with other implementations on computing the Colored Jones polynomial:

Coloring	`KnotTheory`	`ColorJones`	Our implementation
1	0.10473	0.512003	
2	5.75993	0.122924	
3	189.518	0.616438	
4	OOT	4.31470	
5		22.3089	
6		97.3698	
7		406.974	
8		1805.60	
9		OOT	

Table: Run time (in seconds) on the knot $11n34$ with a time constraint of $6 \cdot 10^3$ seconds, where OOT denotes that the time exceeded the time constraint.

Coloring	`KnotTheory`	`ColorJones`	Our implementation (`Python`)
1	0.10473	0.512003	0.002832
2	5.75993	0.122924	0.008507
3	189.518	0.616438	0.033057
4	OOT	4.31470	0.062606
5		22.3089	0.208873
6		97.3698	0.319684
7		406.974	0.942618
8		1805.60	1.19045
9		OOT	3.54350
10			3.90816
11			10.2546
12			9.56505
13			26.3211
14			23.0595
15			62.2023
16			50.9203
17			137.517
18			104.946
19			267.455
20			202.984
21			500.720
22			368.573
23			894.917
24			634.879
25			1440.03
26			1023.49
27			2317.58
28			1649.33
29			3704.88
30			2586.03

Coloring	`KnotTheory`	`ColorJones`	Our implementation (`Mathematica`)	Our implementation (`Python`)
1	0.10473	0.512003	0.977068	0.002832
2	5.75993	0.122924	0.968588	0.008507
3	189.518	0.616438	1.18240	0.033057
4	OOT	4.31470	1.84138	0.062606
5		22.3089	4.18859	0.208873
6		97.3698	9.60003	0.319684
7		406.974	23.4769	0.942618
8		1805.60	50.6798	1.19045
9		OOT	109.212	3.54350
10			215.333	3.90816
11			419.588	10.2546
12			766.792	9.56505
13			1347.75	26.3211
14			2275.95	23.0595
15			3856.06	62.2023
16			OOT	50.9203
17				137.517
18				104.946
19				267.455
20				202.984
21				500.720
22				368.573
23				894.917
24				634.879
25				1440.03
26				1023.49
27				2317.58
28				1649.33
29				3704.88

(Joint work with S. Garoufalidis) Something we can now compute:

(Joint work with S. Garoufalidis) Something we can now compute:

- n-colored Links–Gould polynomial (harder than $(4n-1)$-colored Jones polynomial).

(Joint work with S. Garoufalidis) Something we can now compute:

- n-colored Links–Gould polynomial (harder than $(4n-1)$-colored Jones polynomial).
- The $\mathrm{FK3}$ polynomial (about as hard as 11-colored Jones polynomial).

(Joint work with S. Garoufalidis) Something we can now compute:

- n-colored Links–Gould polynomial (harder than $(4n-1)$-colored Jones polynomial).
- The $\mathrm{FK3}$ polynomial (about as hard as 11-colored Jones polynomial).
- The KM_3 polynomial (harder than 15-colored Jones polynomial).