

Topological Quantum Compilation of Two-Qubit Gates

From exhaustive search over braids to global optimisation



[P1] P. C. Burke, C. Aravanis, J. Aspman, J. Mareček, J. Vala,
Topological quantum compilation of two-qubit gates
Phys. Rev. A **110**, 052616 — [doi:10.1103/PhysRevA.110.052616](https://doi.org/10.1103/PhysRevA.110.052616)



[P2] P. Rytíř, P. C. Burke, C. Aravanis, J. Vala, J. Mareček,
Topological quantum compilation using mixed-integer programming
IEEE Trans. Quantum Eng. **7**, 3104212 (2026) — [doi:10.1109/TQE.2026.3710775](https://doi.org/10.1109/TQE.2026.3710775)

The team

Christos Aravanis	Johannes Aspman	Phillip C. Burke	Jakub Mareček	Pavel Rytíř	Jiří Vala
University of Sheffield CTU in Prague	Czech Technical University in Prague	University College Dublin Maynooth University	Czech Technical University in Prague	Czech Technical University in Prague	Maynooth University DIAS · Tyndall
[P1] [P2]	[P1]	[P1] [P2]	[P1] [P2]	[P2]	[P1] [P2]

P. C. B. acknowledges Taighde Éireann — Research Ireland, grant 21/RP-2TF/10019.

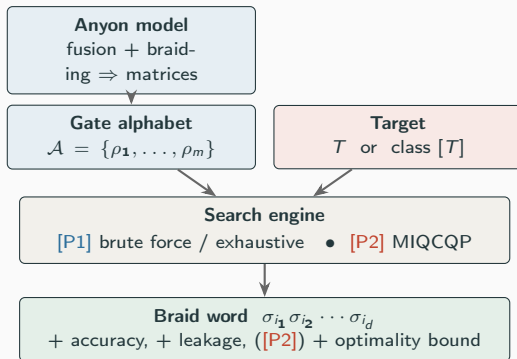
P. R., C. A., J. A. and J. M. acknowledge the Czech Science Foundation, grant 23-0947S.

J. V. acknowledges Enterprise Ireland's DTIF programme, project QColr DT 2019 0090B.

P. C. B. and J. V. thank Paul Watts for fruitful discussions.

The agenda

- **Anyons: Abelian and non-Abelian**
universality, and which models furnish a gate alphabet at all
- **The shared toolbox**
local equivalence classes, Makhlin invariants, the Weyl chamber
- **[P1] Fibonacci anyons by exhaustive search**
- **A primer on integer programming**
what an MILP is, branch & bound, why global bounds matter
- **[P2] Compilation as a mixed-integer program**
- **Synthesis**



Anyons: Abelian and non-Abelian

Non-Abelian anyons: braiding is a matrix



two τ 's can fuse *two* ways

$$\tau \times \tau = \mathbb{1} + \tau$$

$\Rightarrow$ a **degenerate space**
with no local label

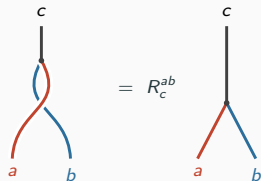
With n anyons at fixed positions the ground space is **degenerate**, of dimension $\dim \mathcal{V}_n \sim d^n$ with d the *quantum dimension*. A braid acts as a **unitary matrix** on that space:

$$\rho(\sigma_k) \in U(\dim \mathcal{V}_n), \quad \rho(\sigma_k)\rho(\sigma_\ell) \neq \rho(\sigma_\ell)\rho(\sigma_k)$$

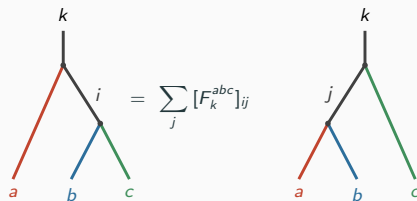
- information lives in the **fusion channel** — global, non-local
- no local perturbation can read or corrupt it: **topological protection**
- the matrices are fixed by two pieces of data: the **R -matrix** (exchange) and the **F -matrix** (re-association)

► Non-Abelian braiding *computes*. The question is which unitaries it reaches.

R — exchange of two anyons



F — change of fusion order



Any braid matrix in any basis is a word in R 's and F 's. For Fibonacci anyons ([P1]):

$$R = \text{diag}(R_{\mathbb{1}}^{\tau\tau}, R_{\tau}^{\tau\tau}) = \begin{pmatrix} e^{-4\pi i/5} & 0 \\ 0 & e^{3\pi i/5} \end{pmatrix}, \quad F = \begin{pmatrix} \phi^{-1} & \sqrt{\phi^{-1}} \\ \sqrt{\phi^{-1}} & -\phi^{-1} \end{pmatrix}, \quad \phi = \frac{1+\sqrt{5}}{2}.$$

► The model supplies a **finite alphabet of fixed unitaries**. Nothing is tunable: no angles, no pulses.

Anyon models and universality

Model	quantum dim. d	braiding rep.	braiding alone gives	universal?
Abelian (e.g. Laughlin $\nu=1/3$)	1	1×1	phases $e^{i\theta}$	no
Ising ($\nu=5/2$ candidate)	$\sqrt{2}$	2×2	Clifford group	no
Fibonacci	$\phi = \frac{1+\sqrt{5}}{2}$	2×2	dense in $SU(2)$	yes
non-semisimple Ising	—	2×2	dense	yes

Why Ising fails

Braiding gives the Clifford group, which is classically simulable (Gottesman–Knill).
Supplementing it costs the protection.

The non-semisimple rescue (*used in [P2]*)

Iulianelli et al.: extend the Ising TQFT by one extra charge α , with $\alpha \times \sigma = (\alpha+1) + (\alpha-1)$. Braiding then becomes **universal**. The proof is **existential**: the gates exist, but are not exhibited.

► Universality asserts *closure*; compilation *constructs*. [P2] supplies the construction.

The compilation problem — and why it is hard

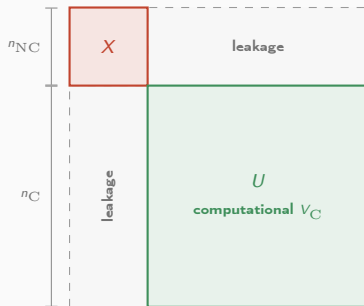
Given a finite alphabet $\mathcal{A} = \{\rho_1, \dots, \rho_m\} \subset \mathbb{C}^{n \times n}$
and a target $T \in \mathbb{C}^{n_C \times n_C}$,

find a word $I : \{1, \dots, d\} \rightarrow \{1, \dots, m\}$ with

$$P = \prod_{i=1}^d \rho_{I(i)} \approx T.$$

Three difficulties:

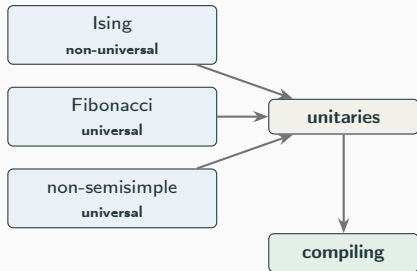
1. **Combinatorial explosion.** m^d words. For $m = 5$, $d = 40$: 10^{28} .
2. **Leakage.** $n > n_C$: the braid acts on a *bigger* space than the two qubits. Need $P = X \oplus U$.
3. **A needle in a 15-dim. haystack.** A specific $T \in SU(4)$ is a single point in a 15-dimensional manifold.



$P = X \oplus U$ is leakage-free
 $\Leftrightarrow$ the dashed blocks vanish

Cui et al. 2019: all leakage-free
Fibonacci braiding gates are *non-entangling*.
Entanglement must be paid for in leakage.

The method does not depend on the anyon model



*Abelian anyons do not enter here:
 $d = 1$, so the fusion space is one-dimensional and carries no qubit*

Neither method asks *which anyons* produced the alphabet. It requires only

- a list of **fixed unitary matrices** ρ_i ,
- the splitting $\mathbb{C}^n = V_C \oplus V_{NC}$,
- a target gate T or a target class $[T]$.

Model-independence, demonstrated

The same machinery is run on **three alphabets**:

Fibonacci, 6 anyons ($n=5$) [P1], [P2]

Fibonacci + inverses ($m=10$) [P1]

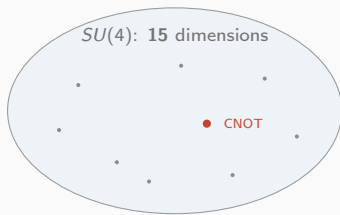
non-semisimple Ising, 5 anyons ($n=6$) [P2]

Nothing in the formulation is topological. Any finite gate set with a computational subspace would do. **What the anyon model decides is whether there is a qubit to compile into at all** — and for that one needs $d > 1$.

The shared toolbox: local equivalence classes

Gate versus local equivalence class

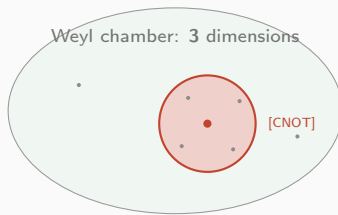
Compile a *gate*



braid words • ; target •

a single point

Compile a *local equivalence class*



a whole $SU(2) \otimes SU(2)$ orbit

a fat target

$$U = k_1 \mathcal{A} k_2, \quad k_1, k_2 \in SU(2) \otimes SU(2) \quad \Rightarrow \quad [T] = \{ k_1 T k_2 : k_1, k_2 \in SU(2) \otimes SU(2) \}$$

► Single-qubit gates are cheap. Ask only for the **entangling content**: $15D \rightarrow 3D$.

Makhlin local invariants

1. Go to the Bell (magic) basis, where $SU(2) \otimes SU(2)$ becomes $SO(4)$:

$$Q = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & i \\ 0 & i & 1 & 0 \\ 0 & i & -1 & 0 \\ 1 & 0 & 0 & -i \end{pmatrix}, \quad U_B = Q^\dagger U Q.$$

2. Form the **Makhlin matrix**, killing one local factor ($O^T O = \mathbb{1}$):

$$m_U = U_B^T U_B.$$

3. Read off the invariants of its characteristic polynomial:

$$g_1 = \operatorname{Re} \left\{ \frac{\operatorname{tr}^2 m_U}{16 \det U} \right\}, \quad g_2 = \operatorname{Im} \left\{ \frac{\operatorname{tr}^2 m_U}{16 \det U} \right\}, \quad g_3 = \frac{\operatorname{tr}^2 m_U - \operatorname{tr} m_U^2}{4 \det U}.$$

The $\det U$ normalisation extends these to all of $U(4)$: it removes the global phase.

Class	g_1	g_2	g_3
[1]	1	0	3
[DCNOT]	0	0	-1
[SWAP]	-1	0	-3
[B]	0	0	0
[CNOT]	0	0	1
[$\sqrt{\text{SWAP}}$]	0	1/4	0

[P1] Table III

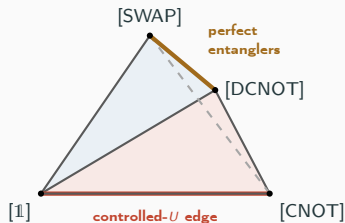
The cost functions

$$d^E(U) = \sum_{i=1}^3 |g_i(E) - g_i(U)|^2$$

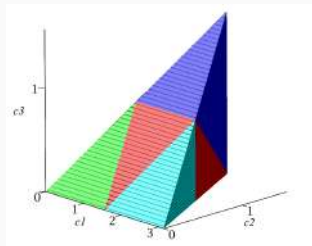
$$D_{\text{CNOT}}(U) = g_1^2 + g_2^2 + (g_3 - 1)^2$$

$$D_{\text{PE}}(U) = (g_1 - g_3 \sqrt{g_1^2 + g_2^2})^2$$

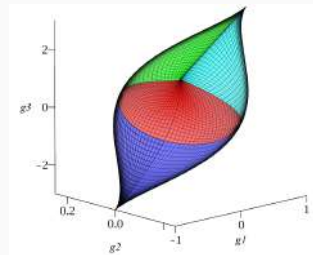
The Weyl chamber



schematic, (c_1, c_2, c_3) picture



(a) Cartan coefficients c_1, c_2, c_3



(b) local invariants g_1, g_2, g_3

[P1] Fig. 2, after Watts et al. (2015); the perfect-entangler region in (b) is the “Eye of Sauron” of Watts & Vala (2013).

- Every two-qubit gate is **one point**; single-qubit gates move nothing.
- Vertices are the familiar classes; the **red region** is the **perfect entanglers** — gates that map some product state to a maximally entangled one.
- [CNOT] sits at the *centre* of that region; [1] and [SWAP] at opposite corners.

**[P1] Fibonacci anyons by
exhaustive search**

Paper 1

Topological quantum compilation of two-qubit gates

P. C. Burke, C. Aravanis, J. Aspman, J. Mareček, J. Vala

- Six **Fibonacci** anyons $\Rightarrow$ 5-dimensional Hilbert space, 4 computational.
- Compile CNOT *exactly* $\rightarrow$ **fails**.
- Compile the *class* [CNOT] $\rightarrow$ **succeeds**, $d \sim 10^{-9}$ at length 30.
- Which classes arise *naturally*? $\rightarrow$ the edges [1]–[CNOT] and [DCNOT]–[SWAP].
- A numerically **exact**, leakage-free [SWAP] in **nine** braid letters.

Fibonacci anyons, and two qubits in six of them

Two charges, $\mathbb{1}$ and τ ; one non-trivial fusion rule

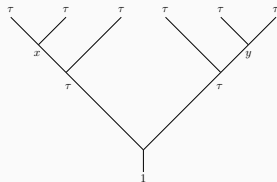
$$\tau \times \tau = \mathbb{1} + \tau.$$

Bonesteel notation: $|(\bullet, \bullet)_{\mathbb{1}}\rangle$ and $|(\bullet, \bullet)_{\tau}\rangle$.

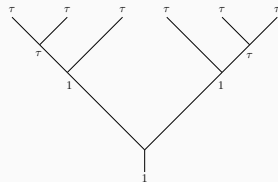
Six anyons $\Rightarrow \dim \mathcal{H} = 5$:

$$\underbrace{\{|11\rangle, |1\tau\rangle, |\tau 1\rangle, |\tau\tau\rangle\}}_{V_C \text{ (two qubits)}} \oplus \underbrace{\{|NC\rangle\}}_{V_{NC}}$$

One leftover dimension (leakage).



(a) $|xy\rangle$, computational

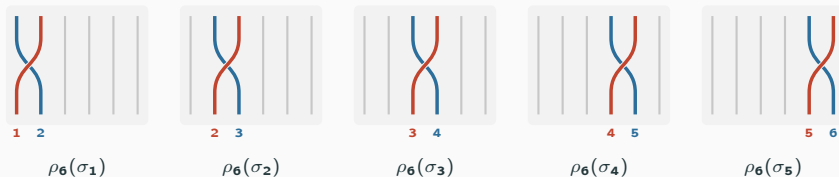


(b) $|NC\rangle$, non-computational

[P1] Fig. 1: basis states of two qubits in the six-anyon system. Encoding due to

Cui et al. (2019).

The gate alphabet: five braids, and their matrices



Six worldlines per panel, read upwards; the crossing pair is picked out, spectators grey.

In the basis $\{|NC\rangle, |11\rangle, |1\tau\rangle, |\tau 1\rangle, |\tau\tau\rangle\}$:

$$\rho_6(\sigma_1) = R_{\tau}^{\tau\tau} \oplus (R \otimes \mathbb{1}_2)$$

$$\rho_6(\sigma_2) = R_{\tau}^{\tau\tau} \oplus (FRF \otimes \mathbb{1}_2)$$

$$\rho_6(\sigma_3) = P_{14}(R_{\tau}^{\tau\tau} \oplus R \oplus FRF)P_{14}$$

$$\rho_6(\sigma_4) = R_{\tau}^{\tau\tau} \oplus (\mathbb{1}_2 \otimes FRF)$$

$$\rho_6(\sigma_5) = R_{\tau}^{\tau\tau} \oplus (\mathbb{1}_2 \otimes R)$$

Two alphabets

basic $\{\sigma_1, \dots, \sigma_5\} \rightarrow \{0, 1, 2, 3, 4\}$, $m = 5$

extended adds the inverses,

$\{\sigma_1, \dots, \sigma_5, \sigma_1^{-1}, \dots, \sigma_5^{-1}\} \rightarrow \{0, \dots, 9\}$,
 $m = 10$

Every letter is itself **leakage-free**.

Products need not be.

What we measure, and how we search

Leakage / unitarity of the 4×4 block A of $M = M_{11} \oplus A$:

$$|M_{11}| \approx 1, \quad d^U(A) = \|A^\dagger A - \mathbb{1}\|_1 = \text{tr}|A^\dagger A - \mathbb{1}|.$$

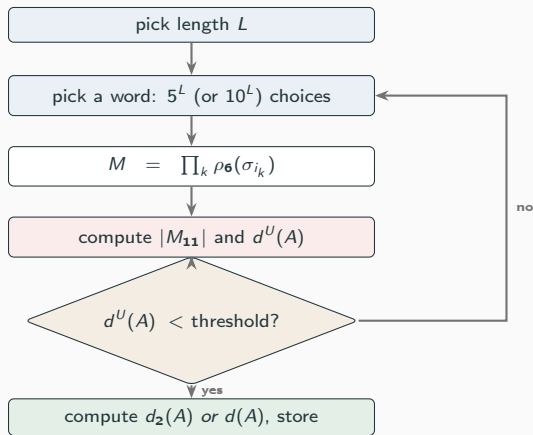
Trace norm, not $\text{tr}(A^\dagger A)/\dim A$ — the trace hides too much.

Distance to a gate (normalised Hilbert–Schmidt, $p = 2$):

$$d_2(A, B) = \left\| \frac{A}{\|A\|_2} - \frac{B}{\|B\|_2} \right\|_2 \in [0, 2].$$

Distance to a class:

$$d^E(A) = \sum_i |g_i(E) - g_i(A)|^2.$$



► Exhaustive to $L = 13$ – 14 ; beyond that random sampling, $\sim 10^{12}$ trials, up to $L = 40$.

Attempt 1: the CNOT gate itself, basic alphabet

No.	L	Operator	$d_2(A)$	$ M_{11} $	$d^U(A)$
1	3	111	1.1756	1	0
2	4	1001	1	1	0
3	5	00000	1	1	0
4	6	443344	1.0759	1	0
5	7	4334300	0.8972	1	0
6	8	33333300	0.8957	1	0
7	9	100100000	1	1	0
8	10	000000000	1	1	0
9	11	3100100000	0.8972	1	0
10	12	32202222222	0.8092	1	0
11	12	011311331113	0.8407	1	0
12	13	4420410200044	0.8271	1	$5.2464 \cdot 10^{-15}$
13	16	3244411044114244	0.8092	1	0
14	17	03043034440033034	0.7985	1	0
15	21	223400344020222234322	0.6321	0.9842	0.0588
16	21	423333044023300323423	0.6525	0.9956	0.0149
17	23	34344031313404411322240	0.7851	0.9538	0.0902
18	27	340113211204400220142121340	0.4841	0.9842	0.0914
19	27	423344224432003320410410133	0.5422	0.9842	0.0626
20	31	2430341014411443041020323332434	0.4823	0.9807	0.0731
21	31	4040232033011013103334443303002	0.6994	0.9899	0.0276

[P1] Table I. Above the rule: exhaustive ($L \leq 13$); below: random sampling. 0104230 = $\sigma_1\sigma_2\sigma_1\sigma_5\sigma_3\sigma_4\sigma_1$.

► Best $d_2(A) \approx 0.48$ out of a range $[0, 2]$, and only by *spending* unitarity ($d^U \sim 0.07$). No convergence.

Attempt 1': the CNOT gate, extended alphabet (with inverses)

No.	L	Operator	$d_2(A)$	$ M_{11} $	$d^U(A)$
1	2	30	0.8092	1	0
2	3	880	1.1667	1	0
3	4	5300	0.8092	1	0
4	5	48440	0.9708	1	0
5	6	553000	0.8092	1	0
6	7	4334300	0.8972	1	0
7	8	37770000	0.7916	0.9539	0.0902
8	9	441001048	0.8191	1	0
9	10	2555337582	0.7916	0.9538	0.0902
10	11	59672955693	0.8191	1	0
11	12	374050007970	0.7916	0.9538	0.0902
12	12	306947382105	0.8092	1	0
13	13	6595888969003	0.7985	1	0
14	14	75139699757375	0.6428	0.9842	0.0588
15	20	78450807782427077787	0.6145	0.9759	0.0939
16	24	287430798424700981936977	0.5497	0.9842	0.0588

[P1] Table II. $\{\sigma_1 \dots \sigma_5, \sigma_1^{-1} \dots \sigma_5^{-1}\} \rightarrow \{0..9\}$. Exhaustive only to $L = 8 - 10^L$ bites hard.

► Doubling the alphabet does not help: best $d_2 \approx 0.55$. More letters buy reach but cost exponentially more search.

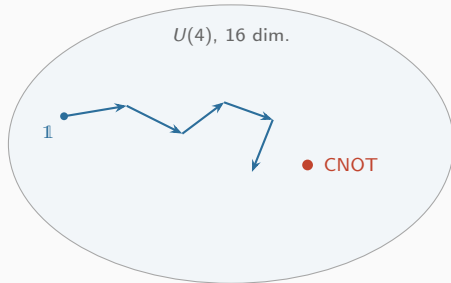
Why exact CNOT is hard at these lengths

A specific $T \in U(4)$ is **one point** in a 16-dimensional manifold (15, after the global phase).

The alphabet offers **five discrete letters**, most of which lie in the single-qubit subgroup $U(2) \otimes U(2)$.

Only **one** direction of the alphabet is genuinely two-qubit.

A needle in fifteen dimensions, sought with a ruler that moves in few directions.



a braid word is a walk
in discrete steps

► So do not ask for the point; ask for the **class**.

Attempt 2: the class [CNOT], basic alphabet

No.	L	Operator	$d(A)$	$ M_{11} $	$d^U(A)$
1	3	000	5	1	0
2	4	0000	5	1	0
3	5	00000	5	1	0
4	6	222000	2.9082	0.9538	0.0902
5	7	0000000	5	1	0
6	8	00000000	5	1	0
7	9	000000000	5	1	0
8	10	2221001222	0.4634	0.9759	0.0476
9	11	22210012220	0.4634	0.9759	0.0476
10	12	223443100122	$2 \cdot 10^{-5}$	0.9759	0.0476
11	13	2213404301242	$2 \cdot 10^{-5}$	0.9759	0.0476
12	17	34224334310031224	$2 \cdot 10^{-5}$	0.9759	0.0476
13	21	242314040312211412221	0.0943	0.9916	0.0293
14	24	422211222020214000112021	0.0078	0.9948	0.0178
15	27	101422102223130431111322222	0.0012	0.9942	0.022
16	29	31423322434332420442310301422	0.0001	0.9952	0.0184
17	30	222223043422422024043333320003	$7.779 \cdot 10^{-9}$	0.998	0.0076
18	31	2344130440331032213334400344312	$2.2 \cdot 10^{-8}$	0.9973	0.0095
19	33	423330314001220244224333334034032	$7.8 \cdot 10^{-9}$	0.998	0.0068
20	39	132424244140142042111112011202122020400	0.0002	0.9995	0.0021
21	39	231333001244422220433403402422343302043	$7.8 \cdot 10^{-9}$	0.998	0.0047

[P1] Table IV. $d(A) = d^{\text{CNOT}}(A)$ in the local invariants.

► $d \sim 10^{-5}$ already at $L = 12$; $d \sim 8 \cdot 10^{-9}$ at $L = 30$ — with d^U simultaneously *improving* to $5 \cdot 10^{-3}$. The gate search never got near its own target; this one essentially reaches its own.

Attempt 2': the class [CNOT], extended alphabet

No.	L	Operator	$d(A)$	$ M_{11} $	$d^U(A)$
1	1	0	5	1	0
2	2	00	5	1	0
3	3	222	2.9082	0.9538	0.0902
4	4	2220	2.9082	0.9538	0.0902
5	5	22200	2.9082	0.9538	0.0902
6	6	222000	2.9082	0.9538	0.0902
7	7	2220000	2.9082	0.9538	0.0902
8	10	6577100177	0.4634	0.9759	0.0476
9	12	208923329892	$2 \cdot 10^{-5}$	0.9759	0.0476
10	13	3734788743705	$2 \cdot 10^{-5}$	0.9759	0.0476
11	14	26952811352642	$2 \cdot 10^{-5}$	0.9759	0.0476
12	17	76177322244377075	0.0678	0.9838	0.0497
13	19	9777577842888487783	0.0158	0.9943	0.0178
14	20	13282844544028208235	0.0012	0.984	0.0458
15	21	726202360623786904622	0.0008	0.9875	0.048
16	21	524546245839171526261	$2 \cdot 10^{-5}$	0.9759	0.0476

[P1] Table V. Exhaustive to $L = 7$ only.

The trade-off, in the columns

Low $d(A)$ and low $d^U(A)$ **pull against each other** — the Cui et al. obstruction. Longer words buy both at once.

Basic vs. extended

No clear advantage from inverses: the alphabet holds effectively *one* two-qubit operation, and is too coarse either way.

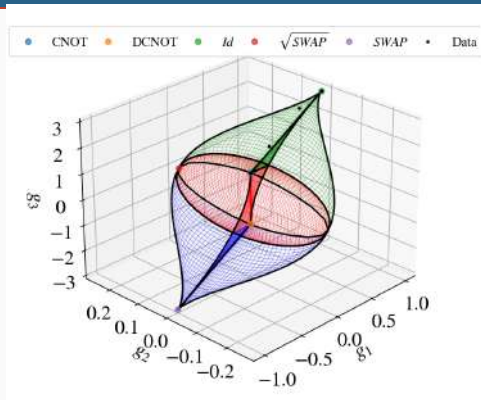
Which classes does braiding reach *naturally*?

No.	L	Operator	g_1	g_2	$\text{Re}(g_3)$
1	3	000	1	0	3
2	4	0000	1	0	3
3	5	00000	1	0	3
4	6	222000	0.7624	0.0201	2.5248
5	7	0000000	1	0	3
6	8	00000000	1	0	3
7	9	000000000	1	0	3
8	10	2221001222	0.3042	0.0112	1.6085
9	11	22210012220	0.3042	0.0112	1.6085
10	12	223443100122	0.0018	0.0011	1.0036
11	13	2213404301242	0.0018	0.0011	1.0036
12	17	34224334310031224	0.0018	0.0011	1.0036
13	21	242314040312211412221	0.1374	-0.0006	1.2747
14	24	422211222020214000112021	0.0396	-0.0008	1.0791
15	27	101422102223130431111322222	0.0157	-0.0003	1.0314
16	29	31423322434332420442310301422	0.0046	0.0003	1.0092
17	30	22223043422422024043333320003	$3.88 \cdot 10^{-5}$	$-7.3 \cdot 10^{-6}$	1.0001
18	31	2344130440331032213334400344312	$6.39 \cdot 10^{-5}$	$1.76 \cdot 10^{-5}$	1.0001
19	33	423330314001220244224333334034032	$3.88 \cdot 10^{-5}$	$-7.3 \cdot 10^{-6}$	1.0001
20	39	132424244140142042111112011202122020400	0.0064	$3.64 \cdot 10^{-5}$	1.0128
21	39	231333001244422220433403402422343302043	$3.88 \cdot 10^{-5}$	$-7.3 \cdot 10^{-6}$	1.0001

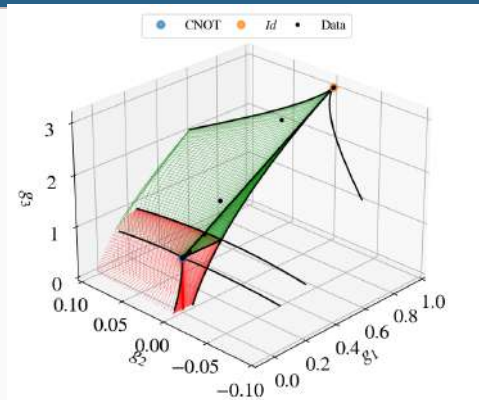
[P1] Table VI: the invariants of the words in Table IV. $\text{Im}(g_3) < 10^{-5}$ throughout, so it is omitted.

► Read the last column: $g_3 : 3 \rightarrow 1$ while $g_1 : 1 \rightarrow 0$ — the words walk from $[1]$ to $[\text{CNOT}]$.

The walk to [CNOT]



(a) with the full Weyl chamber

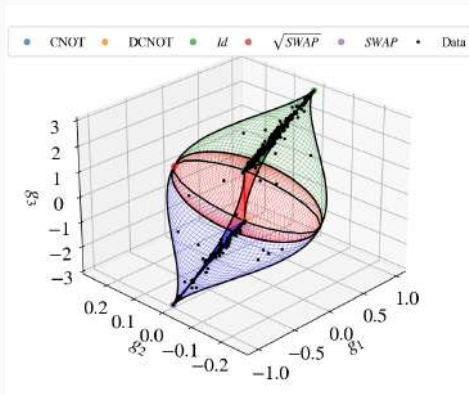


(b) zoom on the top half

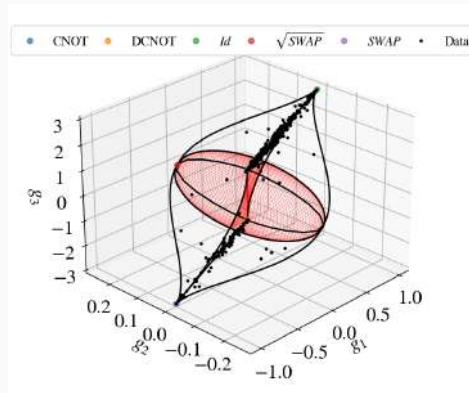
[P1] Fig. 3: invariants of the matrices of Table VI in (g_1, g_2, g_3) space, braids of length 3–39. Perfect-entangler region in red.

► The points pile up on the $[1]$ –[CNOT] edge and converge onto the [CNOT] vertex.

What braiding gives you when you do not ask (1/2)



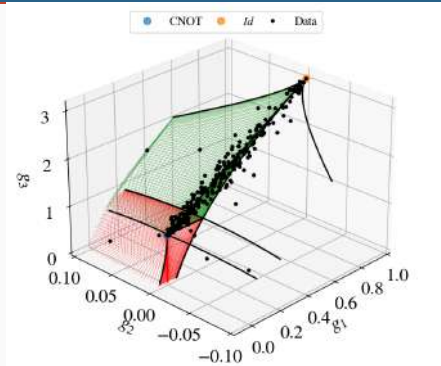
(a) with the full Weyl chamber



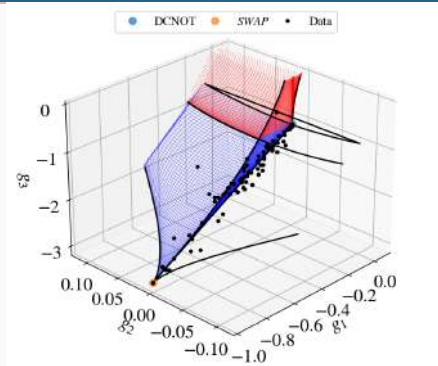
(b) with the perfect-entangler region only — the “Eye of Sauron”

[P1] Fig. 4(a),(b): ~ 1000 random words of length 40 that pass the unitarity test.

What braiding gives you when you do not ask (2/2)



(c) zoom, top half of the chamber



(d) zoom on the CNOT point

[P1] Fig. 4(c),(d).

► Natural operations concentrate on **two edges**: $[1]$ –[CNOT] (controlled-unitaries) and [DCNOT]–[SWAP]. The interior is essentially empty.

Where length-40 words actually land

No.	Operator ($L = 40$)	$d(A)$	Class	$ M_{11} $	$d^U(A)$
1	0131130443333410041340121001420341430104	0	ID	1	$2.43 \cdot 10^{-14}$
2	1332041020222120241220324100023023433334	0	SWAP	1	$2 \cdot 10^{-14}$
3	1433031222342422203023223230240443302034	$4.294 \cdot 10^{-11}$	CNOT	0.9887	0.045
4	2042023223234230444023402233310403144302	$2.197 \cdot 10^{-8}$	CNOT	0.9973	0.009
5	2111210000021122401344030030343310112200	$2.778 \cdot 10^{-8}$	CNOT	0.9904	0.037
6	2000011042011221000220214134423024141113	$4.978 \cdot 10^{-6}$	CNOT	0.9847	0.039
7	4424112204224004142400202443114301430203	$1.22 \cdot 10^{-5}$	CNOT	0.9917	0.048
8	4123222404221021243042034403104041340422	$2.169 \cdot 10^{-5}$	DCNOT	0.9759	0.048
9	4111441434443422140131014432210304341300	$2.169 \cdot 10^{-5}$	CNOT	0.9759	0.048
10	3214304124313214030210210410322210013242	$9.858 \cdot 10^{-5}$	DCNOT	0.9921	0.032
11	3010240424120241112044202220001020142122	0	CNOT	0.991	0.035
12	2303323004100444134324422204022120202124	0.001	CNOT	0.9942	0.035
13	4403124432032044210003142322243041131112	0.001	CNOT	0.9942	0.043
14	1442024304110331441224310343121400044432	0.001	DCNOT	0.9942	0.043

[P1] Table VII: closest equivalence class for a sample of the ~ 1000 random length-40 words.

► Rows 1–2: $d = 0$ with $d^U \sim 10^{-14}$ — **exactly** leakage-free [1] and, remarkably, **exactly** leakage-free [SWAP]. Every [CNOT] row carries $d^U \sim 10^{-2}$.

[SWAP] in nine letters

No.	L	Operator	$d(A)$	Class	$ M_{11} $	$d^U(A)$
1	1	0	0	ID	1	0
2	2	00	0	ID	1	$4 \cdot 10^{-16}$
3	3	000	0	ID	1	$7 \cdot 10^{-16}$
4	4	0000	0	ID	1	$4 \cdot 10^{-16}$
5	5	00000	0	ID	1	0
6	6	000000	0	ID	1	$2 \cdot 10^{-16}$
7	7	0000000	0	ID	1	$7 \cdot 10^{-16}$
8	8	00000000	0	ID	1	$4 \cdot 10^{-16}$
9	9	234123012	0	SWAP	1	$4.7 \cdot 10^{-15}$
10	10	2341230120	0	SWAP	1	$4.6 \cdot 10^{-15}$
11	10	2223443222	0.463	CNOT	0.9759	0.048
12	11	23412301200	0	SWAP	1	$4.1 \cdot 10^{-15}$
13	11	22234430222	0.463	CNOT	0.9759	0.048
14	12	423123041302	0	SWAP	1	$6.2 \cdot 10^{-15}$
15	12	223104403122	$2.169 \cdot 10^{-5}$	CNOT	0.9759	0.048
16	13	1133100000022	$2.169 \cdot 10^{-5}$	CNOT	0.9759	0.048
17	14	03023010000322	$2.169 \cdot 10^{-5}$	CNOT	0.9759	0.048

[P1] Table VIII, exhaustive to $L = 14$.

► [SWAP] is exactly leakage-free (it is non-entangling, consistent with Cui et al.); every *entangling* class is only approximately so.

Row 9: the word 234123012, i.e.

$$\sigma_3\sigma_4\sigma_5\sigma_2\sigma_4\sigma_3\sigma_1\sigma_2\sigma_3,$$

gives $d = 0$, $|M_{11}| = 1$, $d^U = 4.7 \cdot 10^{-15}$.

An example of the resulting $M_{11} \oplus A$ at $L = 40$:

$$\begin{pmatrix} -1.0 & 0 & 0 & 0 & 0 \\ 0 & -.04-.31i & -.36+.18i & .49+.22i & .07-.67i \\ 0 & .49-.22i & -.46-.50i & .04-.31i & -.39+.07i \\ 0 & -.39-.07i & -.04-.31i & .46-.50i & .49+.22i \\ 0 & -.07-.67i & -.49-.22i & -.36-.18i & .04-.31i \end{pmatrix}$$

[P1] Eq. (20) Nothing about A looks like SWAP; it is locally equivalent to it.

[P1] — what we learned

Methodological

- Compiling a **gate** fails; compiling a **class** works. On comparable budgets the gate search stalls at $d_2 \approx 0.48$ of a possible 2, while the class search reaches $d \sim 10^{-9}$. (Different metrics — d_2 to a gate, d to a class — so read the *qualitative* contrast, not a ratio.)
- 15-dimensional search $\rightarrow$ 3-dimensional search.
- Basic vs. extended alphabet: no real difference.

Physical

- Braiding flows along the [1]–[CNOT] and [DCNOT]–[SWAP] edges.
- Exactly leakage-free *and* entangling: not found — supporting Cui et al.'s conjectured negative.

What [P1] cannot tell you

The search is **brute force**: at $L > 14$ it samples $\sim 10^{12}$ words of $5^{40} \approx 10^{28}$, a fraction 10^{-16} . Every number in those tables is therefore an **upper bound**. Nothing says:

- that a shorter word does not exist,
- that a better word of that length does not exist,
- when to stop searching.

[P2] closes that gap, with machinery from optimisation.

An interlude: what is integer programming?

Interlude

What is a mixed-integer program?

Everything in this section is **standard** in mathematical optimisation and **nothing** in it is quantum. It is here because [P2] borrows a sixty-year-old machine.

A solver returns an answer together with a bound on how good that answer is.

Brute-force sampling gives only the answer.

The vocabulary

A **mathematical program** is three things:

Variables the things you get to choose,
 $x_1, \dots, x_N$.

Constraints the rules they must obey,
 $g_j(x) \leq 0$.

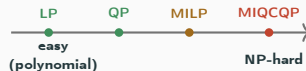
Objective the number you minimise,
 $\min f(x)$.

The **feasible set** is everything obeying the rules.

The **optimum** is the best point in it.

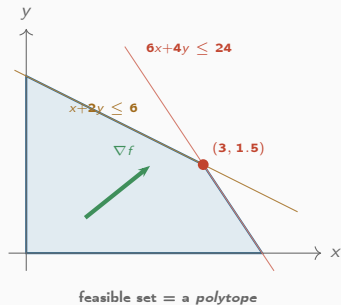
The names are just a taxonomy of how bad f and g are:

	variables	f, g
LP	continuous	linear
QP	continuous	quadratic
ILP	integer	linear
MILP	mixed	linear
MIQCQP	mixed	quadratic



“NP-hard” bounds the *worst* case. Particular instances — including ours — are routinely solved.

Linear programming: the easy case



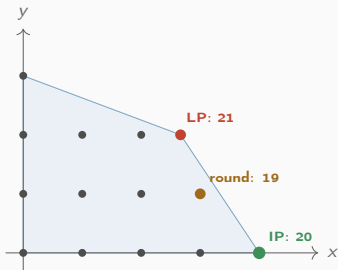
$$\max 5x + 4y \quad \text{s.t.} \quad \begin{cases} 6x + 4y \leq 24 \\ x + 2y \leq 6 \\ x, y \geq 0 \end{cases}$$

- The feasible set is a convex polytope.
- A linear objective is optimised at a **vertex**.
- Simplex walks vertex to vertex; interior-point methods cut through the middle.
- Either way: **fast, and provably optimal**.
Millions of variables are routine.

Here the answer is $x = 3$, $y = 1.5$, objective 21.

But we wanted whole numbers.

Adding integrality



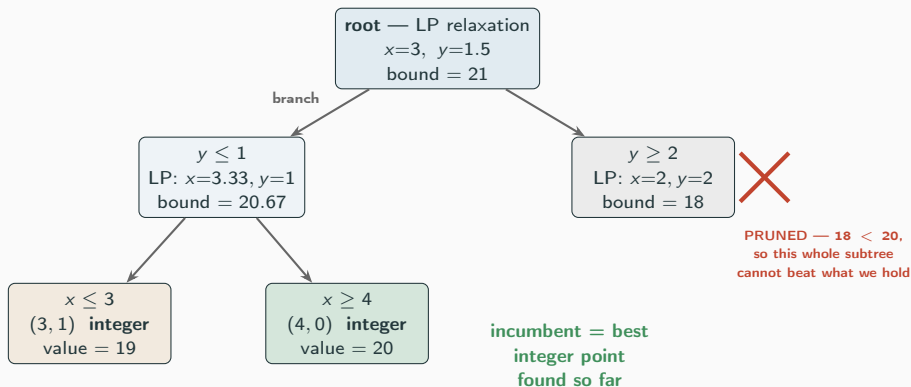
Add $x, y \in \mathbb{Z}$. The feasible set is now a **cloud of points**, not a body — no vertices to walk between, no gradient to follow.

- **LP relaxation** (drop integrality): 21. Always at least as good as the truth, so it is a **valid upper bound**.
- **Rounding** the LP answer $(3, 1.5) \rightarrow (3, 1)$: gives 19. Feasible, but **not optimal**.
- **True optimum**: $(4, 0)$, value 20 — nowhere near the rounded point.

The two bounds

Any feasible point is a **lower** bound on how well you can do (here ≥ 19); the relaxation is an **upper** bound (here ≤ 21). **Optimality is proved when the two meet.**

Branch and bound: divide, bound, prune



At the end: no unexplored node has a bound above 20, and 20 is achieved.
20 is therefore the global optimum.

Why this matters for compilation

Brute force / random sampling [P1]

Returns the best word it happened to see.

$$d^* \leq 0.48$$

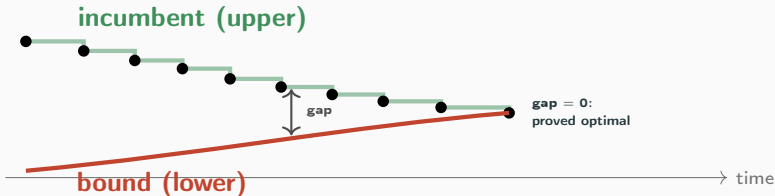
An **upper bound only**: one learns that nothing better was found, not that nothing better exists. Stopping rule: when patience runs out.

Branch and bound [P2]

Maintains *both* bounds throughout:

$$\underbrace{\text{best braid found}}_{\text{upper}} \geq d^* \geq \underbrace{\text{best node bound}}_{\text{lower}}$$

The difference is the **optimality gap**. Gap = 0 means: *no braid of this length is better*. Stopping rule: the gap closes, or time runs out — and one still knows how far off one may be.



Our problem is not linear: bilinear terms

The braid word is grown one **crossing** at a time:

$$Y_t = \left(\sum_{i=1}^m x_{t,i} \rho_i \right) Y_{t-1}$$

- $x_{t,i} \in \{0, 1\}$ — *binary*: “is the t -th crossing the braid i ?”
- $Y_{t-1} \in \mathbb{C}^{n \times n}$ — *continuous*: the braiding matrix so far.

Their product is **bilinear**. That makes this a **MIQCQP** rather than a MILP, and quadratic constraints are much harder: root relaxations need interior-point methods, the dual simplex does not generalise down the tree, and far fewer cutting planes exist.

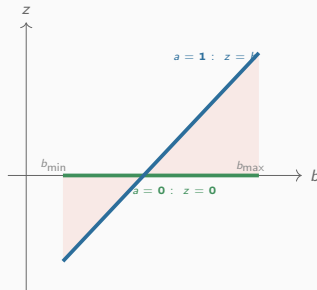
So: turn the bilinear terms into linear ones — exactly.

For $z = ab$ with a **binary**, $b \in [b_{\min}, b_{\max}]$ continuous,

$$\begin{aligned} z &\geq b_{\min} a, & z &\leq b_{\max} a, \\ z &\geq b + b_{\max} a + b_{\min}, & z &\leq b + b_{\min} a + b_{\max} \end{aligned}$$

has **exactly** the same solutions as $z = ab$.

McCormick linearisation



The shaded wedge is the LP relaxation, $a \in [0, 1]$ — where the bound comes from. At $a \in \{0, 1\}$ only the two heavy segments survive.

Why “exactly”

Because a is **binary**. For two *continuous* variables the same four inequalities are only a relaxation — which is why [P2] applies this to the braid-word recursion but **not** to the invariants.

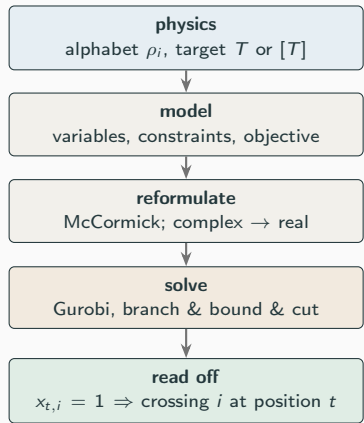
Solvers, and how one drives them

Solvers. Gurobi (used in [P2]), CPLEX, SCIP. All implement **branch-and-bound-and-cut**: branching as above, plus

- *cutting planes* — extra valid inequalities that tighten the relaxation without removing integer points;
- *presolve*, *heuristics*, *symmetry handling*.

Two practical points for [P2]:

- Solvers do **not** do complex arithmetic. Every complex quantity is split into real and imaginary parts.
- Braid length d is **fixed** in advance. To search “length *at most* d ”, put the trivial braid into the alphabet.



In [P2]: 4 cores, 32 GB, 23-hour limit per instance. On timeout you keep the incumbent *and* its gap.

[P2] Compilation as a mixed-integer program

Paper 2

Topological quantum compilation using mixed-integer programming

P. Rytíř, P. C. Burke, C. Aravanis, J. Vala, J. Mareček

- A **general** MIQCQP formulation of quantum compilation — not specific to any anyon model.
- Targets: a gate T ; a local equivalence class $[T]$; the whole set of perfect entanglers.
- Case study: the **non-semisimple Ising** theory, where universality was known only *existentially*. First explicit compilation in it.
- Re-run on the **Fibonacci** alphabet of [P1]: beats the random search.

The formulation: variables

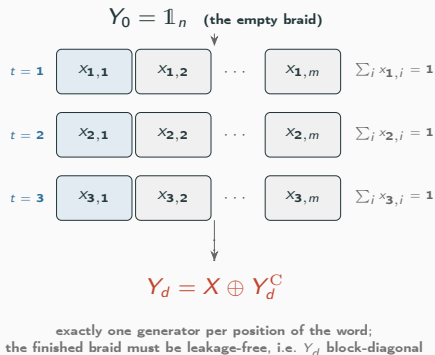
Fix a braid length d : the word is grown one **crossing** at a time, from the alphabet $\mathcal{A} = \{\rho_1, \dots, \rho_m\}$ of elementary braids.

$[Y_t]_{k,\ell} \in \mathbb{C}$ the braiding matrix after t crossings; a new crossing multiplies in on the left,
 $Y_t = \rho_{i_t} Y_{t-1}$, $t = 0, \dots, d$.

$x_{t,i} \in \{0, 1\}$ = 1 iff the t -th crossing is the alphabet's i -th braid;
 $t = 1, \dots, d$, $i = 1, \dots, m$.

The **discrete** decisions are just “which generator at which position of the word”: $d \times m$ binaries. Propagating the Y_t is bookkeeping.

If d is unknown, add the trivial braid $\mathbb{1}$ to the alphabet: the solver may then choose a shorter effective word.



The formulation: constraints and objective

$$\min_{x, Y} J(T, Y_d^C) = \sum_{k, \ell=n_{\text{NC}}+1}^n |T_{k, \ell} - [Y_d]_{k, \ell}|^2 \quad (\text{squared Frobenius distance to the target})$$

$$(i) \quad [Y_t]_{k, \ell} = \sum_{p=1}^n \sum_{i=1}^m x_{t, i} [\rho_i]_{k, p} [Y_{t-1}]_{p, \ell}$$

$$(ii) \quad \sum_{i=1}^m x_{t, i} = 1, \quad t = 1, \dots, d$$

$$(iii) \quad x_{t, i} \in \{0, 1\}$$

$$(iv) \quad [Y_0]_{k, \ell} = \delta_{k, \ell}$$

$$(v) \quad [Y_d]_{k, \ell} = 0, \quad k \leq n_{\text{NC}} < \ell$$

$$(vi) \quad [Y_d]_{k, \ell} = 0, \quad \ell \leq n_{\text{NC}} < k$$

$$(vii) \quad [Y_t]_{k, \ell} \in \mathbb{C}$$

(v)–(vi) force $Y_d = X \oplus Y_d^C$. Since Y_d is unitary, so is Y_d^C ; **no separate unitarity constraint is needed.**

The one nonlinearity

Only (i) is bilinear — binary $\times$ continuous. McCormick replaces each $z_{t, k, \ell}^{p, i} = x_{t, i} [\rho_i]_{k, p} [Y_{t-1}]_{p, \ell}$ by four linear inequalities, **exactly**, and (i) becomes $[Y_t]_{k, \ell} = \sum_p \sum_i z_{t, k, \ell}^{p, i}$.

Reading out the braid

For $t = 1, \dots, d$, the t -th letter of the word is the index i with $x_{t, i} = 1$.

Targeting a *class* instead of a gate

To minimise D_{CNOT} or D_{PE} we must build the invariants g_1, g_2, g_3 *inside* the program. Three things need encoding:

Basis change & m_U $U_B = Q^\dagger U Q$, $m_U = U_B^T U_B$ — same shape as constraint (i).

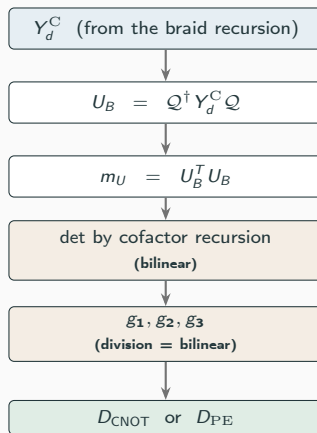
Division $c = a/b$ becomes the bilinear constraint $a = bc$.

Determinant recursive Laplace/cofactor expansion,

$$z_A = \sum_j (-1)^j A_{1j} z_{A,1j},$$

each minor a further bilinear constraint.

Here both factors are continuous, so McCormick would only be a *relaxation*. [P2] leaves these quadratic and lets the solver handle them.



Cost functions:

$$D_{\text{CNOT}}(U) = g_1^2 + g_2^2 + (g_3 - 1)^2$$

$$D_{\text{PE}}(U) = (g_1 - g_3 \sqrt{g_1^2 + g_2^2})^2$$

The test bed: non-semisimple Ising anyons

Ordinary Ising theory has charges $1, \sigma, \psi$ — and is **not** universal. Iulianelli et al. add **one further charge α** , with

$$\alpha \times \sigma = (\alpha + 1) + (\alpha - 1),$$

which makes braiding universal.

One qubit — three anyons:

$$|0\rangle = ((\alpha, \sigma)_{\alpha+1}, \sigma)_{\alpha}, \quad |1\rangle = ((\alpha, \sigma)_{\alpha-1}, \sigma)_{\alpha}$$

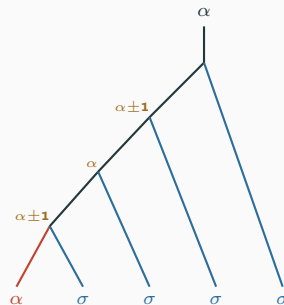
Two qubits — five anyons (α + four σ), $n = 6$:

$$|00\rangle = (((((\alpha, \sigma)_{\alpha+1}, \sigma)_{\alpha}, \sigma)_{\alpha+1}, \sigma)_{\alpha}$$

$$|10\rangle = (((\alpha, \sigma)_{\alpha-1}, \sigma)_{\alpha}, \sigma)_{\alpha+1}, \sigma)_{\alpha}$$

$$|01\rangle = (((\alpha, \sigma)_{\alpha+\mathbf{1}}, \sigma)_{\alpha}, \sigma)_{\alpha-\mathbf{1}}, \sigma)_{\alpha}$$

$$|11\rangle = (((\alpha, \sigma)_{\alpha-1}, \sigma)_{\alpha}, \sigma)_{\alpha-1}, \sigma)_{\alpha}$$

 $|NC_1\rangle, |NC_2\rangle$ (the two non-computational states)

the two $\pm$ choices are the two qubits:

$$n = 6, n_G = 4, n_{NG} = 2$$

► Universality here was proved existentially. [P2] asks what the gates *are*.

The non-semi alphabet

With $q = e^{2\pi i/8}$ and α real,

$$B_{\alpha+1} = \frac{\sqrt{2}}{-1 + \cot \frac{\pi(\alpha+1)}{4}}, \quad B_{\alpha-1} = \frac{\sqrt{2}}{-1 + \cot \frac{\pi\alpha}{4}},$$

$$(b_1^{\alpha\sigma\sigma})^2 = - \begin{pmatrix} q^{\alpha-1} & 0 \\ 0 & q^{-\alpha-1} \end{pmatrix}, \quad b_2^{\alpha\sigma\sigma} = q^{\frac{1}{2}} \begin{pmatrix} \frac{1+q^2}{1-q^{2\alpha}} & q^{-1} \sqrt{B_{\alpha+1}/B_{\alpha-1}} \\ q^{-1} \sqrt{B_{\alpha+1}/B_{\alpha-1}} & \frac{1+q^2}{1-q^{2\alpha}} \end{pmatrix}.$$

On two qubits (first qubit / second qubit):

$$b_1^{(2)2} = (b_1^{\alpha\sigma\sigma})^2 \otimes \mathbb{1}_2 \oplus (b_1^{\alpha\sigma\sigma})^2, \quad b_2^{(2)} = b_2^{\alpha\sigma\sigma} \otimes \mathbb{1}_2 \oplus q^{\frac{1}{2}} \mathbb{1}_2,$$

$$J_4 = \mathbb{1}_2 \otimes (b_1^{\alpha\sigma\sigma})^2 \oplus \begin{pmatrix} q^{1-\alpha} & 0 \\ 0 & q^{1+\alpha} \end{pmatrix}, \quad b_4 = \mathbb{1}_2 \otimes b_2^{\alpha\sigma\sigma} \oplus q^{\frac{1}{2}} \mathbb{1}_2.$$

To these is added the entangling **controlled phase**, built by Iulianelli et al. at $\alpha = 2.4$ via Reichardt's iteration (leakage $\mathcal{O}(10^{-14})$, treated as zero):

$$\text{CPHASE} = \text{diag}(1, 1, e^{-1.772i}, e^{-1.682i}) \oplus V.$$

V acts only on V_{NC} and is taken to be $\mathbb{1}_2$.

Experiment 1: the CNOT gate in non-semi (1/2)

Length	$J(\cdot)$	$d_2(\cdot, \text{CNOT})$	Braid word
150	0.163624	0.202252	$4^{13} 0 4^{23} 3 4^{19} 0 4^9 0 4^5 0 4^{41} 2 4^{23} 0 4^8 2 3$
161	0.353581	0.297313	$4^5 0^2 4^{61} 2 4^{46} 0 4^3 0^2 4^5 3 2 4^{31} 3 4$
88	0.456191	0.337710	$4 0 2 0 4^4 0 4 1 4^4 2 4 2 4 1 2 4^7 2 0 4^6 2 4 3 4^2 0 2^2 4^{14} 2 4^5 0 4 2$ $4 0 4^7 0 4 0 4^3 0 4^2 3^3$
101	0.511175	0.357482	$4^{31} 0 4^{21} 3 4^3 2 4^2 0 4^{10} 2 4^7 2 0 4^{10} 2 4^7 3 4$
59	0.532829	0.364976	$4^3 0 3 0 4^4 2 0 2 4^5 3 4^2 2 4^{19} 2 4^3 0 4^3 3 4^3 0 4 2 0 3^2$
44	0.585746	0.382670	$4^3 0^2 4 3^3 4^{11} 0 4^{10} 0 4^7 3 2^3 3$
61	0.904904	0.475633	$4^8 2 4 3 4 3 4 2 4 0^2 4 3 4^{10} 0 4^8 2 4^9 2 3 0 2 0 4^5 0 3$
68	0.913233	0.477816	$4 3 4^4 3 4^3 2 4^{11} 0 4^{19} 0 2 4^4 0 4 3 4^3 0 4^3 2 4 0 4 0 4^3 2 3$
81	0.927386	0.481505	$4 0 4^8 3^2 4^9 3 4^{13} 3 2 3 4^3 0 4^4 0 2 4^3 0 4^5 3 4^4 0 4^4 2 4 2 0 4 0^3 2$ $4^2 2 3 4$
71	1.087120	0.521326	$4 3 4 3 4^4 0 4^2 0 4^8 2 4^3 1 4^3 2 4^2 1^2 4^3 3 4^2 0 4 1 4 2 4 1 4^9 3 1 4 3$ $0 4^3 1 4^2 2 0^2 4 3$

[P2] Table I (rows 1–10), ordered by $d_2(\cdot, \text{CNOT})$. Lengths 10–200. Sequences shown in *run-length* notation: 4^{13} means thirteen consecutive 4's (CPHASE).

The alphabet, indexed

$$(b_1^{(2)^2}, b_2^{(2)}, J_4, b_4, \text{CPHASE}) \rightarrow (0, 1, 2, 3, 4), \quad m = 5, \quad \alpha \text{ fixed at 2.4 throughout.}$$

Experiment 1: the CNOT gate in non-semi (2/2)

Length	$J(\cdot)$	$d_2(\cdot, \text{CNOT})$	Braid word
56	1.108356	0.526393	$4^4 0 4 0 4^4 2 4^2 0 4^2 3 4^8 2 4 0 4^6 2 4^8 1 3 4^3 3 4 3 4^2 2 1$
132	1.164335	0.539522	$3 0 4 0 1 4^2 2 4 0 2 0 2 0 3 4^3 0^2 4 0 2 4 0 4 1 4 3 4^3 0 4 3 2 0 1 4^8$ $2 4 2 0 2 4 0 4^4 0 4 0^2 4 0^2 4 0 4 2 4 0 4 2 4^2 0 3 0 4 0 4 0^2 1 3 2 4^2$ $2 4 0 4^2 0^2 1 3 0 4^2 0 4 2 0 4 1 0 4 0 4^3 0 1 4 0 1 2 3 0 4 0^2 4^2 0^4 4$ $2 0 3$
37	1.167666	0.540293	$4 0 4^3 0 4^3 0 3 4 0 4^{19} 3 0 4 2 0$
39	1.193768	0.546299	$4 3 4^{12} 3 4^2 2 4^3 0 4^2 3 0 3 4 0 4^2 2 4 3 4^4 3$
72	1.281908	0.566107	$4^3 2 4 0 4^9 0 4^{15} 3 4 3 4^4 3 4^9 0 4^8 3 2^2 0 2 3 4^9$
169	1.288397	0.567538	$0 2 3 1^3 3 2^2 0^2 4 3 0 4^3 0 4 1 4 1 0^3 4^2 0 2 4 0 2^2 4 0 4^4 2 0 4^4 0 4$ $3 2^2 3 4 1 4 0 4 0 4 2 0 2 0^3 4 1 0^2 4^2 0^3 4^3 2 4 0^4 3 4 0 1^2 4^3 0 2 0^2$ $3 4^4 0 4 0 4 0 4^2 2 4^3 0 3 4 1 4 0^2 4^2 0 1 4^4 0^2 1^3 0^2 4^3 0 4 0^2 4 0 3$ $4 2 0^2 4 0 1 0^2 4^7 3 4 0 1 0^2 4^3 2 4$
47	1.313629	0.573068	$4^3 3 4^4 3 4^{16} 3 1 3^2 4 3^2 4 1 4^{11} 3^2$
32	1.336065	0.577942	$3 4^6 2 4^4 3 2 3 4^3 3 2^2 4^7 0 4 3^2$
90	1.343007	0.579441	$4^9 3 4^6 2 4^3 0 4 2 4^6 0 1 4^2 3 4^{11} 0 4^6 1 4^3 0 4 1 4 1 0 3 4^3 0 4 2 3 4$ $0 4^6 3 4^5 0 4^2 3^3$
41	1.355551	0.582140	$3 4^2 3 4^5 0^2 2 3 4 0 3 4 2 4 3 0 4^8 0^2 2 4 0 4^2 3 2 3 0 3$

[P2] Table I (rows 11–20).

► Best $d_2 \approx 0.20$ at length 150. **Failure** — just as in [P1], and for the same reason: a single point in a 15-dimensional manifold, reached with five letters.

Experiment 2: the class [CNOT] in non-semi (1/2)

Length	$D_{\text{CNOT}}(\cdot)$	Braid word
35	1.5617e-09	4^{35}
34	1.3831e-05	4^{34}
36	2.9260e-05	4^{36}
33	2.6959e-04	4^{33}
37	3.9145e-04	4^{37}
32	1.4467e-03	4^{32}
38	1.8533e-03	4^{38}
44	2.1372e-03	$4^{24} 3 4^2 0 4^9 3 4^6$
61	2.3334e-03	$4^7 3 4^{24} 3 4^9 3 4^{10} 3 4^7$
31	4.6776e-03	4^{31}

[P2] Table II (rows 1–10), cost D_{CNOT} , lengths 5–80.

► Length 35: $D_{\text{CNOT}} = 1.6 \cdot 10^{-9}$ — four orders of magnitude below the usual fault-tolerance threshold of 10^{-4} – 10^{-5} . And the word is CPHASE³⁵.

Experiment 2: the class [CNOT] in non-semi (2/2)

Length	$D_{\text{CNOT}}(\cdot)$	Braid word
39	5.6276e-03	4^{39}
40	5.6277e-03	$4^{39} 0$
30	1.1503e-02	4^{30}
29	2.3804e-02	4^{29}
41	2.6880e-02	4^{41}
28	4.3733e-02	4^{28}
42	4.8488e-02	4^{42}
50	6.1524e-02	$3\ 4\ 3\ 4^5\ 3\ 4^{21}\ 3\ 4^{11}\ 0\ 4^5\ 2\ 4$
27	7.3615e-02	4^{27}
43	8.0495e-02	4^{43}

[P2] Table II (rows 11–20). The distance is **monotone in the exponent** about 35, rising on either side: the solver has found a one-parameter family and located its minimum, not a scatter of lucky words.

Experiment 3: an arbitrary perfect entangler (1/2)

Length	$D_{\text{PE}}(\cdot)$	Braid word
77	0.0000e+00	1 4 0 4 ⁴ 0 ² 4 ² 0 2 ² 0 4 ² 0 ³ 4 ² 0 4 ² 2 0 4 ² 0 2 4 0 4 ² 2 0 4 ² 2 0 2 ² 0 4 ² 0 4 ² 0 4 ⁴ 0 ³ 4 0 ² 2 4 2 0 2 4 0 4 0 ² 4 2 4 0 2 ² 1
35	0.0000e+00	4 ³⁵
34	3.0622e-11	4 ³⁴
36	1.3698e-10	4 ³⁶
33	8.3225e-09	4 ³³
37	1.9960e-08	4 ³⁷
32	3.2197e-07	4 ³²
38	5.4960e-07	4 ³⁸
31	3.4437e-06	4 ³¹
40	5.0489e-06	4 ³⁸ 0 4

[P2] Table III (rows 1–10), cost D_{PE} , lengths 5–80.

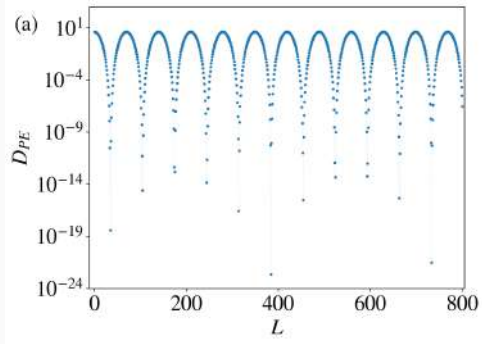
- Two words reach $D_{\text{PE}} = 0$ at machine precision: a length-77 word, and again CPHASE³⁵.

Experiment 3: an arbitrary perfect entangler (2/2)

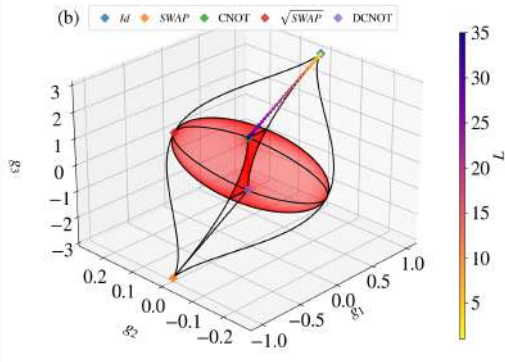
Length	$D_{PE}(\cdot)$	Braid word
39	5.0611e-06	4^{39}
30	2.1170e-05	4^{30}
29	9.0532e-05	4^{29}
41	1.1544e-04	4^{41}
28	3.0602e-04	4^{28}
42	3.7618e-04	4^{42}
27	8.6702e-04	4^{27}
43	1.0365e-03	4^{43}
26	2.1474e-03	4^{26}
44	2.5122e-03	4^{44}

[P2] Table III (rows 11–20). Relaxing from [CNOT] to *any* perfect entangler buys several more orders of magnitude: **the larger the target, the better the compilation.**

Why CPHASE^L?



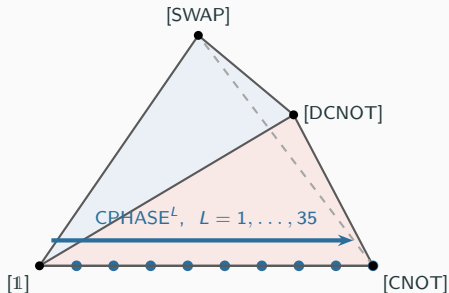
(a) D_{PE} of CPHASE^L vs. L



(b) the same powers in (g_1, g_2, g_3) , up to $L = 35$

[P2] Fig. 1. Perfect-entangler region in red. The invariants of CPHASE are close to $(1, 0, 3)$ — i.e. close to $[1]$. So [CPHASE] sits on the Weyl-chamber **edge** from $[1]$ to [CNOT], and its powers **walk along that edge**, arriving at the [CNOT] vertex at $L \approx 35$.

The same picture, drawn by hand



A sanity check on the method

The optimiser was told only “minimise D_{PE} ”. It returned the answer a theorist would have guessed from the Cartan decomposition — **repeat the native controlled-phase gate**.

And a little more besides

The solver *also* returned words that no intuition would produce — e.g. the length-77 perfect entangler 1404444...221, and the length-44 and length-61 mixed words in Table II.

Experiment 4: the same machine on Fibonacci anyons

To show the formulation is **not** tied to one anyon model, [P2] re-runs it on the FIB alphabet of [P1]:

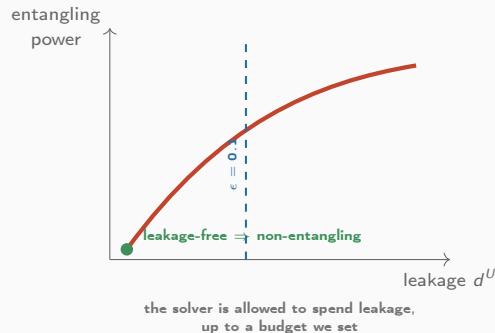
$$(\rho_6(\sigma_1), \dots, \rho_6(\sigma_5)) \rightarrow (0, 1, 2, 3, 4), \quad n = 5, \quad n_C = 4.$$

But here leakage **cannot** be forbidden outright — Cui et al.: leakage-free $\Rightarrow$ non-entangling. So constraints (v)–(vi) are **softened** to tolerate $\epsilon > 0$:

$$(L1) \quad \sum_{k, \ell > n_{NC}} \left| \sum_{p > n_{NC}} [Y_d]_{kp}^* [Y_d]_{p\ell} - \delta_{k\ell} \right| < \epsilon$$

$$(L2) \quad \sum_{k, \ell \leq n_{NC}} \left| \sum_{p \leq n_{NC}} [Y_d]_{kp}^* [Y_d]_{p\ell} - \delta_{k\ell} \right| < \epsilon$$

which give $d^U(A) < \epsilon$ and $d^U(M_{11}) < \epsilon$. In the runs, $\epsilon = 0.1$.



Experiment 4: CNOT with the fib alphabet (1/2)

Length	$J(\cdot)$	$d_2(\cdot, \text{CNOT})$	$ M_{11} $	$d^U(\cdot)$	Braid word
23	1.4234	0.5998	0.9539	0.0902	34030022213410003301100
21	1.5604	0.6280	0.9539	0.0902	24440230433233334020
16	1.8866	0.6906	0.9539	0.0902	2014433003412443
14	1.8895	0.6911	0.9539	0.0902	43242444234303
18	1.9178	0.6963	0.9538	0.0902	323444432343402340
13	2.1081	0.7300	0.9539	0.0902	4334432224034
11	2.1472	0.7368	0.9539	0.0902	34342224034
19	2.2222	0.7495	0.9539	0.0902	1404440331422203444
15	2.3874	0.7769	0.9539	0.0902	233210033143234
38	2.4612	0.7888	0.9538	0.0902	14303110341440410310310043204223433433

[P2] Table IV (rows 1–10), $\epsilon = 0.1$, ordered by $d_2(\cdot, \text{CNOT})$.

► $d_2 = 0.5998$ at length 23, against 0.48 at **length 31** from [P1]'s 10^{12} random samples — at a fraction of the length, and with a bound attached.

Experiment 4: CNOT with the fib alphabet (2/2)

Length	$J(\cdot)$	$d_2(\cdot, \text{CNOT})$	$ M_{11} $	$d^U(\cdot)$	Braid word
23	2.5500	0.7984	1.0000	7.3560e-05	03301443004314340043030
17	2.5501	0.7985	1.0000	6.4084e-05	43343000443303004
33	2.5502	0.7985	1.0000	3.8759e-05	000443440103310300004000040403103
28	2.5855	0.8040	1.0000	1.0921e-04	3333330000040000004043000033
38	2.6064	0.8072	1.0000	1.5908e-05	13330000300001100030003101100043340033
22	2.6191	0.8092	1.0000	9.5051e-05	0000000000000000000003
21	2.6192	0.8092	1.0000	4.9379e-05	022444432042302442044
32	2.6192	0.8092	1.0000	1.2882e-04	00000030000000000000000000000000
16	2.6192	0.8092	1.0000	8.6737e-05	0444241044041243
12	2.6193	0.8092	1.0000	4.7810e-05	044444444443

[P2] Table IV (rows 11–20). The table splits in two: the upper rows **spend** leakage ($d^U \approx 0.09$) and come closer to CNOT; the lower rows are essentially leakage-free ($d^U \sim 10^{-5}$) and stall at $d_2 \approx 0.81$. **The [P1] trade-off, as a discrete jump.**

Experiment 5: the class [CNOT] with the fib alphabet

Length	$D_{\text{CNOT}}(\cdot)$	$ M_{11} $	$d^U(\cdot)$	Braid word
21	1.7083e-05	0.9759	0.0476	220404043041300143224
42	0.4083	0.9718	0.0556	321001404003203240422402014110201121244331
5	2.9065	0.9539	0.0902	12220
6	2.9065	0.9539	0.0902	042224
8	2.9066	0.9539	0.0902	24244442
11	2.9066	0.9539	0.0902	02440200234
20	2.9066	0.9539	0.0902	24310334121244010214
7	2.9066	0.9539	0.0902	4432022
17	2.9066	0.9539	0.0902	24420240444444414
9	2.9066	0.9539	0.0902	213330012

[P2] Table V, cost D_{CNOT} , $\epsilon = 0.1$.

Length 21 reaches $D_{\text{CNOT}} = 1.7 \cdot 10^{-5}$ with $d^U = 0.048$ — comparable to [P1]’s Table IV, obtained without sampling 10^{12} words.

The long tail at 2.9066 is the cluster of short words that never leave the neighbourhood of [1]: visible in [P1] too, as the rows with $d = 2.9082$.

Contributions

- A **general** MIQCQP formulation of quantum compilation: alphabet in, braid word out.
- McCormick makes the braid-word recursion *exactly* linear.
- Local equivalence classes and perfect entanglers encoded as objectives — determinants, divisions and all.
- First explicit braids in the **non-semisimple Ising** theory, turning an existential universality proof into a constructive one.
- Model-independence demonstrated by re-running on Fibonacci.

The recurring pattern

Across both papers and all three alphabets:

target a gate	fails
target a class	$\sim 10^{-9}$
target any perfect entangler	$\sim 10^{-16}$

Honest limitations

23-hour cap per instance: many reported solutions are *incumbents*, not closed-gap optima. The formulation grows with $d \times m \times n^2$, so very large lengths remain out of reach.

Synthesis

Two routes to the same braids

	[P1] exhaustive / random search	[P2] mixed-integer programming
Alphabets	Fibonacci (basic, extended)	non-semisimple Ising; Fibonacci
Reach	exhaustive $L \leq 14$; sampled $L \leq 40$	braid length up to 200, by formulation
Effort	$\sim 10^{12}$ trials	4 cores, 32 GB, ≤ 23 h per instance
Guarantee	upper bound only	upper and lower bounds
Leakage	filtered <i>after</i> the product	a constraint, with budget ϵ
Braid length	implicit in the enumeration	a decision variable (via $\mathbb{1}$ in the alphabet)
Best [CNOT] (Fib.)	$d \approx 8 \cdot 10^{-9}$ at $L = 30$	$D \approx 1.7 \cdot 10^{-5}$ at $L = 21$
Best CNOT (Fib.)	$d_2 \approx 0.48$ at $L = 31$	$d_2 \approx 0.60$ at $L = 23$

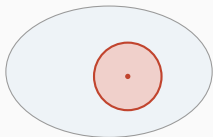
They agree where they overlap

Neither reaches exact CNOT at these lengths; both find the $[\mathbb{1}]$ –[CNOT] edge; both hit the leakage/entanglement trade-off. **Independent methods, same physics.**

They differ in what they can promise

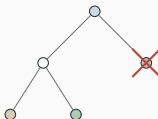
[P1] reports what *was found*; [P2], what *exists*, and when the gap closes, what *cannot* be bettered.

The three lessons



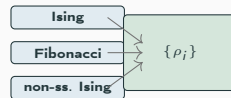
1. Enlarge the target

A gate is a point in 15 dimensions; its class is a point in 3. Every order of magnitude in accuracy came from this one move — in both papers, on all three alphabets.



2. Bounds, not just answers

Sampling 10^{12} of 10^{28} words is a fraction 10^{-16} . Branch and bound *proves* things about words it never enumerates.



3. The method does not care

Ising, Fibonacci, non-semisimple — the compiler sees only a finite set of unitaries and a computational subspace, and nothing topological at all.

Where this goes next

Open questions from the physics

- Are there *any* exactly leakage-free entangling Fibonacci braiding gates? Both papers point to **no** — Cui et al.'s conjecture stands.
- Why do braids concentrate on the [1]–[CNOT] and [DCNOT]–[SWAP] edges? A group-theoretic explanation would be more satisfying than a scatter plot.
- Richer alphabets: the current ones contain effectively *one* two-qubit operation, which may simply be too coarse.

Open questions from the optimisation

- **Lower bounds on braid length**: proof-complexity arguments could show that no word below some length reaches a target.
- Symmetry breaking: the braid relations $\sigma_k \sigma_{k+1} \sigma_k = \sigma_{k+1} \sigma_k \sigma_{k+1}$ generate much symmetry in the search tree; exploiting it should pay.
- Scaling to more qubits, where n grows and the formulation grows as $d m n^2$.
- Global-phase-invariant objectives, warm starts from [P1]-style sampling.

► Compilation turns an existence proof into an explicit braid. That step is an **optimisation problem**, and deserves the tools of one.

Thank you



[P1] P. C. Burke, C. Aravanis, J. Aspman, J. Mareček, J. Vala,
Topological quantum compilation of two-qubit gates
Phys. Rev. A **110**, 052616 — [doi:10.1103/PhysRevA.110.052616](https://doi.org/10.1103/PhysRevA.110.052616)



[P2] P. Rytíř, P. C. Burke, C. Aravanis, J. Vala, J. Mareček,
Topological quantum compilation using mixed-integer programming
IEEE Trans. Quantum Eng. **7**, 3104212 (2026) — [doi:10.1109/TQE.2026.3710775](https://doi.org/10.1109/TQE.2026.3710775)

Building on Cui et al. (2019), Iulianelli et al. (2024), Zhang–Vala–Sastry–Whaley (2003),
Makhlin (2002), Watts et al. (2015), Bonesteel et al. (2005), Reichardt (2012).

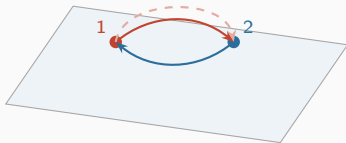
Backup

Backup

Standard background, held in reserve

Two dimensions are special

3D: exchange paths are all equivalent



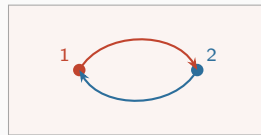
any two exchange paths can be deformed into one another

$\Rightarrow \pi_1$ of configuration space = S_n

two 1D irreps only:

bosons (+1) and fermions (-1)

2D: the path remembers how it wound



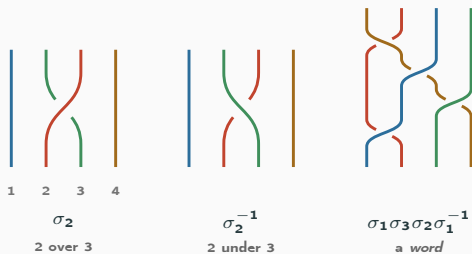
clockwise *cannot* be deformed into anticlockwise without passing through

$\Rightarrow \pi_1$ of configuration space = B_n

the braid group is *infinite*:
hence *anyons*

► In 2D exchange is governed by the **braid group** B_n , not S_n . Its representations are the physics.

The braid group B_n



Generators $\sigma_1, \dots, \sigma_{n-1}$; σ_k exchanges anyons k and $k+1$.

Relations:

$$\sigma_k \sigma_{k+1} \sigma_k = \sigma_{k+1} \sigma_k \sigma_{k+1}$$

$$\sigma_k \sigma_\ell = \sigma_\ell \sigma_k, \quad |k - \ell| \geq 2$$

Crucially, $\sigma_k^2 \neq \mathbb{1}$ — that is exactly what S_n imposes and B_n does not.

A braid word is a **history**: read upwards, quasi-particles dragged about one another.

All that follows is a search for the right *word* in these letters.

Abelian anyons: braiding is a phase



full monodromy (two exchanges)

The ground state is **non-degenerate** for a fixed set of anyon positions. A braid can therefore only multiply the state by a number:

$$|\Psi\rangle \mapsto e^{i\theta} |\Psi\rangle, \quad \theta \in (0, \pi) \text{ generally}$$

- $\theta = 0$: bosons; $\theta = \pi$: fermions; otherwise: **Abelian anyon**
- the B_n representation is **one-dimensional** $\Rightarrow$ all braids commute
- e.g. Laughlin quasiholes at $\nu = 1/3$, $\theta = \pi/3$

Consequence for computing

Braiding gives only $U(1)$ phases: **not universal**, not even entangling. Excellent *memories* — degeneracy on a torus — but not processors.