Trivariate Tricycle Codes

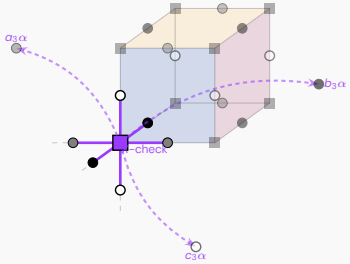
High-rate codes with a cup-product CCZ

Dan Browne

joint work with Abraham Jacob and Campbell McLauchlan

FTQT 2026, Benasque • August 2026 • arXiv:2508.08191

University College London



This talk

1. Motivation: the runtime problem in qLDPC architectures
2. Warm-up: Bivariate Bicycle codes
3. Trivariate Tricycle codes: construction, single-shot decoding, performance and Clifford gates
4. Non-Clifford gates — and a gentle introduction to the cup product
5. *CCZ* gates in TT codes
6. Related work: Magic Tricycles
7. Where we stand — and the gap that is left
8. New codes from old: covering graphs, and work in progress

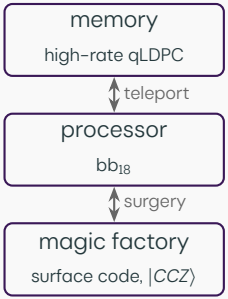
Motivation: the runtime problem

Shor's algorithm on 10,000 qubits

Full architectures for fault-tolerant quantum computing with qLDPC codes have now been proposed[1,2,3], leading to substantial reductions in overhead.

Cain et al. (March 2026): RSA-2048 is possible with ∼10,000 *reconfigurable atomic qubits*.

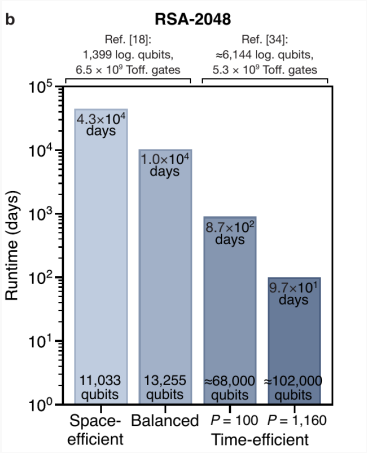
- **Memory**: lifted-product qLDPC, $\mathrm{lp}_{20}^{3,7} = [\![4350, 1224, \leq 20]\!]$ (∼30% rate)
- **Processor**: bivariate bicycle code $\mathrm{bb}_{18} = [\![248, 10, \leq 18]\!]$
- **Factory**: 8*T*-to-*CCZ* distillation on surface-code patches



Logic: CCZ state distillation + generalised lattice surgery

[1]Yoder et al., *Tour de gross*, arXiv:2506.03094. [2]Webster et al., *Pinnacle*, arXiv:2602.11457. [3]Cain et al., arXiv:2603.28627. bb_{18}: Symons, Rajput & Browne, arXiv:2511.13560; Liang et al., arXiv:2503.03827.

The catch



- Space-efficient: 4.3×10^4 days $\approx$ **120 years**
- Balanced: 1.0×10^4 days $\approx$ 27 years
- 97 days? Costs ~102,000 qubits (10× the space-efficient count)

Clear trade-off between *time* and *space*.
Can we reduce *both* space and time?

Cain et al., arXiv:2603.28627, Fig. 3b (annotations to one significant figure).

Where do 100 years go?

Anatomy of one logical Toffoli in the space-efficient architecture:

1. Distil a $|\overline{CCZ}\rangle$ magic state ($8T$-to-CCZ, surface-code factory)
2. Consume it via lattice-surgery Pauli-product measurements — each one surgery cycle of $\frac{2d}{3}$ QEC rounds
3. Amortised over RSA-2048: $\sim$43 sequential surgery cycles per Toffoli, so $\tau_{\mathrm{Toff}} \approx 43 \cdot \frac{2d}{3}$ QEC cycles at 1 ms/cycle $\approx$ 0.5 s per Toffoli

$$6.5 \times 10^9 \text{ Toffolis} \times \text{sequential execution} = \sim 120 \text{ years}$$

The Cliffords are (comparatively) free — **non-Clifford gates dominate the runtime**, through distillation and surgery.

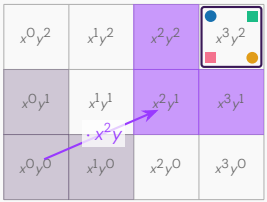
Three ways out

This talk: a code family with **single-shot error correction**, **transversal Cliffords**, and a **native non-Clifford gate** — *Trivariate Tricycle codes*.

Warm-up: Bivariate Bicycle codes

BB codes: polynomials as parity checks

- Qubit positions on an $\ell \times m$ torus $\leftrightarrow$ monomials $x^i y^j$, with $x^\ell = y^m = 1$
- A polynomial = a *set* of positions; multiplying by a monomial *shifts* the set
- $x = S_\ell \otimes I_m$, $y = I_\ell \otimes S_m$ (cyclic shifts) $\Rightarrow$ a polynomial $A(x, y)$ *is* an $\ell m \times \ell m$ binary matrix
- Two qubit blocks L, R: $H_X = [\, A \mid B \,]$, $H_Z = [\, B^\top \mid A^\top \,]$
- $A = 1 + x$, $B = 1 + y \Rightarrow$ the 2D toric code



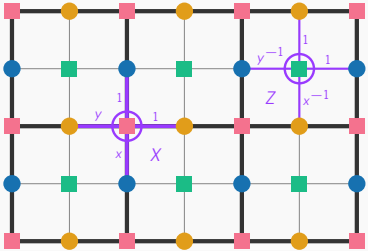
$A = 1{+}x{+}y \longmapsto x^2y\,A \quad (\ell{=}4, m{=}3)$

Every site carries ● L and ● R qubits,

■ X and ■ Z checks

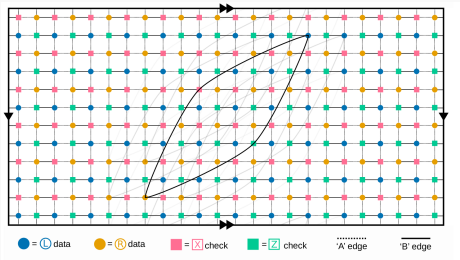
Bravyi, Cross, Gambetta, Maslov, Rall & Yoder, Nature 627, 778 (2024).

The Tanner graph view



toric code: $A = 1+x,\ B = 1+y$

X: $[A \mid B]$ $\quad$ Z: $[B^\top \mid A^\top]$



gross code $[\![144, 12, 12]\!]$: $A = x^3+y+y^2,\ B = y^3+x+x^2$

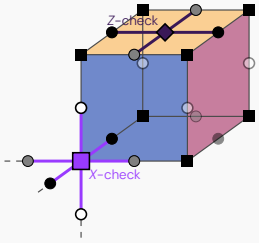
- **Each monomial contributes one edge**: the graph is determined by ℓ, m, A, B — higher monomials give *long-range* edges
- For the toric code the **darker lines** show the Kitaev lattice:
  - **Qubits** sit on **edges**, X**-checks** on **vertices**.

  This is the cohomology picture we use later – keep this in your head!

Right figure: Bravyi, Cross, Gambetta, Maslov, Rall & Yoder, Nature 627, 778 (2024), Fig. 1.

Trivariate Tricycle codes

Why go 3D at all?



the **3D toric code**

The 3D toric code is the canonical 3D CSS code, and it has two properties that BB codes do not have:

- **A native non-Clifford gate**: three copies admit a transversal, constant-depth $\overline{CCZ}$
- **Single-shot decoding** in the Z basis — redundant checks let you *repair* a noisy syndrome rather than repeat the measurement $O(d)$ times

…but the rate is low: $k = 3$, with n growing as d^3.

This work's aim: bring the 3D structure — and both properties — to the BB formalism to find high rate codes with these properties.

Transversal CCZ: Vasmer & Browne, Phys. Rev. A **100**, 012312 (2019). Single-shot decoding: Quintavalle et al., PRX Quantum **2**, 020340 (2021).

Where this starts

Single-Shot Decoding and Fault-tolerant Gates with Trivariate Tricycle Codes

Abraham Jacob,[1,*] Campbell McLauchlan,[1,2,*] and Dan E. Browne[1]

Abraham Jacob

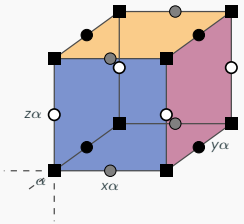
Campbell McLauchlan

A three-variable cousin of the BB codes: high rate, single-shot decoding, and a native non-Clifford gate.

Everything up to the cup product comes from this paper.

A. Jacob, C. McLauchlan & D. E. Browne, *Single-shot decoding and fault-tolerant gates with trivariate tricycle codes*, arXiv:2508.08191.

Generalising polynomial framework from 2D to 3D



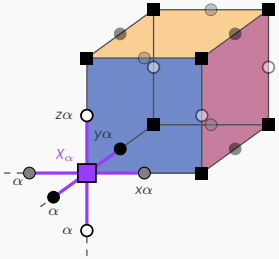
- X-check (vertex)
- L qubit (x-edge)
- C qubit (y-edge)
- R qubit (z-edge)
- Z_a-check (yz-face)
- Z_b-check (xz-face)
- Z_c-check (xy-face)

meta-check = cube

Same basic idea, now with **three** variables x, y, z on a periodic $\ell \times m \times p$ cubic lattice — every edge, vertex and face carries a monomial $x^i y^j z^k$.

- **Qubits on edges** — three families L, C, R, so $n = 3\ell mp$
- **X-checks on vertices**; **Z-checks on faces**, one family per orientation
- Cubes give meta-checks — a fourth level that BB codes don't have

The 3D toric code

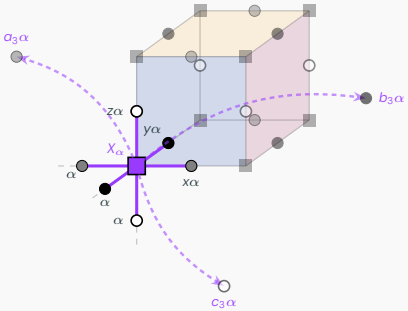


Take $A = 1 + x$, $B = 1 + y$, $C = 1 + z$:

- Each qubit and check is named by its monomial ($x11 = x$, $11z^2 = z^2$, ...)
- A check's polynomial lists the neighbours it touches — exactly as in 2D
- Stabiliser weights: X: **6**, Z: **4**

$$H_X = \begin{bmatrix} A & B & C \end{bmatrix}, \qquad H_Z = \begin{bmatrix} 0 & C^\top & B^\top \\ C^\top & 0 & A^\top \\ B^\top & A^\top & 0 \end{bmatrix}, \qquad M_Z = \begin{bmatrix} A^\top & B^\top & C^\top \end{bmatrix}$$

Trivariate Tricycle codes



Now let A, B, C be *any* three-term polynomials — each the check matrix of a classical code. The TT code is their **balanced product**:

- **Same local monomial layout**, but long-range connections — 3 extra per X-check, 2 per Z-check
- Weights climb: X: **9**, Z: **6**

$$H_X = \begin{bmatrix} A & B & C \end{bmatrix}, \qquad H_Z = \begin{bmatrix} 0 & C^\top & B^\top \\ C^\top & 0 & A^\top \\ B^\top & A^\top & 0 \end{bmatrix}, \qquad M_Z = \begin{bmatrix} A^\top & B^\top & C^\top \end{bmatrix}$$

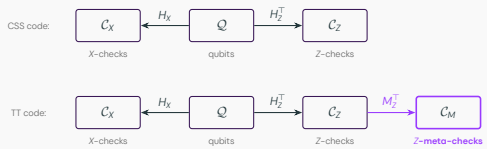
Formally: a balanced product of three group-algebra codes over $\mathbb{Z}_\ell \times \mathbb{Z}_m \times \mathbb{Z}_p$.
The same 4-term complex arises independently as the $D{=}3$ case of *abelian multi-cycle* codes — Lin, Lim, Kovalev & Pryadko, arXiv:2506.16910 — where the focus is $D{=}4$ and single-shot decoding.

A TT code zoo

$[\![n, k, d_Z]\!]$	rate	ℓ, m, p	A	B	C	d_X
$[\![72, 6, 6]\!]$	1/12	4, 3, 2	$1+y+xy^2$	$1+yz+x^2y^2$	$1+xy^2z+x^2y$	12
$[\![180, 12, 8]\!]$	1/15	5, 4, 3	$1+x^2y^3z+x^4y$	$1+x^3+x^4z^2$	$1+x^3y^3+x^4yz^2$	20
$[\![432, 12, 12]\!]$	1/36	6, 6, 4	$1+xyz^3+x^3y^4z^2$	$1+x^3yz^2+x^3y^2z^3$	$1+x^4y^3z^3+x^5z^2$	≤ 36

- Found by random search over polynomials
- $[\![432, 12, 12]\!]$: $48\times$ fewer data qubits than 12 copies of the d=12 3D toric code
- Always $d_X > d_Z$ — natural fit for *biased-noise* hardware

What the extra structure gives you



Single-shot decoding

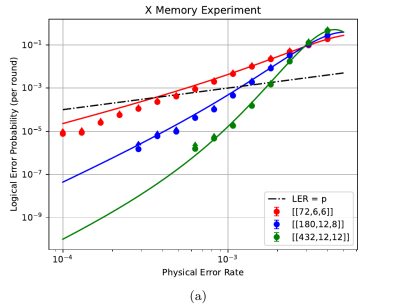
- Syndrome measurements are noisy; usually we repeat $O(d)$ rounds before decoding
- Meta-checks flag *inconsistent* Z-syndromes: repair the syndrome first, then decode $\Rightarrow$ single-shot Z-basis decoding
- **Meta-check distance:** $d_M = d_Z$ (Lemma 2)

Clifford gates

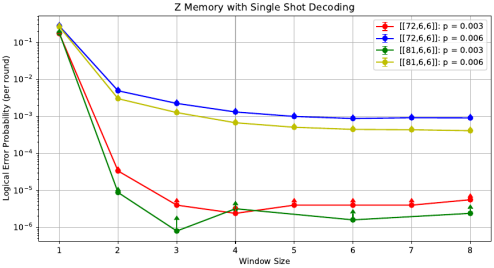
- *Shift automorphisms*: depth-2 SWAP circuits, permuting logicals *within* a block
- *Transversal* $\overline{CZ}$: depth-1, *between* blocks, three families, partially addressable

Automorphism gates: Sayginel et al., arXiv:2409.18175. Fold-transversal gates: Breuckmann & Burton, Quantum **8**, 1372.

Circuit-level performance



(a)



- X memory crossing at 2.9×10^{-3} (3D toric code: 2.7×10^{-3}); Z memory, $(2,1)$-window decoding: 1.1×10^{-2} (plot in the extra slides)
- Single-shot survives circuit-level noise: the error rate plateaus by window size $w \approx 3$–5 — constant-size windows suffice, no full syndrome history

Depth-13 syndrome circuit, found analytically, unoptimised (full circuit in the extra slides). BP+OSD decoding.

Non-Clifford gates — and a gentle intro to the cup product

Routes to a native non-Clifford gate

Cliffords we have. The hard part is a *native, constant-depth* $\overline{CCZ}$ — what would let us drop magic state distillation. How do qLDPC codes get one?

Colour-code lineage

colour codes → **pin codes**
Vuillot & Breuckmann, arXiv:1906.11394

→ rainbow codes
Scruby, Pesah & Webster, arXiv:2408.13130

Algebraic

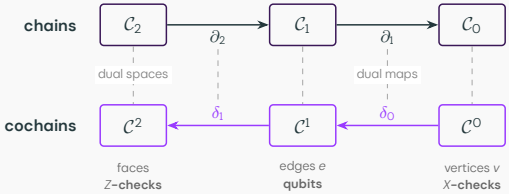
transversal non-Cliffords from **products of algebraic codes**
Golowich & Lin, arXiv:2410.14662; and Lin, via intersection numbers of logical surfaces, arXiv:2410.14631

Cohomological

cup products
Breuckmann, Davydova, Eberhardt & Tantivasadakarn, *Cups & Gates I*, arXiv:2410.16250

We take the cup-product route — based on the same (co)chain-complex language as the codes themselves.

Chains and cochains

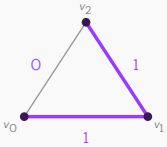


- A **chain** h assigns a bit to every cell — a $\mathbb{Z}_2$ label on each vertex, edge or face. ∂ runs faces $\to$ edges $\to$ vertices
- A **cochain** c is a *function* on chains: give it a chain, it gives back a bit — and its values on the cells pin it down. δ runs the other way, vertices $\to$ edges $\to$ faces
- Two dualities here: each space with its dual, each map with its transpose, $\langle \delta c, h \rangle = \langle c, \partial h \rangle$. Over $\mathbb{Z}_2$ that is completely concrete: δ is just ∂ transposed

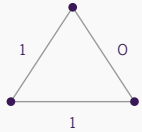
In Dirac language: chains are the *kets*, cochains the *bras*, and $\langle c, h \rangle$ their pairing.

Chain and a cochain example

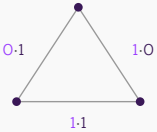
- **1-chain** h: an *element* of C_1, $h = 1 \cdot e_{01} + 1 \cdot e_{12} + 0 \cdot e_{02}$
- **1-cochain** c: a linear map $c : C_1 \to \mathbb{Z}_2$, fixed by its values and extended linearly



1-chain h
a *coefficient* per edge



1-cochain c
a *value* per edge



pairing $\langle c, h \rangle$
$1 + 0 + 0 = 1$

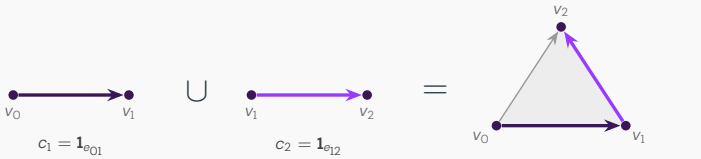
Although over $\mathbb{Z}_2$ chains and cochains are isomorphic, the difference in definition still matters:

functions naturally have a product, sets of coefficients do not.

The cup product

A **product on cochains**, $\cup : \mathcal{C}^j \times \mathcal{C}^k \to \mathcal{C}^{j+k}$ — bilinear and associative. Most intuitively defined for a **simplicial complex**, whose vertices carry an ordering:

$$(c_1 \cup c_2)\big([v_0, \ldots, v_{j+k}]\big) \;=\; c_1\big([v_0, \ldots, v_j]\big)\; c_2\big([v_j, \ldots, v_{j+k}]\big)$$



On a **two-term** complex $\mathcal{C}^0 \to \mathcal{C}^1$ (v a vertex, e an edge) this is just

$v \cup v = v$, $\quad v \cup e = e$ if e leaves v, $\quad e \cup v = e$ if e enters v, $\quad$ every other product is 0.

For **cocycles** c_1, c_2 ($\delta c_1 = 0$) — these are the logical $\overline{X}$ operators:
$(c_1 + \delta b) \cup c_2 \simeq c_1 \cup c_2$, so adding an X-stabiliser changes nothing — the cup product *descends* to cohomology.

The Leibniz rule, and the integral

Leibniz rule. Let's us combine $\cup$ with δ so that cohomology is respected ($\mathbb{Z}_2$ version):

$$\delta(c_1 \cup c_2) \;=\; \delta c_1 \cup c_2 \;+\; c_1 \cup \delta c_2$$

This is why $(c_1 + \delta b) \cup c_2 \simeq c_1 \cup c_2$. ($\delta$ acts like a derivative — Leibniz is its product rule.)

Integral. How to turn a cochain into a *number*: add up its values over every cell of the *highest* dimension — faces in 2D, cubes in 3D.

$$\int \omega \;=\; \sum_{\sigma \,\in\, \text{faces / cubes}} \omega(\sigma) \;\in\; \mathbb{Z}_2$$

- Integrals of coboundaries are zero: $\int \delta\eta = 0$, so $\int \omega$ depends only on the class of ω
- Combined with $\cup$, this gives cohomology invariants $\psi(c_1, \ldots, c_\Lambda) = \int c_1 \cup \cdots \cup c_\Lambda$

Diagonal logical gates from cohomology invariants

- For CSS codes, the cohomology classes let us compactly write logical basis kets:

$$|[\gamma]\rangle \;\propto\; \sum_{c \,\in\, [\gamma]} |c\rangle$$

- A diagonal gate $U|c\rangle = (-1)^{\psi(c)}|c\rangle$ preserves the code space *iff* it acts identically on the terms of each superposition:

$$\psi(\gamma + \delta b) \;=\; \psi(\gamma) \qquad \text{for every } b$$

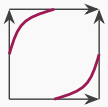
What we want from ψ

1. A **cohomology invariant** — so that U preserves the code space
2. Realised by a **low-depth circuit** of physical gates — so that U is fault tolerant

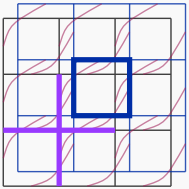
$\psi = \int c_1 \cup \cdots \cup c_\Lambda$ delivers both: an invariant, and — being multilinear — one physical gate per nonvanishing term.

CZ on the torus: the recipe

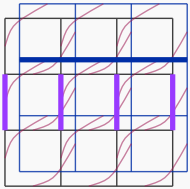
Two copies of the 2D toric code, $\psi(c_1, c_2) = \int c_1 \cup c_2$:



(a) head-to-tail pairs



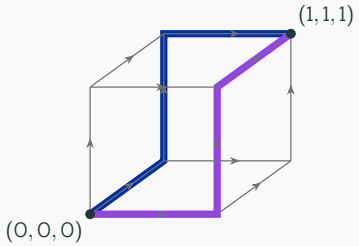
(b) X-check $\mapsto$ X-check $\otimes$ Z-check



(c) $\overline{X} \mapsto \overline{X} \otimes \overline{Z}$

1. Non-zero cup on only **head-to-tail** edges: two pairs per plaquette
2. Put a CZ on every pair where cup product is non-zero
3. $A_v \mapsto A_v \otimes B_p$: an X-check picks up the plaquette above-right — still a stabiliser
4. $\overline{X} \mapsto \overline{X} \otimes \overline{Z}$: a logical $\overline{CZ}$

One dimension up: CCZ in a cube



$e \cup e' \cup e'' \neq 0 \iff$ the three edges form an oriented corner-to-corner path

6 such paths per cube
(yzx and xzy shown)

Three copies of the 3D toric code, $3 \times 3 = 9$ logical qubits. $\int c_1 \cup c_2 \cup c_3$ is nonzero only when the three classes run along distinct directions, so the $3! = 6$ orderings give 6 logical $\overline{CCZ}$s — physical CCZs along the 6 paths of every cube.

Zhu, Sikander, Portnoy, Cross & Brown, arXiv:2310.16982.

Copy-cup gates

Both examples had the same shape. Take Λ copies of a code carrying a Λ-fold cup product, and put one physical $C^{\Lambda-1}Z$ wherever the cup product is nonzero:

$$U_{\cup} = \sum_{c_1,\ldots,c_\Lambda} (-1)^{\int c_1 \cup \cdots \cup c_\Lambda} |c_1,\ldots,c_\Lambda\rangle\langle c_1,\ldots,c_\Lambda|$$

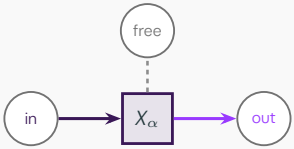
- $\Lambda = 2$: two copies of the 2D toric code, one CZ per plaquette, giving a logical $\overline{CZ}$
- $\Lambda = 3$: three copies of the 3D toric code, CCZs along the paths of every cube, giving a logical $\overline{CCZ}$
- Increasing Λ: $C^{\Lambda-1}Z$ on the Λ-d toric code – Λ-th level of the Clifford hierarchy
- Named copy-cup gates by Breuckmann et al., arXiv:2410.16250.

NB: In this topological setting the **lattice** supplied the ordering the cup product needed.

Cup products beyond topological codes

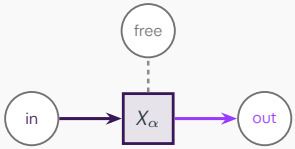
- A general qLDPC code is just a chain complex — nothing supplies an ordering
- **Cups & Gates I**: define $\cup$ on the classical codes behind the product, and replace the ordering by a pre-orientation — a split of each check's qubits, one for *each* of A, B, C:

$A = A_{\text{in}} + A_{\text{out}} + A_{\text{free}}$



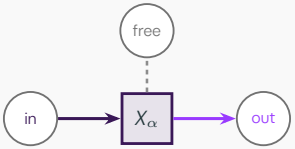
to the **L** qubits

$B = B_{\text{in}} + B_{\text{out}} + B_{\text{free}}$



to the **C** qubits

$C = C_{\text{in}} + C_{\text{out}} + C_{\text{free}}$



to the **R** qubits

Any such split defines a candidate cup product.

Cups & Gates I: Breuckmann, Davydova, Eberhardt & Tantivasadakarn, arXiv:2410.16250.

The cup product on a classical code

The pre-orientation *defines* the cup product — the same three rules as the simplicial case, with the ordering now chosen rather than given:

Simplicial: ordering *given*

$$v \cup v = v$$
$$v \cup e = e \text{ if } e \text{ leaves } v$$
$$e \cup v = e \text{ if } e \text{ enters } v$$

Pre-orientation: ordering *chosen*

$$v \cup v = v$$
$$v \cup e = e \text{ if } e \in \delta_{\text{out}}(v)$$
$$e \cup v = e \text{ if } e \in \delta_{\text{in}}(v)$$

But not every split works. The cup product must satisfy the integrated Leibniz rule:

$$\int \left[\delta c_1 \cup c_2 + c_1 \cup \delta c_2\right] \;=\; \int \delta(c_1 \cup c_2) \;=\; 0$$

For $\Lambda = 2$ that holds exactly when, for all $v_1 \neq v_2 \pmod 2$:

$$|\delta_{\text{in}}(v_1)| + |\delta_{\text{out}}(v_1)| = 0$$

$$|\delta_{\text{in}}(v_1) \cap \delta_{\text{in}}(v_2)| + |\delta_{\text{out}}(v_1) \cap \delta_{\text{out}}(v_2)| + |\delta_{\text{out}}(v_1) \cap \delta_{\text{free}}(v_2)| + |\delta_{\text{free}}(v_1) \cap \delta_{\text{in}}(v_2)| = 0$$

Higher products: the associativity problem

- CCZ gates live at $\Lambda = 3$ - needs a triple cup product: $c_1 \cup c_2 \cup c_3$:
- The simplicial cup product was *associative*, but the one a pre-orientation induces is not: in general $(c_1 \cup c_2) \cup c_3 \neq c_1 \cup (c_2 \cup c_3)$
- But the copy-cup definition carried **no brackets** — $\psi = \int c_1 \cup c_2 \cup c_3$ is ambiguous until we say in which *order* the products are taken

Our choice: evaluate left to right

$$\psi(c_1, c_2, c_3) \;=\; \int \big((c_1 \cup c_2) \cup c_3 \big)$$

The integrated Leibniz rule is then imposed for *this* bracketing.

Next: can any **TT codes** admit such a pre-orientation?

CCZ gates in TT codes

When does a TT code admit a cup product?

A $\overline{CCZ}$ needs the integrated Leibniz rule for the 3-fold product, in our left-to-right bracketing:

$$\int (\delta c_1 \cup c_2) \cup c_3 \;+\; (c_1 \cup \delta c_2) \cup c_3 \;+\; (c_1 \cup c_2) \cup \delta c_3 \;=\; 0$$

Each of A, B, C is one of the classical codes, so δ is multiplication by that polynomial — $\delta(\alpha) = P(\alpha)$ for $P \in \{A, B, C\}$, split by the pre-orientation as $P_{\text{in}}(\alpha) \sqcup P_{\text{out}}(\alpha) \sqcup P_{\text{free}}(\alpha)$. Each polynomial must satisfy, for distinct checks α_i:

$$\begin{aligned}
|P_{\text{in}}(\alpha_1)| + |P_{\text{out}}(\alpha_1)| &= 0 \\
|P_{\text{in}}(\alpha_1) \cap P_{\text{in}}(\alpha_2)| &= 0 \\
|P_{\text{out}}(\alpha_1) \cap P_{\text{out}}(\alpha_2)| + |P_{\text{free}}(\alpha_1) \cap P_{\text{out}}(\alpha_2)| &= 0 \\
|P_{\text{free}}(\alpha_1) \cap P_{\text{in}}(\alpha_2)| &= 0 \\
|P(\alpha_1) \cap P_{\text{in}}(\alpha_2) \cap P_{\text{in}}(\alpha_3)| + |P_{\text{out}}(\alpha_1) \cap P(\alpha_2) \cap P_{\text{in}}(\alpha_3)| &= 0
\end{aligned}$$

$|\cdot|$ counts qubits and $\cap$ takes those the checks share: every overlap must come out even.

The simplest case: weight 2

Take $P = P_1 + P_2$: one term **in**, one **out**, nothing **free**. Distinct checks are shifted apart, so each condition falls out at once:

$\lvert P_{\text{in}}(\alpha_1)\rvert + \lvert P_{\text{out}}(\alpha_1)\rvert = 1+1$	✓ even
$\lvert P_{\text{in}}(\alpha_1) \cap P_{\text{in}}(\alpha_2)\rvert = 0$	✓ $P_1\alpha_1 \neq P_1\alpha_2$
$\lvert P_{\text{out}}(\alpha_1) \cap P_{\text{out}}(\alpha_2)\rvert + \lvert P_{\text{free}}(\alpha_1) \cap P_{\text{out}}(\alpha_2)\rvert = 0$	✓ $P_2\alpha_1 \neq P_2\alpha_2$
$\lvert P_{\text{free}}(\alpha_1) \cap P_{\text{in}}(\alpha_2)\rvert = 0$	nothing free — nothing to check

The triple condition is automatic too: $P_{\text{in}}(\alpha_2)$ and $P_{\text{in}}(\alpha_3)$ are single, *distinct* qubits, so no qubit is shared by three checks.

All three weight 2 gives the (2, 2, 2) codes — named by the weights $(|A|, |B|, |C|)$.

Smallest of them: the 3D toric code, $A = 1+x$, $B = 1+y$, $C = 1+z$.

(2,2,2) codes: beating the 3D toric code

$[\![n, k, d_Z]\!]$	n/n_{3DTC}	d_X	polynomials	logical $\overline{CCZ}$s
$[\![36, 3, 3]\!]$	$4/9$	8	$A{=}1{+}xyz$, $B{=}1{+}x^2z$, $C{=}1{+}x^2y$	6
$[\![48, 3, 4]\!]$	$1/4$	8	$A{=}1{+}x$, $B{=}1{+}xz$, $C{=}1{+}xy$	6
$[\![54, 3, 4]\!]$	$9/32$	9	$A{=}1{+}yz$, $B{=}1{+}xz$, $C{=}1{+}xyz$	6
$[\![60, 3, 4]\!]$	$5/16$	12	$A{=}1{+}xz$, $B{=}1{+}xy$, $C{=}1{+}xyz$	6
$[\![90, 3, 5]\!]$	$6/25$	15	$A{=}1{+}x$, $B{=}1{+}xy$, $C{=}1{+}x^2y^2z$	6

- Up to $4\times$ fewer data qubits than the equal-distance 3D toric code, with more balanced d_X/d_Z
- The catch: $k = 3$ always — these are 3D toric codes on lattices with twisted boundary conditions

Beyond weight 2

We can also admit higher-weight polynomials, provided they carry a specific structure:

Lemma 5

A TT code **admits a cup product** if each of A, B, C is either **weight 2**, or of the form $P = Gf$, where

$$G = \sum_{k=1}^{\mathrm{ord}(g)} g^k = 1 + g + g^2 + \cdots + g^{\mathrm{ord}(g)-1}$$

sums the cyclic group $\langle g \rangle$, with $\mathrm{ord}(g)$ **even** and f any non-zero polynomial.

- Lowest-weight case, $g^2 = 1$: $P = (1+g)(1+\alpha)$ — the $(4, 2, 2)$ codes

(4,2,2) codes: more logical qubits

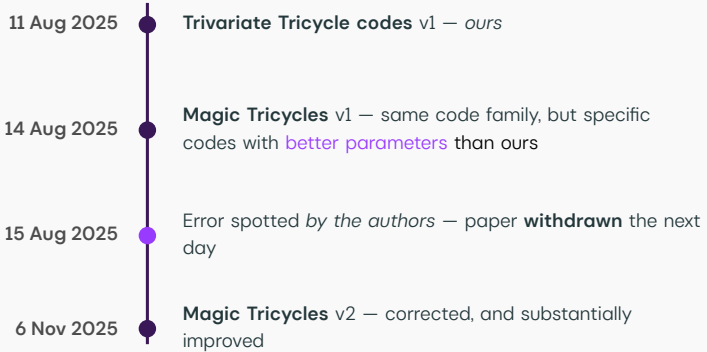
$[\![n, k, d_Z]\!]$	d_X	polynomials	logical $\overline{CCZ}$s
$[\![36, 6, 2]\!]$	3	$A{=}(1{+}z)(1{+}x)$, $B{=}1{+}x$, $C{=}1{+}xyz$	16
$[\![48, 6, 2]\!]$	4	$A{=}(1{+}x^2yz)(1{+}xz)$, $B{=}1{+}x^3$, $C{=}1{+}x^3yz$	10
$[\![54, 9, 2]\!]$	4	$A{=}(1{+}z)(1{+}y)$, $B{=}1{+}xyz$, $C{=}1{+}x^2y^2$	36
$[\![108, 6, 2]\!]$	6	$A{=}(1{+}x^2)(1{+}xz)$, $B{=}1{+}x^2y^2$, $C{=}1{+}x^2y^2z^2$	10
$[\![108, 18, 2]\!]$	4	$A{=}(1{+}x^2)(1{+}xy^2z^2)$, $B{=}1{+}x^2y^2z$, $C{=}1{+}x^2y^2z$	114

- Higher rates, many more logical $\overline{CCZ}$s (more magic per block)
- **But**: Across *every* code we found: either the cup-product $\overline{CCZ}$ has trivial logical action — no encoded CCZ — or the distance is 2. Every row above is of the second kind; in the first, the distance can exceed 2
- The same holds for the $(4, 4, 2)$, $(4, 4, 4)$ and $(6, 2, 2)$ families.

At first this puzzled us. **Tiew & Breuckmann** then proved it had to happen: a *non-associative* cup product forces weight-4 codes to distance 2. arXiv:2602.23307

Related work

Magic Tricycles: a timeline



Menon, Bonilla Ataides, Manovitz, Bluvstein, Lukin et al., arXiv:2508.10714.

What v2 changed: a different associativity rule

Recall our choice: bracket left-to-right, always. Menon et al. choose case by case — a symmetric triple cup-product:

$$
\begin{aligned}
e \cup v \cup v &= (e \cup v) \cup v && \text{bracketing chosen} \\
v \cup v \cup e &= v \cup (v \cup e) && \text{bracketing chosen} \\
v \cup e \cup v &= (v \cup e) \cup v = v \cup (e \cup v) && \text{associative anyway} \\
v \cup v \cup v &= (v \cup v) \cup v = v \cup (v \cup v) && \text{associative anyway}
\end{aligned}
$$

anything else $= 0$; v a vertex, e an edge, as before

- Reversing the order swaps the two chosen cases — they *mirror* each other, which is what makes it symmetric
- Choosing case by case escapes the distance-2 wall that our fixed bracketing *provably* ran into

arXiv:2508.10714v2, Def. 6 (Eqs. 40–44), in the v / e notation used earlier.

Magic Tricycles: the payoff

N	K	D_X	D_Z	Type	CCZ depth	CCZ method
48	6	8	4	$4-2-2$	8	STCP
84	6	12	5	$4-2-2$	8	STCP
108	6	12	6	$4-2-2$	8	STCP
240	6	≤ 22	8	$4-2-2$	8	STCP
480	6	≤ 36	10	$4-2-2$	8	STCP
108	12	11	4	$4-4-2$	16	STCP
180	12	15	6	$4-4-2$	32	STCP
108	15	12	6	$4-4-4$	96	NLR
270	24	15	8	$4-4-4$	96	NLR
324	12	≤ 32	≤ 12	$4-4-4$	128	STCP
480	15	≤ 48	≤ 14	$4-4-4$	128	STCP

Table 1, arXiv:2508.10714v2

- Real distances at last: $[\![48, 6, 4]\!]$, $[\![108, 15, 6]\!]$, $[\![270, 24, 8]\!]$ — no distance-2 wall
- **Single-shot magic state distillation**: 6×10^{-10} logical error at 30% acceptance on $[\![48, 6, 4]\!]$
- $\approx 0.5\%$ circuit-level threshold, and a neutral-atom implementation protocol

STCP = symmetric triple cup-product; NLR = numerical Leibniz rule.

Where we stand

Summary of Trivariate Tricycle codes

Trivariate Tricycle codes: BB-style polynomial codes on a 4-term chain complex.

- **Memory**: thresholds competitive with the 3D toric code at up to 48× fewer qubits; $[\![432, 12, 12]\!]$ reaches LER $\leq 10^{-9}$/round at $p = 10^{-4}$
- **Single-shot**: meta-check distance $= d_Z$; constant decoding windows suffice, even at circuit level — *point (ii) from the intro: each operation gets shorter*
- **Cliffords**: shift automorphisms + three families of transversal $\overline{CZ}$s
- **Non-Clifford**: cup-product $\overline{CCZ}$s in constant depth for (2,2,2) and (4,2,2) codes
- **Magic Tricycles**: the symmetric cup product lifts the distance-2 wall — high-distance codes in the same family, still with a native $\overline{CCZ}$

Open questions

- Is there a Bravyi–König-style bound for this family of codes? (probably)
- Can the periodic boundary conditions be removed — similar to planar BB codes?
- Can we find codes with high rate *and* high distance?
- If we could, what would the details of the architecture look like?

Can we reach the high distances needed for large-scale quantum computation?

Planar BB codes: Liang, Eberhardt & Chen, arXiv:2504.08887.

The gap, concretely

code	source	$[\![n, k, d_Z]\!]$	native *CCZ*?
bb_{18} (BB, processor)	Cain et al.	$[\![248, 10, \leq 18]\!]$	✗
$\mathrm{lp}_{20}^{3,7}$ (lifted product, memory)	Cain et al.	$[\![4350, 1224, \leq 20]\!]$	✗
Magic Tricycle	Menon et al.	$[\![480, 6, 10]\!]^*$ ($d_X \leq 36$)	✓
Magic Tricycle	Menon et al.	$[\![480, 15, \leq 14]\!]^\dagger$ ($d_X \leq 48$)	✓
TT (2, 2, 2)	our work	$[\![90, 3, 5]\!]$ ($d_X = 15$)	✓

* highest *exact* distance in Menon et al.; † highest *bounded* distance.

Open question

Does a code with Cain-et-al.-grade parameters *and* a native unitary fault-tolerant *CCZ* exist? If yes: no factory — and the 120 years collapse.

Higher-distance codes almost certainly exist — but how do we find them?

New codes from old

Where the good codes hide

Sequences of Bivariate Bicycle Codes from Covering Graphs

Benjamin C. B. Symons*[1,3], Abhishek Rajput†[2], and Dan E. Browne[3]

Ben Symons

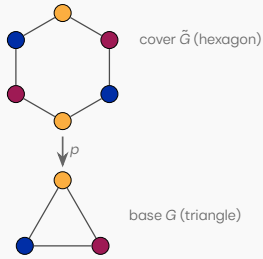
Abhishek Rajput

Using a tool from graph theory — the covering graph — we can build new codes from old ones.

Including bb_{18}, the code at the heart of the Cain et al. 10,000-qubit Shor factoring estimate we opened with.

Symons, Rajput & Browne, arXiv:2511.13560.

Covering graphs: codes that unroll



- A graph $\tilde{G}$ **covers** G if $p : \tilde{G} \to G$ is a bijection on every neighbourhood
- For BB Tanner graphs this is pure algebra — $\tilde{Q}$ is an h-cover of Q iff

$$\tilde{\ell} = u\ell,\ \tilde{m} = tm\ (h = ut),\ \ \tilde{A}_i \equiv A_i,\ \tilde{B}_i \equiv B_i \pmod{x^{\ell}-1,\ y^{m}-1}$$

- Search space collapses from $(n_h/2)^4$ to h^4: for the gross code as a double cover of $[\![72, 12, 6]\!]$, that is 16 candidates instead of 27 million

Two different 2-covers of $[\![18, 8, 2]\!]$ $(A = 1+y+y^2,\ \ \ell = m = 3)$

$\tilde{A} = x^3 + y + y^2$ and $\tilde{A} = x^3 + x^3y + y^2$ (both $[\![36, 8, 4]\!]$, $\tilde{\ell} = 6, \tilde{m} = 3$)

Different polynomials, but $x^3 \equiv 1$ and $x^3y \equiv y \pmod{x^3-1}$ — the covering map leaves *room to choose*.

Symons, Rajput & Browne, arXiv:2511.13560, Examples 2 and 3.

Cover sequences: distance grows, structure travels

- k does not shrink: $k_h \geq k$ — now proved for both odd and even h
- Distance climbs with it: $d \leq d_h \leq hd$ when $k_h = k$ — so far proved for odd h only
- Logical operators *project* and *lift* along the cover; automorphisms lift with identical logical action (h odd, $k_h = k$) — gates ride the sequence for free

Sequence codes first appeared as generalised toric codes: Liang, Liu, Song & Chen, arXiv:2503.03827. Cover-sequence view: Symons, Rajput & Browne, arXiv:2511.13560.

Work in progress: lifting a TT code

Covering graphs gave us **sequences of BB codes**. The same trick works for TT codes — there is one more variable, but the same structure:

$$\ell,\ m,\ p \quad \longrightarrow \quad h_x\ell,\ \ h_y m,\ \ h_z p$$

Each monomial then has $h = h_x h_y h_z$ lifts to choose from:

$$x^i y^j z^k \quad \longmapsto \quad x^{i+\ell\mu_x}\ y^{j+m\mu_y}\ z^{k+p\mu_z}$$

For instance: take Menon et al. $[\![48,6,4]\!]$: $\ell, m, p = 2, 2, 4$, with $A = y+z+xz+xyz^2$. Tripling the z direction ($h_z = 3$) gives $\ell, m, p = 2, 2, 12$, with several choices of lift:

$$\text{e.g. } y+z+xz+xyz^2, \qquad y+z+xz+xyz^6, \qquad y+z+xz+xyz^{10}$$

The cover is again a TT code of the same type.

But does the gate come with it?

- Yes — sometimes. The lift can break the integrated Leibniz rule, but under certain conditions the structure survives (the details depend on which associativity rule you choose)
- But the distance need not follow. Lifting in a single direction can leave the code stuck at the distance it started with
- Lifting in two directions at once looks better — the distance upper bounds seem to grow.

Where we are

Encouraging early results — but so far these are *upper bounds* on distance, not exact, and the full search has yet to be done.

Summary and Outlook

- **The gate**: a native $\overline{CCZ}$ in TT codes, built from cup products — and *which* codes admit one depends on a choice of associativity
- **The path forward**: covering graphs shrink the search space enormously; whether it contains good codes with a $\overline{CCZ}$ is work in progress
- **The target**: one code with high rate, high distance, single-shot error correction *and* native magic — then the factory disappears, and so do the 120 years

Abraham Jacob

Campbell McLauchlan

Ben Symons

Abhishek Rajput

arXiv:2508.08191
arXiv:2511.13560

Thanks to my **co-authors** and to our funders **EPSRC**.

Slides in Beamer Metropolis style built with Claude Code.