REALIZATION CALCULUS

Quantum and Circuits

Arithmetic, Symmetry, and the Cost of Realization

OPERATORS / GATES / OBSTRUCTIONS

Komorebi P.

Preface

A mathematical representation can expose a structure that is expensive to see in another set of coordinates. It can also hide the cost of reaching those coordinates, discarding a residual, or reconstructing an answer. This volume studies both sides of that passage.

The first part follows an arithmetic selector through Fourier synthesis, rational projection, and a normalized complex state. The second asks when symmetry makes a quantum evolution accessible to classical computation. The third compares a restricted circuit with an ambient realization that may leave the restriction temporarily and return. The fourth studies restored reversible circuits, safe discrepancy covers, projection costs, and the limits of multiplicative summaries.

These treatments share the language of realizations, transfers, residuals, and obstructions. Their models remain distinct. A finite Fourier identity is not a gate synthesis algorithm. A small invariant sector is useful for simulation only when its representation, updates, and measurements are accessible. A catalytic saving depends on the chosen gate alphabet and cost. A lower bound for a faithful linear summary has the hypotheses of that summary model.

Each part gives its own definitions and proof conditions. Formal and computational companions follow the mathematical treatments. A shared glossary and collected references close the volume.

How to read the volume

For a concrete entry point, begin with the twelve-dimensional arithmetic example and its visual study. The symmetry treatment develops invariant packets, moving frames, and compressed representations. The catalytic treatment begins with nested realization models before giving the complete linear core and its quantum and reversible extensions. The circuit treatment supplies restoration conventions before deriving its cover and projection results.

A proved statement has a written argument under explicit hypotheses. An exact finite computation concerns the declared bounded state space. A formal result identifies a specified Lean declaration and its proof record. A research proposal or conjecture remains open where its decisive obligation is not established. Panel titles state these distinctions directly.

The mathematical notation is local to each part. In particular, circuit obstructions use the binary field by default, while linear gate theory also works over general fields and rings. Formal coverage applies to the selected declarations, rather than to every written theorem in the volume.

Contents

Preface i

How to read the volume ii

I Arithmetic realizations 1

1 Quantum–Arithmetic Bridge 2

2 Arithmetic in Fourier space 22

II Symmetry collapse in quantum computation 47

3 Foundations of packet simulation 48

4 Exact symmetry families 63

5 States, measurements, and open dynamics 84

6 The complexity boundary 89

7 Algorithms and open directions 95

III Catalytic gate set distortion 103

8 General theory of temporary escape 104

9 The complete linear core 123

10 Quantum and reversible gate families 167

11 Packets, types, holonomy, and restricted realization 196

12 The computational atlas and open directions 203

IV Circuit obstructions 218

13 Scope, notation, and computational model 219

14 Exact cut comparison and safe discrepancies 222

15 Complete safe-packet degree 223

16 Calibration families and exact small oracles 227

17 Homogeneous projection: cost and safety 230

18 Faithful linear gluing algebras 233

19 Weighted witness fingerprints 235

20 The overlap obstruction to composition 237

21 Circuit examples in view 238

V Formal and computational companions 263

A The formal arithmetic bridge 264

B Symmetry: supplementary proofs and computation 285

C Gate matrices, census data, and open obligations 290

D The formal circuit model 301

E Claim coverage and theorem index 330

F Verification and proof boundaries 333

Glossary 334

Sources and references 335

Sources and background references 338

PART I

Arithmetic realizations

An arithmetic selector is followed through Fourier synthesis, exact rational projection, and a normalized complex state. The twelve-dimensional example makes each change of representation explicit.

IN THIS PART

Quantum–Arithmetic Bridge 2

Arithmetic in Fourier space 22

CHAPTER 1

Quantum–Arithmetic Bridge

Overview

A finite arithmetic selector can be transported into a rational signal projector and a complex projective measurement by one explicitly normalized Fourier transformation. This chapter gives the full proof chain and follows a single exact twelve-dimensional example through every representation. The arithmetic input is the coprime residue set modulo twelve. Its Fourier synthesis is the integer-valued Ramanujan signal. The associated circulant matrix is an orthogonal projector of rank four. A normalized version of the signal is a quantum state, and an inverse Fourier transformation recovers the arithmetic support. An unsigned state has the same probabilities in the original basis yet zero probability on the selected support after transformation. The two states are orthogonal. A continuous relative-phase family makes that difference visible without changing the initial probability histogram.

The general finite Fourier, Möbius, divisor-projector, and polyphase identities are proved in the text. The formal companion covers the concrete modulus-twelve realization, including the sound interpretation of exact rational coordinates in the ordinary complex numbers, both unitarity identities, projector transport, state preparation and recovery, and the phase-sensitive probability conclusions. The formal companion contains ninety-six named theorems across five modules, including the twenty-two original bridge statements. The formal appendix identifies its exact coverage and verification evidence.

Ramanujan sums, Fourier orthogonality, rational cyclic projectors, and Ramanujan signal subspaces are classical. The contribution of this presentation is an exact, inspectable realization chain, its theorem-to-code concordance, and its explicit account of the information retained or discarded at each map. It makes no circuit-efficiency or quantum-advantage claim.

1.1 The mathematical object being transported

The word *bridge* is useful only when its arrows have definitions. In the present example the common object is a selector: a diagonal operator that retains certain Fourier coordinates and discards the others. Arithmetic supplies the selected labels. Fourier synthesis expresses that selector in sample coordinates. Complex normalization then supplies a unitary transformation and the state vectors used in a quantum interpretation.

The basic identity is

$$P = FDF^{\dagger}.$$

Here D is a diagonal zero-or-one matrix, F is unitary, and $\dagger$ denotes conjugate transpose. Consequently P is an orthogonal projector. The transformation F is reversible. The selection P need not be. A complete decomposition consists of both Px and $(I - P)x$, whose sum recovers x.

This distinction determines what can be transported. A Fourier basis change preserves dimension, inner products, norms, and the entire vector. A projector preserves only its selected component. Measurement probabilities discard still more information. The same exact operator can appear in arithmetic, signal processing, and finite-dimensional quantum theory, while those uses retain different observables and have different implementation costs.

We work over $\mathbb{C}$ for Fourier transformations and use $\mathbb{Q}$ for the rational kernels. All index sets in the main finite construction are specified, and no limit argument is needed. A packet in this chapter means a component together with its stated realization map. This definition does not turn every kind of component into the same object. Character-order packets and coefficient-residue lanes will be distinguished explicitly in the polyphase section.

1.1.1 Claim labels and evidence classes

The labels QA01 through QA20 refer to mathematical statements in this paper. They are not a count of Lean declarations. A single claim can use several formal lemmas, and a general theorem can have a formally checked specialization rather than a generic Lean proof.

We use three evidence descriptions. A *general prose theorem* has a proof below for all inputs satisfying its hypotheses. A *formal specialization* is an actual declaration in the preserved modulus-twelve source. A *finite computational check* is a bounded verification of an implementation. These descriptions can support the same identity, but they are not interchangeable. The Lean guide gives the exact declaration mapping.

The central example uses the vectors

$$\begin{aligned} c &= (4,0,2,0,-2,0,-4,0,-2,0,2,0),\\ b &= (5,2,-1,2,5,2,-1,2,5,2,-1,2),\\ x &= (9,2,1,2,3,2,-5,2,3,2,1,2). \end{aligned}$$

These are the literal coefficients of the formal source, and $x = c + b$. Keeping one example fixed makes every arrow independently checkable.

1.2 Fourier conventions and the finite proof chain

Let q be a positive integer. Write $I_q = \{0, 1, \dots, q-1\}$, interpreting indices modulo q, and put

$$\zeta_q = e^{2\pi i/q}.$$

A signal is a vector $x : I_q \to \mathbb{C}$. We write Id_q for the identity matrix of dimension q, and I for an identity matrix when its dimension is clear. Our inner product is conjugate-linear in the first argument:

$$\langle x, y\rangle = \sum_{n=0}^{q-1} \overline{x(n)}\, y(n), \qquad \|x\|^2 = \langle x, x\rangle.$$

The unscaled negative-sign discrete Fourier transform is

$$\widehat{x}(k) = \sum_{n=0}^{q-1} x(n)\zeta_q^{-kn}.$$

The positive-sign normalized Fourier matrix is

$$(F_q)_{na} = q^{-1/2}\zeta_q^{na}.$$

Thus F_q synthesizes a signal from normalized frequency coordinates, while $F_q^\dagger x = \widehat{x}/\sqrt{q}$ analyzes it. This convention explains all signs and all square-root factors below. The negative-sign transform and the positive-sign synthesis are adjoints of one another.

Theorem QA01. The order of a finite phase

For $a \in I_q$, the multiplicative order of ζ_q^a is

$$\frac{q}{\gcd(a,q)}.$$

Proof. The equation $(\zeta_q^a)^t = 1$ holds exactly when q divides at. Write $g = \gcd(a,q)$, $q = gq'$, and $a = ga'$, with $\gcd(a',q') = 1$. The divisibility condition is $q' \mid a't$, which is equivalent to $q' \mid t$. Its smallest positive solution is $t = q'$. This includes $a = 0$: then $g = q$ and the order is one. □

Theorem QA02. Character orthogonality

For any integer h,

$$\sum_{n=0}^{q-1} \zeta_q^{hn} = \begin{cases} q, & q \mid h, \\ 0, & q \nmid h. \end{cases}$$

Proof. If $q \mid h$, every summand is one. Otherwise $r = \zeta_q^h \neq 1$ and $r^q = 1$. Multiplying the sum by $1 - r$ gives $1 - r^q = 0$, so the sum is zero. This argument also covers $q = 1$, where only the first case occurs. □

Theorem QA03. Unitarity, inversion, and Parseval

For every positive q,

$$F_q^\dagger F_q = \mathrm{Id}_q, \qquad F_q F_q^\dagger = \mathrm{Id}_q\,.$$

Consequently,

$$x(n) = \frac{1}{q}\sum_{k=0}^{q-1} \widehat{x}(k)\zeta_q^{kn}, \qquad \sum_n |x(n)|^2 = \frac{1}{q}\sum_k |\widehat{x}(k)|^2.$$

Proof. The (a,b) entry of the first matrix product is

$$\frac{1}{q}\sum_{n=0}^{q-1} \zeta_q^{n(b-a)}.$$

By QA02 it is one when $a = b$ and zero otherwise. The (n,m) entry of the second product is the same orthogonality sum with indices interchanged. Since $F_q^\dagger x = \widehat{x}/\sqrt{q}$, the identity $F_q F_q^\dagger x = x$ is the displayed inverse formula. Finally, a unitary matrix preserves inner products:

$$\|F_q^\dagger x\|^2 = x^\dagger F_q F_q^\dagger x = x^\dagger x.$$

Replacing $F_q^\dagger x$ by $\widehat{x}/\sqrt{q}$ proves Parseval. □

These proofs use finite sums, ordinary complex multiplication, and the positive real square root of q. No asymptotic approximation enters. At $q = 1$ the Fourier matrix is the one-by-one identity, and all identities have their literal scalar meanings.

1.3 Coprimality becomes an integer signal

Let

$$U_q = \{a \in I_q : \gcd(a, q) = 1\}, \qquad \varphi(q) = |U_q|.$$

The Ramanujan sum is

$$c_q(n) = \sum_{a \in U_q} \zeta_q^{an}, \qquad n \in \mathbb{Z}.$$

It adds the primitive qth-root phases selected by QA01. Its definition is complex, but its value is an integer. That integrality is the step that makes the sample-coordinate projector rational. Ramanujan sums and their divisor formulas are classical arithmetic constructions [1].

Theorem QA04. Möbius formula and integrality

Let μ be the Möbius function, with $\mu(1) = 1$. For every positive q and every integer n,

$$\boxed{c_q(n) = \sum_{d \mid \gcd(q,n)} d\,\mu(q/d) \in \mathbb{Z}.}$$

In particular c_q has period q, is real-valued, satisfies $c_q(-n) = c_q(n)$, and has $c_q(0) = \varphi(q)$.

Proof. For a positive integer g,

$$\sum_{d \mid g} \mu(d) = \begin{cases} 1, & g = 1, \\ 0, & g > 1. \end{cases}$$

To see this, only squarefree divisors contribute. If g has r distinct prime factors, the sum is $(1-1)^r$, with value one when $r = 0$. Therefore

$$\mathbf{1}_{\gcd(a,q)=1} = \sum_{d \mid \gcd(a,q)} \mu(d).$$

Substitute this into the phase sum over all $a \in I_q$, exchange the finite sums, and write $a = db$ in the terms where $d \mid a$:

$$c_q(n) = \sum_{d|q} \mu(d) \sum_{b=0}^{q/d-1} e^{2\pi i bn/(q/d)}$$
$$= \sum_{\substack{d|q \\ (q/d)|n}} \mu(d)\frac{q}{d}.$$

The second equality is QA02. Replacing q/d by e gives the displayed divisor sum. It is an integer. Periodicity follows already from the phase definition. The divisor formula is unchanged by $n \mapsto -n$, so $c_q(-n) = c_q(n)$. At $n = 0$, every selected phase is one, giving $c_q(0) = |U_q|$. □

At $q = 1$, $U_1 = \{0\}$ because $\gcd(0, 1) = 1$. The phase sum and the divisor formula both give $c_1(n) = 1$. This boundary case is included in the proofs even though the inherited arithmetic regression range begins at modulus two.

Theorem QA05. Exact spectral support and energy

Over one period,

$$\boxed{\widehat{c_q}(k) = q\,\mathbf{1}_{k\in U_q}, \qquad k \in I_q.}$$

Moreover,

$$\sum_{n=0}^{q-1} |c_q(n)|^2 = q\varphi(q).$$

Proof. Substitute the phase definition and exchange two finite sums:

$$\widehat{c_q}(k) = \sum_{a\in U_q} \sum_{n=0}^{q-1} \zeta_q^{(a-k)n}.$$

QA02 makes the inner sum q exactly when $a = k$, and zero otherwise. The energy identity follows from QA03:

$$\|c_q\|^2 = \frac{1}{q}\sum_k |\widehat{c_q}(k)|^2 = \frac{1}{q}\varphi(q)q^2.$$

□

The selector has been recovered by an identity. A numerical spectrum can illustrate the result, but neither a tolerance nor a peak-finding rule defines the arithmetic support.

1.4 Rational character-order projectors

It is helpful to distinguish the size of an ambient signal from the order of a character. Fix a positive ambient length N. For each divisor d of N, define

$$S_d = \{a \in I_N : N/\gcd(a, N) = d\}.$$

The set S_d contains the labels of characters of exact order d. Put $D_d = \mathrm{diag}(\mathbf{1}_{a\in S_d})$ and

$$P_d^{(N)} = F_N D_d F_N^\dagger.$$

The superscript records the ambient dimension. In particular, the kernel normalization below is $1/N$, even when $d < N$.

Theorem QA06. Rational kernels and complete divisor decomposition

For every $d \mid N$,

$$\boxed{(P_d^{(N)})_{nm} = \frac{1}{N} c_d(n-m).}$$

The matrices have rational entries and satisfy

$$\begin{aligned}
(P_d^{(N)})^\dagger &= P_d^{(N)},\\
P_d^{(N)} P_e^{(N)} &= \delta_{de} P_d^{(N)},\\
\sum_{d\mid N} P_d^{(N)} &= \mathrm{Id}_N,\\
\operatorname{rank} P_d^{(N)} &= \operatorname{tr} P_d^{(N)} = \varphi(d).
\end{aligned}$$

Proof. A label in S_d has the unique form $a = (N/d)b$, where $0 \le b < d$ and $\gcd(b,d) = 1$. Indeed,

$$\gcd((N/d)b, N) = (N/d)\gcd(b,d).$$

Direct multiplication therefore gives

$$\begin{aligned}
(F_N D_d F_N^\dagger)_{nm} &= \frac{1}{N}\sum_{a\in S_d} \zeta_N^{a(n-m)}\\
&= \frac{1}{N}\sum_{b\in U_d} \zeta_d^{b(n-m)} = \frac{1}{N} c_d(n-m).
\end{aligned}$$

QA04 proves rationality. Each D_d is a self-adjoint diagonal projector. The sets S_d partition I_N, because every character has exactly one order dividing N. Hence $D_d D_e = \delta_{de} D_d$ and $\sum_{d\mid N} D_d = \mathrm{Id}_N$. Conjugating these identities by the unitary F_N proves the projector identities. The rank is unchanged under multiplication by invertible matrices, so it equals the number of selected diagonal entries, $|S_d| = \varphi(d)$. The diagonal kernel entries are all $c_d(0)/N = \varphi(d)/N$, whose sum is $\varphi(d)$. □

This orthogonal divisor-shell decomposition has established antecedents in Ramanujan subspaces and filter banks [2, 3, 4]. The general theorem above is proved independently in this text. The formal companion certifies the concrete $N = d = 12$ projector and its transport identity, rather than an arbitrary-divisor Lean theorem.

1.4.1 A shift-operator realization and its sign

Let A be the cyclic shift $(Ax)(n) = x(n-1)$. A Fourier column $f_a(n) = N^{-1/2}\zeta_N^{na}$ satisfies $Af_a = \zeta_N^{-a} f_a$. Therefore

$$E_a = \frac{1}{N}\sum_{\ell=0}^{N-1} \zeta_N^{a\ell} A^\ell$$

acts as the identity on f_a and as zero on every other column, by QA02. Summing E_a over S_d gives

$$P_d^{(N)} = \frac{1}{N}\sum_{\ell=0}^{N-1} c_d(\ell) A^\ell.$$

This formula also follows entrywise because $(A^\ell)_{nm}$ is one exactly when $n - m = \ell$ modulo N. A group-algebra convention with the opposite shift sign writes individual idempotents with the opposite phase sign. The divisor shells are unchanged under $a \mapsto -a$, and $c_d(-\ell) = c_d(\ell)$, so the rational shell formula remains the same.

Proposition 21.11 of *Stratified Universality for Resolved Representations*, gives the rational cyclic order-projector formula in the group algebra. Boundary 21.12 distinguishes it from formal-series coefficient-residue selection [5]. The present shift calculation makes the representation and the sign explicit.

1.4.2 Unions of order classes

If $\mathcal{D}$ is any set of divisors of N, then

$$P_{\mathcal{D}} = \sum_{d\in\mathcal{D}} P_d^{(N)}$$

is a rational orthogonal projector of rank $\sum_{d\in\mathcal{D}} \varphi(d)$. Its complement is the sum over the remaining divisors. Individual split Fourier projectors can have nonrational entries. Summing complete order classes is a sufficient mechanism for rationality. No converse classification of every rational selector is needed for the present bridge.

1.5 Selection, residuals, and information loss

For any set $S \subseteq I_N$, let D_S be its diagonal indicator and $P_S = F_N D_S F_N^\dagger$. This construction does not require S to be an order class. Rationality is special, but orthogonal projection is general.

Theorem QA07. Projector geometry and exact residual reconstruction

For every $x \in \mathbb{C}^N$,

$$\begin{aligned} x &= P_S x + (I - P_S)x, \\ \langle P_S x, (I - P_S)x\rangle &= 0, \\ \|x\|^2 &= \|P_S x\|^2 + \|(I - P_S)x\|^2. \end{aligned}$$

Furthermore, P_S is injective exactly when $S = I_N$. If S is proper, its kernel has dimension $N - |S|$.

Proof. The first equality is distributivity. Since $P_S^\dagger = P_S$ and $P_S^2 = P_S$,

$$\langle P_S x, (I - P_S)x\rangle = x^\dagger P_S (I - P_S)x = 0.$$

Expanding the norm square of the orthogonal sum proves the energy identity. In Fourier coordinates, P_S is D_S, whose kernel consists exactly of the vectors supported outside S. Its dimension is $N - |S|$. This kernel is zero exactly when every label is selected. □

This gives a precise reconstruction contract. The pair of packets $(P_S x, (I-P_S)x)$ retains the complete vector, subject to the compatibility condition that the first lies in the range and the second in the kernel. Keeping only the first packet is an exact projection but an incomplete representation of an arbitrary input.

For character-order packets, QA06 gives the stronger complete decomposition

$$x = \sum_{d|N} P_d^{(N)} x, \qquad \|x\|^2 = \sum_{d|N} \|P_d^{(N)} x\|^2.$$

The decomposition is a change in organization rather than a reduction in the dimension of the whole signal. Smaller stored selectors do not make an arbitrary signal contain less information. Compression requires a restriction on the admissible signals or an explicit approximation criterion.

1.6 Coefficient-residue lanes and polyphase transport

There is a second exact decomposition that is common in Komorebi calculations and in signal processing. It separates *positions* by residue class rather than separating *Fourier characters* by order. Both decompositions use projectors, but they select different objects.

1.6.1 Formal coefficient lanes

Let $L \geq 1$ and let $f(z) = \sum_{n\geq 0} a_n z^n$ be a formal power series over a field containing a primitive Lth root ω and $1/L$. For $0 \leq r < L$, define

$$E_r f(t) = \sum_{m\geq 0} a_{r+Lm} t^m, \qquad R_r f(z) = z^r E_r f(z^L).$$

The first expression addresses the lane in a new variable. The second places that lane back at its original coefficient positions.

Theorem QA08. Exact coefficient-residue decomposition

The coefficient-residue maps satisfy

$$\begin{aligned} R_r f(z) &= \frac{1}{L} \sum_{j=0}^{L-1} \omega^{-rj} f(\omega^j z), \\ R_r R_s &= \delta_{rs} R_r, \\ \sum_{r=0}^{L-1} R_r f &= f. \end{aligned}$$

Proof. The coefficient of z^n in the first right-hand side is

$$\frac{a_n}{L}\sum_{j=0}^{L-1}\omega^{j(n-r)}.$$

Character orthogonality makes it a_n when $n \equiv r \pmod{L}$ and zero otherwise. This is precisely the coefficient of $R_r f$. Each coefficient position belongs to a unique lane, so the projector and completeness identities follow coefficientwise. The equality is formal, with no analytic convergence hypothesis. □

For an infinite analytic series, the same identity holds wherever all displayed evaluations are defined, but the formal theorem does not need that interpretation. The lane index is an exponent residue. It is not a multiplicative-order class of a Fourier character.

1.6.2 Finite polyphase lanes

Now suppose $N = LM$ with positive integers L and M. For a vector $x \in \mathbb{C}^N$, define its L lanes by

$$x_r(m) = x(r + Lm), \qquad 0 \le r < L, \quad 0 \le m < M.$$

The addressing map $T : x \mapsto (x_0, \dots, x_{L-1})$ is bijective. Interleaving the lanes reconstructs x, and

$$\sum_{n=0}^{N-1}|x(n)|^2 = \sum_{r=0}^{L-1}\sum_{m=0}^{M-1}|x_r(m)|^2.$$

This energy identity is an exact reindexing. It is useful to preserve it separately from the spectral aliasing identity that follows.

Theorem QA09. Polyphase Fourier identities and decimation aliasing

Write $\widehat{x_r}$ for the unscaled M-point transform of lane r. Then for $k \in I_N$,

$$\boxed{\widehat{x}(k) = \sum_{r=0}^{L-1}\zeta_N^{-kr}\widehat{x_r}(k \bmod M).}$$

Conversely, for $\ell \in I_M$,

$$\boxed{\widehat{x_r}(\ell) = \frac{1}{L}\sum_{j=0}^{L-1}\zeta_N^{r(\ell+jM)}\widehat{x}(\ell + jM).}$$

Proof. Partition the N samples by their unique representation $n = r + Lm$:

$$\begin{aligned}\widehat{x}(k) &= \sum_{r=0}^{L-1}\sum_{m=0}^{M-1}x_r(m)\zeta_N^{-k(r+Lm)}\\ &= \sum_{r}\zeta_N^{-kr}\sum_{m}x_r(m)\zeta_M^{-km}.\end{aligned}$$

The inner sum is $\widehat{x_r}(k \bmod M)$, proving the first identity. For the second, insert Fourier inversion into the lane transform:

$$\widehat{x_r}(\ell) = \sum_{m=0}^{M-1} x(r+Lm)\zeta_M^{-\ell m}$$
$$= \frac{1}{N}\sum_{k=0}^{N-1} \widehat{x}(k)\zeta_N^{kr} \sum_{m=0}^{M-1} \zeta_M^{(k-\ell)m}.$$

QA02 makes the inner sum M exactly when $k = \ell + jM$ for some $0 \le j < L$. Since $M/N = 1/L$, this is the second displayed identity. □

Every lane transform combines an alias class of L full-spectrum coefficients. One decimated lane generally cannot distinguish those coefficients. All L offsets together retain the original signal. For a fixed alias class ℓ, the equations across r are an invertible Fourier mixing system after the nonzero phases $\zeta_N^{r\ell}$ are removed. The apparent collapse is resolved by preserving the complete collection of offsets.

1.6.3 Why the two projectors must not be conflated

A coefficient-residue selector keeps samples such as $x(0), x(3), x(6), x(9)$. A character-order projector keeps spectral modes such as $1, 5, 7, 11$. The former is diagonal in sample coordinates; the latter is diagonal in frequency coordinates. Under a Fourier basis change, a sample-residue mask generally produces a different operator, with its own alias structure.

A valid comparison therefore specifies the conjugating matrix and the chosen selector. The fact that both constructions use roots of unity does not identify their packets. QA08 and QA09 supply an explicit rotation and Fourier calculation where a bridge is needed. Neither statement is claimed as a generic theorem in the inherited Lean package used here.

1.7 The exact modulus-twelve arithmetic calculation

We now fix $q = 12$. The coprime labels are

$$U_{12} = \{1, 5, 7, 11\}, \qquad \varphi(12) = 4, \qquad \zeta = e^{\pi i/6} = \frac{\sqrt{3}}{2} + \frac{i}{2}.$$

The signal is

$$c(n) = \zeta^n + \zeta^{5n} + \zeta^{7n} + \zeta^{11n} = 2\cos(\pi n/6) + 2\cos(5\pi n/6).$$

The cosine formula comes from conjugate pairs. It exposes interference, while the integer divisor formula provides an independent arithmetic route to each value.

Theorem QA10. The exact phase and transform tables

Over one period,

$$c = (4, 0, 2, 0, -2, 0, -4, 0, -2, 0, 2, 0),$$

and

$$\widehat{c} = (0, 12, 0, 0, 0, 12, 0, 12, 0, 0, 0, 12).$$

Proof. For $q = 12$, QA04 becomes

$$c(n) = 2\mathbf{1}_{2|n} - 4\mathbf{1}_{4|n} - 6\mathbf{1}_{6|n} + 12\mathbf{1}_{12|n}.$$

Indeed, the only divisors d of twelve for which $\mu(12/d) \neq 0$ are $2, 4, 6, 12$, with coefficients $2, -4, -6, 12$. Evaluating the divisibility indicators at $0, \dots, 11$ gives the sample table. QA05 gives value twelve at the four coprime labels and zero elsewhere. □

At sample zero, all four phases agree and the value is four. At sample one, they cancel. At sample six, all four are minus one and the value is minus four. Those signs are the relative phase information that will later distinguish two states with the same initial probabilities.

1.7.1 A compact projector matrix

Put $P = P_{12}^{(12)}$. Its entries are

$$P_{nm} = \frac{c(n-m)}{12}.$$

Equivalently, its first row is

$$\left(\frac{1}{3}, 0, \frac{1}{6}, 0, -\frac{1}{6}, 0, -\frac{1}{3}, 0, -\frac{1}{6}, 0, \frac{1}{6}, 0\right),$$

and each subsequent row is the cyclic right shift of the preceding row. This specifies every entry of the twelve-by-twelve rational matrix. It agrees exactly with the literal matrix in the original formal module and with the indexed matrix in the later transport module.

With $D = \operatorname{diag}(\mathbf{1}_{a \in U_{12}})$ and $F = F_{12}$,

$$P = FDF^{\dagger}, \qquad P^2 = P, \qquad P^{\dagger} = P, \qquad \operatorname{rank} P = \operatorname{tr} P = 4.$$

The six divisor shells of length twelve have orders $1, 2, 3, 4, 6, 12$ and ranks $1, 1, 2, 2, 2, 4$. Their ranks add to twelve, and their orthogonal sum is the identity by QA06.

1.7.2 The finite lanes of the same signal

For $L = 3$ and $M = 4$, the coefficient-residue lanes are

$$c_0 = (4, 0, -4, 0),$$
$$c_1 = (0, -2, 0, 2),$$
$$c_2 = (2, 0, -2, 0).$$

Their four-point Fourier transforms are

$$\widehat{c}_0 = (0, 8, 0, 8), \quad \widehat{c}_1 = (0, 4i, 0, -4i), \quad \widehat{c}_2 = (0, 4, 0, 4).$$

For example, a full coefficient at $k = 1$ is reconstructed as

$$\widehat{c}(1) = 8 + 4i\zeta^{-1} + 4\zeta^{-2} = 12.$$

At $k = 9$, the same lane-frequency index 1 gives

$$\widehat{c}(9) = 8 + 4i\zeta^{-9} + 4\zeta^{-18} = 8 - 4 - 4 = 0.$$

The offsets distinguish two full Fourier labels with the same reduced frequency. These equations demonstrate the role of the phases in QA09. They do not identify a lane with a character-order shell.

1.8 Filtering a signal with something to remove

Take the off-band perturbation

$$b(n) = 3\cos(\pi n/2) + 2.$$

The constant occupies frequency zero, and the cosine occupies labels three and nine. These labels all lie outside U_{12}. The perturbation has the literal rational vector

$$b = (5, 2, -1, 2, 5, 2, -1, 2, 5, 2, -1, 2).$$

Set $x = c + b$, giving

$$x = (9, 2, 1, 2, 3, 2, -5, 2, 3, 2, 1, 2).$$

Theorem QA11. Exact filter recovery and orthogonal energy

The test signal satisfies

$$Pc = c, \qquad Pb = 0, \qquad Px = c, \qquad (I - P)x = b,$$

and

$$\langle c, b\rangle = 0, \qquad \|c\|^2 = 48, \qquad \|b\|^2 = 102, \qquad \|x\|^2 = 150 = 48 + 102.$$

Proof. QA05 gives the support of $\widehat{c}$. Write the cosine as

$$3\cos(\pi n/2) = \frac{3}{2}(\zeta^{3n} + \zeta^{9n}).$$

Thus $\widehat{b}$ equals twenty-four at zero, eighteen at three and nine, and zero elsewhere. Multiplication by D fixes the spectrum of c and kills the spectrum of b, proving the filter identities. Their supports are disjoint, so Parseval for inner products gives $\langle c, b\rangle = 0$. QA05 gives $\|c\|^2 = 12 \cdot 4 = 48$, and

$$\|b\|^2 = \frac{24^2 + 18^2 + 18^2}{12} = 102.$$

QA07 then gives $\|x\|^2 = 48 + 102 = 150$. The same numbers follow by direct rational squaring of the displayed vectors. □

The discarded component is part of a complete explanation of the input. It is called a perturbation because the example declares c to be the target. It is a perfectly exact signal component rather than an error introduced by the representation. If the task were to recover all of x, deleting b would lose valid information.

1.9 Normalized states and literal unitary transport

In a twelve-dimensional complex Hilbert space, let $|0\rangle, \ldots, |11\rangle$ be an orthonormal basis. Define

$$|u\rangle = \frac{|1\rangle + |5\rangle + |7\rangle + |11\rangle}{2}.$$

Its squared norm is $4(1/2)^2 = 1$. Define the amplitude states

$$|\psi\rangle = \frac{1}{\sqrt{48}} \sum_{n=0}^{11} c(n)|n\rangle, \qquad |\chi\rangle = \frac{1}{\sqrt{48}} \sum_{n=0}^{11} |c(n)|\,|n\rangle.$$

Both have norm one because $\sum_n c(n)^2 = 48$. The second state is a separately specified state, constructed from the absolute-value vector. No universal quantum operation implementing the nonlinear rule $z \mapsto |z|$ is being asserted.

Theorem QA12. Exact preparation and recovery

With the positive-sign Fourier matrix defined in Section 1.2,

$$\boxed{F|u\rangle = |\psi\rangle, \qquad F^\dagger|\psi\rangle = |u\rangle.}$$

Proof. At coordinate n, the first left-hand side is

$$\frac{1}{2\sqrt{12}} \sum_{a \in U_{12}} \zeta^{na} = \frac{c(n)}{\sqrt{48}}.$$

This is the defining coordinate of ψ. Applying $F^\dagger$ and using $F^\dagger F = I$ proves the second equality. □

The normalization relation is $\sqrt{48} = 2\sqrt{12}$. In the exact coordinate implementation the Fourier scale is $\sqrt{3}/6 = 1/\sqrt{12}$, and the state scale is $\sqrt{3}/12 = 1/\sqrt{48}$. The later formal source proves those scale identities and the exponential entry formula, rather than leaving them as informal interpretations of a rational tuple.

This is a mathematical quantum-state construction. An actual realization would additionally require state preparation, a twelve-dimensional unitary implementation, coherence control, and sampling. A twelve-dimensional Fourier matrix is not the ordinary sixteen-dimensional Fourier transformation on four qubits. The unitary direct sum $F_{12} \oplus I_4$ supplies a mathematical sixteen-dimensional embedding, but it supplies no gate synthesis or resource bound by itself.

1.10 Relative phase is absent from one histogram and present in another

For a normalized state $|v\rangle = \sum_n v(n)|n\rangle$, the standard-basis probabilities are $|v(n)|^2$. In the basis analyzed by $F^\dagger$, the probabilities are $|(F^\dagger v)(k)|^2$.

These describe two measurement procedures on separately prepared copies. Measuring the first histogram on a particular copy does not preserve that copy as a coherent input for the second procedure.

Theorem QA13. Equal initial probabilities and orthogonal states

The states ψ and χ have identical probabilities in the original basis and are orthogonal:

$$|\psi(n)|^2 = |\chi(n)|^2 \quad \text{for all } n, \qquad \langle \psi, \chi \rangle = 0.$$

Proof. Squaring a real number or its absolute value gives the same result, proving the first claim. For orthogonality,

$$\langle \psi, \chi \rangle = \frac{1}{48} \sum_{n=0}^{11} c(n)|c(n)|.$$

The positive entries contribute $16+4+4 = 24$, and the negative entries contribute $-4-16-4 = -24$. Their sum is zero. □

This is a strong information boundary. The states are distinct even after the physically irrelevant overall phase is disregarded. Orthogonal normalized states cannot be scalar multiples of one another. Yet their original-basis probability histograms are identical.

Theorem QA14. Complete arithmetic phase separation

After applying $F^\dagger$,

$$\Pr_{F^\dagger \psi}(k \in U_{12}) = 1, \qquad \Pr_{F^\dagger \chi}(k \in U_{12}) = 0.$$

More explicitly,

$$(F^\dagger \psi)(k) = \begin{cases} 1/2, & k \in \{1, 5, 7, 11\}, \\ 0, & \text{otherwise}, \end{cases}$$

whereas

$$(F^\dagger \chi)(k) = \begin{cases} 2/3, & k \in \{0, 6\}, \\ 1/6, & k \in \{2, 4, 8, 10\}, \\ 0, & \text{otherwise}. \end{cases}$$

Proof. QA12 gives the first amplitude table. For the second, set $a(n) = |c(n)|$. Its vector is

$$a = (4, 0, 2, 0, 2, 0, 4, 0, 2, 0, 2, 0).$$

It has period six. Pairing n and $n + 6$ in its twelve-point transform produces a factor $1 + (-1)^k$, so every odd transform coefficient vanishes. For $k = 2\ell$, its transform is twice the six-point transform of $(4, 0, 2, 0, 2, 0)$. The doubled transform is sixteen when ℓ is divisible by three and four otherwise, as one sees from

$$2\left(4+2e^{-2\pi i\ell/3}+2e^{-4\pi i\ell/3}\right).$$

When $\ell \equiv 0 \pmod 3$, the bracket is eight. Otherwise the two cubic phases sum to minus one and the bracket is two. Thus

$$\widehat{a} = (16, 0, 4, 0, 4, 0, 16, 0, 4, 0, 4, 0).$$

State normalization and the unitary inverse transform divide this table by $\sqrt{48}\sqrt{12} = 24$, producing the second amplitude table. Its support is even, while every label in U_{12} is odd. The selected probability is therefore zero. The signed state has four probabilities of $1/4$, with total one. □

The full probability comparison is:

Label	Initial probability, both states	After $F^\dagger$, signed state	After $F^\dagger$, unsigned state
0	$1/3$	0	$4/9$
1	0	$1/4$	0
2	$1/12$	0	$1/36$
3	0	0	0
4	$1/12$	0	$1/36$
5	0	$1/4$	0
6	$1/3$	0	$4/9$
7	0	$1/4$	0
8	$1/12$	0	$1/36$
9	0	0	0
10	$1/12$	0	$1/36$
11	0	$1/4$	0

Every column sums to one. The Fourier probability supports are disjoint. Negative amplitudes in the original arithmetic signal therefore encode observable relative phase, even though their signs do not appear in the first probability column.

Theorem QA15. No inverse from the initial histogram alone

There is no function from original-basis probability histograms to state vectors that reconstructs both ψ and χ. There is also no reconstruction of both states up to overall phase from that histogram alone.

Proof. The two histograms are equal by QA13. A function assigns the same output to equal inputs, so it cannot output the two distinct vectors. For the phase-insensitive formulation, QA13 shows that the states are orthogonal and normalized. If $\chi = \lambda\psi$ for any complex scalar of modulus one, then $\langle\psi, \chi\rangle = \lambda \neq 0$, a contradiction. □

The first formulation is formalized in the later publication module. The strengthening to equivalence up to global phase is a prose corollary proved here. It is not counted as an additional inherited Lean declaration.

1.11 The selector as a projective measurement

The pair $\{P, I - P\}$ is a projective measurement because its two members are self-adjoint, idempotent, orthogonal, and sum to the identity. For the conditional update we choose the ideal Lüders instrument,

whose unnormalized branches send a density matrix ρ to $P\rho P$ and $(I-P)\rho(I-P)$. The projector effects determine the outcome probabilities; this chosen instrument determines the state update. The probability rule and this conditional update can be stated independently of the arithmetic example.

Theorem QA16. Selection probability and conditional state

Let P be an orthogonal projector on $\mathbb{C}^N$ and let v be a normalized state. For the ideal Lüders instrument just specified, the two probabilities are

$$p = \|Pv\|^2 = \langle v, Pv\rangle, \qquad 1-p = \|(I-P)v\|^2.$$

When $p > 0$, the normalized selected state is $Pv/\sqrt{p}$. When $1-p > 0$, the normalized complementary state is $(I-P)v/\sqrt{1-p}$. The probabilities belong to $[0,1]$ and sum to one.

Proof. Since $P^\dagger P = P$, the quadratic form equals $\|Pv\|^2$ and is nonnegative. QA07 applied to a unit vector gives the probability sum and the upper bound. Division by the positive square root produces a state of squared norm one in each nonzero branch. □

For the explicit normalized test signal $v = x/\sqrt{150}$, QA11 gives

$$p = \frac{48}{150} = \frac{8}{25}, \qquad 1-p = \frac{102}{150} = \frac{17}{25}.$$

The selected conditional state is $c/\sqrt{48} = \psi$, and the complementary conditional state is $b/\sqrt{102}$. An exact classical filter applied to the full vector returns c. A projective measurement on the normalized input instead returns a branch with these probabilities. These are compatible uses of the same operator with different observation rules.

The vector χ makes noninjectivity concrete: $P\chi = 0$ although $\|\chi\| = 1$. A reversible Fourier change of basis cannot be substituted for this loss of information. Conversely, retaining both complementary packets reconstructs every vector, as QA07 and the publication module explicitly state.

The inherited formal source certifies the concrete projectors, states, and probability identities. It does not provide a general quantum measurement library or a generic theorem implementing the conditional state-update rule. QA16 is a general prose theorem, with the usual finite-dimensional measurement interpretation.

1.12 Exact arithmetic coordinates and their complex meaning

The twelfth root contains $\sqrt{3}$ and i. Its relevant exact arithmetic can be represented without floating-point trigonometric evaluation. Let a tuple

$$(a,b,c,d) \in \mathbb{Q}^4$$

denote

$$\iota(a,b,c,d) = a + b\sqrt{3} + i(c + d\sqrt{3}).$$

The coordinate root is $(0, 1/2, 1/2, 0)$, whose interpretation is $\sqrt{3}/2 + i/2$. Coordinate conjugation sends (a,b,c,d) to $(a,b,-c,-d)$. Addition is entrywise, and multiplication is

$$(a,b,c,d)(A,B,C,D) = \big(aA + 3bB - cC - 3dD,\\ aB + bA - cD - dC,\\ aC + 3bD + cA + 3dB,\\ aD + bC + cB + dA\big).$$

Theorem QA17. Sound interpretation of the exact operations

The map ι preserves zero, one, rational scalars, addition, multiplication, conjugation, natural powers, and finite sums. Therefore an equality obtained with these exact coordinate operations yields the corresponding equality in $\mathbb{C}$.

Proof. The claims for zero, one, rational scalars, addition, and conjugation follow directly from the definition. For multiplication, write the two interpreted values as $(a + b\sqrt{3}) + i(c + d\sqrt{3})$ and $(A + B\sqrt{3}) + i(C + D\sqrt{3})$. Their product has real part

$$(a + b\sqrt{3})(A + B\sqrt{3}) - (c + d\sqrt{3})(C + D\sqrt{3})$$

and imaginary part

$$(a + b\sqrt{3})(C + D\sqrt{3}) + (c + d\sqrt{3})(A + B\sqrt{3}).$$

Expanding and using $(\sqrt{3})^2 = 3$ gives exactly the displayed tuple product. Power preservation follows by induction from multiplication and one. Finite-sum preservation follows by induction from addition and zero. Applying a function to equal coordinate tuples gives equal complex numbers. □

Only sound transport of equalities is needed here. The formal source does not install the tuple type as a global field instance, and it does not need an injectivity theorem for ι. It proves preservation of the actual operations and explicitly connects the scales and root to their conventional complex formulas.

The matrix interpretation is entrywise. Since multiplication is a finite sum of products and the adjoint uses conjugation, QA17 also transports coordinate matrix multiplication and adjoints. The same argument transports matrix action on vectors and squared norms. This is what closes the gap between an exact coordinate certificate and an identity about an ordinary complex unitary matrix.

1.13 The central finite bridge theorem

The arithmetic, signal, and state statements can now be gathered without leaving any undefined arrow.

Theorem QA18. Arithmetic, rational signal, and unitary realization

Let F, D, P, c, b, x, u, ψ, and χ have the explicit modulus-twelve definitions above. Then:

1. $F^\dagger F = FF^\dagger = I$.
2. $P = FDF^\dagger$, $P^2 = P$, and $P^\dagger = P$.
3. $Px = c$ and $(I - P)x = b$, with $\|x\|^2 = 48 + 102 = 150$.
4. $Fu = \psi$ and $F^\dagger\psi = u$.
5. u, ψ, and χ are normalized, and ψ is orthogonal to χ.
6. ψ and χ have identical original-basis probabilities.
7. Their inverse-Fourier coprime probabilities are respectively one and zero.

8. $P\psi = \psi$, $P\chi = 0$, and P is not injective.

Proof. QA03 gives the unitarity identities. QA06 gives projector transport and the two projector identities. QA11 gives the filter and energy statements. QA12 gives preparation and recovery; the displayed normalizations and QA13 give statement five and statement six. QA14 gives statement seven. In Fourier coordinates ψ is supported on U_{12} while χ is supported outside it, so $P\psi = \psi$ and $P\chi = 0$. The normalized nonzero vector χ is therefore in the kernel, proving noninjectivity. □

The later formal publication theorem gathers the principal complex transport and probability identities into a single conjunction. Its supporting modules also supply the exact vector coefficients, filter recovery, residual reconstruction, and noninjectivity theorem. The general prose route above explains why the identities hold; the finite formal source certifies this literal example under its exact definitions.

One mathematical object has thus appeared in three realizations. Arithmetic selects coprime character labels. The signal realization is a rational circulant projector. The complex state realization uses a unitary basis change and a projective measurement. The complete vector information is preserved by the basis change, while selection and observation have separately stated losses.

1.14 A continuous phase family with a fixed initial histogram

The two endpoint states are especially simple, but the mechanism is continuous. Define the nonnegative vectors v_+ and v_- by placing $|c(n)|/\sqrt{48}$ respectively on the positive and negative entries of c, with zero elsewhere. Thus

$$\chi = v_+ + v_-, \qquad \psi = v_+ - v_-.$$

Their supports are disjoint, and each has squared norm $1/2$. For real θ, define

$$\chi_\theta = v_+ + e^{i\theta} v_-.$$

It is a normalized state with the same original-basis probabilities for every θ. The endpoints are $\chi_0 = \chi$ and $\chi_\pi = \psi$.

Theorem QA19. Relative-phase response

For this family,

$$\boxed{\Pr_{F^\dagger \chi_\theta}(k \in U_{12}) = \|P\chi_\theta\|^2 = \sin^2(\theta/2).}$$

Proof. Disjoint support gives $\|\chi_\theta\|^2 = \|v_+\|^2 + \|v_-\|^2 = 1$ and unchanged entrywise squared magnitudes. Solving for v_+ and v_- in terms of χ and ψ gives

$$\chi_\theta = \frac{1 + e^{i\theta}}{2}\chi + \frac{1 - e^{i\theta}}{2}\psi.$$

Since $P\chi = 0$ and $P\psi = \psi$,

$$P\chi_\theta = \frac{1 - e^{i\theta}}{2}\psi.$$

Its squared norm is

$$\frac{|1-e^{i\theta}|^2}{4}=\frac{1-\cos\theta}{2}=\sin^2(\theta/2).$$

Finally, $P = FDF^\dagger$ and unitarity imply $\|P\chi_\theta\|^2 = \|DF^\dagger\chi_\theta\|^2$, exactly the sum of the selected Fourier-basis probabilities. □

This is a proved prose extension for the visual guide. It is not presented as an inherited Lean theorem. It shows that a probability histogram can stay fixed while an interference-based observation varies from zero to one.

1.15 Phase replacement and dephasing are different operations

The unsigned endpoint is a coherent state with specified positive amplitudes. It should not be confused with a state obtained by forgetting all relative phases. The latter is represented by a mixed state, not generally by one state vector.

For a normalized vector v, its density matrix is $vv^\dagger$. Complete dephasing in the sample basis keeps only its diagonal entries. In the present example, dephasing either ψ or χ gives the same diagonal matrix

$$\rho_{\text{diag}}=\operatorname{diag}\left(c(n)^2/48\right)_{n=0}^{11}.$$

Its trace is one, and every diagonal entry is nonnegative.

Theorem QA20. The dephased selected probability

The selected projective-measurement probability of this diagonal state is

$$\operatorname{tr}(P\rho_{\text{diag}})=\frac{1}{3}.$$

Proof. Every diagonal entry of P is $c(0)/12 = 4/12 = 1/3$. Multiplication by a diagonal matrix and taking the trace gives

$$\operatorname{tr}(P\rho_{\text{diag}})=\sum_n P_{nn}\frac{c(n)^2}{48}=\frac{1}{3}\sum_n\frac{c(n)^2}{48}=\frac{1}{3}.$$

□

The three outcomes are therefore distinct: the signed coherent state has selected probability one, the unsigned coherent state has probability zero, and the completely dephased state has probability one third. This calculation prevents an incorrect identification of amplitude sign removal with physical decoherence. QA20 is a general finite-dimensional calculation specialized to the example and proved here. Density matrices and dephasing are outside the inherited Lean package.

1.16 What is complete and what remains a separate question

The finite worked bridge is complete as a mathematical result. Its maps are defined, its identities are proved, its literal coefficients agree across the rational and cyclotomic representations, and its principal

modulus-twelve transport and phase claims have formal certificates. The current release provides the traditional paper, visual guide, formal concordance, source, and reproducibility records.

The general prose theorems also establish Fourier unitarity, the Möbius divisor formula, rational divisor-projector completeness, and polyphase aliasing for the declared finite or formal models. Their arbitrary-modulus formulations are not claimed as generic Lean declarations in this package. The continuous phase and dephased-state extensions are proved prose results rather than inherited formal additions. The visual guide illustrates the continuous phase family.

Several broader questions have different proof obligations. Efficient preparation of coprime-support states needs an algorithm and a cost model. Efficient circuit synthesis of the twelve-dimensional transformation or its qubit embedding needs a gate construction and resource bounds. Incomplete-sample recovery needs an observation map that is injective on the declared model, or a stability theorem with stated conditioning. Noisy implementation needs a perturbation analysis. A physical claim needs an experimental protocol and measured data.

These questions can be asked using the same operator language. Their existence does not weaken the exact bridge proved here, and the bridge does not answer them automatically. The present result shows, in one fully specified example, how arithmetic selection becomes rational filtering and phase-sensitive unitary realization while keeping reconstruction, information loss, and measurement distinct.

CHAPTER 2

Arithmetic in Fourier space

One arithmetic selector, three realizations

Four residues modulo twelve will become an integer signal, an exact rational filter, and a normalized quantum state. Every bridge will be written down.

The guide follows one object through three coordinate systems. The starting condition is coprimality. The signal realization adds its selected waves. The quantum realization normalizes the amplitude vector and distinguishes coherent transformation from measurement.

$$U_{12} = \{1, 5, 7, 11\}, \qquad c_{12}(n) = \sum_{a \in U_{12}} e^{2\pi i a n/12}.$$

We will build a projector that keeps exactly those modes, apply it to a signal with a complementary component, and account for all retained and discarded energy. Then we will compare two orthogonal states with identical initial measurement probabilities.

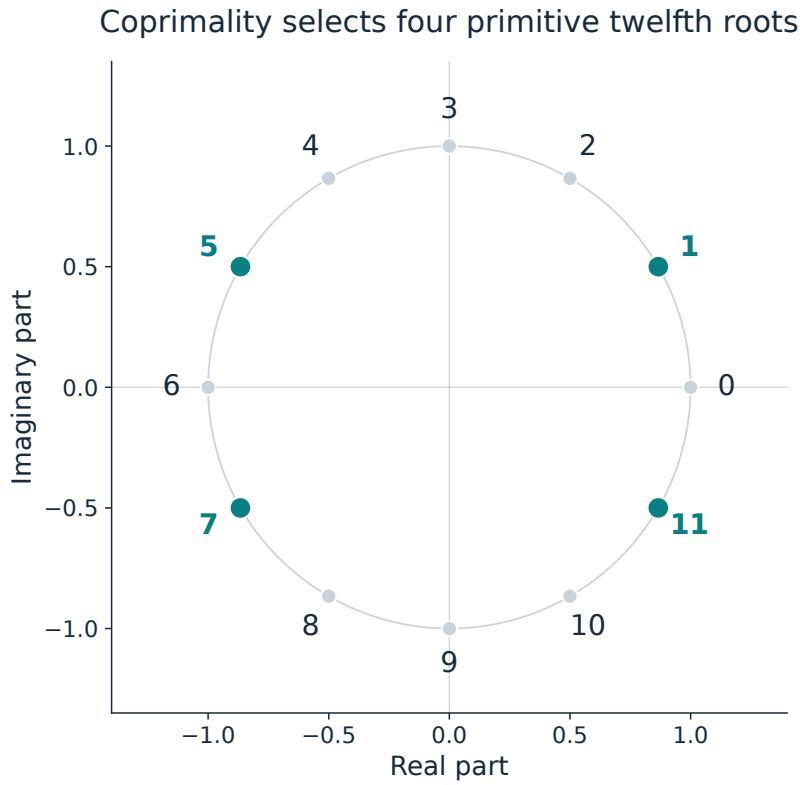
Arithmetic	Signal	Quantum model
Coprime labels	Fourier support	Frequency-basis amplitudes
Rational cyclic projector	Exact convolution filter	Projective measurement
Cyclotomic phase sum	Integer samples	Signed coherent amplitudes

> These are exact finite mathematical constructions. The quantum interpretation uses the usual state-vector and Born-rule model. No hardware experiment, circuit-cost claim, or quantum speedup is implied.

Coprimality selects primitive roots

A selected residue is a label for a wave whose phase has exact order twelve.

Set $\zeta = e^{2\pi i/12}$. The phase ζ^a first returns to 1 after $12/\gcd(a, 12)$ powers. Thus the coprime labels $1, 5, 7, 11$ select exactly the primitive twelfth roots.



The circle is the complex unit circle. Teal points are the four primitive roots; pale points are the other eight roots. Labels are residues a, not measured outcomes.

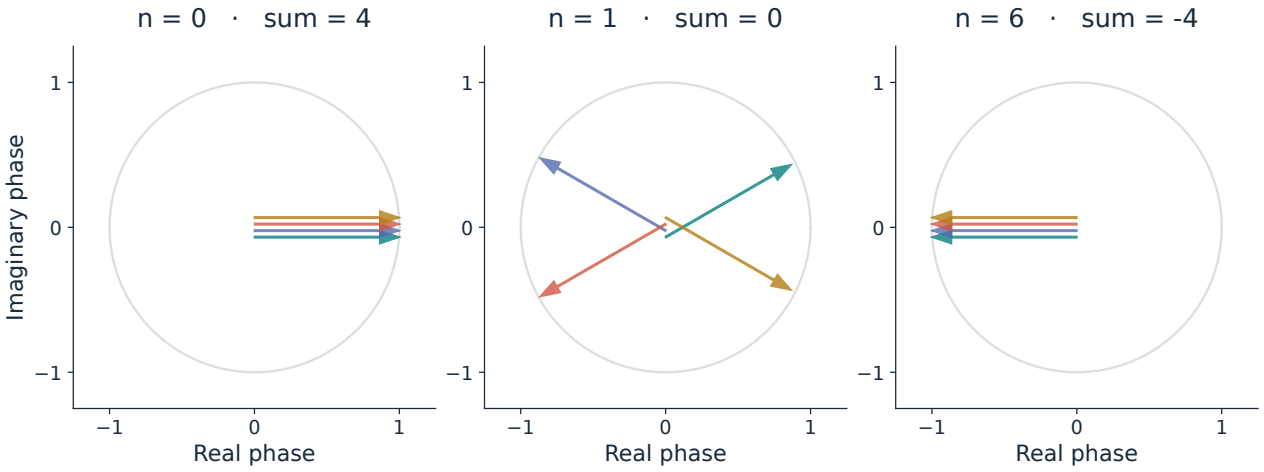
$$\gcd(a, 12) = 1 \quad \Longleftrightarrow \quad \operatorname{ord}(\zeta^a) = 12.$$

Arithmetic has supplied a precise selector. Nothing probabilistic has happened at this stage.

The selected phases add and cancel

The same four waves can reinforce, cancel, or reinforce with the opposite sign.

At each sample n, add the four numbers ζ^{an} with $a \in U_{12}$. Each has magnitude one. Their sum need not have magnitude one.



At $n = 0$ the sum is 4. At $n = 1$ the four phases cancel. At $n = 6$ the sum is -4. Tiny parallel drawing offsets expose coincident arrows at $n = 0$ and $n = 6$; each arrow's displacement is the actual complex phase.

$$c_{12}(0) = 4, \qquad c_{12}(1) = 0, \qquad c_{12}(6) = -4.$$

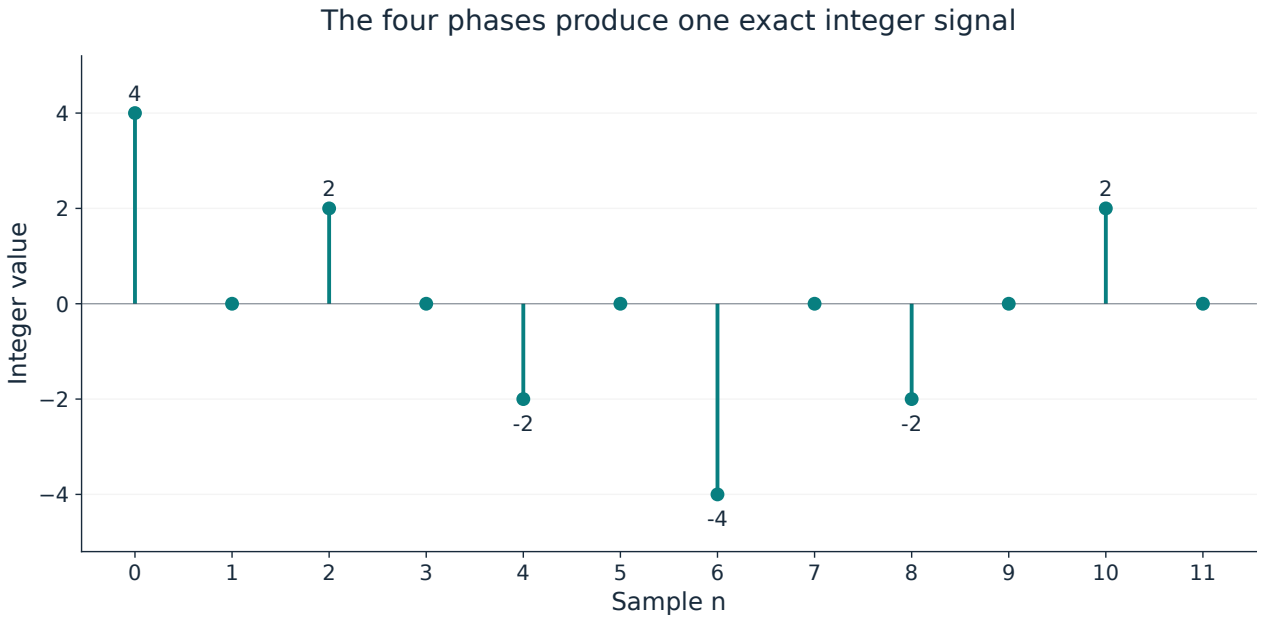
Negative samples encode relative phase. They are part of the information that subsequent transforms use.

Interference yields an integer signal

The samples are exact integers, even though the individual waves have complex amplitudes.

Pairing conjugates gives

$$c_{12}(n) = 2\cos(\pi n/6) + 2\cos(5\pi n/6).$$



One complete period of the Ramanujan signal. The dots are discrete samples, not samples from a continuous probability distribution.

$$(c_{12}(0), \ldots, c_{12}(11)) = (4, 0, 2, 0, -2, 0, -4, 0, -2, 0, 2, 0).$$

The antisymmetry $c_{12}(n+6) = -c_{12}(n)$ will later expose why erasing the signs changes the recovered arithmetic.

Why the phase sum is an integer

A divisor identity certifies integrality without rounding trigonometric values.

For $q \geq 1$, insert the Möbius coprimality identity into the finite phase sum:

$$\mathbf{1}_{\gcd(a,q)=1} = \sum_{d \mid \gcd(a,q)} \mu(d).$$

Write $a = db$ after interchanging the finite sums. Character orthogonality evaluates the inner geometric series:

$$\begin{aligned} c_q(n) &= \sum_{d \mid q} \mu(d) \sum_{b=0}^{q/d-1} e^{2\pi i b n/(q/d)} \\ &= \sum_{\substack{d \mid q \\ q/d \mid n}} \mu(d) \frac{q}{d} = \sum_{e \mid \gcd(q,n)} e\mu(q/e) \in \mathbb{Z}. \end{aligned}$$

Sample	Relevant divisor contribution	Exact result
$n = 0$	$2\mu(6) + 4\mu(3) + 6\mu(2) + 12\mu(1)$	4
$n = 2$	$\mu(12) + 2\mu(6)$	2
$n = 4$	$\mu(12) + 2\mu(6) + 4\mu(3)$	-2
$n = 6$	$\mu(12) + 2\mu(6) + 3\mu(4) + 6\mu(2)$	-4

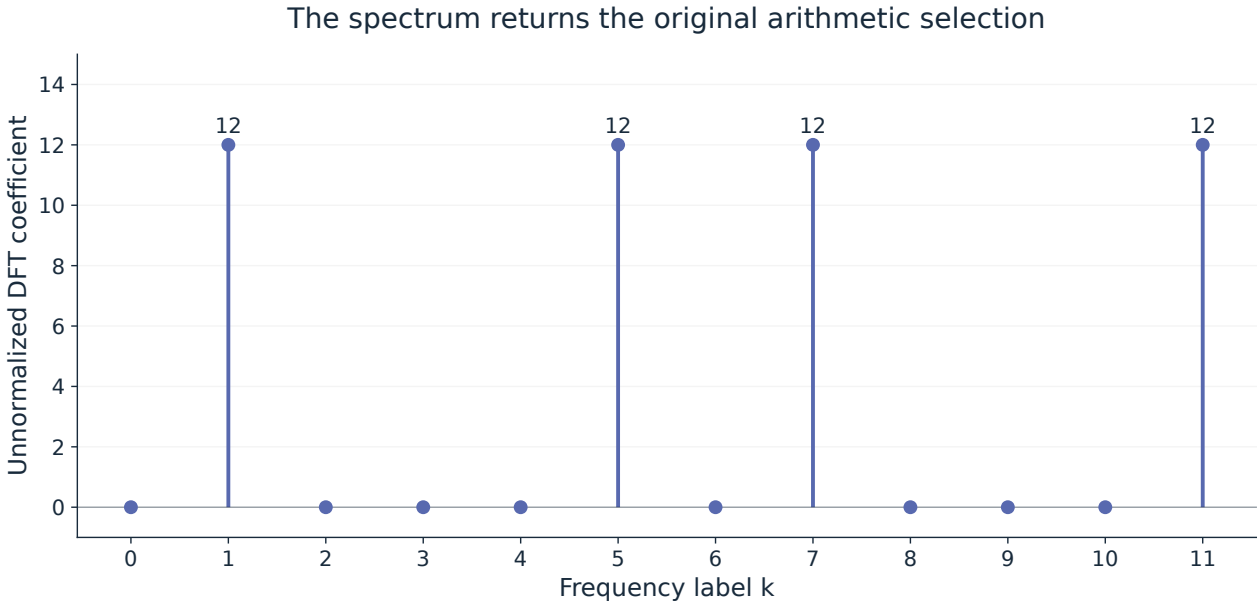
The finite verifier also evaluates the twelve phase sums in the exact ring $\mathbb{Z}[z]/(z^4 - z^2 + 1)$. The plots use floating coordinates for rendering, while the sample identities use exact arithmetic.

The spectrum recovers the selector

The Fourier transform returns exactly the four labels selected by coprimality.

Use the negative-sign, unnormalized discrete Fourier transform

$$\widehat{x}(k) = \sum_{n=0}^{11} x(n) e^{-2\pi i k n/12}.$$



The coefficient is exactly 12 at $k = 1, 5, 7, 11$ and exactly zero elsewhere. This is an algebraic identity, not a spectral thresholding decision.

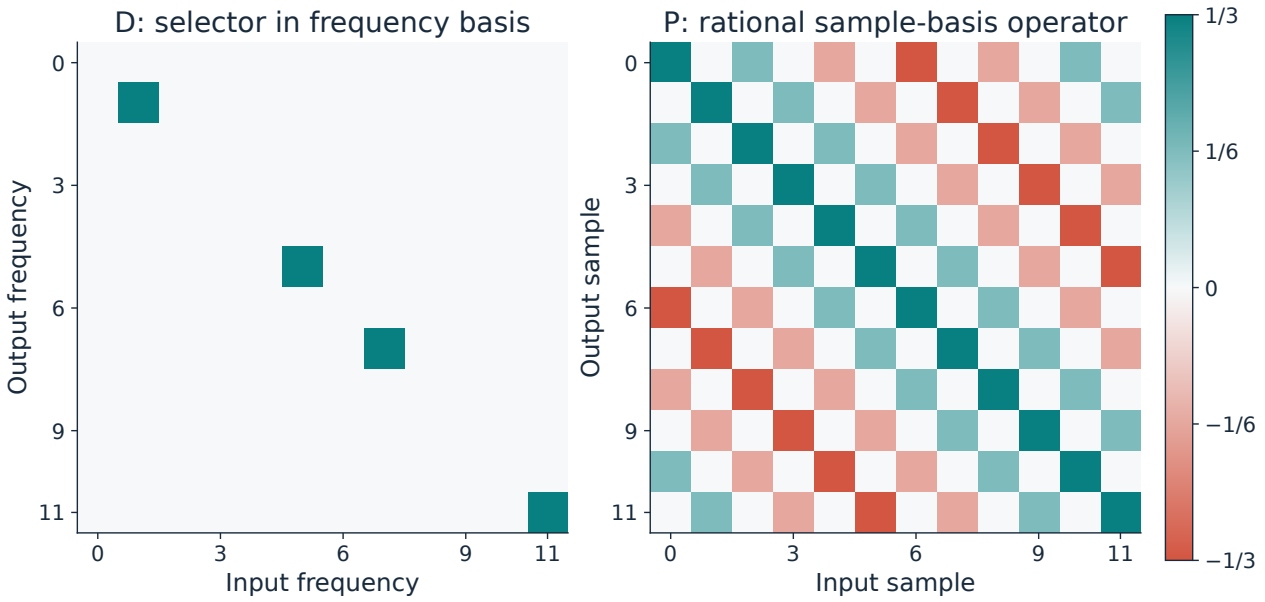
$$\widehat{c_{12}}(k) = \sum_{a \in U_{12}} \sum_{n=0}^{11} \zeta^{(a-k)n} = 12\, \mathbf{1}_{k \in U_{12}}.$$

The inner geometric sum is twelve when $a = k$ modulo twelve and zero otherwise. The selector and the Fourier support are now the same exact data.

A diagonal selector becomes a rational filter

The filter is diagonal in one basis and a rational convolution in the other.

Let $F_{na} = \zeta^{na}/\sqrt{12}$ and $D_{aa} = \mathbf{1}_{a \in U_{12}}$. With the positive-sign unitary matrix F, the negative-sign DFT satisfies $F^\dagger x = \widehat{x}/\sqrt{12}$.



Left: the frequency selector D. Right: its transported sample-basis matrix P. The right colorbar displays its five possible rational entries. The two panels use different scales.

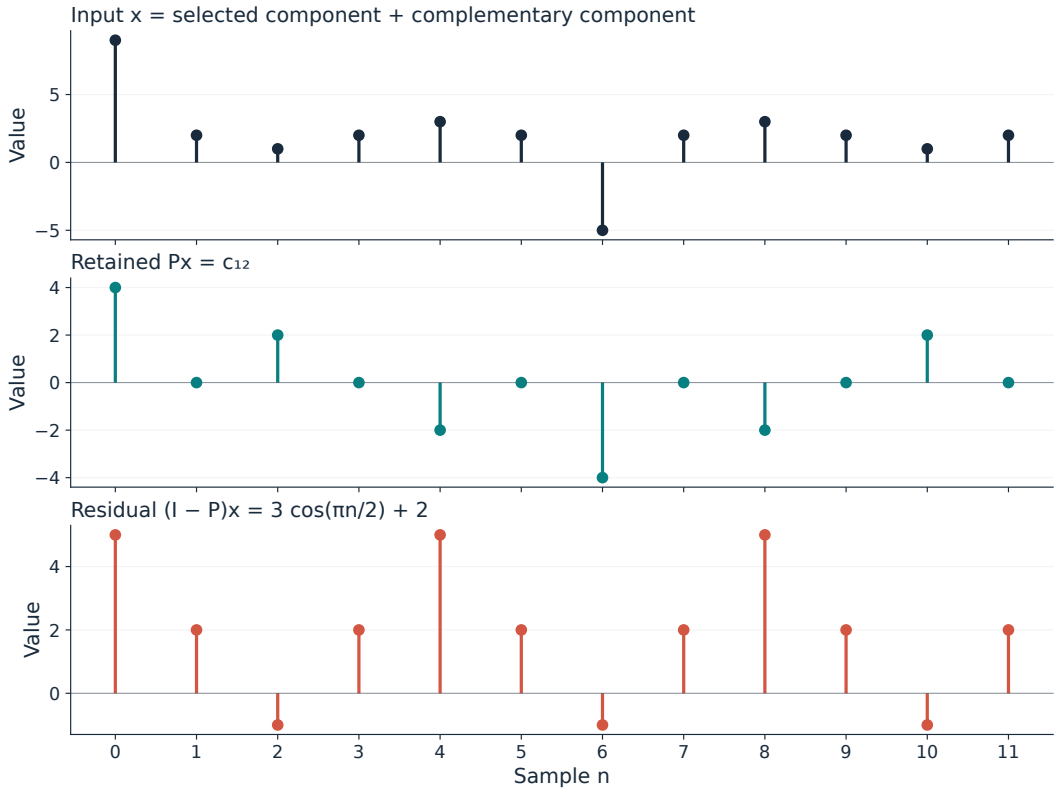
$$P = FDF^\dagger, \qquad P_{nm} = \frac{c_{12}(n-m)}{12}.$$

Because $D^2 = D$ and $F^\dagger F = I$, we obtain $P^2 = P$. Because $D^\dagger = D$, we also obtain $P^\dagger = P$. This is an orthogonal projector of rank four.

Give the projector something to remove

The input contains an arithmetic component and a deliberately complementary component.

Take $x(n) = c_{12}(n) + 3\cos(\pi n/2) + 2$. Its extra cosine uses modes $3, 9$ and its constant uses mode 0. All three are outside U_{12}.



The projector retains the primitive-order component exactly. The bottom panel shows everything it removes, so the loss remains visible.

$$Px = c_{12}, \qquad (I - P)x = 3\cos(\pi n/2) + 2.$$

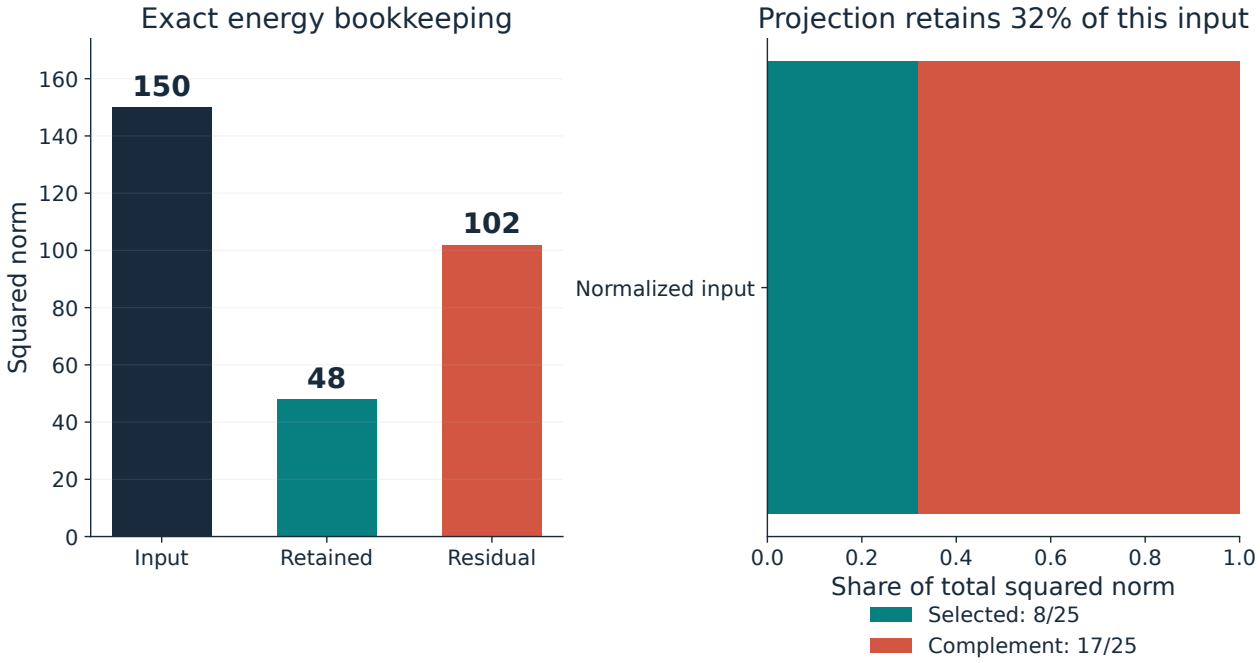
The exact input is $(9, 2, 1, 2, 3, 2, -5, 2, 3, 2, 1, 2)$. Calling the complementary component "noise" is a modeling choice. Arbitrary noise need not lie in the kernel of P.

Account for the entire input

Exact filtering removes energy as well as modes.

The selected and complementary components are orthogonal. With the unnormalized squared norm $\|x\|^2 = \sum_{n=0}^{11} |x(n)|^2$,

$$\|x\|^2 = 150, \qquad \|Px\|^2 = 48, \qquad \|(I-P)x\|^2 = 102.$$



The energy identity is $150 = 48 + 102$. The normalized input has selected squared-norm share $48/150 = 8/25$ and complementary share $17/25$.

On the normalized state $x/\sqrt{150}$, the ideal Lüders projective measurement $\{P, I-P\}$ has selected-outcome probability $8/25$. Conditional on that outcome, the resulting normalized state is $c_{12}/\sqrt{48}$.

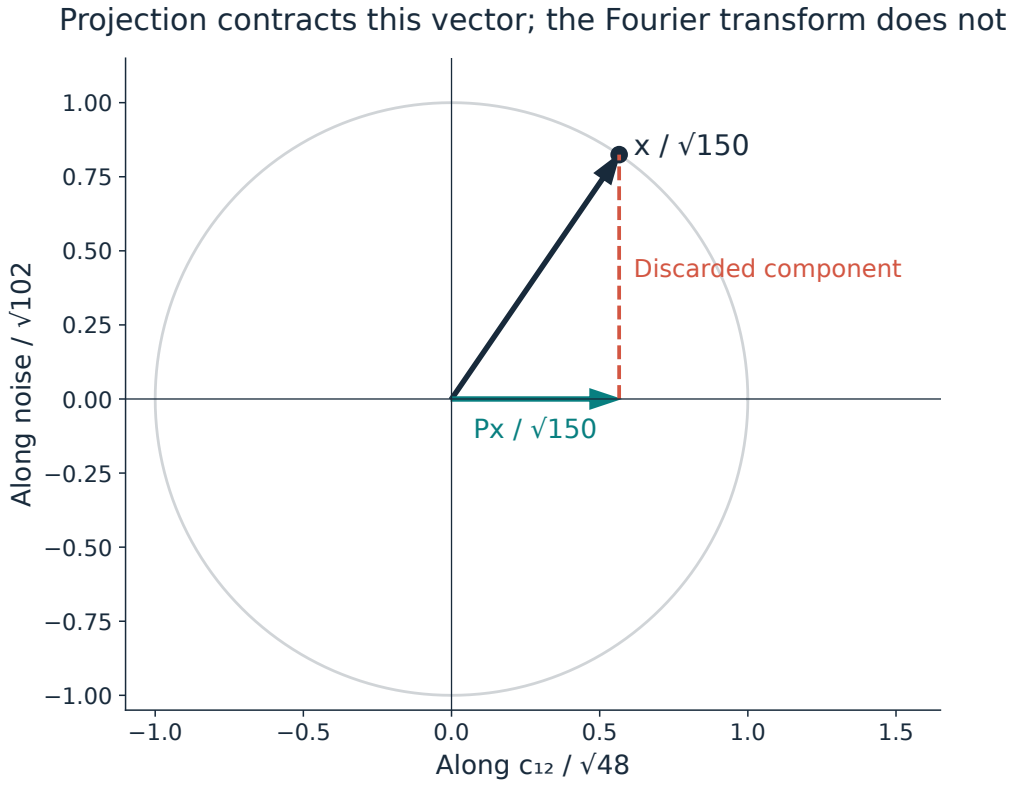
> This conditional measurement is not deterministic unitary denoising. The transform F is reversible; the projection P discards the complementary subspace.

Projection and unitary change are different operations

A basis change preserves the whole vector. Projection keeps only one orthogonal component.

Let $e_s = c_{12}/\sqrt{48}$ and $e_r = (I - P)x/\sqrt{102}$. They are orthonormal, so the normalized input has the exact two-dimensional representation

$$\frac{x}{\sqrt{150}} = \sqrt{\frac{8}{25}}e_s + \sqrt{\frac{17}{25}}e_r.$$



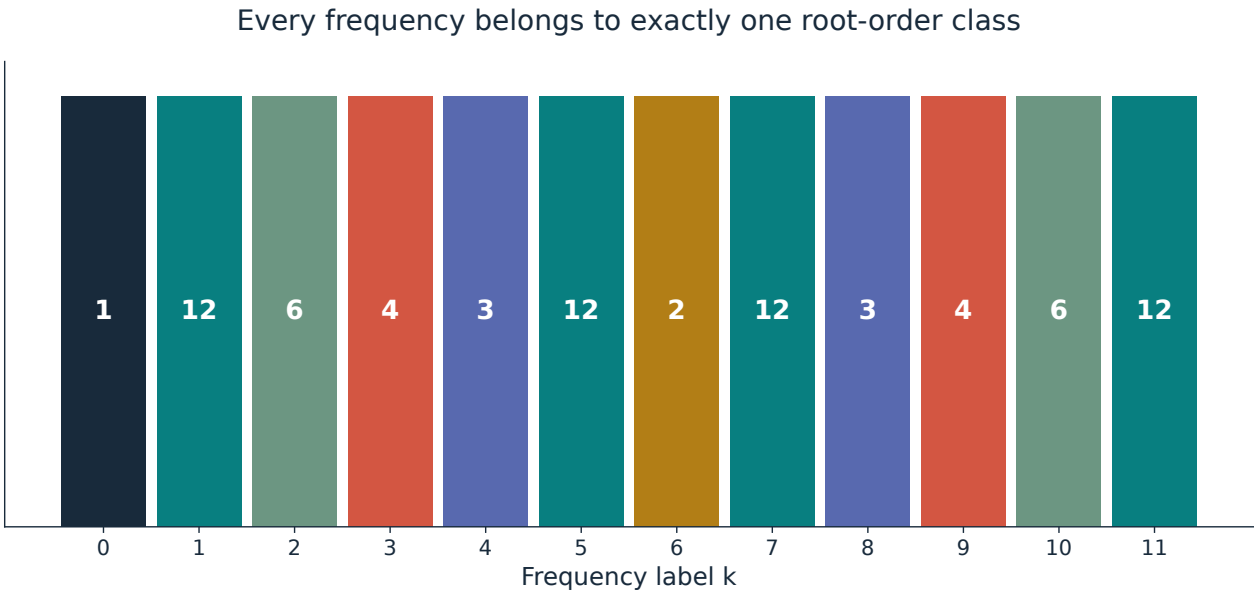
This plane is the actual span of the selected signal and its complementary component. The dashed segment is the discarded component. Its length is $\sqrt{17/25}$; the retained vector has length $\sqrt{8/25}$.

The Fourier transform preserves the input norm and permits exact inversion. The projector maps every complementary vector to zero. That distinction controls what may legitimately be called reconstruction, filtering, or measurement.

All twelve modes belong to six order classes

The primitive-twelve projector is one packet in a complete orthogonal decomposition.

A frequency label k has root order $d = 12/\gcd(k, 12)$. The possible orders are the divisors $1, 2, 3, 4, 6, 12$.



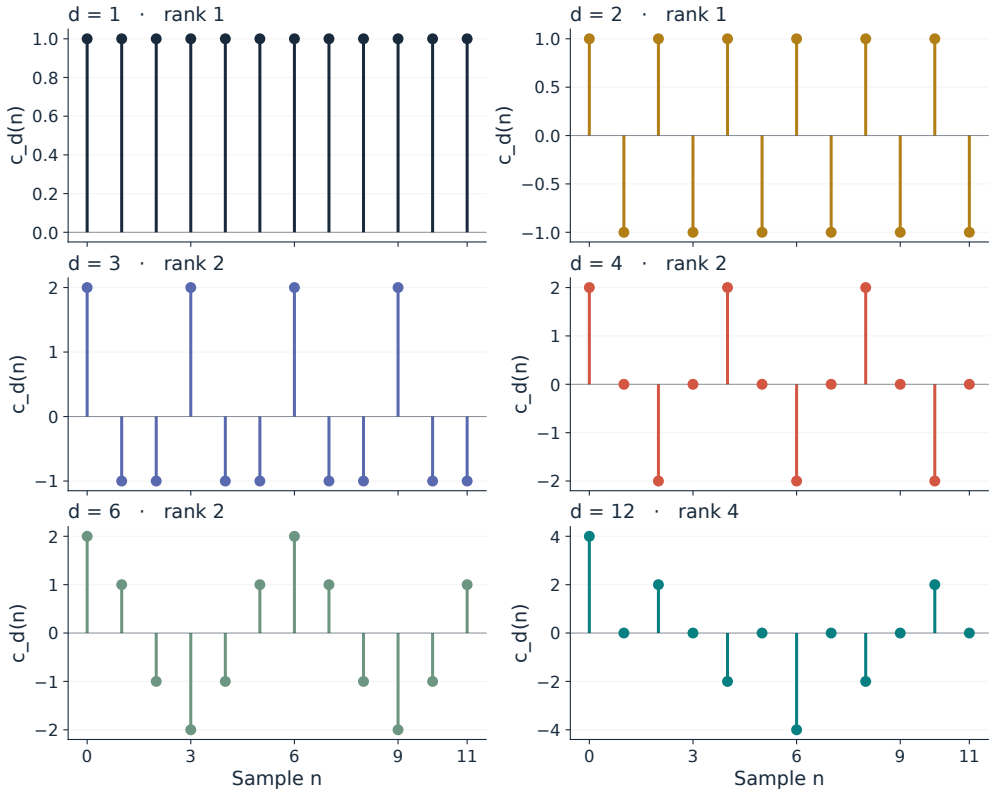
The number inside each tile is its root order d. The horizontal label is frequency k. Every label belongs to one class; no labels are dropped.

$$P_{dnm} = \frac{c_d(n-m)}{12}, \quad P_d P_e = \delta_{de} P_d, \quad \sum_{d|12} P_d = I.$$

These are character-order projectors. They select Fourier modes by root order. They are different from coefficient-residue selectors, which select formal-series coefficients by their index modulo an integer.

Six exact filters in one ambient space

Every order class has an integer Ramanujan kernel and a rational projector.

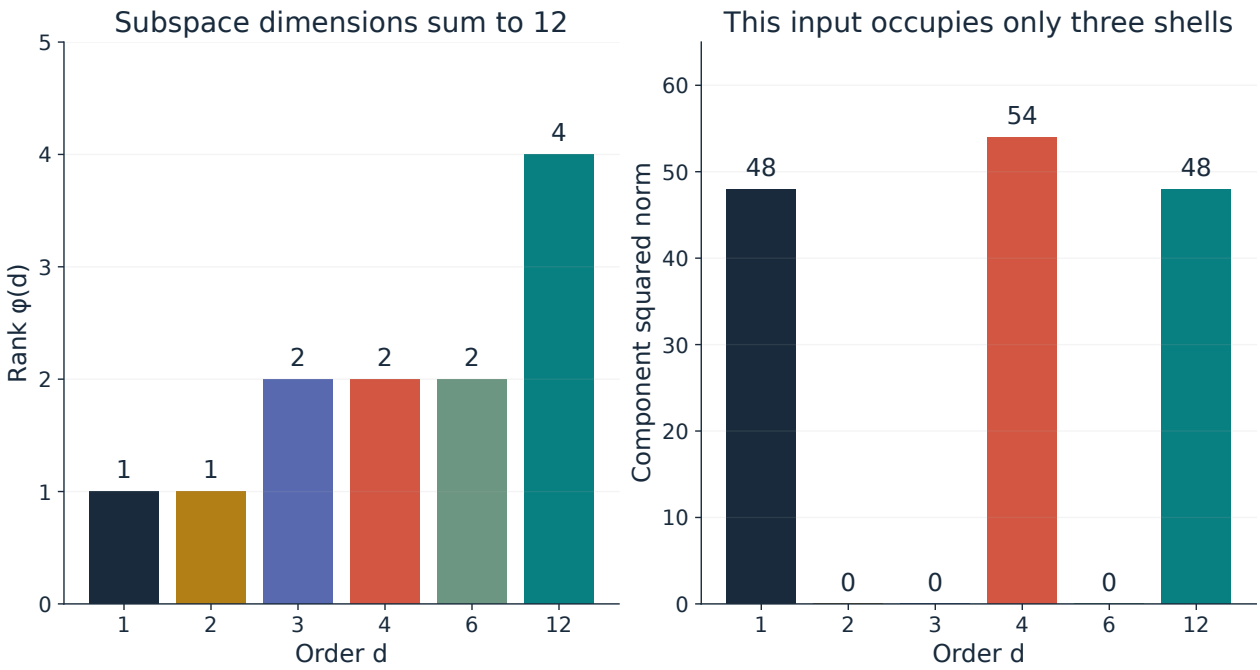


Each panel displays $c_d(n)$ over the common twelve-sample ambient period. The associated matrix is $P_{dnm} = c_d(n-m)/12$, so every kernel uses the same denominator twelve.

For $d < 12$, the kernel repeats with period dividing d. Its projector still acts on all twelve sample coordinates. The ranks are $1, 1, 2, 2, 2, 4$, respectively.

A packet's capacity is not its occupied energy

Ranks describe the subspaces. Energies describe a particular input.



Left: the six subspace dimensions sum to twelve. Right: the particular input x has energy 48 in order one, 54 in order four, and 48 in order twelve. The other packets exist but have zero component for this input.

$$x = \sum_{d|12} P_d x, \qquad \|x\|^2 = \sum_{d|12} \|P_d x\|^2.$$

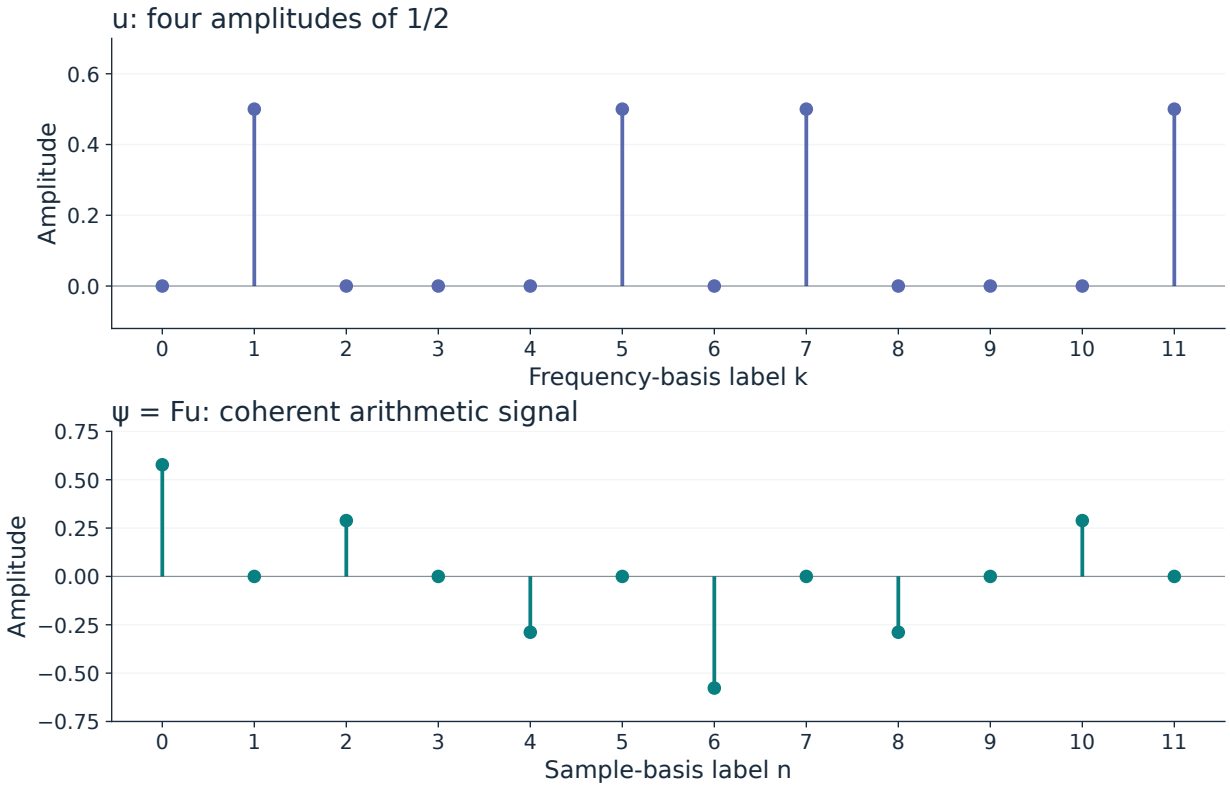
The constant 2 contributes $12 \cdot 2^2 = 48$ to order one. The cosine contributes 54 to order four. The Ramanujan component contributes 48 to order twelve. Keeping all six packets gives exact synthesis. Keeping only one packet performs a lossy selection.

The arithmetic signal is a coherent amplitude vector

Normalization makes the signal a unit vector without changing its relative phases.

Define $u_k = 1/2$ for $k \in U_{12}$ and zero otherwise. Then $\|u\| = 1$, and

$$\psi = Fu, \qquad \psi_n = \frac{c_{12}(n)}{\sqrt{48}}, \qquad F^\dagger \psi = u.$$

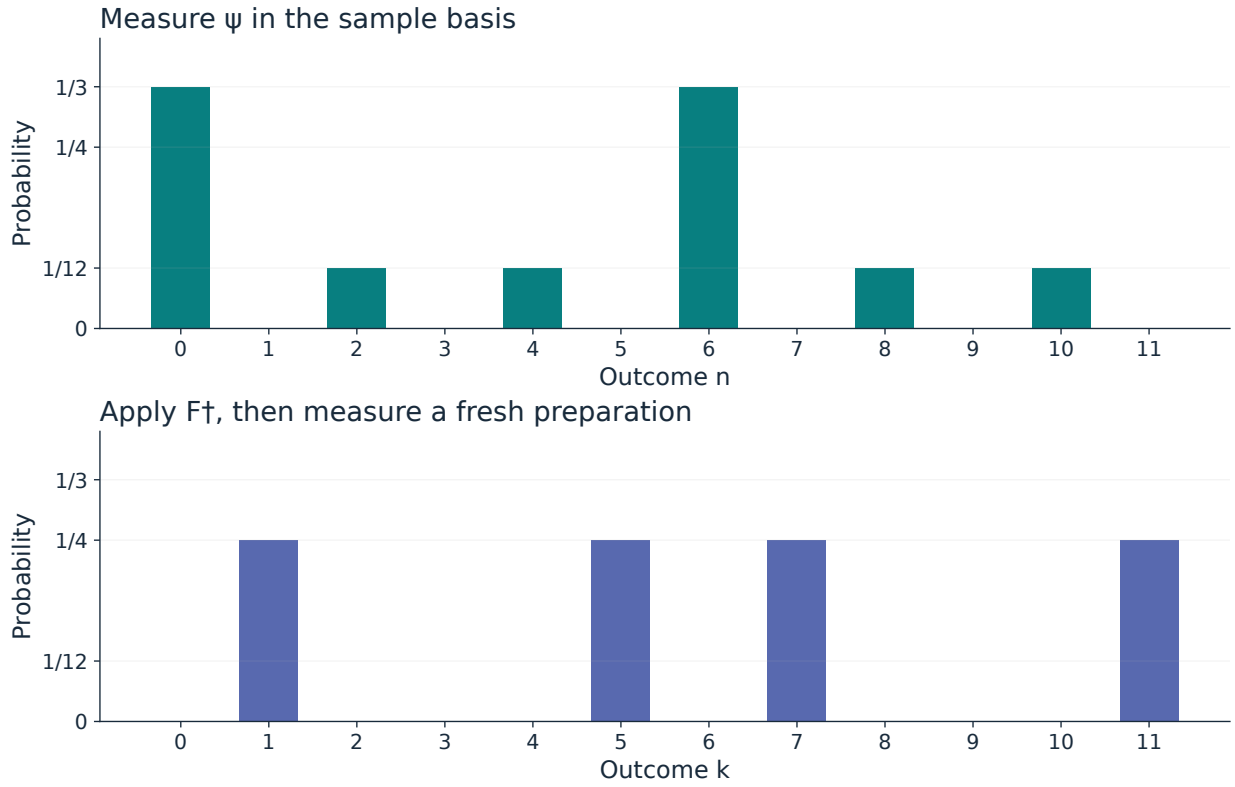


Separate panels preserve the different basis labels: k for the frequency-basis coordinates and n for the sample-basis coordinates. Heights are amplitudes, including their signs.

The normalization is exact because $\sum_n c_{12}(n)^2 = 48$. The matrix F is unitary, so no probability mass is lost during this coherent change of basis.

Amplitudes and probabilities tell different stories

Measurement squares magnitudes; interference depends on coherent amplitude before that step.



Two alternative procedures on fresh preparations. The first measures ψ directly. The second applies $F^\dagger$ before measurement. Neither histogram is the amplitude plot from the previous page.

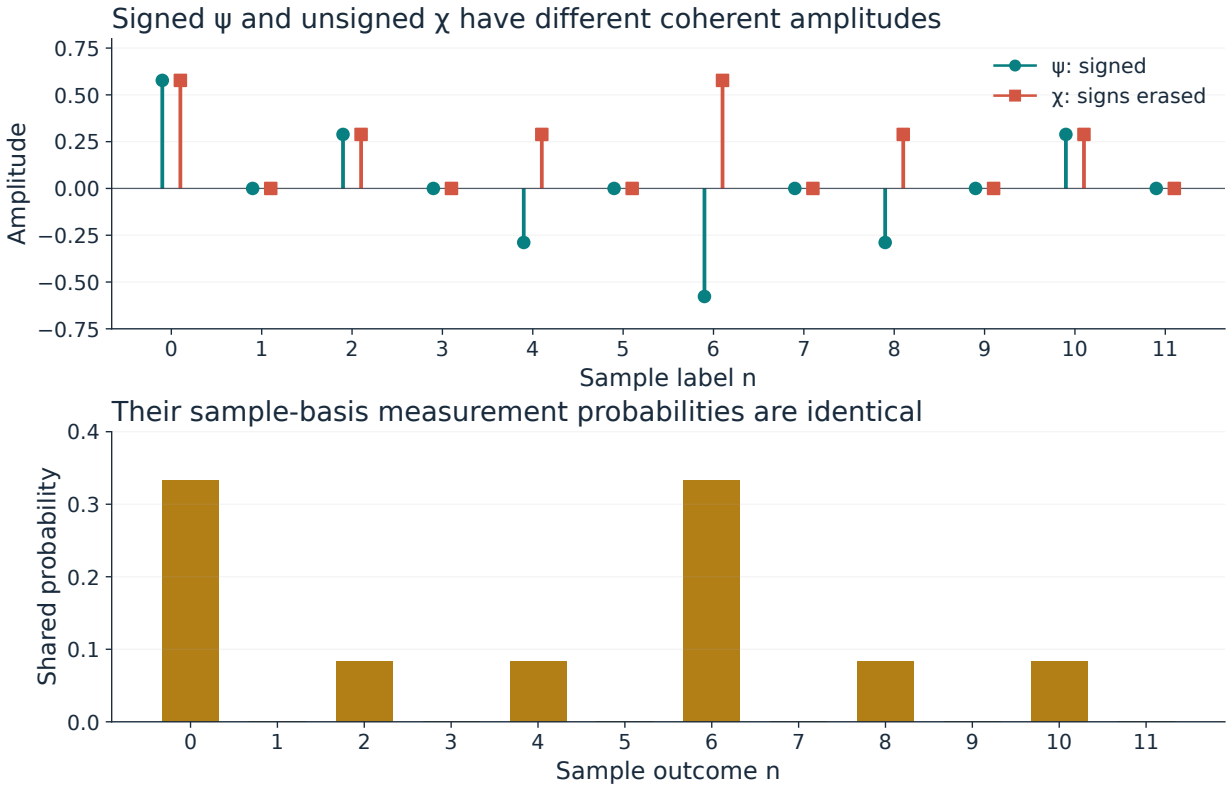
The first procedure gives probabilities $c_{12}(n)^2/48$. The second gives probability $1/4$ at each of $1, 5, 7, 11$ and zero elsewhere. One run returns one outcome. Measuring the sample basis first does not leave the original coherent state available for the second procedure. These distributions are alternative experimental choices, rather than successive readouts of one unchanged state.

Identical histograms can hide orthogonal states

Erasing the signs leaves sample probabilities unchanged and changes the state.

Let $\chi_n = |c_{12}(n)|/\sqrt{48}$. Both states are normalized and

$$|\psi_n|^2 = |\chi_n|^2 \quad \text{for every } n, \qquad \langle \psi, \chi \rangle = 0.$$

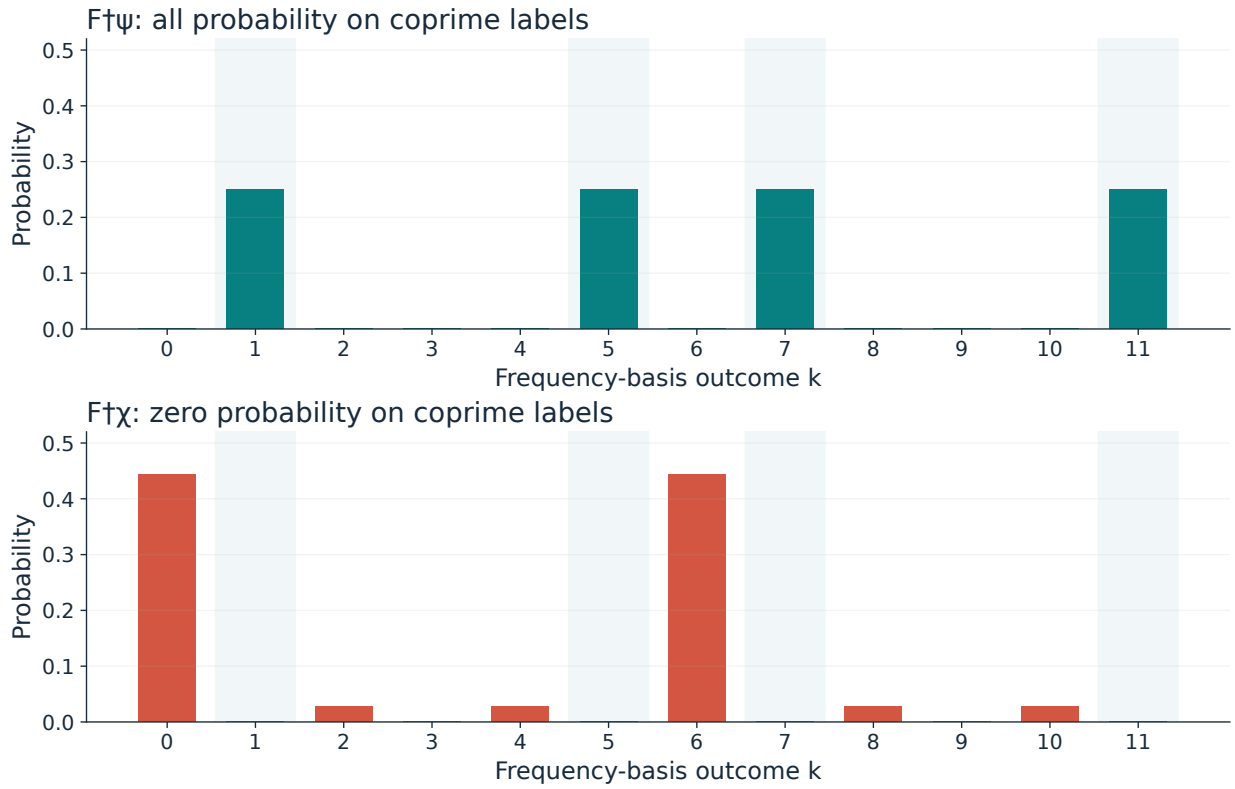


The top panel shows the phase information carried by negative amplitudes. The bottom panel is shared by the two states because measurement in this basis depends only on squared magnitudes.

The orthogonality follows from equal positive-support and negative-support energies: each contributes 24 before division by 48. A sample-basis probability histogram therefore cannot reconstruct the coherent state.

A basis change makes the hidden distinction observable

The signed state recovers all four coprime labels. The unsigned state recovers none.



The pale vertical strips mark the coprime labels. After $F^{\dagger}$, the signed state assigns them total probability one; the unsigned state assigns them total probability zero.

The exact unnormalized DFT of the unsigned integer signal is

$$(16, 0, 4, 0, 4, 0, 16, 0, 4, 0, 4, 0).$$

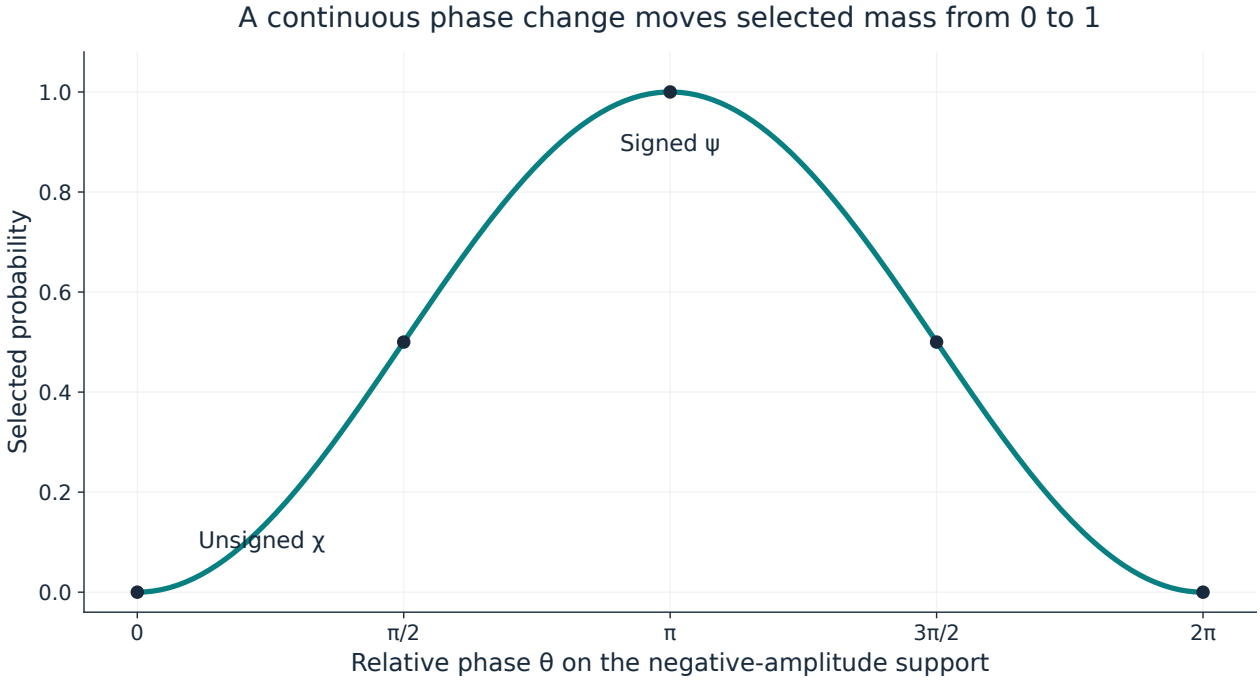
After state normalization and unitary transformation, its probabilities are $4/9$ at $0, 6$, $1/36$ at $2, 4, 8, 10$, and zero elsewhere. Since $|c_{12}(n)|$ has period six, its odd Fourier labels vanish.

Move continuously between the two states

Changing only relative phase moves the selected probability from zero to one.

Let v_+ contain the positive entries of $c_{12}/\sqrt{48}$ and v_- contain the absolute values of its negative entries. Their supports are disjoint and each has squared norm $1/2$. Define

$$\chi_\theta = v_+ + e^{i\theta} v_-, \qquad \chi_0 = \chi, \quad \chi_\pi = \psi.$$



The selected probability is the exact formula $\sin^2(\theta/2)$. The continuous curve is its floating rendering. Black dots mark exactly checked quarter-turn cases $0, 1/2, 1, 1/2, 0$.

Indeed, $\chi_\theta = \frac{1+e^{i\theta}}{2}\chi + \frac{1-e^{i\theta}}{2}\psi$. Since $P\chi = 0$ and $P\psi = \psi$, we get $\|P\chi_\theta\|^2 = |1-e^{i\theta}|^2/4 = \sin^2(\theta/2)$.

> The endpoint statements are covered by the inherited Lean development. This continuous family is an elementary prose derivation and a checked illustration, not an additional inherited Lean theorem.

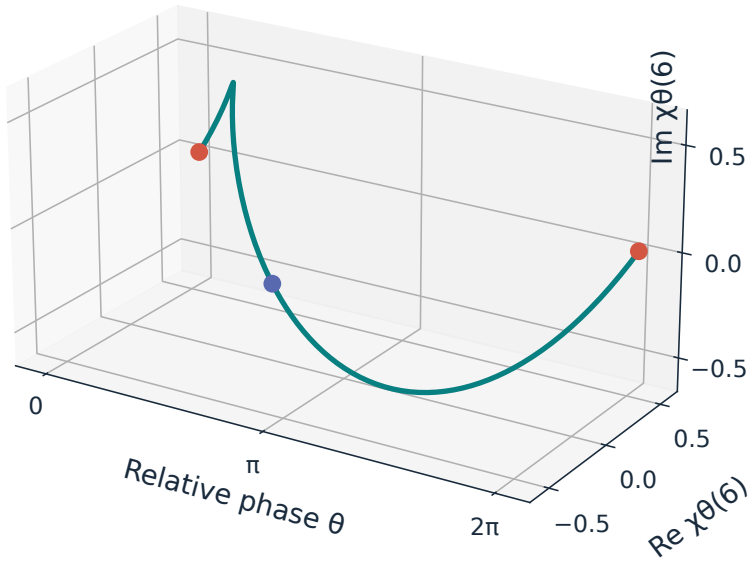
A real three-dimensional view of one amplitude

The plot follows an actual complex amplitude as the relative phase advances.

At sample $n = 6$, the phase-family amplitude is

$$\chi_\theta(6) = \frac{e^{i\theta}}{\sqrt{3}}.$$

One amplitude traces a circle as phase advances



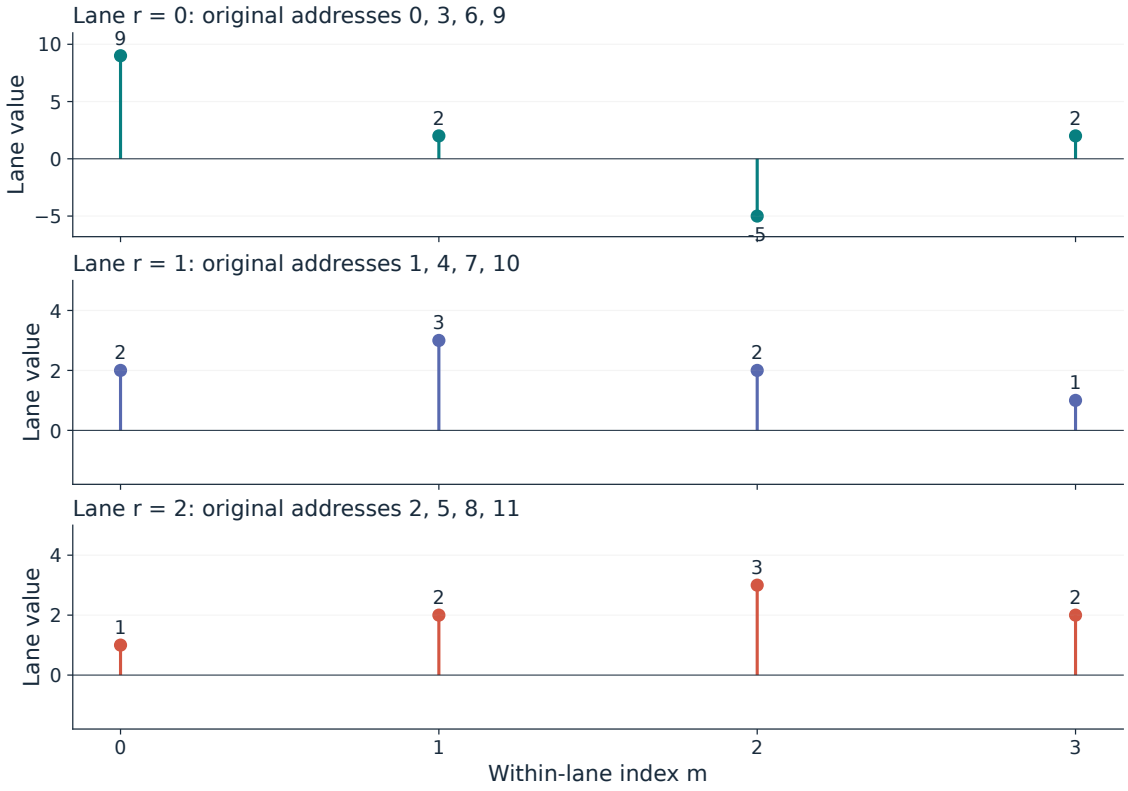
Coordinates are $(\theta, \operatorname{Re}\chi_\theta(6), \operatorname{Im}\chi_\theta(6))$. In the real-imaginary plane the amplitude traces a radius-$1/\sqrt{3}$ circle. Advancing the phase coordinate produces the three-dimensional trajectory.

Its magnitude is constant, so the sample-six probability stays $1/3$ throughout. Nevertheless, interference changes the probability of the selected subspace according to the preceding page. The geometry separates phase change from magnitude change.

Rearrange twelve samples into three exact lanes

Lane extraction changes addresses while preserving all twelve values.

For $r = 0, 1, 2$ and $m = 0, 1, 2, 3$, define $(E_r x)(m) = x(r + 3m)$.



The lane numbers and original addresses are explicit. Interleaving all three lanes reconstructs the original vector exactly. Keeping fewer lanes would be a different, lossy observation.

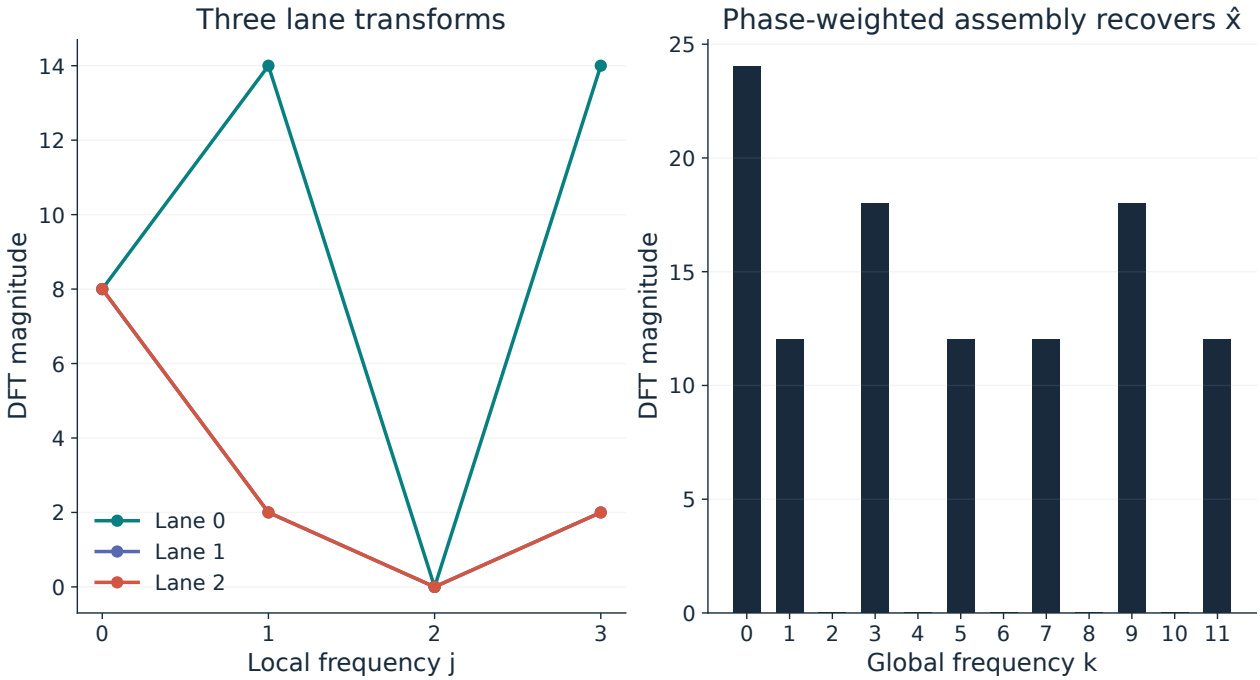
The lanes are $(9, 2, -5, 2)$, $(2, 3, 2, 1)$, and $(1, 2, 3, 2)$. Twelve values remain twelve values. A shorter address in each lane does not by itself reduce information content or computation cost.

Phase factors connect the lane spectra

Each local Fourier transform contributes to every appropriate global frequency.

With four-point negative-sign DFTs in the lanes,

$$\widehat{x}(k) = \sum_{r=0}^{2} e^{-2\pi i k r/12} \widehat{E_r x}(k \bmod 4).$$

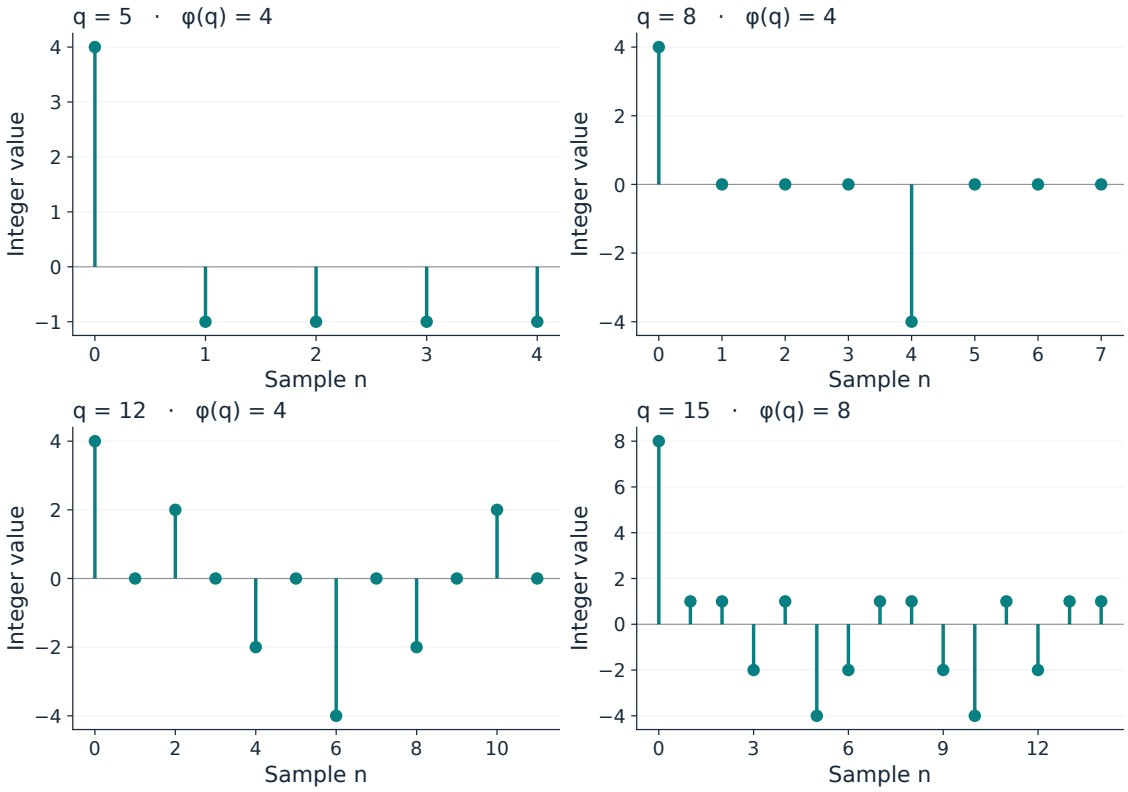


Left: magnitudes of the three local transforms. Right: the recovered global spectrum. Assembly uses the complex coefficients and phase factors; adding the displayed magnitudes would be incorrect.

The identity follows by writing $n = r + 3m$ in the full DFT sum. Its phase factors retain the distinction between the lanes. The diagnostic replay compares this assembly with an independent direct twelve-point DFT at a declared tolerance of 10^{-12}.

The construction extends beyond this picture

The integer waveform changes with the modulus, while the same arithmetic identity supplies it.



Four exact Ramanujan signals constructed by the divisor formula. Their periods and amplitudes differ. These examples illustrate the general formulas; their finite verification does not prove those formulas universally.

For general q, character orthogonality gives

$$\widehat{c_q}(k) = q\mathbf{1}_{\gcd(k,q)=1}, \qquad \sum_{n=0}^{q-1} |c_q(n)|^2 = q\varphi(q).$$

The associated primitive-order projector has rank $\varphi(q)$. The general mathematical conclusions follow from the finite-sum derivations in the main paper. The source verifier separately checks its declared finite range $2 \le q \le 64$.

State exactly what each passage preserves

The value of a bridge is the contract it supplies.

Passage	Preserves	Additional requirement or loss
Selector D to $P = FDF^\dagger$	Rank, idempotence, orthogonality	A specified unitary Fourier basis
Vector x to $F^\dagger x$	Full vector, inner product, norm	Retain all complex coefficients
Vector x to Px	Selected component	Complementary component removed
Vector to all polyphase lanes	All sample values	Every lane and its address retained
Amplitudes to probabilities	Outcome probabilities in one basis	Relative phase discarded
Projective measurement	Born probabilities and conditional state	One outcome per run, conditional branch

The two states ψ and χ give a concrete obstruction: the map from coherent amplitudes to a single-basis probability histogram is not injective. Their common histogram cannot determine which arithmetic selection will appear after the Fourier transform.

> The guide uses an actual twelve-dimensional Hilbert space. A four-qubit realization would require a specified embedding and implementation. Neither a gate synthesis nor an efficiency bound is supplied here.

Exact data, floating rendering, and formal coverage

Three kinds of evidence support different parts of the guide.

Exact Python data. The source verifier uses integers, rational fractions, and cyclotomic quotient-ring arithmetic. The release visualization script adds an independent rational-matrix implementation for all six shell products and completeness, component energy bookkeeping, the unsigned Fourier table, and the phase-family quarter-turn cases. The assertion counts and hashes are recorded in the accompanying verification results.

Floating diagnostics. Fourier matrix checks, lane assembly, phase curves, and figure coordinates use NumPy and Matplotlib. Their residuals are reported against an explicit 10^{-12} tolerance. Passing that tolerance is numerical evidence, not exact arithmetic or an interval certificate.

Inherited Lean coverage. The later five-module source contains exact rational and cyclotomic calculations, normalized finite Fourier statements, and semantic transport into ordinary complex matrices and state vectors. The Lean guide maps the checked declarations and records the inherited certificate. This visual release does not claim a fresh Lean compilation.

> The script is `checks/make_visuals.py`. Its final bounded-run manifest and scientific result are in `verification/visual_release_run_v2/`. Every scientific figure is supplied as PDF, SVG, and PNG. No image-generation model was used.

What the construction establishes

Coprimality, signal filtering, and coherent measurement share a specified exact operator.

For the twelve-sample model, the guide has constructed every ingredient: the primitive-root selector, integer phase-sum signal, rational projector, complementary residual, complete order packets, normalized states, and alternative measurement probabilities. The retained arithmetic is not inferred from a suggestive resemblance. It follows from explicit finite Fourier identities.

The phase-erasure example is decisive. Two orthogonal normalized states have identical sample-basis probabilities. One inverse Fourier transform places all mass on the coprime labels for the signed state and none there for the unsigned state. Relative phase is therefore an operational part of the realization, not optional decoration.

The construction also states its losses. Projection removes a complementary subspace. Measurement returns one outcome and generally changes the state. Erasing phase destroys information. Keeping all complex Fourier coefficients or all polyphase lanes permits exact inversion.

> Ramanujan sums, Fourier orthogonality, spectral projection, and the quantum state-vector rules are classical. The paper's purpose is their exact integration within Realization Calculus. Efficient preparation, gate synthesis, incomplete-data recovery, hardware robustness, and broad quantum advantage require separate results.

Read together. The traditional paper supplies the full derivations and scope. The Lean guide supplies formal declaration contracts and certificate provenance. This guide supplies the exact worked example and its reproducible geometry.

PART II

Symmetry collapse in quantum computation

Invariant packets, moving frames, and compressed representations organize quantum evolution. The treatment keeps the cost of coordinates, updates, and measurement alongside the dimension of the retained state space.

IN THIS PART

Foundations of packet simulation 48

Exact symmetry families 63

States, measurements, and open dynamics 84

The complexity boundary 89

Algorithms and open directions 95

CHAPTER 3

Foundations of packet simulation

3.1 The symmetry-collapse question

3.1.1 The computational mismatch

An n-qubit pure state has 2^n complex amplitudes in the computational basis. That description is universal but frequently redundant. A number-preserving circuit started with two excitations never visits basis strings of weight zero, one, three, or more. A permutation-symmetric circuit started in a symmetric state never distinguishes two strings that differ only by a relabeling of qubits. A quantum walk on a highly symmetric graph may evolve entirely in the span of a polynomial number of orbit states. In each case the full Hilbert space is a valid container but not the intrinsic state space of the computation.

The question is therefore not whether a quantum circuit acts on an exponential ambient space. It is whether the reachable part of that space, together with the gate action and measurement, has a polynomial classical representation.

Definition 3.1 (Exact symmetry collapse)**.** A uniform quantum circuit family exhibits an exact symmetry collapse when its acceptance probability can be computed by an exact classical simulation on an invariant representation whose description and update costs are polynomial in the input length.

Definition 3.2 (Approximate symmetry collapse)**.** A uniform circuit family exhibits an approximate symmetry collapse to error $\varepsilon(n)$ when a polynomial-size classical representation reproduces its acceptance probability to additive error $\varepsilon(n)$, with the error below the promise gap required by the decision problem.

3.1.2 Four conclusions that must not be conflated

The phrase “symmetry reduction” is used for several logically different achievements.

1. **Structural reduction.** A projector, quotient, or representation-theoretic coordinate change removes redundant coordinates.
2. **Practical acceleration.** A concrete implementation uses less time or memory on tested instances.
3. **Polynomial classical simulation.** A uniform family admits a deterministic polynomial-time classical algorithm.
4. **Universal complexity collapse.** Every polynomial-size quantum circuit admits such a representation.

The first three occur in this monograph for declared subclasses. The fourth would imply $BQP \subseteq P$ and is not established. A benchmark can support the second conclusion without proving the third. A small invariant label set can support the first conclusion while leaving exponentially large internal blocks and therefore failing to prove the third.

3.1.3 The four audit quantities

For a proposed reduction, define the schematic complexity record

$$\mathfrak{C}_n = \big(N_{\text{packets}}, D_{\text{internal}}, S_{\text{transfer}}, L_{\text{leakage}}\big).$$

Here:

- N_{packets} is the number of reachable symmetry labels.
- D_{internal} is the packet dimension or the size of a compressed internal representation.
- S_{transfer} records the sparsity and cost of gate-induced transitions.
- L_{leakage} measures departure from the proposed representation.

A complete claim must also state the cost of discovering this structure from the input. A decomposition that is small but exponentially hard to find does not yield a polynomial algorithm.

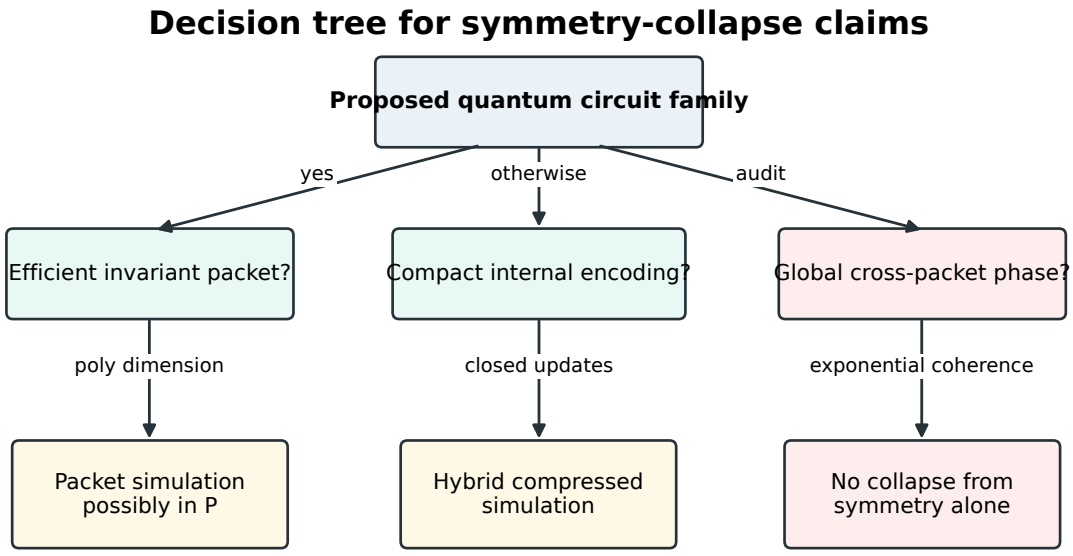


Figure 3.1. A disciplined decision tree for proposed symmetry-collapse claims.

3.1.4 Why symmetry is only the beginning

Suppose a group G acts on the Hilbert space and the circuit commutes with this action. Representation theory decomposes the space into isotypic pieces. This gives a canonical block structure, but three questions remain.

First, how many pieces are reachable from the input? Second, how large is each piece or how compactly can its states be encoded? Third, can each gate and measurement be applied in those coordinates without reconstructing the full state? The monograph repeatedly returns to this three-stage distinction:

$$\text{symmetry label} \longrightarrow \text{reachable representation} \longrightarrow \text{efficient classical algorithm}.$$

Principle 3.3 (No free collapse)**.** *Block diagonality is a structural statement. Polynomial-time classical simulation requires an efficient basis or encoding, efficient updates, and efficient output evaluation in addition to the block decomposition.*

3.2 Projectors and quantum packets

3.2.1 Orthogonal packet decompositions

Let $\mathcal{H}$ be a finite-dimensional Hilbert space and let $\{P_\alpha\}_{\alpha\in A}$ be orthogonal projectors satisfying

$$P_\alpha P_\beta = \delta_{\alpha\beta} P_\alpha, \qquad \sum_{\alpha\in A} P_\alpha = I.$$

Every state and operator decompose as

$$|\psi\rangle = \sum_\alpha P_\alpha|\psi\rangle, \qquad X = \sum_{\alpha,\beta} P_\alpha X P_\beta.$$

The diagonal blocks act within packets. The off-diagonal blocks transfer amplitude or coherence between packets.

Define the packet-transfer graph of an operator X by placing an edge $\beta \to \alpha$ whenever

$$P_\alpha X P_\beta \neq 0.$$

An exactly symmetry-preserving operator has only diagonal edges. A sparse transfer graph may still permit efficient simulation if the active packets and internal states remain compact.

3.2.2 Conserved observables

Let

$$Q = \sum_k q_k P_k$$

be a self-adjoint observable with spectral projectors P_k.

Theorem 3.4 (Invariant quantum packets)**.** *If a unitary U satisfies $[U, Q] = 0$, then*

$$P_i U P_j = 0 \qquad (i \neq j),$$

and

$$U = \bigoplus_i U_i, \qquad U_i = P_i U P_i\big|_{P_i\mathcal{H}}.$$

If $P_k|\psi\rangle = |\psi\rangle$, then $P_k U|\psi\rangle = U|\psi\rangle$.

Proof. Since U commutes with Q, it commutes with every spectral projector obtained from the functional calculus of Q. Thus

$$P_i U P_j = U P_i P_j = 0$$

for $i \neq j$. Summing the diagonal blocks gives the direct-sum form. The final statement follows from

$$P_k U|\psi\rangle = U P_k|\psi\rangle = U|\psi\rangle.$$

□

3.2.3 Residue and character packets

Suppose Q has integer spectrum. For a modulus M, define

$$B_r^{(M)} = \sum_{k \equiv r \pmod{M}} P_k.$$

Let

$$V_M = \exp\left(\frac{2\pi i}{M} Q\right), \qquad \omega_M = e^{2\pi i/M}.$$

Then

$$B_r^{(M)} = \frac{1}{M} \sum_{\ell=0}^{M-1} \omega_M^{-r\ell} V_M^{\ell}. \tag{3.1}$$

On a Q-eigenvector, the right side is the roots-of-unity filter. The same finite-character formula selects residue classes in signal polyphase analysis and charge sectors in quantum theory.

3.2.4 Finite-group projectors

Let a finite group G act unitarily by U_g. For an irreducible character χ of degree d_χ, the central isotypic projector is

$$P_\chi = \frac{d_\chi}{|G|} \sum_{g \in G} \overline{\chi(g)} U_g. \tag{3.2}$$

Character orthogonality gives

$$P_\chi P_\psi = \delta_{\chi\psi} P_\chi, \qquad \sum_\chi P_\chi = I$$

on the represented space. If an operator A commutes with every U_g, then

$$P_\chi A P_\psi = 0 \qquad (\chi \neq \psi).$$

This framework includes momentum, parity, total spin, point-group symmetry, permutation symmetry, and stabilizer syndrome labels. The isotypic projector identifies a symmetry type. It does not necessarily resolve the multiplicity space within that type.

3.2.5 Nested refinements

If one conserved label is too coarse, intersect commuting projector families. For mutually commuting observables $Q_1, \ldots, Q_s$, a joint packet has the form

$$P_{\boldsymbol{\alpha}} = P_{\alpha_1}^{(1)} \cdots P_{\alpha_s}^{(s)}.$$

The dimension can decrease dramatically, but each extra label must be efficiently computable and preserved. A refinement that requires exponential preprocessing merely relocates the complexity.

Remark 3.5. A packet decomposition is basis independent as an orthogonal direct sum, although a practical simulator must choose a concrete basis or compressed coordinate system within each active packet.

3.3 Uniform circuits and the classical cost model

3.3.1 Circuit families

A quantum decision procedure is a polynomial-time uniform family of circuits $\{C_x\}$, where the input string x determines a circuit on $n = n(|x|)$ qubits with at most $\mathrm{poly}(|x|)$ gates. Let

$$C_x = G_m \cdots G_1.$$

A classical simulation theorem must describe an algorithm receiving the same succinct description of x and the circuit. It is not enough to assume that an exponentially large matrix or a complete orbit list has already been supplied.

3.3.2 The packet access model

For each input length, suppose there is a packet projector P_n and a basis

$$\mathcal{B}_n = \{|b_1\rangle, \ldots, |b_{D_n}\rangle\}$$

of $P_n\mathcal{H}_n$. The simulation interface has four routines.

1. **Discovery and indexing.** Produce or index $\mathcal{B}_n$ in polynomial time.
2. **Initialization.** Express the initial state in this basis.
3. **Gate update.** Apply each restricted gate $P_nG_jP_n$ to the packet coordinates in polynomial time.
4. **Measurement.** Compute the final acceptance probability to the needed precision.

The basis need not be listed explicitly if there is a compact data structure supporting these operations. Stabilizer tableaux and covariance matrices are examples of representations that encode exponentially many computational-basis amplitudes without storing them individually.

3.3.3 Precision and promise gaps

A BQP decision problem has an inverse-constant promise gap after standard amplification. A deterministic classical simulator need only estimate the acceptance probability to an additive error smaller than this gap. If the gate entries are algebraic or otherwise polynomial-time computable, ordinary numerical analysis gives polynomial bit complexity for polynomially many updates, provided condition numbers and error amplification are controlled.

3.3.4 State-vector cost inside a packet

If the simulator stores D amplitudes and a local gate can be applied in $O(nD)$ arithmetic operations, a depth-d packet simulation has schematic cost

$$C_{\text{packet}} = O(dnD) + C_{\text{compile}}.$$

The corresponding full state-vector path is

$$C_{\text{full}} = O(dn2^n).$$

These formulas are intentionally schematic. Cache behavior, indexing, vectorization, and basis compilation determine actual runtime. The asymptotic distinction is exact when D is polynomial and 2^n is exponential.

3.3.5 Reuse and amortization

Packet compilation can be expensive. Repeated parameter sweeps, time stepping, variational optimization, and repeated likelihood evaluation benefit from cached basis maps and gate plans. A one-shot simulation can be slower even when each subsequent packet update is much cheaper. The benchmark chapter preserves both one-shot and repeated-application measurements for this reason.

> **Scope boundary**
>
> A polynomial-dimensional invariant sector does not guarantee a practical speedup at small n. It guarantees that the state-vector representation within that sector has polynomial asymptotic size under the stated access assumptions.

3.4 Exact packet simulation and polynomial collapse

3.4.1 Gatewise invariance

Let P be an orthogonal projector and suppose the initial state obeys $P|\psi_0\rangle = |\psi_0\rangle$. If every gate satisfies

$$G_j P = P G_j P, \tag{3.3}$$

then G_j maps $P\mathcal{H}$ into itself. For a general linear map, this equation asserts only invariance of the selected subspace. For the finite-dimensional unitary gates considered here, invariance also implies invariance of the orthogonal complement, so it is equivalent to commutation with P.

Theorem 3.6 (Invariant-basis simulation). *Let $U = G_m \cdots G_1$ and assume Equation (3.3) for every gate. If $P|\psi_0\rangle = |\psi_0\rangle$, then*

$$U|\psi_0\rangle = (PG_mP)\cdots(PG_1P)|\psi_0\rangle.$$

In any basis of $P\mathcal{H}$, the restricted circuit gives exactly the same amplitudes after embedding.

Proof. Inductively, if $P|\psi_{j-1}\rangle = |\psi_{j-1}\rangle$, then

$$G_j|\psi_{j-1}\rangle = G_jP|\psi_{j-1}\rangle = PG_jP|\psi_{j-1}\rangle,$$

so the next state remains in $P\mathcal{H}$. Iterating proves the identity. □

3.4.2 Polynomial packet-simulation theorem

Theorem 3.7 (Polynomial packet-simulation theorem). *Let $\{C_x\}$ be a polynomial-time uniform family of quantum circuits. Suppose there is an efficiently computable packet representation such that:*

1. *the initial state lies in the represented packet and its coordinates are computable in polynomial time;*

2. *every gate preserves the representation;*
3. *the representation size is at most* $\mathrm{poly}(|x|)$*;*
4. *each gate update is computable to polynomial precision in polynomial time;*
5. *the final acceptance probability is computable from the representation in polynomial time.*

Then the language decided by the circuit family belongs to P*.*

Proof. The simulator stores a polynomial-size classical description of the evolving state. It performs polynomially many polynomial-time gate updates and evaluates the final measurement to inverse-polynomial precision. Choosing an error smaller than the amplified promise gap yields a deterministic polynomial-time decision algorithm. □

Proved statement

The theorem is a genuine quantum-to-classical complexity collapse for the declared subclass. It does not assume that a full 2^n state vector is ever constructed.

3.4.3 Dimension-only corollary

A convenient special case uses an explicit basis of a packet of dimension D_n.

Corollary 3.8 (Polynomial-dimensional invariant packet)**.** *If* $D_n \leq \mathrm{poly}(n)$*, the packet basis is efficiently indexed, every gate has a polynomial-time sparse or structured action in that basis, and the measurement is efficiently evaluable, then the circuit family is classically polynomial-time simulable.*

3.4.4 A broader compressed representation

The state description need not be a vector of packet amplitudes. Let $\mathcal{R}_n$ be a polynomial-size classical representation class with an embedding into the packet Hilbert space. If initialization, gate updates, and measurement stay within $\mathcal{R}_n$ and are efficient, the same proof applies. This observation will later combine packet labels with stabilizer, Gaussian, or tensor-network coordinates.

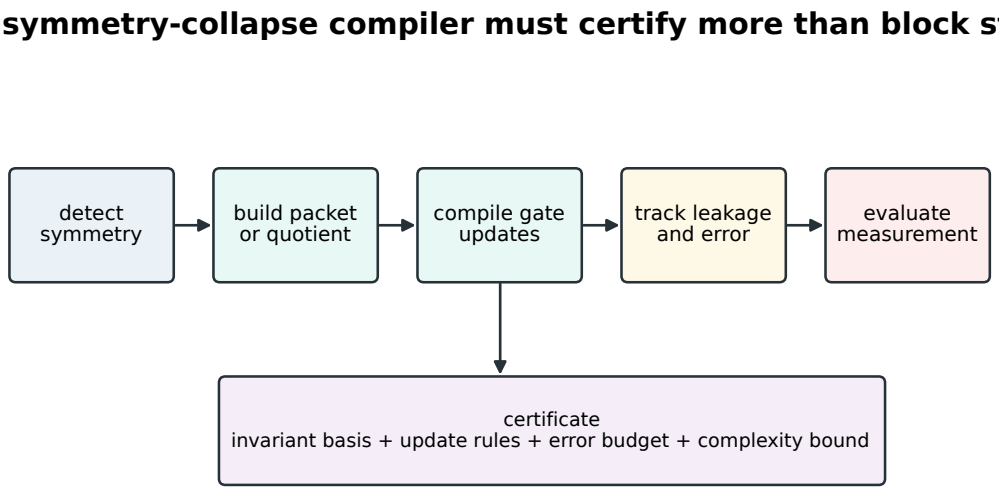


Figure 3.2. A complete collapse certificate includes discovery, update, leakage, measurement, and complexity information.

3.5 Approximate symmetry and leakage certificates

3.5.1 Hamiltonian leakage

Exact conservation is not the only useful regime. Let P be an orthogonal projector and let

$$U(t) = e^{-itH}$$

for a bounded self-adjoint Hamiltonian H. The leakage operator from the packet is

$$L_P(t) = (I - P)e^{-itH}P.$$

Theorem 3.9 (Commutator leakage bound). *For every real t,*

$$\|(I - P)e^{-itH}P\| \le |t|\,\|[H, P]\|.$$

Proof. Define

$$F(s) = e^{-i(t-s)H}Pe^{-isH}P.$$

Then $F(0) = e^{-itH}P$ and $F(t) = Pe^{-itH}P$. Differentiation gives

$$F'(s) = ie^{-i(t-s)H}[H, P]e^{-isH}P.$$

Therefore

$$(I - P)e^{-itH}P = -(I - P)\int_0^t F'(s)\,ds.$$

The unitary factors and projectors have norm at most one, so the integral is bounded by $|t|\|[H, P]\|$. □

3.5.2 Circuit leakage

For a gate G_j, define

$$\delta_j = \|(I - P)G_jP\|.$$

Let $V_j = PG_jP$. A telescoping expansion gives

$$G_m \cdots G_1P - V_m \cdots V_1P = \sum_{k=1}^{m} G_m \cdots G_{k+1}(G_k - V_k)V_{k-1} \cdots V_1P.$$

Taking norms yields the following.

Proposition 3.10 (Sum-of-leakages bound). *For an initial state in $P\mathcal{H}$,*

$$\|G_m \cdots G_1|\psi_0\rangle - V_m \cdots V_1|\psi_0\rangle\| \le \sum_{j=1}^{m} \delta_j.$$

This is conservative because leakage contributions may interfere destructively. Its value is that it is composable and can be checked gate by gate.

3.5.3 Decision-problem consequence

If the packet representation is polynomial, the restricted updates are efficient, and

$$\sum_j \delta_j \le \frac{1}{\mathrm{poly}(n)},$$

then the approximate packet simulator reproduces the final acceptance probability to inverse-polynomial accuracy. After ordinary promise-gap amplification, this suffices for a deterministic polynomial-time decision algorithm.

3.5.4 Adaptive packet enlargement

A practical simulator need not fix one packet forever. It can maintain an active set A_t of packets, estimate transfers $P_\alpha G_t P_\beta$, and add a packet when its predicted mass exceeds a tolerance. The proof obligation becomes a global accounting identity:

$$\text{total error} \le \text{discarded transfer} + \text{numerical error} + \text{measurement approximation}.$$

The key research problem is to retain polynomial active size while certifying the tail.

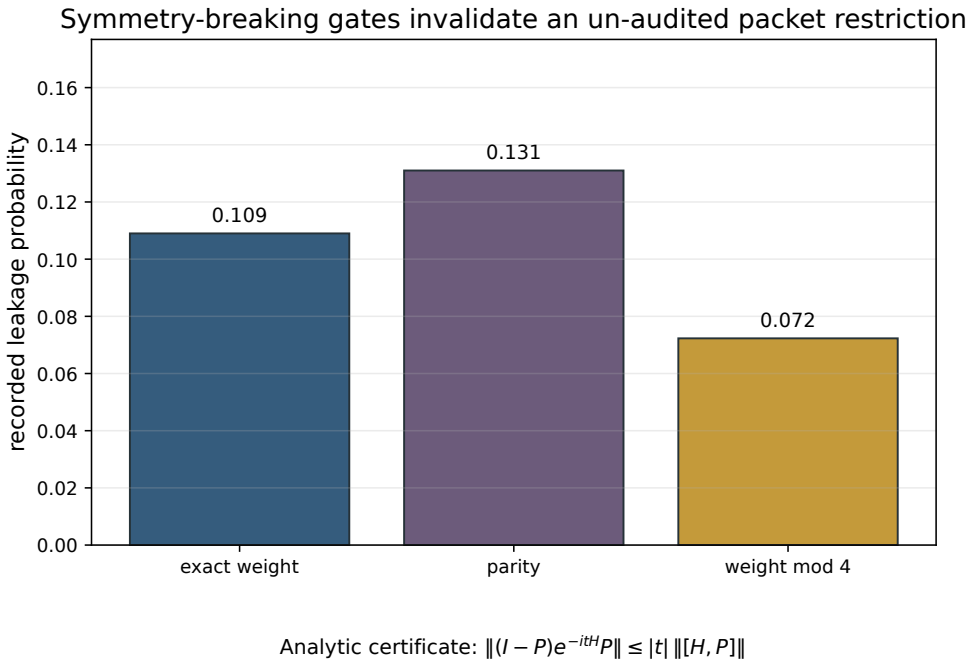


Figure 3.3. Recorded leakage examples and the analytic commutator certificate.

3.6 Moving packets, dynamic frames, and sparse transfer graphs

3.6.1 Why one fixed projector is not the end of the theory

The exact packet theorem in Section 3.4 follows a state inside one fixed subspace. That is sufficient for a conserved charge, but it is not necessary for classical collapse. A circuit may move through a

sequence of small subspaces, or it may preserve a small packet only after a time-dependent change of coordinates. The correct object is therefore a *packet schedule*, not necessarily one static projector.

Definition 3.11 (Packet schedule)**.** For a circuit

$$U = G_m \cdots G_1$$

on a Hilbert space $\mathcal{H}$, a packet schedule is a sequence of orthogonal projectors

$$P_0, P_1, \ldots, P_m.$$

It is exact when

$$G_t P_{t-1} = P_t G_t P_{t-1} \qquad (1 \le t \le m). \tag{3.4}$$

The condition says that gate G_t maps the currently active packet into the next active packet. The dimensions may vary with time.

3.6.2 Exact dynamic-packet simulation

Choose isometries

$$B_t : \mathbb{C}^{D_t} \longrightarrow \mathcal{H}, \qquad B_t B_t^* = P_t, \qquad B_t^* B_t = I_{D_t},$$

where $D_t = \operatorname{rank} P_t$. Define the rectangular coordinate update

$$M_t = B_t^* G_t B_{t-1}. \tag{3.5}$$

Theorem 3.12 (Exact dynamic-packet theorem)**.** *Suppose $P_0|\psi_0\rangle = |\psi_0\rangle$ and Equation (3.4) holds for every gate. Put*

$$z_0 = B_0^*|\psi_0\rangle, \qquad z_t = M_t z_{t-1}.$$

Then

$$G_t \cdots G_1 |\psi_0\rangle = B_t z_t$$

for every t.

Proof. The claim is true at $t = 0$. If the state after $t-1$ gates is $B_{t-1}z_{t-1}$, then

$$G_t B_{t-1} z_{t-1} = P_t G_t P_{t-1} B_{t-1} z_{t-1} = B_t B_t^* G_t B_{t-1} z_{t-1} = B_t M_t z_{t-1}.$$

This is $B_t z_t$. □

Theorem 3.13 (Polynomial moving-packet collapse)**.** *Let $\{C_x\}$ be a polynomial-time uniform circuit family. Suppose an exact packet schedule can be produced from x in polynomial time such that:*

1. $\max_t D_t \le \operatorname{poly}(|x|)$*;*
2. *the coordinate description of the initial state is computable in polynomial time;*
3. *each rectangular update M_t can be applied to polynomial precision in polynomial time;*
4. *the final acceptance probability is computable from z_m in polynomial time.*

Then the language decided by the circuit family belongs to P.

Proof. The simulator stores at most polynomially many coordinates at every step and applies polynomially many polynomial-time updates. Theorem 3.12 proves exactness before numerical rounding. Polynomial precision below the amplified promise gap gives a deterministic polynomial-time decision procedure. □

Derived theorem or corollary

A circuit does not need one globally conserved sector. It is enough that the state follow an efficiently discoverable path through polynomial-dimensional sectors.

3.6.3 Moving frames and hidden symmetry

The packet schedule has a useful coordinate form. Let P be one fixed projector and let W_t be unitaries with

$$P_t = W_t P W_t^*.$$

In the moving frame, the transformed gate is

$$\widetilde{G}_t = W_t^* G_t W_{t-1}. \tag{3.6}$$

The schedule condition is equivalent to

$$\widetilde{G}_t P = P \widetilde{G}_t P.$$

Corollary 3.14 (Moving-frame collapse)**.** *Suppose a polynomial-time uniform circuit admits efficiently described frames W_t for which the transformed gates in Equation (3.6) preserve one fixed polynomial-dimensional packet, the initial state has efficiently computable coordinates in $W_0 P\mathcal{H}$, and their restricted matrices and the transformed final measurement are efficiently computable. Then the circuit family is classically simulable in polynomial time.*

This is the appropriate theorem for a symmetry that changes orientation with time. Examples to test include Clifford frames, Gaussian mode frames, rotating spin axes, and gauge-fixed local coordinates.

Scope boundary

The frame must be independently discoverable and efficiently usable. Choosing $W_t = G_t \cdots G_1$ makes the transformed state trivial but is circular: computing that frame is the original simulation problem. The theorem is useful only when the frames have a compact classical description and their restricted actions can be computed without constructing the full state.

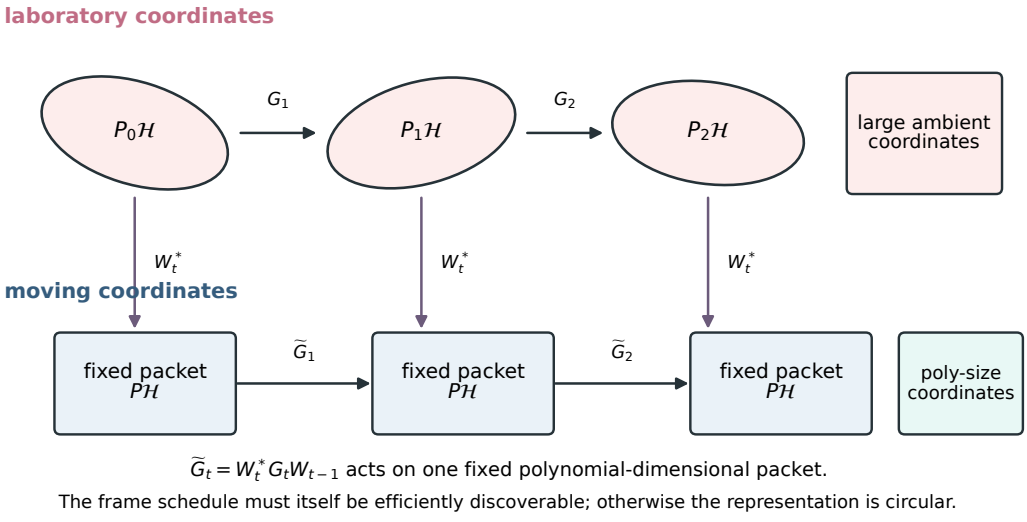


Figure 3.4. A circuit may fail to preserve one laboratory-frame packet while remaining inside a small packet in a computable moving frame.

3.6.4 Sparse packet-transfer graphs

Let

$$I = \sum_{\alpha \in \Lambda} P_\alpha$$

be an orthogonal packet decomposition. Write the gate blocks as

$$G_{t;\beta\alpha} = P_\beta G_t P_\alpha.$$

Starting from an active packet set S_0, define recursively

$$S_t = \{\beta : \exists \alpha \in S_{t-1} \text{ with } G_{t;\beta\alpha} \neq 0\}. \tag{3.7}$$

Corollary 3.15 (Reachable packet-graph collapse)**.** *Suppose the packet sets S_t and nonzero transfer blocks are computable in polynomial time, and*

$$D_t = \sum_{\alpha \in S_t} \operatorname{rank} P_\alpha \leq \operatorname{poly}(n)$$

for every t. If all active block updates and the final measurement are efficiently computable, then the circuit family is classically polynomial-time simulable.

Proof. Take

$$P_t = \sum_{\alpha \in S_t} P_\alpha.$$

The definition of S_t gives $G_t P_{t-1} = P_t G_t P_{t-1}$. Apply Theorem 3.13. □

This theorem permits exact sector changes. What matters is not that one charge be fixed, but that the union of reachable sectors remain polynomial.

Corollary 3.16 (Bounded-excitation collapse). *Consider n qubits and suppose the initial state and every circuit prefix are certified to lie in*

$$\mathcal{H}_{\le K} = \bigoplus_{k=0}^{K} \mathcal{H}_{n,k}$$

for a fixed constant K. If initialization, the gate actions, and the measurement are computable in polynomial time in the occupation basis, then the circuit family lies in P, even when individual gates change excitation number.

Proof. The active dimension is

$$\sum_{k=0}^{K} \binom{n}{k} = O(n^K).$$

Let P project onto $\mathcal{H}_{\le K}$. By the hypothesis on the actual circuit prefixes, $P\psi_t = \psi_t$ at every step. Consequently $PG_tP\psi_{t-1} = G_t\psi_{t-1}$ on the evolving state, even if the gate does not preserve all of $P\mathcal{H}$. The restricted update therefore reproduces that trajectory exactly. The hypothesis supplies efficient initialization, updates, and measurement in these polynomially many coordinates. □

For example, a fixed-weight input followed by only a constant number of pair-creation or pair-annihilation events remains in a constant collection of low-weight sectors. Exact number conservation is therefore stronger than necessary.

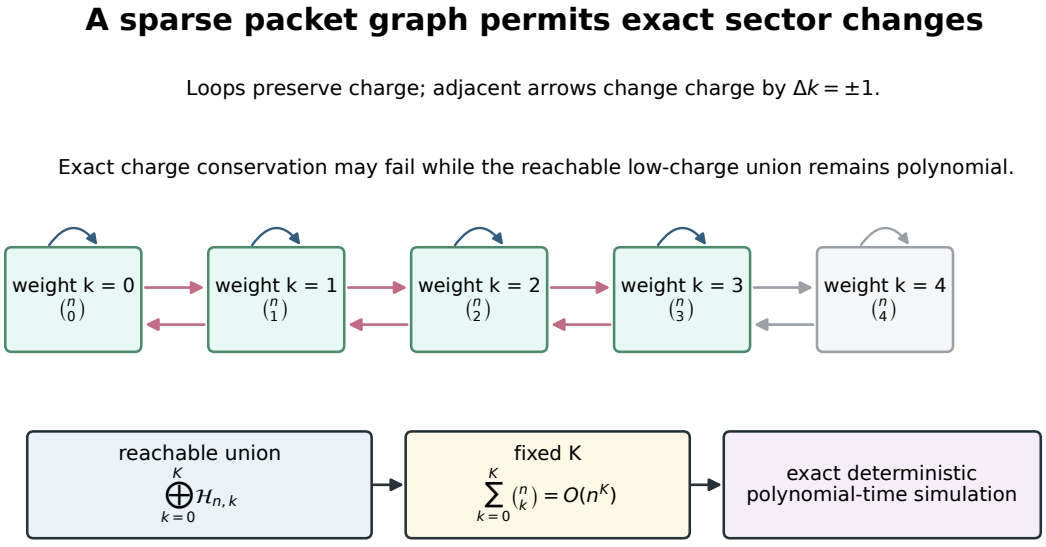


Figure 3.5. Sector-changing gates remain classically tractable when the reachable packet graph and total active dimension stay polynomial.

3.6.5 Approximate dynamic packets

Define the one-step leakage from the scheduled packet by

$$\delta_t = \|(I - P_t)G_tP_{t-1}\|.$$

Let

$$\widehat{G}_t = P_tG_tP_{t-1}.$$

Theorem 3.17 (Dynamic sum-of-leakages bound). *For an initial unit vector in $P_0\mathcal{H}$,*

$$\left\| G_m \cdots G_1 |\psi_0\rangle - \widehat{G}_m \cdots \widehat{G}_1 |\psi_0\rangle \right\| \leq \sum_{t=1}^{m} \delta_t.$$

For any acceptance operator $0 \leq M \leq I$, the corresponding acceptance probabilities differ by at most

$$2 \sum_{t=1}^{m} \delta_t.$$

Proof. A telescoping expansion inserts $G_t - \widehat{G}_t$ at one time. On $P_{t-1}\mathcal{H}$ this difference equals $(I - P_t)G_t P_{t-1}$, while all other unitary or projected factors have norm at most one. Summing the operator norms proves the state bound. If ψ and ϕ have norm at most one, then

$$|\langle \psi, M\psi \rangle - \langle \phi, M\phi \rangle| \leq (\|\psi\| + \|\phi\|)\|\psi - \phi\| \leq 2\|\psi - \phi\|.$$

□

Corollary 3.18 (Approximate moving-packet collapse). *If the scheduled dimensions and updates are polynomial and*

$$\sum_t \delta_t \leq \frac{1}{\mathrm{poly}(n)},$$

then, after ordinary promise-gap amplification, the circuit family remains deterministically classically simulable in polynomial time.

The result is conservative but fully composable. It extends the fixed-projector estimate of Section 3.5 to moving frames and changing packet sets.

3.6.6 An adaptive certificate from discarded state vectors

The operator-norm leakage certificate is uniform over every input in a packet. An adaptive simulator can use a less expensive certificate that follows its actual approximate state. The discarded vector must be accounted for before any normalization.

Theorem 3.19 (Adaptive discarded-state certificate). *Let $G_1, \ldots, G_m$ be unitary operators on a finite-dimensional Hilbert space, and let $\psi_0 = \phi_0$ be a unit vector. At each step choose an orthogonal projector P_t, possibly as a function of the previously computed approximate states. Define*

$$\psi_t = G_t \psi_{t-1}, \qquad \phi_t = P_t G_t \phi_{t-1}, \qquad \eta_t = \|(I - P_t)G_t\phi_{t-1}\|.$$

Then

$$\|\psi_m - \phi_m\| \leq \sum_{t=1}^{m} \eta_t.$$

For an effect $0 \leq M \leq I$,

$$|\langle \psi_m, M\psi_m \rangle - \langle \phi_m, M\phi_m \rangle| \leq 2 \sum_{t=1}^{m} \eta_t.$$

No renormalization of ϕ_t is performed. The second quadratic form is an unnormalized acceptance estimate.

Proof. Fix the projectors chosen during the particular run. Put $d_t = (I - P_t)G_t\phi_{t-1}$. Subtraction gives the exact recurrence

$$\psi_t - \phi_t = G_t(\psi_{t-1} - \phi_{t-1}) + d_t.$$

Since $\psi_0 - \phi_0 = 0$, iteration yields

$$\psi_m - \phi_m = \sum_{t=1}^{m} G_m \cdots G_{t+1} d_t.$$

Every unitary product preserves norm, so the triangle inequality proves the state estimate. Also $\|\psi_m\| = 1$ and $\|\phi_m\| \le 1$ because each projector is contractive. Expanding the difference of the two quadratic forms and using $\|M\| \le 1$ gives the factor two. This proof applies to the chosen run even when the projectors were selected adaptively. □

Example 3.20 (A rational two-step certificate)**.** Take

$$G = \begin{pmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{pmatrix}, \qquad P = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \psi_0 = \phi_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Using $G_1 = G_2 = G$ and $P_1 = P_2 = P$ gives

$$\phi_2 = \begin{pmatrix} 9/25 \\ 0 \end{pmatrix}, \qquad \psi_2 = \begin{pmatrix} -7/25 \\ 24/25 \end{pmatrix}, \qquad \eta_1 = 4/5, \quad \eta_2 = 12/25.$$

The squared state error is $832/625$, below $(\eta_1 + \eta_2)^2 = 1024/625$. The rational validation script checks these values exactly. The deliberately large tail illustrates a valid but loose certificate, not an efficient approximation.

The theorem closes global error accounting for any adaptive projection run whose discarded vectors or certified norm bounds can be computed. It does not prove that the retained representation stays polynomial, that the discarded norm is cheap to evaluate, or that a small tail can always be found. Those are separate algorithmic obligations. This is a written theorem with exact finite controls, not an additional Lean formalization.

CHAPTER 4

Exact symmetry families

4.1 Fixed charge and exact-number circuits

4.1.1 Excitation-number packets

For n qubits, define

$$\widehat{N} = \sum_{q=1}^{n} \frac{I - Z_q}{2}.$$

Its eigenvalues are $k = 0, 1, \ldots, n$. The projector onto Hamming weight k is

$$P_{n,k} = \sum_{|x|=k} |x\rangle\langle x|,$$

and

$$\dim P_{n,k}\mathcal{H} = \binom{n}{k}.$$

The decomposition

$$(\mathbb{C}^2)^{\otimes n} = \bigoplus_{k=0}^{n} \mathcal{H}_{n,k}$$

recovers the full dimension through

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n.$$

4.1.2 Number-preserving gates

Single-qubit R_Z gates commute with $\widehat{N}$. A two-qubit exchange rotation acts in the ordered basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ as

$$U_{XY}(\theta, \phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -ie^{-i\phi}\sin\theta & 0 \\ 0 & -ie^{i\phi}\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

It rotates $|01\rangle$ and $|10\rangle$ while fixing $|00\rangle$ and $|11\rangle$, so it preserves exact Hamming weight.

Corollary 4.1 (Fixed-charge polynomial simulation)**.** *For every fixed k, a polynomial-time uniform family of number-preserving circuits initialized in $\mathcal{H}_{n,k}$ and measured by an efficiently computable observable is classically simulable in polynomial time, provided the initial weight-k coordinates can be computed efficiently and each restricted local gate is efficiently applied in that basis.*

Proof. For fixed k,

$$\dim \mathcal{H}_{n,k} = \binom{n}{k} = O(n^k).$$

The weight-k basis is efficiently indexed by k-element subsets. A local number-preserving gate changes only basis states containing the affected occupation pattern. Apply Theorem 3.7. □

4.1.3 Sparse update structure

A one-qubit diagonal phase changes one amplitude at a time. An exchange gate on modes p, q couples a basis state only to the state obtained by moving one excitation between those modes. Thus every column of the restricted gate has at most two nonzero entries. For fixed k, the total update cost is polynomial in n.

4.1.4 Uniformity and input encoding

The fixed-charge corollary assumes k is a fixed constant independent of n. It also applies when $k = O(1)$ is supplied by the circuit family. If k is written in binary and allowed to scale as $n/2$, the sector can become exponential. The theorem is about the dimension and update cost of the actual active sector, not the name of the symmetry.

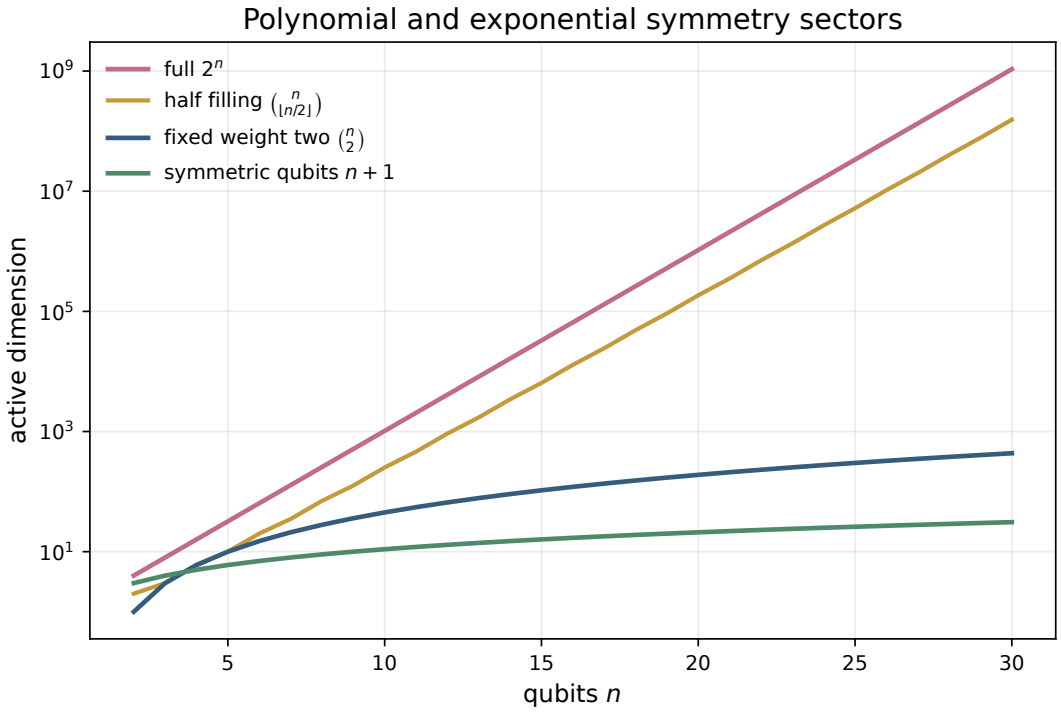


Figure 4.1. Fixed weight and permutation symmetry can be polynomial, while half filling remains exponential.

4.1.5 Observables

A diagonal observable depending on occupations is immediate. Few-body number-preserving observables are sparse in the subset basis. More global observables may require additional structure. The measurement hypothesis in Theorem 3.7 cannot be omitted: a compactly evolved state does not help if the requested output itself hides exponential work.

4.2 The exact weight-two benchmark

4.2.1 Purpose and status

The benchmark record compares a full state-vector simulator with a simulator restricted to the exact weight-two packet. Both use the same local gate arithmetic. The benchmark is not a hardware claim and not a universal performance model. It shows that the exact dimension reduction can translate into large memory and repeated-application gains in an ordinary implementation.

Recorded computation

At $n = 20$, the full state has $1{,}048{,}576$ amplitudes and the exact weight-two packet has 190. The archived repeated-application gain is 958.8, the one-shot gain is 54.8, and the amplitude-memory ratio is 5518.8. The recorded relative reconstruction error is zero at the displayed precision.

4.2.2 Recorded table

Table 4.1. Recorded exact-weight-two benchmark.

n	full dim.	packet dim.	apply gain	one-shot gain	memory gain	rel. error
8	256	28	1.11	1.64	9.14	0
10	1,024	45	1.51	2.02	22.8	0
12	4,096	66	2.95	3.98	62.1	0
14	16,384	91	8.55	11.0	180.0	0
16	65,536	120	33.0	36.9	546.1	0
18	262,144	153	131.3	39.1	1,713.4	0
20	1,048,576	190	958.8	54.8	5,518.8	0

4.2.3 What is machine independent

The dimension identity

$$\frac{2^n}{\binom{n}{2}}$$

is independent of implementation. It predicts an exponential memory advantage for fixed $k = 2$. The measured runtime ratio need not equal the dimension ratio. Packet indexing, compilation, sparse update kernels, allocation, and cache behavior determine the constant factors.

4.2.4 Repeated application versus one shot

Let C_{compile} be the cost of constructing the packet basis and restricted gate plans, and let C_{apply} be one packet application. Reusing the compiled plan R times costs

$$C_{\text{packet}}(R) = C_{\text{compile}} + RC_{\text{apply}}.$$

If the full path costs RC_{full}, the packet method breaks even when

$$R > \frac{C_{\text{compile}}}{C_{\text{full}} - C_{\text{apply}}}.$$

This is why time evolution, parameter sweeps, variational loops, and repeated likelihood evaluations are natural applications.

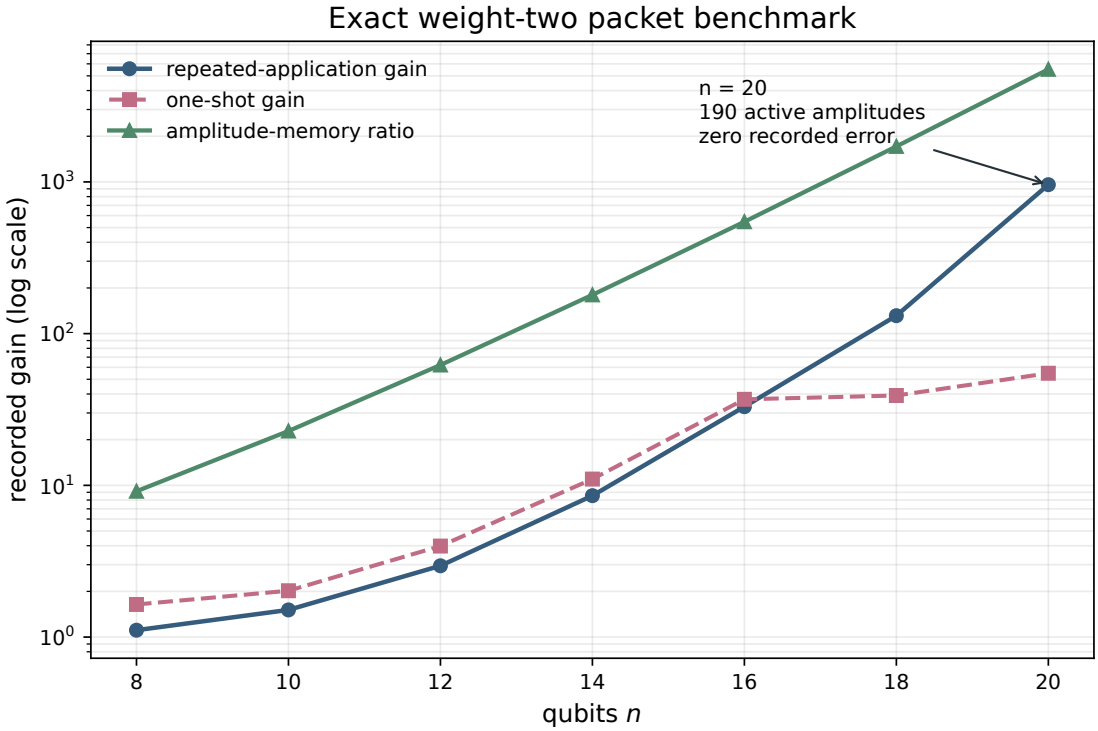


Figure 4.2. Recorded runtime and memory gains for the exact-weight-two packet.

4.2.5 Independent reconstruction check

The release code generates a random packet state, embeds it in the full Hilbert space, applies the same R_Z and exchange gates through both arithmetic paths, and compares the final states. The representative checks at $(n, k) = (6, 1), (8, 2), (10, 2)$ produce errors near machine precision.

Scope. The recorded benchmark supports exactness and implementation value for a symmetry-preserving subclass. It does not show that generic quantum circuits admit comparable reductions.

4.3 Coarse packets and the honest limitation

4.3.1 Parity

The even- and odd-parity packets each have dimension

$$2^{n-1}.$$

Parity removes half the state vector. This can produce a useful constant factor, but it does not change exponential scaling.

A pair-creation rotation mixes $|00\rangle$ and $|11\rangle$ and therefore changes excitation number by two while preserving parity. The packet is exact for a parity-preserving circuit, but its internal dimension remains exponential.

4.3.2 Weight modulo four

A four-qubit rotation between $|0000\rangle$ and $|1111\rangle$ changes excitation number by four. It can mix exact-weight sectors while preserving weight modulo four. A single residue packet contains asymptotically one quarter of all basis states. Again, the reduction is a constant factor.

4.3.3 Half filling

At half filling,

$$\binom{n}{\lfloor n/2 \rfloor} \sim \frac{2^n}{\sqrt{\pi n/2}}.$$

The packet is smaller by a polynomial factor, but it remains exponential. At $n = 20$, the half-filled packet contains $184{,}756$ amplitudes. The archived repeated-application gain is approximately 7.396, while the one-shot gain is 0.6529, meaning the packet path was slower in that single-run test after compilation overhead.

Recorded computation

The half-filled benchmark demonstrates a necessary distinction: dimension reduction can be exact and useful without yielding polynomial complexity or a one-shot speedup.

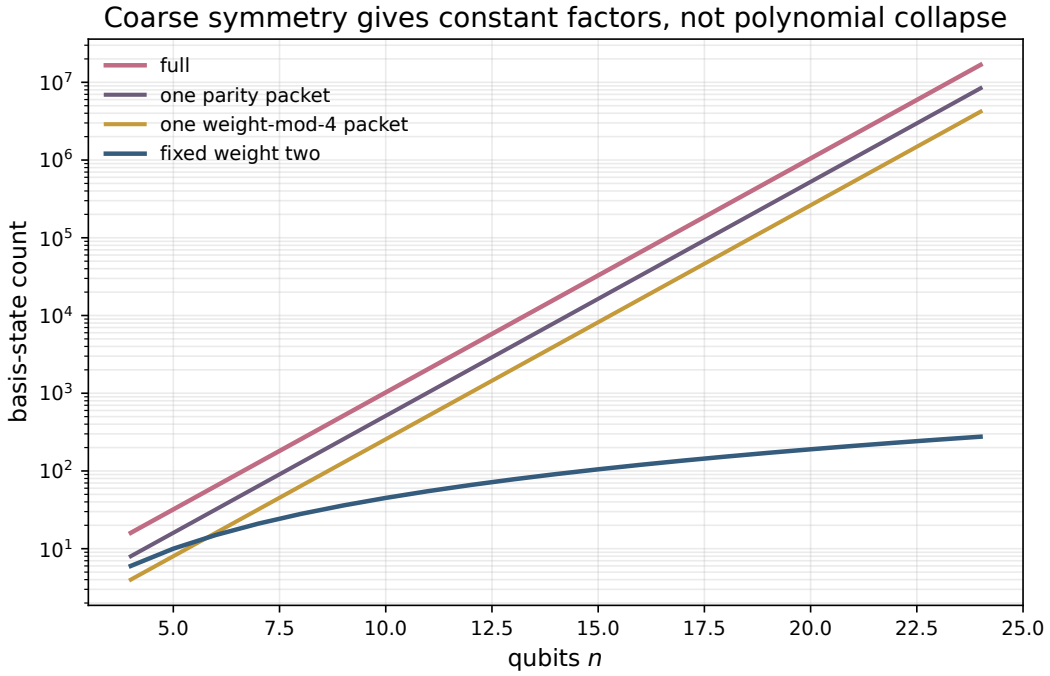


Figure 4.3. Parity and mod-four packets retain an exponential number of basis states.

4.3.4 Entanglement inside a packet

A small number of symmetry labels does not imply low entanglement. The half-filled packet can support volume-law entanglement. Tensor-network bond dimension and packet dimension are independent resources. A robust simulator should therefore track both

$$(D_{\text{packet}}, \chi_{\text{internal}}),$$

where χ_{internal} measures the cost of representing the state inside the packet.

4.3.5 The no-label shortcut

A stabilizer syndrome is only a label. A momentum quantum number is only a label. A total charge residue is only a label. The internal multiplicity space may still contain exponentially many degrees of freedom. The packet-simulation theorem becomes polynomial only when the complete active representation, not merely the label set, has polynomial classical cost.

4.4 Permutation-symmetric circuits and Dicke coordinates

4.4.1 The symmetric subspace

Let the symmetric group S_n act by permuting the n tensor factors. The fully symmetric qubit subspace is spanned by the Dicke states

$$|D_n^k\rangle = \binom{n}{k}^{-1/2} \sum_{|x|=k} |x\rangle, \qquad 0 \le k \le n.$$

Its dimension is $n+1$.

Derived theorem or corollary

If the initial state is permutation symmetric with efficiently computable Dicke coordinates, every gate preserves the fully symmetric subspace, the gate matrices in the Dicke basis are computable in polynomial time, and the final measurement is permutation invariant and efficient, then the circuit family lies in P.

Theorem 4.2 (Permutation-symmetric collapse)**.** *Under the hypotheses above, an n-qubit circuit can be simulated using an $(n+1)$-component classical state vector and polynomial-time updates.*

Proof. The projector onto the trivial S_n representation has rank $n+1$. The Dicke basis is explicitly indexed by $k = 0, \dots, n$. Apply Theorem 3.7. □

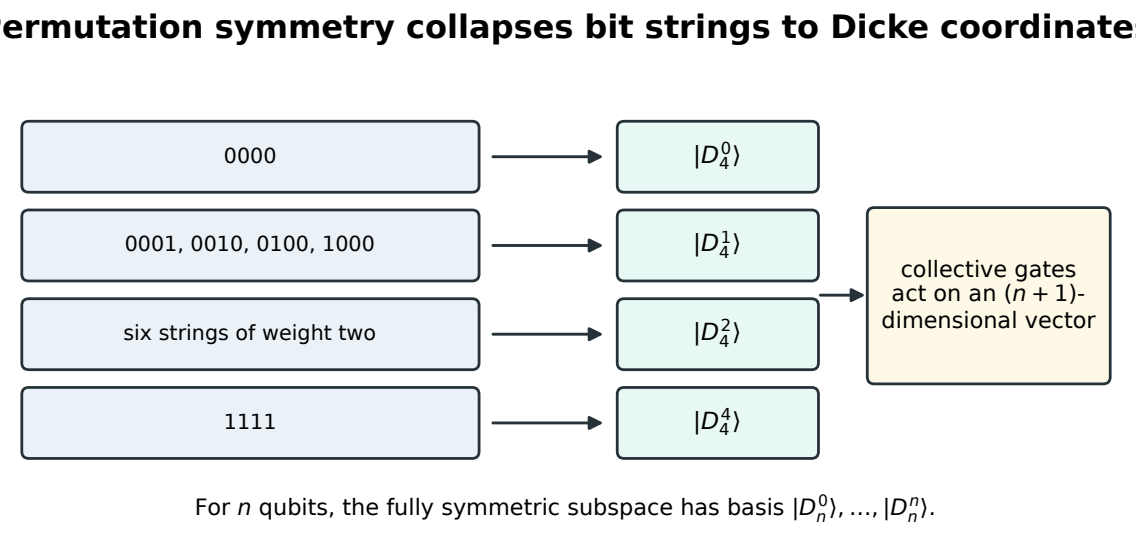


Figure 4.4. Computational-basis orbits become Dicke coordinates.

4.4.2 Collective spin operators

Define

$$J_a = \frac{1}{2}\sum_{q=1}^{n} \sigma_a^{(q)}, \qquad a \in \{x, y, z\}.$$

The operators J_a preserve the symmetric subspace. We use the computational convention $\sigma_z|0\rangle = |0\rangle$ and $\sigma_z|1\rangle = -|1\rangle$. Thus $J_z|D_n^k\rangle = (n/2 - k)|D_n^k\rangle$. For the following occupation formulas define $J_+ = J_x - iJ_y$ and $J_- = J_x + iJ_y$, so that the subscript records increasing or decreasing excitation number rather than magnetic quantum number. Then

$$J_+|D_n^k\rangle = \sqrt{(n-k)(k+1)}|D_n^{k+1}\rangle,$$
$$J_-|D_n^k\rangle = \sqrt{k(n-k+1)}|D_n^{k-1}\rangle.$$

Thus collective rotations and polynomial functions of collective spin admit banded or structured updates on $n+1$ amplitudes.

4.4.3 Symmetric qudits

For local dimension d, the symmetric subspace has occupation-number basis

$$(n_1, \ldots, n_d), \qquad n_1 + \cdots + n_d = n,$$

and dimension

$$\dim \mathrm{Sym}^n(\mathbb{C}^d) = \binom{n+d-1}{d-1}. \tag{4.1}$$

For fixed d, this is polynomial in n.

Corollary 4.3 (Fixed-local-dimension symmetric qudits). *A uniform family of fully symmetric d-level circuits is classically polynomial-time simulable for fixed d, provided its occupation-basis updates and measurement are polynomial-time computable.*

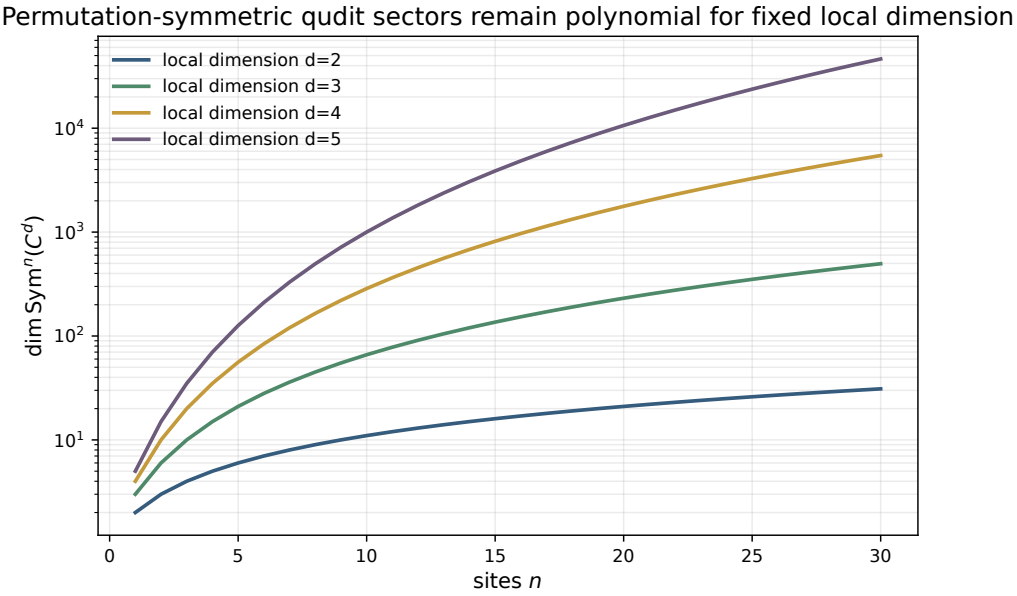


Figure 4.5. Symmetric qudit sectors remain polynomial for every fixed local dimension.

4.4.4 What full symmetry excludes

A generic two-qubit gate acting on one selected pair does not preserve the fully symmetric subspace. Symmetry is preserved by collective gates or by circuits arranged so that the total operation commutes with all permutations. A gatewise packet simulator requires gatewise invariance, or else it must combine several elementary gates into a symmetry-preserving layer and audit intermediate leakage.

4.5 Schur–Weyl coordinates and total-spin packets

4.5.1 Decomposition

For n qudits of local dimension d, Schur–Weyl duality decomposes

$$(\mathbb{C}^d)^{\otimes n} \cong \bigoplus_{\lambda \vdash n,\, \ell(\lambda) \le d} \mathcal{U}_\lambda \otimes \mathcal{V}_\lambda,$$

where $\mathcal{U}_\lambda$ carries an irreducible unitary-group representation and $\mathcal{V}_\lambda$ carries a symmetric-group representation. A permutation-invariant operator acts as

$$A = \bigoplus_\lambda A_\lambda \otimes I_{\mathcal{V}_\lambda}.$$

The label λ can be polynomially short even when the multiplicity space $\mathcal{V}_\lambda$ is exponentially large. This is a canonical example of the difference between a small packet label and a small active representation.

4.5.2 Qubit total spin

For qubits, the decomposition may be labeled by total spin j and a multiplicity index. A collective $SU(2)$ operation acts nontrivially only on the $(2j+1)$-dimensional spin factor and as the identity

on multiplicity. If the initial state and observable are insensitive to multiplicity, the effective dimension can be small. If arbitrary information is stored in multiplicity, total-spin labeling alone does not compress it.

Principle 4.4 (Multiplicity audit)**.** *A representation-theoretic packet must be recorded as*

$$\textit{irrep label} \quad + \textit{irrep carrier} \quad + \textit{multiplicity space}.$$

Dropping the multiplicity space is valid only when the dynamics, input, and measurement make it redundant.

4.5.3 Efficient Schur simulation regime

A Schur-coordinate simulation is polynomial when the number of active partitions is polynomial, the retained carrier and multiplicity dimensions are polynomial or compactly represented, and each gate has an efficient action in these coordinates. Fully symmetric dynamics is the one-row partition and therefore the simplest case.

4.5.4 Possible compiler use

A symmetry compiler can detect collective layers, transform them into Schur coordinates, apply small representation matrices, and transform only when required. The transformation itself must be costed. In a purely classical simulator, one may never materialize the full Schur transform if the initial state is already specified by occupation or spin coordinates.

> **Open program or boundary**
>
> A broad Schur-based collapse theorem requires explicit uniform algorithms for active-partition discovery, multiplicity handling, and local-gate updates. The decomposition alone is not the algorithm.

4.6 Permutation-invariant mixed states and commutant compression

4.6.1 Why multiplicity can disappear without being discarded

The fully symmetric subspace is only one Schur–Weyl packet. A general permutation-invariant mixed state can occupy every Schur label while still admitting a polynomial classical description for fixed local dimension. The reason is that the exponentially large symmetric-group multiplicities occur as identity factors rather than independent dynamical coordinates.

Let S_n act on

$$\mathcal{H}_{n,d} = (\mathbb{C}^d)^{\otimes n}$$

by permuting tensor factors. Schur–Weyl duality gives

$$\mathcal{H}_{n,d} \cong \bigoplus_{\lambda \vdash n,\, \ell(\lambda) \le d} \mathcal{U}_\lambda \otimes \mathcal{V}_\lambda, \tag{4.2}$$

where S_n acts trivially on the first tensor factor and irreducibly on $\mathcal{V}_\lambda$.

4.6.2 The permutation twirl

Define the twirling channel

$$\mathcal{T}_n(X) = \frac{1}{n!} \sum_{\pi \in S_n} U_\pi X U_\pi^*. \tag{4.3}$$

Its image is the commutant of the permutation action.

Lemma 4.5 (Invariant observables see only the twirl)**.** *If A commutes with every U_π, then*

$$\mathrm{Tr}(A\rho) = \mathrm{Tr}\big(A\mathcal{T}_n(\rho)\big)$$

for every density operator ρ.

Proof. Using cyclicity of the trace and $U_\pi^* A U_\pi = A$,

$$\mathrm{Tr}\big(A\mathcal{T}_n(\rho)\big) = \frac{1}{n!}\sum_\pi \mathrm{Tr}(AU_\pi \rho U_\pi^*) = \frac{1}{n!}\sum_\pi \mathrm{Tr}(U_\pi^* A U_\pi \rho) = \mathrm{Tr}(A\rho).$$

□

If a unitary U and acceptance operator M are permutation invariant, then U^*MU is invariant. Therefore the final probability depends on the input only through its twirl. This permits an observable-specific collapse even when the original pure input is not itself symmetric.

4.6.3 Exact commutant record

Let

$$u_\lambda = \dim \mathcal{U}_\lambda, \qquad f_\lambda = \dim \mathcal{V}_\lambda.$$

Schur's lemma gives the following normal form.

Lemma 4.6 (Permutation-invariant state normal form)**.** *A density operator ρ commutes with every permutation if and only if it can be written as*

$$\rho = \bigoplus_\lambda p_\lambda\, \rho_\lambda \otimes \frac{I_{\mathcal{V}_\lambda}}{f_\lambda}, \tag{4.4}$$

where $p_\lambda \geq 0$, $\sum_\lambda p_\lambda = 1$, and each ρ_λ is a density operator on $\mathcal{U}_\lambda$ whenever $p_\lambda > 0$. The weights p_λ and the unnormalized blocks $p_\lambda \rho_\lambda$ are unique. The normalized block ρ_λ is unique only when $p_\lambda > 0$ and is omitted when $p_\lambda = 0$.

The complete state record is therefore

$$\{(p_\lambda, \rho_\lambda)\}_\lambda .$$

The multiplicity dimension f_λ affects normalization and degeneracy, but no $f_\lambda \times f_\lambda$ matrix needs to be stored.

A permutation-invariant unitary has the form

$$U = \bigoplus_\lambda U_\lambda \otimes I_{\mathcal{V}_\lambda}, \tag{4.5}$$

and a permutation-invariant measurement operator has the analogous form

$$M = \bigoplus_{\lambda} M_\lambda \otimes I_{\mathcal{V}_\lambda}.$$

The output probability is

$$\operatorname{Tr}(MU\rho U^*) = \sum_{\lambda} p_\lambda \operatorname{Tr}\big(M_\lambda U_\lambda \rho_\lambda U_\lambda^*\big). \tag{4.6}$$

4.6.4 Polynomial parameter bound

The number of partitions of n with at most d parts is bounded by the number of weak compositions:

$$\#\{\lambda : \ell(\lambda) \le d\} \le \binom{n+d-1}{d-1} = O_d(n^{d-1}).$$

The Weyl dimension formula gives

$$u_\lambda = \prod_{1 \le i < j \le d} \frac{\lambda_i - \lambda_j + j - i}{j - i} = O_d\left(n^{d(d-1)/2}\right). \tag{4.7}$$

Therefore

$$\sum_{\lambda} u_\lambda^2 = O_d\left(n^{d^2-1}\right). \tag{4.8}$$

Theorem 4.7 (Schur-commutant collapse)**.** *Fix the local dimension d. Consider a polynomial-time uniform family of permutation-invariant quantum circuits or channels on n qudits. Suppose:*

1. *the Schur-block record of the twirled input can be constructed in polynomial time;*
2. *the induced update on the invariant operator record in Equation (4.4) is computable to polynomial precision in polynomial time;*
3. *the final measurement is permutation invariant and its Schur blocks are efficiently computable.*

Then the associated decision problem belongs to P.

Proof. Permutation covariance keeps the density operator inside the commutant algebra. By Equation (4.8), that algebra has polynomial dimension for fixed d. The hypotheses supply polynomial-time initialization, updates, and measurement on this record. Apply Theorem 6.1. □

Derived theorem or corollary

For fixed local dimension, permutation-invariant mixed-state dynamics can be classically polynomial even when the state has support across exponentially large symmetric-group multiplicity spaces. Those multiplicities are exact identity factors in the invariant operator record.

4.6.5 Qubit corollary

For qubits, Schur labels are total spins j. The carrier dimension is

$$u_j = 2j + 1 \le n + 1,$$

and there are at most $\lfloor n/2 \rfloor + 1$ labels. Hence

$$\sum_j (2j+1)^2 = O(n^3). \tag{4.9}$$

Corollary 4.8 (Permutation-invariant qubit channels)**.** *A uniform permutation-covariant qubit channel family with efficiently computable total-spin block updates, efficiently available twirled input blocks, and a permutation-invariant measurement is classically simulable using $O(n^3)$ density-matrix parameters, up to constant factors and numerical precision.*

Collective Hamiltonians that are polynomial expressions in J_x, J_y, J_z are a basic test class because their total-spin matrices can be generated directly without constructing the full $2^n \times 2^n$ operator.

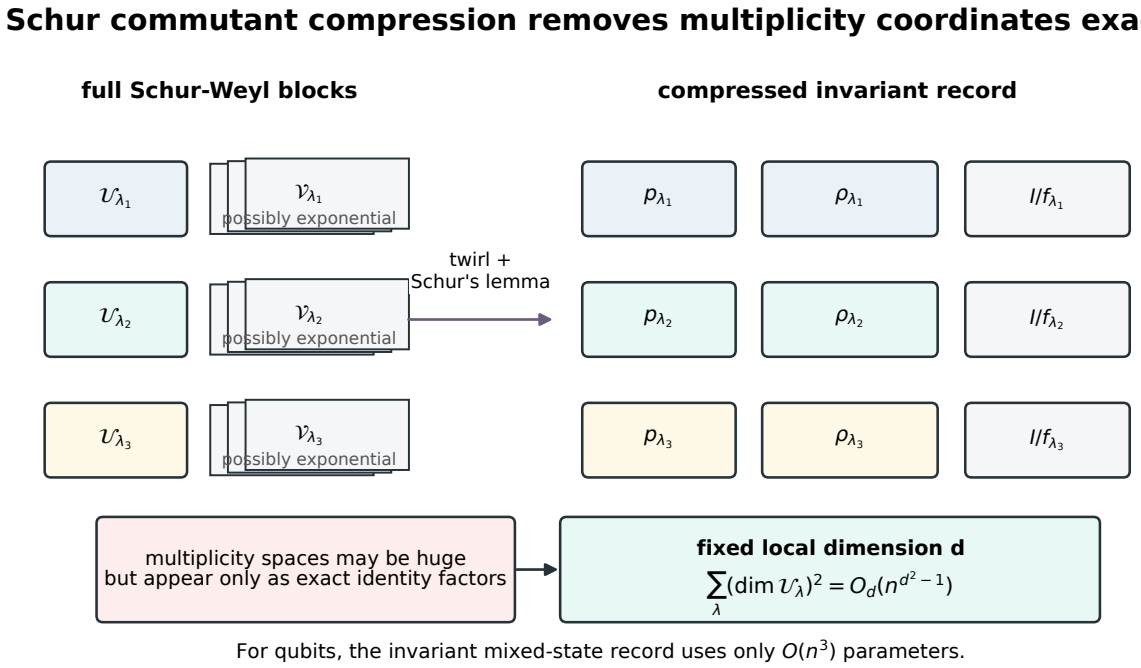


Figure 4.6. Permutation-invariant density operators retain the polynomial carrier blocks and replace each possibly exponential multiplicity space by an exact identity factor.

4.6.6 What this theorem does not cover

The theorem does not make an arbitrary local circuit easy. A gate acting on one selected pair generally fails to commute with all permutations. An arbitrary input may have Schur blocks that are difficult to compute from its succinct description. A non-invariant measurement can read information stored in multiplicity coordinates. Finally, allowing local dimension d to grow with n can make the commutant record superpolynomial.

The result is therefore a complete collapse theorem for a declared symmetry and access model, not a claim that Schur–Weyl decomposition alone solves generic many-body simulation.

4.7 Orbit-quotient quantum walks

4.7.1 Orbit states

Let a group G act by automorphisms on a finite graph with vertex set V. The orbits are $\mathcal{O}_1, \dots, \mathcal{O}_r$. Define normalized orbit states

$$|\mathcal{O}_a\rangle = \frac{1}{\sqrt{|\mathcal{O}_a|}} \sum_{x \in \mathcal{O}_a} |x\rangle.$$

If the adjacency or Hamiltonian H commutes with the group action, the span of these states is invariant whenever the partition is equitable.

The quotient matrix is

$$\widetilde{H}_{ab} = \langle \mathcal{O}_a | H | \mathcal{O}_b \rangle.$$

Theorem 4.9 (Orbit-quotient collapse)**.** *Suppose a uniformly described quantum walk has a polynomial number $r(n)$ of relevant orbits, an orbit-uniform initial state, an orbit-invariant final measurement, and polynomial-time algorithms for orbit sizes, quotient entries, initial orbit coordinates, and the restricted final measurement. For continuous-time evolution, also assume a polynomial bound on $|t|\|\widetilde{H}\|$ and efficient entry access to the precision required by the simulation. Then the walk can be simulated classically in polynomial time by the $r(n) \times r(n)$ quotient dynamics.*

Proof. The orbit-state span is invariant. The state is represented by $r(n)$ amplitudes. Polynomial-time computation of quotient entries and ordinary polynomial-precision linear algebra satisfy Theorem 3.7. □

4.7.2 Hypercube example

The n-dimensional hypercube is invariant under coordinate permutations. Its vertex orbits are Hamming-weight layers $0, 1, \dots, n$. In the normalized orbit basis, the quotient adjacency has couplings

$$\langle k+1 | A | k \rangle = \sqrt{(k+1)(n-k)}. \tag{4.10}$$

Thus a walk on 2^n vertices initialized at 0^n evolves in an $(n+1)$-dimensional subspace.

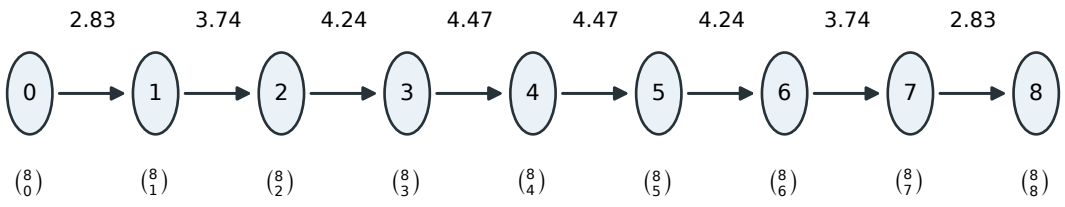


Figure 4.7. The hypercube walk collapses to a weighted path of Hamming-weight orbits.

4.7.3 The succinct-input caveat

If the full graph is supplied explicitly, a polynomial-in-$|V|$ algorithm may still be exponential in the succinct parameter n. The theorem requires quotient data computable directly from the succinct graph description. Discovering graph automorphisms or orbit partitions can itself be difficult. For regular families such as hypercubes, Johnson graphs, association schemes, and Cayley graphs with known character theory, the quotient is explicit.

4.7.4 Fourier packets

On a cycle, translation characters diagonalize the adjacency operator. The continuous-time walk

$$|\psi(t)\rangle = e^{-itA}|0\rangle$$

is a sum of Fourier-character packets. This is the same discrete Fourier transform that appears as a signal-processing coordinate change, a group-character decomposition, and a quantum basis transformation.

4.8 Few bosons and few fermions

4.8.1 Bosonic occupation space

For N indistinguishable bosons in m modes, a Fock basis state is an occupation vector

$$(n_1, \dots, n_m), \qquad n_1 + \cdots + n_m = N.$$

The dimension is

$$D_{\text{boson}}(m, N) = \binom{N+m-1}{N}. \tag{4.11}$$

For fixed N, this is $O(m^N)$.

4.8.2 Fermionic occupation space

For N fermions in m modes, each mode is occupied at most once and

$$D_{\text{fermion}}(m, N) = \binom{m}{N}. \tag{4.12}$$

For fixed N, this is also polynomial in m.

Corollary 4.10 (Few-particle collapse)**.** *A polynomial-size number-preserving bosonic or fermionic circuit with a fixed number of particles is classically polynomial-time simulable if its initial occupation coordinates, mode-gate updates, and final observable are computable in polynomial time in the occupation basis.*

Proof. Use Equations (4.11) and (4.12) and Theorem 3.7. □

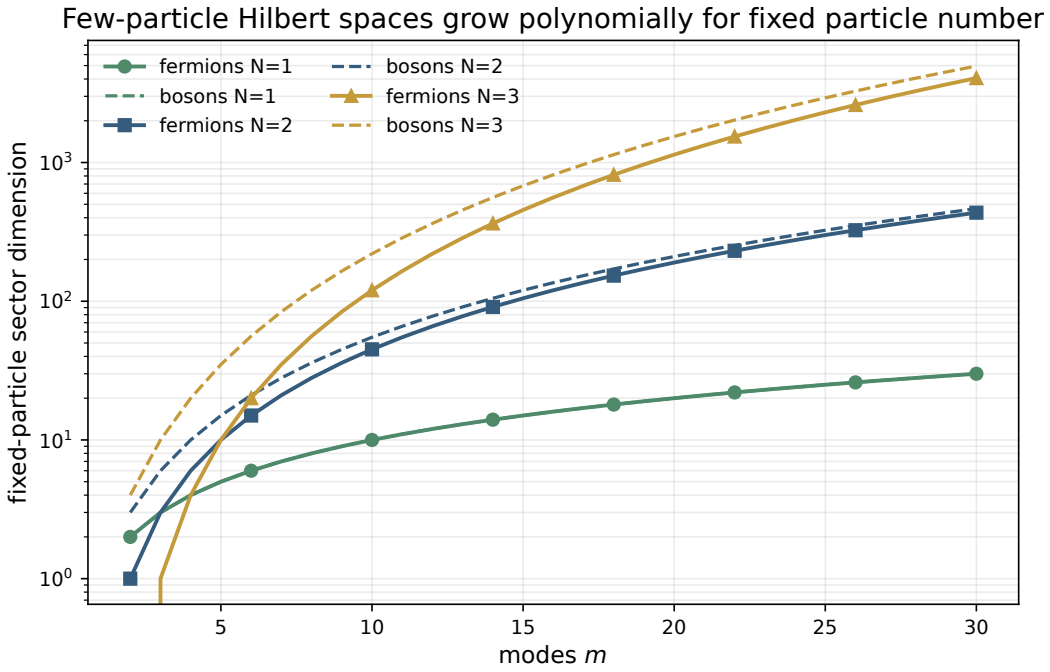


Figure 4.8. Fixed-particle bosonic and fermionic sectors grow polynomially with the number of modes.

4.8.3 Gate updates

A number-preserving one-body unitary acts on mode creation operators by

$$a_i^\dagger \longmapsto \sum_j U_{ji} a_j^\dagger.$$

A generic dense mode transformation can induce many occupation-basis transitions, but for fixed N the basis dimension remains polynomial. Local hopping and phase gates are sparse. Few-body interactions remain polynomial to apply for fixed N, although the exponent can grow with N.

4.8.4 Where the theorem stops

If N scales linearly with m, both dimensions become exponential. The fixed-number symmetry still reduces the full Fock space, but it does not prove polynomial simulation. Boson-sampling-type interference also shows that a fixed total particle number is not enough when the particle count itself scales with input size and the output asks for detailed many-body amplitudes.

> **Scope boundary**
>
> The corollary is a fixed-particle theorem. It is not a claim that general many-boson or many-fermion quantum dynamics are classically easy.

4.9 Scaling phase diagrams for charge, particles, and local dimension

4.9.1 Exact sector size is a complexity phase variable

The statement "fixed particle number is polynomial" has a sharp extension. Let the particle number depend on the system size:

$$k = k(n).$$

The explicit-amplitude simulation cost of one fermionic or qubit weight packet is governed by

$$D_{n,k} = \binom{n}{k}.$$

For $1 \le k \le n$, the standard bounds

$$\left(\frac{n}{k}\right)^k \le \binom{n}{k} \le \left(\frac{en}{k}\right)^k \tag{4.13}$$

show that the growth changes continuously between polynomial and exponential regimes.

Because

$$\binom{n}{k} = \binom{n}{n-k},$$

write

$$r(n) = \min\{k(n), n - k(n)\}.$$

Theorem 4.11 (Explicit fixed-charge phase diagram)**.** *For explicit amplitude storage in a weight-$k(n)$ packet:*

1. *the packet dimension is polynomial in n if $r(n) = O(1)$;*
2. *the packet dimension is superpolynomial whenever $r(n) \to \infty$;*
3. *if $r(n) = O(\log n)$, then*
$$D_{n,k} = 2^{O((\log n)^2)},$$
so the explicit packet simulation is quasipolynomial;
4. *if $r(n) = O(n^\alpha)$ for a fixed $0 < \alpha < 1$, then*
$$D_{n,k} = 2^{O(n^\alpha \log n)},$$
which is subexponential;
5. *if $k(n)/n \to p \in (0, 1)$, then*
$$\log_2 D_{n,k} = nH_2(k(n)/n) + O(\log n) = nH_2(p) + o(n),$$
where H_2 is binary entropy, and the packet remains exponential.

Proof. The upper bounds follow from Equation (4.13). If $r = O(1)$, then $\binom{n}{r} = O(n^r)$. Conversely, suppose $r \to \infty$ and $r \le n/2$. If $r \le \sqrt{n}$, then

$$\log \binom{n}{r} \ge r \log(n/r) \ge \frac{r}{2} \log n,$$

whose ratio to $\log n$ diverges. If $r > \sqrt{n}$, then

$$\log \binom{n}{r} \ge r \log 2 \ge \sqrt{n} \log 2,$$

which also dominates every constant multiple of $\log n$. For the final assertion, $p_n = k(n)/n$ stays in a compact subinterval of $(0, 1)$. Stirling's formula gives

$$\log_2 \binom{n}{k(n)} = nH_2(p_n) - \frac{1}{2} \log_2\big(2\pi n p_n (1 - p_n)\big) + O(1/n).$$

The logarithmic term is $O(\log n)$, uniformly on that subinterval. Continuity of H_2 gives $nH_2(p_n) = nH_2(p) + o(n)$. Replacing p_n by its limit inside an $O(\log n)$ estimate would require a convergence rate, which is not assumed here. □

Derived theorem or corollary

For the explicit occupation-amplitude representation, "few particles" means a bounded number of particles or holes. A slowly growing charge can still improve on full state-vector scaling, but it no longer guarantees membership in P.

4.9.2 A bounded-excitation hierarchy

The union of the first K weight sectors has dimension

$$D_{\le K}(n) = \sum_{k=0}^{K} \binom{n}{k}.$$

For fixed K,

$$D_{\le K}(n) = O(n^K).$$

If $K = O(\log n)$, the same union is quasipolynomial. This refines Corollary 3.16: bounded symmetry breaking preserves polynomial simulation, while logarithmically growing excitation gives a controlled quasipolynomial regime.

4.9.3 Bosonic and fermionic analogues

For N fermions in m modes,

$$D_{\mathrm{F}}(m, N) = \binom{m}{N}.$$

For N bosons in m modes,

$$D_{\mathrm{B}}(m, N) = \binom{m + N - 1}{N}.$$

The same binomial analysis gives:

Corollary 4.12 (Few-particle scaling)**.** *With the number of modes as the input-size parameter:*

1. *fixed N gives polynomial explicit-amplitude simulation;*
2. *$N = O(\log m)$ gives quasipolynomial sector size;*
3. *$N = O(m^\alpha)$, $0 < \alpha < 1$, gives subexponential sector size;*
4. *finite-density fermionic sectors remain exponential.*

For bosons, the statements hold in the dilute regime with the same occupation-basis access assumptions.

The adjective "polynomial" here refers to the sector dimension and efficient gate action. It does not assert that every interacting Hamiltonian has efficiently computable matrix elements or that every compact description remains closed.

4.9.4 Symmetric qudits with growing local dimension

The symmetric d-level subspace has dimension

$$S(n, d) = \binom{n + d - 1}{d - 1}.$$

For fixed d, this is polynomial in n. If $d = d(n)$ grows, the same phase transition appears.

Proposition 4.13 (Growing-local-dimension boundary)**.** *In the explicit occupation basis:*

1. *fixed d gives $S(n, d) = O(n^{d-1})$;*
2. *$d = O(\log n)$ gives $S(n, d) = 2^{O((\log n)^2)}$;*
3. *unbounded d is not covered by the fixed-d polynomial-collapse theorem without an additional internal compression.*

Thus permutation symmetry removes the exponential tensor-product multiplicity only when the number of local levels is controlled, or when the occupation record itself admits a second compression.

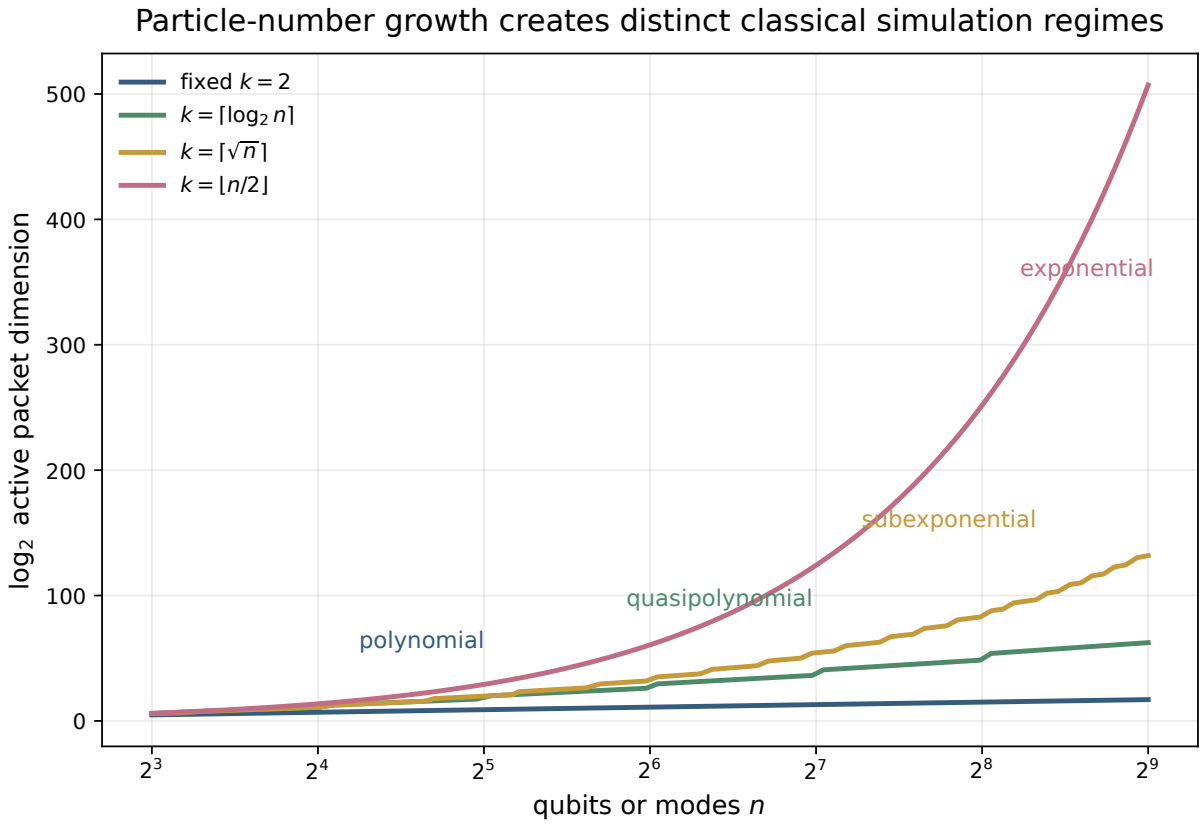


Figure 4.9. The active dimension passes through polynomial, quasipolynomial, subexponential, and exponential regimes as particle number grows.

4.9.5 Complexity class versus practical runtime

A polynomial sector can still have a large exponent. Fixed $k = 20$ gives $O(n^{20})$ explicit amplitudes, which is in P but may be impractical. Conversely, a subexponential sector may be useful at moderate size without changing the formal complexity class.

A complete performance claim should therefore report both:

asymptotic class and actual representation size and update cost.

This is why the benchmark chapters keep wall-clock evidence separate from the dimension theorems.

4.10 Stabilizer syndromes and Clifford packets

4.10.1 Syndrome projectors

Let $S_1, \dots, S_m$ be commuting Pauli stabilizers with eigenvalues ± 1. A syndrome $s \in \{0,1\}^m$ defines

$$P_s = \prod_{j=1}^{m} \frac{I + (-1)^{s_j} S_j}{2}.$$

These projectors are orthogonal and sum to the identity. Syndrome measurement selects one packet, while recovery applies a syndrome-dependent transfer toward the code packet.

4.10.2 The syndrome label is not the logical state

If the code encodes k logical qubits, a fixed syndrome packet can have dimension 2^k. The syndrome has only m bits, but arbitrary logical information remains inside the packet. Syndrome packetization therefore does not by itself prove efficient simulation.

4.10.3 Tableau compression

Clifford circuits provide a different route. A stabilizer state may occupy exponentially many computational-basis strings, yet it is represented by a polynomial-size stabilizer tableau. Clifford gates update the tableau in polynomial time, and Pauli measurements are efficiently computable. This is an example of the compressed-packet theorem rather than a small-dimensional packet theorem.

> **Derived theorem or corollary**
>
> Packet labels and internal encoding are complementary. A Clifford circuit restricted to selected syndromes may be simulated by a tableau augmented with the active syndrome graph, even when the packet dimensions are exponential.

4.10.4 Non-Clifford residues

A few non-Clifford gates can be handled by stabilizer decompositions, quasiprobability expansions, or magic-state rank methods. The cost grows with a non-Clifford resource such as stabilizer rank or robustness. Symmetry packets may reduce the number of relevant branches, but they do not eliminate the intrinsic cost of the non-Clifford residue.

4.10.5 Error-correction interpretation

The packet view separates three objects:

$$\text{syndrome label}, \qquad \text{logical state}, \qquad \text{recovery transfer}.$$

This separation is useful for compressed simulation of fault-tolerant protocols, but a full complexity statement must include repeated measurements, classical control, and the size of the logical state.

4.11 Gaussian, fermionic, and matchgate compression

4.11.1 Polynomial covariance representations

Gaussian bosonic and fermionic states are specified by first and second moments rather than by all Fock-basis amplitudes. A bosonic covariance matrix has size polynomial in the number of modes, and a fermionic Gaussian state is represented by an antisymmetric Majorana covariance matrix. Gaussian unitaries act by symplectic or orthogonal transformations:

$$\Gamma \longmapsto S\Gamma S^T \quad \text{or} \quad \Gamma \longmapsto O\Gamma O^T.$$

This is a compact internal encoding even when the corresponding Hilbert-space packet is exponentially large.

4.11.2 Hybrid symmetry and Gaussianity

Number conservation, parity, translation symmetry, and point-group symmetry can further block the covariance matrices. A simulator can therefore combine two reductions:

$$\text{symmetry packetization} + \text{Gaussian internal encoding}.$$

The first removes unreachable sectors. The second compresses states within the remaining sectors.

4.11.3 Matchgate circuits

Nearest-neighbor matchgate circuits have polynomial classical representations through fermionic Gaussian dynamics. Geometry matters: adding SWAP, fermionic SWAP, long-range gates, or a non-matchgate resource changes the simulation problem. A symmetry-collapse atlas should therefore record the gate library and connectivity as part of the model, not as implementation details.

4.11.4 Low-rank non-Gaussian corrections

One research direction is to represent a weakly non-Gaussian state as

$$\rho = \sum_{a=1}^{R} c_a \rho_a^{\text{Gauss}}$$

with small Gaussian rank R, or as a Gaussian reference plus sparse packet corrections. A polynomial collapse requires R and the update cost to remain polynomial.

> **Open program or boundary**
>
> Gaussianity is an internal representation class, not a symmetry projector. A complete hybrid theorem must prove closure of the chosen compressed class under every allowed gate and bound the growth of any non-Gaussian residue.

4.11.5 Few-particle versus Gaussian routes

Fixed-particle simulation and Gaussian simulation are independent. A fixed two-fermion state may be non-Gaussian but live in a polynomial-dimensional sector. A half-filled Gaussian state may live in an exponential number sector but have a polynomial covariance description. The two-axis model in Section 6.1 is designed to capture this distinction.

CHAPTER 5

States, measurements, and open dynamics

5.1 Density operators: populations and coherences

5.1.1 Two packet indices

A pure state uses one packet label. A density operator uses two:

$$\rho = \sum_{\alpha,\beta} P_\alpha \rho P_\beta.$$

The diagonal block $P_\alpha \rho P_\alpha$ is population inside packet α. The off-diagonal block $P_\alpha \rho P_\beta$ is coherence between packets α and β.

Discarding off-diagonal blocks is not merely a change of coordinates. It is a physical operation such as dephasing. An exact simulator must preserve all coherence blocks relevant to the final observable.

5.1.2 Hilbert–Schmidt decomposition

Orthogonality of the left-right operator packets gives

$$\|\rho\|_2^2 = \sum_{\alpha,\beta} \|P_\alpha \rho P_\beta\|_2^2.$$

This identity supplies an exact coherence-energy ledger. It can be used to decide which blocks are large, to audit dephasing, or to bound truncation errors in approximate packet simulations.

5.1.3 Symmetry-preserving unitary dynamics

If U commutes with every packet projector, then

$$P_\alpha U \rho U^\dagger P_\beta = U(P_\alpha \rho P_\beta)U^\dagger.$$

The left and right packet labels are preserved. Populations and coherences evolve independently by their declared labels.

5.1.4 Observable sparsity

For an observable M,

$$\mathrm{Tr}(M\rho) = \sum_{\alpha,\beta} \mathrm{Tr}(P_\beta M P_\alpha \, P_\alpha \rho P_\beta).$$

Only operator packets coupled by M contribute. A measurement that is diagonal in the symmetry labels depends only on populations. A phase-sensitive measurement can require off-diagonal coherences.

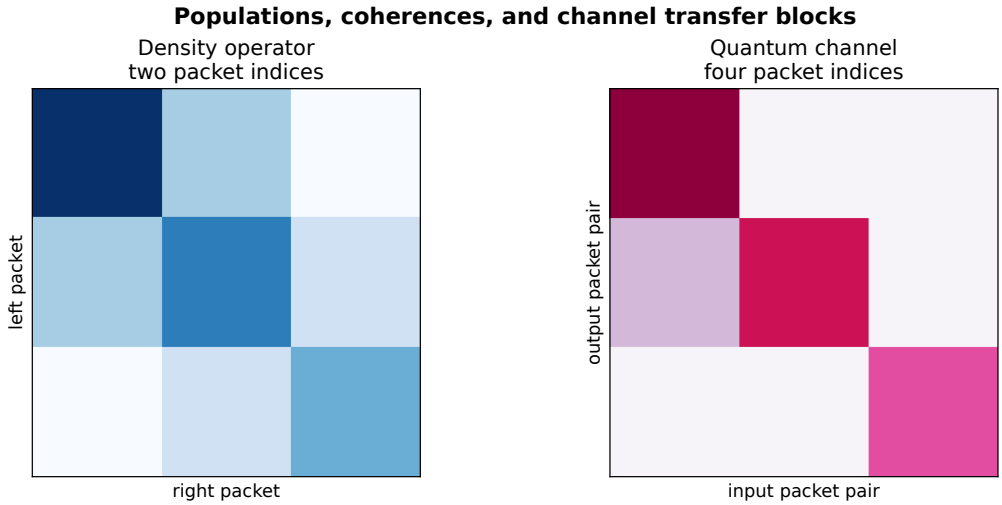


Figure 5.1. State operators carry two packet indices, while channels act between pairs of packet labels.

5.1.5 Polynomial operator-space collapse

If only polynomially many density blocks are reachable and every block has a polynomial representation, then mixed-state evolution can also be simulated in polynomial time. The representation size is not the square of the full Hilbert dimension. It is the total cost of the active left-right blocks.

> **Derived theorem or corollary**
>
> For a polynomial active set A_n of packet labels, a density-matrix simulator has at most $|A_n|^2$ label pairs. Polynomial-time simulation follows when each active coherence block has a polynomial classical representation and channel updates remain sparse and efficient.

5.2 Channels, Kraus maps, and Lindblad generators

5.2.1 Four-index transfer

A quantum channel with Kraus operators K_μ acts by

$$\Phi(\rho) = \sum_\mu K_\mu \rho K_\mu^\dagger.$$

Relative to operator packets

$$\mathcal{P}_{\alpha\beta}(X) = P_\alpha X P_\beta,$$

the channel has blocks

$$\Phi_{\gamma\delta,\alpha\beta} = \mathcal{P}_{\gamma\delta}\,\Phi\,\mathcal{P}_{\alpha\beta}.$$

Thus a channel carries four packet labels: an input coherence pair and an output coherence pair.

5.2.2 Symmetry-preserving channels

If every Kraus operator commutes with the symmetry action, the channel is diagonal in the isotypic labels. More generally, Kraus operators may carry definite charges, producing a sparse selection rule.

Amplitude damping transfers population between number sectors. Dephasing damps selected off-diagonal blocks. Particle loss moves weight-k population to weight $k-1$.

5.2.3 Lindblad dynamics

A Markovian open system obeys

$$\frac{d\rho}{dt} = \mathcal{L}(\rho) = -i[H,\rho] + \sum_a \left(L_a \rho L_a^\dagger - \frac{1}{2}\{L_a^\dagger L_a, \rho\} \right).$$

Packet selection rules of H and L_a determine a sparse directed graph on coherence pairs. If the reachable graph and each block representation remain polynomial, a classical integrator can evolve the reduced system.

5.2.4 Error accounting for truncated channels

Let Π_A project operator space onto an active set of coherence blocks. A reduced evolution uses

$$\mathcal{L}_A = \Pi_A \mathcal{L} \Pi_A.$$

The discarded transfer is

$$(I - \Pi_A)\mathcal{L}\Pi_A.$$

Duhamel's formula yields an error bound analogous to the Hamiltonian leakage theorem. The resulting estimate depends on the induced norm of the discarded superoperator transfer and the integration time.

5.2.5 Syndrome-conditioned recovery

Error correction naturally fits the four-index model. Syndrome measurement selects a population packet, classical processing chooses a recovery, and the recovery transfers the state toward the code packet. A full simulation must still retain the logical state inside each syndrome sector.

> **Open program or boundary**
>
> A useful open-system collapse theorem should state the norm, time horizon, packet-discovery algorithm, integration precision, and growth of reachable coherence pairs. Sparsity of a displayed block matrix is not enough by itself.

5.3 Measurement, conditioning, and classical feedback

5.3.1 Projective measurement

For projectors $\{P_\alpha\}$, the probability of outcome α is

$$p_\alpha = \mathrm{Tr}(P_\alpha \rho),$$

and the conditioned state is

$$\rho_\alpha = \frac{P_\alpha \rho P_\alpha}{p_\alpha}$$

when $p_\alpha > 0$. Measurement is therefore packet selection followed by normalization.

5.3.2 Branching complexity

A measurement-assisted circuit produces a tree of outcomes. Classical simulation is efficient when the branch distribution can be sampled or summed without enumerating exponentially many relevant histories. Symmetry can merge histories that lead to the same packet state, but only if the feedforward policy respects the same equivalence.

5.3.3 Dynamic programming on packet states

Suppose each branch is represented by a pair (α, r), where α is a packet label and r is a compact internal state. If polynomially many distinct pairs occur at every depth, identical pairs can be merged and their probabilities added. This gives a dynamic program instead of an exponential branch tree.

Proposition 5.1 (Branch-merging simulation)**.** *Suppose a measurement-and-feedback circuit has polynomial depth and polynomially many distinct reachable packet representations at each step. Assume that the initial representation is efficiently constructible, each representation has a polynomial-time canonical key sufficient for all future transitions, all successor representations and their transition probabilities can be enumerated in polynomial time, and the final predicate is efficient. Then the final acceptance probability is classically computable to the requested polynomial precision in polynomial time. The same conclusion holds for any polynomially supported final output distribution.*

Proof. Maintain a probability dictionary over distinct packet representations. Each measurement or controlled gate updates every entry into polynomially many successors. Merge equal successors. The dictionary remains polynomial by hypothesis. □

5.3.4 Reset and dissipation

Reset is nonunitary but often simplifies a classical representation by removing entropy or closing branches. Conversely, measurement may expose a classical label while leaving a complex logical state. The resource value of measurement depends on the target, ancilla convention, and cost function.

5.3.5 Connection to catalytic gate models

The larger gate-set monograph studies temporary excursions into measurement, ancilla, or symmetry-breaking resources. The present monograph asks a different question: when does the resulting model remain classically simulable? A temporary resource may shorten a quantum circuit without making the computation classically easy.

5.4 Block encodings and polynomial operator transforms

5.4.1 Projector block form

A block encoding places a generally nonunitary operator A inside a larger unitary U so that

$$(\langle 0| \otimes I)U(|0\rangle \otimes I) = \frac{A}{\alpha}.$$

With

$$P_0 = |0\rangle\langle 0| \otimes I,$$

this is the selected block

$$P_0 U P_0 = \frac{A}{\alpha}$$

after identifying the encoded subspace.

5.4.2 Structural connection

The encoded operator is literally a projector block of a unitary. Polynomial transformations of singular values can be implemented by controlled phase sequences once the block encoding and access model are available. This makes block encodings a natural part of a projector calculus.

5.4.3 Why block encoding does not imply classical simulation

A block encoding can be efficient on a quantum computer because U is supplied by an oracle or a succinct circuit. A classical algorithm may not have efficient access to the entries of A, to the block state, or to the polynomial transform. The selected block may still have exponential dimension.

> **Scope boundary**
>
> The statement "A is a block of U" is structural. A classical collapse requires a polynomial classical description of the active encoded state, efficient application of the selected block, and efficient evaluation of the output.

5.4.4 Symmetry-resolved block encodings

If A commutes with a symmetry group, both the encoded system and the ancilla can be decomposed into packets. A symmetry-resolved block encoding may reduce normalization, memory, or simulation cost for selected sectors. The relevant objects are

$$P_\chi P_0 U P_0 P_\psi.$$

Exact symmetry removes off-diagonal $\chi \neq \psi$ blocks. Approximate symmetry yields leakage terms that must be included in the error budget.

5.4.5 Research direction

A useful classical theorem would identify access models in which the active block has polynomial packet dimension or a compact internal representation. This could cover structured sparse matrices, low-rank updates, association-scheme operators, and fixed-particle Hamiltonians.

CHAPTER 6

The complexity boundary

6.1 Compressed packet simulation

6.1.1 Beyond packet dimension

The small-packet theorem stores one amplitude per packet-basis state. This is not the only classical representation. A state in an exponentially large packet may lie on a polynomially parameterized manifold. Stabilizer states, Gaussian states, matrix-product states of bounded bond dimension, and low-rank operator states are standard examples of compact internal encodings.

Let $\mathcal{R}_n$ be a class of classical descriptions, and let

$$\text{decode} : \mathcal{R}_n \to P_n\mathcal{H}_n$$

associate a quantum state with each description.

Theorem 6.1 (Compressed-packet simulation)**.** *A polynomial-time uniform quantum circuit family lies in P if there is a polynomial-size classical representation satisfying:*

1. *the initial state has an efficiently constructible description $r_0 \in \mathcal{R}_n$;*
2. *every gate induces a polynomial-time update $r \mapsto T_j(r)$ with*

$$\text{decode}(T_j(r)) = G_j \,\text{decode}(r);$$

3. *all intermediate description sizes remain polynomial;*
4. *the acceptance probability is computable from the final description in polynomial time.*

Proof. Iterate the polynomial-time classical updates on the polynomial-size descriptions and evaluate the final measurement. The algorithm never reconstructs the full state vector. □

6.1.2 Two-axis complexity

The records suggest the pair

$$(D_{\text{packet}}, \chi_{\text{internal}}),$$

where D_{packet} measures active packet size or count and χ_{internal} measures the representation cost inside the packet. Efficient simulation may follow from either coordinate:

- fixed weight has polynomial D_{packet} and can use explicit amplitudes;
- stabilizer states may have exponential support but polynomial χ_{internal};
- Gaussian states may occupy large number sectors but have polynomial covariance descriptions;
- half-filled generic states may have both coordinates exponential.

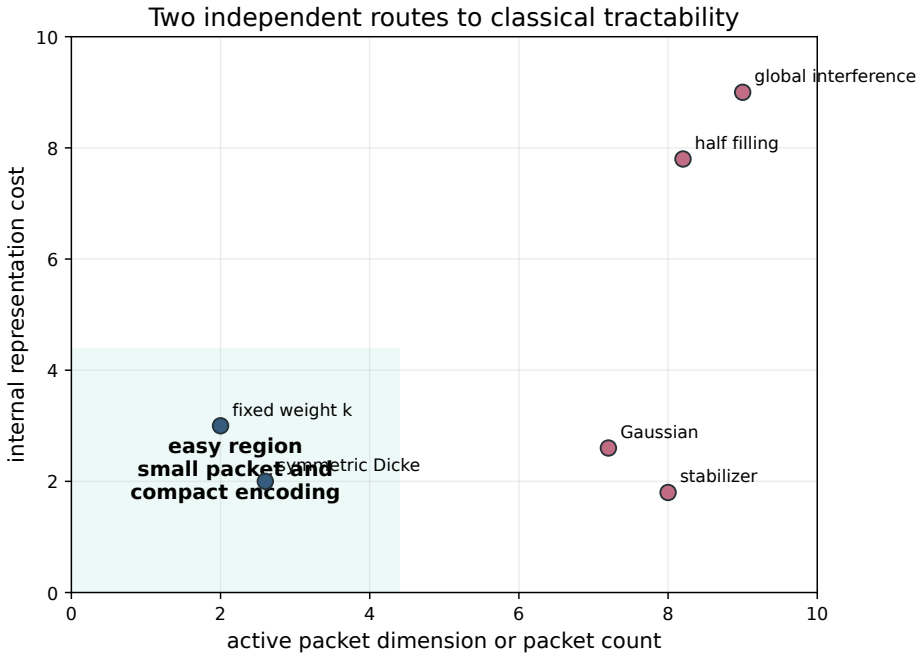


Figure 6.1. Packet dimension and internal representation cost are independent classical resources.

6.1.3 Hybrid transfer graphs

Suppose packets are nodes of a sparse graph and each node carries an internal representation. A gate may update a node internally or transfer a compact component to neighboring nodes. The simulation cost depends on

$$\sum_{\alpha \in A_t} \text{size}(r_\alpha) + \#\{\text{nonzero transfers at time } t\}.$$

This model combines symmetry selection rules with compact state classes.

6.1.4 Closure is the decisive condition

A compact representation may be destroyed by one gate. Clifford tableaux fail under generic non-Clifford rotations. Gaussian covariance matrices fail under generic non-Gaussian operations. Low-bond tensor networks may acquire exponential bond dimension. The theorem therefore requires closure or a proved polynomial growth bound, not merely a compact initial state.

Principle 6.2 (Representation growth audit)**.** *The correct complexity variable is the maximum representation size reached along the computation, not the size of the initial or final state alone.*

6.2 Global interference and the failure of local packet collapse

6.2.1 A different quantum mechanism

Invariant-packet reduction succeeds when large parts of Hilbert space are never occupied or can be compressed locally. A different regime stores the answer in coherent relations among many paths. The individual amplitudes may be simple, while their signed sum is the computational object.

6.2.2 Forrelation amplitude

Let $f, g : \{0,1\}^n \to \{-1,1\}$ and $N = 2^n$. The circuit with alternating Hadamards and phase oracles has zero-state amplitude

$$\Phi(f,g) = \frac{1}{N^{3/2}} \sum_{x,y} f(x)(-1)^{x\cdot y} g(y). \tag{6.1}$$

The circuit does not evaluate the N^2 terms separately. It preserves phases and lets them interfere.

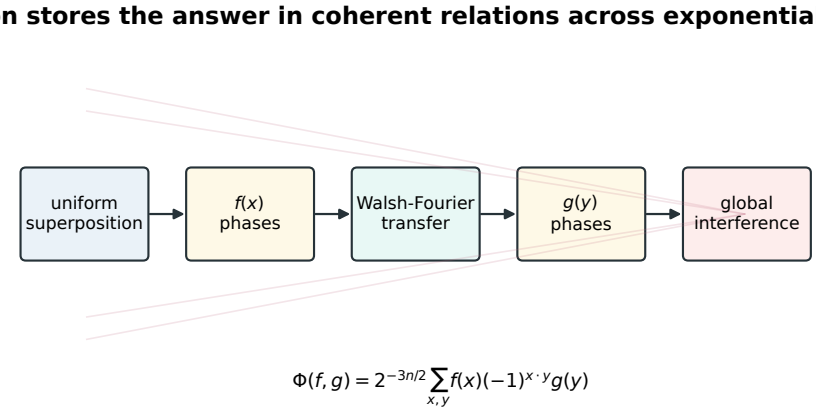


Figure 6.2. Forrelation is governed by coherent Fourier transfer rather than a small invariant sector.

6.2.3 Why Fourier structure is not automatically easy

The Walsh transform is explicit and highly symmetric. That does not imply that the bilinear expression in Equation (6.1) is classically polynomial-time computable when f and g are supplied through restricted query access. The difficulty lies in the access model and the coherent global relation, not in the absence of algebraic structure.

6.2.4 Truth tables versus succinct access

If the complete truth tables of f and g are part of the classical input, their length is already N, and an $O(N \log N)$ Walsh transform is polynomial in the explicit input length. In a query or succinct-circuit model, n is the parameter and $N = 2^n$. The same formula then represents exponentially many values.

This distinction illustrates a general rule: complexity statements must be measured against the input encoding. A quotient of exponential dimension may be efficient relative to an explicit matrix input but exponential relative to a succinct circuit input.

6.2.5 Cross-packet coherence

Suppose a projector family gives many packets P_α. Even if each diagonal block is simple, the output can depend on the phases of

$$P_\alpha|\psi\rangle \quad \text{relative to} \quad P_\beta|\psi\rangle$$

for exponentially many pairs. A simulator that retains only populations

$$\|P_\alpha|\psi\rangle\|^2$$

loses this information. The hard-side invariant must measure classical compression of coherent relations, not merely active dimension.

Open program or boundary

A representation-independent lower bound on interference compression for an explicit, succinct, uniform circuit family would be a major complexity result. The packet framework states the missing object but does not provide that lower bound.

6.3 Simulation complexity versus decision complexity

6.3.1 Why simulator lower bounds are insufficient

Showing that a circuit defeats one simulation strategy does not prove that its decision problem is outside P. A classical algorithm may compute the answer without reconstructing the quantum state. It is therefore insufficient to rule out only:

- full state-vector simulation,
- one projector family,
- bounded-width tensor-network contraction,
- path sampling,
- low-rank transfer matrices,
- stabilizer decompositions,
- any fixed catalogue of known simulators.

A proof of $P \neq BQP$ would need a lower bound against every deterministic polynomial-time classical algorithm for an explicit language.

6.3.2 A representation-relative invariant

For a declared classical representation class $\mathcal{R}$, define

$$\mathrm{TC}^{\mathcal{R}}_{\varepsilon}(C) = \min\left\{\mathrm{cost}(R) : R \in \mathcal{R},\ |R(C) - p_C| \leq \varepsilon\right\},$$

where p_C is the target acceptance probability. This permits a disciplined ladder:

1. exact polynomial-packet simulation for easy sectors;
2. lower bounds for local packet-query methods;
3. lower bounds for broader tensor, sampling, low-degree, and communication representations;
4. a representation-independent lower bound;
5. a separation of P and BQP.

Only the first step is completed here.

6.3.3 Universal packetization and its consequence

Consider the statement:

Conjecture 6.3 (Universal polynomial packetization). Every polynomial-time uniform quantum circuit family admits an efficiently discoverable exact or approximate packet representation of polynomial active size and polynomial transfer complexity, sufficient to compute its acceptance probability.

If true, Theorems 3.7 and 6.1 would imply $BQP \subseteq P$, and hence $P = BQP$. The records do not establish this conjecture and explicitly identify global interference as an obstacle.

6.3.4 The opposite route

An explicit uniform family with a proved representation-independent classical lower bound and a polynomial quantum algorithm would establish $P \neq BQP$. The candidate succinct transfer-interference problem in the records illustrates the form of such a target, but the required lower bound is the unresolved theorem itself.

> **Scope boundary**
>
> This monograph proves polynomial classical simulation for declared subclasses. It neither proves nor assumes a universal collapse of quantum computation.

6.3.5 Nonrelativizing issues

Oracle separations show that query complexity can distinguish quantum and classical access. They also show why a black-box argument alone cannot settle ordinary P versus BQP. A full separation would have to use explicit features such as uniform circuit generation, arithmetic structure, algebraic degree, communication constraints, or another nonrelativizing mechanism.

6.4 Existence defects and quantum combinatorial realizations

6.4.1 Metric collapse and existence collapse

The simulation theorems compare costs of two realizations that both exist. A broader restricted-realization framework also allows the restricted object to be absent. If

$$\mathcal{R}(x) = \varnothing, \qquad \mathcal{A}(x) \neq \varnothing,$$

then enlarging the realization category changes feasibility rather than merely runtime.

6.4.2 Quantum Latin-square example

The order-six quantum-Latin-square record supplies an existence-level example. Entangled quantum Latin squares of order six exist, while mutually orthogonal non-entangled quantum Latin squares of order six do not. This is an ambient-versus-restricted realization gap, not a proof of classical simulation and not a gate-count shortcut.

The proof architecture uses support patterns, unitary-pattern constraints, graph orthonormal representations, projective-line closure, dimension contradictions, and a finite reduction to twelve classical Latin-square classes. Its conceptual relevance here is limited but useful: enlarging the allowed quantum resource can change whether a structure exists at all.

6.4.3 Why this does not imply computational advantage

Existence of an entangled combinatorial design does not by itself provide an algorithmic speedup. One must still specify an input problem, circuit family, output rule, and classical comparison. Likewise, nonexistence of a product realization does not prove classical hardness of the entangled realization.

> **Scope boundary**
>
> The quantum Latin-square result is included as a contextual example of restricted-realization failure. It is not used as a premise in any packet-simulation theorem.

6.4.4 Unified terminology

For a target x, let $c_{\mathrm{res}}(x)$ and $c_{\mathrm{amb}}(x)$ be minimum costs, with $+\infty$ assigned when a realization does not exist. Then

$$\Delta(x) = c_{\mathrm{res}}(x) - c_{\mathrm{amb}}(x)$$

includes both finite metric defects and infinite existence defects. Symmetry collapse concerns the special case where the ambient quantum description has a cheaper classical realization because its reachable representation is compressed.

CHAPTER 7

Algorithms and open directions

7.1 Architecture of a symmetry-collapse compiler

7.1.1 The five-stage pipeline

A practical compiler must perform more than symbolic block diagonalization.

1. Detect exact or approximate symmetries.
2. Construct an invariant basis, orbit quotient, or compact internal representation.
3. Compile each gate into that representation.
4. Track leakage, truncation, and numerical error.
5. Evaluate the requested measurement and emit a complexity certificate.

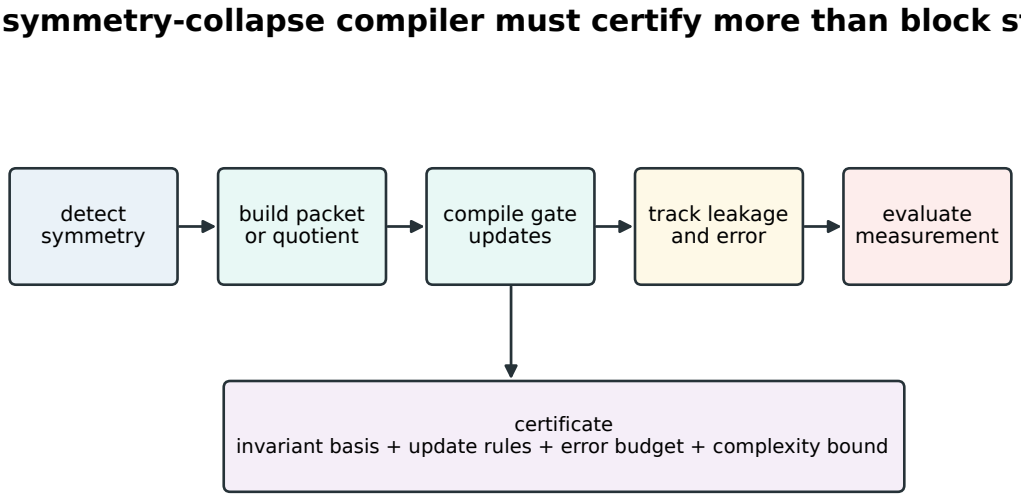


Figure 7.1. A symmetry-collapse compiler and its required certificate.

7.1.2 Symmetry detection

Detection may use declared conserved quantities, commutators, finite-group generators, graph automorphisms, stabilizer relations, occupation conservation, or numerical approximate symmetries. Exact symbolic detection is preferable when available. Numerical detection must specify tolerances and how they propagate.

7.1.3 Basis construction

Examples include:

- subset indexing for fixed weight,
- occupation vectors for bosons,
- Dicke states for full permutation symmetry,
- Schur coordinates for broader permutation symmetry,
- orbit states for graph walks,
- stabilizer tableaux,
- fermionic or bosonic covariance matrices,
- matrix-product states for bounded entanglement.

The compiler should avoid materializing an exponentially large change-of-basis matrix. It needs direct indexing and update formulas.

7.1.4 Gate plan caching

For a repeated circuit family, compile a packet plan containing affected index pairs, sparse matrix blocks, phase tables, and quotient couplings. The plan can be reused across parameter values when the sparsity pattern is fixed.

7.1.5 Fallback and adaptive enlargement

If a gate leaks from the current representation, the compiler can reject the reduction, merge packets, add neighboring sectors, or switch internal representations. The choice must preserve a certified error budget.

7.1.6 Output certificate

A useful machine-readable certificate should include:

- the symmetry or invariant-subspace proof,
- active dimensions or representation sizes,
- gate update complexity,
- leakage or approximation bounds,
- measurement complexity,
- benchmark and hardware metadata,
- the final complexity conclusion and its scope.

7.2 Benchmark and validation protocol

7.2.1 Fair comparison

A packet simulator and a full simulator should use the same precision, gate parameters, initial state, observable, and arithmetic path wherever possible. Report compilation separately from repeated application.

7.2.2 Required metrics

A complete benchmark should report:

1. full and reduced dimensions,

2. packet-discovery and compilation time,
3. one-shot runtime,
4. repeated-application runtime,
5. peak memory,
6. reconstruction or observable error,
7. symmetry leakage,
8. hardware and software versions,
9. number of repetitions over which compilation is amortized.

7.2.3 Algebraic and numerical validation paths

For manageable n, embed a random packet state into the full space, apply identical gates in both representations, and compare final states. The included numerical validation program performs this check for structurally invariant fixed-weight circuits using complex floating-point arithmetic, with reconstruction tolerance $2 \cdot 10^{-12}$. The separate rational validation uses exact fractions and rational orthogonal gates. Exact subspace invariance is a property of the model, while near-machine-precision agreement is numerical evidence about this implementation.

7.2.4 Scaling studies

A scaling plot should include both fine packets and coarse packets. Fixed weight two demonstrates polynomial dimension. Parity demonstrates a constant-factor packet. Half filling demonstrates an exponential packet with a polynomial memory reduction factor. These controls prevent overinterpreting a speedup as a complexity collapse.

7.2.5 Leakage tests

Deliberately append a gate that breaks the proposed symmetry and verify that the simulator detects nonzero leakage. The archived tests break exact weight, parity, and weight modulo four, producing retained probabilities below one. A simulator that reports zero error after a symmetry-breaking gate without enlarging its representation is incorrect.

7.2.6 Reproducibility contract

The release preserves:

- the benchmark JSON data,
- the Python figure generator,
- packet and gate utilities,
- exact rational checks and separately labeled numerical validation tests,
- LaTeX source,
- deterministic build commands,
- checksums of final artifacts.

Recorded computation

The release validation reproduces fixed-weight basis dimensions, residue partitions, symmetric/bosonic/fermionic dimension formulas, hypercube quotient symmetry, and full-vs-packet state reconstruction near machine precision.

7.3 Atlas of candidate symmetry collapses

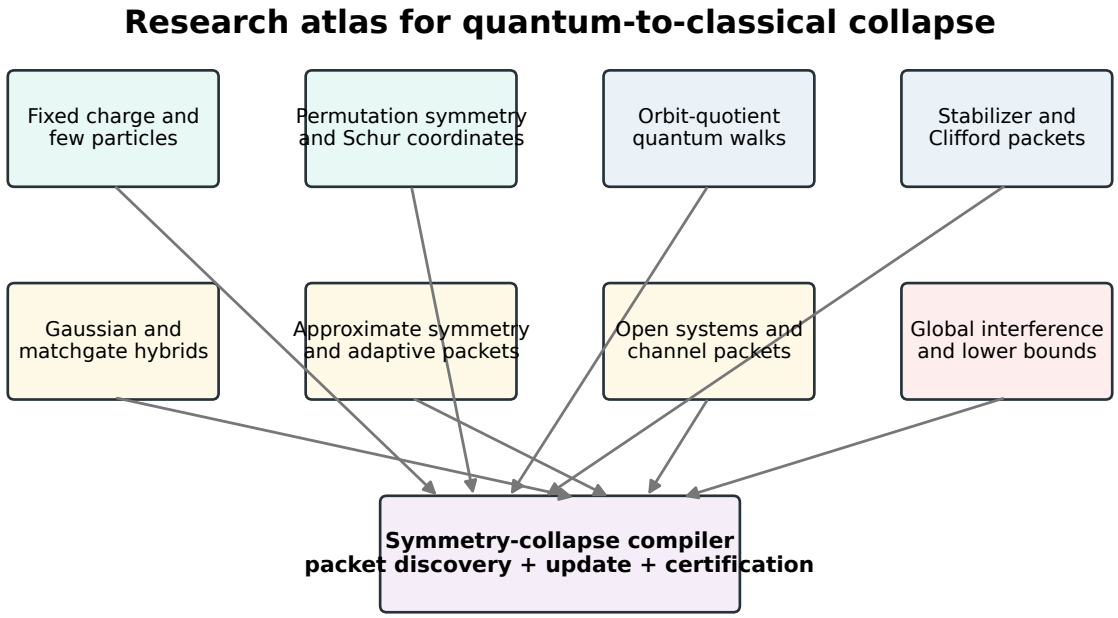


Figure 7.2. The main theorem and implementation directions organized around one certified compiler.

7.3.1 High-confidence theorem families

Table 7.1. High-priority collapse families.

Family	Compression mechanism	Decisive condition
Fixed charge $k = O(1)$	Explicit subset basis of size $\binom{n}{k}$	Number-preserving gates and efficient measurement
Permutation-symmetric qubits	Dicke basis of size $n + 1$	Collective or permutation-invariant dynamics
Symmetric qudits, fixed d	Occupation basis $\binom{n+d-1}{d-1}$	Efficient occupation-space updates
Polynomial-orbit walks	Quotient Hamiltonian on orbit states	Succinct quotient entries and orbit-invariant input/output
Few bosons or fermions	Fixed-particle occupation basis	Fixed particle number and efficient few-body updates

Family	Compression mechanism	Decisive condition
Stabilizer circuits	Tableau representation	Clifford closure and efficient Pauli measurement
Gaussian circuits	Covariance representation	Gaussian closure and efficient observable evaluation

7.3.2 Hybrid theorem targets

The strongest new classes may combine several resources:

- charge packets plus Gaussian covariance,
- permutation packets plus bounded-bond tensor networks,
- stabilizer tableaux plus a small syndrome transfer graph,
- orbit quotients plus Fourier diagonalization,
- adaptive approximate packets plus rigorous leakage accounting.

7.3.3 Negative controls

A research atlas should preserve cases that do not collapse:

- parity alone,
- weight modulo a fixed constant,
- half filling with generic internal states,
- a syndrome label with exponentially many logical qubits,
- a block encoding without classical access,
- global phase interference without a compact representation.

7.3.4 Priority order

The recommended sequence is:

1. implement the Dicke and orbit-quotient simulators;
2. combine fixed charge with Gaussian and stabilizer encodings;
3. develop adaptive leakage-certified packet growth;
4. build open-system four-index packet compilers;
5. search for broader representations of global interference;
6. separate simulator-specific lower bounds from decision-problem lower bounds.

7.4 Open problems

7.4.1 Efficient hidden-symmetry discovery

Question 7.1. Which uniform circuit families have polynomial-dimensional invariant subspaces that can be discovered in polynomial time from the circuit description?

A nonconstructive invariant subspace does not yield an algorithm. The problem includes symbolic commutant computation, graph-orbit discovery, representation decomposition, and approximate invariant-subspace detection.

7.4.2 Moving-frame discovery

Question 7.2. Which circuit families admit efficiently computable frames W_t that expose one fixed polynomial packet even though no fixed laboratory-frame symmetry is preserved?

The theorem is exact, but the discovery problem is the substance. Useful frames must have compact descriptions and restricted gate matrices that can be generated without solving the original simulation problem.

7.4.3 Permutation-commutant compilers

Question 7.3. For which succinct permutation-invariant Hamiltonians and channels can all Schur-commutant block updates be generated in polynomial time without explicitly performing the full Schur transform?

The parameter count is polynomial for fixed local dimension. The remaining algorithmic issue is efficient block access.

7.4.4 Dynamic packet graphs

Question 7.4. Can local selection rules guarantee polynomial growth of the reachable packet graph for polynomially long circuits with weak but repeated symmetry breaking?

Bounded excitation gives one exact theorem. Broader results may require locality, energy penalties, or cancellation of packet-transfer paths.

7.4.5 Minimal packet complexity

Question 7.5. Given a circuit and error tolerance ε, what is the minimum active representation size among all efficiently applicable exact or approximate packet decompositions?

This optimization should include internal encoding and transfer sparsity, not only projector rank.

7.4.6 Adaptive leakage

Question 7.6. Can one prove polynomial active packet growth for physically local Hamiltonians with weak symmetry breaking over polynomial time?

The commutator bound is local in time and often conservative. Stronger bounds may use energy gaps, locality, Lieb–Robinson estimates, or destructive interference among leakage channels.

7.4.7 Permutation and Schur compilers

Question 7.7. Which noncollective but globally permutation-invariant circuit layers admit sparse Schur-basis updates without constructing the full Schur transform?

7.4.8 Orbit quotients under succinct access

Question 7.8. For which succinct graph families are orbit counts, orbit sizes, and quotient couplings computable in polynomial time without enumerating the graph?

7.4.9 Hybrid Gaussian and packet methods

Question 7.9. Can a polynomial number of non-Gaussian packet corrections be maintained on top of a Gaussian reference for broad interacting few-particle dynamics?

7.4.10 Open systems

Question 7.10. When does the number of reachable population and coherence blocks remain polynomial under a local Lindblad generator?

7.4.11 Universal packetization

Question 7.11. Is every polynomial-size quantum circuit classically compressible by some efficiently discoverable representation of polynomial size?

A positive answer would imply $P = BQP$. The record does not support this as a universal empirical principle.

7.4.12 Interference lower bounds

Question 7.12. Can one prove an explicit representation-independent exponential lower bound for the classical cost of the coherent interference needed by a uniform BQP language?

This is the hard-side frontier. Lower bounds against one simulator family are intermediate results, not the final theorem.

Open program or boundary

The most achievable next theorem is not a universal complexity result. It is a complete compiler theorem for one new symmetry family, with explicit discovery, update, measurement, and leakage algorithms.

7.5 Conclusion

The quantum state vector is a universal coordinate system, not always the computationally intrinsic one. Conserved charges, finite-group representations, permutation orbits, few-particle sectors, and operator packets can expose much smaller dynamics. When the packet is efficiently discoverable, polynomial in size, closed under the gates, and compatible with the measurement, the quantum computation collapses exactly to a deterministic classical polynomial-time simulation.

The strongest current examples are concrete. Fixed charge at constant particle number is polynomial. Exact charge conservation is not necessary: moving packet schedules, efficiently computable moving frames, and bounded unions of low-excitation sectors also collapse to polynomial classical simulation. Fully symmetric fixed-local-dimension dynamics is polynomial. A stronger mixed-state theorem compresses all permutation-invariant Schur sectors into a polynomial commutant record, with $O(n^3)$ density-matrix parameters for qubits under efficient block access. Polynomial-orbit walks are polynomial under quotient-access hypotheses. Few-particle bosonic and fermionic systems are polynomial under efficient updates, while growing particle number gives a sharp progression through quasipolynomial, subexponential, and exponential sector sizes. Stabilizer and Gaussian representations show that an exponential packet can also be easy when its internal states admit compact closed coordinates.

Recorded number-conserving benchmarks show large memory and repeated-application gains with zero reconstruction error at the recorded precision.

The boundaries are equally sharp. Parity, modulo-four charge, and half filling do not generally change exponential scaling. Entanglement can remain hard inside one packet. A syndrome label does not compress logical information. A block encoding does not provide a classical access model. Global Fourier interference may be fully structured yet resist small packetization. Symmetry collapse is therefore not a slogan that all quantum computation is classical. It is a theorem architecture for identifying substantial subclasses that are.

The core research invariant is the complete classical representation record

$$\big(\text{discovery cost}, N_{\text{packets}}, D_{\text{internal}}, S_{\text{transfer}}, L_{\text{leakage}}, \text{measurement cost}\big).$$

A future compiler that computes and certifies this record can decide when the full state vector is unnecessary, when a hybrid representation is preferable, and when symmetry alone is insufficient.

PART III

Catalytic gate set distortion

A restricted realization can be compared with an ambient implementation that leaves the restriction and returns. The resulting savings depend on the gate alphabet, resource convention, and cost assigned to each step.

IN THIS PART

General theory of temporary escape 104

The complete linear core 123

Quantum and reversible gate families 167

Packets, types, holonomy, and restricted realization 196

The computational atlas and open directions 203

CHAPTER 8

General theory of temporary escape

8.1 Nested gate models and distortion invariants

8.1.1 Why a separate theory is needed

A target transformation often belongs to a visibly structured class even when the machine used to realize it is capable of much more. A lower-triangular linear transformation may be synthesized by an unrestricted CNOT network. A diagonal unitary may be realized by a non-diagonal Clifford+T circuit. A charge-preserving unitary may be compiled by gates that temporarily visit other charge sectors. A Gaussian target may be realized by a circuit that briefly uses a non-Gaussian primitive. These situations share a question that is not answered by ordinary exact synthesis:

> How much does it cost to require the *entire path* of a computation to retain the structure possessed by its endpoint?

The distinction is path dependent. It is not determined only by membership of the final target in a subgroup, a subcategory, or a model class.

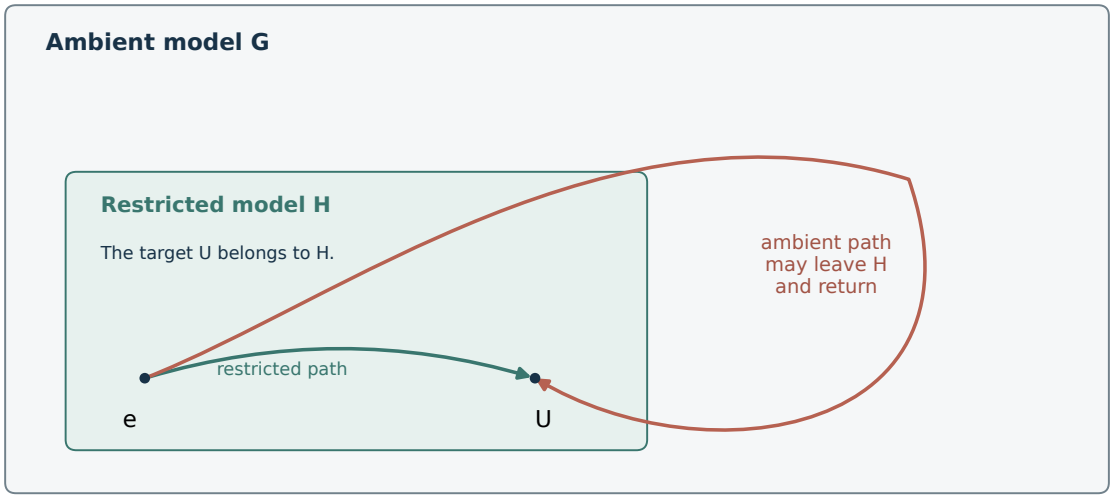


Figure 8.1. A restricted model inside an ambient model. The target lies in the restricted class, while an ambient implementation may leave it temporarily and return. Each gate family requires its own specified inclusion and costs.

8.1.2 Exact gate models

Definition 8.1 (Gate model)**.** An *exact gate model* is a tuple

$$\mathfrak{M} = (\mathcal{X}, \mathcal{S}, c, \circ),$$

where $\mathcal{X}$ is a set of states or morphisms, $\mathcal{S}$ is a specified family of elementary generators, $c : \mathcal{S} \to \mathbb{R}_{\geq 0}$ is a cost function, and $\circ$ is the composition law. A word

$$w = s_L \cdots s_1$$

has cost

$$c(w) = \sum_{j=1}^{L} c(s_j).$$

For a target $U \in \mathcal{X}$, the exact synthesis cost is

$$\ell_{\mathfrak{M}}(U) = \inf\{c(w) : w = U\}.$$

The value is $+\infty$ when the target has no realization.

The composition law may be group multiplication, composition of permutations, composition of quantum channels, tensor-network contraction, or composition in a dagger symmetric monoidal category. The additive cost may count all gates, only entangling gates, only non-Clifford gates, weighted error, or a vector of several resources. Depth fits this additive model by taking admissible parallel layers as generators, rather than charging one unit to each individual gate.

Definition 8.2 (Nested realization pair)**.** A *nested realization pair* consists of restricted and ambient models

$$\mathfrak{M}_{\text{res}} \subseteq \mathfrak{M}_{\text{amb}}$$

with a common target class $\mathcal{T}$ of finite ambient cost and the same cost on common generators. For $U \in \mathcal{T}$, define

$$\ell_{\text{res}}(U) = \ell_{\mathfrak{M}_{\text{res}}}(U), \qquad \ell_{\text{amb}}(U) = \ell_{\mathfrak{M}_{\text{amb}}}(U).$$

The additive defect is

$$\Delta(U) = \ell_{\text{res}}(U) - \ell_{\text{amb}}(U).$$

When both costs are finite, $\Delta(U)$ is a metric or optimization defect. If $\ell_{\text{res}}(U) = +\infty$ and $\ell_{\text{amb}}(U) < +\infty$, it is an existence defect. This convention unifies finite gate savings with the failure of a restricted realization category.

Proposition 8.3 (Monotonicity)**.** *For every target in the common domain,*

$$\ell_{\text{amb}}(U) \leq \ell_{\text{res}}(U).$$

Consequently $\Delta(U) \geq 0$ *whenever both models use the same nonnegative cost on their common generators.*

Proof. Every restricted word is also an ambient word. □

8.1.3 Cost models are part of the theorem

A gate-set comparison is incomplete until the cost convention is fixed. The same pair of groups can display a defect under entangling count and no defect under total primitive count. A SWAP gate may

be primitive, may cost three CNOTs, or may be implemented by physical motion. A local Clifford may be free in a fault-tolerant resource estimate and nonfree in a pulse-level model.

Definition 8.4 (Resource vector)**.** Let

$$\mathbf{c}(s) = (c_1(s), \dots, c_r(s)) \in \mathbb{R}^r_{\geq 0}$$

be the resource vector of a gate. The achievable resource region of a target is

$$\mathcal{P}_{\mathfrak{M}}(U) = \left\{ \sum_{j=1}^{L} \mathbf{c}(s_j) : s_L \cdots s_1 = U \right\}.$$

Its nondominated points form the Pareto boundary.

For a weight vector $\lambda \in \mathbb{R}^r_{\geq 0}$, scalarization gives

$$\ell^{\lambda}_{\mathfrak{M}}(U) = \inf_{x \in \mathcal{P}_{\mathfrak{M}}(U)} \lambda \cdot x.$$

The scalar defect $\Delta^{\lambda}(U)$ can vary with λ even when the target and gate alphabets are fixed.

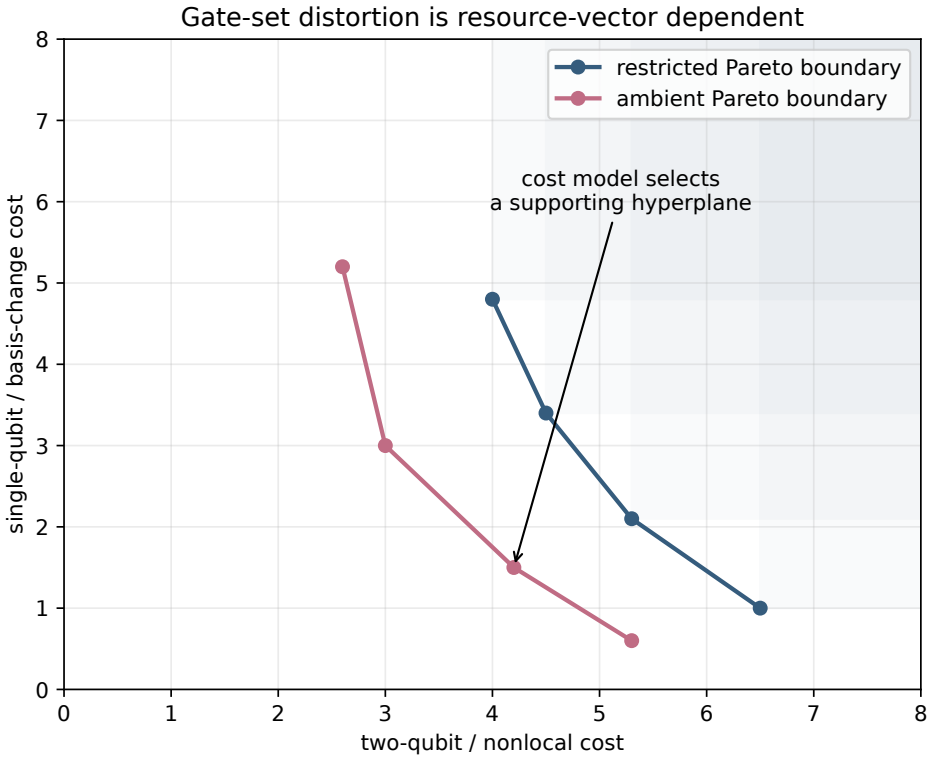


Figure 8.2. Restricted and ambient Pareto regions can differ without one model dominating by the same amount under every resource weighting.

Principle 8.5 (Model specificity)**.** *Every claimed shortcut must state the target class, elementary gate alphabet, ancilla convention, exact or approximate synthesis criterion, and cost vector. Changing any one of these may remove, create, or reverse a defect.*

8.1.4 Ancilla and restoration conventions

Temporary workspace is meaningful only after its restoration rule is specified.

Definition 8.6 (Ancilla conventions)**.** An ancilla is

1. *clean* if it begins in a specified state and must return to that state,
2. *borrowed* or *dirty* if its initial state is arbitrary but must be restored,
3. *consumable* if restoration is not required,
4. *catalytic* if it is restored while enabling a cheaper realization of the target.

The present theory focuses on restoration. If an auxiliary resource may be discarded, the comparison becomes a different channel or isometry synthesis problem.

8.1.5 Worst-case and threshold functions

For a size parameter n, let $\mathcal{T}_n$ denote the target family.

Definition 8.7 (Extremal distortion)**.** Define

$$D_n = \sup_{U \in \mathcal{T}_n} \Delta(U)$$

and, when $0 < \ell_{\text{amb}}(U) < \infty$,

$$R_n = \sup_{U \in \mathcal{T}_n} \frac{\ell_{\text{res}}(U)}{\ell_{\text{amb}}(U)}.$$

The first defective size is

$$\tau = \min\{n : D_n > 0\}.$$

We set a threshold to $+\infty$ when its defining set is empty.

Several threshold notions should not be conflated:

$$\begin{aligned} \tau_{\text{metric}} &= \min\{n : \exists U,\ 0 < \Delta(U) < \infty\}, \\ \tau_{\text{exist}} &= \min\{n : \exists U,\ \ell_{\text{res}}(U) = \infty,\ \ell_{\text{amb}}(U) < \infty\}, \\ \tau_q &= \min\{n : \exists U,\ \ell_{\le q}(U) < \ell_{\le 0}(U)\}, \end{aligned}$$

where $\ell_{\le q}$ permits at most q ambient-only gates. A model may have an early metric threshold and a later existence threshold, or no threshold at all.

8.1.6 Equivalence, orbits, and classification

A finite census should be quotiented by transformations preserving both gate models and the cost. Depending on the problem, these may include wire permutations, local Clifford conjugation, isotopy, simultaneous row and column relabeling, or graph automorphisms.

Definition 8.8 (Cost-preserving equivalence)**.** Targets U and V are equivalent if an automorphism φ of the ambient realization system satisfies

$$\varphi(U) = V, \qquad \varphi(\mathcal{S}_{\text{res}}) = \mathcal{S}_{\text{res}}, \qquad \varphi(\mathcal{S}_{\text{amb}}) = \mathcal{S}_{\text{amb}},$$

and preserves the gate costs.

Under this equivalence, both lengths and the defect are orbit invariants. A useful threshold theorem should classify minimal defective orbits, not merely list raw matrices.

8.1.7 Approximate synthesis and border defects

For continuous gate models, exact word length can differ from the minimum length achievable as a limit.

Definition 8.9 (Border cost)**.** Let $\overline{\mathcal{W}}_L$ be the topological or Zariski closure of targets realizable with cost at most L. The border cost is

$$\underline{\ell}(U) = \inf\{L : U \in \overline{\mathcal{W}}_L\}.$$

The border defect is

$$\underline{\Delta}(U) = \underline{\ell}_{\text{res}}(U) - \underline{\ell}_{\text{amb}}(U).$$

This is the circuit analogue of the distinction between tensor rank and border rank. An exact defect can disappear under degeneration, persist, or become larger after approximation changes the relevant cost geometry.

8.1.8 Metric and existence defects in one category

Let $\mathcal{C}_{\text{res}} \subseteq \mathcal{C}_{\text{amb}}$ be categories with weighted generating morphisms. For a morphism f in the restricted target class, the same definition

$$\Delta(f) = c_{\text{res}}(f) - c_{\text{amb}}(f)$$

includes group-word metrics, adaptive quantum channels, tensor-network realizations, and orthogonality designs. This categorical form is essential when the ambient model permits measurement, branching, postselection, or entangled objects that do not lie in a common group.

> **Scope boundary**
>
> The abstract definition does not prove that a defect exists. It provides a common language in which a finite search, a symbolic identity, and a lower-bound certificate can be compared without changing the meaning of the target or the cost.

8.1.9 A standard theorem package

A mature gate-set distortion result should contain as many of the following components as the model allows:

1. a precisely defined nested gate pair and cost vector,
2. a smallest defective target or a no-defect theorem below a threshold,
3. an explicit ambient word,
4. a restricted lower-bound certificate,
5. classification modulo cost-preserving equivalence,
6. a reusable catalytic normal form,
7. direct-sum, recursive, or tensor amplification,
8. generic-rigidity or sparsity results,
9. a certificate-producing implementation,
10. a clear account of which statements are symbolic, computed, or conjectural.

The remainder of the monograph follows this package.

8.2 Catalytic holonomy, excursion geometry, and resource reuse

8.2.1 Subgroup metrics and Schreier loops

Assume first that the restricted and ambient models are groups

$$H = \langle \mathcal{S}_H \rangle \leq G = \langle \mathcal{S}_G \rangle.$$

For $h \in H$, the intrinsic and ambient word metrics are

$$\ell_H(h) = |h|_{\mathcal{S}_H}, \qquad \ell_G(h) = |h|_{\mathcal{S}_G}.$$

In the group-metric statements, the generating alphabets are symmetric and a generator and its inverse have equal cost. The quotient geometry is the left Schreier graph G/H, with edges $gH \longmapsto sgH$. For the base-fiber assertion below, assume also $\mathcal{S}_H = \mathcal{S}_G \cap H$. This last condition excludes additional ambient primitives that already lie in H but have no equally cheap restricted realization.

Theorem 8.10 (Holonomy-compression identity)**.** *Let every generator have unit cost. For $h \in H$, $\ell_G(h)$ is the minimum length of a based loop in G/H whose lifted word has holonomy h. The intrinsic length $\ell_H(h)$ is the minimum length among loops that remain in the base fiber.*

Proof. An ambient word $s_k \cdots s_1 = h$ projects to the loop

$$H, s_1 H, s_2 s_1 H, \ldots, hH = H.$$

Conversely, a based Schreier loop has product in H. If every prefix lies in H, each edge label is the product of two elements of H and belongs to $\mathcal{S}_G \cap H = \mathcal{S}_H$. Conversely every restricted word remains in the base fiber. Minimizing the labeled words with endpoint h gives the two asserted lengths. □

Thus

$$\Delta_{H \hookrightarrow G}(h) = \text{intrinsic fiber cost} - \text{shortest holonomy loop cost}.$$

This is the geometric content of a dirty-workspace shortcut.

8.2.2 The basic conjugation inequality

Let $g \in G$ and $k \in H$ satisfy $gkg^{-1} \in H$. Then

$$\ell_G(gkg^{-1}) \leq 2\ell_G(g) + \ell_H(k).$$

Define the intrinsic conjugation stretch

$$\mathrm{Stretch}_g(k) = \ell_H(gkg^{-1}) - \ell_H(k).$$

Proposition 8.11 (Catalytic stretch criterion)**.** *If*

$$\mathrm{Stretch}_g(k) > 2\ell_G(g),$$

then gkg^{-1} has positive defect.

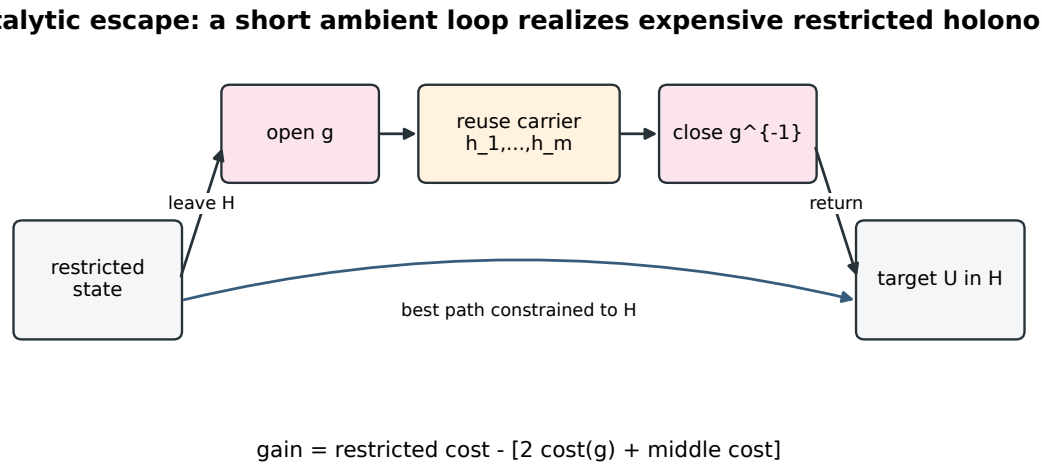


Figure 8.3. A catalytic excursion leaves a restricted fiber, reuses an ambient state, and returns with nontrivial holonomy.

The opening and closing cost is paid once. Every useful consumer inside the excursion contributes to the stretch.

8.2.3 Many consumers and catalytic excess

Suppose the middle word is

$$k = h_m \cdots h_1, \qquad h_i \in H.$$

The ambient word

$$g h_m \cdots h_1 g^{-1}$$

has cost at most

$$2c(g) + \sum_i c(h_i).$$

If the restricted images $g h_i g^{-1}$ require costs r_i and a lower-bound invariant proves that their obligations cannot be shared, then

$$\ell_H(g k g^{-1}) \geq \sum_i r_i.$$

A sufficient inequality is

$$\sum_i (r_i - c(h_i)) > 2c(g).$$

The left side is the gross catalytic benefit. The right side is the restoration cost.

Definition 8.12 (Catalytic excess)**.** For a normal form with r matched excursions and C useful cross interactions, define

$$\mathcal{E} = C - 2r$$

when each consumer contributes one unit and each opening or closing costs one unit.

Positive excess is necessary in the simplest transvection model, but it is not sufficient. An all-cross configuration may admit a mirror word of equal restricted length. Temporal asymmetry or phase holonomy is also required.

8.2.4 Excursion budgets

Counting ambient-only gates is only one measure of how far a circuit leaves its restricted model.

Definition 8.13 (Excursion profile)**.** For a circuit path $X_0, X_1, \dots, X_L$, an excursion profile may include

1. the number q of ambient-only gates,
2. the maximum Schreier distance $d(X_t H, H)$,
3. the rank of a forbidden block,
4. the bandwidth of a forbidden dependency,
5. the number of simultaneously open catalytic intervals,
6. the intersection graph of those intervals,
7. the maximum algebraic degree created temporarily,
8. the maximum ancilla dimension or branch count.

The established linear family shows that a rank-one, bandwidth-one excursion using exactly two forbidden gates can save linearly many operations. Thus geometric excursion size need not control computational benefit.

8.2.5 Catalytic interval geometry

Associate to every matched opening and closing operation an interval

$$I_j = [t_j^{\text{open}}, t_j^{\text{close}}].$$

The family of intervals can be disjoint, nested, laminar, or crossing.

Principle 8.14 (First higher-order frontier)**.** *Independent or disjoint intervals often reduce to additive copies of a one-catalyst analysis. Crossing intervals are the first place where one dirty resource can alter another and where a multiplicative distortion larger than the clean one-carrier value may occur.*

Define the catalytic width

$$w_{\text{cat}} = \max_t \#\{j : t \in I_j\}.$$

One may then study

$$D_n^{(w)} = \max_U \left(\ell_{\text{res}}(U) - \ell_{\text{amb}}^{(w)}(U) \right),$$

where the ambient circuit has width at most w. The unitriangular snapshot family gives $D_n^{(1)} = \Omega(n)$ in its corresponding interval model.

8.2.6 Four recurrent catalytic mechanisms

Basis-change catalysis

An ambient gate g changes the frame in which several restricted operations are cheap:

$$h_i \mapsto g h_i g^{-1}.$$

Hadamard and Fourier transforms are the basic quantum examples. The frame is restored at the end.

Commutator catalysis

An ambient commutator creates a new direction:

$$[g, h] = ghg^{-1}h^{-1}.$$

For continuous controls,

$$e^{\varepsilon X}e^{\varepsilon Y}e^{-\varepsilon X}e^{-\varepsilon Y} = e^{\varepsilon^2[X,Y]+O(\varepsilon^3)}.$$

The desired effective generator may lie in the restricted algebra even though the path does not.

Degree catalysis

A Toffoli, CCZ, or non-Gaussian gate can create higher algebraic degree or higher moments temporarily. If the nonlinear object is reused and canceled, the final target may remain affine, diagonal, or Gaussian.

Branch and workspace catalysis

Ancillas, measurements, reset, teleportation, and classical feedforward enlarge the realization category. The resource is a temporary branch, subsystem, or borrowed state rather than a group element.

8.2.7 When defects are impossible

A useful positive theorem is a criterion guaranteeing geodesic convexity.

Proposition 8.15 (One-Lipschitz retraction criterion)**.** *Suppose there is a group homomorphism $\pi : G \to H$ with $\pi|_H = \mathrm{id}_H$ such that every ambient generator $s \in \mathcal{S}_G$ satisfies*

$$\ell_H(\pi(s)) \le c(s).$$

Then

$$\ell_H(h) \le \ell_G(h)$$

for every $h \in H$, hence the two metrics agree on H.

Proof. For an ambient word $h = s_L \cdots s_1$, multiplicativity gives $h = \pi(h) = \pi(s_L)\cdots\pi(s_1)$. Subadditivity and the generator bounds give $\ell_H(h) \le \sum_j c(s_j)$. Taking the infimum over ambient words proves the claim. □

The unitriangular counterexample proves that no such generator-wise one-Lipschitz retraction exists for the lower transvection model inside the all-transvection model.

8.2.8 Weighted restoration and phase transitions

Let every restricted gate cost one and every ambient-only gate cost λ. For a catalyst used by v consumers with unit benefit, the displayed construction has net gain at least

$$v - 2\lambda.$$

The critical weight is therefore target and mechanism dependent.

Definition 8.16 (Critical catalyst penalty)**.** For a target family U_n, define

$$\lambda_c(U_n) = \sup\{\lambda : \ell_{\lambda,\mathrm{amb}}(U_n) < \ell_{\mathrm{res}}(U_n)\}.$$

Determining λ_c exactly is a refined synthesis problem. It can be more physically meaningful than unweighted gate count.

8.2.9 Restoration is the source of nontriviality

If the ambient resource need not be restored, the opening cost can dominate the comparison and many examples become trivial. Catalytic escape is strongest when the following all hold:

1. the final target lies entirely in the restricted class,
2. every auxiliary subsystem or violated constraint is restored,
3. the ambient-only resource is used more than once,
4. the restricted lower bound charges the same repeated obligations separately,
5. the gain survives a realistic weighted cost.

> **Conjectural frontier**
>
> In many finite gate hierarchies, minimal defects should admit a bounded-width catalytic normal form. Proving or refuting this would turn an unrestricted synthesis search into a search over a finite family of excursion motifs.

8.3 First arrival, spectral packets, algebraic rarity, and tensor regularization

8.3.1 Word distance as first arrival

Let G be finite and let $S = S^{-1}$ be a symmetric generating multiset. In the right regular representation, define

$$A_S = \sum_{s \in S} \rho(s).$$

Then

$$\langle \delta_g, A_S^k \delta_e \rangle$$

counts length-k words ending at g.

Theorem 8.17 (Endpoint Green valuation)**.** *The word distance is*

$$d_S(e, g) = \mathrm{ord}_{z=0} \left\langle \delta_g, (I - zA_S)^{-1} \delta_e \right\rangle.$$

Proof. Expand the resolvent as a formal power series:

$$(I - zA_S)^{-1} = \sum_{k \geq 0} z^k A_S^k.$$

The first nonzero endpoint coefficient occurs at the shortest word length. □

For strictly positive integer costs, replace each generator s by the monomial $z^{c(s)}\rho(s)$. Every coefficient then counts finitely many words of a given total cost, and the same formal inverse is defined.

Zero-cost cycles require separate treatment: they can create infinitely many words of cost zero, so the ordinary counting series need not exist. The finite atlases with free gates use shortest-path search with zero-cost edges instead. Multivariate variables can track several resources when the grading gives every generator positive total degree.

8.3.2 Why eigenvalues alone are insufficient

If

$$A_S = \sum_{\lambda} \lambda \Pi_\lambda,$$

then

$$\mathcal{G}_{S,g}(z) = \sum_{\lambda} \frac{\langle \delta_g, \Pi_\lambda \delta_e \rangle}{1 - \lambda z}.$$

The eigenvalues and multiplicities do not determine the endpoint coefficients

$$\langle \delta_g, \Pi_\lambda \delta_e \rangle.$$

Two Cayley or Schreier systems can therefore be cospectral and still have different first-arrival behavior at distinguished targets.

Principle 8.18 (Embedded packet principle)**.** *For threshold questions, a spectral label is usually weaker than an embedded spectral projector. The matrix coefficients of the projector relative to the initial and target states retain positional information that multiplicities erase.*

This distinction recurs in the Latin-square example discussed later. The natural commuting row, column, and symbol operators have universal eigenvalue packets at fixed order, while the embedded incidence and transition data still vary.

8.3.3 Subgroup-holonomy Fourier packets

Let $H \leq G$. Let $\mu_k(h)$ count ambient words of length k with product $h \in H$. For every irreducible representation $\tau \in \widehat{H}$, define

$$\widehat{\mu}_k(\tau) = \sum_{h \in H} \mu_k(h) \tau(h).$$

Fourier inversion gives

$$\mu_k(h) = \frac{1}{|H|} \sum_{\tau \in \widehat{H}} \dim(\tau)\, \mathrm{tr}\big(\tau(h^{-1}) \widehat{\mu}_k(\tau)\big).$$

The matrix-valued Green packet

$$\widehat{\mathcal{G}}_\tau(z) = \sum_{k \geq 0} \widehat{\mu}_k(\tau) z^k$$

recovers every endpoint series after Fourier inversion.

The trivial representation records only the total number of loops returning to the subgroup. Nontrivial H-types retain their holonomy. This yields a hierarchy:

$$\text{spherical return data} \prec \text{type-resolved holonomy packets} \prec \text{exact endpoint realization}.$$

8.3.4 Word maps and product loci

Let K be a field and let $X \subseteq G$ be an algebraic gate family. For a fixed length L, multiplication defines

$$\mu_L : X^L \to G, \qquad (x_L, \dots, x_1) \mapsto x_L \cdots x_1.$$

The exact length-L locus is $\mu_L(X^L)$, and the length-at-most-L locus is a finite union of such images when the gate family has finitely many components.

Theorem 8.19 (Dimension per gate)**.** *Suppose every irreducible component of X has dimension at most r. Then the Zariski closure of the locus reachable in at most L gates has dimension at most rL.*

Proof. Every component of X^L has dimension at most rL. A morphism cannot increase dimension. A finite union has dimension equal to the maximum dimension of its components. □

For scaled transvections, $r = 1$. This immediately yields generic lower bounds in a target variety whose dimension exceeds L.

8.3.5 Generic rigidity

Work over an algebraically closed field. Suppose the restricted target variety V is irreducible of dimension N, every target has a restricted realization using at most N unit-cost gates, and the ambient alphabet is an algebraic family with finitely many components of dimension at most one. Then ambient realizations of length below N occupy a proper closed subset of V.

Theorem 8.20 (Abstract generic rigidity)**.** *Under the preceding hypotheses, there is a nonempty Zariski-open subset $\Omega \subseteq V$ such that*

$$\ell_{\text{res}}(U) = \ell_{\text{amb}}(U) = N \qquad (U \in \Omega).$$

The unitriangular theorem in the core chapter is a concrete instance with

$$N = \binom{n}{2}.$$

8.3.6 Defect codimension

If every target has restricted cost at most N, then defect at least d forces ambient cost at most $N - d$. The dimension-per-gate theorem gives

$$\operatorname{codim}_V \overline{\{U : \Delta(U) \geq d\}} \geq d$$

when each ambient gate carries one parameter.

Principle 8.21 (Rare but amplifiable)**.** *At fixed dimension, large defects require increasingly special algebraic conditions. Under direct sums, recursive lifts, or tensor products, a point on a thin exceptional stratum can nevertheless seed a macroscopic family.*

This is one of the central tensions of the subject. Catalysis is neither generic nor negligible.

8.3.7 Finite-field sparsity

A word-counting analogue applies over $\mathbb{F}_q$. If the ambient alphabet has at most $M(q-1)$ nontrivial scaled gates, then the number of targets reached in L gates is at most

$$\sum_{j=0}^{L}[M(q-1)]^j.$$

When the target space has q^N points and defect d implies ambient length at most $N-d$, the crude bound is

$$\mathbb{P}[\Delta \geq d] = O_{N,M}(q^{-d}).$$

The estimate can be sharpened by orbit counting, stabilizers, and the geometry of word-map fibers.

8.3.8 Direct sums and monoidal amplification

Suppose targets admit a direct sum $U \oplus V$ and a homomorphism from the large restricted group onto the two factor groups sends $U \oplus V$ to (U, V). Require the aggregate generator bound

$$\ell_{\text{res}}(\pi_1(s)) + \ell_{\text{res}}(\pi_2(s)) \leq c(s)$$

for every large restricted generator s. Assume that both restricted and ambient factor circuits embed into the corresponding large model without increasing cost. Then

$$\ell_{\text{res}}(U \oplus V) = \ell_{\text{res}}(U) + \ell_{\text{res}}(V).$$

Indeed, projecting any word and summing the two factor bounds gives the lower bound, while concatenating the embedded factor circuits gives the upper bound. Separate one-Lipschitz projections alone do not imply this sum inequality. Under the ambient embedding assumption, the constructive upper bound is

$$\ell_{\text{amb}}(U \oplus V) \leq \ell_{\text{amb}}(U) + \ell_{\text{amb}}(V),$$

so

$$\Delta(U \oplus V) \geq \Delta(U) + \Delta(V).$$

One finite defect therefore gives an unbounded additive family.

8.3.9 Tensor products and normalized rates

Let U have dimension m and V dimension n. In linear transvection models with tree-tight certificates,

$$\ell_{\text{res}}(U \otimes V) = n\ell_{\text{res}}(U) + m\ell_{\text{res}}(V).$$

The ambient model satisfies the corresponding inequality. Normalizing by dimension gives an entropy-like law:

$$\frac{\ell_{\text{res}}(U \otimes V)}{mn} = \frac{\ell_{\text{res}}(U)}{m} + \frac{\ell_{\text{res}}(V)}{n}.$$

The normalized defect is superadditive under tensor products.

Definition 8.22 (Tensor synergy)**.** Define

$$\text{Syn}(U, V) = n\ell_{\text{amb}}(U) + m\ell_{\text{amb}}(V) - \ell_{\text{amb}}(U \otimes V).$$

Positive synergy means that the joint ambient realization improves on independent lifted circuits.

Restricted tree-tight metrics have zero tensor synergy. Ambient synergy is a possible source of collective activation.

Definition 8.23 (Stable tensor cost)**.** For a size-m target U, define

$$\tau_{\otimes}(U) = \lim_{r\to\infty} \frac{\ell_{\mathrm{amb}}(U^{\otimes r})}{rm^{r-1}},$$

when the subadditivity argument applies. Define

$$\kappa_{\otimes}(U) = \ell_{\mathrm{res}}(U) - \tau_{\otimes}(U).$$

Fekete's lemma gives the limit from the tensor subadditivity inequality.

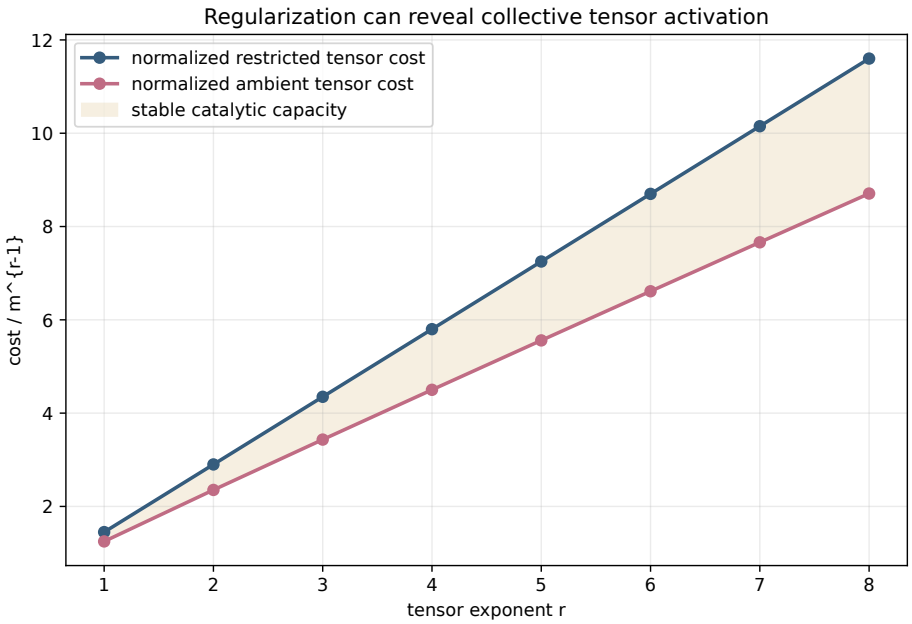


Figure 8.4. Schematic stable tensor capacity. The figure illustrates definitions, not measured data.

Definition 8.24 (Tensor activation)**.** A target is tensor activated if

$$\Delta(U) = 0 \quad \text{but} \quad \kappa_{\otimes}(U) > 0.$$

This is analogous to regularized capacities and asymptotic ranks. It asks whether collective synthesis can unlock a catalyst invisible in one copy.

8.3.10 Border catalysis

Let X_{res} and X_{amb} be algebraic gate varieties. The exact product rank is

$$\mathrm{rk}_X(U) = \min\{L : U \in X^L\},$$

while border product rank is

$$\underline{\mathrm{rk}}_X(U) = \min\{L : U \in \overline{X^L}\}.$$

The four possible comparisons are:

1. exact and border defects both positive,
2. exact defect positive but border defect zero,
3. one-copy defect zero but tensor-regularized defect positive,
4. ambient and restricted border ranks both smaller than exact ranks but by different amounts.

Each requires a separate certificate. Numerical approximation alone does not distinguish them.

8.3.11 A packet-resolved threshold principle

Combining the preceding constructions gives the following architecture.

1. A coarse commutative algebra supplies spherical spectral packets.
2. Embedded projectors retain incidence and endpoint position.
3. Noncommutative transition data record holonomy.
4. A realization theorem decides whether the packet has a restricted representative and at what cost.
5. Direct sums and tensor products propagate local defects.

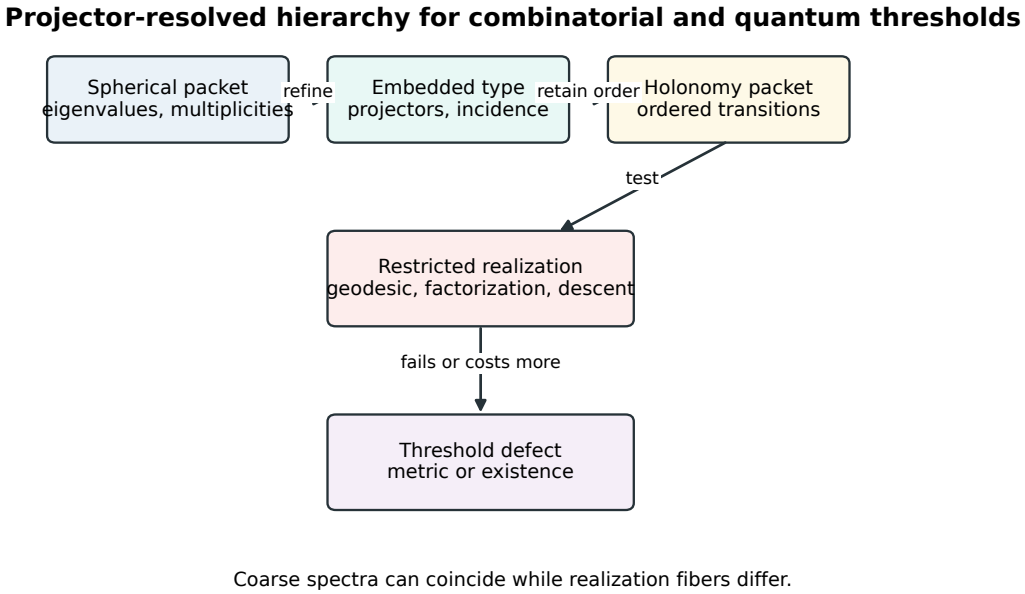


Figure 8.5. The spectral-to-realization hierarchy.

> **Scope boundary**
>
> A packet hierarchy organizes evidence. It does not replace the final lower bound, positivity argument, orthogonality construction, or descent theorem. Classification and realization remain distinct layers.

8.4 Matroidal state geometry, quiver factorizations, and root-system catalysis

8.4.1 Intermediate states as a representable matroid

A linear circuit visits vectors that satisfy exact dependence relations. If $\mathcal{V}$ is the set of all row states appearing during a computation, then $\mathcal{V}$ determines a representable matroid. A transvection

$$u \longmapsto u + av$$

creates a three-point projective line

$$[u],\ [v],\ [u + av].$$

The ambient circuit chooses a tour through this projective configuration. A flag-preserving circuit must lift the same dependencies through prescribed pivot positions.

Definition 8.25 (Flagged matroid tour)**.** Let M be a representable matroid with designated initial basis, target basis, and consumer states. A *flagged tour* is a sequence of basis exchanges or transvections in which the ith carrier remains inside a specified filtration layer. An ambient tour drops this constraint.

The six-row atom realizes the graphic regular matroid $M(K_4)$. Its ambient state tour is shorter than every compatible flag lift. Regularity explains why the same dependence pattern survives every field and, through integer matrices, every nonzero commutative coefficient ring in the stated model.

8.4.2 Matroidal classification problem

The first general classification problem is:

> Which regular matroids admit designated carrier and consumer roles for which an ambient tour is shorter than every flag-compatible tour?

Natural subclasses are graphic, cographic, and network matrices. Potential invariants include branch width, circuit size, excluded minors, and the number of consumer incidences supported by one projective line.

Conjectural frontier

Minimal field-universal catalytic atoms should be governed by regular matroids, while characteristic-dependent atoms should arise from representable matroids with excluded-characteristic minors.

This conjecture is a programmatic guide. The core paper establishes only the $M(K_4)$ instance.

8.4.3 Dynamic rank-one factorization

Consider a height-two block shear

$$E(C) = \begin{pmatrix} I_s & 0 \\ C & I_t \end{pmatrix}.$$

A restricted circuit may change source and output bases while transmitting rank-one terms. Let $P_r \in U_s$ be the source basis before the rth transmission and let $Q_{>r} \in U_t$ be the future output basis. A cross gate contributes

$$Q_{>r} e_{i_r} e_{j_r}^T P_{<r}$$

to the final block. Hence

$$C = \sum_{r=1}^{c} Q_{>r} e_{i_r} e_{j_r}^T P_{<r},$$

with closed basis paths

$$P_{\text{final}} = I_s, \qquad Q_{\text{final}} = I_t.$$

Definition 8.26 (Dynamic rank-one cost)**.** The dynamic rank-one cost of C is the minimum of

$$\text{source basis motion} + \text{output basis motion} + \text{number of transmissions}$$

over such timed decompositions.

Ordinary rank retains only the number of transmissions after arbitrary bases are given for free. Circuit complexity charges the closed paths needed to expose and restore those bases.

8.4.4 Quiver gate models

Let $Q = (Q_0, Q_1)$ be a quiver. Associate a vector space V_v to every vertex. Internal gates change bases inside V_v, while edge gates transmit low-rank maps along arrows. A target is a representation of the path algebra or an automorphism of a filtered quiver module.

This generalization includes:

- block-triangular linear circuits,
- network coding with reversible local state,
- layered dataflow architectures,
- incidence-algebra transformations,
- tensor networks with reusable bond states.

A catalytic excursion can temporarily activate a forbidden arrow or enlarge a vertex space. The restored target still lies in the original quiver representation class.

> **Research template**
>
> A quiver threshold theorem should specify a forbidden-arrow motif, an exact ambient word, a projection-based restricted lower bound, and an amplification operation under quiver substitution. The unitriangular snapshot family is the linearly ordered one-dimensional-vertex case.

8.4.5 Pattern groups and architecture constraints

For a poset P, define the pattern group

$$U_P(K) = \{I + A : A_{ij} = 0 \text{ unless } j <_P i\}.$$

When the allowed positions are transitively closed, U_P is a group. Compare the intrinsic metric generated by P-allowed transvections with the metric induced by all transvections.

The first threshold now depends on the forbidden-edge geometry of P. One may ask whether a six-role catalytic motif embeds as a generator-preserving retract. If it does, the larger architecture inherits a defect.

Proposition 8.27 (Retraction inheritance)**.** *Suppose a smaller nested pair $H \leq G$ embeds into $H' \leq G'$ by homomorphisms preserving the inclusion, the ambient embedding does not increase gate cost, and there is a homomorphic retraction $\pi : H' \to H$ onto the embedded copy. Require $\ell_H(\pi(s)) \leq c(s)$ for every restricted generator s of H'. Then every defect in $H \leq G$ gives a defect at least as large in $H' \leq G'$.*

The proof projects a restricted word in the larger system and embeds the shorter ambient word from the smaller system.

8.4.6 Root subgroups and Chevalley relations

Let G be a split reductive group [24, 31] with root system Φ. Let $\Psi \subseteq \Phi$ be a closed set of allowed roots and let U_Ψ be generated by the root groups $x_\beta(t)$ for $\beta \in \Psi$. Define

$$\ell_\Psi(u) = \min\{L : u = x_{\beta_L}(t_L) \cdots x_{\beta_1}(t_1),\ \beta_i \in \Psi\},$$

and let ℓ_Φ permit all roots.

The Chevalley conjugation formula has the schematic form

$$x_\alpha(a) x_\beta(b) x_\alpha(a)^{-1} = x_\beta(b) \prod_{k>0} x_{\beta+k\alpha}(c_k a^k b).$$

A forbidden root α can therefore create several allowed root effects before being removed.

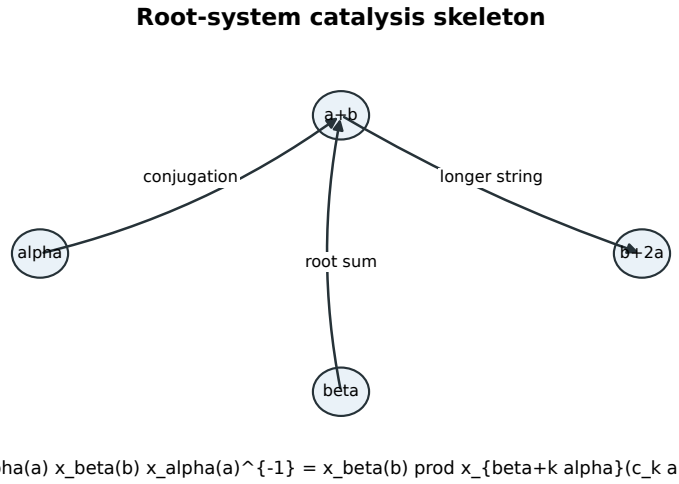


Figure 8.6. A root-string template for catalytic generation. Longer strings may support more gain per opening and closing pair.

8.4.7 Type A and beyond

Row transvections are root-group elements of type A. The established unitriangular atom occurs in A_5. In nonsimply laced systems, root strings can have length greater than two. This raises the possibility that one ambient root can create multiple restricted effects per consumer.

Conjecture 8.28 (Long-root catalytic gain). There exist gate alphabets in types B, C, or G_2 for which a clean one-catalyst multiplicative distortion exceeds the type-A snapshot limit of $3/2$.

No lower-bound certificate is presently supplied. The conjecture identifies a concrete reason to test nonsimply laced root systems rather than merely repeating type A computations.

8.4.8 Symplectic root systems and Clifford logic

The binary Clifford group modulo phases acts through $\mathrm{Sp}(2n, 2)$. A parabolic subgroup stabilizing a Lagrangian is the symplectic analogue of a Borel or flag-preserving circuit class. Hadamard gates leave this parabolic, while S, CZ, and CX preserve it. The exact Clifford censuses in Section 10.2 are therefore a first root-system extension of the linear theory.

8.4.9 Categorical realization defects

The group framework does not cover measurement, feedforward, postselection, or entangled design objects. Let

$$\mathcal{C}_{\mathrm{res}} \subseteq \mathcal{C}_{\mathrm{amb}}$$

be monoidal categories with costed generators. A target morphism can have a finite metric defect or an infinite existence defect.

The categorical viewpoint supports:

- unitary versus measurement-assisted circuits,
- product-state versus entangled realizations,
- clean versus borrowed ancilla categories,
- symmetry-preserving versus symmetry-breaking channels,
- exact versus approximate completion.

The common invariant is still the cost of a restored target. What changes is the geometry of the path and the meaning of holonomy.

8.4.10 A general research dichotomy

The broad program separates into two tasks.

1. **Construction:** exhibit an ambient excursion whose repeated consumers produce a cheaper target.
2. **Certification:** prove that the restricted category cannot share the same obligations as efficiently.

Conjugation identities often solve the first task. The second is the difficult part and requires model-specific invariants such as interval-tree rank, symplectic flag length, algebraic degree, phase-polynomial rank, covariance constraints, or branch complexity.

CHAPTER 9

The complete linear core

9.1 Introduction

A CNOT circuit on n labeled wires represents an element of $\mathrm{GL}_n(\mathbb{F}_2) = \mathrm{SL}_n(\mathbb{F}_2)$, and minimizing its gate count is the word-metric problem for transvection generators. This formulation supports asymptotically efficient synthesis, architecture-aware elimination, computable lower bounds, and exact small-width optimization [40, 38, 32, 29, 21, 23, 25, 44]. Recent work has also produced stronger explicit lower-bound families and established NP-completeness of exact all-to-all CNOT distance [33, 27, 18]. The present paper studies a different comparison: the target is lower unitriangular, but the circuit is either required to respect that ordered structure or allowed to leave it temporarily.

Let R be a nonzero commutative ring with identity. A row transvection is

$$\tau_{t,s}(a) = I + aE_{t,s}, \qquad t \neq s, \qquad a \in R \setminus \{0\},$$

acting by

$$R_t \longleftarrow R_t + aR_s.$$

The transvection is *lower directed* when $s < t$. The lower-directed gates generate $\mathrm{U}_n(R)$. We compare

$$\ell_{\downarrow}(A) = \min\{\text{number of lower-directed transvections producing } A\}$$

with

$$\ell_*(A) = \min\{\text{number of arbitrary row transvections producing } A\}, \qquad A \in \mathrm{U}_n(R).$$

For a field $\mathbb{F}$, the first metric is intrinsic to the unipotent radical and the second is induced from the elementary group $\mathrm{E}_n(\mathbb{F}) = \mathrm{SL}_n(\mathbb{F})$.

The equality $\ell_{\downarrow}(A) = \ell_*(A)$ is tempting because the target and the obvious Gaussian-elimination paths are lower unitriangular. It is false. The failure first occurs at six wires over $\mathbb{F}_2$ and $\mathbb{F}_3$, and it is caused by a reusable temporary carrier rather than by a better ordering of the same lower gates.

Main results

The first result is entirely analytic and does not require a field.

Theorem 9.1 (Universal catalytic atom over rings)**.** *For every nonzero commutative ring R with identity, let*

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \in \mathrm{U}_6(R). \tag{9.1}$$

Then

$$\ell_{\downarrow}(C) = 7, \qquad \ell_{*}(C) = 6.$$

The six-gate ambient circuit uses row 1 as a dirty workspace. It first adds row 2 into row 1, exports the carrier e_1+e_2, changes the carrier to $e_0+e_1+e_2$, exports it twice, and finally removes the temporary e_2 component. No extra wire is introduced. The workspace is one of the target rows and is restored to its prescribed value.

The next result is computer assisted. All state spaces are finite, every transition is exact, and the complete source is included in the ancillary archive.

Theorem 9.2 (Exact binary threshold and census)**.** *Over $\mathbb{F}_2$:*

(i) $\ell_{\downarrow}(A) = \ell_{*}(A)$ *for every* $A \in \mathrm{U}_n(\mathbb{F}_2)$ *when* $n \le 5$.
(ii) *Among the* $2^{15} = 32768$ *matrices in* $\mathrm{U}_6(\mathbb{F}_2)$, *exactly twenty satisfy* $\ell_{\downarrow}(A) > \ell_{*}(A)$.
(iii) *Every defective target satisfies*

$$\ell_{\downarrow}(A) = 7, \qquad \ell_{*}(A) = 6.$$

No other distance pair is defective.

The twenty targets have a uniform combinatorial explanation.

Theorem 9.3 (Minimal catalytic classification)**.** *The twenty defective targets in $\mathrm{U}_6(\mathbb{F}_2)$ are exactly the targets generated by a simple one-catalyst word having three cross interactions, one phase interaction, and a nontrivial before/after split across the phase. The numbers arising from the three possible left/right auxiliary splits are*

$$6, \qquad 8, \qquad 6,$$

for a total of twenty.

The field-independent atom and exact ternary search also give the same finite threshold over $\mathbb{F}_3$.

Theorem 9.4 (Exact ternary threshold)**.** *For $n \le 5$ and every $A \in \mathrm{U}_n(\mathbb{F}_3)$,*

$$\ell_{\downarrow}(A) = \ell_{*}(A).$$

The matrix $C \in \mathrm{U}_6(\mathbb{F}_3)$ of Equation (9.1) has distances $(7, 6)$. Hence six is the first defective dimension over $\mathbb{F}_3$.

To pass from the first threshold to arbitrarily large connected examples, we introduce a multiscale rank certificate and a family that exports successive carrier states.

Theorem 9.5 (Connected snapshot family)**.** *Let $u \ge 1$, let $1 \le v \le u+1$, and choose distinct phases*

$$0 \le s_1 < \cdots < s_v \le u.$$

There is a connected target

$$A(u; s_1, \dots, s_v) \in \mathrm{U}_{u+v+2}(\mathbb{F})$$

with

$$\ell_{\downarrow}\big(A(u; s_1, \dots, s_v)\big) = u + 2v$$

and

$$\ell_{*}\big(A(u; s_1, \dots, s_v)\big) \le u + v + 2.$$

Hence its certified defect is at least $v - 2$.

Corollary 9.6 (Connected defects in every dimension). *For every field and every $n \geq 7$, there is a connected target $A_n \in \mathrm{U}_n(\mathbb{F})$ satisfying*

$$\ell_{\downarrow}(A_n) - \ell_*(A_n) \geq \left\lfloor \frac{n-5}{2} \right\rfloor.$$

The same construction remains effective under stringent restrictions on the ambient resource.

Theorem 9.7 (Two forbidden gates and weighted penalties). *For every field and every $n \geq 7$, there is a connected target $A_n \in \mathrm{U}_n(\mathbb{F})$ such that*

$$\ell_{\leq 0}(A_n) - \ell_{\leq 2}(A_n) \geq \left\lfloor \frac{n-5}{2} \right\rfloor,$$

where $\ell_{\leq q}$ permits at most q upper-directed gates. More generally, if lower gates have cost 1 and upper gates have cost $\lambda > 0$, then

$$\ell_{\downarrow}(A_n) - \ell_{\lambda}(A_n) \geq \left\lfloor \frac{n-1}{2} \right\rfloor - 2\lambda.$$

Thus no fixed upper-gate budget of two, and no fixed finite upper-gate penalty, suppresses catalytic escape asymptotically.

The same paths remain extremely close to the restricted subgroup. For a matrix X over a field, let X^+ be its strict upper part, let $r^+(X) = \operatorname{rank} X^+$, and let $b^+(X)$ be the largest upper displacement $j - i$ with $X_{ij} \neq 0$. If $\ell^{(r,b)}$ denotes the minimum length among circuits satisfying $r^+(X_t) \leq r$ and $b^+(X_t) \leq b$ at every prefix, then the connected family can be chosen so that

$$\ell^{(0,0)}(A_n) - \ell^{(1,1)}(A_n) \geq \left\lfloor \frac{n-5}{2} \right\rfloor.$$

Thus a rank-one excursion across one adjacent flag layer can create a linear saving.

At fixed dimension, however, defects occupy thin algebraic strata.

Theorem 9.8 (Generic rigidity and defect codimension). *Let K be an infinite field and put $N = \binom{n}{2}$. There is a nonempty Zariski-open subset $\Omega_n \subseteq \mathrm{U}_n$ with $\Omega_n(K) \neq \varnothing$ such that*

$$\ell_{\downarrow}(A) = \ell_*(A) = N \qquad (A \in \Omega_n(K)).$$

Moreover, after base change to an algebraic closure $\overline{K}$, for

$$\mathcal{D}_{n,\geq d}(\overline{K}) = \{A \in \mathrm{U}_n(\overline{K}) : \Delta(A) \geq d\},$$

one has

$$\operatorname{codim}_{(\mathrm{U}_n)_{\overline{K}}} \overline{\mathcal{D}_{n,\geq d}(\overline{K})} \geq d.$$

Corollary 9.9 (Finite-field sparsity). *Let $N = \binom{n}{2}$ and $M = n(n-1)$. For $1 \leq d \leq N$,*

$$\mathbb{P}_{A \in \mathrm{U}_n(\mathbb{F}_q)}[\Delta(A) \geq d] \leq \min\{1, M^{N-d} q^{-d}\}.$$

For fixed n, a uniformly random target therefore has defect zero with probability $1 - O_n(q^{-1})$ as $q \to \infty$.

Tensor products produce a stronger asymptotic separation.

Theorem 9.10 (Tensor-amplified snapshot family)**.** *Let $A = A(u; s_1, \dots, s_v)$ have dimension $m = u + v + 2$. For $N = m^r$,*

$$\ell_{\downarrow}(A^{\otimes r}) = r m^{r-1}(u + 2v),$$

while

$$\ell_{*}(A^{\otimes r}) \le r m^{r}.$$

Consequently,

$$\ell_{\downarrow}(A^{\otimes r}) - \ell_{*}(A^{\otimes r}) \ge \frac{v-2}{m \log_2 m} N \log_2 N.$$

Corollary 9.11 (Explicit $n \log n$ distortion)**.** *For*

$$D_n(\mathbb{F}) = \sup_{A \in \mathrm{U}_n(\mathbb{F})} \big(\ell_{\downarrow}(A) - \ell_{*}(A)\big),$$

one has, for every field and every n,

$$D_n(\mathbb{F}) \ge \frac{8}{21 \log_2 21} n \log_2 n - \frac{82}{105} n.$$

The coefficient is attained within the snapshot family by the twenty-one-dimensional seed

$$A(9; 0, 1, \dots, 9).$$

Conceptual interpretation

The ambient circuit is shorter because one forbidden-direction gate creates a carrier that is read by several later gates before it is cleaned. The saving is therefore a reuse effect. The relevant object is not an ancillary wire in the usual sense. It is a temporary excursion from a flag-preserving frame space into the full frame graph.

Over $\mathbb{F}_2$, the carrier states used by the canonical word form six of the seven points of $\mathrm{PG}(2,2)$. The four projective lines supported on those six points form a Pasch configuration. The same six vectors represent the graphic matroid $M(K_4)$ over every field; this regular-matroid structure isolates the characteristic-independent core of the atom. At the spectral level, the distance to a target is the first nonzero coefficient of an endpoint Green kernel. For C, the ambient kernel begins in degree six and the restricted kernel begins in degree seven.

The higher-dimensional families reveal three additional principles. First, an interval-tree certificate can make the restricted metric exactly additive across scales even when the target is connected. Second, two forbidden gates can support a linear saving, even if each forbidden gate carries any fixed finite penalty. Third, tensor products add normalized restricted complexity while ambient complexity remains subadditive. This converts one local carrier advantage into a superlinear global defect. Finally, the algebraic-geometric bounds show that these amplifiable atoms are exceptional at fixed dimension: a defect of size d imposes at least d algebraic codimensions. The resulting picture is one of algebraic rarity coupled with asymptotic amplification.

Scope

The results concern in-place reversible linear circuits on labeled rows. They do not by themselves imply a faster general CNOT synthesis algorithm, a lower bound for unrestricted XOR directed acyclic

graphs, or a result about integer multiplication. Their content is a precise comparison between a flag-preserving gate model and the ambient transvection model.

The paper is organized as follows. Section 9.2 fixes the circuit and metric conventions and gives the Schreier-holonomy interpretation. Section 9.3 proves the universal ring-valued atom. Section 9.4 develops the catalytic normal form and counts the twenty minimal motifs. Section 9.5 gives the exact binary and ternary enumerations and proves the search procedure complete. Section 9.6 gives the projective, regular-matroid, and Green-kernel formulations. Section 9.7 develops interval-tree certificates, exact cases through dimension nine, and connected families in every larger dimension. Section 9.8 proves bounded, weighted, and microscopic-excursion theorems. Section 9.9 proves generic rigidity, defect-codimension bounds, and finite-field sparsity. Section 9.10 proves block, recursive, and tensor amplification, introduces stable tensor capacity, and derives the $n \log n$ extremal bound. The appendices provide the exact target list, algorithms, and reproducibility data.

Proof and computation status

The universal atom, catalytic identities, interval-tree certificates, connected families, resource bounds, algebraic-rarity theorems, amplification results, and stable tensor limit are proved symbolically. Complete search is used only for the first-threshold censuses, the equality between the twenty computed defects and the twenty analytic motifs, and the exact ambient distances of the selected seven-, eight-, and nine-dimensional targets. Every computer-assisted statement is paired with a finite-state completeness argument and a reproducible implementation in the supplement.

9.2 Circuit model and metric comparison

9.2.1 Transvections and row conventions

Fix a nonzero commutative ring R with identity and an integer $n \geq 2$. Rows and columns are indexed by

$$0, 1, \ldots, n-1.$$

For $s \neq t$ and $a \in R \setminus \{0\}$, set

$$\tau_{t,s}(a) = I + aE_{t,s}.$$

Left multiplication by $\tau_{t,s}(a)$ performs

$$R_t \longleftarrow R_t + aR_s.$$

We call s the source and t the target. A circuit word

$$w = (\tau_{t_m,s_m}(a_m), \ldots, \tau_{t_1,s_1}(a_1))$$

produces

$$[w] = \tau_{t_m,s_m}(a_m) \cdots \tau_{t_1,s_1}(a_1).$$

Let

$$S_n^{\downarrow}(R) = \{\tau_{t,s}(a) : 0 \leq s < t < n,\ a \in R \setminus \{0\}\}$$

and

$$S_n^{*}(R) = \{\tau_{t,s}(a) : 0 \leq s, t < n,\ s \neq t,\ a \in R \setminus \{0\}\}.$$

Both sets are symmetric because

$$\tau_{t,s}(a)^{-1} = \tau_{t,s}(-a).$$

Let

$$\mathrm{E}_n(R) = \langle S_n^*(R) \rangle.$$

When $R = \mathbb{F}$ is a field and $n \geq 2$, elementary elimination gives

$$\mathrm{E}_n(\mathbb{F}) = \mathrm{SL}_n(\mathbb{F}).$$

The ambient metric below is therefore the transvection word metric in $\mathrm{SL}_n(\mathbb{F})$, viewed inside $\mathrm{GL}_n(\mathbb{F})$. When $\mathbb{F} = \mathbb{F}_2$, one has $\mathrm{SL}_n(\mathbb{F}_2) = \mathrm{GL}_n(\mathbb{F}_2)$, every generator is an involution, and it is exactly a CNOT gate under the usual row-action convention.

Lemma 9.12. *The set $S_n^{\downarrow}(R)$ generates $\mathrm{U}_n(R)$. Every prefix of a lower-directed circuit belongs to $\mathrm{U}_n(R)$.*

Proof. A lower-directed transvection is lower unitriangular, and products of lower-unitriangular matrices remain lower unitriangular. Conversely, ordinary elimination of the strict lower entries, processed in a compatible order, expresses every element of $\mathrm{U}_n(R)$ as a product of lower-directed transvections. □

Definition 9.13 (Intrinsic and ambient lengths)**.** For $A \in \mathrm{U}_n(R)$, define

$$\ell_{\downarrow}(A) = |A|_{S_n^{\downarrow}(R)}, \qquad \ell_*(A) = |A|_{S_n^*(R)}.$$

The *catalytic defect* is

$$\Delta(A) = \ell_{\downarrow}(A) - \ell_*(A) \geq 0.$$

The inequality follows because $S_n^{\downarrow}(R) \subseteq S_n^*(R)$. The central question is whether the inclusion can be strict on a lower-unitriangular target.

9.2.2 Frame and flag interpretation

For the geometric interpretation in this subsection, assume $R = \mathbb{F}$ is a field. An invertible matrix is an ordered basis

$$(R_0, \ldots, R_{n-1})$$

of $\mathbb{F}^n$. A transvection replaces one basis vector by its sum with a nonzero multiple of another. Let

$$V_i = \mathrm{span}(e_0, \ldots, e_i).$$

A matrix lies in $\mathrm{U}_n(\mathbb{F})$ exactly when

$$R_i \in e_i + V_{i-1}$$

for every i. Thus lower circuits move among frames compatible with the standard complete flag

$$V_0 \subset V_1 \subset \cdots \subset V_{n-1}.$$

An upper-directed gate leaves this fiber of the flag map. A strict defect means that the flag-compatible frame subgraph is not geodesically convex in the full transvection graph.

9.2.3 Schreier holonomy

Set

$$G = \mathrm{E}_n(R), \qquad H = \mathrm{U}_n(R).$$

Consider the right Schreier graph on the left cosets $H\backslash G$: a generator $s \in S_n^*(R)$ gives an edge

$$Hg \longrightarrow Hgs.$$

A word representing $h \in H$ traces a based loop at the coset H. Its label product is the holonomy h.

Proposition 9.14 (Holonomy-compression interpretation)**.** *For $h \in H$, the ambient length $\ell_*(h)$ is the least length of a labeled based loop in $H\backslash G$ with holonomy h. The restricted length $\ell_\downarrow(h)$ is the least length among such words whose every prefix remains in H. Consequently,*

$$\Delta(h) = \textit{intrinsic flag-preserving cost} - \textit{shortest ambient holonomy-loop cost}.$$

Proof. A word $w = s_1 \cdots s_k$ ends at the coset Hw. It is a based loop exactly when $w \in H$, and its label product is then the represented element of H. Minimizing over all transvection words representing h gives $\ell_*(h)$. Requiring every generator to be lower directed keeps every prefix in H and gives $\ell_\downarrow(h)$. □

A catalytic shortcut is therefore a nonconstant quotient excursion whose accumulated holonomy is more expensive to produce inside the base fiber.

9.2.4 Two elementary counting lemmas

The proofs of the universal atom use only target counts and pivot support.

Lemma 9.15 (Target count)**.** *Suppose a circuit starts at the identity and finishes at A.*

(i) *If row i of A differs from e_i, then row i is targeted at least once.*
(ii) *If row i of A equals e_i and is targeted, then it is targeted at least twice.*

Proof. A row changes only when it is targeted. Every row of an invertible matrix over R is unimodular. If $aR_s = 0$ for such a row, then multiplying a unimodular relation for R_s by a gives $a = 0$. Hence a gate with nonzero coefficient changes its target by a nonzero vector, so one targeting cannot leave that row equal to its initial value. □

Lemma 9.16 (One-target source condition)**.** *Let $W \leq R^n$. Suppose row t starts at e_t, is targeted exactly once, and finishes at*

$$e_t + w, \qquad w \in W,$$

where at least one coordinate of w is a unit of R. At the moment of its unique targeting, the gate coefficient is a unit and the source row is a unit multiple of w.

Proof. If the unique gate is $R_t \leftarrow R_t + aR_s$, then $aR_s = w$. Choose a coordinate j for which w_j is a unit. Since $a(R_s)_j = w_j$, the factor a is a unit in the commutative ring R. Hence $R_s = a^{-1}w$. □

The second lemma is especially restrictive when W is a low-dimensional source space and output rows carry unique pivot coordinates outside W.

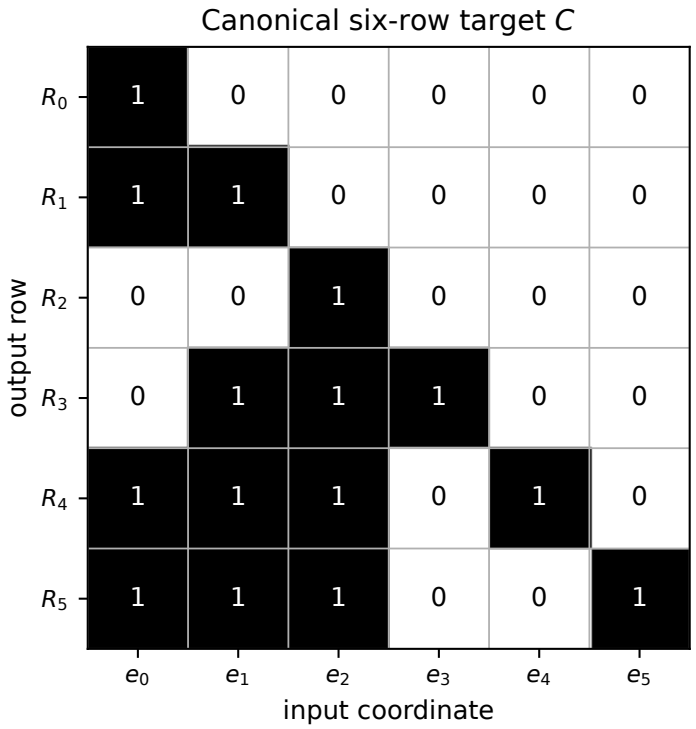


Figure 9.1. The canonical target C. Black cells are nonzero. Rows 0 and 2 are restored exactly, while rows $3, 4, 5$ require two distinct additions v and w from the source space W.

9.3 The universal six-row atom

Let R be a nonzero commutative ring with identity, and let

$$v = e_1 + e_2, \qquad w = e_0 + e_1 + e_2, \qquad W = \operatorname{span}(e_0, e_1, e_2).$$

The target C of Equation (9.1) has rows

$$\begin{aligned} R_0 &= e_0, & R_1 &= e_1 + e_0, & R_2 &= e_2, \\ R_3 &= e_3 + v, & R_4 &= e_4 + w, & R_5 &= e_5 + w. \end{aligned}$$

9.3.1 A six-gate ambient word

Proposition 9.17. *For every nonzero commutative ring R with identity,*

$$\ell_*(C) \leq 6.$$

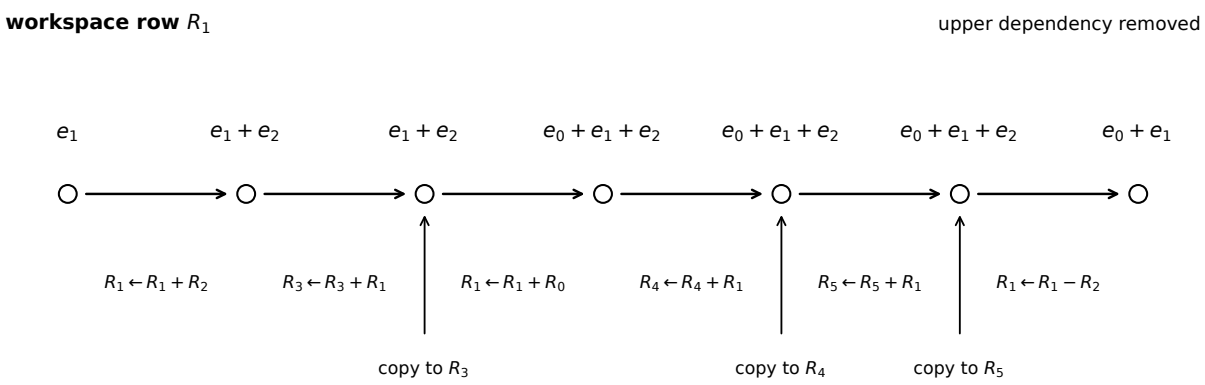


Figure 9.2. Carrier timeline for the six-gate ambient word. The two upper-directed operations are the opening and closing gates on row 1. The state $e_1 + e_2$ is exported once and $e_0 + e_1 + e_2$ is exported twice.

Proof. Apply the following six gates:

$$R_1 \leftarrow R_1 + R_2, \tag{9.2}$$
$$R_3 \leftarrow R_3 + R_1, \tag{9.3}$$
$$R_1 \leftarrow R_1 + R_0, \tag{9.4}$$
$$R_4 \leftarrow R_4 + R_1, \tag{9.5}$$
$$R_5 \leftarrow R_5 + R_1, \tag{9.6}$$
$$R_1 \leftarrow R_1 - R_2. \tag{9.7}$$

After Equation (9.2), row 1 contains v. Gate Equation (9.3) copies v to row 3. Gate Equation (9.4) changes the carrier to w, which is copied to rows 4 and 5 by Equations (9.5) and (9.6). Gate Equation (9.7) removes the temporary e_2 component, leaving $e_1 + e_0$. Every row now agrees with C. □

9.3.2 A seven-gate lower word

Proposition 9.18. *For every nonzero commutative ring R with identity,*

$$\ell_{\downarrow}(C) \leq 7.$$

Proof. The lower-directed word

$$\begin{aligned}
&R_1 \leftarrow R_1 + R_0,\\
&R_2 \leftarrow R_2 + R_1,\\
&R_3 \leftarrow R_3 - R_0,\\
&R_3 \leftarrow R_3 + R_2,\\
&R_4 \leftarrow R_4 + R_2,\\
&R_5 \leftarrow R_5 + R_2,\\
&R_2 \leftarrow R_2 - R_1
\end{aligned}$$

produces C. After the first two gates, row 2 contains w. Rows 4 and 5 copy that state. The two gates on row 3 cancel the unwanted e_0 and leave v. The final gate restores row 2. □

9.3.3 Restricted lower bound

Proposition 9.19. *For every nonzero commutative ring R with identity,*

$$\ell_{\downarrow}(C) \geq 7.$$

Proof. Rows $1, 3, 4, 5$ differ from their initial values, so they must each be targeted. Assume for contradiction that a lower-directed word of length at most six produces C.

First suppose row 2 is never targeted. Row 0 remains e_0, row 2 remains e_2, and every state ever assumed by row 1 lies in $\mathrm{span}(e_0, e_1)$ because the only possible source below row 1 is row 0. Every output row retains coefficient 1 on its own pivot coordinate in a lower-directed circuit, so no output row can lie in W. Hence no available source row is a unit multiple of either

$$v = e_1 + e_2 \quad \text{or} \quad w = e_0 + e_1 + e_2.$$

By Lemma 9.16, each of rows $3, 4, 5$ must be targeted at least twice. Together with the required targeting of row 1, this gives at least seven gates.

Now suppose row 2 is targeted. Since its final value is e_2, Lemma 9.15 forces at least two targetings of row 2. A six-gate word must therefore target row 2 exactly twice and each of rows $1, 3, 4, 5$ exactly once. Each output row $3, 4, 5$ must receive its complete addition in its unique targeting. Row 0 cannot supply v or w, and row 1 never contains an e_2 component. Therefore row 2 must supply all three outputs, up to unit scaling. But two targetings of row 2 expose at most one noninitial intermediate state before row 2 returns to e_2, while the outputs require both distinct states v and w. This is impossible. □

9.3.4 Ambient lower bound

Proposition 9.20. *For every nonzero commutative ring R with identity,*

$$\ell_{*}(C) \geq 6.$$

Proof. Assume that an ambient word of length at most five produces C. Rows $1, 3, 4, 5$ must each be targeted. Rows 0 and 2 are unchanged. If either were targeted, it would require at least two targetings, giving at least six gates in total. Thus rows 0 and 2 are never targeted. If all four changed rows are targeted once, each has only its initial and final state. No such source is a unit multiple of $v = e_1 + e_2$ or $w = e_0 + e_1 + e_2$, so the output rows cannot be completed. In the only remaining case, exactly one of rows $1, 3, 4, 5$ is targeted twice and the other three are targeted once.

Project the rows to the quotient module R^6/W. Rows $0, 1, 2$ have zero image, while rows $3, 4, 5$ begin and end at the three distinct quotient basis vectors of the free quotient module. An output row targeted exactly once must therefore use a source lying in W, by Lemma 9.16.

Suppose first that row 1 is the twice-targeted row. The three output rows are each targeted once. Rows 0 and 2 supply only scalar multiples of e_0 and e_2. Row 1 begins at e_1, has one intermediate state, and ends at $e_1 + e_0$. Thus at most one of the two required unit-multiple classes of v and w can appear as an available source. The three outputs require both, a contradiction.

Suppose instead that an output row, say row $r \in \{3, 4, 5\}$, is targeted twice. The other two outputs and row 1 are targeted once. We claim that row r cannot lie in W between its two targetings. To cancel

its unique quotient pivot e_r at the first targeting, the source would already have to contain an e_r component. Before row r is first targeted, the only way another row can acquire such a component is to be targeted using row r as source. That other row is targeted only once and would then retain an unwanted e_r component in its final value. Therefore row r never becomes a W-valued carrier. The only W-valued source states are rows 0 and 2 and the two states of singly targeted row 1, namely its initial and final states. None supplies both v and w. The two singly targeted outputs cannot be completed, again a contradiction. □

Proof of Theorem 9.1. Combine Propositions 9.17 to 9.20. □

Remark 9.21. The proof uses only unimodular rows, unit coordinates, and pivot support. It does not use division or a characteristic assumption. In characteristic two the closing subtraction equals addition.

Corollary 9.22 (Failure of geodesic convexity)**.** *For every field $\mathbb{F}$ and every $n \geq 6$, the lower-unitriangular frame subgraph is not geodesically convex in the transvection Cayley graph of $\mathrm{E}_n(\mathbb{F}) = \mathrm{SL}_n(\mathbb{F})$.*

Proof. Embed C as $C \oplus I_{n-6}$. A six-gate ambient word acts on the first six rows. In a lower-directed circuit, rows with index at least six cannot be used as sources for targets among the first six rows, so the restricted lower bound of seven persists. □

For the remainder of the paper, unless explicitly stated otherwise, the coefficient domain is a field $\mathbb{F}$.

9.4 Catalytic normal forms and the six-role threshold

The universal atom is one instance of a root-conjugation mechanism. This section isolates that mechanism and explains the count twenty.

9.4.1 Root conjugation identities

Let $p < q < y$ and $x < p < q$. Direct multiplication gives

$$\tau_{p,q}(-a)\,\tau_{y,p}(b)\,\tau_{p,q}(a) = \tau_{y,p}(b)\,\tau_{y,q}(ab), \tag{9.8}$$

and

$$\tau_{p,q}(-a)\,\tau_{q,x}(b)\,\tau_{p,q}(a) = \tau_{q,x}(b)\,\tau_{p,x}(-ab). \tag{9.9}$$

A single forbidden upper transvection can therefore turn one lower interaction into two lower effects. Interactions of the forms $x \to p$ and $q \to y$ commute through the excursion without duplication. We call the duplicated interactions *cross interactions* and the unduplicated interactions *phase interactions*.

Definition 9.23 (Simple one-catalyst word)**.** A simple one-catalyst word has the form

$$\tau_{p,q}(-a)\,h_m \cdots h_1\,\tau_{p,q}(a), \qquad p < q,$$

where every h_i is lower directed and is one of

$$\tau_{q,x}(b), \quad \tau_{p,x}(b), \quad \tau_{y,p}(b), \quad \tau_{y,q}(b)$$

with $x < p < q < y$. Auxiliary rows are distinct unless stated otherwise.

The circuit is read from right to left in the matrix product and from left to right as a gate list. The opening and closing operations temporarily place the q component in row p.

Proposition 9.24 (Cross-interaction accounting)**.** *Let a simple one-catalyst word have m middle interactions, of which c are cross interactions. Its target admits a lower-directed realization of length at most*

$$m + c.$$

Consequently, if the ambient word of length $m + 2$ is strictly shorter than every lower word, then

$$c \geq 3.$$

Proof. Conjugate each middle generator using Equations (9.8) and (9.9). A phase interaction remains one lower transvection, while a cross interaction becomes a product of two lower transvections. This gives length at most $m + c$. If $c \leq 2$, then $m + c \leq m + 2$, so the ambient excursion cannot be strictly better. □

Three consumers are therefore the first possible point at which the cost two of opening and closing can be amortized.

9.4.2 Why an all-cross word is insufficient

Within the distinct-leaf normal form, cross count alone is not enough.

Lemma 9.25 (All-cross mirror)**.** *Suppose a simple one-catalyst word contains only cross interactions, with left inputs X and right outputs Y. Then its target has a lower-directed realization with the same number of gates.*

Proof. Write the opening coefficient as $a \in \mathbb{F}^{\times}$, the left-update coefficients as b_x, and the right-read coefficients as c_y. Since the all-cross middle gates commute past one another, the ambient gate list may be written

$$R_p \leftarrow R_p + aR_q; \quad [R_q \leftarrow R_q + b_x R_x]_{x\in X}; \quad [R_y \leftarrow R_y + c_y R_p]_{y\in Y}; \quad R_p \leftarrow R_p - aR_q.$$

Put $B = \sum_{x\in X} b_x e_x$. Its final nontrivial rows are

$$R_p = e_p - aB, \qquad R_q = e_q + B, \qquad R_y = e_y + c_y e_p + ac_y e_q \quad (y \in Y).$$

The following lower-directed gate list produces the same rows:

$$R_q \leftarrow R_q + a^{-1}R_p; \quad [R_y \leftarrow R_y + ac_y R_q]_{y\in Y}; \quad [R_p \leftarrow R_p - ab_x R_x]_{x\in X}; \quad R_q \leftarrow R_q - a^{-1}R_p.$$

Indeed, after the first gate $R_q = e_q + a^{-1}e_p$, so the output gates create exactly $c_y e_p + ac_y e_q$. The source updates then leave $R_p = e_p - aB$, and the final gate changes R_q to $e_q + B$. Both words have $|X| + |Y| + 2$ gates. □

Thus a minimal strict shortcut needs both positive catalytic excess and a phase switch that makes temporal order matter.

Proposition 9.26 (Six-role threshold in the simple model)**.** *A strict shortcut in the distinct-leaf simple one-catalyst model requires at least six row roles:*

$$2 \textit{ catalyst rows} + 3 \textit{ cross interactions} + 1 \textit{ phase interaction}.$$

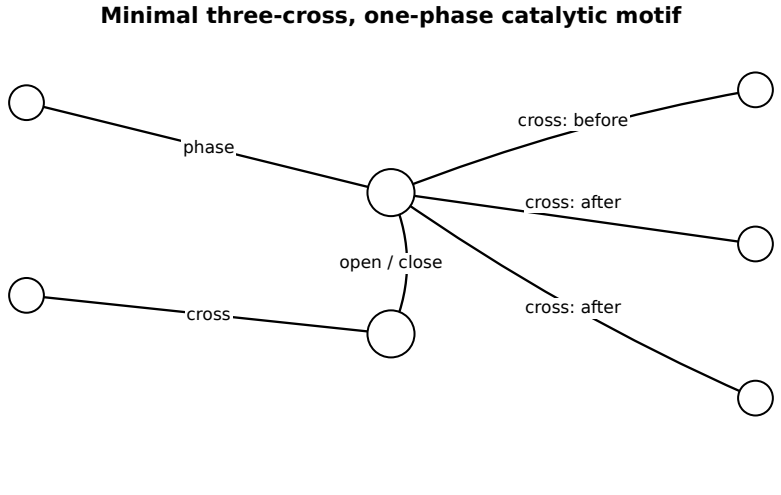


Figure 9.3. The minimal one-catalyst motif. Cross interactions reuse the opened carrier. The phase gate separates consumers that observe two different carrier states and prevents the all-cross mirror.

The bound is attained by the matrix C.

Proof. By Proposition 9.24, at least three cross interactions are required. By Lemma 9.25, at least one phase interaction is required. With distinct leaves these use four auxiliary rows in addition to the two catalyst rows. The word of Proposition 9.17 has exactly this form. □

9.4.3 Enumeration of the minimal motifs

Fix six ordered rows. The catalyst pair must be adjacent in a minimal six-role word. Write

$$p < q = p + 1,$$

let L be the set of rows left of p, and let R be the set of rows right of q. Then

$$|L| + |R| = 4, \qquad |L|, |R| > 0.$$

There are two orientations.

Left phase. Choose a phase leaf $x \in L$. The remaining $|L|-1$ left leaves interact by $u \to q$. Every right leaf is a cross read $p \to y$. To make the phase effective, the right leaves are divided into a nonempty set read before $x \to p$ and a nonempty set read after it. The number of targets is

$$|L|\big(2^{|R|} - 2\big).$$

Right phase. Choose a phase leaf $y \in R$. Every other right leaf is a cross read. The left updates into q are divided nontrivially before and after the phase gate $q \to y$. The number is

$$|R|\big(2^{|L|} - 2\big).$$

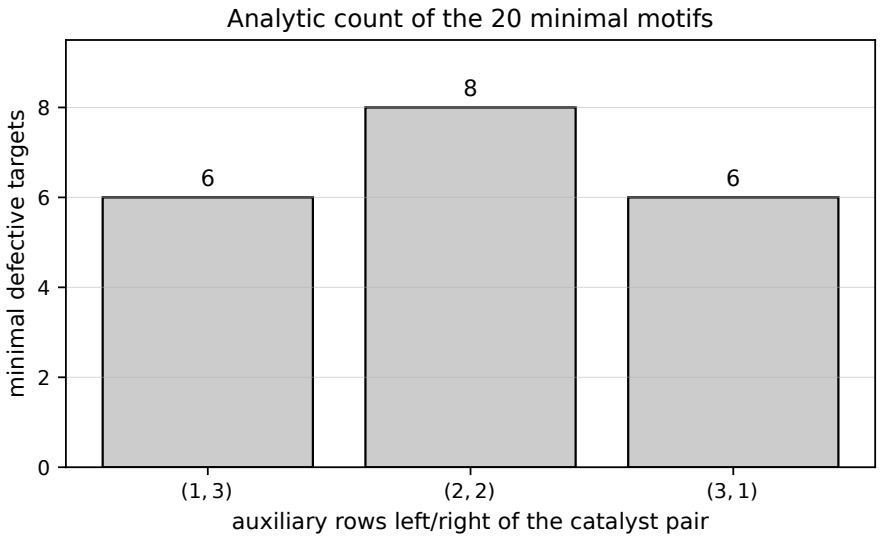


Figure 9.4. The analytic count of the twenty minimal catalytic targets.

Proposition 9.27 (Motif count)**.** *For a split $(L, R) = (|L|, |R|)$ with $L + R = 4$, the number of minimal phase-separated targets is*

$$N_{L,R} = L(2^R - 2) + R(2^L - 2).$$

For the three possible splits,

$$N_{1,3} = 6, \qquad N_{2,2} = 8, \qquad N_{3,1} = 6.$$

Hence there are twenty targets.

Proof. The preceding left-phase and right-phase constructions are disjoint because the phase row lies on opposite sides of the catalyst. Within each orientation, the final support records the phase leaf and the before/after subset, so distinct choices give distinct targets. Summing gives the formula and the stated values. □

Every target counted by Proposition 9.27 has an explicit six-gate ambient word and a seven-gate lower word obtained by eliminating the three cross interactions. The nontrivial statement is that there are no other defects in dimension six and that none of these targets has a lower word of length six. That is the exact classification established in Section 9.5.

9.5 Exact finite enumeration

This section describes the computer-assisted part of the paper. The computations use integer encodings only. No floating-point decisions occur.

9.5.1 Restricted breadth-first search

Over $\mathbb{F}_2$, an element of $\mathrm{U}_n(\mathbb{F}_2)$ is determined by its $n(n-1)/2$ strict lower entries. For $n = 6$ there are $2^{15} = 32768$ states. Breadth-first search from the identity using the fifteen lower transvections gives every restricted distance and the restricted diameter.

For $n = 6$, the restricted diameter is eight. Therefore every lower-unitriangular target has ambient distance at most eight. This makes a radius-four meet-in-the-middle search complete.

9.5.2 Ambient radius-four ball and right-coset pairing

Let

$$B_4 = \{g \in \mathrm{GL}_6(\mathbb{F}_2) : \ell_*(g) \le 4\}.$$

Exact breadth-first search gives

$$|B_4| = 110626.$$

Group B_4 by right cosets of $\mathrm{U}_6(\mathbb{F}_2)$. If a, b lie in the same right coset, then

$$b^{-1}a \in \mathrm{U}_6(\mathbb{F}_2),$$

and

$$\ell_*(b^{-1}a) \le \ell_*(b) + \ell_*(a) \le 8.$$

The implementation uses a canonical right-coset key obtained by reducing matrix columns from right to left. Right multiplication by a lower-unitriangular matrix changes column j only by a combination of columns with larger index, so this reduction is constant on right cosets and separates them.

Lemma 9.28 (Meet-in-the-middle completeness)**.** *For every $u \in \mathrm{U}_6(\mathbb{F}_2)$,*

$$\ell_*(u) = \min\{\ell_*(a) + \ell_*(b) : a, b \in B_4,\ b^{-1}a = u\}.$$

Proof. Every pair on the right gives a word for u, so the right-hand side is at least $\ell_*(u)$. Conversely, $\ell_*(u) \le 8$. Take a shortest path from the identity to u and split it at a midpoint. The prefix endpoint a has distance at most four. The inverse of the remaining suffix is an element b of distance at most four, and $u = b^{-1}a$. Hence the minimum is at most $\ell_*(u)$. □

The radius-four ball splits into 6948 right-coset buckets. The largest bucket has 3775 elements. The computation checks 45178326 ordered pairs and recovers an ambient distance for every one of the 32768 targets.

9.5.3 The exact binary census

Proof of Theorem 9.2. For $n = 2, 3, 4, 5$, complete restricted breadth-first search and complete ambient meet-in-the-middle search return zero defects. The state counts are respectively $2, 8, 64, 1024$. For $n = 6$, Lemma 9.28 proves that the ambient calculation is exhaustive. The resulting distribution is Table 9.1; its counts sum to 32768, no target is missing, and the only strict pair is $(7, 6)$ with multiplicity twenty. □

Proof of Theorem 9.3. The analytic generator of Proposition 9.27 produces twenty distinct targets without reference to the distance census. Exact set comparison shows that these twenty targets are precisely the twenty off-diagonal targets in Table 9.1. Each therefore has distances $(7, 6)$, and no other dimension-six target is defective. □

9.5.4 Ternary and additional field checks

For $\mathbb{F}_3$, every nonzero scaled transvection is included. Complete searches give the data in Table 9.2.

Table 9.1. Complete distance distribution in $\mathrm{U}_6(\mathbb{F}_2)$.

$\ell_\downarrow$	ℓ_*	number of targets
0	0	1
1	1	15
2	2	125
3	3	720
4	4	2914
5	5	7851
6	6	12656
7	6	20
7	7	7926
8	8	540

Table 9.2. Complete no-defect computations below dimension six.

field	n	$\lvert \mathrm{U}_n(\mathbb{F})\rvert$	restricted diameter
$\mathbb{F}_2$	2	2	1
$\mathbb{F}_2$	3	8	2
$\mathbb{F}_2$	4	64	4
$\mathbb{F}_2$	5	1024	6
$\mathbb{F}_3$	2	3	1
$\mathbb{F}_3$	3	27	3
$\mathbb{F}_3$	4	729	5
$\mathbb{F}_3$	5	59049	7

The $n = 5$ ternary computation uses radius-three and radius-four balls, checks 38415205 right-coset pairs, and recovers all 59049 targets. It finds no defect. The field-independent Theorem 9.1 supplies a defect at $n = 6$, proving Theorem 9.4. Separate exact searches over $\mathbb{F}_3$ and $\mathbb{F}_5$ verify the distances of the canonical atom directly. The $\mathbb{F}_5$ calculation is an audit of the atom, not a complete threshold computation below dimension six.

9.5.5 Verification status

The analytic proofs of Theorem 9.1 and Corollary 9.52 do not depend on computation. The claims in Theorems 9.2 to 9.4 depend on exact finite enumeration. The source archive contains:

- the C++ programs for every breadth-first and meet-in-the-middle search;
- the Python program that independently generates the twenty motifs and compares sets;
- symbolic checks for the bounded-resource formulas and the totally unimodular $M(K_4)$ representation;
- exact stdout files and JSON packet data;
- SHA-256 checksums;
- build and reproduction commands.

The programs were rebuilt from source and rerun independently when this manuscript was assembled; the outputs matched byte for byte.

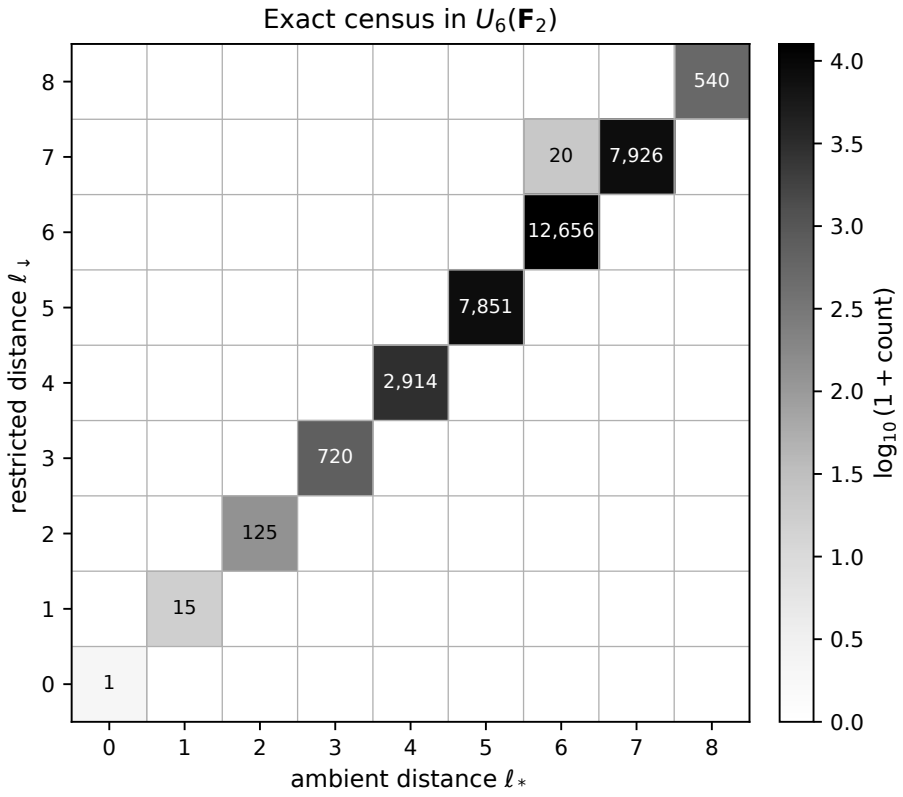


Figure 9.5. Complete binary dimension-six census. The only off-diagonal cell contains the twenty targets with distance pair $(7, 6)$.

9.6 Projective, matroidal, and spectral structure

9.6.1 The flagged Pasch packet

Over $\mathbb{F}_2$, write

$$x = e_0, \qquad p = e_1, \qquad q = e_2.$$

The canonical ambient word uses the six nonzero vectors

$$x, \quad p, \quad q, \quad p+q, \quad p+q+x, \quad p+x$$

inside $\mathrm{span}(x, p, q) \cong \mathbb{F}_2^3$. The only missing nonzero vector is $x + q$.

In $\mathrm{PG}(2, 2)$, three distinct points form a line exactly when their vector representatives sum to zero. The six points support the four lines

$$\{x, p, x+p\}, \quad \{x, p+q, x+p+q\}, \quad \{p, q, p+q\}, \quad \{x+p, q, x+p+q\}.$$

Every point lies on exactly two of these lines. This is the Pasch configuration, equivalently the Fano plane with one point and its three incident lines removed from the full incidence structure.

The configuration is flagged because the rows have fixed roles. Points x and q are stationary sources, p is the carrier, and the remaining points occur as carrier states. Forgetting those roles loses the circuit ordering information.

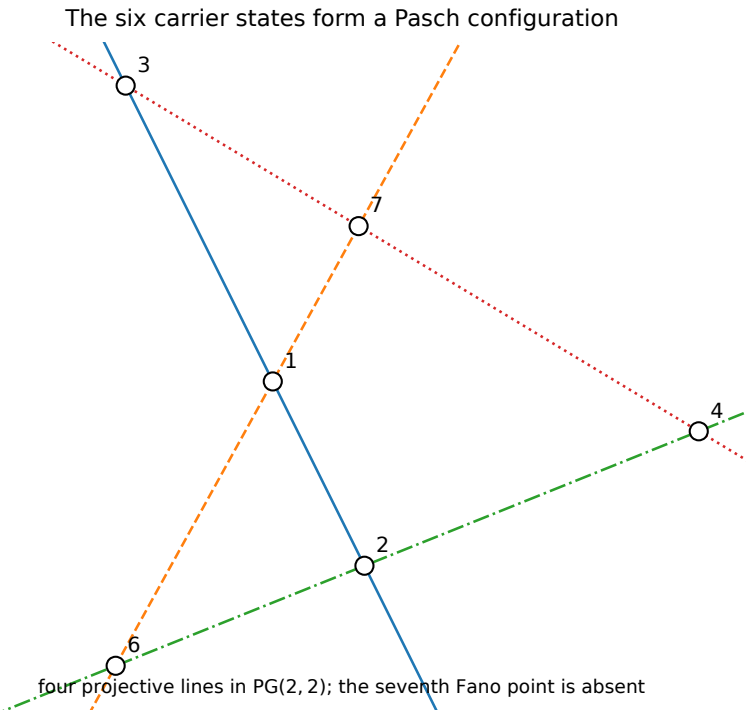


Figure 9.6. A real drawing of the four-line Pasch incidence structure. The labels are the binary point encodings $1, 2, 3, 4, 6, 7$; point 5 is the missing Fano point.

9.6.2 The regular matroid $M(K_4)$

The six carrier states admit a field-independent representation. In the ordered basis (x, p, q), place them as the columns of

$$B = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \tag{9.10}$$

corresponding respectively to

$$x, \quad p, \quad q, \quad x+p, \quad p+q, \quad x+p+q.$$

Proposition 9.29 (Regular-matroid core)**.** *Every square minor of B belongs to $\{0, \pm 1\}$, so B is totally unimodular. Hence the column-dependence matroid is the same over every field. Under the edge labeling*

state	p	x	$x+p$	$p+q$	q	$x+p+q$
K_4 *edge*	12	13	23	14	24	34,

it is the graphic matroid $M(K_4)$.

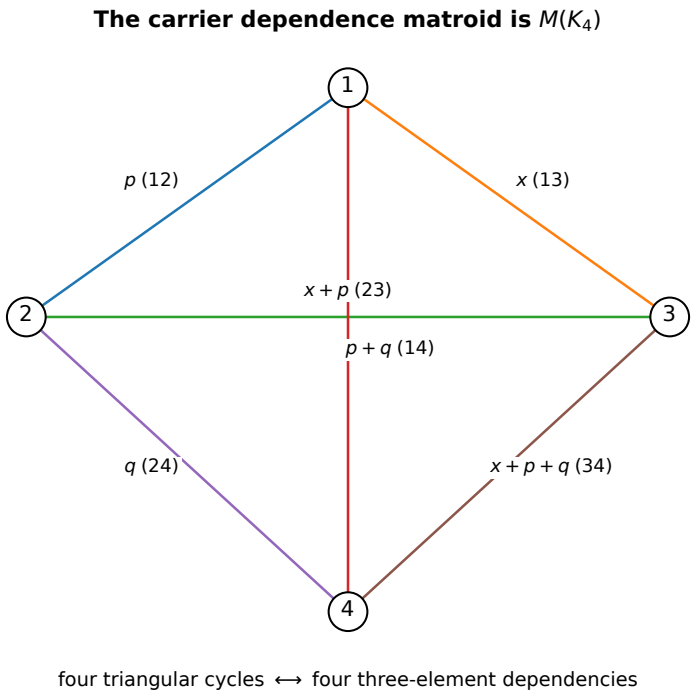


Figure 9.7. The six carrier states as the edges of K_4. Each triangular cycle is one of the four three-element dependencies in the Pasch configuration. The resulting regular matroid is representable over every field.

Proof. A direct calculation of all square minors of B gives only $0, 1$, and -1. Thus every subdeterminant has characteristic-independent vanishing, and a column set is independent over one field exactly when it is independent over every field. The four minimal dependent triples are

$$\{x, p, x+p\}, \quad \{p, q, p+q\}, \quad \{x, p+q, x+p+q\}, \quad \{q, x+p, x+p+q\}.$$

Under the displayed edge labeling, these are precisely the four triangular cycles of K_4. They determine the rank-three cycle matroid $M(K_4)$, which is regular because it has a totally unimodular representation; see, for example, [37]. □

The field-independent circuit theorem is stronger than matroid representability alone, because it also depends on the ordering and the designated carrier role. Nevertheless, Proposition 9.29 explains why the local dependence pattern survives unchanged in every characteristic.

9.6.3 Distance as first arrival

Let G be a finite group and $S = S^{-1}$ a symmetric generating set. On $\ell^2(G)$, let λ be the left regular representation,

$$\lambda(s)\delta_h = \delta_{sh},$$

and define

$$A_S = \sum_{s \in S} \lambda(s).$$

Table 9.3. The five endpoint Green-signature packets. Each row occurs for four defective targets.

	restricted coefficients				ambient coefficients		
packet	$[z^7]$	$[z^8]$	$[z^9]$	$[z^{10}]$	$[z^6]$	$[z^7]$	$[z^8]$
1	326	15430	316118	7576819	4	570	23318
2	326	15536	321870	7727183	4	570	23318
3	543	25080	510703	10824940	2	776	39454
4	543	25639	511524	10984988	2	776	39454
5	543	26091	511819	11121486	2	776	39454

For $g \in G$, define the endpoint Green series

$$\mathcal{G}_{S,g}(z) = \left\langle \delta_g, (I - zA_S)^{-1}\delta_e \right\rangle .$$

Proposition 9.30 (First-arrival identity)**.** *The coefficient of z^k in $\mathcal{G}_{S,g}(z)$ is the number of length-k words in S with product g. Hence*

$$d_S(e, g) = \operatorname{ord}_{z=0} \mathcal{G}_{S,g}(z).$$

Proof. Formally,

$$(I - zA_S)^{-1} = \sum_{k\geq 0} z^k A_S^k.$$

The matrix coefficient $\langle \delta_g, A_S^k \delta_e \rangle$ counts k-step walks from e to g, which are exactly length-k words with product g. The first nonzero coefficient occurs at the word distance. □

If

$$A_S = \sum_{\lambda} \lambda \Pi_\lambda$$

is the spectral resolution, then

$$\mathcal{G}_{S,g}(z) = \sum_{\lambda} \frac{\langle \delta_g, \Pi_\lambda \delta_e \rangle}{1 - \lambda z}.$$

Thus eigenvalues alone do not determine first arrival. The endpoint matrix coefficients of the spectral projectors are essential.

For the canonical target C, exact word counts begin

$$\mathcal{G}_{\downarrow,C}(z) = 543z^7 + 25080z^8 + 510703z^9 + 10824940z^{10} + \cdots ,$$
$$\mathcal{G}_{*,C}(z) = 2z^6 + 776z^7 + 39454z^8 + \cdots .$$

The metric defect is the valuation drop

$$\operatorname{ord}_0 \mathcal{G}_{\downarrow,C} - \operatorname{ord}_0 \mathcal{G}_{*,C} = 1.$$

Across the twenty defects, the truncated endpoint data split into five Green-signature classes, each of multiplicity four. The signatures are listed in Table 9.3.

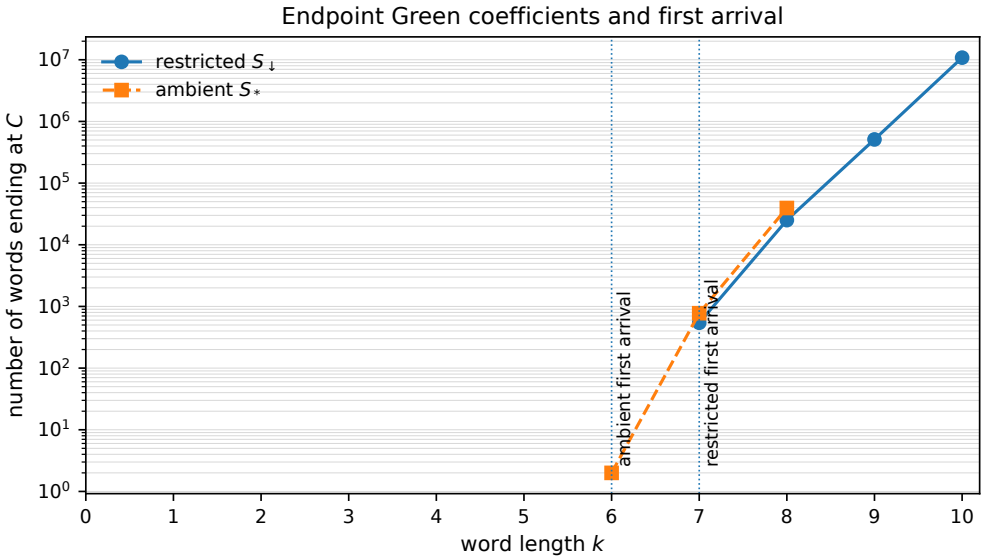


Figure 9.8. Exact endpoint word counts for the canonical target. The ambient walk first arrives at length six. The restricted walk first arrives at length seven.

9.7 Interval-tree certificates and connected higher-dimensional families

From this point through the rank and tensor arguments, the coefficient domain is a field $\mathbb{F}$. The six-row atom is the first defect, but it is not an isolated low-dimensional accident. This section gives a field-independent family of connected targets in every dimension at least seven. The lower bound is certified by a multiscale rank potential adapted to the ordered flag.

9.7.1 Interval-tree rank potential

Definition 9.31 (Interval tree)**.** An *interval tree* on the ordered coordinates $0, 1, \dots, n-1$ is a rooted binary tree whose leaves occur in that order and whose every node is a contiguous interval. If an internal node v splits its interval into consecutive left and right children L_v and R_v, define

$$\Phi_T(A) = \sum_{v \in \mathrm{Int}(T)} \operatorname{rank} A[R_v, L_v], \qquad A \in \mathrm{U}_n(\mathbb{F}). \tag{9.11}$$

Lemma 9.32 (One-cut locality)**.** *Let $X \in \mathrm{U}_n(\mathbb{F})$ and let X' be obtained from X by one lower-directed transvection. Then*

$$\left|\Phi_T(X') - \Phi_T(X)\right| \le 1.$$

Proof. Suppose the gate is $s \to t$ with $s < t$. Let v_0 be the lowest node of T whose two children separate s and t. The order of the leaves forces $s \in L_{v_0}$ and $t \in R_{v_0}$. At this one node, a single row of the cross block $X[R_{v_0}, L_{v_0}]$ changes, so its rank changes by at most one.

At every ancestor of v_0, the two rows lie in the same child. If that child is the right child, the gate is an internal row operation on the relevant cross block and preserves rank; if it is the left child, the cross block is unchanged. At a descendant containing t but not s, every column of the descendant interval lies strictly to the right of s. Since X remains lower triangular, row s vanishes on those columns, so the descendant block is unchanged. All other descendants are also unchanged. Hence only the term at v_0 can vary. □

Corollary 9.33 (Interval-tree lower bound)**.** *For every $A \in \mathrm{U}_n(\mathbb{F})$ and every interval tree T,*

$$\ell_{\downarrow}(A) \geq \Phi_T(A).$$

Proof. The identity has potential zero. Apply Lemma 9.32 along any lower-directed word from the identity to A. □

Definition 9.34 (Tree-tight target)**.** A lower-unitriangular target A is *tree tight* if some interval tree T satisfies

$$\Phi_T(A) = \ell_{\downarrow}(A).$$

Tree tightness is stronger than possessing a rank lower bound. It says that one ordered hierarchy accounts for the entire restricted word length. The family below is tree tight by construction.

9.7.2 The snapshot family

Fix integers

$$u \geq 1, \qquad 1 \leq v \leq u+1,$$

and choose distinct phases

$$0 \leq s_1 < s_2 < \cdots < s_v \leq u.$$

Order the coordinates as

$$x_1, \ldots, x_u,\ p,\ q,\ t_1, \ldots, t_v.$$

For $0 \leq s \leq u$, put

$$w_s = e_p + \sum_{i=1}^{s} e_{x_i}. \tag{9.12}$$

Define

$$A(u; s_1, \ldots, s_v) \in \mathrm{U}_{u+v+2}(\mathbb{F}) \tag{9.13}$$

by

$$R_{x_i} = e_{x_i}, \qquad R_p = w_u, \qquad R_q = e_q,$$

and

$$R_{t_j} = e_{t_j} + e_q + w_{s_j} \qquad (1 \leq j \leq v). \tag{9.14}$$

The output rows record selected intermediate versions of the carrier p.

Proof of Theorem 9.5. For the ambient construction, first apply

$$q \to p.$$

Sweep through the updates

$$x_1 \to p, \dots, x_u \to p.$$

Whenever phase s_j is reached, apply

$$p \to t_j.$$

Finally apply $q \to p$ with coefficient -1. At phase s, the carrier is $e_q + w_s$, so every output has the required value, and the final gate removes e_q from row p. The word has length

$$1 + u + v + 1 = u + v + 2.$$

Thus

$$\ell_*\big(A(u; s_1, \dots, s_v)\big) \le u + v + 2.$$

For a lower word, perform the same u updates without opening the catalyst. At each selected phase, use the two gates

$$p \to t_j, \qquad q \to t_j.$$

This gives length $u + 2v$.

It remains to prove optimality in the lower model. At the root, split

$$\{x_1, \dots, x_u, p\} \quad | \quad \{q, t_1, \dots, t_v\}.$$

The nonzero rows of the root cross block are $w_{s_1}, \dots, w_{s_v}$. Distinct prefix vectors are linearly independent: in a linear relation, inspect successively a coordinate occurring in w_{s_j} but in none of the earlier prefixes, starting from the largest phase. Hence the root contributes v.

Inside the left child, use the comb that splits off x_1, then x_2, and so on while retaining p in the remaining interval. Every one of the u cuts has rank one because the final row p contains every x_i. Inside the right child, repeatedly split off the last output while retaining q in the left interval. Each of the v cuts has rank one because every output contains q. Therefore

$$\Phi_T\big(A(u; s_1, \dots, s_v)\big) = v + u + v = u + 2v.$$

The interval-tree lower bound matches the displayed lower circuit. □

Remark 9.35 (Support connectivity). For $u, v \ge 1$, the undirected support graph of $A(u; s_1, \dots, s_v) - I$ is connected. Row p joins all update coordinates x_i, while every output joins both p and q. The separation below therefore does not arise from a direct sum of independent six-row gadgets.

9.7.3 Exact higher binary examples

The interval tree gives the exact restricted distance in all dimensions. For the first three higher targets, a separate bidirectional search also proves that the displayed ambient construction is optimal.

Theorem 9.36 (Exact binary cases in dimensions seven through nine)**.** *Over $\mathbb{F}_2$, let*

$$H_7 = A(2; 0, 1, 2), \qquad H_8 = A(3; 0, 1, 2), \qquad H_9 = A(3; 0, 1, 2, 3).$$

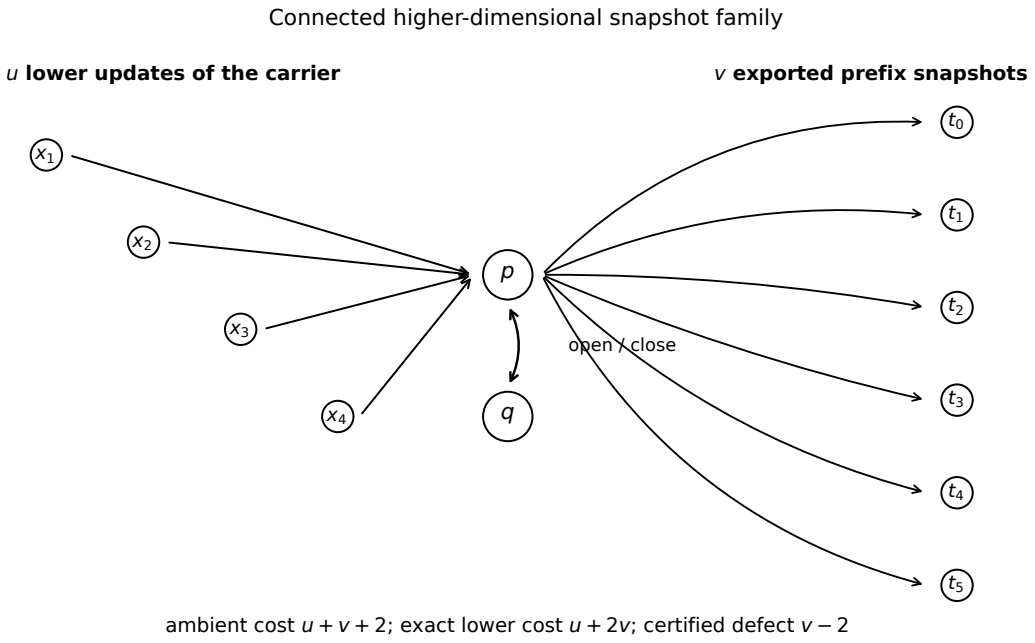


Figure 9.9. The connected snapshot family. One temporary upper dependency lets the ambient circuit export v selected prefix states. A lower-only circuit must insert the q component separately into every output.

Then

$$(\ell_{\downarrow}(H_7), \ell_*(H_7)) = (8, 7),$$
$$(\ell_{\downarrow}(H_8), \ell_*(H_8)) = (9, 8),$$

and

$$(\ell_{\downarrow}(H_9), \ell_*(H_9)) = (11, 9).$$

Proof. The restricted values follow from Theorem 9.5. The ambient words have lengths 7, 8, and 9. To exclude shorter words, the supplement computes exact ambient balls and checks that the two relevant balls do not intersect. The certificate sizes are

n	tested split	left ball	right ball
7	$3+3$	23276	23276
8	$3+4$	52277	1174097
9	$4+4$	2994949	2994949

Every intersection is empty. Since the generating sets are symmetric, an empty radius-r versus radius-s intersection rules out every word of length at most $r+s$. □

These are exact higher-dimensional examples, not merely padded copies of C.

Table 9.4. Successive connected cases from the optimized snapshot family. The restricted value is exact; the ambient column is a certified construction and is exact for $n = 7, 8, 9$ over $\mathbb{F}_2$.

n	u	v	$\ell_\downarrow$	ambient upper bound / certified gap
7	2	3	8	7 / 1
8	3	3	9	8 / 1
9	3	4	11	9 / 2
10	4	4	12	10 / 2
11	4	5	14	11 / 3
12	5	5	15	12 / 3
13	5	6	17	13 / 4
15	6	7	20	15 / 5
17	7	8	23	17 / 6
19	8	9	26	19 / 7
21	9	10	29	21 / 8

9.7.4 A connected defect in every dimension

Proof of Corollary 9.6. For $n \geq 7$, put

$$v_n = \left\lfloor \frac{n-1}{2} \right\rfloor, \qquad u_n = n - v_n - 2,$$

and take the phases

$$0, 1, \ldots, v_n - 1.$$

The inequality $v_n \leq u_n + 1$ holds, so the snapshot target is defined. By Theorem 9.5,

$$\ell_\downarrow(A_n) = u_n + 2v_n = n + v_n - 2,$$

while

$$\ell_*(A_n) \leq u_n + v_n + 2 = n.$$

Hence

$$\ell_\downarrow(A_n) - \ell_*(A_n) \geq v_n - 2 = \left\lfloor \frac{n-5}{2} \right\rfloor.$$

Connectivity follows from the preceding remark. □

The ratio also increases beyond the block-sum value $7/6$.

Corollary 9.37 (Asymptotic multiplicative distortion)**.** *For every field,*

$$\limsup_{n\to\infty} \sup_{A \in \mathrm{U}_n(\mathbb{F}),\, A \neq I} \frac{\ell_\downarrow(A)}{\ell_*(A)} \geq \frac{3}{2}.$$

Proof. Along the optimized connected family,

$$\frac{\ell_\downarrow(A_n)}{\ell_*(A_n)} \geq \frac{n + \lfloor (n-1)/2 \rfloor - 2}{n},$$

which tends to $3/2$. □

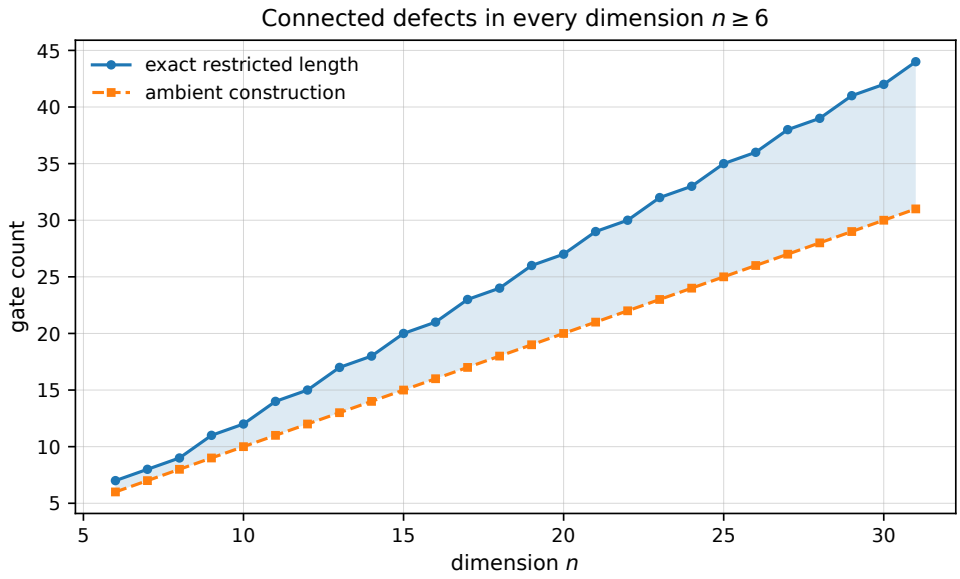


Figure 9.10. The six-row atom is the first exact defect. From dimension seven onward, the connected snapshot family gives an exact restricted length and a shorter ambient construction in every dimension.

9.8 Bounded and weighted catalytic resources

The connected snapshot family uses a single matched excursion: one upper-directed gate opens the dirty carrier and one upper-directed gate closes it. This permits a sharper statement than the unrestricted inequality $\ell_* < \ell_\downarrow$.

9.8.1 A bounded number of upper-directed gates

For a circuit word w, let $u(w)$ denote its number of upper-directed gates.

Definition 9.38 (Upper-gate budget)**.** For $q \geq 0$ and $A \in \mathrm{U}_n(\mathbb{F})$, define

$$\ell_{\leq q}(A) = \min\{|w| : [w] = A,\ u(w) \leq q\}.$$

Thus

$$\ell_{\leq 0}(A) = \ell_\downarrow(A), \qquad \ell_{\leq q+1}(A) \leq \ell_{\leq q}(A), \qquad \ell_*(A) \leq \ell_{\leq q}(A).$$

Proposition 9.39 (Two upper gates give a linear advantage)**.** *For every snapshot target* $A = A(u; s_1, \ldots, s_v)$*,*

$$\ell_{\leq 0}(A) = u + 2v, \qquad \ell_{\leq 2}(A) \leq u + v + 2.$$

Consequently,

$$\ell_{\leq 0}(A) - \ell_{\leq 2}(A) \geq v - 2.$$

For the optimized connected target A_n *of Corollary 9.6,*

$$\ell_{\leq 0}(A_n) - \ell_{\leq 2}(A_n) \geq \left\lfloor \frac{n-5}{2} \right\rfloor.$$

Proof. The exact restricted value is Theorem 9.5. Its ambient construction has precisely two upper-directed gates, namely the opening and closing operations on the carrier row; all u updates and all v reads are lower directed. The optimized choice has $v = \lfloor (n-1)/2 \rfloor$. □

Corollary 9.40 (No budget-only additive simulation bound)**.** *There is no finite constant K such that*

$$\ell_{\downarrow}(A) \le \ell_{\le 2}(A) + K$$

for all dimensions, fields, and lower-unitriangular targets A.

Proof. The left-minus-right difference in Proposition 9.39 tends to infinity with n. □

Thus the power of the excursion is not controlled by the number of forbidden gates. It is controlled by how many useful carrier states are exported between opening and closing.

9.8.2 Rank-one, bandwidth-one flag escape

The number of upper gates does not measure how far a circuit travels from the lower subgroup. We now control every intermediate state.

For $X \in \mathrm{SL}_n(\mathbb{F})$, let X^+ be the matrix obtained by retaining only the strict upper entries of X. Define

$$r^+(X) = \operatorname{rank} X^+$$

and

$$b^+(X) = \max\{j - i : X_{ij} \neq 0,\ i < j\},$$

with $b^+(X) = 0$ when $X^+ = 0$. Let $\ell^{(r,b)}(A)$ be the minimum length of a transvection circuit from I to A such that every prefix matrix X satisfies

$$r^+(X) \le r, \qquad b^+(X) \le b.$$

Lemma 9.41. *For $A \in \mathrm{U}_n(\mathbb{F})$,*

$$\ell^{(0,0)}(A) = \ell_{\downarrow}(A).$$

Proof. A lower-directed gate preserves lower triangularity. Conversely, if X is lower unitriangular and an upper-directed gate $R_t \leftarrow R_t + aR_s$ with $t < s$ is applied, then the new (t, s) entry equals

$$X_{t,s} + aX_{s,s} = a \neq 0.$$

The next state is therefore not lower triangular. A circuit whose every prefix has zero strict upper part uses only lower-directed gates. □

Theorem 9.42 (Microscopic escape, macroscopic saving)**.** *For every field $\mathbb{F}$ and every $n \ge 7$, there is a connected target $A_n \in \mathrm{U}_n(\mathbb{F})$ such that*

$$\ell^{(0,0)}(A_n) - \ell^{(1,1)}(A_n) \ge \left\lfloor \frac{n-5}{2} \right\rfloor.$$

The length-n circuit witnessing the second term uses exactly two upper-directed gates.

Proof. Use the connected family from Corollary 9.6, with the carrier coordinates $p < q$ adjacent. Between the opening and closing gates, the only strict upper entry is the coefficient at position (p, q).

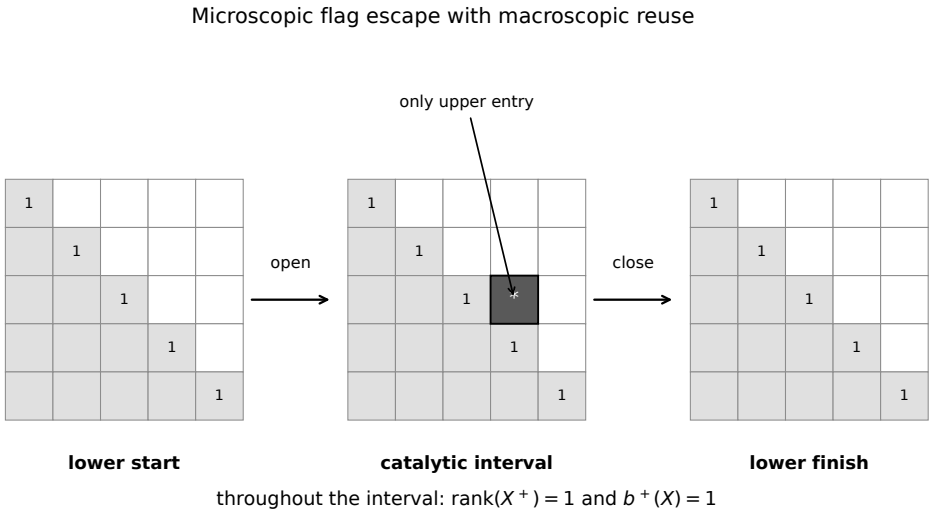


Figure 9.11. The snapshot circuit leaves the lower subgroup through one adjacent upper entry. Throughout the excursion the strict upper part has rank one and bandwidth one, yet the accumulated saving grows linearly.

Hence every prefix has upper rank at most one and upper bandwidth at most one. The displayed ambient circuit has length n, whereas

$$\ell^{(0,0)}(A_n) = \ell_{\downarrow}(A_n) = n + \left\lfloor \frac{n-1}{2} \right\rfloor - 2.$$

Subtracting proves the claim. □

9.8.3 A weighted upper-gate metric

A hardware or fault model may penalize an upper-directed interaction more heavily than a lower-directed one. Fix $\lambda > 0$. If w has $d(w)$ lower-directed gates and $u(w)$ upper-directed gates, put

$$\mathrm{cost}_\lambda(w) = d(w) + \lambda u(w),$$

and define

$$\ell_\lambda(A) = \inf\{\mathrm{cost}_\lambda(w) : [w] = A\}.$$

At $\lambda = 1$ this is the ordinary ambient length, while large λ increasingly penalizes flag excursions.

Proposition 9.43 (Fixed penalties do not suppress catalysis)**.** *For every snapshot target $A = A(u; s_1, \dots, s_v)$ and every $\lambda > 0$,*

$$\ell_\lambda(A) \le u + v + 2\lambda.$$

Consequently,

$$\ell_{\downarrow}(A) - \ell_\lambda(A) \ge v - 2\lambda.$$

For the optimized connected family,

$$\ell_{\downarrow}(A_n) - \ell_\lambda(A_n) \ge \left\lfloor \frac{n-1}{2} \right\rfloor - 2\lambda.$$

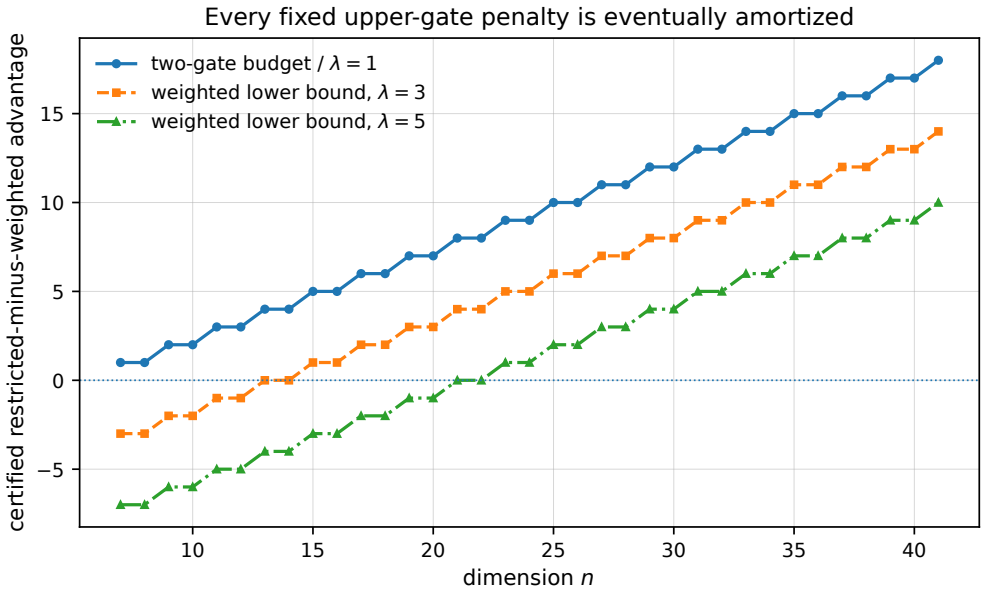


Figure 9.12. Certified lower-minus-weighted advantage for the optimized connected family. Every fixed upper-gate penalty is eventually amortized because the same matched excursion serves a growing number of output snapshots.

In particular, for every fixed finite λ, the weighted advantage grows linearly with n.

Proof. The displayed catalytic word contains $u + v$ lower-directed gates and two upper-directed gates, so its weighted cost is $u + v + 2\lambda$. Subtract this from the exact restricted value $u + 2v$. The optimized family again has $v = \lfloor (n-1)/2 \rfloor$. □

Remark 9.44 (Penalty threshold). For the displayed realization of a snapshot target, catalysis beats the optimal lower-only cost whenever

$$\lambda < \frac{v}{2}.$$

The critical penalty therefore scales with the number of consumers, not with the dimension of the dirty workspace.

Proof of Theorem 9.7. Combine Propositions 9.39 and 9.43. □

9.9 Algebraic rarity of catalytic defects

The preceding sections construct highly structured targets with large defects. This section proves the complementary statement that catalytic defects are nongeneric at every fixed dimension. Fix a field K. Algebraic dimensions and closures are taken after base change to an algebraic closure of K. Put

$$N = \dim \mathrm{U}_n = \binom{n}{2}, \qquad M = n(n-1).$$

9.9.1 Transvection product loci

For a direction skeleton

$$\sigma = ((t_1, s_1), \dots, (t_L, s_L)), \qquad t_i \neq s_i,$$

define the polynomial word map

$$\Phi_\sigma : \mathbb{A}^L \longrightarrow \mathrm{SL}_n, \qquad (a_1, \dots, a_L) \longmapsto (I + a_L E_{t_L, s_L}) \cdots (I + a_1 E_{t_1, s_1}).$$

The coefficients are allowed to vanish, so a word of length at most L can be padded to length L. Let

$$\mathcal{R}^*_{n,\le L} = \mathrm{U}_n \cap \bigcup_{\sigma \in \{(t,s) : t \neq s\}^L} \operatorname{im} \Phi_\sigma$$

be the ambient reachability locus, and define $\mathcal{R}^{\downarrow}_{n,\le L}$ analogously using only lower directions.

Equivalently, put

$$X_* = \bigcup_{t \neq s} \{I + aE_{t,s} : a \in K\}, \qquad X_{\downarrow} = \bigcup_{s < t} \{I + aE_{t,s} : a \in K\}.$$

Then $\ell_*(A)$ and $\ell_{\downarrow}(A)$ are the corresponding transvection product lengths: the least L for which $A \in X_*^L$ or $A \in X_{\downarrow}^L$.

Proposition 9.45 (Dimension of bounded-length loci)**.** *For every $L \ge 0$, the sets $\mathcal{R}^*_{n,\le L}$ and $\mathcal{R}^{\downarrow}_{n,\le L}$ are constructible and*

$$\dim \overline{\mathcal{R}^*_{n,\le L}} \le L, \qquad \dim \overline{\mathcal{R}^{\downarrow}_{n,\le L}} \le L.$$

Proof. There are finitely many direction skeletons of length L. For each skeleton, Φ_σ is a morphism from an L-dimensional affine space. Chevalley's theorem makes its image constructible, and the closure of the image has dimension at most L [30]. Finite unions preserve constructibility and the maximum dimension bound. Intersecting with the closed subvariety U_n cannot increase dimension. □

The argument is an instance of a general parameter-counting principle.

Proposition 9.46 (Dimension per gate)**.** *Let G be an algebraic group, let $V \subseteq G$ be an irreducible subvariety of dimension D, and let $X \subseteq G$ be a finite union of locally closed subvarieties, each of dimension at most r, with the identity in X. Write*

$$X^{(L)} = \{x_L \cdots x_1 : x_1, \dots, x_L \in X\}.$$

Then

$$\dim \overline{V \cap X^{(L)}} \le rL.$$

In particular, if $rL < D$, a Zariski-generic point of V has X-product length greater than L.

Proof. Every irreducible component of the Cartesian power $X^{\times L}$ has dimension at most rL. Multiplication $G^L \to G$ is a morphism, so the closure of the image of each component has dimension at most rL. There are finitely many components, and intersecting the resulting finite union with V cannot increase dimension. Since the identity belongs to X, products of length at most L are already contained in $X^{(L)}$. □

For transvections, the discrete choice of direction contributes finitely many components and each component carries one scalar parameter. Thus $r = 1$. Every element of $\mathrm{U}_n(K)$ has lower length at most

N: process the strict lower entries in any root order compatible with elimination. The dimension-per-gate bound shows that this worst-case value is generic.

Theorem 9.47 (Generic rigidity)**.** *Let K be an infinite field. There is a nonempty Zariski-open subvariety $\Omega_n \subseteq \mathrm{U}_n$, defined over K, such that $\Omega_n(K)$ is nonempty and*

$$\ell_*(A) = \ell_\downarrow(A) = N \qquad (A \in \Omega_n(K)).$$

The same statement holds with exact product length replaced by border product length.

Proof. After base change to an algebraic closure $\overline{K}$, Proposition 9.45 gives

$$\dim \overline{\mathcal{R}^*_{n,\le N-1}} \le N - 1 < N = \dim \mathrm{U}_n\,.$$

Because the word maps are defined over K, this proper closed set descends to a proper closed subset of U_n over K. Its complement is a nonempty Zariski-open set Ω_n. Since K is infinite and $\mathrm{U}_n \cong \mathbb{A}^N$, the set $\Omega_n(K)$ is nonempty and Zariski dense. For any $A \in \Omega_n(K)$, one has $\ell_*(A) \ge N$. The inequalities

$$\ell_*(A) \le \ell_\downarrow(A) \le N$$

force equality. Replacing each image by its Zariski closure gives the border-length statement by the same argument. □

9.9.2 Defect codimension

For an algebraically closed field K and $d \ge 1$, define

$$\mathcal{D}_{n,\ge d} = \{A \in \mathrm{U}_n(K) : \Delta(A) \ge d\}.$$

Theorem 9.48 (Defect–codimension bound)**.** *Let K be algebraically closed. For every $1 \le d \le N$,*

$$\dim \overline{\mathcal{D}_{n,\ge d}} \le N - d.$$

Equivalently,

$$\operatorname{codim}_{\mathrm{U}_n} \overline{\mathcal{D}_{n,\ge d}} \ge d.$$

More finely, on the stratum where $\ell_\downarrow(A) = L$ and $\Delta(A) \ge d$, the closure has dimension at most $L - d$.

Proof. If $\Delta(A) \ge d$, then

$$\ell_*(A) \le \ell_\downarrow(A) - d \le N - d.$$

Thus $\mathcal{D}_{n,\ge d} \subseteq \mathcal{R}^*_{n,\le N-d}$, and Proposition 9.45 gives the first assertion. On the intrinsic-length-L stratum one instead has $\ell_*(A) \le L - d$, proving the refinement. □

9.9.3 Finite-field sparsity

The dimension argument has an elementary counting counterpart. Allowing zero coefficients, a padded word of length L is specified by L choices among M directed positions and L coefficients in $\mathbb{F}_q$. Consequently

$$|\mathcal{R}^*_{n,\le L}(\mathbb{F}_q)| \le (Mq)^L.$$

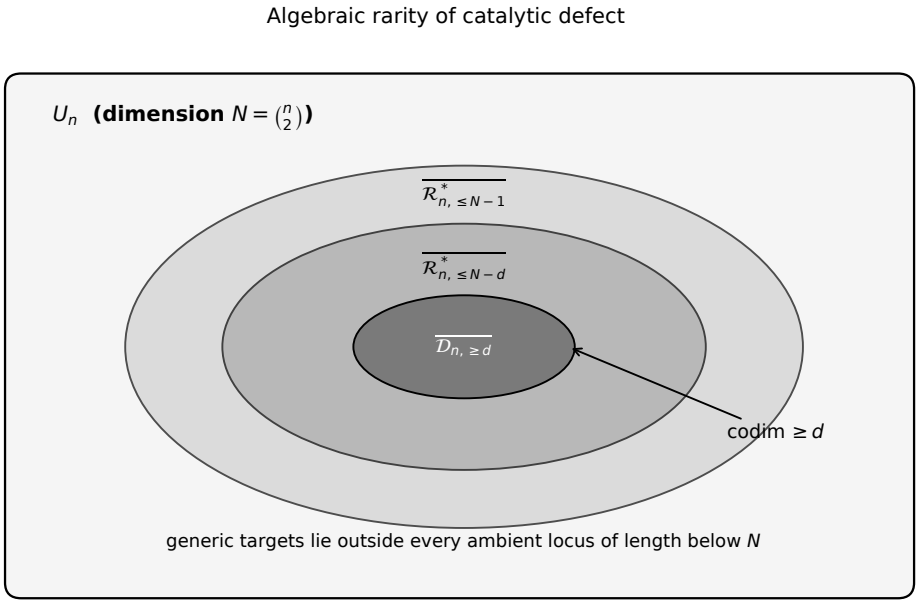


Figure 9.13. Bounded ambient product loci form nested algebraic strata inside U_n. A defect of size at least d lies inside the ambient reachability locus of length $N - d$, whose closure has codimension at least d.

Corollary 9.49 (Finite-field defect density)**.** *For* $1 \le d \le N$*,*

$$\mathbb{P}_{A \in \mathrm{U}_n(\mathbb{F}_q)}[\Delta(A) \ge d] \le \min\{1, M^{N-d} q^{-d}\}.$$

In particular, for fixed n*,*

$$\mathbb{P}[\Delta(A) > 0] = O_n(q^{-1}), \qquad \mathbb{E}[\Delta(A)] = O_n(q^{-1})$$

as $q \to \infty$*.*

Proof. The defect condition implies ambient length at most $N - d$, so the number of such targets is at most $(Mq)^{N-d}$. Divide by

$$|\,\mathrm{U}_n(\mathbb{F}_q)| = q^N.$$

The expectation bound follows from

$$\mathbb{E}[\Delta] = \sum_{d=1}^{N} \mathbb{P}[\Delta \ge d].$$

□

The same counting argument also gives

$$\mathbb{P}_{A \in \mathrm{U}_n(\mathbb{F}_q)}[\ell_*(A) = \ell_\downarrow(A) = N] \ge 1 - M^{N-1} q^{-1},$$

whenever the right side is positive. Thus maximal ambient and intrinsic length is not only Zariski generic over infinite fields but asymptotically typical over growing finite fields at fixed dimension.

9.9.4 Border catalysis

For an algebraic generating set X, define the border product length

$$\underline{\ell}_X(A) = \min\{L : A \in \overline{X^L}\}.$$

This is analogous to the passage from tensor rank to border rank [35]. The generic value is still N by Theorem 9.47, but exact and border defects can differ on exceptional strata. It is presently open whether the canonical atom C belongs to the closure of the six-factor lower transvection locus. A negative answer would show that its one-gate defect is stable under algebraic degeneration; a positive answer would exhibit a rank-versus-border-rank phenomenon inside reversible linear synthesis.

Remark 9.50 (Rare but amplifiable). The codimension theorem and the tensor construction point in opposite directions without contradiction. At each fixed n, a defect of size d lies on a codimension-at-least-d subset. Carefully selected points on these thin strata can nevertheless be tensorized into families with $\Omega(n \log n)$ additive defect. Catalytic compression is therefore algebraically rare but asymptotically amplifiable.

9.10 Recursive, block, and tensor amplification

9.10.1 Restricted block additivity

The lower metric is exactly additive on contiguous block-diagonal targets.

Proposition 9.51 (Block additivity). *Let $A_i \in \mathrm{U}_{n_i}(\mathbb{F})$ for $1 \le i \le k$. Then*

$$\ell_{\downarrow}\big(\mathrm{diag}(A_1, \dots, A_k)\big) = \sum_{i=1}^{k} \ell_{\downarrow}(A_i).$$

Proof. The upper bound follows by running optimal lower circuits independently inside the blocks.

For the lower bound, view an element of $\mathrm{U}_{n_1+\cdots+n_k}(\mathbb{F})$ as a lower block-triangular matrix and project to its diagonal blocks:

$$\pi(X) = (X_{11}, \dots, X_{kk}).$$

Because diagonal blocks multiply independently in a lower block-triangular product, π is a homomorphism. A lower transvection internal to one block projects to one generator in that factor, while a crossing transvection projects to the identity in every factor. If m_i gates survive in factor i, then the total word length is at least

$$\sum_i m_i \ge \sum_i \ell_{\downarrow}(A_i).$$

□

The ambient metric is subadditive:

$$\ell_*\big(\mathrm{diag}(A_1, \dots, A_k)\big) \le \sum_i \ell_*(A_i).$$

Cross-block ambient workspace can only improve this upper bound.

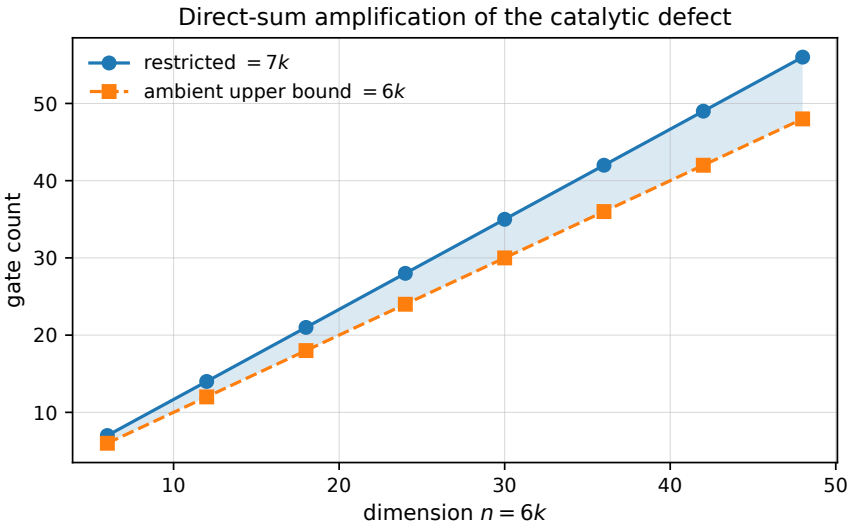


Figure 9.14. Block-sum amplification. The restricted length is exactly $7k$, while the displayed ambient construction has length $6k$.

Corollary 9.52 (Block-sum amplification). *For every field and every $n \geq 6$, there is an $A_n \in \mathrm{U}_n(\mathbb{F})$ with*

$$\ell_{\downarrow}(A_n) - \ell_*(A_n) \geq \left\lfloor \frac{n}{6} \right\rfloor .$$

For $n = 6k$, the target $C^{\oplus k}$ satisfies

$$\ell_{\downarrow}(C^{\oplus k}) = 7k, \qquad \ell_*(C^{\oplus k}) \leq 6k.$$

Proof. For $n = 6k + r$, take $C^{\oplus k} \oplus I_r$ and apply block additivity together with k copies of the ambient six-gate word. □

9.10.2 A recursive zeta-shaped lift

For $A \in \mathrm{U}_m(\mathbb{F})$, define

$$\mathcal{Z}(A) = \begin{pmatrix} A & 0 \\ A & A \end{pmatrix} \in \mathrm{U}_{2m}(\mathbb{F}). \tag{9.15}$$

Proposition 9.53 (Exact restricted lift). *For every $A \in \mathrm{U}_m(\mathbb{F})$,*

$$\ell_{\downarrow}(\mathcal{Z}(A)) = 2\ell_{\downarrow}(A) + m,$$

and

$$\ell_*(\mathcal{Z}(A)) \leq 2\ell_*(A) + m.$$

Consequently,

$$\Delta(\mathcal{Z}(A)) \geq 2\Delta(A).$$

Proof. The construction first realizes $\mathrm{diag}(A, A)$ and then applies the m root-cross gates that add top row i into bottom row i.

For the lower bound, write a circuit state in $m \times m$ blocks as

$$X = \begin{pmatrix} P & 0 \\ Q & R \end{pmatrix}.$$

Projection to the two diagonal blocks forces at least $2\ell_{\downarrow}(A)$ internal gates. A top-internal gate leaves Q unchanged. A bottom-internal gate performs an invertible row operation on Q and preserves its rank. A root-cross gate changes Q by one rank-one row contribution, so it can increase rank Q by at most one. The final cross block is A, of rank m, hence at least m root-cross gates are necessary. □

Let $A^{[0]} = A$ and $A^{[r+1]} = \mathcal{Z}(A^{[r]})$. If A has size m, restricted length a, and ambient length b, then

$$\ell_{\downarrow}(A^{[r]}) = 2^r a + rm2^{r-1}, \tag{9.16}$$

and

$$\ell_*(A^{[r]}) \le 2^r b + rm2^{r-1}. \tag{9.17}$$

Thus

$$\Delta(A^{[r]}) \ge 2^r \Delta(A).$$

For the canonical atom C, the first values are

r	dimension	restricted exact	ambient upper bound	certified gap
0	6	7	6	1
1	12	20	18	2
2	24	52	48	4
3	48	128	120	8
4	96	304	288	16

This gives arbitrarily deep exact recursion with identical diagonal children and a full-rank cross block while preserving a positive defect.

9.10.3 Tensor products of tree-tight targets

The tensor operation gives the stronger asymptotic amplification.

Theorem 9.54 (Exact restricted tensor rule)**.** *Let $A \in \mathrm{U}_m(\mathbb{F})$ and $B \in \mathrm{U}_n(\mathbb{F})$ be tree tight. Then $A \otimes B$ is tree tight and*

$$\ell_{\downarrow}(A \otimes B) = n\ell_{\downarrow}(A) + m\ell_{\downarrow}(B). \tag{9.18}$$

Moreover,

$$\ell_*(A \otimes B) \le n\ell_*(A) + m\ell_*(B), \tag{9.19}$$

so

$$\Delta(A \otimes B) \ge n\Delta(A) + m\Delta(B). \tag{9.20}$$

Proof. Use lexicographic order on the mn coordinates. Substitute a copy of a certificate tree for B at every leaf of a certificate tree for A. At an outer node, the cross block is

$$A[R, L] \otimes B,$$

whose rank is n rank $A[R, L]$ because B is invertible. The outer nodes therefore contribute $n\ell_{\downarrow}(A)$. The m diagonal copies of the inner tree contribute $m\ell_{\downarrow}(B)$. This proves the lower bound.

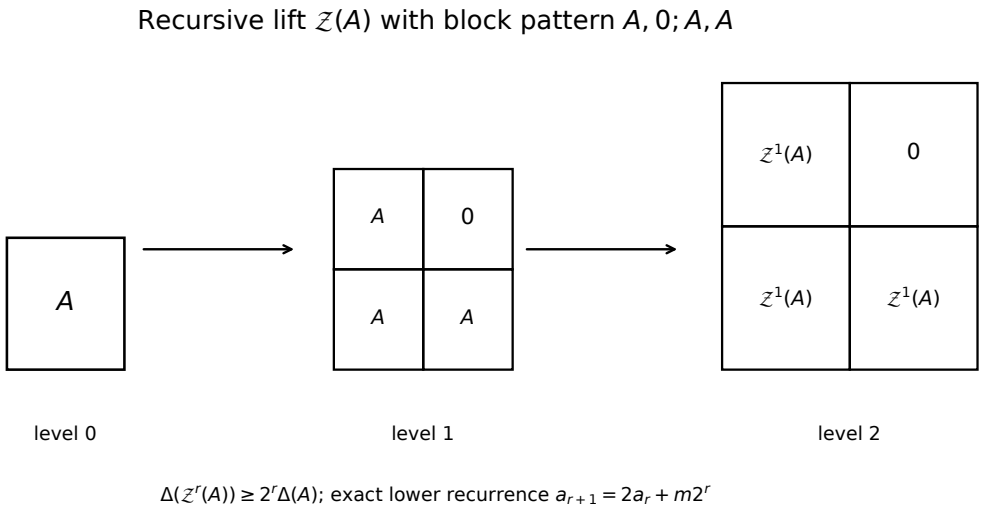


Figure 9.15. The recursive lift has the same outer shape as a Boolean zeta recursion. A finite defect at the leaf is doubled at every lift.

For the matching lower construction, first realize $I_m \otimes B$ using m copies of a circuit for B. Then realize $A \otimes I_n$. Each gate of a circuit for A lifts to n coordinatewise gates. Since

$$(A \otimes I_n)(I_m \otimes B) = A \otimes B,$$

the cost is the right side of Equation (9.18). The same construction with ambient circuits proves Equation (9.19). □

If A is tree tight of dimension m, with restricted length a and an ambient construction of length b, then

$$\ell_{\downarrow}(A^{\otimes r}) = rm^{r-1}a, \tag{9.21}$$

and

$$\ell_{*}(A^{\otimes r}) \leq rm^{r-1}b. \tag{9.22}$$

A positive one-copy defect therefore becomes an order-$N \log N$ defect at dimension $N = m^r$.

Proof of Theorem 9.10. Apply Theorem 9.54 repeatedly to the tree-tight snapshot target $A = A(u; s_1, \ldots, s_v)$, for which

$$a = u + 2v, \qquad b = u + v + 2 = m.$$

Then

$$\ell_{\downarrow}(A^{\otimes r}) - \ell_{*}(A^{\otimes r}) \geq rm^{r-1}(v-2).$$

Since $N = m^r$ and $r = \log_2 N / \log_2 m$, this is

$$\frac{v-2}{m \log_2 m} N \log_2 N.$$

□

Within the snapshot family, the best coefficient is obtained by the twenty-one-dimensional seed

$$A_{21} = A(9; 0, 1, \dots, 9),$$

for which

$$\ell_{\downarrow}(A_{21}) = 29, \qquad \ell_{*}(A_{21}) \le 21, \qquad \Delta(A_{21}) \ge 8.$$

Proposition 9.55 (Optimal snapshot seed for tensor rate)**.** *Among all snapshot targets, the coefficient*

$$\frac{v-2}{m \log_2 m}, \qquad m = u + v + 2,$$

is maximized at

$$(m, u, v) = (21, 9, 10).$$

Proof. For fixed m, maximize v subject to $v \le u + 1$ and $u + v + 2 = m$. Thus

$$v = \left\lfloor \frac{m-1}{2} \right\rfloor.$$

If m is even and at least eight, its coefficient is strictly smaller than that of $m-1$, because the numerator is the same while

$$(m-1)\log(m-1) < m \log m.$$

It therefore suffices to consider odd m. Apart from the constant change of logarithm base, the odd-dimensional envelope is

$$f(x) = \frac{x-5}{2x \log x}.$$

Its derivative has the sign of

$$5(1 + \log x) - x.$$

For $x > 5$ this expression crosses zero exactly once, between 19 and 21. Direct comparison gives $f(21) > f(19)$, after which the envelope decreases. Hence the unique integer maximizer is $m = 21$. □

The first tensor powers are

r	dimension 21^r	restricted exact	ambient upper bound	certified gap
1	21	29	21	8
2	441	1218	882	336
3	9261	38367	27783	10584
4	194481	1074276	777924	296352

9.10.4 Stable tensor capacity and activation

The tensor inequality also defines a regularized ambient invariant. Let $A \in \mathrm{U}_m(\mathbb{F})$ and put

$$b_r(A) = \frac{\ell_*(A^{\otimes r})}{m^{r-1}}.$$

Applying Equation (9.19) to $A^{\otimes r} \otimes A^{\otimes s}$ gives

$$b_{r+s}(A) \le b_r(A) + b_s(A).$$

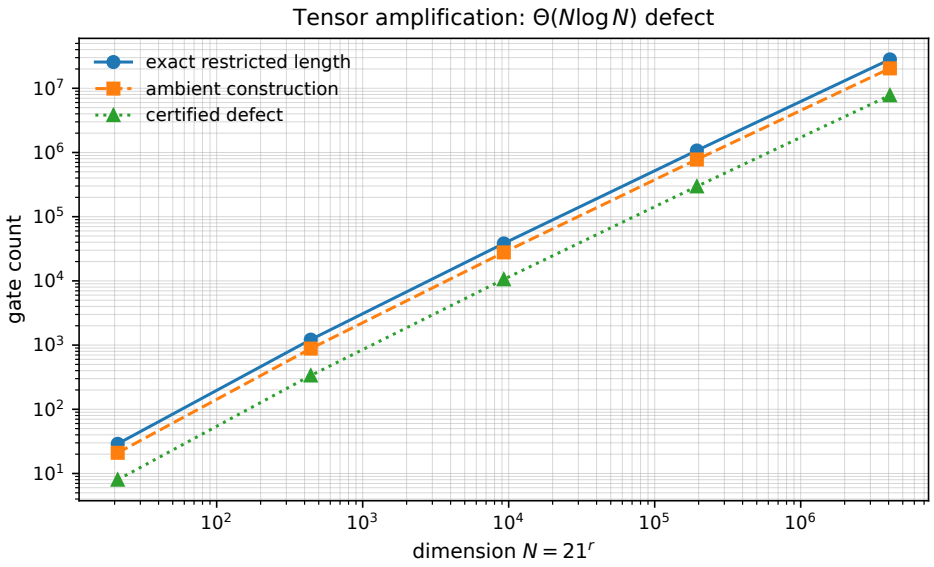


Figure 9.16. Tensor powers of the twenty-one-dimensional snapshot seed. Both displayed circuit sizes are of order $N \log N$, and their difference has a positive $N \log N$ coefficient.

Proposition 9.56 (Stable tensor length)**.** *The limit*

$$\tau_*(A) = \lim_{r\to\infty} \frac{\ell_*(A^{\otimes r})}{rm^{r-1}} = \inf_{r\geq 1} \frac{\ell_*(A^{\otimes r})}{rm^{r-1}}$$

exists. If A is tree tight, then the stable catalytic capacity

$$\kappa_\otimes(A) = \ell_\downarrow(A) - \tau_*(A)$$

satisfies

$$\kappa_\otimes(A) \geq \Delta(A).$$

Proof. The sequence $b_r(A)$ is subadditive, so Fekete's lemma gives the displayed limit. Since the one-copy lifted construction implies $\tau_*(A) \leq \ell_*(A)$, one has

$$\kappa_\otimes(A) \geq \ell_\downarrow(A) - \ell_*(A) = \Delta(A).$$

For a tree-tight target, Equation (9.21) identifies $\ell_\downarrow(A)$ with its exact normalized restricted tensor cost. □

For the twenty-one-dimensional seed A_{21},

$$\kappa_\otimes(A_{21}) \geq 8.$$

This regularization introduces a new possibility: a target may have $\Delta(A) = 0$ but $\kappa_\otimes(A) > 0$. In that case one copy is rigid while many tensor copies admit a collective shortcut. We call this *tensor activation*.

9.10.5 Extremal consequences

Define

$$D_n(\mathbb{F}) = \sup_{A\in \mathrm{U}_n(\mathbb{F})} \big(\ell_\downarrow(A) - \ell_*(A)\big).$$

For finite fields the supremum is a maximum. The tensor family gives the following explicit all-dimension bound.

Proof of Corollary 9.11. Let $m = 21$ and write

$$n = \sum_{j=0}^{r} d_j m^j, \qquad 0 \le d_j < m.$$

Take the block sum of d_j copies of $A_{21}^{\otimes j}$ for every j, with the convention $A_{21}^{\otimes 0} = I_1$. Block additivity and the tensor estimate give

$$D_n(\mathbb{F}) \ge \frac{8}{21} \sum_{j=0}^{r} j d_j 21^j.$$

Put $S = \sum_j j d_j 21^j$. Since $r = \lfloor \log_{21} n \rfloor$,

$$rn - S = \sum_{j<r} (r-j) d_j 21^j \le 20 \sum_{h \ge 1} h 21^{r-h} \le \frac{21^{r+1}}{20} \le \frac{21}{20} n.$$

Also $r \ge \log_{21} n - 1$. Therefore

$$S \ge n \log_{21} n - \frac{41}{20} n.$$

Multiplying by $8/21$ yields

$$D_n(\mathbb{F}) \ge \frac{8}{21 \log_2 21} n \log_2 n - \frac{82}{105} n.$$

□

Numerically,

$$\frac{8}{21 \log_2 21} = 0.0867315\ldots.$$

The elementary bottom-to-top construction gives

$$D_n(\mathbb{F}) \le \frac{n(n-1)}{2}.$$

Thus the current general range is

$$0.0867315\, n \log_2 n - O(n) \le D_n(\mathbb{F}) \le \frac{n(n-1)}{2}. \tag{9.23}$$

9.11 Discussion and open problems

9.11.1 What the threshold and higher families establish

The six-row atom proves that an optimal circuit for a lower-unitriangular target need not remain in the lower-unitriangular subgroup. The exact binary classification shows that this first failure is rigid: it occurs for exactly twenty targets, every defect is $(7, 6)$, and every target is generated by the same three-cross, one-phase mechanism.

The higher-dimensional results change the scale of the phenomenon. The connected snapshot family shows that exactly two upper-directed gates can support a defect linear in dimension, and that

the advantage persists under every fixed finite penalty on those gates. The recursive lift shows that a defective leaf survives arbitrarily many identical full-rank dyadic levels. Tensor products then turn any tree-tight positive-defect atom into an order-$N \log N$ additive separation. Thus the defect is not a finite anomaly and not merely linear in the worst case. At the same time, generic rigidity and the defect–codimension theorem show that large defects require correspondingly special algebraic conditions. The central synthesis is therefore not ubiquity but amplification: thin exceptional strata can control large structured families.

9.11.2 Relation to ordinary CNOT synthesis

Exact CNOT synthesis usually minimizes ℓ_* on all of $\mathrm{GL}_n(\mathbb{F}_2)$. The present problem fixes a structured target class and compares the ambient optimum with a structure-preserving optimum. Recent exact methods explore the full Cayley graph using symmetries, bidirectional search, A* methods, SAT, and related techniques [25, 44, 43]. The right-coset computation in this paper is tailored to a subgroup comparison, while the higher-dimensional exact audits use direct bidirectional balls for selected targets.

The 2026 NP-completeness result for all-to-all CNOT distance shows that cancellation and unrestricted parity reuse must be handled in full generality [18]. The present constructions isolate explicit reusable-carrier mechanisms under an ordered restriction. They do not resolve ambient synthesis complexity.

9.11.3 Open directions

Question 9.57 (Universal six-dimensional threshold). Is six the first defective dimension over every field, or even over every nonzero commutative ring? The atom gives a universal defective target in dimension six. Exact no-defect searches below six are proved here for $\mathbb{F}_2$ and $\mathbb{F}_3$.

Question 9.58 (Geometry of the defect locus). For fixed n and d, determine the irreducible components and exact dimension of $\overline{\mathcal{D}_{n,\geq d}}$. When is the lower bound codim $\geq d$ sharp?

Question 9.59 (Border catalysis). Does the canonical atom have lower border product length six or seven? More generally, when can exact catalytic defects disappear under algebraic degeneration?

Question 9.60 (Correct growth of the maximum defect). Is $D_n(\mathbb{F})$ of order $n \log n$, or can interacting catalysts produce a larger power of n? The current bounds are

$$\Omega(n \log n) \leq D_n(\mathbb{F}) \leq O(n^2).$$

Question 9.61 (Multiplicative distortion). Is

$$\sup_{n, A \in \mathrm{U}_n(\mathbb{F})} \frac{\ell_\downarrow(A)}{\ell_*(A)}$$

bounded? The connected snapshot family gives an asymptotic lower bound of $3/2$. Exceeding $3/2$ appears to require a mechanism beyond one clean carrier exporting independent snapshots.

Question 9.62 (Exact ambient complexity of snapshot targets). For which phase sets is the ambient construction of length $u + v + 2$ optimal? It is exact for the displayed binary targets in dimensions seven, eight, and nine, but the general ambient optimum remains open.

Question 9.63 (Tensor synergy and activation). Can $A \otimes B$ admit an ambient circuit shorter than the lifted bound

$$n\ell_*(A) + m\ell_*(B)?$$

Such synergy would only increase the defect, but understanding it is necessary for exact tensor complexity. Does there exist a tree-tight target with $\Delta(A) = 0$ but $\kappa_{\otimes}(A) > 0$?

Question 9.64 (Higher catalytic normal forms). Can interacting or crossing catalytic excursions produce asymptotic behavior impossible for a single clean carrier? A useful classification would separate laminar, crossing, and singular flag excursions.

Question 9.65 (Two-gate multiplicative ceiling). Among targets realizable with at most two upper-directed gates, is the asymptotic multiplicative distortion at most $3/2$? The snapshot family reaches this value from below, but the present paper proves no matching upper bound.

Question 9.66 (Matroidal classification). Which regular matroids admit a flagged realization whose shortest ambient state tour is strictly shorter than every flag-compatible lift? The first atom is the graphic matroid $M(K_4)$ with designated source, carrier, and output roles.

Question 9.67 (Weighted phase transition). For a fixed target family, determine the exact critical penalty λ_c at which a flag excursion ceases to improve the optimal weighted cost. The snapshot construction proves only the lower bound $\lambda_c \geq v/2$ for its displayed carrier word.

Question 9.68 (Other root systems). Let G be a split reductive group with unipotent radical U. Compare word length in positive root subgroups with ambient word length using all root subgroups. Which root systems admit a catalytic defect, and what is the first rank?

Question 9.69 (Certificate extraction). Can exact ambient lower bounds for large structured targets be accompanied by short human-readable certificates rather than exhaustive balls? The interval-tree method completely handles the restricted side, while the ambient side still lacks an equally general certificate.

9.12 The twenty binary dimension-six defects

A lower-unitriangular binary matrix is encoded by reading the strict lower entries row by row:

$$(1,0),(2,0),(2,1),\dots,(5,0),\dots,(5,4).$$

The resulting fifteen-bit integer is denoted by $\mathrm{code}(A)$. A row is displayed as the integer whose binary expansion is its support vector. Table 9.5 lists all defects. Every row has $\ell_{\downarrow} = 7$ and $\ell_{*} = 6$.

Table 9.5. The twenty defective matrices in $\mathrm{U}_6(\mathbb{F}_2)$.

code	R_0	R_1	R_2	R_3	R_4	R_5
6585	1	3	4	15	22	38
6641	1	3	4	14	23	38
6649	1	3	4	15	23	38
7601	1	3	4	14	22	39
7609	1	3	4	15	22	39
7665	1	3	4	14	23	39
10014	1	2	7	11	28	41
11038	1	2	7	11	28	42
12894	1	2	7	11	25	44
12958	1	2	7	11	26	44
13142	1	2	7	10	29	44

code	R_0	R_1	R_2	R_3	R_4	R_5
13198	1	2	7	9	30	44
14102	1	2	7	10	28	45
15118	1	2	7	9	28	46
17912	1	2	4	15	23	49
18936	1	2	4	15	23	50
19960	1	2	4	15	23	51
20984	1	2	4	15	23	52
22008	1	2	4	15	23	53
23032	1	2	4	15	23	54

For example, the canonical target has row encoding

$$(1, 3, 4, 14, 23, 39)$$

and code 7601.

9.13 Search algorithms and correctness

9.13.1 Restricted search

For $\mathbb{F}_2$, each lower-unitriangular matrix is stored as a bit string. Standard breadth-first search gives exact distance because every edge has unit weight. The algorithm terminates after visiting all states.

Algorithm 9.70 (Restricted breadth-first search)**.** Initialize $d[I] = 0$ and a queue containing I. Repeatedly remove X from the queue. For every lower transvection $s \to t$, form $Y = \tau_{t,s}(1)X$. If Y has not been visited, set $d[Y] = d[X] + 1$ and append Y. Continue until the queue is empty.

9.13.2 Right-coset key

For an invertible binary matrix A, let c_j be column j. Starting at $j = n - 1$ and moving left, reduce c_j by the already canonicalized columns $c_{j+1}, \dots, c_{n-1}$, using their pivot coordinates. The resulting ordered reduced columns form the key $\kappa(A)$.

Lemma 9.71. *For $A, B \in \mathrm{GL}_n(\mathbb{F}_2)$,*

$$\kappa(A) = \kappa(B) \quad \Longleftrightarrow \quad AU_n = BU_n.$$

Proof. Right multiplication by $U \in \mathrm{U}_n$ replaces column j by itself plus a combination of columns with index greater than j. Right-to-left reduction is therefore unchanged on a right coset. Conversely, equal reduced columns imply that the original column frames differ by a unique lower-unitriangular column transformation, so $A = BU$ for some $U \in \mathrm{U}_n$. □

9.13.3 Ambient search

Algorithm 9.72 (Radius-four right-coset meet in the middle)**.** Compute the exact radius-four ball B_4 in $\mathrm{GL}_6(\mathbb{F}_2)$. Bucket its elements by κ. For each bucket and every ordered pair (b, a) in that bucket, compute $u = b^{-1}a \in \mathrm{U}_6$ and update

$$d_*(u) \leftarrow \min\bigl(d_*(u), d(b) + d(a)\bigr).$$

Correctness is Lemma 9.28. The computation also checks that every generated pair lies in U_6 and that every one of the 32768 targets receives a finite distance.

9.13.4 Higher-dimensional bidirectional audits

For the targets H_7, H_8, H_9 of Theorem 9.36, the program `small_snapshot_exact.cpp` constructs exact ambient balls from the identity and from the target. If radius-r and radius-s balls are disjoint, symmetry of the generating set rules out every word of length at most $r + s$. The tested splits are $3 + 3$, $3 + 4$, and $4 + 4$, respectively. The source stores matrices in one 128-bit word and uses exact row-addition transitions; no probabilistic or floating-point step occurs.

9.13.5 Symbolic higher-family checks

The program `higher_family_tests.py` generates the snapshot targets over $\mathbb{F}_2$, $\mathbb{F}_3$, and $\mathbb{F}_5$, evaluates both displayed circuits, computes the interval-tree potentials, audits a tensor certificate for the seven-dimensional seed, verifies the recursive cross rank, and reproduces the twenty-one-dimensional rate optimization.

9.13.6 Further generalization audits

The program `further_generalization_tests.py` verifies the universal circuits over $\mathbb{Z}/4\mathbb{Z}$, $\mathbb{Z}/6\mathbb{Z}$, and $\mathbb{Z}/8\mathbb{Z}$, audits the rank-one bandwidth-one excursion, checks the finite-field counting bounds symbolically, and verifies the stable tensor subadditivity identities.

9.13.7 Motif comparison

The motif program independently loops over the three adjacent catalyst positions, the phase orientation, the phase leaf, and every nonempty proper before/after subset. It evaluates each six-gate word from the identity and stores its target. The resulting set has size twenty. Exact comparison with the twenty census defects returns equality.

9.14 Reproducibility ledger

The ancillary directory `supplement/` contains portable source, frozen expected outputs, and a one-command driver:

```
cd supplement
./run_all.sh
```

The driver compiles every C++ search, executes the Python motif, higher-family, matroid, and weighted-resource checks, and compares all generated outputs byte for byte with the frozen records.

The supplement README gives the individual commands and the role of every program. A SHA-256 manifest covers every included source and expected-output file.

Acknowledgement of computational status

All theorem statements are labeled by proof type in the body. The universal ring-valued atom, Schreier-holonomy interpretation, regular-matroid identification, interval-tree family, bounded, weighted, and microscopic resource theorems, algebraic-rarity bounds, block additivity, recursive lift, tensor rule, stable tensor limit, and asymptotic amplification are proved symbolically. The first-threshold classification, finite censuses, and exact ambient values in dimensions seven through nine are computer assisted, with complete search-space coverage justified in Section 9.13.

Data and code availability

The source archive accompanying this paper contains the complete LaTeX source, all vector figures, every C++ and Python verification program, frozen expected outputs, and SHA-256 manifests. The command

```
cd supplement
./run_all.sh
```

rebuilds the finite searches and symbolic audits and compares every output byte for byte with its frozen record. No randomized or floating-point decision enters a theorem certificate.

CHAPTER 10

Quantum and reversible gate families

10.1 Qudit SUM gates and ring-valued linear logic

10.1.1 From transvections to quantum gates

Let $d \geq 2$. On n qudits with computational basis labeled by $\mathbb{Z}_d^n$, define the scaled SUM gate

$$\mathrm{SUM}_{s\to t}^{(a)}|x_0,\ldots,x_{n-1}\rangle = |x_0,\ldots,x_t + ax_s,\ldots,x_{n-1}\rangle, \qquad a \in \mathbb{Z}_d.$$

It is a permutation unitary. Its action on basis labels is the row transvection

$$I + aE_{t,s}.$$

For $d = 2$ and $a = 1$, it is CNOT. For prime d, these gates lie in the standard qudit Clifford group. For composite d, they remain exact reversible modular-addition gates.

> **Proved statement**
>
> The universal six-row atom in the core chapter is valid over every nonzero commutative ring with identity. Taking $R = \mathbb{Z}_d$ gives a six-qudit target with restricted directed-SUM length seven and ambient all-direction SUM length six for every $d \geq 2$.

10.1.2 The universal six-qudit circuit

Use the coordinate names

$$x, p, q, t_0, t_1, t_2.$$

The target acts on basis labels by

$$\begin{aligned} p &\mapsto p + x, \\ t_0 &\mapsto t_0 + p + q, \\ t_1 &\mapsto t_1 + x + p + q, \\ t_2 &\mapsto t_2 + x + p + q, \end{aligned}$$

with x and q unchanged. An ambient word is

$$\mathrm{SUM}_{q\to p},\quad \mathrm{SUM}_{p\to t_0},\quad \mathrm{SUM}_{x\to p},\quad \mathrm{SUM}_{p\to t_1},\quad \mathrm{SUM}_{p\to t_2},\quad \mathrm{SUM}_{q\to p}^{(-1)}.$$

The first gate adds the dirty q component to p. The middle gates export two different carrier states. The final inverse gate restores p.

A lower-only word needs seven gates. The exact proof is coefficient-domain independent because it uses pivot support and the number of distinct carrier states rather than division.

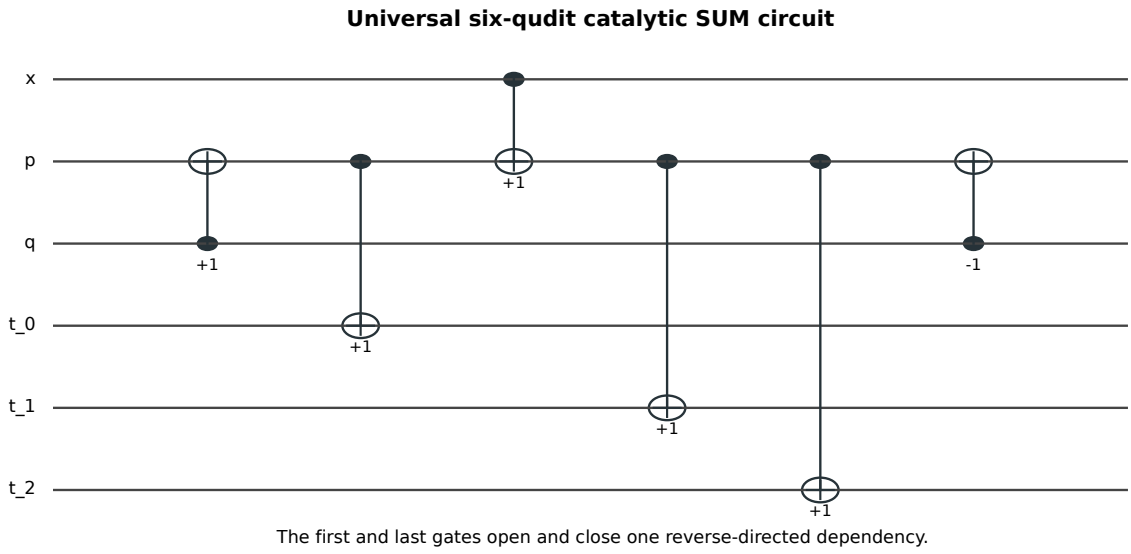


Figure 10.1. The universal six-qudit catalytic circuit. Coefficients are interpreted modulo d.

10.1.3 Connected higher-dimensional qudit families

The snapshot target $A(u; s_1, \dots, s_v)$ from the core chapter is defined over any coefficient ring for which the displayed additions make sense. Over $\mathbb{Z}_d$, it becomes a modular SUM circuit. Its explicit ambient word uses exactly two reverse-directed SUM gates and has length

$$u + v + 2.$$

The interval-tree proof of exact restricted length is most transparent over fields, but the displayed upper-bound and resource-amortization mechanism remain valid over every $\mathbb{Z}_d$.

For a field $\mathbb{F}_q$, the exact formula is

$$\ell_{\downarrow}(A) = u + 2v, \qquad \ell_{*}(A) \le u + v + 2,$$

so

$$\Delta(A) \ge v - 2.$$

The connected family has asymptotic ratio approaching $3/2$.

10.1.4 Weighted physical costs

Suppose forward SUM gates cost one and reverse-directed SUM gates cost λ. The displayed word has weighted cost

$$u + v + 2\lambda,$$

while the restricted construction costs $u + 2v$. Hence the gain is at least

$$v - 2\lambda.$$

This formula can be combined with hardware calibration. If reverse orientation requires echo pulses or routing, the catalyst remains useful when the number of exported snapshots exceeds twice the effective penalty.

Scope boundary

A gate-count saving does not imply a lower physical error or depth. Device-specific conclusions require a cost vector containing duration, infidelity, crosstalk, and routing. The algebraic theorem identifies candidate rewrites and the exact break-even inequality.

10.1.5 Pauli action for prime qudits

For odd prime d, let X and Z denote generalized Pauli operators. The SUM gate acts by

$$X_s \mapsto X_s X_t^a, \qquad X_t \mapsto X_t,$$
$$Z_s \mapsto Z_s, \qquad Z_t \mapsto Z_s^{-a} Z_t.$$

Thus the modular linear action embeds into a symplectic transformation. The catalytic row excursion has a dual Pauli interpretation: a temporary reverse dependency changes both X propagation and the contragredient Z propagation, then restores them.

This makes the SUM family a bridge between the transvection paper and the Clifford chapter. It is the only quantum extension in this monograph that follows directly from the ring-universal theorem without a new finite search.

10.1.6 Continuous-variable shear analogue

On continuous variables, a linear coordinate map A embeds symplectically as

$$A \longmapsto \begin{pmatrix} A & 0 \\ 0 & A^{-T} \end{pmatrix}.$$

A row shear therefore gives a Gaussian SUM transformation. The six-row word identity remains a matrix identity in the corresponding shear subgroup.

Research template

To claim an advantage in a full Gaussian gate library, one must compare against rotations, squeezers, beam splitters, and general symplectic decompositions. The current theorem proves a defect only inside the directed-shear versus all-shear comparison.

10.1.7 Compiler primitive

A compiler can search for a carrier row p, a temporary source q, and output rows requiring several successive states of p. The symbolic pattern is:

1. add q into p in the disallowed direction,
2. update p through a sequence of allowed additions,
3. copy selected intermediate states of p into outputs,
4. subtract q from p,
5. verify that the outputs contain the intended modular combinations.

The Python function `snapshot_target` constructs the target and `snapshot_ambient_word` emits the word.

Listing 10.1. Constructing a modular snapshot family.

```
from catalytic_gate_atlas.linear import (
   snapshot_target,
   snapshot_ambient_word,
   apply_row_word,
)

u = 4
phases = [0, 2, 4]
modulus = 5
A = snapshot_target(u, phases, modulus)
word = snapshot_ambient_word(u, phases, modulus)
assert apply_row_word(len(A), word, modulus) == A
```

10.1.8 Open qudit questions

Question 10.1. Does the first defective dimension remain six for every finite field and every modular ring $\mathbb{Z}_d$ under the directed-SUM metric?

Question 10.2. Can qudit Clifford gates outside the SUM subgroup produce a larger defect for a SUM target under total, entangling, or depth cost?

Question 10.3. Do composite dimensions admit characteristic-dependent catalytic motifs unavailable over fields?

10.2 Catalytic basis escape in symplectic Clifford synthesis

10.2.1 Binary symplectic representation

Modulo global phase and Pauli translations, an n-qubit Clifford operation acts on Pauli exponent vectors

$$v = (x_0, \ldots, x_{n-1} \mid z_0, \ldots, z_{n-1})^T \in \mathbb{F}_2^{2n}$$

through a symplectic matrix

$$M \in \mathrm{Sp}(2n, 2).$$

The elementary gates used here act as follows:

$$\begin{aligned} H_i &: (x_i, z_i) \mapsto (z_i, x_i), \\ S_i &: z_i \mapsto z_i + x_i, \\ \mathrm{CX}_{i\to j} &: x_j \mapsto x_j + x_i, \quad z_i \mapsto z_i + z_j, \\ \mathrm{CZ}_{ij} &: z_i \mapsto z_i + x_j, \quad z_j \mapsto z_j + x_i. \end{aligned}$$

Let $P_n \leq \mathrm{Sp}(2n, 2)$ be the parabolic subgroup stabilizing the Z Lagrangian

$$L_Z = \{(0 \mid z) : z \in \mathbb{F}_2^n\}.$$

The gates S, CZ, and CX preserve L_Z. Hadamard gates do not.

Definition 10.4 (Parabolic and ambient Clifford metrics)**.** The restricted alphabet is

$$\mathcal{S}_P = \{S_i, \mathrm{CZ}_{ij}, \mathrm{CX}_{i\to j}\},$$

and the ambient alphabet is

$$\mathcal{S}_C = \mathcal{S}_P \cup \{H_i\}.$$

Targets lie in P_n. We study both total primitive count and entangling count, where H_i and S_i have cost zero while CX and CZ have cost one.

This is a precise basis-escape model. It is not the unique meaning of "CSS preserving" or "Clifford cost." It is chosen because both groups are finite and the restricted subgroup has a clean Lagrangian interpretation.

10.2.2 Exact group sizes

The finite state spaces are

$$|P_n| = 2^{n(n+1)/2} |\operatorname{GL}(n,2)|,$$

and

$$|\operatorname{Sp}(2n,2)| = 2^{n^2} \prod_{j=1}^{n} (2^{2j}-1).$$

For $n = 1, 2, 3$ this gives:

n	$\lvert P_n \rvert$	$\lvert \operatorname{Sp}(2n,2) \rvert$
1	2	6
2	48	720
3	10,752	1,451,520

These sizes permit exhaustive breadth-first or zero-one search through three qubits.

10.2.3 Total primitive count

Exact finite computation

With every gate in $\mathcal{S}_P$ and $\mathcal{S}_C$ assigned unit cost and with SWAP excluded as a primitive, exhaustive search gives:

n	restricted diameter	ambient diameter	defective targets	maximum defect
1	1	3	0	0
2	5	8	0	0
3	8	12	24	1

At $n = 3$, six targets have distance pair $(7, 6)$ and eighteen have pair $(8, 7)$.

Thus three qubits are the first total-count threshold in this exact swap-free model.

A three-qubit witness

One defective target has symplectic matrix

$$M_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

A restricted word of length seven is

$$\mathrm{CX}_{0\to2}, \mathrm{CZ}_{12}, \mathrm{CX}_{2\to0}, \mathrm{CX}_{0\to2}, \mathrm{CX}_{0\to1}, \mathrm{CX}_{1\to0}, \mathrm{CX}_{0\to1}\,.$$

An ambient word of length six is

$$H_1, \mathrm{CX}_{1\to2}, \mathrm{CX}_{2\to1}, \mathrm{CX}_{2\to0}, \mathrm{CX}_{0\to2}, H_0.$$

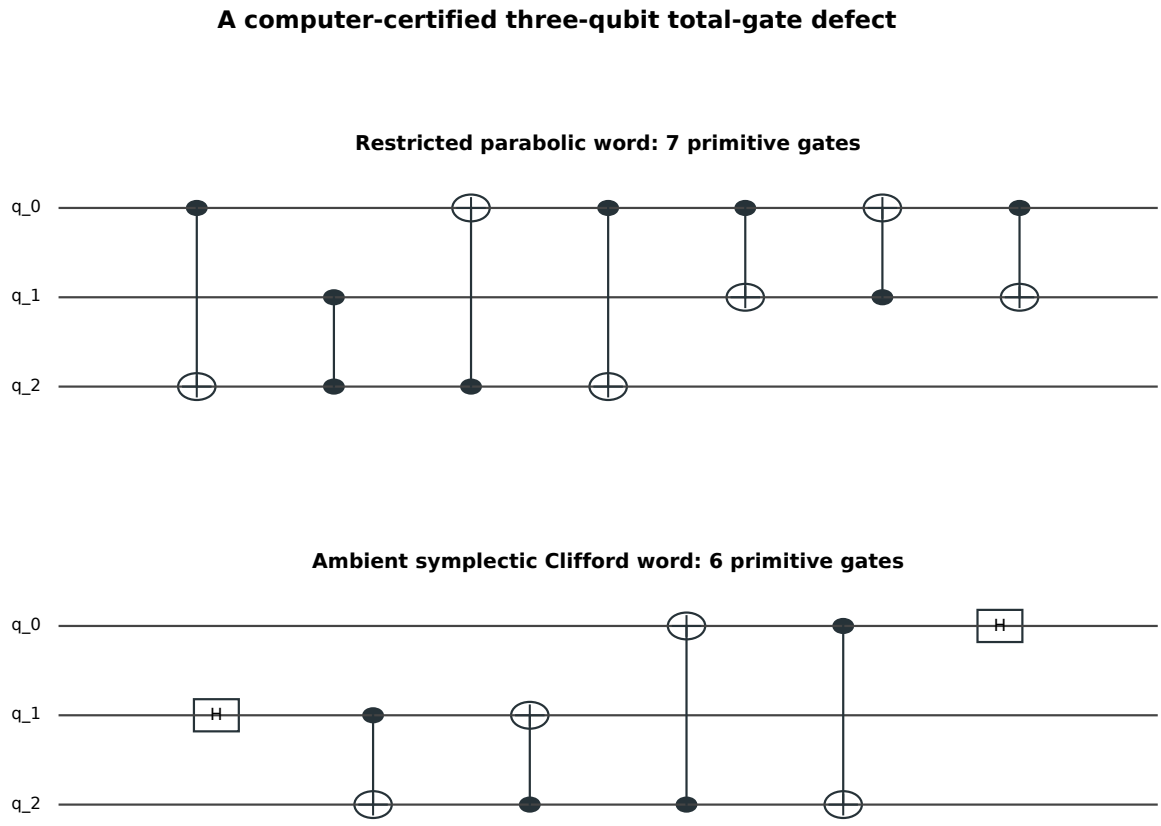


Figure 10.2. A swap-free three-qubit parabolic target with total primitive-count defect one.

The Hadamards do not form a matched pair on the same wire. They open and close the Lagrangian excursion in different coordinates. This is a useful warning against assuming that every catalyst has the literal form $g \cdots g^{-1}$.

10.2.4 SWAP sensitivity

Add SWAP_{ij} to both alphabets as a unit-cost primitive. The target groups remain unchanged, but the metric changes.

Exact finite computation

For $n = 3$ with unit-cost SWAP gates, the restricted diameter drops from eight to seven, the ambient diameter drops from twelve to eleven, and every one of the $10,752$ parabolic targets has zero total-count defect.

The preceding twenty-four defects therefore measure a real gate-alphabet distinction, not an invariant of the subgroup inclusion alone. In many architectures SWAP is not elementary and costs three CNOTs, while in others qubit movement or relabeling changes the accounting. The theorem must retain the convention.

10.2.5 Entangling-count defects

Now assign cost zero to H_i and S_i and cost one to CX and CZ.

Exact finite computation

The exact censuses are:

n	defective targets	distance pairs	maximum defect
1	0	none	0
2	4	$(3,2)$ with multiplicity 4	1
3	1,624	$(3,2):24,\ (4,3):432,\ (5,4):1056,$ $(6,4):48,\ (6,5):64$	2

The entangling metric has an earlier threshold and a larger exceptional set.

A two-qubit witness

A canonical target is

$$M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

A restricted realization is

$$\mathrm{CX}_{0\to1}, S_1, \mathrm{CX}_{1\to0}, \mathrm{CX}_{0\to1},$$

with three entangling gates. An ambient realization is

$$H_1, \mathrm{CX}_{1\to0}, \mathrm{CX}_{0\to1}, H_0, S_1, S_0,$$

with two entangling gates.

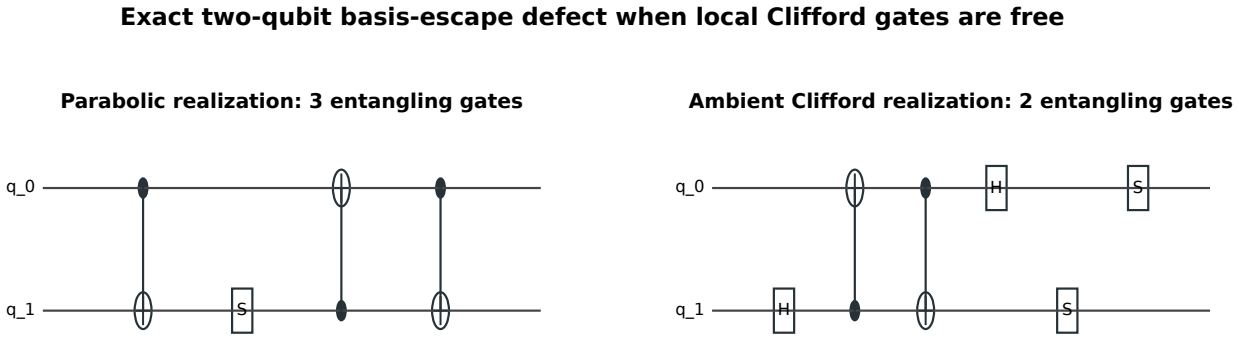


Figure 10.3. A two-qubit entangling-count defect. Local Clifford gates are free in this cost model.

10.2.6 Census summary

An analytic certificate for the four two-qubit defects

The four exceptional matrices admit one common lower-bound proof. Work in the declared symplectic quotient and entangling metric, with H_i and S_i free and SWAP excluded as a primitive. Put

$$J_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad D = \mathrm{diag}(d_0, d_1), \qquad d_0, d_1 \in \mathbb{F}_2,$$

and define

$$M_D = \begin{pmatrix} J_2 & 0 \\ I_2 + DJ_2 & J_2 \end{pmatrix}.$$

Theorem 10.5 (Two-qubit orbit and exact costs)**.** *Each of the four distinct matrices M_D lies in P_2 and has restricted entangling cost three and ambient entangling cost two. They form one orbit under left multiplication by the free restricted subgroup generated by S_0, S_1.*

Proof. The ambient sequence

$$H_1,\ \mathrm{CX}_{1\to 0},\ \mathrm{CX}_{0\to 1},\ H_0$$

has matrix $M_0 = \left(\begin{smallmatrix} J_2 & 0 \\ I_2 & J_2 \end{smallmatrix}\right)$. Applying $S_0^{d_0} S_1^{d_1}$ afterwards left multiplies by $\left(\begin{smallmatrix} I_2 & 0 \\ D & I_2 \end{smallmatrix}\right)$ and gives M_D. These matrices are distinct because J_2 is invertible. This also proves their orbit statement and the ambient upper bound two. The displayed restricted sequence for M_2 above gives M_{I_2} using three CNOT gates and one free S_1. Free final phase gates then give every M_D, proving the restricted upper bound three.

For the restricted lower bound, the upper-left block of a parabolic product is the product of its upper-left blocks. Each S or CZ contributes I_2, while the two directed CNOTs contribute the elementary matrices

$$L = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Products using at most two CNOTs belong to $\{I_2, L, U, LU, UL\}$, which does not contain J_2. Thus any restricted circuit for M_D requires at least three charged gates.

For the ambient lower bound, reorder the coordinates as $(x_0, z_0 \mid x_1, z_1)$. A local Clifford acts by an invertible block-diagonal matrix in this order. In M_D, each of the two diagonal 2×2 blocks is $\left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$ and has rank one. In either directed CNOT or CZ, both diagonal blocks are I_2 and have rank two. Left and

right multiplication by local Cliffords preserve these diagonal-block ranks. A circuit with one charged gate has the form $K_1 E K_2$ with K_1, K_2 local and E a CNOT or CZ, so it cannot equal M_D. A circuit with no charged gate is local and also has rank-two diagonal blocks. The ambient cost is therefore at least two. □

The complete finite census finds exactly four defective targets. Since the theorem supplies four distinct defective targets, it identifies the entire exceptional set in that census. The exact costs and orbit description above have analytic proofs, while completeness of the finite classification retains the exhaustive-search evidence.

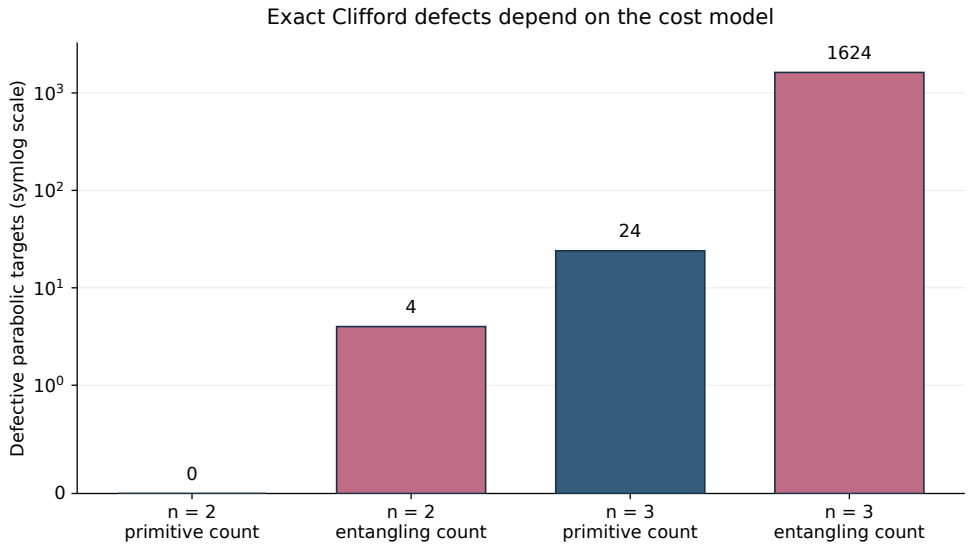


Figure 10.4. Exact parabolic-versus-ambient Clifford census. The vertical scale is symlogarithmic. At three qubits, adding SWAP as a unit-cost primitive removes all total-count defects.

The results establish three distinct lessons.

1. Basis escape produces genuine Clifford defects, not only linear CNOT defects.
2. The first threshold depends on whether local gates are free.
3. Adding a SWAP primitive can eliminate a finite total-count threshold.

10.2.7 Search completeness

The search state is the exact $2n \times 2n$ binary symplectic matrix. Every generator acts by a fixed row update. Ordinary breadth-first search gives exact unit-cost distances. Zero-one breadth-first search gives exact entangling costs. Since every state of P_n and $\mathrm{Sp}(2n, 2)$ is reached, comparing the two distance maps is a complete census.

The optimized C++ verifier represents a 6×6 matrix by a 36-bit integer. It enumerates the full three-qubit group in a few seconds in the present environment. The Python implementation is slower but independently expresses the same state model.

10.2.8 Interpretation as symplectic holonomy

The parabolic subgroup is a fiber over the Lagrangian Grassmannian. A Hadamard moves one coordinate out of the chosen Lagrangian. The ambient path traverses other Lagrangian frames and returns to a target stabilizing L_Z.

This is the symplectic analogue of the unitriangular Schreier loop. The catalyst is a temporary exchange of X and Z roles. The consumers are entangling gates that become cheaper in the changed frame.

10.2.9 CSS-preserving extensions

The parabolic model fixes one Lagrangian, but CSS circuit classes can be defined more narrowly or more broadly. Possible restricted alphabets include:

1. CNOT and Pauli gates only,
2. CNOT, S, and diagonal controlled phases,
3. block-diagonal symplectic transformations preserving separate X and Z codes,
4. logical Clifford gates preserving a CSS stabilizer decomposition,
5. architecture-constrained versions of the preceding models.

Each produces a different subgroup and metric.

> **Research template**
>
> A full CSS threshold paper should classify cost-preserving orbits of minimal defects, identify a symplectic-flag lower-bound invariant, and determine which defects survive when SWAP, qubit permutation, or local Cliffords are assigned alternative costs.

10.2.10 Depth and parallelism

Gate-count defects do not automatically imply depth defects. Given a gate sequence, define its dependency graph and schedule commuting or disjoint gates in parallel. The appropriate quantities are

$$\Delta_{\text{depth}}(U), \qquad \Delta_{\text{2q}-\text{depth}}(U), \qquad \Delta_{\text{space}-\text{time}}(U).$$

A basis change applied transversally to many qubits may have small depth even when its total count is large. Conversely, a two-gate count saving can increase depth under locality constraints.

10.2.11 Open Clifford problems

Question 10.6. What are the defect orbits of the twenty-four three-qubit total-count targets under qubit permutations and parabolic conjugation?

Question 10.7. Does the maximum parabolic-versus-Clifford defect grow linearly, as $n \log n$, or faster under total primitive count?

Question 10.8. Can one prove a symbolic family with entangling-count defect greater than two, rather than relying on finite census data?

Question 10.9. Which finite symplectic parabolics are geodesically convex after SWAP is included as a primitive?

Conjectural frontier

The observed three-qubit defects should admit a small collection of symplectic root motifs analogous to the three-cross, one-phase classification in the unitriangular case. The present monograph records the exact census and witnesses but does not claim that classification.

10.3 Temporary nonlinearity: affine logic versus Toffoli-assisted synthesis

10.3.1 The question

Let

$$\mathrm{AGL}(n,2) = \{x \mapsto Ax + b : A \in \mathrm{GL}(n,2),\ b \in \mathbb{F}_2^n\}$$

be the affine reversible group. It is generated by NOT and CNOT gates. The ambient gate set adds Toffoli. On three bits it generates the full reversible group. On $n \geq 4$ bits with no ancillas, each NOT, CNOT, or Toffoli gate is an even permutation, so this alphabet does not generate the full symmetric group.

The catalytic question is:

> Can a circuit temporarily create nonlinear Boolean terms with Toffoli gates, reuse them, cancel them completely, and implement a final affine permutation with fewer gates than every affine circuit?

This is the nonlinear analogue of dirty linear memory.

10.3.2 Algebraic normal form

Every Boolean coordinate function has a unique algebraic normal form

$$f_i(x) = \bigoplus_{S \subseteq [n]} a_{i,S} \prod_{j \in S} x_j.$$

Affine maps have degree at most one. A Toffoli gate targeting t with controls a, b performs

$$x_t \longmapsto x_t + x_a x_b,$$

which can raise the degree of later coordinates. The final affine target requires every higher-degree coefficient to cancel.

A candidate lower-bound invariant must do more than track maximum degree. One Toffoli creates degree two and a second can remove it. Useful refinements include:

- the dimension of the quadratic monomial span,
- directional derivatives $D_u f(x) = f(x+u) + f(x)$,
- the rank of the bilinear polarization,
- the number of outputs consuming a nonlinear workspace term,
- restoration obligations for every nonlinear carrier.

10.3.3 Exact three-bit census

On three bits,

$$|\mathrm{AGL}(3,2)| = 2^3|\mathrm{GL}(3,2)| = 1344,$$

while the group generated by NOT, CNOT, and Toffoli is the full permutation group

$$S_8, \qquad |S_8| = 40320.$$

Both spaces can be enumerated exactly.

Exact finite computation

For three bits, every affine target has the same minimum cost in the affine gate set and in the Toffoli-assisted gate set under each of the following metrics:

1. total primitive gate count, where NOT, CNOT, and Toffoli cost one,
2. entangling count, where NOT is free and CNOT and Toffoli cost one.

There are no defective affine targets among all 1344 targets.

The restricted total-count diameter is seven and the ambient diameter on S_8 is eight. Under entangling count, both relevant diameters are six. Equality on affine targets is stronger than a diameter comparison because every target is checked individually.

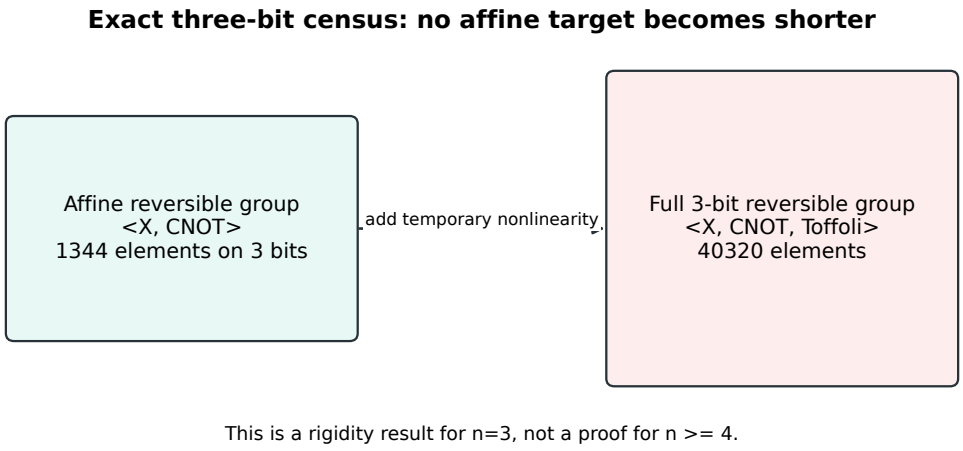


Figure 10.5. The complete three-bit affine-versus-Toffoli census is rigid.

10.3.4 Interpretation of the negative result

The census does not show that temporary nonlinearity is useless in general. It shows that three bits do not provide enough state space for a nonlinear carrier to repay its creation and cancellation cost under the stated alphabets.

The negative result is valuable for three reasons.

1. It gives a certified lower threshold: any total- or entangling-count defect requires at least four bits.
2. It prevents a forced analogy with the six-wire linear catalyst.

3. It supplies a complete regression test for future synthesis code.

10.3.5 A formal nonlinear catalyst

A clean one-workspace template is

$$T\, h_m \cdots h_1\, T^{-1},$$

where T is Toffoli and the middle gates use the quadratic term created in the workspace. Since Toffoli is self-inverse, the displayed circuit uses two nonlinear gates.

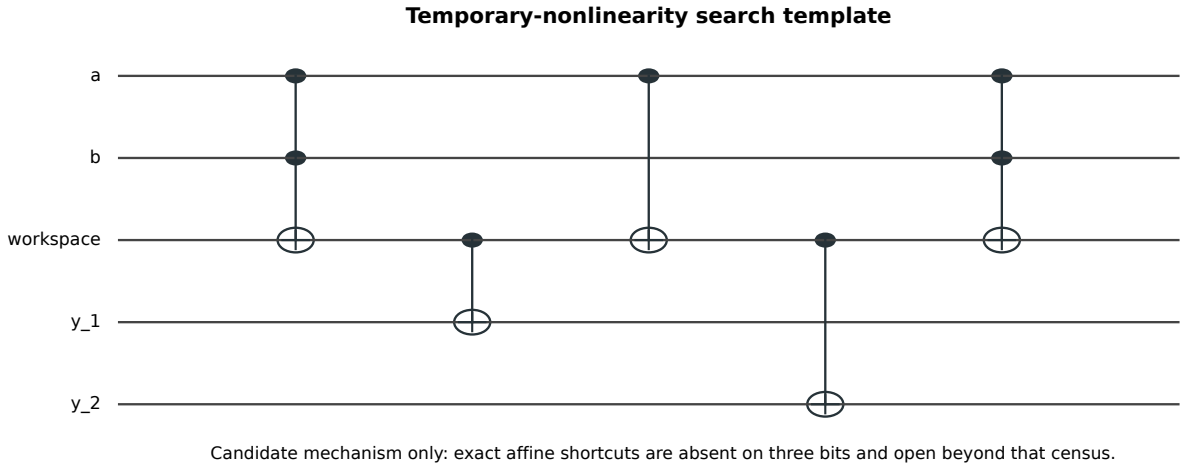


Figure 10.6. A temporary-nonlinearity template. It is a search skeleton, not a proved shortcut.

The key difficulty is that a final affine map cannot retain any product term. Every output that consumed the nonlinear carrier must be repaired, or the nonlinear contributions must cancel collectively.

10.3.6 Conjugation formulas

Let $T_{ab\to t}$ denote Toffoli. Conjugating an affine gate by T can produce nonlinear permutations. For example, if X_t flips the target,

$$T_{ab\to t} X_t T_{ab\to t} = X_t,$$

so no gain occurs. If a CNOT changes a control or exports the target, the conjugated map can contain several quadratic terms. The exact formula is easiest in algebraic normal form: substitute the Toffoli-updated coordinate into the affine map, then apply the inverse substitution.

A systematic search should classify pairs (T, h) for which

$$ThT^{-1} \in \mathrm{AGL}(n, 2).$$

For one Toffoli and an affine middle word, this is a normalizer-intersection problem. A useful next bounded test is to examine simple two-Toffoli sandwiches with an affine middle word of restricted length at most five. The present archive contains no reproducible implementation or receipt for this four-bit test, so no negative result from it is claimed here.

10.3.7 Potential lower-bound invariants

Quadratic support debt

Let $Q(f)$ be the span of quadratic monomials appearing across the coordinate functions. A Toffoli changes $Q(f)$ by at most a controlled rank-one contribution before subsequent substitutions. If several outputs consume the same quadratic term, a dynamic debt invariant may charge its creation, distribution, and cancellation.

Derivative rank

For an affine map, every derivative $D_u f$ is constant. During a nonlinear excursion, the derivative map

$$u \mapsto D_u f(x)$$

can depend on x. One can track the dimension of the nonconstant derivative space and require it to return to zero.

Higher-order Möbius coefficients

The ANF coefficients are the Möbius transform of the truth table. Projectors onto degree layers give an exact packet decomposition

$$f = f^{(0)} + f^{(1)} + f^{(2)} + \cdots .$$

A restricted affine circuit remains in degrees zero and one. The ambient circuit can populate higher packets temporarily. A successful lower bound must charge the useful flow through those packets, not merely their instantaneous dimension.

10.3.8 Four-bit search architecture

The full permutation group on four-bit states has size $16!$. The declared no-ancilla gate alphabet lies in its alternating subgroup: NOT, CNOT, and Toffoli are respectively products of eight, four, and two disjoint transpositions. Even this reduced ambient search is large enough to require a more structured approach. Several exact approaches remain available.

1. Meet in the middle for a fixed candidate target and length.
2. SAT or SMT encoding of a length-L circuit with affine endpoint constraints.
3. Bidirectional search restricted to states within a bounded nonlinear-degree or Toffoli-count budget.
4. Enumeration of two- or four-Toffoli normal forms modulo affine conjugation.
5. Mixed-integer or algebraic solving of ANF coefficient equations.

Protocol 10.10 (First four-bit threshold search)**.** 1. Compute exact affine distances for all $322{,}560$ affine targets.
2. Enumerate ambient words with two Toffoli gates modulo affine gates on the left and right.
3. Retain words whose endpoint is affine.
4. Compare with the exact affine distance.
5. If no defect appears, increase the Toffoli budget and produce an unsatisfiability certificate for every shorter length.

This protocol is a research plan. The packaged executable census stops at three bits, and neither the four-bit normal-form search nor its lower-bound certificates is supplied.

10.3.9 Resource variants

Temporary nonlinearity may be useful under one metric and useless under another. Natural costs include:

$$\text{Toffoli count}, \quad \text{CNOT count}, \quad T\text{-count after decomposition}, \quad T\text{-depth},$$
$$\text{ancilla count}, \quad \text{total depth}.$$

A Toffoli-assisted affine shortcut in total Boolean gates might disappear after fault-tolerant decomposition. Conversely, a circuit with more Boolean primitives can reduce T-depth.

10.3.10 Related nonlinear gate families

The same program applies to:

- CCZ-assisted diagonal Boolean phase maps,
- relative-phase Toffoli gates,
- multi-controlled X and Z gates,
- Reed–Muller degree hierarchies,
- reversible arithmetic over $\mathbb{Z}_d$,
- polynomial automorphisms over finite rings.

> **Conjectural frontier**
>
> There exists an affine target on finitely many bits whose total or fault-tolerant resource cost is reduced by a restored nonlinear excursion. The exact three-bit census rules out the smallest possible instance but supplies no evidence for a particular larger threshold.

10.3.11 Reproducible code

The implementation represents every reversible map as an exact permutation of 2^n basis states. Breadth-first search enumerates both the affine group and the full three-bit reversible group.

Listing 10.2. Exact three-bit affine-versus-Toffoli census.

```
from catalytic_gate_atlas import exact_affine_toffoli_census

print(exact_affine_toffoli_census(3, entangling_cost=False))
print(exact_affine_toffoli_census(3, entangling_cost=True))
```

The expected defect count is zero in both outputs.

10.4 Phase-polynomial and Clifford+T catalysis

10.4.1 Why the phase setting is different

Fault-tolerant quantum compilation rarely assigns equal cost to every gate. Clifford operations are often treated as inexpensive relative to non-Clifford gates, while connectivity and parity computation can dominate the remaining cost. Consequently there is no single Clifford+T defect. One must specify at least

$$\mathbf{c}(U) = \big(T\text{-count}, T\text{-depth}, \text{two-qubit count}, \text{two-qubit depth}, \text{ancillas}\big).$$

A catalytic excursion may improve one component and worsen another. The natural comparison is therefore Pareto-valued rather than scalar unless a hardware-specific weight vector is fixed.

Let $\omega = e^{i\pi/4}$. A phase-polynomial circuit on n qubits has the form [19, 41]

$$U|x\rangle = \omega^{P(x)}|Ax + b\rangle, \tag{10.1}$$

where $A \in \mathrm{GL}(n, 2)$, $b \in \mathbb{F}_2^n$, and $P : \mathbb{F}_2^n \to \mathbb{Z}_8$ is represented as a sum of binary linear forms with coefficients in $\mathbb{Z}_8$. The diagonal case has $A = I$ and $b = 0$.

The restricted model may be chosen in several ways:

1. diagonal phase gadgets only;
2. CNOT-dihedral circuits generated by X, CNOT, and T;
3. parity networks in which every intermediate transformation remains affine on computational basis labels;
4. a prescribed family of commuting Pauli rotations.

The ambient model may add Hadamards, arbitrary Clifford conjugations, measurements, or ancillas. These choices define different problems.

10.4.2 Pauli-rotation language

For a Pauli operator P and angle θ, write

$$R_P(\theta) = e^{-i\theta P/2}.$$

Every Clifford C satisfies

$$CR_P(\theta)C^\dagger = R_{CPC^\dagger}(\theta). \tag{10.2}$$

Thus a short ambient basis change can rotate a large family of phase gadgets into a different commuting or nearly commuting family, apply them there, and rotate back.

Proved statement

Equation (10.2) is an exact catalytic transport identity. It does not by itself prove a gate-count defect. A defect additionally requires a lower bound showing that the transported family is more expensive in the restricted representation than the opening and closing Clifford layers plus the transported implementation.

The simplest symbolic sandwich is

$$C\left(\prod_{j=1}^{m} R_{P_j}(\theta_j)\right)C^\dagger = \prod_{j=1}^{m} R_{CP_jC^\dagger}(\theta_j).$$

If the P_j commute, their conjugates commute. A catalyst can therefore change the geometry of the support without changing the algebra of the angles.

10.4.3 Phase polynomials and parity carriers

A CNOT network computes linear parities

$$\lambda(x) = a \cdot x \in \mathbb{F}_2.$$

Applying T^k to a wire carrying $\lambda(x)$ contributes

$$\omega^{k\lambda(x)}$$

to the phase polynomial. A parity network must create each required parity, expose it to a phase gate, and restore the linear map required at the endpoint.

This makes phase-polynomial synthesis a close relative of the snapshot families in the linear core. A parity row may evolve through

$$\lambda_0, \lambda_1, \ldots, \lambda_m,$$

with selected states consumed by phase gates. The difference is that consumption need not copy the parity to another row; it can deposit phase while leaving the carrier unchanged.

Research template

A candidate phase catalyst consists of:

1. a Clifford C that opens a non-diagonal frame;
2. an evolving Pauli or parity carrier in that frame;
3. several phase rotations consuming intermediate carrier states;
4. $C^\dagger$ restoring the original diagonal or CNOT-dihedral model;
5. a certificate that no restricted parity network attains the same resource vector.

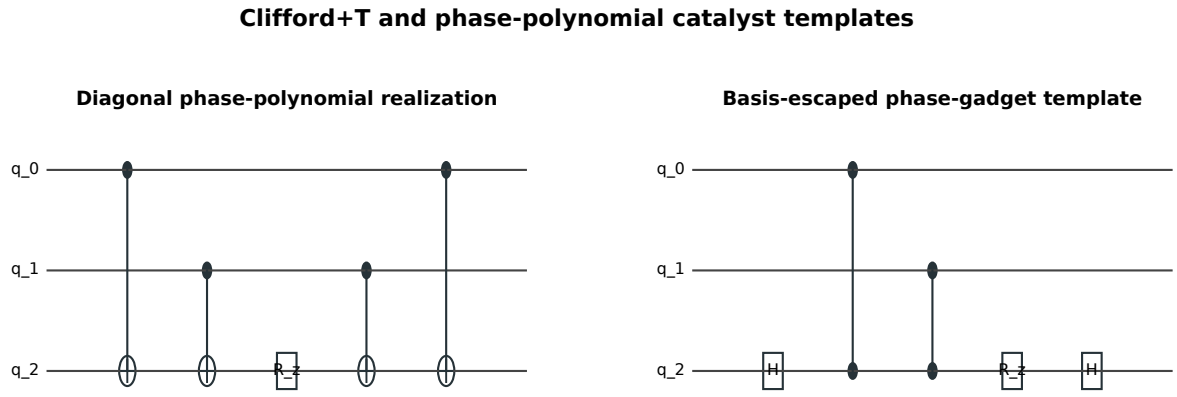


Figure 10.7. Symbolic Clifford+T catalyst: a basis excursion transports a family of phase obligations and is removed at the endpoint.

10.4.4 T-count as a coding problem

For diagonal Clifford+T synthesis without ancillas, equivalent phase-polynomial descriptions can differ by identities whose coefficients vanish modulo eight on Boolean inputs. Minimizing the number of

odd coefficients is naturally expressed as a coset-decoding problem in a punctured Reed–Muller-type code.

Let $a \in \mathbb{Z}_8^{2^n-1}$ index nonzero linear forms and let $\mathrm{odd}(a)$ denote the binary vector recording which coefficients are odd. An abstract restricted T-count is

$$\ell_T(P) = \min_{a:\, P_a = P} \mathrm{wt}(\mathrm{odd}(a)).$$

An ambient Clifford excursion changes not only the coefficient vector but also the Pauli frame in which the rotations are represented. A lower bound that survives ambient Clifford conjugation must therefore be invariant under the relevant symplectic action or must explicitly charge the frame change.

10.4.5 T-depth and commuting layers

Two phase gadgets may share a T layer when a suitable Clifford network maps their parities to distinct wires or when ancilla resources permit parallel extraction. Define the conflict graph Γ_P whose vertices are odd phase terms and whose edges encode inability to realize two terms in one layer under the chosen ancilla and connectivity rules. Then

$$T\text{-depth}(P) \geq \chi(\Gamma_P)$$

for any conflict graph whose edge relation is sound. Catalysis can alter Γ_P by changing the Pauli frame. The right question is not merely whether C preserves the number of terms, but whether it reduces the chromatic or scheduling obstruction enough to repay its own depth.

10.4.6 A vector-valued defect

For a target U in a restricted phase model, define

$$\mathcal{C}_{\mathrm{res}}(U) = \{\mathbf{c}(\mathcal{W}) : \mathcal{W} \text{ is a restricted circuit for } U\}$$

and similarly $\mathcal{C}_{\mathrm{amb}}(U)$. Their Pareto frontiers are denoted $\mathrm{Pareto}_{\mathrm{res}}(U)$ and $\mathrm{Pareto}_{\mathrm{amb}}(U)$.

Definition 10.11 (Pareto catalytic defect)**.** A target has a Pareto catalytic defect when the ambient Pareto frontier contains a point strictly dominating every restricted cost vector in at least one declared region of resource space.

Scalarization by positive weights w gives

$$\Delta_w(U) = \min_{c \in \mathcal{C}_{\mathrm{res}}(U)} w \cdot c - \min_{c \in \mathcal{C}_{\mathrm{amb}}(U)} w \cdot c.$$

Different w can change the magnitude of the defect and whether it is zero or positive. It remains nonnegative because every restricted realization is also an ambient realization with the same resource vector.

10.4.7 Exact finite-search model

A finite exact atlas can be built [34, 20] for small n by representing Clifford+T operators in one of three ways.

1. Exact matrices over the cyclotomic ring $\mathbb{Z}[1/\sqrt{2}, i]$ modulo global phase.
2. Normal forms combining a binary symplectic Clifford with a phase-polynomial record.

3. Quotients by left and right Clifford actions when only T-count is charged.

The third model can be dramatically smaller but requires care: quotienting is sound only when the free Clifford actions in the cost model agree with the quotient relation.

Protocol 10.12 (Small-width Clifford+T threshold atlas)**.**
1. Fix exact versus approximate synthesis, ancilla rules, and the resource vector.
2. Enumerate the restricted target subgroup or a complete set of normal forms.
3. Compute restricted Pareto frontiers by multiobjective shortest path.
4. Enumerate ambient states or use meet in the middle for candidate targets.
5. Store an optimal witness and a lower-bound certificate for every defect.
6. Classify defects under qubit permutation and free Clifford equivalence.

10.4.8 Lower-bound candidates

Odd-support rank

The span of odd phase terms is invariant under some restricted rewrites but transforms covariantly under linear changes of variables. Its dimension gives a coarse obstruction.

Pauli support geometry

Represent a commuting Pauli family by an isotropic subspace of $\mathbb{F}_2^{2n}$. Restricted diagonal circuits remain in the Z Lagrangian. Ambient Clifford circuits move through the symplectic Grassmannian. Distance from the Z Lagrangian, together with the number of deposited rotations, is the phase analogue of microscopic flag escape.

Moment and derivative packets

For a phase function P, finite differences

$$D_u P(x) = P(x+u) - P(x) \pmod 8$$

separate affine, quadratic, and higher interactions. A hierarchy of derivative tensors can record temporary flow into nonlinear phase packets.

Architecture-aware parity rank

On a coupling graph, each parity has a Steiner cost. A catalyst can be profitable by making several required parities share a Steiner tree. A sound lower bound may combine cut rank with edge congestion and restoration debt.

10.4.9 Tensor and repeated-module questions

For targets U and V, independent synthesis gives

$$\ell_T(U \otimes V) \le \ell_T(U) + \ell_T(V)$$

when the systems are disjoint and local Clifford cost is ignored. Joint synthesis may be better because phase identities can couple terms across modules. This motivates a stable regularization

$$\tau_T(U) = \liminf_{r\to\infty} \frac{\ell_T(U^{\otimes r})}{r}.$$

A one-copy rigid phase target could still display tensor activation.

Conjectural frontier

There are diagonal or CNOT-dihedral targets whose Pareto frontier is strictly improved by a restored non-diagonal Clifford excursion. The smallest qubit threshold and the resource component in which the first defect appears are open in this monograph.

10.4.10 What would count as a theorem

A publishable first result in this direction should contain all of the following:

1. an exact target family and gate alphabets;
2. explicit ambient circuits;
3. a restricted lower bound for the same declared cost;
4. exact treatment of global phase and ancillas;
5. classification below the first threshold or a proof of minimality;
6. reproducible symbolic code.

Merely rewriting a phase gadget with Hadamards is not sufficient. The decisive content is the optimality comparison.

10.5 Gaussian, matchgate, and fermionic catalytic models

10.5.1 Two Gaussian settings

Quantum Gaussianity occurs [45] in two closely related but algebraically different settings.

For n bosonic modes, a Gaussian unitary acts on quadratures

$$\mathbf{r} = (q_1, \dots, q_n, p_1, \dots, p_n)^T$$

by an affine symplectic transformation

$$\mathbf{r} \longmapsto S\mathbf{r} + d, \qquad S \in \mathrm{Sp}(2n, \mathbb{R}).$$

For n fermionic modes, a fermionic Gaussian unitary acts on $2n$ Majorana operators by

$$\gamma \longmapsto O\gamma, \qquad O \in SO(2n).$$

Nearest-neighbor matchgates give a circuit presentation of a fermionic Gaussian subgroup on qubits.

In both cases one can choose a restricted subgroup preserving a flag, mode ordering, particle number, or bipartition, and compare it with a larger Gaussian or non-Gaussian gate set.

10.5.2 An inherited symplectic shear theorem

Every invertible linear transformation $A \in \mathrm{GL}(n, R)$ has a canonical symplectic embedding

$$\iota(A) = \begin{pmatrix} A & 0 \\ 0 & A^{-T} \end{pmatrix}. \tag{10.3}$$

Indeed, with

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix},$$

one checks

$$\iota(A)^T J \iota(A) = J.$$

Thus every row-transvection identity in the established linear theory induces an exact identity between symplectic shear circuits.

Proved statement

Under a Gaussian gate alphabet in which each embedded elementary transvection $\iota(I + aE_{ij})$ is one primitive and direction is the only restriction, the universal six-row catalytic atom embeds into a twelve-quadrature Gaussian atom with restricted length seven and ambient length six.

This is a formal inheritance statement for the declared shear alphabet. It does not assert optimality after arbitrary beam splitters, squeezers, Fourier transforms, or local Gaussian gates are admitted at unit cost.

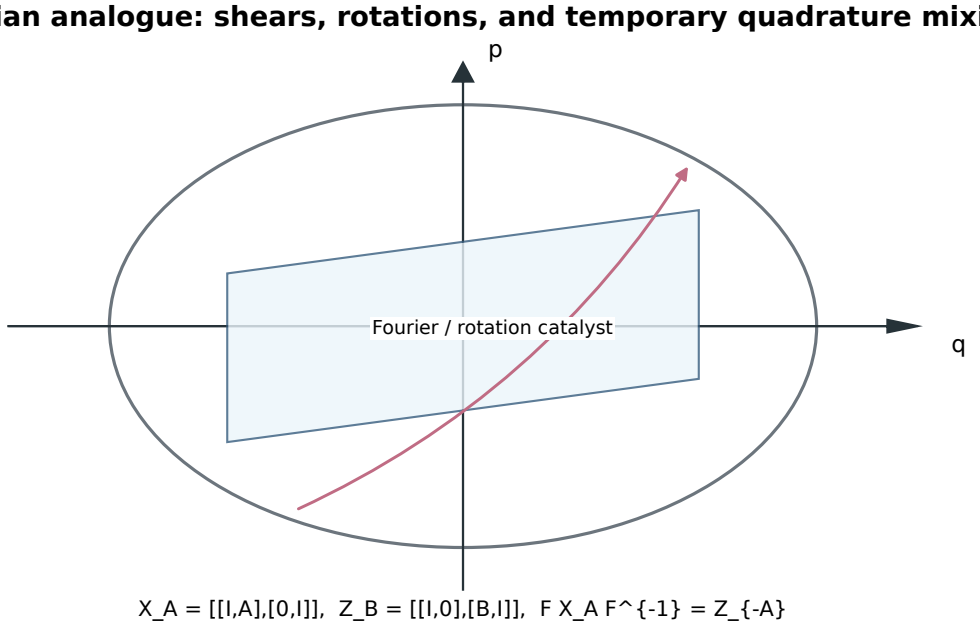


Figure 10.8. The row-transvection atom embedded as a Gaussian symplectic shear circuit.

10.5.3 Bosonic gate hierarchies

Natural nested models include

triangular shears $\subset$ all Gaussian unitaries $\subset$ Gaussian plus a non-Gaussian primitive $\subset U(\mathcal{H})$.

Typical primitive resources are:

- SUM or controlled displacement gates,

- beam splitters,
- single-mode and two-mode squeezers,
- Fourier transforms,
- cubic phase gates,
- photon counting or non-Gaussian state injection.

Each inclusion supports a different defect. A target Gaussian transformation might be shortened by leaving a triangular decomposition but remaining Gaussian. A more radical question asks whether temporary non-Gaussianity can shorten a final Gaussian target.

10.5.4 Bloch–Messiah versus catalytic synthesis

A bosonic Gaussian unitary admits a Bloch–Messiah decomposition

$$S = K_1 D K_2,$$

where K_1, K_2 are passive symplectic transformations and D is a diagonal squeezing layer. This normal form optimizes conceptual structure, not necessarily a weighted gate count on a specific architecture.

Catalytic synthesis asks whether a short excursion into a basis where many shear or squeezing obligations align can reduce total cost. The opening and closing passive layers play the role of the catalyst.

Research template

A Gaussian catalytic theorem requires:

1. a declared primitive library and cost vector;
2. a target subgroup, such as triangular, number conserving, or local;
3. an explicit sandwich KHK^{-1} returning to the target subgroup;
4. a lower bound based on symplectic rank, singular values, graph cuts, or mode support;
5. treatment of approximation error when continuous parameters are synthesized numerically.

10.5.5 Fermionic Gaussian and matchgate circuits

Let $c_j, c_j^\dagger$ be fermionic creation and annihilation operators. Quadratic Hamiltonians generate Gaussian unitaries, and their action is determined by an orthogonal transformation on Majoranas. Number-conserving Gaussian operations correspond to a unitary action on modes, while general Gaussian operations allow particle-hole mixing.

This yields the hierarchy

$$U(n) \text{ mode transformations} \subset SO(2n) \text{ fermionic Gaussian transformations} \subset \text{parity-preserving unitaries}.$$

A temporary Bogoliubov transformation can violate number conservation while preserving parity. The target may nevertheless return to the number-conserving subgroup.

Definition 10.13 (Bogoliubov catalytic defect)**.** For a number-conserving Gaussian target U, compare the minimum cost using number-conserving primitives with the minimum cost using general Gaussian primitives, requiring the endpoint to conserve number exactly.

10.5.6 Matchgates and geometry

Nearest-neighbor matchgate circuits [16, 10] have strong classical simulation properties. Their geometry changes when SWAP, fermionic SWAP, long-range connectivity, or a single non-matchgate resource is added. Candidate restricted/ambient pairs include:

1. nearest-neighbor matchgates versus all-to-all matchgates;
2. number-conserving matchgates versus general matchgates;
3. matchgates versus parity-preserving two-qubit gates;
4. Gaussian circuits versus Gaussian circuits with one reusable non-Gaussian ancilla.

A compiler application is immediate: a temporary mode permutation or particle-hole rotation may allow a batch of otherwise expensive interactions to share one routing or basis-change layer.

10.5.7 Covariance-matrix lower bounds

For a Gaussian state, the covariance matrix Γ transforms as

$$\Gamma \longmapsto S\Gamma S^T$$

in the bosonic case and

$$\Gamma \longmapsto O\Gamma O^T$$

in the fermionic case. Possible restricted lower bounds include:

- rank change of off-diagonal covariance blocks,
- symplectic singular-value obligations,
- growth of mode support,
- graph bandwidth and cut rank,
- number of nontrivial canonical angles between invariant subspaces.

One elementary gate changes only a controlled portion of these data. A useful invariant must also charge preparation of a carrier that makes progress on many cuts simultaneously.

10.5.8 Temporary non-Gaussianity

Suppose a non-Gaussian primitive g is introduced and later removed:

$$U = gh_m \cdots h_1 g^{-1}, \qquad U \in G_{\text{Gauss}}.$$

If $gh_i g^{-1}$ generates several useful Gaussian effects, the excursion may be catalytic. However, many standard non-Gaussianity monotones vanish only on states rather than unitaries, and cancellation at the endpoint complicates direct lower bounds.

Candidate monotones and certificates include:

- degree in the Weyl characteristic function,
- Wigner negativity generated during the path,
- operator Schmidt rank across mode cuts,
- polynomial degree of Heisenberg-evolved quadratures,
- stabilizer rank for encoded finite-dimensional subspaces.

10.5.9 Exact and numerical search

Finite exact enumeration is available for binary symplectic or orthogonal quotients, but continuous Gaussian groups require a different strategy.

1. Fix a symbolic gate skeleton and solve polynomial equations for parameters.
2. Use interval arithmetic to certify nonexistence below a length.
3. Exploit canonical decompositions to quotient local or passive layers.
4. For discrete-angle Gaussian libraries, use exact algebraic-number arithmetic.
5. Store both an upper-bound circuit and a dual or algebraic certificate.

10.5.10 Architecture and routing

On a line or grid, Gaussian and matchgate circuits may be dominated by routing. Let G_{conn} be the hardware graph. A mode permutation π can be treated as a catalyst if it is opened, many interactions are executed in the reordered frame, and π^{-1} restores the logical labeling. The benefit is

$$\text{routing saved across consumers} - 2\,\mathrm{cost}(\pi).$$

This is the same amortization law as the linear snapshot family, now applied to mode geometry.

> **Conjectural frontier**
>
> There exist number-conserving or nearest-neighbor Gaussian targets whose minimum weighted cost is reduced by a restored Bogoliubov, mode-permutation, or non-Gaussian excursion. The first exact threshold depends on the primitive library and is not determined here.

10.5.11 Research deliverables

A complete Gaussian extension should produce:

1. a finite or semialgebraic model with exact endpoint equality;
2. at least one nontrivial lower-bound invariant;
3. a minimal catalytic atom;
4. a weighted hardware interpretation;
5. a tensor or repeated-mode amplification theorem;
6. code using exact arithmetic or certified intervals.

10.6 Symmetry, ancillas, measurement, and architecture

10.6.1 Beyond subgroup word metrics

Some of the most important quantum resource enlargements are not inclusions of groups. Measurement introduces branches, reset is irreversible, and ancillas enlarge the Hilbert space. The appropriate abstraction is a nested pair of realization categories

$$\mathcal{C}_{\mathrm{res}} \subseteq \mathcal{C}_{\mathrm{amb}},$$

with a cost assigned to generating morphisms.

For a target channel Φ in the restricted category, define

$$c_{\mathrm{res}}(\Phi) = \inf\{\mathrm{cost}(W) : W \text{ realizes } \Phi \text{ in } \mathcal{C}_{\mathrm{res}}\}$$

and similarly $c_{\mathrm{amb}}(\Phi)$. The value $+\infty$ is allowed. A finite positive difference is a metric defect; an infinite difference is an existence defect.

10.6.2 Symmetry-preserving circuits

Let Q be a conserved charge and let

$$H_Q = \{U : [U, Q] = 0\}.$$

Restricted circuits use gates commuting with Q. Ambient circuits may temporarily leave H_Q but must return to a charge-preserving endpoint.

Examples include:

- particle-number conservation,
- parity conservation,
- total spin or magnetization,
- lattice gauge constraints,
- stabilizer or CSS sector preservation,
- block-diagonal representation sectors.

The Hilbert space decomposes as

$$\mathcal{H} = \bigoplus_q \mathcal{H}_q.$$

A restricted circuit is block diagonal. An ambient catalyst may couple sectors, use the enlarged route, and decouple them at the endpoint.

Research template

A symmetry-breaking catalyst has three obligations:

1. quantify how far the path leaves the symmetry algebra;
2. prove exact return, not merely return on a chosen input state;
3. compare the cost with every symmetry-preserving realization of the full target operator.

10.6.3 Sector-flow invariants

Let P_q be the projector onto sector q. For an intermediate unitary U_t, define transfer blocks

$$T_{q,r}(t) = P_q U_t P_r.$$

A restricted path has $T_{q,r} = 0$ for $q \neq r$. Ambient paths may populate off-diagonal blocks. Possible excursion measures include

$$\sum_{q \neq r} \operatorname{rank} T_{q,r}, \qquad \max_{q \neq r} \|T_{q,r}\|, \qquad \text{bandwidth in } |q - r|.$$

These are analogues of strict-upper rank and bandwidth in the unitriangular theory.

10.6.4 Clean and dirty ancillas

A clean ancilla starts in a specified state, usually $|0\rangle$, and must be returned to that state. A dirty ancilla starts in an unknown state and must be restored for every input.

Let U act on the data register. A clean-ancilla realization is a unitary W satisfying

$$W(|\psi\rangle \otimes |0\rangle) = (U|\psi\rangle) \otimes |0\rangle \quad \text{for every } |\psi\rangle.$$

A dirty-ancilla realization requires

$$W(|\psi\rangle \otimes |\phi\rangle) = (U|\psi\rangle) \otimes |\phi\rangle \quad \text{for every } |\psi\rangle, |\phi\rangle.$$

The second condition is strictly stronger.

Ancillas define a family of defects

$$\Delta_{a,\text{clean}}(U), \qquad \Delta_{a,\text{dirty}}(U),$$

where a is the number of ancillary qubits or modes. Stable ancilla capacity asks whether a fixed number of reusable ancillas can yield an unbounded saving as the data size grows.

10.6.5 Measurement and feedforward

A measurement-assisted realization is described by an instrument with classical outcomes m, conditional channels Φ_m, and feedforward operations. Exact deterministic implementation of a unitary target requires every branch, after correction, to induce the same channel.

The cost vector may include

$$(T\text{-count}, \text{two-qubit count}, \text{measurement rounds}, \text{classical latency}, \text{ancilla qubits}).$$

A measurement can create an existence defect by realizing a channel unavailable unitarily under the same locality or depth constraints, or a metric defect by reducing one resource while increasing another.

Definition 10.14 (Branch-catalytic realization)**.** A measurement-assisted circuit is branch catalytic when the intermediate classical or quantum branch information is absent from the final data channel and all declared work systems are reset or discarded according to the model.

10.6.6 Teleportation as a canonical ambient excursion

Gate teleportation replaces direct implementation by:

$$\text{prepare resource state} \to \text{entangle and measure} \to \text{classical correction}.$$

The ambient category contains resources not present in the target channel: entanglement, measurement outcomes, and feedforward. The endpoint can nevertheless be a deterministic unitary.

This is an archetypal realization defect, but careful accounting is mandatory. A claimed saving must include resource-state preparation or explicitly treat the state as an external catalyst.

10.6.7 Architecture-constrained gates

Let $\Gamma = (V, E)$ be a hardware graph. A restricted two-qubit gate may act only on edges of E. The ambient model may permit a small number of nonlocal interactions, temporary logical relabelings, teleportation edges, or bus-mediated gates.

For a target U, define

$$\ell_\Gamma(U) = \text{minimum hardware-respecting cost},$$

and compare it with an enlarged graph Γ'. A matched routing excursion has the form

$$\pi\, W\, \pi^{-1},$$

where π is a logical permutation. The excursion is profitable when many gates in W become local in the permuted frame.

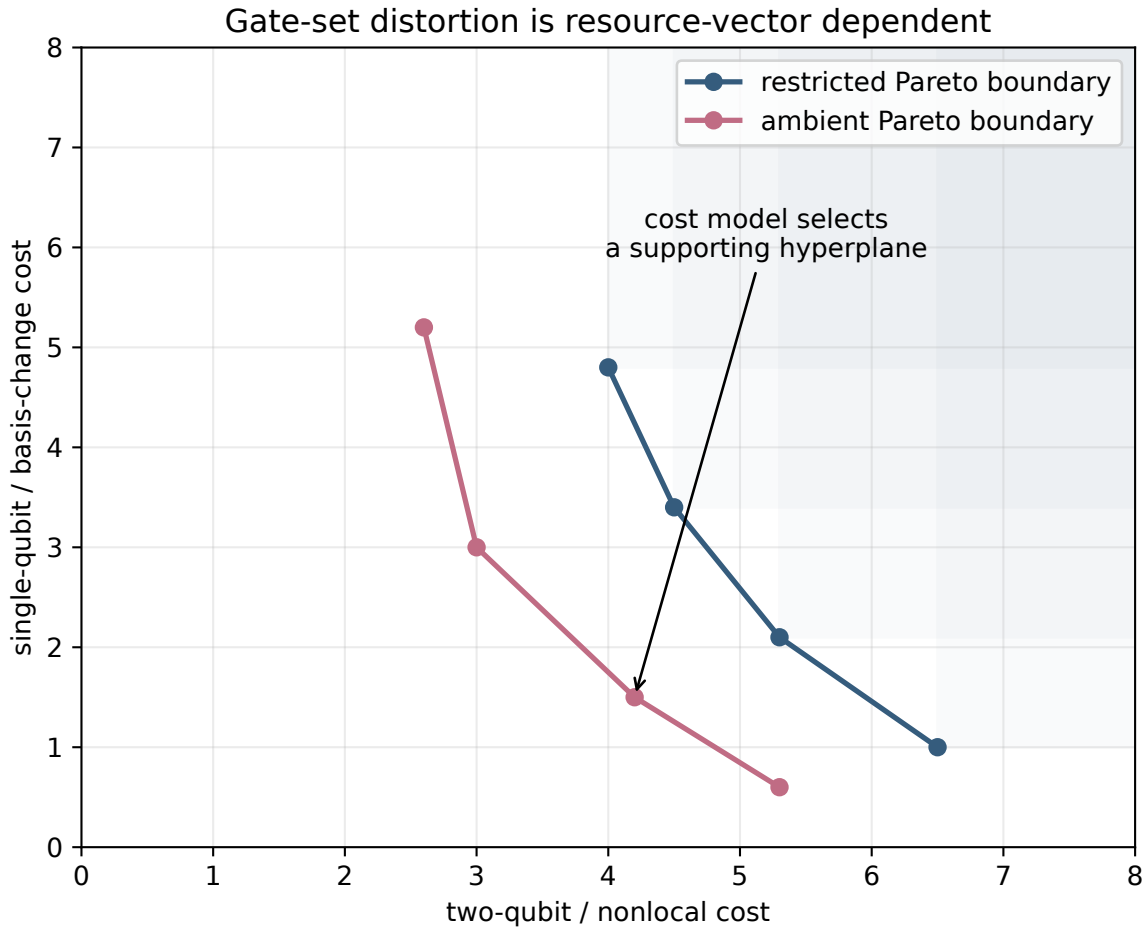


Figure 10.9. Gate-set defects generally live in a multi-resource Pareto geometry.

10.6.8 Depth, count, and space-time volume

Let g_t be the set of gates scheduled in layer t. Common scalar costs are

$$\sum_t |g_t|, \qquad \#\{t : g_t \neq \varnothing\}, \qquad \sum_t \#\{\text{active qubits in layer } t\}.$$

A catalyst may reduce count but increase depth, or vice versa. Space-time volume is often the more stable hardware quantity.

For stochastic error rates p_g, a first-order weighted cost is

$$C_{\text{err}}(W) = \sum_{g \in W} -\log(1 - p_g).$$

This additive approximation can be inserted directly into Dijkstra search. It does not capture coherent or correlated errors, but it produces a declared and reproducible metric.

10.6.9 Approximate synthesis

For continuous gate sets, define

$$\ell_{\text{res},\varepsilon}(U) = \min\{\text{cost}(W) : \|W - U\| \leq \varepsilon\}.$$

There are two distinct border phenomena:

1. algebraic border length, where a target lies in the closure of a bounded exact word locus;
2. analytic approximation length, where an error tolerance is explicitly permitted.

They should not be conflated.

A catalyst can improve the asymptotic Solovay–Kitaev or number-theoretic approximation constant even when exact synthesis is impossible in both models.

10.6.10 Resource theories and monotones

A quantum resource theory provides free states, free operations, and monotones. Gate-set distortion adds a path-sensitive question: the endpoint may be free even though the path temporarily creates resource.

A state monotone $M(\rho)$ alone may fail because the catalyst acts at the channel level. Useful channel-level quantities include:

- robustness or magic capacity,
- entangling power,
- operator Schmidt rank,
- Choi-state resource measures,
- symmetry asymmetry of the channel,
- amortized resource generation.

A lower bound must relate the cumulative or peak resource to the declared gate cost and account for cancellation.

10.6.11 A categorical amplification principle

Suppose a monoidal category admits a tensor product and a cost satisfying

$$c_{\mathrm{amb}}(f \otimes g) \leq c_{\mathrm{amb}}(f) + c_{\mathrm{amb}}(g).$$

If restricted cost is additive on a target family, then any one-copy defect amplifies linearly under independent tensor powers. Superlinear amplification requires a normalization in which dimensions multiply, as in the matrix tensor rule of the core theory, or a joint-synthesis synergy across copies.

10.6.12 Experimental standards

Every architecture, measurement, or ancilla claim should state:

1. gate alphabet and native connectivity;
2. clean, dirty, catalytic, or consumable ancilla convention;
3. deterministic or postselected implementation;
4. exact or approximate endpoint condition;
5. total, weighted, depth, or Pareto cost;
6. whether resource-state preparation is included;
7. whether qubit relabeling is free;
8. certificate of restoration on all inputs.

Without these declarations, two reported "gate counts" may refer to incomparable problems.

> **Conjectural frontier**
>
> For several common symmetry and architecture restrictions, a constant-size restored excursion should yield an unbounded saving on structured target families. The unitriangular construction proves this phenomenon for directed linear information flow; analogous theorems for charge sectors, measurement branches, and nonlocal routing remain open.

CHAPTER 11

Packets, types, holonomy, and restricted realization

11.1 Packets, types, holonomy, and restricted realization

11.1.1 Why coarse spectra are not enough

A finite symmetry problem often comes with a natural commuting operator algebra. Its joint spectrum is attractive because it is canonical, compact, and computationally cheap. Yet the joint eigenvalues and their multiplicities can be identical across objects whose realization behavior is different.

This chapter develops the hierarchy

$$\text{spherical packet} \prec \text{embedded type packet} \prec \text{holonomy packet} \prec \text{realization fiber}. \tag{11.1}$$

The relation $\prec$ means "retains strictly more potentially relevant information," not that every refinement is complete.

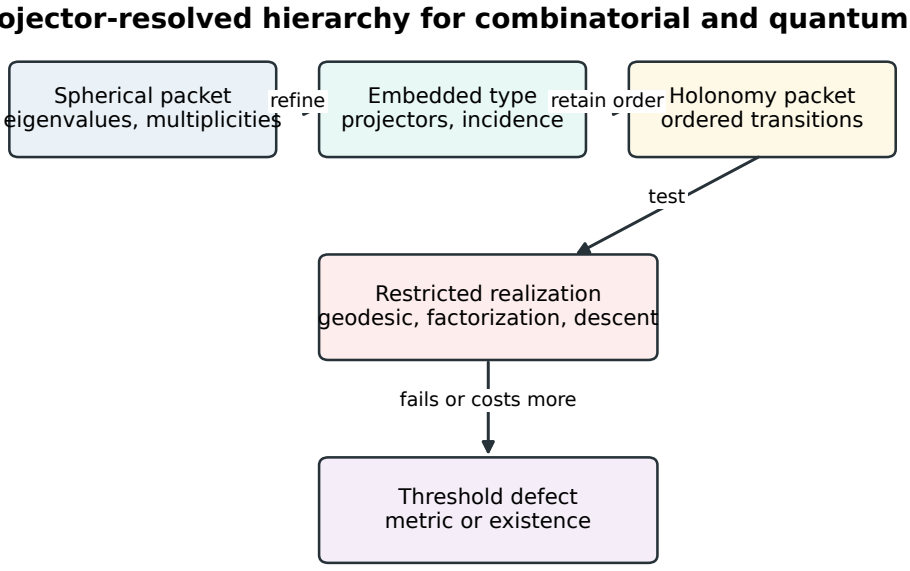


Figure 11.1. The packet hierarchy used to organize finite threshold systems.

11.1.2 Spherical packets

Let $T_1, \dots, T_m$ be commuting normal operators on a finite-dimensional Hilbert space. Their unique coarsest joint spectral resolution is

$$T_j = \sum_{\lambda} t_j(\lambda) B_\lambda, \qquad \sum_{\lambda} B_\lambda = I.$$

The tuple

$$\lambda = (t_1(\lambda), \dots, t_m(\lambda))$$

is the spherical label. The abstract spectral packet records these labels and the ranks of B_λ.

Two different embedded systems can have the same abstract packet. The actual matrices B_λ, endpoint coefficients, and transition operators may still differ.

Principle 11.1 (Spherical collapse)**.** *Passing from embedded operators to eigenvalues and multiplicities can erase exactly the incidence or transition data that controls restricted realization.*

11.1.3 The Latin-square association algebra

A Latin square L of order n has n^2 cells. Let R, C, S be adjacency matrices for equality of row, column, and symbol. The algebra generated by I, J, R, C, S is commutative and its abstract multiplication table depends only on n, not on the paratopy class of L.

Define

$$Q_R = \frac{I + R}{n}, \qquad Q_C = \frac{I + C}{n}, \qquad Q_S = \frac{I + S}{n}, \qquad B_0 = \frac{J}{n^2}.$$

Then the five joint projectors are

$$B_0, \quad Q_R - B_0, \quad Q_C - B_0, \quad Q_S - B_0, \quad I - Q_R - Q_C - Q_S + 2B_0.$$

Their multiplicities are

$$1, \quad n-1, \quad n-1, \quad n-1, \quad (n-1)(n-2).$$

The Latin-square graph $R + C + S$ has universal spectrum

$$3(n-1)^1, \qquad (n-3)^{3(n-1)}, \qquad (-3)^{(n-1)(n-2)}.$$

At $n = 6$ this becomes

$$15^1, \qquad 3^{15}, \qquad (-3)^{20}.$$

Thus the graph spectrum cannot distinguish the twelve main classes of order six.

> **Proved statement**
>
> The displayed projector formulas depend only on the Latin-square axioms and the order. They therefore prove that the abstract R, C, S joint eigenvalue packets cannot distinguish squares of the same order. This conclusion does not require an enumeration of the order-six main classes.

11.1.4 Transition fusion also collapses

Let ρ_i be the row permutations of a Latin square. The symmetrized aggregate of all pair transitions satisfies

$$\sum_{i<j} \left(P_{\rho_i \rho_j^{-1}} + P_{\rho_j \rho_i^{-1}} \right) = n(J - I). \tag{11.2}$$

Hence first-order fusion destroys the class information as well.

The lesson is general: summing typed transitions can be as destructive as discarding the embedded projectors. Order and labels matter.

11.1.5 Embedded type data

A type packet retains the actual projectors, support partitions, distinguished incidence sets, or local representation types. Examples include:

- the row, column, and symbol partitions of a Latin square;
- a projective-line closure system;
- the flag stabilized by a circuit subgroup;
- support patterns of unitary or quantum-Latin entries;
- ramification or block data in a finite group action.

A projective-line closure procedure can supply additional type operators, such as adjacency operators of a line spread. Whether a particular added operator separates two classes is a further question. The present archive does not include the separate order-six projective computation, so it supplies no new finite certification of that comparison.

11.1.6 Holonomy packets

Let a family of local charts or rows be represented by permutations ρ_i. Define normalized transitions

$$h_{ij} = \rho_i \rho_j^{-1}$$

and the transition group

$$H_r(L) = \langle h_{ij} : i, j \rangle.$$

A holonomy packet may include:

- the isomorphism type or order of $H_r(L)$,
- the cycle-type inventory of the distinguished h_{ij},
- block systems and orbit partitions,
- characters or matrix coefficients of words in the h_{ij},
- noncommutative moments
$$\mathrm{tr}(h_{i_1 j_1} \cdots h_{i_k j_k}),$$
- the corresponding packets for row, column, and symbol roles.

Isotopy conjugates these transition groups; paratopy permutes the three roles. Therefore the unordered multiset of role signatures is a paratopy invariant.

Research template

For a specified collection of Latin squares, one can compute transition-group orders, pair-transition cycle inventories, and nontrivial block sizes, then test whether the resulting role signatures separate the collection. The separate order-six input tables and verifier are not included in this archive, so pairwise separation of all twelve main classes is an open verification obligation here.

Even a complete discrete classification would not prove orthonormal representability. Noncommutative transition data are candidate refinements of the information lost under spherical fusion.

11.1.7 Imprimitivity as hidden decomposition

A transitive permutation group is imprimitive when it preserves a nontrivial block system. In combinatorial designs, imprimitivity often signals a subsquare, product, congruence, or module decomposition.

For a Latin square of order $2m$, normalize one row to the identity permutation on the symbols. An $m \times m$ Latin subsquare determines a partition of the symbols into its m symbols and their complement. Rows meeting the subsquare preserve the two corresponding column blocks, and the remaining rows exchange them. Thus normalized row transitions preserve this two-block system. Conversely, if all normalized row transitions preserve a partition into two blocks of size m, exactly m rows send a fixed column block into a fixed symbol block, producing an $m \times m$ Latin subsquare. This is a structural criterion. Counting the order-six classes with this property requires separate input representatives.

Principle 11.2 (Clifford-type decomposition)**.** *Combinatorial substructure can be detected as imprimitivity of a transition packet. The spherical packet sees aggregate multiplicities; the holonomy packet sees how those multiplicities are assembled into blocks.*

11.1.8 Realization fibers

Let P denote packet data and let

$$\mathrm{Real}_{\mathrm{res}}(P)$$

be the space of restricted realizations compatible with P. The packet may completely classify the discrete object while the realization fiber remains empty, positive dimensional, disconnected, or singular.

Examples of realization conditions include:

- a flag-preserving geodesic with a declared length;
- a product-state orthonormal representation;
- a positive semidefinite Gram matrix with a zero pattern;
- a tensor factorization;
- a global descent object compatible with local packets;
- a circuit satisfying ancilla restoration on every input.

Thus classification and realization are distinct layers.

11.1.9 Quantum Latin squares as an existence defect

A quantum Latin square is an array [11] of vectors whose rows and columns are orthonormal bases. Ball and Simoens prove that a pair of non-entangled mutually orthogonal quantum Latin squares of

order six does not exist, while earlier work provides entangled order-six solutions and hence $\mathrm{AME}(4,6)$ states [8, 12, 39].

This is naturally expressed as

$$\mathrm{Real}_{\mathrm{product}}(P_6) = \varnothing, \qquad \mathrm{Real}_{\mathrm{entangled}}(P_6) \neq \varnothing.$$

It is an existence defect, not a gate-count defect.

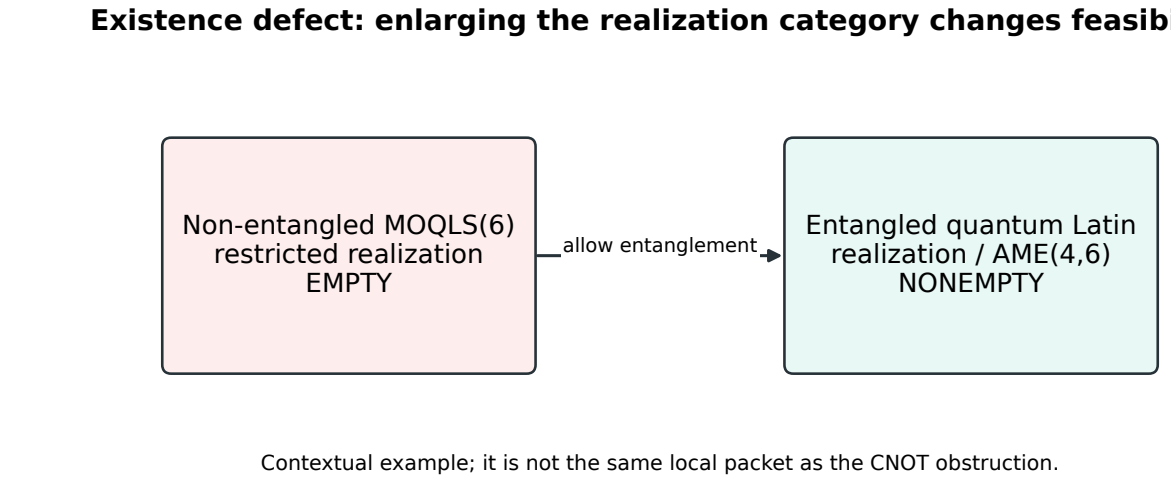


Figure 11.2. The order-six quantum-Latin distinction is a restricted-versus-ambient realization gap.

The source proof uses support patterns, a reduction to one classical square, orthonormal representations of complements of Latin-square graphs, projective-line closure, finite case analysis over twelve main classes, and a continuous argument for the two remaining subsquare cases. No theorem in the unitriangular or Clifford chapters is inferred from this result.

11.1.10 First-arrival packets

For a finite generator set S, the endpoint Green kernel

$$\mathcal{G}_{S,g}(z) = \langle \delta_g, (I - zA_S)^{-1}\delta_e \rangle$$

contains more information than the spectrum of A_S. Its first nonzero degree is the distance to g.

If

$$A_S = \sum_{\lambda} \lambda B_\lambda,$$

then

$$\mathcal{G}_{S,g}(z) = \sum_{\lambda} \frac{\langle \delta_g, B_\lambda \delta_e \rangle}{1 - \lambda z}.$$

The endpoint matrix coefficients of the embedded projectors determine cancellation between spectral components. This is the metric analogue of retaining a type rather than only an unramified eigencharacter.

11.1.11 Typed Green kernels

Let $H \leq G$ and let $\mu_k(h)$ count ambient words of length k with holonomy $h \in H$. For an irreducible representation τ of H, define

$$\widehat{\mu}_k(\tau) = \sum_{h \in H} \mu_k(h)\tau(h)$$

and

$$\widehat{\mathcal{G}}_\tau(z) = \sum_{k \geq 0} \widehat{\mu}_k(\tau) z^k.$$

Fourier inversion recovers every $\mu_k(h)$. The trivial representation records only total return to the subgroup; nontrivial types resolve the holonomy.

Proved statement

For finite H, the full family of typed Green kernels over $\widehat{H}$ determines every endpoint Green series and therefore every ambient word distance from the identity to an element of H.

11.1.12 Operational Langlands analogy

The hierarchy resembles the local organization of representation theory:

$$\text{spherical character} \to \text{type} \to \text{ramification or monodromy} \to \text{realization and descent.}$$

The analogy is useful because it prevents coarse spectral data from being mistaken for a complete correspondence. It does not imply that global Langlands reciprocity predicts either the circuit threshold or the quantum-Latin result.

Scope boundary

The word "Langlands" in this monograph refers to an operational packet architecture: commuting local spectra, type refinement, noncommutative holonomy, tensor assembly, and descent obligations. No general automorphic or Galois correspondence is claimed or used.

11.1.13 A general finite threshold system

A packet-resolved threshold system consists of:

1. a finite group, groupoid, or association scheme;
2. a commutative spherical algebra with exact projectors;
3. embedded incidence or type data;
4. a transition group or monodromy representation;
5. restricted and ambient realization categories;
6. a metric, multiplicity, or descent obstruction.

The first threshold may then be defined as

$$\tau = \min\{n : \exists P \text{ with } c_{\text{res}}(P) > c_{\text{amb}}(P)\}.$$

The cost can be finite or infinite.

11.1.14 Open problems

Question 11.3. For which finite association schemes does the full embedded Bose–Mesner algebra determine the restricted realization fiber?

Question 11.4. What bounded degree of noncommutative transition moments separates all Latin squares of a fixed order?

Question 11.5. Can typed Green kernels yield new lower bounds without enumerating the full ambient group?

Question 11.6. When does imprimitivity of a transition packet characterize a tensor or subsquare decomposition in quantum combinatorial designs?

CHAPTER 12

The computational atlas and open directions

12.1 The exact computational atlas

12.1.1 Purpose and design

The accompanying code is an audit-oriented finite mathematics package rather than a production compiler. Its goals are:

1. exact state representations with no floating-point ambiguity;
2. complete finite search where the state space is tractable;
3. explicit witnesses and reconstructible paths;
4. small, dependency-free core algorithms;
5. frozen JSON results and regression tests;
6. separation between proved, computed, and proposed claims.

The package is named `catalytic_gate_atlas`. Binary matrices are tuples of integer row bitmasks, and reversible Boolean maps are exact permutations of basis-state indices.

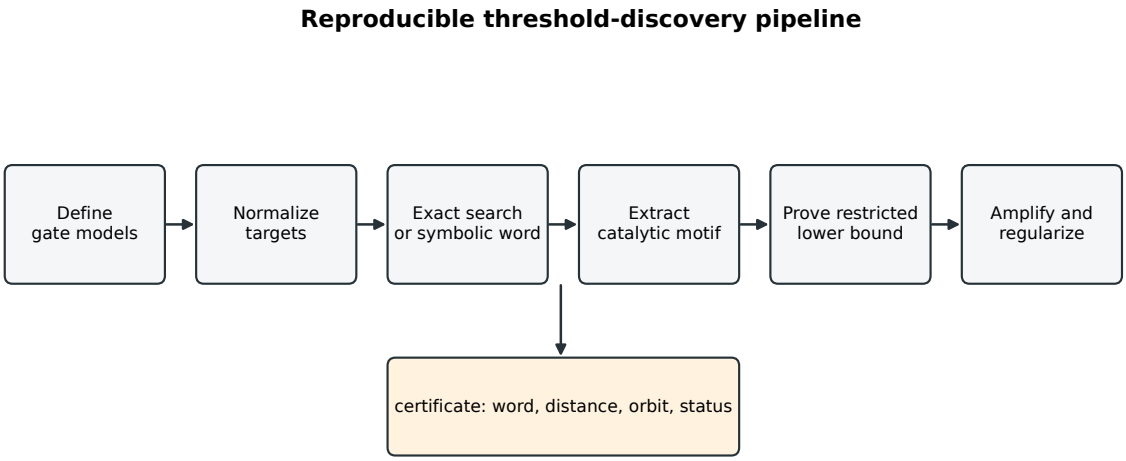


Figure 12.1. The finite certificate pipeline used throughout the computational atlas.

12.1.2 State models

Binary matrices

An $m \times m$ matrix over $\mathbb{F}_2$ is stored as

$$(r_0, \ldots, r_{m-1}),$$

where bit j of r_i is the (i, j) entry. Matrix multiplication is exact XOR accumulation.

Symplectic Clifford states

An n-qubit Clifford modulo global phase and Pauli translation is represented by its action on Pauli exponent vectors

$$(x \mid z) \in \mathbb{F}_2^{2n}.$$

The state space is $\mathrm{Sp}(2n, 2)$. The restricted parabolic group is generated by S, CZ, and CNOT; the ambient group additionally permits H.

Classical reversible states

A reversible map on n bits is stored as a permutation of $\{0, \ldots, 2^n - 1\}$. Composition is exact. The affine subgroup is generated by X and CNOT, and the ambient three-bit group adds Toffoli.

12.1.3 Shortest-path engines

The package implements:

- breadth-first search for unit costs;
- 0–1 breadth-first search when gates are free or unit cost;
- Dijkstra search for general nonnegative integer weights;
- parent maps for witness reconstruction;
- defect distributions on a restricted target set.

Algorithm 12.1 (Breadth-first defect census)**.** Given restricted generators S_R, ambient generators S_A, and identity e:

1. compute d_R on $\langle S_R \rangle$;
2. compute d_A on $\langle S_A \rangle$;
3. for every restricted state x, compare $d_R(x)$ and $d_A(x)$;
4. record the pair $(d_R(x), d_A(x))$ and reconstruct witnesses for strict inequalities.

For a finite group this is complete because every vertex reachable from the identity is visited exactly once at its minimum distance.

12.1.4 Zero-cost gates

When local Clifford gates are free and entangling gates cost one, ordinary BFS is invalid. The package uses a deque-based 0–1 BFS. A zero-cost edge is pushed to the front and a unit-cost edge to the back. The resulting labels are exact shortest entangling costs.

A common error is to quotient by free gates without proving that the quotient preserves the target subset and all left/right actions. The atlas avoids this issue by running 0–1 BFS directly on the finite group for the reported small cases.

12.1.5 The optimized Clifford verifier

The three-qubit symplectic group has

$$|\mathrm{Sp}(6,2)| = 1{,}451{,}520$$

states. A compact C++ implementation enumerates the full group and the parabolic subgroup exactly. It supports:

- total primitive count;
- total count with unit-cost SWAP;
- entangling count with H and S free;
- exact distance distributions and JSON output.

The C++ program and the independent Python implementation use the same gate matrices but different traversal code. Witness reconstruction is supplied by the Python implementation, not by the C++ census executable.

12.1.6 Computed Clifford census

n	Metric	SWAP	$\lvert P_n\rvert$	$\lvert\mathrm{Sp}(2n,2)\rvert$	Defects	Max defect
1	total	no	2	6	0	0
1	entangling	no	2	6	0	0
2	total	no	48	720	0	0
2	entangling	no	48	720	4	1
3	total	no	10,752	1,451,520	24	1
3	total	yes	10,752	1,451,520	0	0
3	entangling	no	10,752	1,451,520	1,624	2

The detailed defect distributions are:

model	restricted $\to$ ambient pairs
$n = 2$ entangling	$3 \to 2 : 4$
$n = 3$ total	$7 \to 6 : 6,\quad 8 \to 7 : 18$
$n = 3$ entangling	$3 \to 2 : 24,\ 4 \to 3 : 432,\ 5 \to 4 : 1056,\ 6 \to 4 : 48,\ 6 \to 5 : 64.$

Exact finite computation

Every entry in the table is a complete finite enumeration in the declared binary symplectic model. Phases and Pauli translations are not part of the state space. These are exact results for that quotient model, not full hardware Clifford optimizations.

12.1.7 Computed affine-versus-Toffoli census

For three bits:

$$|\mathrm{AGL}(3,2)| = 1344, \qquad |S_8| = 40320.$$

The complete census finds no affine target with a strict defect under total primitive count or under entangling count with NOT free.

Metric	Restricted size	Ambient size	Restricted diameter	Defects
Total	1,344	40,320	7	0
Entangling	1,344	40,320	6	0

12.1.8 Linear symbolic checks

The Python module verifies that the displayed six-row ambient and restricted words produce the same canonical target modulo

$$2, 3, 5, 8.$$

The lower bounds and ring universality are proved in the core chapter; these modular checks are regression tests rather than the logical basis of the theorem.

12.1.9 Witness format

A witness record contains:

- the target matrix or permutation;
- restricted and ambient costs;
- a named restricted gate path;
- a named ambient gate path;
- the metric and gate alphabet;
- the state-model convention.

For example, the two-qubit entangling Clifford witness has restricted path

$$\mathrm{CX}_{0\to1},\ S_1,\ \mathrm{CX}_{1\to0},\ \mathrm{CX}_{0\to1}$$

and ambient path

$$H_1,\ \mathrm{CX}_{1\to0},\ \mathrm{CX}_{0\to1},\ H_0,\ S_1,\ S_0.$$

Only two gates in the ambient path are charged by the entangling metric.

12.1.10 Certificates beyond full enumeration

For larger systems, the following certificate styles are supported in principle.

1. **Meet in the middle.** Store forward and backward balls whose radii sum to the candidate length.
2. **SAT/SMT.** Encode gate choices and endpoint equations; use an UNSAT certificate below the claimed optimum.
3. **Integer programming.** Useful when a fixed gate skeleton leaves discrete routing choices.
4. **Algebraic elimination.** Eliminate continuous gate parameters from bounded-length word equations.
5. **Dual potentials.** Exhibit a one-Lipschitz potential attaining the claimed lower bound.
6. **Quotient search.** Search cosets only after proving that the quotient is compatible with the cost and target class.

12.1.11 Multiobjective search

Let each gate have a vector cost in $\mathbb{N}^r$. At each state, retain only nondominated labels. The algorithm terminates for finite state spaces with a bounded search region, but the frontier can grow exponentially.

A practical exact strategy is to scalarize with several positive weight vectors, use the resulting optima to identify candidate Pareto points, and then certify each candidate by constrained search.

12.1.12 Reproducibility commands

From the `code` directory:

Listing 12.1. Python atlas commands.

```
python -m pip install -e . --no-build-isolation
python scripts/run_atlas.py --output ../results/atlas_small.json
python -m pytest tests
```

The optimized three-qubit verifier can be compiled by

Listing 12.2. C++ Clifford verifier.

```
g++ -O3 -std=c++20 scripts/clifford_census.cpp -o scripts/clifford_census
./scripts/clifford_census --n 3 --metric total --output ../results/clifford_n3_total.json
./scripts/clifford_census --n 3 --metric entangling --output ../results/clifford_n3_entangling.json
```

12.1.13 Validation policy

The release uses four layers of validation.

1. Unit tests for matrix, permutation, and path primitives.
2. Cross-checks of group sizes against known finite formulas.
3. Independent Python and C++ computations for selected Clifford cases.
4. Frozen JSON outputs and SHA-256 manifests in the release archive.

12.1.14 Scaling limits

Full group enumeration grows rapidly:

$$|\operatorname{Sp}(2n,2)| = 2^{n^2}\prod_{j=1}^{n}(4^j-1).$$

The four-qubit symplectic group is far beyond direct Python BFS. Future work should exploit:

- double-coset normal forms,
- Bruhat or parabolic decompositions,
- admissible heuristics for A* search,
- SAT encodings,
- orbit reduction,
- tensor-network representations,
- distributed meet in the middle.

12.1.15 What the atlas does not prove

A finite census establishes only the declared finite model. It does not prove:

- an asymptotic theorem without a separate proof,
- a hardware advantage under a different cost,
- robustness to approximation,
- equivalence after adding ancillas,
- a result for global phases or Pauli translations if they were quotiented out.

The status metadata in every result file is intended to make these boundaries machine readable.

12.2 Reusable theorem templates

12.2.1 Purpose

This chapter collects proof architectures that recur across gate logics. Each template separates the easy upper-bound identity from the difficult lower-bound obligation. The templates are not claims that every gate hierarchy satisfies the hypotheses.

12.2.2 Template I: catalytic conjugation

Let $H \le G$ have restricted and ambient costs with $\ell_{\mathrm{amb}}(g^{-1}) = \ell_{\mathrm{amb}}(g)$. Choose $g \in G$ and $h_1, \dots, h_m \in H$ such that

$$U = g h_m \cdots h_1 g^{-1} \in H.$$

Then

$$\ell_{\mathrm{amb}}(U) \le 2\ell_{\mathrm{amb}}(g) + \sum_i \ell_{\mathrm{amb}}(h_i).$$

Theorem 12.2 (Catalytic comparison template)**.** *Assume a restricted lower bound of the form*

$$\ell_{\mathrm{res}}(U) \ge \sum_i r_i - \beta,$$

where r_i is the restricted cost of the ith transported obligation and β is the maximum certified sharing. Then

$$\Delta(U) \ge \sum_i r_i - \beta - 2\ell_{\mathrm{amb}}(g) - \sum_i \ell_{\mathrm{amb}}(h_i).$$

The proof is immediate. The research content lies in bounding β.

12.2.3 Template II: amortized consumers

Suppose a catalyst costs $c_{\mathrm{open}} + c_{\mathrm{close}}$ and each of m consumers saves at least s_i restricted cost. Then

$$\Delta(U) \ge \sum_{i=1}^{m} s_i - c_{\mathrm{open}} - c_{\mathrm{close}} - r,$$

where r is repair or interference cost. A threshold occurs when the right side first becomes positive.

This template explains the three-cross rule in the linear atom and guides searches in phase, Gaussian, and routing models.

12.2.4 Template III: first-arrival Green valuation

Let S be a finite symmetric generating set and

$$A_S = \sum_{s \in S} \rho(s).$$

Then

$$\mathcal{G}_{S,g}(z) = \sum_{k \geq 0} \langle \delta_g, A_S^k \delta_e \rangle z^k.$$

Theorem 12.3 (First-arrival template)**.** *If word multiplicities are nonnegative, then*

$$d_S(e, g) = \operatorname{ord}_{z=0} \mathcal{G}_{S,g}(z).$$

For restricted and ambient sets,

$$\Delta(g) = \operatorname{ord}_0 \mathcal{G}_{S_R,g} - \operatorname{ord}_0 \mathcal{G}_{S_A,g}.$$

This template provides an exact spectral restatement. To turn it into a lower bound without enumeration, one needs control of the endpoint projector coefficients.

12.2.5 Template IV: one-Lipschitz potentials

Let F be a function on circuit states satisfying

$$F(I) = 0, \qquad |F(gs) - F(g)| \leq c(s)$$

for every allowed restricted gate s. Then

$$\ell_{\text{res}}(U) \geq F(U).$$

Potential design steps:

1. choose a decomposition into cuts, packets, or degree layers;
2. prove a gate changes the total charge by at most its cost;
3. ensure cancellations cannot create free progress on many layers;
4. evaluate the target exactly.

Static sums of ranks are often too coarse in ambient models; time-expanded or target-relative potentials may be required.

12.2.6 Template V: dimension per gate

Let X be an algebraic family of elementary gates whose irreducible components have dimension at most r. Let V be an irreducible target variety of dimension N.

Theorem 12.4 (Dimension-per-gate template)**.** *The closure of the locus reachable in L gates has dimension at most rL. If $rL < N$, a generic point of V is not reachable in L gates.*

If every target in V has a restricted construction of length at most N/r, then generic restricted and ambient lengths coincide at that value, provided the same parameter bound applies to the ambient alphabet.

12.2.7 Template VI: defect codimension

Assume every target has restricted cost at most M. A defect of size at least d implies ambient cost at most $M - d$. If the $(M - d)$-gate ambient locus has dimension at most $N - d$, then

$$\operatorname{codim} \overline{\mathcal{D}_{\geq d}} \geq d.$$

This is the general mechanism behind algebraic rarity.

12.2.8 Template VII: finite-field sparsity

Suppose there are at most B gate directions and at most q^r parameter choices per gate over $\mathbb{F}_q$. Then the number of words of length at most L is at most

$$\sum_{j=0}^{L} (Bq^r)^j.$$

If the target family has q^N points and defect d forces $L \leq (N - d)/r$, this yields a density bound of order q^{-d} up to a fixed combinatorial factor.

The estimate counts words, not distinct endpoints, and is therefore an upper bound without injectivity assumptions.

12.2.9 Template VIII: block retraction

Let a large restricted group contain a smaller one through a homomorphic embedding ι, and suppose a homomorphic retraction π satisfies $\pi\iota = \mathrm{id}$. Require that every large restricted generator maps to a small restricted word of no greater cost. Then

$$\ell_{\text{large,res}}(\iota U) \geq \ell_{\text{small,res}}(U).$$

If the ambient embedding preserves gate cost,

$$\Delta_{\text{large}}(\iota U) \geq \Delta_{\text{small}}(U).$$

This transfers a catalytic atom to architectures containing it as a retract.

12.2.10 Template IX: exact block coupling

For block-lower matrices

$$M = \begin{pmatrix} A & 0 \\ XA & D \end{pmatrix},$$

projection to diagonal blocks forces the costs of A and D, while rank of the cross block forces at least rank X cross transmissions. If X is a partial monomial matrix, the lower bound is attained by direct construction.

The same logic extends to trees of blocks whenever every coupling layer has an exact rank certificate and the projections are homomorphisms.

12.2.11 Template X: tensor Leibniz rule

Suppose a restricted complexity is tree tight and satisfies

$$\ell_{\mathrm{res}}(A \otimes B) = n\ell_{\mathrm{res}}(A) + m\ell_{\mathrm{res}}(B)$$

for A of size m and B of size n. If ambient complexity satisfies the corresponding upper bound, then

$$\Delta(A \otimes B) \geq n\Delta(A) + m\Delta(B).$$

For tensor powers,

$$\Delta(A^{\otimes r}) \geq rm^{r-1}\Delta(A).$$

This is the standard route from a finite atom to an $N \log N$ defect.

12.2.12 Template XI: stable capacity

If

$$a_{r+s} \leq m^s a_r + m^r a_s,$$

then

$$b_r = \frac{a_r}{m^{r-1}}$$

is subadditive and

$$\lim_{r\to\infty} \frac{a_r}{rm^{r-1}}$$

exists by Fekete's lemma. This defines a regularized ambient cost and exposes possible tensor activation.

12.2.13 Template XII: vector costs

Let gates carry costs in $\mathbb{R}^k_+$. A lower bound may be a vector inequality

$$F(U) \preceq \sum_{g\in W} c(g)$$

or a family of scalar inequalities indexed by dual weights $w \in \mathbb{R}^k_+$:

$$w \cdot F(U) \leq w \cdot c(W).$$

A complete Pareto proof may require enough supporting weights to separate the restricted and ambient convex lower hulls.

12.2.14 Template XIII: ancilla restoration

Let W act on data D and ancilla A. To certify a dirty catalyst, prove the operator identity

$$W = U_D \otimes I_A$$

on the full joint space, not only on one ancilla input. For a clean catalyst, prove

$$W(|\psi\rangle \otimes |0\rangle) = (U|\psi\rangle) \otimes |0\rangle$$

for all $|\psi\rangle$. The distinction changes both the target equations and lower bounds.

12.2.15 Template XIV: symmetry-sector escape

Given projectors P_q, define off-sector transfer blocks

$$T_{qr} = P_q U P_r.$$

A restricted gate preserves all diagonal sectors. An ambient gate changes a controlled collection of blocks. A candidate potential is a weighted sum of ranks or norms of obligations relative to the final target. The proof must charge both creation and useful consumption of off-sector coherence.

12.2.16 Template XV: exhaustive threshold classification

A finite first-threshold theorem should contain:

1. no defects below size n_0;
2. at least one defect at n_0;
3. exact classification of every defect at n_0 modulo declared equivalence;
4. analytic generation of the exceptional motifs;
5. equality between the analytic set and computational census;
6. witness and completeness certificates.

This is the standard achieved by the six-wire linear core and is the target standard for future gate logics.

12.2.17 A proof-obligation ledger

Before promoting a catalytic identity to a theorem, record:

Obligation	Required evidence
Endpoint membership	Symbolic proof that the target lies in the restricted class.
Ambient upper bound	Explicit circuit evaluated exactly.
Restricted lower bound	Potential, rank, algebraic, search, or certificate argument.
Restoration	Identity on all data and work inputs under the chosen convention.
Cost compatibility	Same gate alphabet, weights, and free operations in both metrics.
Minimal threshold	Exhaustive or symbolic exclusion of smaller instances.
Equivalence classification	Orbit or normal-form analysis of all minimal defects.
Amplification	Proof that the lower bound and upper construction compose.
Reproducibility	Source, frozen outputs, and deterministic commands.

12.3 Research program and paper architecture

12.3.1 The central research problem

For a nested pair of gate models, determine the distortion function

$$D_n = \sup_{U \in \mathcal{T}_n} \big(c_{\mathrm{res}}(U) - c_{\mathrm{amb}}(U) \big),$$

where $\mathcal{T}_n$ is the target class. The basic questions are:

1. Does any defect exist?
2. What is the first size threshold?
3. What is the minimal catalytic motif?
4. Is the effect generic, sparse, or confined to special strata?
5. How does it scale under direct sum, recursion, and tensor products?
6. Which resource vector reveals the first defect?
7. Can the effect survive realistic architecture and fault-tolerant costs?

12.3.2 Established base camp

The unitriangular theory provides a complete first example:

- a universal six-row atom over nonzero commutative rings;
- exact binary and ternary first-threshold evidence;
- exact classification of twenty binary minimal defects;
- connected families with two reverse gates and linear saving;
- microscopic rank-one, bandwidth-one escape;
- generic rigidity and defect codimension;
- recursive and tensor amplification to $\Omega(n \log n)$.

The small Clifford atlas adds a second gate logic with genuine exact defects and shows that changing the primitive library by adding SWAP can remove them.

12.3.3 Program A: complete the Clifford threshold theory

The highest-priority quantum project is a paper on parabolic-versus-ambient Clifford synthesis.

Immediate tasks

1. Extend the analytic four-target free-phase orbit certificate to other cost-preserving Clifford equivalences and larger widths.
2. Classify the twenty-four three-qubit total-count defects under parabolic conjugation and qubit permutations.
3. Derive analytic normal forms for the witness circuits.
4. Prove lower bounds independent of full group enumeration.
5. Test depth, weighted, and architecture-aware metrics.
6. Determine whether a finite defective atom tensor-amplifies.

Longer-term questions

Conjecture 12.5. The parabolic subgroup of $\mathrm{Sp}(2n, 2)$ has unbounded additive distortion in the full symplectic group for at least one natural Clifford cost model.

Question 12.6. Does the maximum total-count defect grow as n, $n \log n$, or faster?

Question 12.7. Is the disappearance of three-qubit defects after adding unit-cost SWAP a small-width accident or a structural geodesic-convexity phenomenon?

12.3.4 Program B: temporary nonlinearity

The three-bit affine-versus-Toffoli census is rigid. The next target is a complete four-bit normal-form search without enumerating $16!$ permutations.

Algorithmic plan

1. Compute exact affine distances for all $|\mathrm{AGL}(4,2)| = 322{,}560$ targets.
2. Enumerate Toffoli sandwiches modulo affine left and right actions.
3. Use ANF equations to retain only affine endpoints.
4. Compare total, Toffoli, CNOT, and fault-tolerant costs.
5. Produce SAT certificates below every claimed optimum.

Question 12.8. What is the first bit width at which a restored nonlinear excursion shortens an affine target?

Question 12.9. Can temporary degree two yield a family with unbounded saving while the endpoint degree remains one?

12.3.5 Program C: phase-polynomial catalysts

The phase setting is likely to produce a defect first in a resource other than total primitive count.

Priority metrics:

$$T\text{-depth}, \quad \text{CNOT count at fixed } T\text{-count}, \quad \text{two-qubit depth}, \quad \text{architecture-weighted cost}.$$

A first paper should choose one metric and avoid a diffuse comparison across all resources.

Question 12.10. Can a restored Hadamard or Clifford frame reduce T-depth of a diagonal target without additional ancillas?

Question 12.11. Does tensor activation occur for phase-polynomial T-count?

12.3.6 Program D: Gaussian and matchgate systems

Two concrete first searches are recommended.

1. Number-conserving versus general fermionic Gaussian synthesis on a small number of modes.
2. Nearest-neighbor versus all-to-all matchgate circuits with weighted fermionic SWAP cost.

Both admit exact orthogonal-matrix representations and have clear architecture relevance.

12.3.7 Program E: symmetry and measurement

This program should begin with a narrowly declared resource theory, such as:

- excitation-preserving two-qubit gates versus arbitrary two-qubit gates;
- unitary Clifford+T versus measurement-assisted Clifford+T;
- clean versus dirty ancillas for a fixed arithmetic family;
- local architecture versus one reusable nonlocal bus gate.

The first objective is not maximal generality but one theorem with exact restoration and cost accounting.

12.3.8 Program F: root-system catalysis

For a split reductive group with allowed roots $\Psi \subset \Phi$, classify configurations in which a forbidden root acts as a reusable catalyst through Chevalley commutator relations.

Suggested sequence:

$$A_2 \to B_2 = C_2 \to G_2 \to A_r.$$

Low-rank nonsimply laced systems are attractive because longer root strings may create more than one useful effect per consumer.

Question 12.12. Is there a root-system analogue of the three-cross, one-phase threshold?

Question 12.13. Can a G_2 root string produce a multiplicative distortion exceeding the clean type-A value $3/2$?

12.3.9 Program G: algebraic defect loci

For each finite-dimensional gate hierarchy, define bounded word loci and defect strata. Determine:

- irreducible components,
- exact dimension and codimension,
- singularities and degenerations,
- border length,
- finite-field point counts,
- behavior under tensor products.

Conjecture 12.14. In a broad class of one-parameter elementary gate varieties, defect at least d is generically a codimension-at-least-d condition, with equality on explicit families of components.

12.3.10 Program H: packet-resolved realization

The packet hierarchy should be developed as a separate mathematical framework rather than appended to every circuit paper.

Core objectives:

1. define spherical, type, and holonomy functors for finite transition groupoids;
2. prove information-loss theorems for commutativization;
3. classify when typed Green kernels are sufficient for distance;
4. relate imprimitivity to combinatorial decomposition;
5. formalize realization fibers and descent obstructions.

12.3.11 Recommended paper sequence

Paper	Working title	Core result
I	Catalytic escape from unitriangularity	Completed linear threshold, rarity, and asymptotic distortion.
II	Catalytic basis escape in Clifford synthesis	Exact parabolic-versus-Clifford thresholds and analytic motifs.
III	Temporary nonlinearity in reversible circuits	Affine targets with or without Toffoli-assisted defects.

IV	Gate-set distortion and catalytic holonomy	Abstract subgroup, groupoid, and vector-cost theory.
V	Spherical collapse and transition holonomy	Packet refinement, Latin-square holonomy, and realization fibers.
VI	Algebraic defect loci	Word varieties, codimension, border catalysis, and point counts.
VII	Root-system catalytic complexity	Chevalley commutators and thresholds beyond type A.

12.3.12 Software roadmap

Version 0.1 of the atlas supports exact small finite groups. A mature package should add:

1. generic multiobjective Dijkstra search;
2. disk-backed and parallel BFS;
3. SAT and SMT export;
4. group action and orbit classification;
5. symbolic word-map elimination;
6. circuit diagrams in Quantikz and SVG;
7. architecture graphs and routing costs;
8. certificate verification independent of the search engine;
9. a machine-readable theorem-status ledger.

12.3.13 Standards for claims

A result should be labeled *proved* only when every lower bound and restoration claim has a proof. A complete finite census should identify the state quotient, enumerate its full size, and make the traversal reproducible. A symbolic skeleton should be valued as a research object but not described as an established shortcut.

12.3.14 Potential applications

If developed, the theory may contribute to:

- exact and heuristic quantum circuit compilers,
- architecture-aware routing,
- parity and phase-network synthesis,
- stabilizer and Clifford optimization,
- reversible arithmetic,
- Gaussian and fermionic simulation,
- lower-bound methodology,
- geometric group theory of gate sets,
- algebraic geometry of circuit word maps.

The near-term value is methodological and algorithmic. Hardware benefit requires a device-specific cost and noise model.

12.3.15 Twenty central open problems

1. Prove or refute a universal six-dimensional linear threshold over all coefficient rings.
2. Determine the true asymptotic order of the maximum unitriangular defect.
3. Decide whether multiplicative distortion is bounded.
4. Determine the lower border length of the canonical six-row atom.
5. Classify tree-tight catalytic matroids.
6. Find a one-copy rigid target with tensor activation.
7. Extend the two-qubit analytic orbit certificate to larger Clifford systems.
8. Analytically classify the three-qubit total-count Clifford defects.
9. Determine whether SWAP restores rigidity for all n in the chosen primitive model.
10. Find the first affine target shortened by temporary nonlinearity, or prove none exists.
11. Find the first diagonal target improved by non-diagonal Clifford catalysis.
12. Determine a phase-polynomial Pareto defect.
13. Find a number-conserving Gaussian target shortened by Bogoliubov escape.
14. Find a matchgate routing catalyst with a certified architecture advantage.
15. Establish a symmetry-sector microscopic-escape theorem.
16. Classify when a fixed number of dirty ancillas gives unbounded saving.
17. Develop typed Green-kernel lower bounds that avoid full enumeration.
18. Determine finite moment bounds sufficient for transition-holonomy classification.
19. Classify catalytic motifs in B_2, C_2, and G_2.
20. Prove a general theorem connecting algebraic codimension and catalytic defect.

12.3.16 Closing perspective

The common phenomenon is not that every restriction can be beaten. It is that the path space matters. A target may obey a symmetry, flag, degree bound, locality rule, or factorization constraint while the cheapest route to that target does not.

The disciplined research question is therefore:

> Which temporary resources can be opened, reused, and exactly removed at a net computational gain?

The established linear theory answers this question completely for one important model. The monograph supplies the symbolic and computational machinery to ask it across quantum gate logic without conflating identities, experiments, and theorems.

PART IV

Circuit obstructions

Restored reversible circuits lead to cut comparisons, safe discrepancy covers, and projection costs. The later chapters examine faithful linear summaries and the obstruction to composing multiplicative descriptions.

IN THIS PART

Scope, notation, and computational model 219

Exact cut comparison and safe discrepancies 222

Complete safe-packet degree 223

Calibration families and exact small oracles 227

Homogeneous projection: cost and safety 230

Faithful linear gluing algebras 233

Weighted witness fingerprints 235

The overlap obstruction to composition 237

Circuit examples in view 238

CHAPTER 13

Scope, notation, and computational model

13.1 Boolean and algebraic conventions

A Boolean function is a map $f : \{0,1\}^n \to \{0,1\}$. Unless a different field is explicitly stated, polynomials are the unique multilinear representatives in

$$\mathcal{R}_n = \mathbb{F}_2[x_1, \ldots, x_n]/(x_i^2 - x_i : 1 \le i \le n).$$

Addition in this ring is XOR. Multiplication is AND. Boolean OR is written $\vee$ and satisfies $a \vee b = a + b + ab$. A statement $h \le f$ is pointwise and means that h has no positive value at a negative input of f.

The degree of a nonzero function is its multilinear algebraic degree. A nonzero constant has degree zero. For mathematical statements involving a constant-zero function we use $\deg 0 = -\infty$; stored profile tables and Python helpers use their documented sentinels instead. Those implementation conventions do not change the inequalities.

For $S \subseteq [n]$, write $x^S = \prod_{i \in S} x_i$ and let 1_S be the input with support S. A minterm of a monotone function means an inclusion-minimal positive support, not an arbitrary positive input.

Symbol	Meaning
$C(f)$	Minimum fan-in-two AND/OR/NOT gate count, allowing fan-out.
$L(f)$	Minimum total NOT/CNOT/Toffoli count of a restored implementation.
T	Number of Toffoli gates in a specified implementation.
$T_{\text{rest}}(f)$	Minimum such Toffoli count, allowing any finite zero workspace.
$M(f)$	Minimum AND count in an XOR–AND computation, with affine operations uncharged.
I_f	Boolean functions supported only on positive inputs of f.
$\alpha_1(f)$	Minimum degree of one nonzero member of I_f.
$\beta_v(f)$	Minimum degree of a member of I_f that is one at v.
$\beta(f), \beta^*(f)$	Complete-cover degree and its affine-output optimization.
$[P]_2$	Homogeneous degree-two component of the multilinear polynomial P in the specified coordinates.

13.2 The restored classical oracle

A legal NOT changes one target bit. A CNOT adds a control bit to a distinct target. A Toffoli adds the product of two distinct controls to a third distinct wire. Gates have positive controls and are self-inverse. Their composition is a permutation of all register states.

A restored oracle is a circuit satisfying

$$(x, 0^a, y) \longmapsto (x, 0^a, y + f(x)) \tag{13.1}$$

for every logical input x and both initial target values $y \in \{0, 1\}$. The zero-workspace condition is imposed at the boundary, not at intermediate times. The circuit may disturb logical inputs, use output feedback, reuse intermediate values, and make arbitrary reversible excursions before restoring the required registers.

NOT and CNOT are free only when counting T. They are charged in L. This distinction is essential in the projection example. There are no measurements, quantum phases, discarded work registers, or unverified ancilla initializations in this model.

13.3 Comparison with ordinary circuits

Theorem 13.1 (Classical/restored simulation comparison)**.** *For every Boolean function f,*

$$\frac{C(f)}{5} \le L(f) \le 6C(f) + 1.$$

Proof. Compute every Boolean gate onto a fresh zero wire. An AND costs one Toffoli. A NOT of an existing signal is obtained with one NOT and one CNOT. An OR is computed as $a + b + ab$, using two CNOTs and one Toffoli. Repeated operands are simplified to a copy, not encoded as an illegal repeated-control gate. Compute the output, copy it to y, and reverse the first computation. This uses at most $6C(f) + 1$ gates and restores every work register.

Conversely, express each reversible gate as an ordinary Boolean update. XOR has a four-gate AND/OR/NOT realization. A Toffoli needs that XOR plus one AND, hence at most five gates. Substitute zero for initial ancillas and $y = 0$, then read the final target. The resulting Boolean circuit has at most $5L(f)$ gates. □

13.4 The degree bound used throughout

Lemma 13.2 (Degree per multiplication)**.** *An XOR–AND computation with t AND gates has output degree at most $t + 1$, allowing constants and arbitrary sharing. The same bound holds for each coordinate of a reversible computation containing t Toffoli gates.*

Proof. Introduce a formal Boolean variable for each AND output. Its defining expression is a product of two affine forms in the logical inputs and earlier gate variables. The final output is affine in the inputs and all gate variables.

Substitute the gate variables in reverse topological order. A multilinear monomial contains the variable being replaced at most once. Replacing it by a quadratic expression raises degree by at most one; multilinear reduction cannot raise it. After t substitutions the degree is at most $t + 1$.

A Toffoli contributes one multiplication to such a straight-line computation, while NOT and CNOT are affine. The argument also applies to a suffix executed backwards because the gates are self-inverse. □

The source identifies this standard degree bound with the XOR–AND background in [52].

13.5 A degree invariant that retains sharing

The preceding Boolean substitution proof applies to general XOR–AND straight-line computations. The later formal development proves the classical reversible specialization directly on the evolving register state. This route makes the resource accounting visible without unfolding reused registers into duplicated formulas.

Theorem 13.3 (Register-product degree invariant). *Let $s_i(x)$ be the contents of a finite register after t Toffoli gates, with arbitrary intervening NOT and CNOT gates. Initial register contents are affine functions of x. For every finite list L of r register indices, including repeated indices,*

$$\deg\left(\prod_{i\in L} s_i(x)\right) \le t + r.$$

The zero-function convention is as above. In the formal source the conclusion is the concrete representation certificate `DegreeLE`.

Proof. Initially a product of r affine functions has degree at most r. Suppose a gate toggles register u by a product of q controls, with $q = 0, 1, 2$ for NOT, CNOT, and Toffoli respectively. If u does not occur in L, the product is unchanged. If it occurs, Boolean idempotence removes all its repetitions. Let R be the remaining list, so $|R| + 1 \le r$. The updated product is

$$s_u \prod_{i\in R} s_i + \left(\prod_{j\in \text{controls}} s_j\right) \prod_{i\in R} s_i.$$

By the old invariant, these two summands have degrees at most $t+1+|R|$ and $t+q+|R|$. The latter is at most $t+r+\max\{q-1, 0\}$. NOT and CNOT therefore consume no degree budget; a Toffoli consumes at most one. Induction along the actual gate list proves the claim. The empty list is the constant-one product and is also covered. □

For $r = 1$ this recovers the coordinate bound. The initial target can be fixed to either boundary bit while the inputs vary. Temporary input disturbance, dirty intermediate workspace, output controls, and repeated use of earlier products are all allowed. The formal locators are `workspaceDegree_run` and `canonical_run_coordinate_degree`.

> **The representation is part of the theorem**
>
> `DegreeLE(d,f)` is tied by proved equivalences to an explicit finite XOR of monomials with length at most d. It is not an assumed degree oracle. Canonical algebraic-normal-form extraction and a separate generic XOR–AND simulation compiler are outside this formal block.

CHAPTER 14

Exact cut comparison and safe discrepancies

CLAIM C01

Theorem 14.1 (Cut-comparison identity and complete coverage)**.** *Let a restored oracle be written $\mathcal{C} = Q \circ P$, and put $R = Q^{-1}$. Suppress zero ancillas in boundary arguments and define*

$$A(x) = P(x, 0), \qquad B(x) = R(x, 0), \qquad D(x) = P(x, 1) + P(x, 0).$$

These are vectors indexed by all registers. Then

$$A(x) + B(x) = f(x)D(x), \qquad D(x) \neq 0. \tag{14.1}$$

Consequently, the coordinate discrepancies $H_i = A_i + B_i$ satisfy

$$H_i \leq f \quad \text{for every } i, \qquad \bigvee_i H_i = f. \tag{14.2}$$

Proof. For a boundary state with target z, correctness of the full circuit gives

$$R(x, z) = P(x, z + f(x)).$$

When $f(x) = 0$, this says $B(x) = A(x)$. When $f(x) = 1$, it says $B(x) = P(x, 1)$, and the discrepancy is $D(x)$. Both cases are summarized by (14.1).

The prefix P is a permutation. Its images of the two distinct boundary states $(x, 0)$ and $(x, 1)$ are different, so $D(x)$ is a nonzero vector.

If $f(x) = 0$, every discrepancy is zero. If $f(x) = 1$, at least one discrepancy is one. These are precisely the safety and complete-coverage assertions in (14.2). □

14.1 What the comparison does and does not require

The suffix need not reverse the prefix. Output feedback and a target direction distributed among many registers are permitted. Only boundary correctness and reversibility are used. Equivalently,

$$f(x) = 1 \quad \Longleftrightarrow \quad P(x, 0) \neq Q^{-1}(x, 0).$$

This is a necessary representation, not a circuit lower bound from packet count. Expanded monomial counts, individual degrees, and algebra dimensions remain distinct from the shared cost of constructing the complete family. The declared finite cut checks are reported in Section E.4.

CHAPTER 15

Complete safe-packet degree

CLAIM C02

15.1 Pointwise coverage rather than one easy packet

Let

$$I_f = \{h \in \mathcal{R}_n : h(1+f) = 0\}.$$

This is the ideal of functions supported only on the accepting set of f. For nonzero f, the one-packet parameter is

$$\alpha_1(f) = \min\{\deg h : 0 \neq h \in I_f\}.$$

It asks for one safe nonzero polynomial, which may detect only an easy portion of the accepting set. One-sided polynomial immunity is established terminology in this setting; see the background reference [54].

For a positive input v, define instead

$$\beta_v(f) = \min\{\deg h : h \in I_f,\ h(v) = 1\}, \qquad \beta(f) = \max_{v:f(v)=1} \beta_v(f).$$

Set $\beta(0) = 0$. Thus $\beta(f)$ is the degree needed before safe polynomials collectively detect *every* positive input. Finally, put

$$\beta^*(f) = \max_{\ell \in \mathrm{Aff}_n} \beta(f + \ell),$$

where Aff_n is the set of affine Boolean functions on the logical inputs.

Proposition 15.1 (Basic size bounds)**.** *For nonzero f on n inputs,*

$$\alpha_1(f) \leq \beta(f) \leq n.$$

If f has at least two positive inputs, then $\beta(f) \leq n - 1$. In particular $\beta^*(f) \leq n$.

Proof. Any polynomial detecting a specified positive input is nonzero, giving the first inequality. For $v \in \{0,1\}^n$, its point indicator is

$$\delta_v(x) = \prod_{i:v_i=1} x_i \prod_{i:v_i=0} (1 + x_i).$$

If $f(v) = 1$, then $\delta_v \in I_f$ and has degree n, proving the upper bound.

If v and w are distinct positive inputs, $\delta_v + \delta_w$ is safe and equals one at v. The degree-n coefficients of its two summands are both one, so they cancel. Its degree is at most $n - 1$. Do this for each positive v. The last assertion applies the general bound to every affine adjustment, with the stated zero convention. □

Proposition 15.2 (Affine input invariance)**.** *Invertible affine changes of logical input coordinates preserve α_1, β, and β^*, with the pointwise parameters transported by the corresponding bijection of inputs.*

Proof. Composition with an affine map cannot increase multilinear degree. Composition with its affine inverse proves equality of degrees. Such composition bijects the accepting and rejecting inputs, hence maps safe polynomials bijectively to safe polynomials and preserves their pointwise detecting property. It also permutes the affine output masks used in the definition of β^*. □

15.2 The restored lower bound

Theorem 15.3 (Affine-optimized complete-cover bound)**.** *Every restored implementation of f with T Toffoli gates satisfies*

$$T \geq \max\{0,\ \deg f - 1,\ 2\beta^*(f) - 3\}. \tag{15.1}$$

In particular,

$$T \geq \max\{0,\ \deg f - 1,\ 2\beta(f) - 3,\ 2\beta(1+f) - 3\}.$$

Proof. Cut the implementation after $\lfloor T/2 \rfloor$ Toffolis. The forward prefix and the inverse suffix use at most $\lceil T/2 \rceil$ Toffolis each. By Lemma 13.2, each of their coordinates, and therefore each discrepancy H_i, has degree at most

$$\left\lceil \frac{T}{2} \right\rceil + 1.$$

Theorem 14.1 gives $H_i \leq f$ and a complete cover of the positive inputs. Every positive v is therefore detected by a safe polynomial within this degree bound, so

$$\beta(f) \leq \left\lceil \frac{T}{2} \right\rceil + 1.$$

Integer rounding gives $T \geq 2\beta(f) - 3$.

At the end of a restored oracle the logical inputs are available unchanged. CNOTs from those inputs and, when needed, a target NOT append any affine ℓ to the target at no Toffoli cost. Apply the same inequality to each $f + \ell$ and maximize. Finally, reading the target with initial value zero computes f using T multiplications, so Lemma 13.2 gives $T \geq \deg f - 1$. □

> **Scope of the lower bound**
>
> The circuit may use output feedback and arbitrary dirty intermediate states, but it must satisfy the restored classical oracle convention. The inequality does not apply automatically to a model with discarded garbage or measurement. Because $\beta^*(f) \leq n$, this parameter alone cannot prove a superpolynomial lower bound in the actual input length.

15.3 Exact evaluation-span certificates

For an integer $d \geq 0$, form the evaluation vector

$$v_d(x) = \left(x^S\right)_{S \subseteq [n],\ |S| \leq d},$$

and write $U = f^{-1}(0)$.

Theorem 15.4 (Dual certificate for failure of safe detection)**.** *For every positive input v,*

$$\beta_v(f) > d \quad \Longleftrightarrow \quad v_d(v) \in \operatorname{span}_{\mathbb{F}_2}\{v_d(u) : u \in U\}. \tag{15.2}$$

Proof. A degree-at-most-d polynomial is a coefficient covector on the evaluation space. It is safe exactly when it annihilates every negative evaluation vector, equivalently their span. Such a covector can equal one on $v_d(v)$ exactly when $v_d(v)$ lies outside that span. This is finite-dimensional linear duality over $\mathbb{F}_2$. □

Corollary 15.5 (Finite support of a dual witness)**.** *When the right side of* (15.2) *holds, there is $W \subseteq U$ with*

$$v_d(v) = \sum_{u \in W} v_d(u), \qquad |W| \le \sum_{j=0}^{\min(d,n)} \binom{n}{j}.$$

The cardinality $|W|$ is odd.

Proof. Choose a basis of the span from negative evaluation vectors and express $v_d(v)$ in that basis. Over $\mathbb{F}_2$, the nonzero coefficients select a subset W of size at most the ambient dimension. The constant-monomial coordinate of every evaluation vector is one; comparing it on both sides shows that $|W|$ is odd. □

Such a relation is replayable by exact linear arithmetic. It does not provide an efficient way to enumerate all negatives of an arbitrary SAT verifier. The theorem assumes that the negative vectors or their proved structure are available.

15.4 Monotone specialization

Theorem 15.6 (Safe monomial ideal for a monotone function)**.** *If f is monotone, I_f has a monomial basis indexed by the positive supports of f. For each positive v,*

$$\beta_v(f) = \min\{|S| : S \subseteq \operatorname{supp}(v),\ f(1_S) = 1\}.$$

Consequently, for nonzero f,

$$\alpha_1(f) = \textit{minimum minterm size}, \qquad \beta(f) = \textit{maximum minterm size}.$$

Proof. For any Boolean polynomial h, its coefficient on x^S is

$$[x^S]h = \sum_{T \subseteq S} h(1_T)$$

over $\mathbb{F}_2$. If S is negative for monotone f, every subset T is negative. A safe h vanishes at each 1_T, so its x^S coefficient is zero. Conversely, a monomial on a positive support is safe by monotonicity. This proves the basis assertion.

If a safe h satisfies $h(v) = 1$, at least one of its nonzero monomials is supported inside $\operatorname{supp}(v)$. That support is positive and has size at most $\deg h$. Conversely, a smallest such positive-support monomial detects v. This proves the pointwise formula. Taking the appropriate minima and maxima gives the minterm statements: a minimal positive support cannot contain a smaller positive one, and every positive input contains a minterm. □

Corollary 15.7 (Exact monomial oracle cost)**.** *For $k \geq 2$,*

$$T_{\text{rest}}(x_1 x_2 \cdots x_k) = 2k - 3.$$

Proof. The unique positive input of the k-variable monomial needs a safe polynomial of degree k. The complete-cover lower bound gives $2k - 3$. For the matching construction, successively compute $x_1 x_2$, then $x_1 x_2 x_3$, and so on up to a $(k-1)$-fold product on clean work wires. Toggle the output using that product and x_k, then uncompute the $k - 2$ work products. The count is $(k-2) + 1 + (k-2) = 2k - 3$. For $k = 2$, a single Toffoli suffices. □

CHAPTER 16

Calibration families and exact small oracles

CLAIM C03 and C04

16.1 An unbalanced disjunction

For $k \geq 2$ and $n = k + 2$, define

$$B = z_1 \cdots z_k, \qquad F_k(a, b, z) = (ab) \vee B = ab + B + abB, \qquad G_k = F_k + a.$$

This is a deliberately easy family with two differently sized accepting regions. It separates the cost of finding one safe packet from the degree needed for complete coverage.

Proposition 16.1 (Raw calibration parameters)**.** *For this family,*

$$\alpha_1(F_k) = 2, \qquad \beta(F_k) = k, \qquad \deg F_k = k + 2.$$

Proof. The minterms have supports $\{a, b\}$ and $\{z_1, \ldots, z_k\}$. The first two assertions follow from Theorem 15.6. The monomial abB has degree $k + 2$ and does not cancel with either lower-degree term in the displayed polynomial. □

Theorem 16.2 (Affine optimization of the one-packet parameter)**.** *For every $k \geq 2$,*

$$\max_{\ell \in \mathrm{Aff}_n} \alpha_1(F_k + \ell) = 3.$$

Proof. First restrict to the face $z_1 = 0$. There $F_k + \ell$ becomes $ab + \ell|_{z_1=0}$, a nonzero quadratic. If the restricted affine term contains a remaining z variable, set $a = 0$ and set the remaining affine form to one. This gives an accepting affine subspace of codimension at most two inside the face. If it contains no remaining z variable, fix a, b to an accepting pair instead. The indicator of the resulting affine subspace in the full input space has degree at most three and is safe. Thus every affine adjustment has a nonzero safe polynomial of degree at most three.

For equality, take $\ell = a$. The degree-three polynomial

$$a(1 + b)(1 + z_1)$$

is safe for G_k. Suppose a safe h of degree at most two existed. On $a = 0$, acceptance requires $B = 1$. For $k \geq 3$, the monotone-ideal argument forces $h|_{a=0} = 0$, hence $h = aL$ with L affine in the remaining variables. The negative face $a = b = 1$ forces $L(1, z) = 0$, so $L = c(1 + b)$. The negative point $a = 1, b = 0, z = (1, \ldots, 1)$ then forces $c = 0$.

When $k = 2$, the restriction can initially be $h|_{a=0} = cB$. Write $h = cB + aL$. On the same negative face $a = b = 1$, the quadratic term cB cannot be cancelled by affine L, so $c = 0$. The previous argument then applies. There is no nonzero safe polynomial of degree at most two, proving equality. □

Theorem 16.3 (Affine optimization of complete coverage)**.** *For every $k \geq 2$,*

$$\beta^*(F_k) = k + 1.$$

Proof. Restrict $G_k = F_k + a$ to $b = 1$. Then

$$G_k|_{b=1} = (1 + a)B.$$

This is the indicator of one point on $k + 1$ variables. At the positive input $a = 0, b = 1, z = (1, \ldots, 1)$, a safe detecting polynomial must restrict to that singleton indicator and therefore have degree at least $k + 1$. Thus $\beta(G_k) \geq k + 1$.

For the upper bound, Theorem 16.2 shows that no $F_k + \ell$ is a singleton indicator: a singleton would have one-packet degree $n = k + 2 > 3$. These functions are also nonzero because F_k is not affine. Each has at least two positive inputs, so Proposition 15.1 gives $\beta(F_k + \ell) \leq n - 1 = k + 1$. □

Corollary 16.4 (Restored cost interval for the calibration family)**.** *For $n = k + 2$,*

$$2n - 5 \leq T_{\text{rest}}(F_k) \leq 2n - 3.$$

Proof. The lower bound follows from Theorems 15.3 and 16.3. For the upper bound, compute $A = ab$ with one multiplication and B using $k-1$ multiplications. Toggle the output by $A+B+AB$, requiring one output Toffoli and affine additions. Uncompute A and B. The total Toffoli count is $k + 1 + k = 2k + 1 = 2n - 3$. □

The preceding combined bound $\max\{0, \deg F_k - 1, 2 \max_\ell \alpha_1(F_k + \ell) - 3\}$ gives $n-1$ for this family; its immunity-only term is the constant 3. The complete-cover bound gives $2n-5$. The stronger estimate can also be recognized by restriction to $b = 1$ followed by a free affine output adjustment. No exact optimum is claimed when the displayed endpoints differ.

Inputs n	$\max_\ell \alpha_1$	β^*	One-packet bound	Complete-cover bound
4	3	3	3	3
5	3	4	4	5
6	3	5	5	7
7	3	6	6	9
8	3	7	7	11
9	3	8	8	13

The formulas are proved for all $k \geq 2$. The source evaluates the designated family truth tables through nine inputs, but checks every affine mask only through eight. The ninth-input affine maximum is justified by the proof, not by a stored exhaustive mask search.

16.2 An explicit moment certificate

On the face $b = 1$, every point except $v = (a = 0, z = (1, \dots, 1))$ is negative for G_k. A polynomial of degree at most k sums to zero over the $(k+1)$-dimensional cube: each monomial omits some coordinate and contributes an even number of ones. Therefore

$$v_k(v) = \sum_{u \neq v \text{ on the face}} v_k(u).$$

This is the certificate from Theorem 15.4, proving that no safe degree-k polynomial detects v.

The finite four-input computation found $\max_\ell \alpha_1(f + \ell) = \beta^*(f)$ for every function in that domain. The family above disproves extrapolation of that equality: it already fails at five inputs, with values three and four, and the difference grows without bound.

16.3 All functions on at most three inputs

Theorem 16.5 (Exact small restored-oracle calibration)**.** *For every Boolean function on one, two, or three logical inputs, the bound in Theorem 15.3 is attained by an explicit restored implementation. The optimum Toffoli counts are zero for affine functions, one for nonaffine quadratics, and three for three-input cubics.*

Proof. An affine function is added to the target using only NOT and CNOT gates, so its optimum is zero.

A nonaffine quadratic on at most three variables has an alternating polar matrix of rank two. In suitable invertible linear coordinates, its quadratic part is a product of two linear forms; the remaining part is affine. A single Toffoli, together with affine preparation and restoration, computes the oracle. The degree bound requires at least one.

A cubic on three variables has the form

$$x_1 x_2 x_3 + q(x) + \ell(x),$$

with q quadratic and ℓ affine. Translating $x_i \mapsto x_i + c_i$ adjusts independently the coefficients of $x_j x_k$, so all quadratic terms can be cancelled. The resulting function is a three-variable monomial plus an affine function. Affine input changes and affine output additions cost no Toffolis. Corollary 15.7 gives both the lower bound three and a matching restored construction. □

There are $4 + 16 + 256 = 276$ functions over the three input sizes. The distribution is

Optimum Toffolis	0	1	3
Number of functions	28	120	128.

The source constructs and checks the actual circuits. Optimality is obtained from the written lower bounds and matching constructions, not from exhaustive enumeration of all reversible circuits.

CHAPTER 17

Homogeneous projection: cost and safety

CLAIM C05

This chapter separates the nonlinear cost of a complete function from the nonlinear cost of one extracted homogeneous component. It also shows that an algebraic projection need not preserve safe support.

17.1 Two supporting lower bounds

Lemma 17.1 (Affine subspace after multiplication restrictions)**.** *If a Boolean function on n inputs is computed with t AND gates in an XOR–AND circuit, there is an affine subspace of dimension at least $n - t$ on which the output is affine.*

Proof. Process the AND gates in topological order while maintaining an affine subspace on which all preceding product outputs are affine. The operands of the next gate are then affine on the current subspace. If an operand is constant, their product is already affine. Otherwise impose one nonempty affine equation setting that operand to zero. This loses at most one dimension and makes the product affine. Earlier affine outputs stay affine under restriction. After at most t restrictions the output is affine on a subspace of dimension at least $n - t$. □

This is the affine-restriction method used in the background literature on multiplicative complexity [53]. The proof is reproduced here in the scalar convention needed below.

Proposition 17.2 (Exact cost of a nondegenerate paired quadratic)**.** *For even n, let*

$$Q(x) = x_1x_2 + x_3x_4 + \cdots + x_{n-1}x_n.$$

Then

$$M(Q) = T_{\text{rest}}(Q) = n/2.$$

Proof. The polar bilinear form of Q is the full-rank alternating form made of $n/2$ disjoint 2×2 blocks. If Q is affine on an affine subspace with direction space U, its second differences vanish on U, so U is totally isotropic. Nondegeneracy gives $\dim U \le n/2$.

Lemma 17.1 therefore implies $n - t \le n/2$, or $t \ge n/2$, for any XOR–AND computation. Every restored Toffoli implementation gives such a computation by fixing its ancillary inputs and initial target, so the same lower bound applies to T_{rest}. Directly add the $n/2$ disjoint pair products to the target with one Toffoli each. This attains both bounds and requires no nonlinear work register. □

17.2 A compact whole function

Let $m \geq 2$, let $a = (1, \ldots, 1) \in \mathbb{F}_2^m$, and introduce one logical bit x_v for each

$$v \in V = \mathbb{F}_2^m \setminus \{0, a\}.$$

There are $n = 2^m - 2$ such bits. Define

$$L_j(x) = \sum_{v : v_j = 1} x_v, \qquad P_m(x) = \prod_{j=1}^{m} L_j(x).$$

Theorem 17.3 (Exact nonlinear cost of the whole function)**.** *For the family above,*

$$M(P_m) = m - 1, \qquad T_{\text{rest}}(P_m) = 2m - 3.$$

Proof. The linear map $x \mapsto (L_1(x), \ldots, L_m(x))$ has rank m, since its columns include the standard basis vectors. Extend these forms to an invertible system of linear coordinates. In those coordinates P_m is the product of the first m variables, with the other $n - m$ variables unused.

Degree is invariant under an invertible affine change, so $\deg P_m = m$ and Lemma 13.2 gives $M(P_m) \geq m - 1$. A product chain attains it.

For the restored lower bound, the safe ideal of the product in these coordinates consists of polynomials divisible by the m-variable monomial. Any nonzero safe polynomial has degree at least m. Theorem 15.3 gives $2m - 3$. Compute the m parities onto clean registers by CNOT, apply the optimal monomial oracle there, and uncompute the parities. No additional Toffolis are needed. □

17.3 Extracting the quadratic grade

Theorem 17.4 (Exact cost of the extracted grade)**.** *Let $Q_m = [P_m]_2$ denote the homogeneous quadratic part in the original x_v coordinates. Then*

$$Q_m = \sum_{\{v, v+a\} \subseteq V} x_v x_{v+a},$$

where every unordered pair is used once, and

$$M(Q_m) = T_{\text{rest}}(Q_m) = n/2 = 2^{m-1} - 1.$$

Proof. The excluded columns give $P_m(0) = P_m(e_v) = 0$. For distinct u, v, Boolean coefficient inversion therefore gives

$$[x_u x_v] P_m = P_m(e_u + e_v) = [u + v = a].$$

The map $v \mapsto v + a$ is a fixed-point-free involution of V, so the displayed terms form $n/2$ disjoint pairs. Proposition 17.2 supplies the exact costs. □

Counterexample 17.5 (Projection does not preserve safe support)**.** *The quadratic extraction $[P_m]_2$ need not be supported on the positive set of P_m.*

Proof. Take $m = 3$, identify the columns with the integers $1, \ldots, 6$ using their three-bit expansions, and set $x_1 = x_3 = x_4 = 1$ with all other inputs zero. Their column sum is $1 + 3 + 4 = 6$ under bitwise XOR, not $a = 7$, so $P_3 = 0$. Among the complementary pairs $(1, 6), (2, 5), (3, 4)$, exactly $(3, 4)$ has both

inputs one. Thus $Q_3 = 1$. In the source ordering this is the logical assignment encoded by integer 13. □

m	Actual inputs n	$T_{\text{rest}}(P_m)$	$T_{\text{rest}}(Q_m)$	Constructed total gates for P_m
2	2	1	1	5
3	6	3	3	21
4	14	5	7	61
6	62	9	31	381
8	254	13	127	2,045
10	1,022	17	511	10,237

These are exact cost formulas. The large-input entries were not obtained by searching for optimal circuits. The full construction uses

$$2m(2^{m-1} - 1)$$

CNOTs as well as $2m - 3$ Toffolis, giving total count $O(n \log n)$. The projected quadratic has $n/2$ total gates.

What the counterexample excludes

The nonlinear projection cost grows exponentially in m, but only linearly in actual input length n. The example does not give an exponential separation in ordinary total gate count. It rules out treating this grade extraction as having overhead bounded only by a polynomial in the original Toffoli count, independently of linear and input-size resources. It also shows that homogeneous extraction may destroy safety. It does not exclude a bound polynomial in $n+T$ or every possible graded invariant.

The source checks all pair coefficients and exact ranks for $2 \leq m \leq 10$. Full truth tables and complete restored evaluations are restricted to $m = 2, 3, 4$. At $m = 4$, the quadratic accepts 7,616 inputs that the whole function rejects; that count is finite evidence, distinct from the explicit universal mechanism above.

CHAPTER 18

Faithful linear gluing algebras

CLAIM C06

This chapter states an exponential dimension result for a specified summary model. The qualifications are part of the theorem's meaning: a lower bound on an algebra's dimension is not automatically a lower bound on the length of a circuit describing one of its elements.

18.1 The witness algebra

Let K be a field and let $W = \{0,1\}^m$. The full witness algebra is

$$\mathcal{B} = K^W,$$

with pointwise addition and multiplication. For each witness v, let δ_v be its indicator function. These form a basis of $\mathcal{B}$ and satisfy

$$\delta_v^2 = \delta_v, \qquad \delta_v \delta_u = 0 \quad (u \neq v), \qquad \sum_{v \in W} \delta_v = 1.$$

In characteristic two, this is equivalently the Boolean quotient algebra

$$K[w_1, \ldots, w_m]/(w_i^2 - w_i : 1 \leq i \leq m).$$

Theorem 18.1 (Dimension of a faithful multiplicative summary)**.** *Let $\Phi : \mathcal{B} \to A$ be a K-linear multiplicative map into a K-algebra. Suppose*

$$\Phi(\delta_v) \neq 0 \qquad \textit{for every } v \in W.$$

Then

$$\dim_K A \geq 2^m.$$

No unital hypothesis on Φ is required.

Proof. Write $e_v = \Phi(\delta_v)$. Multiplicativity makes these nonzero orthogonal idempotents. If

$$\sum_{v \in W} c_v e_v = 0,$$

multiply the equality by e_u. Orthogonality annihilates every term except $c_u e_u$, which must be zero. Since K is a field and $e_u \neq 0$, $c_u = 0$. This holds for every u, so the 2^m vectors e_v are linearly independent. □

Corollary 18.2 (Scalar unital summaries are evaluations)**.** *Every unital K-linear multiplicative map $\Phi : K^W \to K$ is evaluation at exactly one witness.*

Proof. An idempotent scalar in a field is either zero or one. Orthogonality allows at most one $\Phi(\delta_v)$ to be one, and preservation of the unit requires exactly one. If it occurs at v_0, linearity gives

$$\Phi(g) = \Phi\left(\sum_v g(v)\delta_v\right) = g(v_0).$$

□

For $m \geq 1$, such a scalar summary loses at least one singleton witness other than v_0. At $m = 0$, the unique empty witness is preserved by the identity map. The dimension theorem is an elementary product-algebra fact, not a claim of new algebra or a universal computational lower bound.

18.2 The quantified model boundary

The same map Φ here serves *all* predicates on a fixed witness space. It is linear, multiplicative, and retains every primitive witness indicator. Under exactly these assumptions, a polynomial-dimensional summary is impossible as m grows.

An arbitrary algorithm is not required to satisfy these hypotheses. It may use nonlinear encodings, syntax-dependent operations, changing spaces, or succinct descriptions inside an exponentially large algebra. A description of the algebra and a description of one relevant element can be much shorter than a full coordinate listing.

> **No dimension-to-circuit inference**
>
> The theorem excludes a fixed faithful linear multiplicative compression of the full witness algebra into fewer than 2^m dimensions. It does not exclude every polynomial-time decision procedure, does not prove $P \neq NP$, and does not justify treating the algebra's dimension as the circuit cost of an individual decision function.

The finite scalar controls enumerate 256 linear maps from a two-point witness algebra to GF(16), 65,536 maps for four points, and 256 maps from an eight-point algebra to $\mathbb{F}_2$. The respective counts of unital algebra homomorphisms are two, four, and eight, matching the evaluation classification. The all-size result rests on the proof above.

CHAPTER 19

Weighted witness fingerprints

CLAIM C07

19.1 A polynomial that distinguishes witnesses

Let $R : \{0,1\}^m \to \{0,1\}$ be a Boolean verifier, and let K be a characteristic-two field. Define the formal polynomial

$$P_R(z_1, \dots, z_m) = \sum_{w \in \{0,1\}^m} R(w) \prod_{i=1}^{m} z_i^{w_i} \in K[z_1, \dots, z_m].$$

The variables z_i in this expression are ordinary polynomial variables, not reduced by $z_i^2 = z_i$.

Theorem 19.1 (Exact witness fingerprint)**.** *The polynomial P_R is identically zero if and only if R has no accepting witness. Its total degree is at most m.*

Proof. Different Boolean witnesses have different exponent vectors, so their monomials are distinct. Each accepting witness contributes coefficient one to its own monomial and cannot cancel another witness's term. The polynomial is zero exactly when there are no such terms. Each monomial uses at most m variables. □

This avoids the cancellation of an unweighted parity count. It does not, by itself, give a small circuit for evaluating the sum.

19.2 Exact top-coefficient extraction

Lemma 19.2 (Top coefficient as a Boolean-cube sum)**.** *Let $G(w)$ be a multilinear polynomial over a characteristic-two field. Then*

$$[w_1 \cdots w_m]G = \sum_{v \in \{0,1\}^m} G(v).$$

Proof. It is enough to check a monomial. A monomial of proper support is one on an even number $2^{m-|S|}$ of Boolean inputs, so its cube sum is zero in characteristic two. The full-support monomial is one at exactly one input and has cube sum one. Linearity gives the identity. □

Theorem 19.3 (Weighted top-coefficient identity)**.** *Let Top_w extract the coefficient of $w_1 \cdots w_m$ after reduction in the Boolean w algebra. Then*

$$P_R(z) = \mathrm{Top}_w \left[R(w) \prod_{i=1}^{m} (1 + (1 + z_i)w_i) \right]. \tag{19.1}$$

Proof. At $w_i = 0$, the ith factor is one. At $w_i = 1$, it is z_i. Lemma 19.2 therefore converts the right side to exactly the weighted sum defining P_R. □

The expression inside Top_w has a circuit with $O(|R| + m)$ operations when the verifier has circuit size $|R|$. The cost of applying Top_w to that succinct expression is not bounded by writing down the operator. The source evaluator explicitly stores 2^m coefficients and remains exponential in witness length.

19.3 The conditional algorithmic consequence

Lemma 19.4 (Finite-field root bound)**.** *If a nonzero polynomial $P \in K[z_1, \ldots, z_m]$ has total degree at most d, where K is finite, then*

$$\Pr_{z \text{ uniform in } K^m}[P(z) = 0] \le \frac{d}{|K|}.$$

Proof. Induct on the number of variables. For one variable this is the elementary root bound. In the inductive step, write

$$P = \sum_{j=0}^{r} P_j(z_1, \ldots, z_{m-1}) z_m^j, \qquad P_r \neq 0.$$

The leading coefficient has total degree at most $d - r$, so it vanishes at a random first $(m-1)$-tuple with probability at most $(d-r)/|K|$. Otherwise the remaining univariate polynomial is nonzero with at most r roots. The union bound gives $(d - r + r)/|K| = d/|K|$. □

Proposition 19.5 (Conditional randomized satisfiability test)**.** *Suppose the weighted expression in* (19.1) *can be evaluated exactly and uniformly in time polynomial in the verifier description and witness length, over explicitly represented characteristic-two fields of polynomial size. Then choosing $|K| \ge 4m$ gives a one-sided randomized polynomial-time satisfiability test. It never accepts an unsatisfiable verifier and accepts a satisfiable one with probability at least $3/4$ per evaluation.*

Proof. Choose a uniformly random $z \in K^m$, evaluate $P_R(z)$ under the assumed algorithm, and accept when it is nonzero. If no witness accepts, Theorem 19.1 makes the polynomial zero identically, so there is no false acceptance. Otherwise Lemma 19.4 bounds the rejection probability by $m/|K| \le 1/4$. A field of polynomial size uses $O(\log m)$ bits per element, so the random input has $O(m \log m)$ bits. Under the explicit efficient-field and evaluator hypotheses, the procedure has polynomial running time. □

> **Conditional means the evaluator is still missing**
>
> The proposition assumes a uniform efficient evaluator that has not been constructed here. The implemented evaluator is exponential. Even with the assumed evaluator, the stated conclusion is randomized with one-sided error, not a deterministic proof of $P = NP$.

The exact tests use $\mathrm{GF}(16) = \mathbb{F}_2[t]/(t^4 + t + 1)$. They evaluate every specified verifier through three witness bits at every field point, while the independent top-coefficient check uses all field points through two witness bits and the declared smaller coordinate grid at three bits. These domains must not be conflated.

CHAPTER 20

The overlap obstruction to composition

CLAIM C08

The weighted fingerprint is compatible with products on disjoint witness coordinates. The same formula fails when the factors share a coordinate. The distinction isolates a concrete compatibility obligation, rather than asserting that all compression is impossible.

Proposition 20.1 (Disjoint-variable factorization)**.** *If predicates $A(u)$ and $B(v)$ use disjoint witness-variable sets, then their weighted fingerprints satisfy*

$$P_{A\wedge B}(z_u, z_v) = P_A(z_u)P_B(z_v).$$

Proof. Expand the left side as a sum over the Cartesian product of witnesses:

$$\sum_{u,v} A(u)B(v)z_u^u z_v^v = \left(\sum_u A(u)z_u^u\right)\left(\sum_v B(v)z_v^v\right).$$

Disjointness makes the exponent data independent and the factorization exact. □

Counterexample 20.2 (Shared-variable multiplication fails)**.** *For a single witness bit, define $\mathrm{Tr}_z(A) = A(0) + zA(1)$. Then in the Boolean w algebra,*

$$\mathrm{Tr}_z(w) = z, \qquad \mathrm{Tr}_z(1+w) = 1,$$

but

$$\mathrm{Tr}_z\big(w(1+w)\big) = 0 \neq z = \mathrm{Tr}_z(w)\,\mathrm{Tr}_z(1+w)$$

whenever $z \neq 0$. Also,

$$\mathrm{Tr}_z(w^2) = z, \qquad \mathrm{Tr}_z(w)^2 = z^2,$$

which differ for field elements z with $z^2 \neq z$.

Proof. Evaluate w and $1+w$ at zero and one. In the Boolean quotient $w(1+w) = w + w^2 = 0$ and $w^2 = w$. The displayed identities follow directly. In $\mathrm{GF}(16)$, any element outside $\mathbb{F}_2$ supplies the second inequality. □

The input predicates are individually easy and their separate fingerprints are explicit. Their conjunction nevertheless imposes a common-witness requirement that ordinary multiplication of the summaries fails to preserve.

Theorem 18.1 explains a related all-size restriction: a fixed linear multiplicative summary retaining every witness cannot reduce the full witness algebra below dimension 2^m. That statement does not rule out all context-dependent nonlinear evaluation methods. It rules out the displayed context-free multiplicative compiler under its exact hypotheses.

The eight maintained claim groups end here. Their proofs do not assume a superpolynomial shared-generation bound or a polynomial-time weighted-trace evaluator. The appendices identify the exact source coverage, evidence, and closed-scope boundaries.

CHAPTER 21

Circuit examples in view

The model: boundary restoration and Boolean degree

A state has logical input wires x, zero-initialized ancillary wires, and a target bit y. A restored oracle is a legal classical circuit satisfying

$$C(x, 0^a, y) = (x, 0^a, y + f(x)) \qquad \text{for every } x \text{ and both } y.$$

The input and scratch restoration conditions hold at the boundary. Wires may change internally, and the target may control an intermediate gate. Each primitive gate is an involution:

$$y \leftarrow y + 1, \quad y \leftarrow y + x_i, \quad y \leftarrow y + x_i x_j.$$

In a CNOT the control differs from the target. In a Toffoli all three wires are distinct. These are construction rules, not informal assumptions.

All polynomial arithmetic in the Boolean state model is over $\mathbb{F}_2$, with $x_i^2 = x_i$. Every function has a unique multilinear algebraic normal form. Its degree is the largest support size with a nonzero coefficient; the zero function has mathematical degree -1 in the Python helpers. A formal degree bound uses a natural number, so zero has a degree-zero certificate.

Quantity	Meaning
T	Number of Toffoli gates; NOT and CNOT have zero cost in this quantity
L	Total number of NOT, CNOT, and Toffoli gates
M	Number of binary AND operations in an XOR–AND computation
n	Actual number of input bits, including all variables in a displayed family
$\dim_K A$	Dimension of a specified linear summary algebra over K

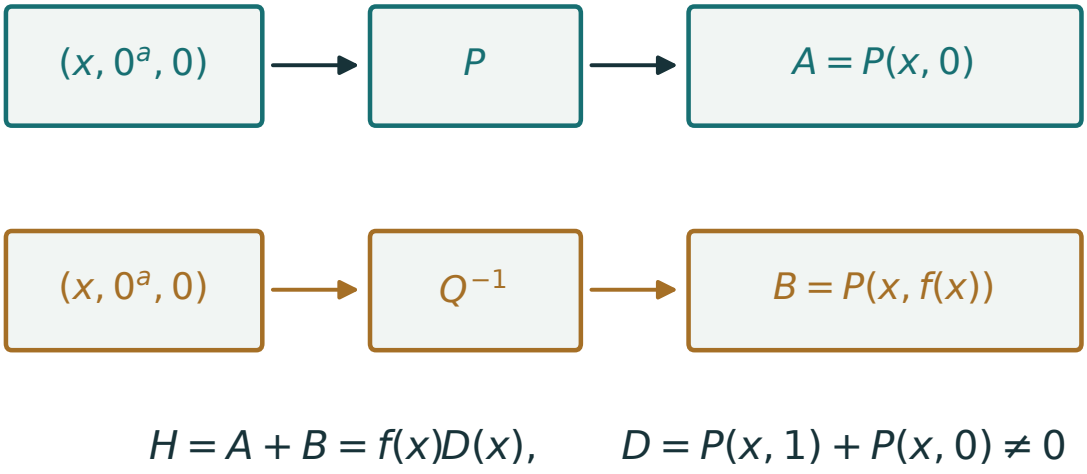
A bound on one row of this table cannot silently be substituted for a bound on another row. In particular, a computation can have few Toffolis and many CNOTs. An exponentially large ambient algebra can have succinct expressions. Those distinctions explain the projection and gluing examples later.

An arbitrary cut compares two reversible computations

Write the circuit as $C = Q \circ P$. The prefix P runs forward; the inverse suffix Q^{-1} starts from the same zero-target boundary state. Define

$$A(x) = P(x, 0), \qquad B(x) = Q^{-1}(x, 0), \qquad D(x) = P(x, 1) + P(x, 0).$$

The zero ancillas are suppressed here. The two executions generally arrive at different interior states, even though they start at matching boundaries.



The exact C01 objects. Both arrows remain inside the legal reversible state space.

Correctness on both target branches gives

$$Q^{-1}(x, b) = P(x, b + f(x)).$$

To derive it, apply C to $(x, b + f(x))$ and then apply Q^{-1} to the result. For $b = 0$, $B = A$ when $f(x) = 0$, while $B = P(x, 1)$ when $f(x) = 1$. Thus

$$H(x) := A(x) + B(x) = f(x)D(x).$$

The two original states differ in their target. Since P is injective, $D(x) \neq 0$. Each coordinate discrepancy is supported on positives, and some coordinate discrepancy detects every positive input.

General written identity; archived formal declarations `cut_transport_identity`, `direction_nonzero`, and `complete_safe_cover`.

Why completeness is pointwise, and why no decoder is needed

For a coordinate i, let $H_i = A_i + B_i$. The cut identity says

$$H_i(x) = f(x)D_i(x), \qquad H_i \le f, \qquad \bigvee_i H_i = f.$$

The last equation is pointwise. For every fixed accepted x, at least one coordinate has $D_i(x) = 1$. The detecting coordinate may depend on x and may be an input, scratch, or target wire. The argument does not require a single fixed coordinate to detect all positives.

At an empty prefix, the direction is simply the target unit vector. At a full prefix, all non-target wires have restored and the target discrepancy is f. Interior cuts can spread the direction through the workspace. These minimum cases verify the composition order and the role of the two target branches.

The zero-input cases are also valid. With only a target wire, the empty circuit realizes the constant-zero oracle and one NOT realizes the constant-one oracle. At every cut, the direction is nonzero and the discrepancy cover has the correct accepting set.

> **Essential hypothesis.** The endpoint condition holds for both initial target values. The inverse-suffix derivation uses the branch $b + f(x)$. Correctness only for initial target zero is not the stated model.

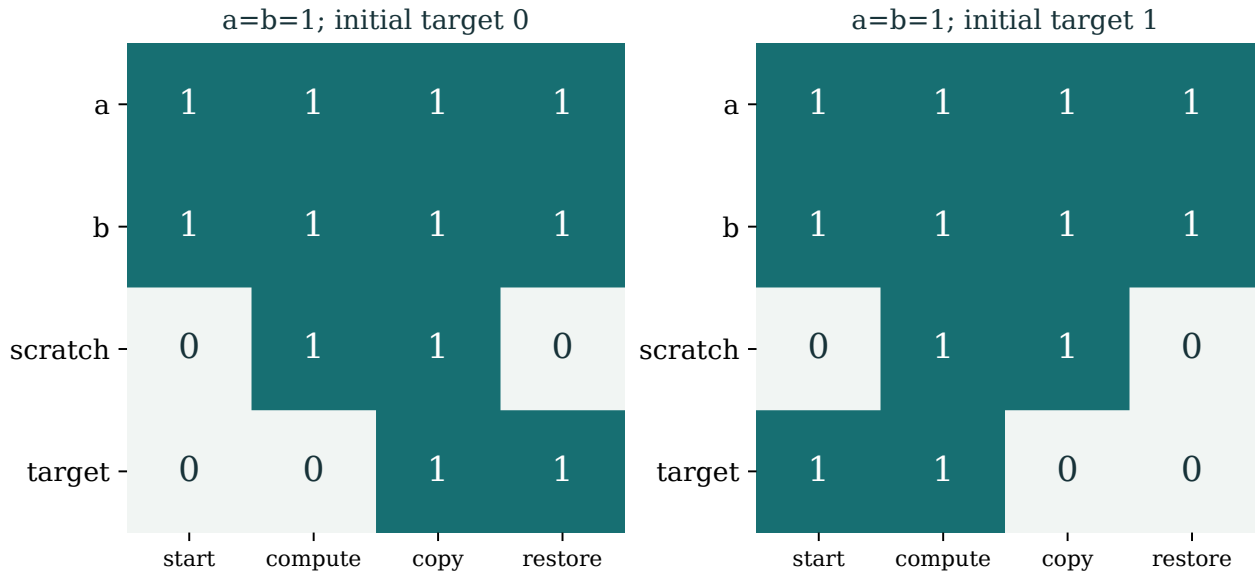
The theorem also does not require scratch to remain zero at the cut. Imposing that extra condition would exclude admitted restored circuits. Conversely, dropping target separation from the boundary encoding permits a vacuous endpoint equation: an encoding that ignores y can make the empty circuit appear to compute a nonzero function. The formal `Boundary.target_eq` property prevents precisely this degeneracy.

A dirty scratch wire can be fully restored

Consider the four-wire oracle in the Lean examples. First compute ab into scratch s, copy s into the target, and uncompute s:

$$(a, b, s, y) \xrightarrow{T(a,b;s)} (a, b, s + ab, y) \xrightarrow{C(s;y)} (a, b, s + ab, y + s + ab) \xrightarrow{T(a,b;s)} (a, b, s, y + s + ab).$$

The displayed final expression specializes to the required oracle when the boundary scratch is zero. On the positive input $a = b = 1$, the scratch is dirty between the first and last gates.



Exact states for the accepting input, in wire order (a, b, s, y). White cells are zero; teal cells are one.

The two panels check both initial targets. The final target is toggled, the logical inputs are unchanged, and scratch returns to zero. A proof that used an intermediate clean-workspace assumption would fail to cover this explicit legal fixture.

This circuit uses two Toffolis. It is a useful restoration witness, not the optimal two-input AND oracle: the direct gate $T(a, b; y)$ computes that oracle with one Toffoli and no scratch. The guide distinguishes an exact example from an optimization claim.

Exact new trace assertion imported from the source gate semantics; the same fixture is formalized by `Examples.andOracle_restores`.

The output direction can move away from the target

Two data/target wires already suffice. A SWAP is implemented by three legal CNOTs. Let P be that SWAP, and let Q be NOT on the data wire followed by another SWAP. Then

$$(x, y) \xrightarrow{P} (y, x) \xrightarrow{Q} (x, 1 + y).$$

The whole circuit is a restored constant-one oracle with zero Toffolis. At the cut after the first SWAP,

$$A = P(x, 0) = (0, x), \qquad B = Q^{-1}(x, 0) = (1, x), \qquad H = (1, 0).$$

The target discrepancy is zero on every input, while the data discrepancy is one on every input. Reading only the target at this cut would reject the constant-one function.

x	A	B	H	named target discrepancy
0	$(0, 0)$	$(1, 0)$	$(1, 0)$	0
1	$(0, 1)$	$(1, 1)$	$(1, 0)$	0

This example uses the target as a control during each SWAP. That behavior is expressly allowed by the model. Its relevance is logical: completeness belongs to the full register comparison, not to an arbitrarily chosen wire. The coordinate packets remain perfectly safe even though the detection direction has moved.

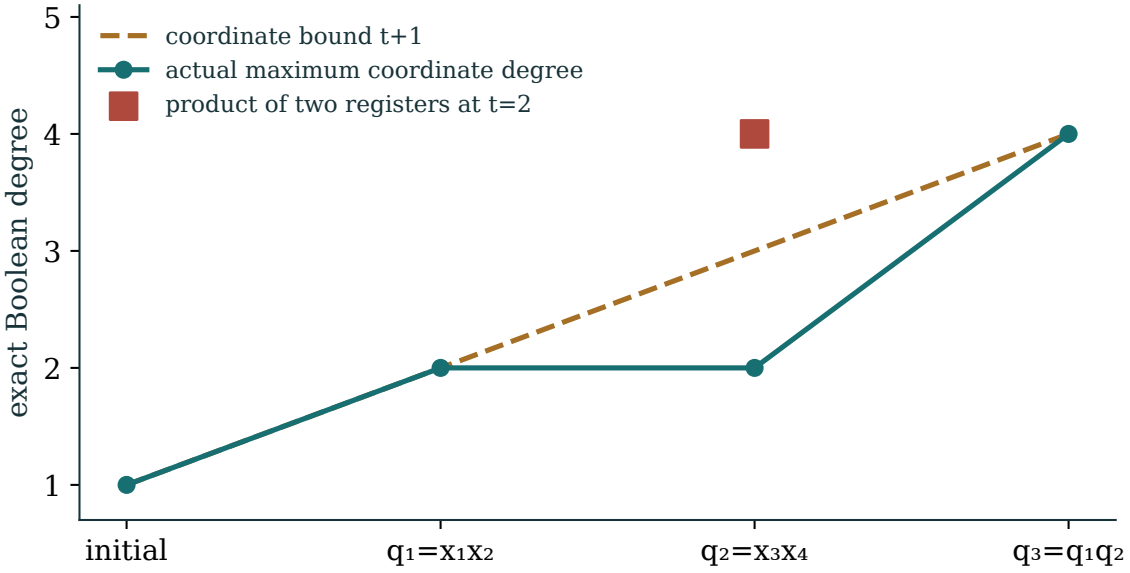
Constructive formal counterexample to target-only reconstruction: `outputActiveOracle_restores`, `moved_discrepancy_data`, `moved_discrepancy_target`.

Degree bounds must retain shared register products

Take four independent input bits. Compute

$$q_1 = x_1x_2, \qquad q_2 = x_3x_4, \qquad q_3 = q_1q_2.$$

After two ANDs, each nonlinear register has degree two, but their product has degree four. After the third AND, a coordinate of degree four appears.



Exact degrees from the complete four-input truth tables. The square marker is a two-register product, not a coordinate.

The correct invariant tracks a product of r current registers after t Toffolis:

$$\deg\left(\prod_{i\in L} s_i(x)\right) \le t + |L| = t + r.$$

At $t = 2$, the product q_1q_2 has $r = 2$ and saturates $t + r = 4$. At $t = 3$, the resulting coordinate has $r = 1$ and saturates $t + 1 = 4$.

This explains why tracking only individual register degrees and then adding their separate histories can lose the shared accounting. The invariant applies to the actual evolving state. It does not unfold a reused register into a new copy of the entire computation.

Exact finite fixture plus the general register-product invariant; the source proves `workspaceDegree_run` for arbitrary gate lists.

The invariant handles duplicates, NOT, and CNOT

Suppose a gate toggles target u by a product of c controls. In a product list L that contains u, remove all occurrences of u to obtain R. Boolean idempotence leaves one updated target factor, and

$$\prod_{i\in L} s_i' = s_u \prod_{i\in R} s_i + \left(\prod_{j\in\text{controls}} s_j\right) \prod_{i\in R} s_i.$$

The old invariant bounds these terms by $t+1+|R|$ and $t+c+|R|$. Since $|R|+1 \le |L|$, the budget rises by at most $\max\{c-1, 0\}$.

Gate	c	toggle expression	extra budget
NOT	0	1	0
CNOT	1	one control	0
Toffoli	2	two controls	1

If the target is absent from L, the product is unchanged. An empty list is the constant-one product. Repeated target indices are removed using idempotence; repeated other indices can only reduce multilinear degree. Controls already appearing in R also cause no difficulty.

For general XOR–AND computations, the written proof introduces a Boolean variable for each AND output. Every wire is affine in inputs and preceding gate variables. Substitute those variables in reverse topological order. A squarefree monomial contains each substituted variable at most once, so replacing it by a quadratic raises degree by at most one. After t substitutions, the affine output has degree at most $t+1$.

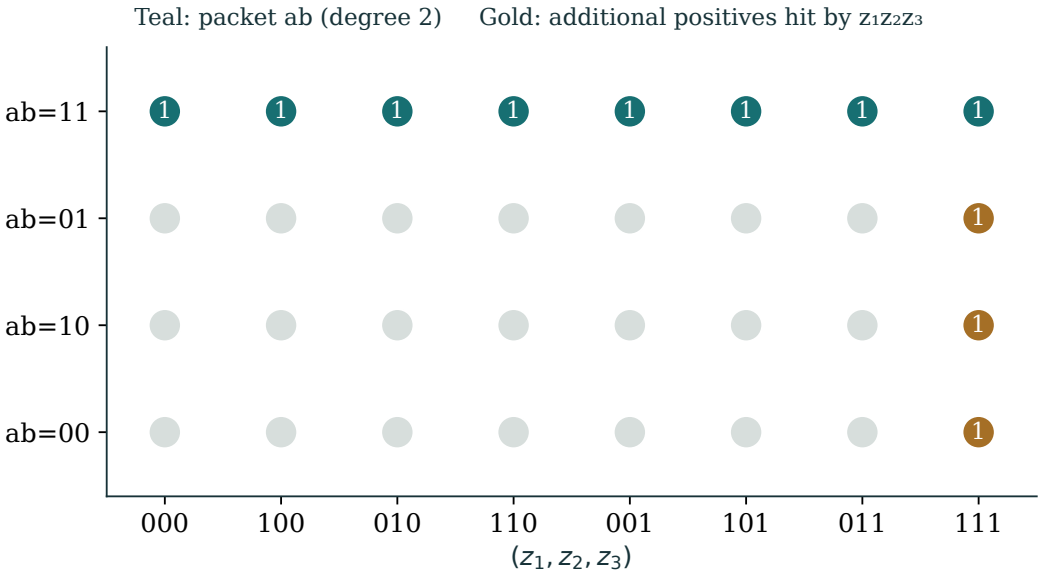
These are two precise proof routes. The general syntax argument is written; the maintained NCT register argument is in Lean. The formalization does not assert a separate arbitrary XOR–AND compiler or a canonical ANF algorithm.

One safe packet need not cover every accepting point

For $k = 3$ consider

$$F_3(a, b, z_1, z_2, z_3) = ab \text{ OR } z_1 z_2 z_3.$$

The packet ab is safe and has degree two. It detects every teal point below. It misses three additional positives, which require the other packet $z_1 z_2 z_3$.



All 32 assignments. There are 11 positives: 8 in the teal row and 3 additional gold points. The coordinate labels use (z_1, z_2, z_3) order.

The one-packet parameter α_1 asks for any nonzero safe polynomial. The complete parameter β asks for safe detection of every positive:

$$\beta_v(f) = \min\{\deg h : h \le f,\ h(v) = 1\}, \qquad \beta(f) = \max_{v : f(v)=1} \beta_v(f).$$

For this monotone function, its minimal positive supports have sizes two and three. Thus $\alpha_1(F_3) = 2$ while $\beta(F_3) = 3$. Selecting one easy accepting region loses the pointwise obstruction at the other region.

All 32 points and the two parameter values were freshly recomputed; the general monotone minterm characterization is a written proof for nonzero functions.

Balanced cuts turn coverage into a Toffoli bound

Choose a cut after $\lfloor T/2 \rfloor$ Toffolis. Both sides have at most $\lceil T/2 \rceil$ Toffolis. Reversing the suffix preserves its count because every gate is self-inverse. Each forward or inverse-suffix coordinate has degree at most $\lceil T/2 \rceil + 1$, and their XOR has the same upper bound.

For every positive v, completeness chooses a discrepancy H_i that detects v. Its safety and degree bound imply

$$\beta(f) \le \left\lceil \frac{T}{2} \right\rceil + 1, \qquad 2\beta(f) \le T + 3.$$

The integer rounding matters. If $T = 2r$, the first bound gives $2\beta \le T + 2$. If $T = 2r + 1$, it gives $2\beta \le T + 3$. The uniform bound uses the latter constant.

An affine output mask $\ell(x)$ costs zero additional Toffolis: append CNOTs from restored inputs and, if needed, a target NOT. This constructs a restored oracle for $f + \ell$, so the bound applies to every affine mask. Define

$$\beta^*(f) = \max_{\ell \text{ affine}} \beta(f + \ell).$$

Together with the output degree bound,

$$\boxed{T \ge \max\{0, \deg f - 1, 2\beta^*(f) - 3\}.}$$

There are finitely many affine masks. The Lean definition uses the least upper bound of their beta values, which equals this finite maximum. Its affine-output theorem constructs legal gates rather than assuming masks are free transformations of an arbitrary interior computation.

Written theorem and archived endpoint `restored_degree_and_cover_bound`; the dual is a separate written result.

An affine shift reveals a harder accepting point

For $k \geq 2$, let $B = z_1 \cdots z_k$ and $F_k = ab$ OR B on $n = k + 2$ inputs. Add the affine mask a to obtain $G_k = F_k + a$. On the face $b = 1$,

$$G_k|_{b=1} = (1 + a)B.$$

This is a singleton indicator on the $k + 1$ variables (a, z). At $v = (a = 0, b = 1, z = 1^k)$, every safe detecting polynomial must restrict to that singleton indicator, so its degree is at least $k + 1$.

The general two-positive indicator construction gives the matching upper bound $\beta(G_k) \leq n - 1 = k + 1$. The maintained family proves

$$\max_{\ell} \alpha_1(F_k + \ell) = 3, \qquad \beta^*(F_k) = k + 1.$$

The gap between one-packet and complete degree therefore grows with k.

k	n	β^*	complete bound $2k - 1$	combined degree/one-packet bound
2	4	3	3	3
3	5	4	5	4
4	6	5	7	5
6	8	7	11	7

For $k = 3$, all 64 affine masks were freshly checked and the optimized parameters are exactly $(3, 4)$. The all-size family proof is not inferred from those 64 cases. Its restored construction has $T \leq 2n-3$, so this remains a linear-size easy family. The exact optimum between the lower and upper constructions is not asserted when they differ.

A degree-one dual certificate for two-input AND

Let $f(a,b) = ab$, with unique positive point $(1,1)$. For degree at most one, the evaluation vector is $v_1(a,b) = (1,a,b)$. The three negative points already span the positive vector.

point (a,b)	1	a	b	AND(a,b)
(0, 0)	1	0	0	0
(1, 0)	1	1	0	0
(0, 1)	1	0	1	0
(1, 1)	1	1	1	1

$$v_1(11) = v_1(00) + v_1(10) + v_1(01)$$

The first three rows are negative inputs. Their XOR equals the last evaluation row.

Any affine polynomial is a coefficient covector $c = (c_0, c_a, c_b)$. Safety means that it pairs to zero with each negative row. The displayed dependence therefore forces its value at $(1,1)$ to be zero. No safe affine packet detects the positive point.

The degree-two polynomial ab does detect it, so $\beta_{11}(f) = \beta(f) = 2$. Since $\beta^* \le n = 2$ and $\beta^* \ge \beta$, $\beta^* = 2$. The complete bound gives $T \ge 1$, attained by one direct Toffoli to the target.

The constant coordinate also checks parity: this dependence uses three negative rows, an odd number. Without the constant coordinate, that parity conclusion would not follow.

Exact finite dependence plus written duality. Lean proves the corresponding AND2 degree and beta exclusions by four-corner parity, without formalizing the general evaluation-span dual.

The finite evaluation-span dual in full

Fix a positive v and integer $d \geq 0$. Let

$$v_d(x) = (x^S)_{S\subseteq[n],\ |S|\leq d}, \qquad U = f^{-1}(0), \qquad N = \operatorname{span}_{\mathbb{F}_2}\{v_d(u) : u \in U\}.$$

The constant monomial $S = \varnothing$ is included. Degree-d coefficient covectors form the finite dual of this evaluation space. A covector is safe exactly when it annihilates N.

If $v_d(v) \in N$, every safe covector vanishes at v. If $v_d(v) \notin N$, extend a basis of N by $v_d(v)$ and then to a basis of the full space. Assign zero to the N basis and one to $v_d(v)$. The resulting covector is a safe detector. Hence

$$\boxed{\beta_v(f) > d \quad \Longleftrightarrow \quad v_d(v) \in N.}$$

Choose a basis of N from negative evaluation rows. A span representation of the positive row gives a subset $W \subseteq U$ with

$$v_d(v) = \sum_{u\in W} v_d(u), \quad |W| \text{ odd}, \quad |W| \leq \sum_{j=0}^{\min(d,n)} \binom{n}{j}.$$

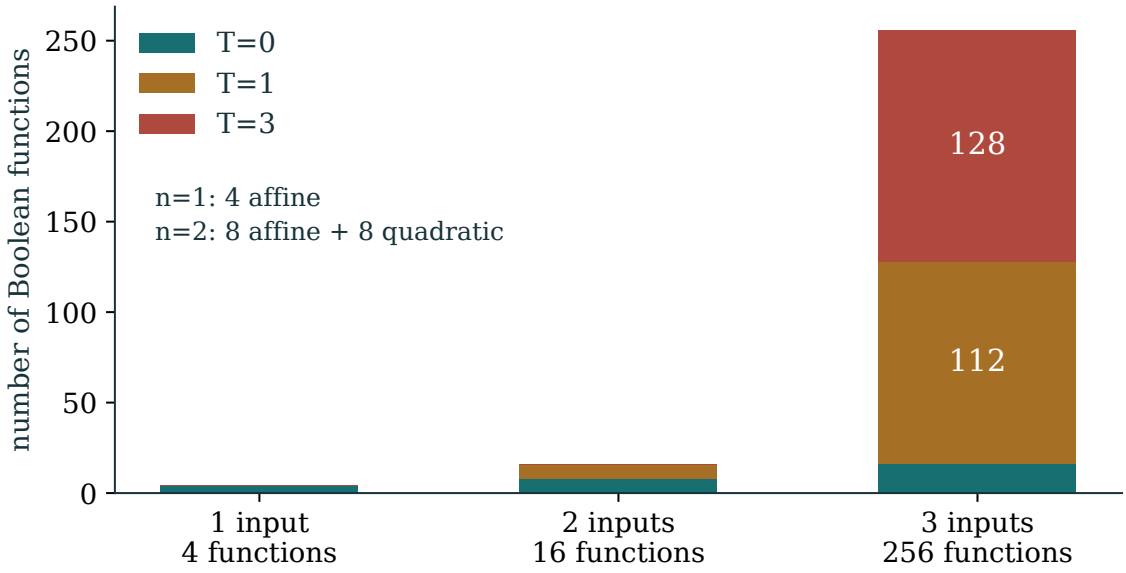
It is a replayable finite obstruction to every safe degree-d detector.

At $d = 0$, a negative exists exactly when the constant-one detector is unsafe. At $d \geq n$, the full evaluation rows are independent and no positive row lies in the negative span. This matches $\beta_v \leq n$ from point indicators. For $f = 0$, there is no positive v and the separate convention is $\beta(0) = 0$.

Fresh enumeration checked all 278 functions on zero through three inputs, all positives, and every $d = 0, \ldots, n$: 4,201 primal/dual checks agree. Those bounded checks support the independently supplied universal proof.

Every function through three inputs has a matching oracle

The small calibration concerns every truth table on one, two, or three inputs. It is not a search through all reversible circuit words. A theorem lower bound and an explicit construction coincide.



Reconstructed source constructions for all 276 functions. Totals are 28 with $T = 0$, 120 with $T = 1$, and 128 with $T = 3$.

Affine functions use target CNOTs and possibly NOT. A nonaffine quadratic through three bits has alternating quadratic rank two, so it is a product of two independent linear forms plus an affine residual. Compute those forms on clean wires, apply one Toffoli, uncompute, and append the residual.

Every three-bit cubic is a translated three-variable monomial plus an affine residual. The singleton monomial needs three Toffolis by complete coverage, and compute/toggle/uncompute attains that number.

The new figure script rebuilds and validates all 276 circuits and both target branches, covering 4,240 scalar input/target cases via exact truth tables. Its totals match the frozen baseline. Zero-input constants were checked in the separate dual suite, rather than being added to this 276-function count.

Matching written lower bounds and exact constructed implementations; not exhaustive optimal-circuit synthesis. The generic monomial cost theorem is not newly Lean-certified by this edition.

Projection is defined in the original coordinates

For $m \geq 2$, put $a = (1, \ldots, 1)$ and

$$V = \mathbb{F}_2^m \setminus \{0, a\}, \qquad n = 2^m - 2.$$

There is one input bit x_v for each $v \in V$. Define

$$L_j(x) = \sum_{v : v_j = 1} x_v, \qquad P_m(x) = \prod_{j=1}^{m} L_j(x).$$

The linear map $x \mapsto L(x)$ has rank m, because the standard basis columns are present. In the L coordinates, P_m is a monomial in m independent bits. Its restored nonlinear cost is $2m - 3$.

Now extract the quadratic homogeneous ANF component Q_m in the *original* x_v coordinates. Since $P_m(0) = P_m(e_v) = 0$, for distinct u, v its quadratic coefficient is

$$[x_u x_v] P_m = P_m(e_u + e_v) = [u + v = a].$$

The involution $v \mapsto v + a$ pairs every variable with one other variable:

$$Q_m = \sum_{\{v, v+a\}} x_v x_{v+a}.$$

It is a quadratic inner product on $n/2$ disjoint pairs. Its alternating polar matrix has full rank n.

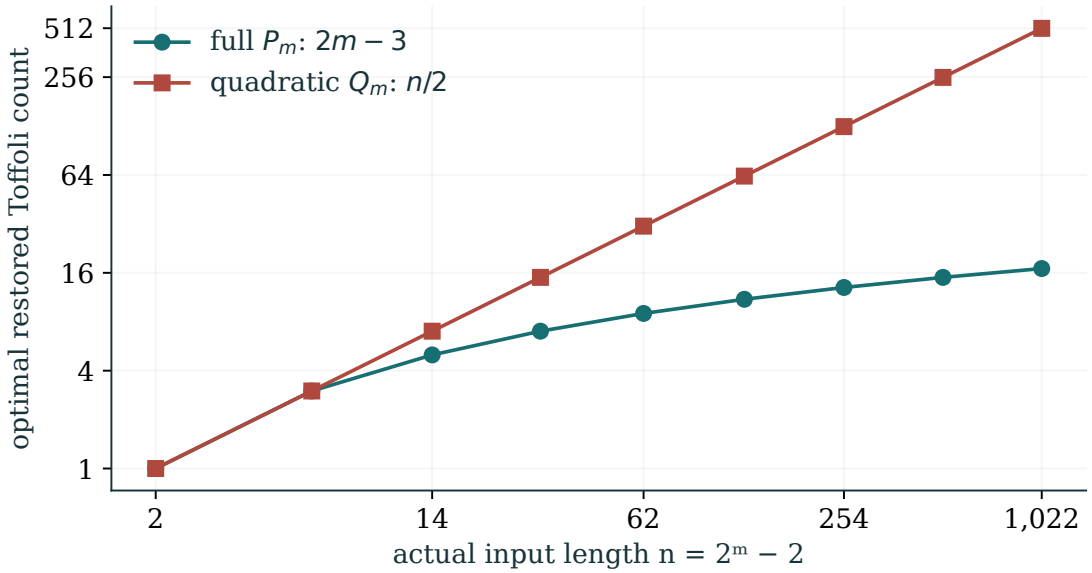
The coordinate choice is essential. Homogeneous grade is not invariant under arbitrary affine changes of variables. A cheap representation in the L coordinates does not imply cheap independent generation of the quadratic grade in the original coordinates. The comparison keeps the definition of projection fixed throughout.

Exact nonlinear projection costs versus actual input length

The source proves the all-size formulas

$$M(P_m) = m - 1, \quad T_{\text{rest}}(P_m) = 2m - 3, \qquad M(Q_m) = T_{\text{rest}}(Q_m) = n/2.$$

The chart uses actual input length $n = 2^m - 2$ on its horizontal axis.



Exact formulas for $m = 2, \ldots, 10$. Both axes are logarithmic; the plotted points are not optimal-circuit search results.

For Q_m, each disjoint quadratic pair is accumulated directly into the target with one Toffoli. The lower bound allows nonlinear intermediates: any computation with t multiplications is affine on an affine subspace of codimension at most t. A nondegenerate quadratic can be affine only on a subspace with isotropic direction space, whose dimension is at most $n/2$. Thus $t \geq n/2$.

The separation grows exponentially in m, but n itself grows exponentially in m. The quadratic cost is linear in actual input length. The conclusion is about nonlinear generation and a specified grade extraction operation, not a superpolynomial unrestricted circuit bound.

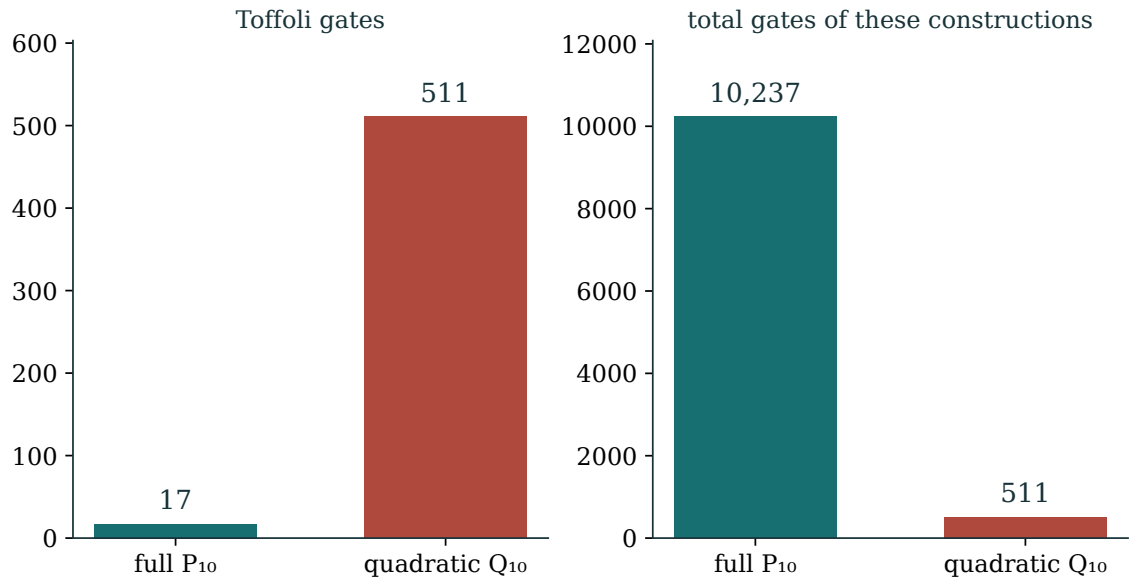
All-size written cost proofs; constructor gate counts freshly asserted through $m = 10$. No full truth-table enumeration at $m > 4$ is claimed.

At $m = 10$, the total-gate accounting points the other way

Here $n = 1,022$. The full function uses 17 Toffolis; its quadratic grade needs 511. However, computing and uncomputing the m parities also uses

$$2m(2^{m-1} - 1) = 20 \cdot 511 = 10,220 \quad \text{CNOTs.}$$

The explicit full construction has $10,237$ total gates. The quadratic accumulation uses only 511 total gates.



Two different resource comparisons at the same input length. The right panel reports the totals of the stated constructions, not an asserted optimal total for P_{10}.

The left panel is a valid nonlinear-cost separation. The right panel prevents interpreting it as an exponential ordinary total-gate separation. The total full construction is $O(n \log n)$, whereas the quadratic construction is $n/2$ gates.

A proposed invariant that charges an original computation for every separately extracted grade would have to justify that extraction's cost. These formulas refute an overhead bounded only by a polynomial in the original nonlinear count, independently of input and linear resources. They do not exclude an overhead polynomial in $n + T$ or a different precisely defined invariant.

Exact counts obtained from the source constructors; optimal Toffoli formulas have written proofs. The full total-gate count is an upper construction count.

Quadratic projection can destroy safety

At $m = 3$, variables are ordered by source labels $v = 1, 2, 3, 4, 5, 6$. Assignment index 13 is read least-significant-bit first:

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (1, 0, 1, 1, 0, 0).$$

The active labels have syndrome $1 + 3 + 4 = 6$, not $a = 7$.

input	x_1	x_2	x_3	x_4	x_5	x_6
label v	1	2	3	4	5	6
x (13)	1	0	1	1	0	0

$$L = (0, 1, 1) \Rightarrow P_3 = 0; \quad Q_3 = x_1x_6 + x_2x_5 + x_3x_4 = 1$$

A concrete accepted point for Q_3 that is rejected by P_3. The variable ordering matches the source checker.

The three linear forms evaluate to $L = (0, 1, 1)$, so $P_3 = L_1L_2L_3 = 0$. The quadratic grade has pairs $(1, 6)$, $(2, 5)$, $(3, 4)$, so

$$Q_3 = x_1x_6 + x_2x_5 + x_3x_4 = 0 + 0 + 1 = 1.$$

Thus $Q_3 \notin I_{P_3}$: homogeneous extraction does not preserve the positive-support ideal. The whole polynomial is supported on its own accepting set, but its isolated homogeneous components need not be.

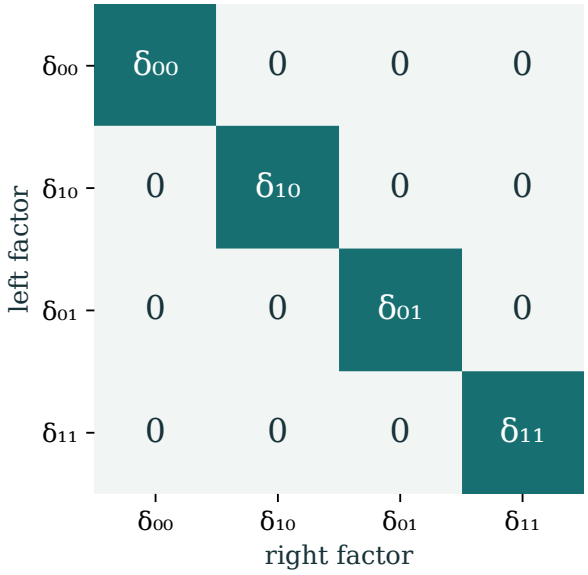
The new script checks all 64 inputs here, finds 24 false positives, and confirms that 13 is the first in the stated integer ordering. The frozen $m = 4$ control similarly has 7,616 false positives among its 2^{14} assignments. These are separate finite statements with exact ranges.

Fresh complete $m = 3$ truth-table check; baseline $m = 4$ check was reproduced by the full source verifier. Safety failure is already established by the single valid assignment above.

Witness indicators are orthogonal idempotents

Let K be a field and $W = \{0,1\}^m$. The full witness algebra is $B = K^W$ with pointwise addition and multiplication. For each witness v, the indicator δ_v is one at v and zero elsewhere. Then

$$\delta_v^2 = \delta_v, \qquad \delta_u \delta_v = 0 \ (u \neq v), \qquad \sum_v \delta_v = 1.$$



The four-witness multiplication table at $m = 2$. Each nonzero diagonal entry is its own point indicator.

The four indicator functions form a basis of K^4. General m has 2^m such basis elements. A verifier is the sum of the indicators of its accepted witnesses, and pointwise multiplication models intersection of accepting sets.

This algebra retains exact witness compatibility: disjoint singleton predicates multiply to zero. A summary that preserves that product operation while retaining every singleton must preserve the same orthogonal structure. The following dimension theorem says exactly how much fixed linear space that requires.

Exact $m = 2$ multiplication table; general product-algebra identities hold over any field. No characteristic-two assumption is needed for the dimension proof.

Faithful linear multiplicative summaries need 2^m dimensions

Suppose $\Phi : K^W \to A$ is K-linear and multiplicative, and $\Phi(\delta_v) \neq 0$ for every $v \in W$. Put $e_v = \Phi(\delta_v)$. The images are nonzero orthogonal idempotents.

If $\sum_v c_v e_v = 0$, multiply by e_u. Orthogonality removes every other term and idempotence gives $c_u e_u = 0$. Since K is a field and $e_u \neq 0$, $c_u = 0$. Hence all e_v are linearly independent and

$$\boxed{\dim_K A \geq |W| = 2^m.}$$

No unital hypothesis is needed. At $m = 0$, there is one empty witness and the bound is simply $\dim_K A \geq 1$.

For a unital scalar map to K, scalar idempotents are zero or one. Orthogonality permits at most one nonzero image, and preservation of one requires exactly one. Such a map is evaluation at one witness. For $m \geq 1$ it loses at least one other singleton verifier. For $m = 0$, there is no other witness to lose.

> **Model boundary.** The theorem quantifies a fixed summary serving all predicates at a witness length, preserving linear combinations and exact pointwise multiplication. An unrestricted algorithm need not use such a summary. Syntax, nonlinear encodings, changing spaces, context-dependent operations, and succinct expressions are outside these hypotheses.

An exponentially large algebra can still contain short circuit expressions. Therefore exponential dimension does not imply exponential expression length or a lower bound for every algorithm on one explicit decision family.

Distinct witness monomials prevent formal cancellation

Over a characteristic-two field, define

$$P_R(z) = \sum_{w \in \{0,1\}^m} R(w) \prod_i z_i^{w_i}.$$

The z_i are formal indeterminates, not Boolean variables. Different witnesses produce different monomials, so $P_R = 0$ exactly when there is no accepting witness.

witness (w_1,w_2,w_3)	R(w)	retained monomial
000	0	—
100	1	z1
010	1	z2
110	0	—
001	0	—
101	1	z1z3
011	1	z2z3
111	0	—

An exact three-bit verifier accepts witness indices $1, 2, 5, 6$ in least-significant-bit order. Four accepts have parity zero, but their monomials are distinct.

Here

$$P_R = z_1 + z_2 + z_1 z_3 + z_2 z_3 = (z_1 + z_2)(1 + z_3) \neq 0.$$

The ordinary unweighted characteristic-two sum of the four accepts is zero. The formal polynomial avoids this cancellation by retaining witness labels as monomial supports.

Nonzero as a formal polynomial does not mean nonzero at every field point. For instance, this example vanishes whenever $z_1 = z_2$ or $z_3 = 1$. Random evaluation is governed by a root bound, not by the formal identity alone. Evaluating the polynomial efficiently remains a separate task.

Exact support fixture; general no-cancellation argument is written. The source finite evaluator retains 2^m coefficients.

A short expression inside a trace is not a computed trace

Let τ_w extract the top multilinear coefficient in the Boolean witness ring. For any $G(w)$ over a characteristic-two field,

$$\tau_w(G) = \sum_{v \in \{0,1\}^m} G(v).$$

A proper-support monomial evaluates to one an even number of times; the full-support monomial does so once. This proves the identity directly.

Applying it to a weighted expression gives

$$P_R(z) = \tau_w \left[R(w) \prod_{i=1}^{m} \big(1 + (1 + z_i) w_i\big) \right].$$

At $w_i = 0$, a multiplier equals one. At $w_i = 1$, it equals z_i. Summing evaluations therefore produces exactly the witness monomials. For the preceding verifier, the four surviving products are $z_1, z_2, z_1 z_3, z_2 z_3$.

The circuit expression *inside* the trace adds only $O(m)$ operations to the verifier. That observation bounds the expression's construction, not the cost of applying τ_w. The finite checker computes with a complete table of 2^m witness coefficients.

Established	Not supplied by the expression
Small weighted circuit inside τ	A polynomial-time evaluator for the trace
Formal polynomial detects nonemptiness	A deterministic identity test from one evaluation
Exact finite table algorithm	Polynomial storage in the witness length

At $m = 0$, the empty product is one, the top coefficient is the constant itself, and there is a single empty witness. The definitions remain valid.

Disjoint-variable factorization uses each predicate's own block

Suppose A depends on witness variables in block S and B on a disjoint block T. Write $\tau_{z_S}(A) = \sum_s A(s) z_S^s$ for the weighted trace on S, distinct from the unweighted top-coefficient notation τ_w. Weighted sum over the union factors as

$$\sum_{s,t} A(s)B(t) z_S^s z_T^t = \left(\sum_s A(s) z_S^s\right)\left(\sum_t B(t) z_T^t\right).$$

Each factor is traced on its own variable block. The union trace is then their product. This is an ordinary factorization of a finite double sum.

For $A = w_1$ and $B = w_2$,

$$\tau_{z_1}(A) = z_1, \qquad \tau_{z_2}(B) = z_2, \qquad \tau_{z_1,z_2}(AB) = z_1 z_2.$$

Tracing both factors over the full two-bit ambient domain introduces unused variable sums:

$$\tau_{z_1,z_2}(A) = z_1(1+z_2), \qquad \tau_{z_1,z_2}(B) = z_2(1+z_1).$$

Their product generally differs from the union trace. At $z_1 = z_2 = 1$, each full ambient factor is zero, while the trace of AB is one. Disjoint dependence alone does not justify multiplying traces with mismatched domains.

The source checker tests all four one-bit predicates per block and every pair of GF(16) weights: $4 \cdot 4 \cdot 16^2 = 4,096$ own-block factorization cases. It does not claim an ambient-domain product identity.

General written factorization under the own-block convention; exact finite source check and an explicit ambient-domain counterexample.

Overlap breaks the attempted context-free product rule

One shared witness bit already defeats multiplicativity. With $\tau_z(P) = P(0) + zP(1)$,

$$\tau_z(w) = z, \qquad \tau_z(1+w) = 1, \qquad \tau_z(w(1+w)) = 0.$$

The product of the separate summaries is z. Also $w^2 = w$ in the Boolean ring, so $\tau_z(w^2) = z$, whereas $\tau_z(w)^2 = z^2$.

expression	trace after product	product of traces
w(1+w)	0	θ
$w^2 = w$	θ	θ^2

GF(16) = F_2[θ]/(θ⁴+θ+1); θ = integer 2, θ² = integer 4

A concrete field point $z = \theta$ with $\theta \neq 0, 1$. The two columns are unequal in both rows.

The field is $\mathbb{F}_{16} = \mathbb{F}_2[\theta]/(\theta^4 + \theta + 1)$. The checker stores field elements as four-bit coefficient vectors: θ is integer 2 and θ^2 is integer 4. These integers encode field elements; they are not ordinary integer multiplication. Exact field multiplication gives the reported mismatch.

The first row loses witness compatibility: w and $1 + w$ have no common accepting witness, yet their separate weighted summaries multiply to a nonzero value. The second row loses Boolean idempotence. This is a genuine counterexample to a context-free multiplicative trace compiler, not a floating-point effect or an implementation anomaly.

Exact GF(16) fixture imported from the source field routines. The general overlap failure is symbolic and does not depend on the selected field.

Randomized detection is conditional on efficient evaluation

The witness polynomial has total degree at most m. Over a finite field K, a nonzero polynomial of total degree at most m vanishes on at most an $m/|K|$ fraction of independent uniform field points. The standard inductive proof separates the leading coefficient in the last variable and applies the univariate root bound.

If a uniform polynomial-time exact evaluator for the weighted trace were available, with efficient field representation and arithmetic and $|K| \geq 4m$, one evaluation would accept a satisfiable verifier with probability at least $3/4$. An unsatisfiable verifier has the zero polynomial, so it never produces a false acceptance. Repetition with independent samples reduces the false-rejection probability to at most 4^{-r} after r trials.

Witness bits	GF(16) points	maximum zeros in source tests	degree-bound zeros
1	16	1	1
2	256	31	32
3	4,096	721	768

These source tests enumerate every verifier at each displayed witness length and all field points for the polynomial evaluations. The independent coefficient-extraction comparison has a narrower three-variable grid: coordinates $0, 1, 2, 3$, rather than all GF(16) points. Both scopes are recorded.

This is a conditional randomized algorithmic implication. The required efficient trace evaluator is not supplied. The existing exact evaluator is exponential in witness length, and no deterministic $P = NP$ conclusion is drawn. Random bits and polynomial output description alone do not account for the evaluation operation.

Reproduce the figures and inspect their exact evidence

All 12 figure pairs are vector PDFs plus raster PNG previews. Their single generation command, from the publication-set root, is

```
python scripts/make_figures.py
```

The script imports the original gate, profile, construction, and GF(16) functions without invoking their release-writing entry points. It writes only new figures and `reviews/visual_examples.json`. That record contains fixture truth-table integers, ordered bits, exact gate lists, scalar counts, the source hashes, and the distinction between finite assertions and plotted all-size formulas.

The separate dual sanity suite runs as

```
python scripts/core_dual_checks.py \
  --output reviews/core_dual_checks.json
```

It compares exhaustive degree-bounded safe truth tables with a separate Gaussian-elimination calculation of the negative evaluation span. Its range is all truth tables at $n = 0, 1, 2, 3$, not an extrapolation.

Evidence layer	Publication treatment
General mathematics	Main paper written proofs with explicit model boundaries
Formal subset	C01, maintained NCT degree, and numerical C02; fresh compiler and kernel replay passed
New figures	12 exact generated examples/formula plots, each with fixture provenance
Fresh source checks	Complete source `--all` verification and preserved historical outputs passed in the current replay
Rendering	This guide is compiled and visually checked separately from mathematical and kernel verification

The source release, earlier companions, and archived evidence remain unchanged. Publication corrections clarify evidence labels, the nonzero monotone minterm domain, the $m = 0$ scalar-summary boundary, and trace-variable domains. No new conjecture or extension to unrestricted hardness is added.

Figure and manuscript sources are included for review and reproduction.

PART V

Formal and computational companions

The companions identify formal coverage, exact finite computations, and the obligations that remain open. Verification records establish the stated results within their declared models and bounds.

IN THIS PART

The formal arithmetic bridge 264

Symmetry: supplementary proofs and computation 285

Gate matrices, census data, and open obligations 290

The formal circuit model 301

Claim coverage and theorem index 330

Verification and proof boundaries 333

APPENDIX A

The formal arithmetic bridge

A.1 What this guide certifies

The completed fixed example is a bridge between exact arithmetic and ordinary complex linear algebra. It certifies a twelve-dimensional Fourier matrix, a Ramanujan projector, two normalized states with the same initial probabilities, and their opposite outcomes after a basis change. This guide maps those claims to actual Lean declarations rather than treating numerical agreement as a proof.

The five-module formal package contains **96 named mathematical theorems**. Its current verification rebuilt the source, inspected the axiom dependencies of every named theorem, and replayed all five modules. Historical proof records remain separate from this fresh execution.

Production module	Named theorems
`QuantumArithmeticBridge`	22
`FiniteFourier`	17
`SemanticBridge`	30
`Conventions`	17
`Publication`	10
Total	96

These counts include supporting lemmas. They count named source declarations, not new discoveries, compiler jobs, or declarations generated internally by Lean. The namespace is `Komorebi.QuantumArithmetic` throughout.

A.2 The typed mathematical target

The complex matrices act on vectors indexed by $\mathrm{Fin}(12)$. The Fourier convention is

$$F_{nk} = \frac{1}{\sqrt{12}} \exp\left(\frac{2\pi i n k}{12}\right), \qquad n, k \in \{0, \ldots, 11\}.$$

The adjoint F^* therefore carries the negative exponential sign. The selector D is diagonal with value one precisely at the residues $U = \{1, 5, 7, 11\}$. The transported operator is $P = FDF^*$.

Write

$$c = (4, 0, 2, 0, -2, 0, -4, 0, -2, 0, 2, 0), \qquad \psi = \frac{c}{\sqrt{48}}, \qquad \chi = \frac{|c|}{\sqrt{48}}.$$

Both vectors have norm one. Their initial probability histograms are identical, but their inner product is zero. The vector u has entry $1/2$ on U and zero elsewhere. Lean proves the actual complex identities $Fu = \psi$ and $F^*\psi = u$, as well as transformed coprime probability one for ψ and zero for χ.

The statements are finite and exact. No parameterized arbitrary-modulus theorem, general circuit construction, physical phase-removal process, or quantum speedup is included in these formal claims.

A.3 Revision reconciliation

Two retrieved archives carry version `0.2.0`, but they do not have the same active source. The earlier four-module checkpoint has 59 mathematical theorems and four separately checked refutation controls. The later five-module checkpoint has 96 mathematical theorems and three rejected control files. Its proof receipt is the later `verification/final/REPORT.json`.

The original 22 finite-table statements are retained inside both checkpoints. They are not counted a second time. Likewise, $59 + 96$ is not a meaningful theorem total: these are overlapping revisions of one formal development.

The earlier proof boundary disclosed that explicit projector transport and normalized-state matrix actions had not yet been supplied. The later source supplies both, connects the Fourier matrix to the usual complex exponential, and proves a histogram nonreconstruction theorem. The earlier boundary remains a record of its own revision, rather than a gap list for the current publication.

The entire later checkpoint is preserved under `lean/checked_checkpoint`. The source files are not edited to match this guide.

A.4 Architecture of the proof

The proof crosses two representations. First, rational coordinates support exact finite calculations. Second, a semantics map transfers those calculations to ordinary complex matrices and vectors.

Layer	Role
Rational tables	Signal, projector entries, filtering, and energies.
Exact coordinates	Algebra in the basis $1, \sqrt{3}, i, i\sqrt{3}$.
Finite matrices	All entries indexed by $\mathrm{Fin}(12)$, with exact finite sums.
Complex semantics	Preservation of multiplication, adjoint, matrix action, and squared norm.
Conventions	Identification of roots and normalization with standard exponentials and square roots.
Publication theorem	Bundling of the actual complex identities and reconstruction obstructions.

The active import chain is `QuantumArithmeticBridge` into `FiniteFourier`, then `SemanticBridge`, then `Conventions`, then `Publication`. Supporting Mathlib imports enter at the layer that uses them. There is no circular theorem dependency.

A.5 The rational signal and its filter

The first layer uses literal rational lists. The chosen perturbation is $r_n = 3\cos(\pi n/2) + 2$, whose displayed rational entries can be certified without numerical trigonometric evaluation. Its sum with c is the test signal.

Literal source. `QuantumArithmeticBridge.lean`, physical lines 51–61.

```
/-- Ramanujan sum c_12(n), n = 0,...,11. -/
def ramanujan12 : QVector :=
  [4, 0, 2, 0, -2, 0, -4, 0, -2, 0, 2, 0]

/-- A deliberately off-band perturbation: 3 cos(pi n/2) + 2. -/
def noise12 : QVector :=
  [5, 2, -1, 2, 5, 2, -1, 2, 5, 2, -1, 2]

/-- Signal used in the essay's filter demonstration. -/
def testSignal12 : QVector :=
  [9, 2, 1, 2, 3, 2, -5, 2, 3, 2, 1, 2]
```

The cyclic matrix has entry $c_{(n-m) \bmod 12}/12$. The original layer proves its symmetry, idempotence, preservation of c, rejection of the chosen perturbation, and exact recovery from the displayed contaminated signal. The exact energy contract is $150 = 48 + 102$.

Literal source. `QuantumArithmeticBridge.lean`, physical lines 92–126.

```
theorem projector12_idempotent :
    matMulN 12 projector12 projector12 = projector12 := by
  decide +kernel

/-- The arithmetic signal lies in the selected subspace. -/
theorem projector12_fixes_ramanujan :
    matVec projector12 ramanujan12 = ramanujan12 := by
  decide +kernel

/-- The chosen constant-plus-frequency-3 perturbation lies in the complementary subspace. -/
theorem projector12_kills_noise :
    matVec projector12 noise12 = List.replicate 12 0 := by
  decide +kernel

/-- Therefore the exact filter recovers the arithmetic component. -/
theorem projector12_recovers_test_signal :
    matVec projector12 testSignal12 = ramanujan12 := by
  decide +kernel

/-- The arithmetic amplitude vector has squared norm 48. -/
theorem ramanujan12_energy : sqEnergy ramanujan12 = 48 := by
  decide +kernel

/-- The perturbation has squared norm 102. -/
theorem noise12_energy : sqEnergy noise12 = 102 := by
  decide +kernel

/-- The full signal has squared norm 150. -/
theorem testSignal12_energy : sqEnergy testSignal12 = 150 := by
  decide +kernel

/-- Exact orthogonal energy decomposition 150 = 48 + 102. -/
theorem energy_decomposition :
    sqEnergy testSignal12 = sqEnergy ramanujan12 + sqEnergy noise12 := by
  decide +kernel
```

The list operations use finite lists, including zero defaults and zip-based dot products. These facts are therefore claims about the supplied twelve-entry objects. They should not be silently interpreted as

universal matrix theorems for arbitrary ragged lists. The later Fin(12) layer supplies the typed matrix identities used in the publication.

A.6 The exact phase arithmetic

The structure `Cyclo12` represents

$$(a,b,c,d) \longmapsto a + b\sqrt{3} + i(c + d\sqrt{3}), \qquad a,b,c,d \in \mathbb{Q}.$$

Its multiplication explicitly incorporates $(\sqrt{3})^2 = 3$ and $i^2 = -1$. It is a concrete arithmetic representation. The development does not need or install a global field instance to certify the finite example.

Literal source. `QuantumArithmeticBridge.lean`, physical lines 130–141.

```
/--
An exact coordinate model for Q(sqrt(3), i).  The tuple (a,b,c,d) denotes
  a + b*sqrt(3) + i*(c + d*sqrt(3)).
Only the concrete arithmetic needed for the q=12 certificate is used below;
we do not install a global field instance.
-/
structure Cyclo12 where
  a : ℚ
  b : ℚ
  c : ℚ
  d : ℚ
  deriving DecidableEq, Repr
```

Literal source. `QuantumArithmeticBridge.lean`, physical lines 155–173.

```
/-- Multiplication in Q(sqrt(3), i). -/
def cMul (x y : Cyclo12) : Cyclo12 :=
  ⟨x.a*y.a + 3*x.b*y.b - x.c*y.c - 3*x.d*y.d,
   x.a*y.b + x.b*y.a - x.c*y.d - x.d*y.c,
   x.a*y.c + 3*x.b*y.d + x.c*y.a + 3*x.d*y.b,
   x.a*y.d + x.b*y.c + x.c*y.b + x.d*y.a⟩

def cConj (x : Cyclo12) : Cyclo12 := ⟨x.a, x.b, -x.c, -x.d⟩

def cPow (x : Cyclo12) : Nat → Cyclo12
  | 0 => cOne
  | n + 1 => cMul (cPow x n) x

def cSum : List Cyclo12 → Cyclo12
  | [] => cZero
  | x :: xs => cAdd x (cSum xs)

/-- zeta = exp(2*pi*i/12) = sqrt(3)/2 + i/2. -/
def zeta12 : Cyclo12 := ⟨0, (1/2 : ℚ), (1/2 : ℚ), 0⟩
```

The phase $\zeta = (0, 1/2, 1/2, 0)$ has complex meaning $\sqrt{3}/2 + i/2$. Exact finite certificates establish its twelfth power, inverse, and absence of a smaller positive power below twelve. Later the convention module identifies it with $\exp(\pi i/6)$.

The Ramanujan vector is obtained by summing powers of this phase over $1, 5, 7, 11$. The negative-sign unnormalized DFT has value 12 at these four frequencies and zero elsewhere. The absolute-value

vector has Fourier support on the even frequencies instead. These are exact phase calculations over rational coordinates.

A.7 Finite indices and normalized Fourier certificates

The finite layer uses `Index := Fin 12`. Its normalized Fourier scale is represented exactly as $\sqrt{3}/6 = 1/\sqrt{12}$, and its state scale as $\sqrt{3}/12 = 1/\sqrt{48}$. The use of a finite index type makes every row and column well defined without list-length side conditions.

Literal source. `FiniteFourier.lean`, physical lines 58–93.

```
/-- The stored arithmetic selector is exactly the coprimality predicate. -/
theorem coprime_residues_exact :
    (List.range 12).filter (fun a => Nat.gcd a 12 == 1) = unitResidues12 := by
  decide +kernel

theorem root_exact_order :
    ∀ n : Fin 12, 0 < n.val → cPow zeta12 n.val ≠ cOne := by
  decide +kernel

theorem normalized_fourier_left :
    ∀ i j, kMatMul (kAdjoint exactFourier) exactFourier i j = kIdentity i j := by
  decide +kernel

theorem normalized_fourier_right :
    ∀ i j, kMatMul exactFourier (kAdjoint exactFourier) i j = kIdentity i j := by
  decide +kernel

def selectedKernel : KMatrix := fun i j =>
  kSum (fun k => if Nat.gcd k.val 12 = 1 then
    cMul (exactFourier i k) (cConj (exactFourier j k)) else cZero)

theorem exact_projector_transport :
    ∀ i j, selectedKernel i j = exactProjector i j := by
  decide +kernel

theorem exact_projector_matches_rational_table :
    ∀ i j, exactProjector i j = cRat ((projector12.getD i.val []).getD j.val 0) := by
  decide +kernel

theorem exact_state_preparation :
    ∀ i, kMatVec exactFourier exactUnitState i = exactPsi i := by
  decide +kernel

theorem exact_state_recovery :
    ∀ i, kMatVec (kAdjoint exactFourier) exactPsi i = exactUnitState i := by
  decide +kernel
```

The two Fourier identities above quantify over all 12×12 entries. Kernel reduction proves these concrete universal statements over a finite type. The selected-kernel equality and state actions are also complete finite certificates, rather than floating-point matrix comparisons.

A.8 Why the coordinate certificates mean complex identities

The semantics map is explicitly defined with the positive real square root. Soundness for addition, multiplication, conjugation, powers, and finite sums prevents the exact coordinate layer from being a detached calculation with an unproved interpretation.

Literal source. `SemanticBridge.lean`, physical lines 14–56.

```
/-- The exact coordinate arithmetic is interpreted in the ordinary complex numbers. -/
noncomputable def interpret (x : Cyclo12) : ℂ :=
  ⟨(x.a : ℝ) + (x.b : ℝ) * Real.sqrt 3,
   (x.c : ℝ) + (x.d : ℝ) * Real.sqrt 3⟩

@[simp] theorem interpret_zero : interpret cZero = 0 := by
  apply Complex.ext <;> simp [interpret, cZero]

@[simp] theorem interpret_one : interpret cOne = 1 := by
  apply Complex.ext <;> simp [interpret, cOne]

@[simp] theorem interpret_rat (r : ℚ) : interpret (cRat r) = (r : ℂ) := by
  apply Complex.ext <;> simp [interpret, cRat]

@[simp] theorem interpret_add (x y : Cyclo12) :
    interpret (cAdd x y) = interpret x + interpret y := by
  apply Complex.ext <;> simp [interpret, cAdd] <;> ring

@[simp] theorem interpret_mul (x y : Cyclo12) :
    interpret (cMul x y) = interpret x * interpret y := by
  apply Complex.ext <;> simp [interpret, cMul, Complex.mul_re, Complex.mul_im] <;>
    ring_nf <;> norm_num [Real.sq_sqrt]

@[simp] theorem interpret_conj (x : Cyclo12) :
    interpret (cConj x) = star (interpret x) := by
  apply Complex.ext <;> simp [interpret, cConj]
  ring

@[simp] theorem interpret_pow (x : Cyclo12) (n : Nat) :
    interpret (cPow x n) = interpret x ^ n := by
  induction n with
  | zero => simp [cPow]
  | succ n ih => simp [cPow, ih, pow_succ]

@[simp] theorem interpret_sum (xs : List Cyclo12) :
    interpret (cSum xs) = (xs.map interpret).sum := by
  induction xs with
  | nil => simp [cSum]
  | cons x xs ih => simp [cSum, ih]

@[simp] theorem interpret_kSum (f : Index → Cyclo12) :
    interpret (kSum f) = ∑ i, interpret (f i) := by
  simp only [kSum, interpret_sum, List.map_ofFn, List.sum_ofFn]
```

The multiplication proof uses complex extensionality, real polynomial normalization, and the square-root identity. It proves preservation for arbitrary coordinates, not merely for the handful of roots that occur in the example. The power and finite-sum proofs use induction.

Injectivity of this interpretation is not needed to transfer an equality: if two coordinate expressions are proved equal, applying the interpretation gives equal complex expressions. No claim of a globally packaged field equivalence is required by the theorem chain.

A.9 Matrix transport and actual unitarity

The semantics layer proves that coordinate matrix multiplication, adjoint, identity, and matrix-vector action have their ordinary complex meanings. The matrices F, P, D and the normalized states are then defined directly in that complex representation.

Literal source. `SemanticBridge.lean`, physical lines 65–91.

```
@[simp] theorem complexMatrix_mul (A B : KMatrix) :
    complexMatrix (kMatMul A B) = complexMatrix A * complexMatrix B := by
  ext i j
  simp [complexMatrix, kMatMul, Matrix.mul_apply]

@[simp] theorem complexMatrix_adjoint (A : KMatrix) :
    complexMatrix (kAdjoint A) = (complexMatrix A)ᴴ := by
  ext i j
  simp [complexMatrix, kAdjoint, Matrix.conjTranspose_apply]

@[simp] theorem complexMatrix_identity :
    complexMatrix kIdentity = 1 := by
  ext i j
  by_cases h : i = j <;> simp [complexMatrix, kIdentity, h, Matrix.one_apply]

@[simp] theorem complexVector_mul (A : KMatrix) (v : KVector) :
    complexVector (kMatVec A v) = (complexMatrix A).mulVec (complexVector v) := by
  ext i
  simp [complexVector, complexMatrix, kMatVec, Matrix.mulVec, dotProduct]

noncomputable def F : Matrix Index Index ℂ := complexMatrix exactFourier
noncomputable def P : Matrix Index Index ℂ := complexMatrix exactProjector
noncomputable def D : Matrix Index Index ℂ :=
  Matrix.diagonal (fun i => if Nat.gcd i.val 12 = 1 then 1 else 0)
noncomputable def psi : Index → ℂ := complexVector exactPsi
noncomputable def chi : Index → ℂ := complexVector exactChi
noncomputable def unitState : Index → ℂ := complexVector exactUnitState
```

Consequently, the finite certificates produce actual complex unitarity and projector transport. The following excerpt contains the full proofs of both unitary identities, the cross-representation projector identity, and the two state-action identities.

Literal source. `SemanticBridge.lean`, physical lines 93–120.

```
/-- Actual complex unitarity, not an assertion only about the coordinate checker. -/
theorem fourier_unitary_left : Fᴴ * F = 1 := by
  have h : kMatMul (kAdjoint exactFourier) exactFourier = kIdentity :=
    funext (fun i => funext (normalized_fourier_left i))
  simpa [F] using congrArg complexMatrix h

theorem fourier_unitary_right : F * Fᴴ = 1 := by
  have h : kMatMul exactFourier (kAdjoint exactFourier) = kIdentity :=
    funext (fun i => funext (normalized_fourier_right i))
```

```
  simpa [F] using congrArg complexMatrix h

theorem projector_transport : F * D * Fᴴ = P := by
  ext i j
  have h := congrArg interpret (exact_projector_transport i j)
  simp only [selectedKernel, interpret_kSum] at h
  rw [Matrix.mul_apply]
  simpa [D, P, F, complexMatrix, Matrix.mul_diagonal, Matrix.conjTranspose_apply,
    apply_ite interpret, mul_ite] using h

theorem state_preparation : F.mulVec unitState = psi := by
  have h : kMatVec exactFourier exactUnitState = exactPsi :=
    funext exact_state_preparation
  simpa [F, unitState, psi] using congrArg complexVector h

theorem state_recovery : Fᴴ.mulVec psi = unitState := by
  have h : kMatVec (kAdjoint exactFourier) exactPsi = exactUnitState :=
    funext exact_state_recovery
  simpa [F, unitState, psi] using congrArg complexVector h
```

This is the key strengthening over the earlier checkpoint. The rational matrix, the exact coordinate kernel, and the complex matrix FDF^* are explicitly connected. State preparation and recovery are equalities of actual complex vectors. The theorem name `state_preparation` describes a matrix identity, not an efficient hardware preparation algorithm.

A.10 The exponential sign and square-root normalization

The usual Fourier matrix is not left as an informal interpretation. The convention module proves that the coordinate root equals the positive exponential root, including the selected square-root branch.

Literal source. `Conventions.lean`, physical lines 10–24.

```
/-- The coordinate root is exactly the usual exponential twelfth root. -/
theorem root_is_exponential :
    interpret zeta12 = Complex.exp (((Real.pi / 6 : ℝ) : ℂ) * Complex.I) := by
  rw [Complex.exp_mul_I]
  rw [← Complex.ofReal_cos, ← Complex.ofReal_sin, Real.cos_pi_div_six, Real.sin_pi_div_six]
  apply Complex.ext <;> simp [interpret, zeta12]
  ring

theorem root_power_twelve : interpret zeta12 ^ 12 = 1 := by
  simpa using congrArg interpret zeta12_order_certificate

theorem rootPower_interpret (n : Nat) :
    interpret (rootPower n) = interpret zeta12 ^ n := by
  simp only [rootPower, interpret_pow]
  exact (pow_eq_pow_mod n root_power_twelve).symm
```

Its normalization lemmas prove $\sqrt{12} = 2\sqrt{3}$ and $\sqrt{48} = 4\sqrt{3}$, then identify the exact scales with their ordinary reciprocals. The matrix entry formula and state formula below are therefore literal equalities in $\mathbb{C}$.

Literal source. `Conventions.lean`, physical lines 42–54.

```
theorem fourierScale_interpret :
    interpret fourierScale = ((1 / Real.sqrt 12 : ℝ) : ℂ) := by
  apply Complex.ext <;> simp [interpret, fourierScale, sqrt_twelve]
  have h : Real.sqrt 3 ≠ 0 := by positivity
  field_simp
  nlinarith [Real.sq_sqrt (show (0 : ℝ) ≤ 3 by norm_num)]

theorem stateScale_interpret :
    interpret stateScale = ((1 / Real.sqrt 48 : ℝ) : ℂ) := by
  apply Complex.ext <;> simp [interpret, stateScale, sqrt_fortyeight]
  have h : Real.sqrt 3 ≠ 0 := by positivity
  field_simp
  nlinarith [Real.sq_sqrt (show (0 : ℝ) ≤ 3 by norm_num)]
```

Literal source. `Conventions.lean`, physical lines 56–65.

```
/-- The matrix checked in the certificate is the standard positive sign Fourier matrix. -/
theorem fourier_entry_formula (i j : Index) :
    F i j = ((1 / Real.sqrt 12 : ℝ) : ℂ) *
      Complex.exp (2 * (Real.pi : ℂ) * Complex.I * (i.val : ℂ) * (j.val : ℂ) / 12) := by
  simp only [F, complexMatrix, exactFourier, interpret_mul, fourierScale_interpret,
    rootPower_interpret, root_is_exponential]
  rw [← Complex.exp_nat_mul]
  congr 2
  push_cast
  ring
```

Literal source. `Conventions.lean`, physical lines 67–82.

```
theorem psi_entry_formula (i : Index) :
    psi i = ((1 / Real.sqrt 48 : ℝ) : ℂ) * (ramanujan12.getD i.val 0 : ℂ) := by
  simp [psi, complexVector, exactPsi, stateScale_interpret]

theorem chi_entry_formula (i : Index) :
    chi i = ((1 / Real.sqrt 48 : ℝ) : ℂ) * (absRamanujan12.getD i.val 0 : ℂ) := by
  simp [chi, complexVector, exactChi, stateScale_interpret]

theorem unitState_entry_formula (i : Index) :
    unitState i = if Nat.gcd i.val 12 = 1 then (1/2 : ℂ) else 0 := by
  by_cases h : Nat.gcd i.val 12 = 1 <;>
    simp [unitState, complexVector, exactUnitState, h]

theorem projector_entry_formula (i j : Index) :
    P i j = (ramanujan12.getD ((i.val + 12 - j.val) % 12) 0 : ℂ) / 12 := by
  simp [P, complexMatrix, exactProjector]
```

The convention is positive sign for F and negative sign for F^*. The earlier unnormalized `dft12` uses the negative sign. The normalization and adjoint are the precise link between those notations.

A.11 Hermitian projection and phase-sensitive states

The selector is Hermitian and idempotent. Transport through a unitary matrix then proves the corresponding facts for P. They are ordinary matrix identities, not claims that every rational list operation is an orthogonal projection.

Literal source. `Conventions.lean`, physical lines 84–108.

```
theorem selector_idempotent : D * D = D := by
  unfold D
  rw [Matrix.diagonal_mul_diagonal]
  congr 1
  funext i
  by_cases h : Nat.gcd i.val 12 = 1 <;> simp [h]

theorem selector_selfAdjoint : Dᴴ = D := by
  ext i j
  by_cases h : i = j
  · subst j
    simp [D, Matrix.conjTranspose_apply, apply_ite star]
  · simp [D, Matrix.conjTranspose_apply, h, Ne.symm h]

theorem projector_idempotent_complex : P * P = P := by
  rw [← projector_transport]
  calc
    (F * D * Fᴴ) * (F * D * Fᴴ) = F * D * (Fᴴ * F) * D * Fᴴ := by
      simp only [Matrix.mul_assoc]
    _ = F * D * Fᴴ := by rw [fourier_unitary_left, Matrix.mul_one, Matrix.mul_assoc F D D,
      selector_idempotent]

theorem projector_selfAdjoint_complex : Pᴴ = P := by
  rw [← projector_transport]
  simp [Matrix.conjTranspose_mul, selector_selfAdjoint, Matrix.mul_assoc]
```

The vectors ψ and χ are separately defined states. One is the signed arithmetic amplitude vector and the other its entrywise absolute value. They have the same squared amplitudes at each initial index but retain different relative phase. The development neither claims nor requires a universal quantum operation implementing the nonlinear map $v \mapsto |v|$.

Literal source. `SemanticBridge.lean`, physical lines 145–174.

```
theorem psi_normalized : totalProbability psi = 1 := by
  have h := congrArg interpret exact_psi_normalized
  simpa [psi] using congrArg Complex.re h

theorem chi_normalized : totalProbability chi = 1 := by
  have h := congrArg interpret exact_chi_normalized
  simpa [chi] using congrArg Complex.re h

theorem unitState_normalized : totalProbability unitState = 1 := by
  have h := congrArg interpret exact_unit_normalized
  simpa [unitState] using congrArg Complex.re h

theorem same_initial_probabilities (i : Index) :
    Complex.normSq (psi i) = Complex.normSq (chi i) := by
  have h := congrArg interpret (exact_same_initial_probabilities i)
  simpa [psi, chi, complexVector] using congrArg Complex.re h

theorem states_orthogonal : ∑ i, star (psi i) * chi i = 0 := by
  have h := congrArg interpret exact_orthogonal_states
  simpa [psi, chi, complexVector] using h

theorem signed_state_recovers_arithmetic :
```

```
    coprimeProbability (Fᴴ.mulVec psi) = 1 := by
  have h := congrArg interpret exact_coprime_mass_psi
  simpa [F, psi] using congrArg Complex.re h

theorem unsigned_state_erases_arithmetic :
    coprimeProbability (Fᴴ.mulVec chi) = 0 := by
  have h := congrArg interpret exact_coprime_mass_chi
  simpa [F, chi] using congrArg Complex.re h
```

The transformed probability statement is exact. The measurement event is the coprime-frequency subset. After applying F^*, its probability is one for ψ and zero for χ. This basis change reveals information that their shared initial histogram does not contain.

A.12 Two exact information-loss statements

The formal development isolates distinct obstructions. First, the initial histogram cannot be inverted on both states by any reconstruction function. Second, the projector itself is noninjective because it annihilates the nonzero normalized state χ.

Literal source. `Publication.lean`, physical lines 39–50.

```
/-- No state reconstruction function can invert the initial probability histogram on both states. -/
theorem no_reconstruction_from_initial_probabilities :
    ¬ ∃ R : (Index → ℝ) → (Index → ℂ),
      R (fun i => Complex.normSq (psi i)) = psi ∧
      R (fun i => Complex.normSq (chi i)) = chi := by
  rintro ⟨R, hpsi, hchi⟩
  have heq : (fun i => Complex.normSq (psi i)) =
      (fun i => Complex.normSq (chi i)) := funext same_initial_probabilities
  have hstate : psi = chi := by rw [← hpsi, ← hchi, heq]
  have h := signed_state_recovers_arithmetic
  rw [hstate, unsigned_state_erases_arithmetic] at h
  norm_num at h
```

Literal source. `Publication.lean`, physical lines 52–57.

```
/-- Exact probabilities in the recovered basis, connected to the normalized state. -/
theorem recovered_probability_at (i : Index) :
    Complex.normSq (Fᴴ.mulVec psi i) =
      if Nat.gcd i.val 12 = 1 then (1/4 : ℝ) else 0 := by
  rw [state_recovery, unitState_entry_formula]
  by_cases h : Nat.gcd i.val 12 = 1 <;> norm_num [h, Complex.normSq_apply]
```

Literal source. `Publication.lean`, physical lines 59–65.

```
/-- The operator is not injective. Its nonzero normalized kernel state is explicit. -/
theorem projector_not_injective : ¬ Function.Injective P.mulVec := by
  intro h
  have hz : chi = 0 := h (by simp [projector_kills_chi])
  have hp := chi_normalized
  rw [hz] at hp
  norm_num [totalProbability] at hp
```

The nonreconstruction theorem quantifies over every function from real-valued histograms to complex-valued vectors. It uses equality of the two histograms and the incompatible transformed probabilities. This is a mathematical obstruction to a specified inverse, not a statistical claim about estimating a state from many measurements in several bases.

The traditional paper also proves that the two states cannot both be recovered up to an overall phase from this histogram. That strengthening is a prose corollary of their orthogonality and normalization. It is not separately exported as a quotient-by-global-phase Lean theorem.

Projection becomes reconstructive when its residual is retained. The next identity holds for every twelve-dimensional complex vector, including vectors outside the arithmetic signal subspace.

Literal source. `Publication.lean`, physical lines 33–37.

```
/-- Every signal is reconstructible from the retained and discarded components together. -/
theorem projector_residual_reconstructs (v : Index → ℂ) :
    P.mulVec v + (1 - P).mulVec v = v := by
  rw [Matrix.sub_mulVec, Matrix.one_mulVec]
  exact add_sub_cancel _ _
```

Its proof is algebraic. The identity $Pv + (I - P)v = v$ itself does not require idempotence. Hermitian idempotence supplies the stronger interpretation as orthogonal retained and discarded components. This separates the reconstruction identity from the additional geometric structure.

A.13 The publication theorem in full

The final declaration bundles the formal core of the worked example. Every conjunct is supported by a previously proved declaration rather than by a new numerical test or an implicit physical assumption.

Literal source. `Publication.lean`, physical lines 67–80.

```
/-- Publication theorem gathering the complex identities stated in the worked example. -/
theorem publication_bridge :
    Fᴴ * F = 1 ∧ F * Fᴴ = 1 ∧ F * D * Fᴴ = P ∧
    P * P = P ∧ Pᴴ = P ∧ F.mulVec unitState = psi ∧
    Fᴴ.mulVec psi = unitState ∧ totalProbability psi = 1 ∧
    totalProbability chi = 1 ∧
    (∀ i, Complex.normSq (psi i) = Complex.normSq (chi i)) ∧
    (∑ i, star (psi i) * chi i) = 0 ∧
    coprimeProbability (Fᴴ.mulVec psi) = 1 ∧
    coprimeProbability (Fᴴ.mulVec chi) = 0 := by
  exact (fourier_unitary_left, fourier_unitary_right, projector_transport,
    projector_idempotent_complex, projector_selfAdjoint_complex, state_preparation,
    state_recovery, psi_normalized, chi_normalized, same_initial_probabilities,
    states_orthogonal, signed_state_recovers_arithmetic, unsigned_state_erases_arithmetic)
```

The bundled theorem has a fixed dimension and explicitly named operators and states. It establishes the bridge used in the paper. Broader arbitrary-modulus results remain separate prose theorems, with their own proofs and evidence.

The manuscript's QA18 additionally includes filtering and energy, normalization of u, the projector actions on both states, and noninjectivity. Those clauses use separately exported declarations, including `projector12_recovers_test_signal`, `energy_decomposition`, `unitState_normalized`, `projector_fixes_psi`, `projector_kills_chi`, and `projector_not_injective`. The complete machine concordance lists these supporting results. The manuscript bundle is not identified with a literal statement that omits its extra clauses.

A.14 Manuscript-to-Lean concordance

The main paper labels its results QA01 through QA20. The table below distinguishes the exact fixed-example formal coverage from the more general mathematical development. "Not formalized here" is a scope label, not a claim that the prose result is false.

Claim	Subject	Formal coverage
QA01	Primitive root	Exact order 12 and standard exponential root.
QA02	Character orthogonality	Fixed-example content through both exact unitary identities; general theorem not separately exported.
QA03	Fourier theory	Both ordinary complex unitary identities and standard entry formula; general Parseval theorem not separately exported.
QA04	Möbius formula	General modulus formula not formalized; exact integer table for 12 certified.
QA05	Ramanujan support	Exact signed and unsigned twelve-frequency DFT tables and arithmetic energy 48.
QA06	Divisor projectors	Primitive-order projector at 12 certified; the full divisor family not a general Lean family.
QA07	Projector geometry	Hermitian idempotence, displayed energies, explicit noninjectivity, and all-vector residual reconstruction.
QA08	Residue recombination	No theorem in this checkpoint.
QA09	Polyphase aliasing	No theorem in this checkpoint.
QA10	Arithmetic example	Phase sum connected to the exact rational sample table.

Claim	Subject	Formal coverage
QA11	Exact filtering	Displayed contaminated signal and orthogonal energy decomposition.
QA12	State preparation	Ordinary complex normalized vector actions and recovery.
QA13	Shared histogram	Entrywise probability equality and state orthogonality.
QA14	Phase separation	Actual transformed coprime probabilities 1 versus 0.
QA15	Reconstruction barrier	No vector-valued initial-histogram inverse works on both specified states; the global-phase corollary is prose only.
QA16	Measurement law	Relevant projector algebra and state probabilities; no general conditional-state-update API.
QA17	Arithmetic semantics	Soundness for scalar, matrix, vector, adjoint, and mass operations.
QA18	Central bridge	Publication bundle plus separately exported filter, energy, normalization, projector-action, and noninjectivity results.
QA19	Relative-phase family	Prose extension $\sin^2(\theta/2)$; not an inherited Lean theorem.
QA20	Dephased probability	Prose density-matrix calculation $1/3$; not an inherited Lean theorem.

The machine concordance includes all theorem statements verbatim, per-theorem axiom records, source hashes, and exact physical line locators. Its mapping records formal scope; it does not upgrade general manuscript claims merely because a nearby finite special case is checked.

A.15 Trust boundary and current verification

The current replay uses Lean 4.34.0-rc2, compiler commit `6a10ac8c22beadecabdbb0919c2b50214762f91d`, and Mathlib revision `aeb72f1e9981b6d35b426b16dca7759be3e7ba06`. It built the five source modules with warnings treated as errors, printed axiom dependencies for all 96 named theorems, and replayed each module with `leanchecker`. The compiler and imported dependency cache were not rebuilt from their source. Kernel replay uses Lean's own kernel, not a second independent proof checker.

The axiom union is exactly `propext`, `Classical.choice`, and `Quot.sound`. This is the standard foundation used by the audited proof terms, not an axiom-free claim. The active source contains no proof placeholders, custom axiom declarations, native decision proofs, or unsafe declarations. Historical failed or native-decision drafts are excluded from the active build.

Current source reconciliation checked all five source hashes and all eleven archived check-log hashes against the archived report. All sixteen match. The current inventory contains exactly 96 unique names and 96 archived axiom records. All line numbers agree with the archived source index.

The pinned portable runtime was restored and verified against its combined archive hash. A fresh build, all 96 theorem axiom inspections, and all five kernel replays passed. The complete receipt and logs accompany the source package. Earlier transfer failures remain historical records.

A.16 Controls, failed attempts, and reproducibility

The latest archive rejected a deliberately false signal sign and a proof using `sorry` under warnings as errors. Its false Fourier-mode control also failed, but the compiler stopped at opaque `Decidable` reduction. That failure alone is not a direct mathematical certificate that the proposition is false. The positive `ramanujan_dft_table` theorem separately establishes the correct value 12.

The control runner's broad diagnostic expression can match the word "false" in a filename. The guide does not interpret that match as a stronger arithmetic test than the actual log provides. The earlier checkpoint's four explicit refutation proofs are retained as earlier evidence and are not silently counted as current production theorems.

A supplementary current check closes the diagnostic gap for the false-mode control. It derives the exact mode-one value and its negation from the proved Fourier table, replays the supplementary module through the kernel, and rejects a false rational-coordinate equality with an explicit false-value diagnostic. These two verification lemmas do not increase the monograph count of 96 theorems.

The supplied verifier creates a fresh source project, checks dependency revisions, builds, prints all theorem axiom dependencies, replays all five modules, and runs the controls. It refuses to overwrite an existing fresh build directory and treats timeouts as failures. It makes no network requests or installations. With the pinned runtime restored, the documented command is:

```
python scripts/verify_lean.py \
  --toolchain /path/to/Komorebi_Portable_Lean_4.34.0-rc2/toolchain \
  --packages /path/to/Komorebi_Portable_Lean_4.34.0-rc2/workspace/.lake/packages \
  --out verification/reproduction_new
```

The runner was exercised under Linux x86-64. It uses POSIX process groups and filesystem symlinks. Native Windows behavior is not tested. WSL2 with the pinned Linux runtime is the documented route for a Windows machine.

Full proof source, manifests, archived build and replay logs, the static concordance builder, and the current successful replay receipt are included. The runtime itself remains an external dependency.

A.17 Complete active theorem inventory

All names in the following table have prefix `Komorebi.QuantumArithmetic`. File and physical source line identify the exact declaration. The full literal statements and axiom lists are also available in `Lean_Concordance.json` and `Lean_Concordance.csv`.

A.17.1 `QuantumArithmeticBridge`: 22 theorems

Declaration	Line
`ramanujan12_length`	80
`projector12_shape`	82
`projector12_symmetric`	88
`projector12_idempotent`	92
`projector12_fixes_ramanujan`	97
`projector12_kills_noise`	102
`projector12_recovers_test_signal`	107
`ramanujan12_energy`	112
`noise12_energy`	116
`testSignal12_energy`	120
`energy_decomposition`	124
`zeta12_order_certificate`	177
`zeta12_inverse_certificate`	180
`ramanujanC12_table`	190

Declaration	Line
ramanujan_dft_table	205
abs_ramanujan_dft_table	217
same_computational_basis_probabilities	231
abs_ramanujan12_energy	235
recovered_mode_probability	243
recovered_total_probability	248
phase_stripping_coprime_probability	253
arithmetic_signal_quantum_bridge	258

A.17.2 `FiniteFourier`: 17 theorems

Declaration	Line
coprime_residues_exact	59
root_exact_order	63
normalized_fourier_left	67
normalized_fourier_right	71
exact_projector_transport	79
exact_projector_matches_rational_table	83
exact_state_preparation	87
exact_state_recovery	91
exact_unit_normalized	95
exact_psi_normalized	98
exact_chi_normalized	101
exact_same_initial_probabilities	104
exact_orthogonal_states	108
exact_coprime_mass_psi	112
exact_coprime_mass_chi	116
exact_projector_fixes_psi	120
exact_projector_kills_chi	124

A.17.3 `SemanticBridge`: 30 theorems

Declaration	Line
interpret_zero	19
interpret_one	22
interpret_rat	25
interpret_add	28
interpret_mul	32
interpret_conj	37
interpret_pow	42
interpret_sum	48
interpret_kSum	54
complexMatrix_mul	65
complexMatrix_adjoint	70
complexMatrix_identity	75

Declaration	Line
complexVector_mul	80
fourier_unitary_left	94
fourier_unitary_right	99
projector_transport	104
state_preparation	112
state_recovery	117
interpret_normSq	122
interpret_mass	133
interpret_coprimeMass	137
psi_normalized	145
chi_normalized	149
unitState_normalized	153
same_initial_probabilities	157
states_orthogonal	162
signed_state_recovers_arithmetic	166
unsigned_state_erases_arithmetic	171
projector_fixes_psi	176
projector_kills_chi	181

A.17.4 Conventions: 17 theorems

Declaration	Line
root_is_exponential	11
root_power_twelve	18
rootPower_interpret	21
sqrt_twelve	26
sqrt_fortyeight	34
fourierScale_interpret	42
stateScale_interpret	49
fourier_entry_formula	57
psi_entry_formula	67
chi_entry_formula	71
unitState_entry_formula	75
projector_entry_formula	80
selector_idempotent	84
selector_selfAdjoint	91
projector_idempotent_complex	98
projector_selfAdjoint_complex	106
complete_phase_separation	111

A.17.5 `Publication`: 10 theorems

Declaration	Line
`exact_phase_signal_lookup`	9
`complex_ramanujan_signal`	14
`absolute_signal_lookup`	22
`retained_discarded_orthogonal`	27
`signal_decomposition`	30
`projector_residual_reconstructs`	34
`no_reconstruction_from_initial_probabilities`	40
`recovered_probability_at`	53
`projector_not_injective`	60
`publication_bridge`	68

A.18 What completion means here

The fixed twelve-dimensional bridge is formally covered in the preserved later checkpoint. This guide and machine concordance expose its proof and trust boundaries.

General-modulus Möbius and divisor-projector theory, polyphase identities, and a general measurement API remain outside this Lean package. The traditional paper supplies separate proofs. The current pinned replay passed for the fixed formal package.

A.19 The original finite bridge and its extension

A.20 Formal coverage, lineage, and reproducibility

A.20.1 Two inherited checkpoints and the expanded boundary

The earlier checked package has four mathematical production modules and fifty-nine named theorem declarations. Its first module preserves twenty-two original statements. Its proof-boundary document correctly says that explicit matrix transport between independently defined representations and normalized state-action identities are not supplied as additional theorems in that package. Those limitations must not be erased when describing the earlier evidence.

The later checked package has five production modules and ninety-six named theorem declarations:

Module	Named theorems	Principal role
`QuantumArithmeticBridge`	22	Literal rational projector and cyclotomic tables
`FiniteFourier`	17	Indexed exact Fourier and normalized state identities
`SemanticBridge`	30	Sound complex interpretation and transported identities
`Conventions`	17	Exponential entries, scales, and complex projector laws
`Publication`	10	Phase lookup, residual reconstruction, nonreconstruction, and central bundle
Total	96	Finite modulus-twelve proof chain, including helpers

The later source explicitly supplies $FDF^{\dagger} = P$, both normalized state-action identities, the conventional exponential formula for F, identical initial probabilities, orthogonality, exact selected probability one versus zero, histogram nonreconstruction, and projector noninjectivity. It therefore closes the representation-transport obligations that the earlier package leaves outside its formal boundary.

The ninety-six count includes helper lemmas and bundled propositions. It is not a count of ninety-six independent discoveries. Nor should the earlier fifty-nine and later ninety-six counts be added as if their shared declarations were disjoint.

A.20.2 Current and inherited checks

This edition preserves the source, theorem index, verification records, and compiled-artifact lineage of the checked checkpoints. The archived records provide the earlier compiler and kernel-checking evidence. Current static audits and independent Python replay provide current bounded evidence about the downloaded files and the example implementation. These are disclosed separately in the release records.

The pinned runtime was restored for this completion pass. Fresh source compilation, axiom inspection, and replay of all five modules passed. The separate historical failures are preserved in the source records.

The exact source uses kernel-reduction decision proofs for finite identities and ordinary proof terms for the transport lemmas. The declaration audit rejects admitted proofs and unapproved project-specific axioms. Standard foundational dependencies are reported in the formal guide. Compiled artifact checking reuses Lean's kernel; it is not an independently implemented proof assistant.

The later checkpoint preserves three negative controls with distinct meanings. The false-sign claim is a directly false finite statement. The false-mode case is an operational rejection whose failure mechanism must be read alongside its diagnostic and the positive checked transform table. The admitted-proof case tests the verification policy for incomplete proof evidence. These controls are not combined with the four controls of the earlier checkpoint into an artificial total.

A.20.3 Independent arithmetic evidence

The original reproducibility package records 3,003 exact Python assertions, separately from floating-point checks. Its exact verifier represents the twelfth root in a cyclotomic quotient, checks the signal and transform tables, the rational projector identities, the test-signal decomposition, divisor-shell completeness, and the phase-sensitive example. Its arithmetic regression range is bounded to moduli two through sixty-four.

The release also includes the current replay and any added finite checks with their own records. Exact rational or cyclotomic arithmetic and numerical residuals are reported separately. A small floating-point residual verifies the implementation at its declared tolerance; it does not replace the exact proof of the general formula. Likewise, a finite modulus sweep does not prove the arbitrary-modulus theorem. QA01 through QA09 provide the general proofs needed here.

A.20.4 The twenty-two preserved original statements

The original bridge module's statements remain available under their original names. The following list records their substantive content rather than treating each source helper as a separate main theorem.

Original declaration	Preserved statement
`ramanujan12_length`	The rational signal list has length twelve
`projector12_shape`	The literal matrix has twelve rows and twelve columns
`projector12_symmetric`	The rational matrix equals its transpose
`projector12_idempotent`	Exact rational multiplication gives $P^2 = P$
`projector12_fixes_ramanujan`	The matrix fixes the signal c
`projector12_kills_noise`	The matrix sends the displayed perturbation b to zero
`projector12_recovers_test_signal`	Applying it to x returns c
`ramanujan12_energy`	$\|c\|^2 = 48$
`noise12_energy`	$\|b\|^2 = 102$
`testSignal12_energy`	$\|x\|^2 = 150$
`energy_decomposition`	The exact energy balance is $150 = 48 + 102$
`zeta12_order_certificate`	The coordinate root raised to the twelfth power is one
`zeta12_inverse_certificate`	The coordinate root times its conjugate is one
`ramanujanC12_table`	Four primitive phase sums give the exact twelve-entry signal table
`ramanujan_dft_table`	The negative-sign DFT gives twelve exactly at $1, 5, 7, 11$
`abs_ramanujan_dft_table`	The unsigned DFT gives the exact even-support table
`same_computational_basis_probabilities`	Signed and unsigned rational entries have the same squares
`abs_ramanujan12_energy`	The unsigned signal also has squared energy forty-eight
`recovered_mode_probability`	The scalar normalization gives $12^2/(48 \cdot 12) = 1/4$
`recovered_total_probability`	Four recovered mode probabilities sum to one
`phase_stripping_coprime_probability`	Zero transformed coefficients give zero selected mass
`arithmetic_signal_quantum_bridge`	The original finite projector, recovery, DFT, and equal-square claims are bundled

The later source strengthens the specification around these statements. It proves that the four labels are exactly the coprime residues, that the coordinate root has no smaller positive order, that the literal rational and indexed matrices agree, and that the probability consequences belong to the actual normalized complex state vectors.

The list operations in the original module use explicit finite literal lists. Their generic helper definitions are not an assertion of arbitrary-dimensional linear algebra for unsized lists, where zip operations could truncate. General vector identities in this chapter use fixed finite index sets, and the later matrix source uses `Fin 12` for exactly that reason.

APPENDIX B

Symmetry: supplementary proofs and computation

B.1 Proof details and auxiliary derivations

B.1.1 Residue projector formula

Let $Q|\psi_k\rangle = k|\psi_k\rangle$. Then

$$\frac{1}{M}\sum_{\ell=0}^{M-1}\omega_M^{-r\ell}e^{2\pi i\ell Q/M}|\psi_k\rangle = \left(\frac{1}{M}\sum_{\ell=0}^{M-1}\omega_M^{(k-r)\ell}\right)|\psi_k\rangle.$$

The bracket equals one when $k \equiv r \pmod{M}$ and zero otherwise. This proves Equation (3.1) and the orthogonality of the residue packets.

B.1.2 Circuit leakage telescoping

Let $V_j = PG_jP$. Insert and subtract one factor at a time:

$$G_m\cdots G_1P - V_m\cdots V_1P = \sum_{k=1}^{m} G_m\cdots G_{k+1}(G_k - V_k)V_{k-1}\cdots V_1P.$$

Every G_j has norm one and every V_j has norm at most one. On $P\mathcal{H}$,

$$(G_k - V_k)P = (I - P)G_kP.$$

Taking norms gives Proposition 3.10.

B.1.3 Hypercube quotient coupling

Let

$$|k\rangle = \binom{n}{k}^{-1/2}\sum_{|x|=k}|x\rangle.$$

Every weight-k string has $n-k$ neighbors of weight $k+1$, while every weight-$(k+1)$ string has $k+1$ neighbors of weight k. Counting the bipartite edges yields

$$\langle k+1|A|k\rangle = \frac{\binom{n}{k}(n-k)}{\sqrt{\binom{n}{k}\binom{n}{k+1}}} = \sqrt{(k+1)(n-k)},$$

proving Equation (4.10).

B.1.4 Symmetric-subspace dimension

A symmetric basis state of n qudits is indexed by occupation numbers

$$(n_1, \dots, n_d), \qquad n_1 + \cdots + n_d = n.$$

The stars-and-bars count is

$$\binom{n+d-1}{d-1},$$

proving Equation (4.1).

B.1.5 Bosonic and fermionic dimensions

Bosonic occupations are weak compositions of N into m parts, giving $\binom{N+m-1}{N}$. Fermionic occupations are N-element subsets of m modes, giving $\binom{m}{N}$.

B.1.6 Acceptance-probability stability

If normalized states $|\psi\rangle, |\phi\rangle$ satisfy

$$\||\psi\rangle - |\phi\rangle\| \le \varepsilon$$

and $0 \le M \le I$, then

$$|\langle\psi|M|\psi\rangle - \langle\phi|M|\phi\rangle| \le 2\varepsilon.$$

Indeed, add and subtract $\langle\phi|M|\psi\rangle$ and use Cauchy–Schwarz. This converts state-vector leakage bounds into measurement-error bounds.

B.1.7 Orbit quotient invariance

For an automorphism group G, the adjacency matrix commutes with the permutation representation. The fixed subspace of the action is spanned by normalized orbit states. Therefore adjacency preserves that span. Polynomial-time simulation additionally requires direct computation of the quotient entries.

B.2 Python API and reproduction commands

The release code is intentionally compact. Its purpose is to verify the algebra and provide a reusable seed rather than to compete with optimized quantum simulators.

B.2.1 Core API

Listing B.1. Core packet API.

```
from symmetry_packets import (
   weight_basis,
   residue_basis,
   symmetric_subspace_dimension,
   boson_fixed_number_dimension,
   fermion_fixed_number_dimension,
   PacketPlan,
   rz,
```

```
   xy_exchange,
   hypercube_quotient,
   circuit_leakage_bound,
   hamiltonian_leakage_bound,
)
```

B.2.2 Exact packet plan

Listing B.2. Fixed-weight packet compilation.

```
plan = PacketPlan.fixed_weight(n=10, k=2)
packet_state = random_packet_state(plan, seed=12)
full_state = plan.embed(packet_state)

restricted_gate = plan.restrict_gate(
   xy_exchange(theta=0.2),
   qubits=(3, 4),
)
packet_state = restricted_gate @ packet_state
```

B.2.3 Validation

From the project root, run

```
PYTHONPATH=code python3 code/validate.py
PYTHONPATH=code python3 code/validate_exact.py
PYTHONPATH=code python3 code/validate_boundaries.py
```

The expected output reports successful formula checks and reconstruction errors near machine precision for the representative fixed-weight circuits.

B.2.4 Figure generation

All figures in the PDF are generated by

```
python3 code/make_figures.py
```

The generator writes vector PDF figures into the `figures` directory.

B.2.5 Build

The complete rebuild command is

```
./run_all.sh
```

It validates the Python code, regenerates figures, compiles the LaTeX source, and updates the checksums.

B.3 Benchmark data ledger

B.3.1 Exact weight two

The JSON file `data/benchmark_records.json` contains the values in Table 4.1, together with the parity, half-filled, and leakage records.

B.3.2 Half filling

Table B.1. Recorded half-filled benchmark.

n	k	full dim.	sector dim.	apply gain	one-shot gain	memory gain
8	4	256	70	1.212	1.083	3.657
10	5	1,024	252	1.234	0.7513	4.063
12	6	4,096	924	2.083	0.5092	4.433
14	7	16,384	3,432	3.24	0.2092	4.774
16	8	65,536	12,870	7.041	1.501	5.092
18	9	262,144	48,620	7.757	0.7458	5.392
20	10	1,048,576	184,756	7.396	0.6529	5.675

All recorded relative errors were zero at the displayed precision. The one-shot result can be below one because compilation dominates a single application.

B.3.3 Parity

Table B.2. Recorded even-parity benchmark.

n	full dim.	packet dim.	apply gain	one-shot gain	memory gain
8	256	128	1.06	1.19	2.0
10	1,024	512	1.15	1.33	2.0
12	4,096	2,048	1.49	1.37	2.0
14	16,384	8,192	1.78	1.75	2.0
16	65,536	32,768	2.0	3.7	2.0

B.3.4 Symmetry-breaking controls

Test	retained probability	leakage
Break exact weight with pair gate	0.891	0.109
Break parity with X rotation	0.869	0.131
Break weight modulo four with pair gate	0.928	0.0723

These controls demonstrate that a packet restriction must be audited after every symmetry-breaking operation.

B.4 Theorem and evidence-status ledger

Table B.3. Claim-status ledger.

Statement	Status	Qualification
Invariant-basis simulation	Proved	Exact gatewise subspace invariance.
Polynomial packet simulation	Proved	Requires efficient discovery, updates, measurement, and polynomial representation size.

Statement	Status	Qualification
Fixed k number-preserving circuits lie in P	Proved corollary	k fixed and local restricted gates efficiently applicable.
Approximate circuit leakage bounded by sum of gate leakages	Proved	Operator-norm state-error certificate.
Hamiltonian leakage bounded by $\|t\|\|[H, P]\|\|$	Proved	Bounded self-adjoint Hamiltonian.
Permutation-symmetric qubits lie in P	Derived theorem	Gate matrices and measurement efficient in Dicke coordinates.
Fixed-d symmetric qudits lie in P	Derived theorem	Fixed local dimension and efficient occupation updates.
Polynomial-orbit quantum walks lie in P	Derived theorem	Quotient data computable from succinct input.
Fixed-particle boson and fermion circuits lie in P	Derived theorem	Particle number fixed and updates efficient.
Compressed-packet theorem	Proved abstraction	Representation class closed under gates with polynomial size.
Weight-two runtime and memory gains	Recorded computation	Implementation-specific archival benchmark.
Half-filling remains exponential	Proved dimension statement	Runtime constants remain implementation dependent.
Parity gives only a factor-two dimension reduction	Proved dimension statement	Does not preclude further internal compression.
Quantum Latin order-six product realization absent, entangled realization present	External theorem	Contextual existence defect, not a simulation theorem.
Adaptive discarded-state error certificate	Proved	Unnormalized trajectories, explicitly accounted discarded norms.
Universal polynomial packetization	Open conjecture	Would imply $P = BQP$.
Representation-independent interference lower bound	Open	Required for a full hard-side complexity conclusion.

B.4.1 Publication discipline

Every extracted theorem paper should declare the circuit model, input encoding, gate precision, measurement, symmetry discovery method, packet representation, error norm, and complexity parameter. Benchmarks should be accompanied by code and frozen data. Open conjectures should not be phrased as consequences of finite experiments.

APPENDIX C

Gate matrices, census data, and open obligations

C.1 Gate matrices and symbolic conventions

C.1.1 Row-transvection convention

Matrices act on column coordinate vectors, while row operations left multiply the current matrix. A row transvection

$$\tau_{t,s}(a) = I + aE_{t,s}$$

performs

$$R_t \leftarrow R_t + aR_s.$$

In the binary CNOT interpretation, s is the control and t is the target.

The canonical six-row target is

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

One ambient word is

$$\tau_{1,2}(1), \tau_{3,1}(1), \tau_{1,0}(1), \tau_{4,1}(1), \tau_{5,1}(1), \tau_{1,2}(-1).$$

A restricted word of length seven is

$$\tau_{1,0}(1), \tau_{2,1}(1), \tau_{3,0}(-1), \tau_{3,2}(1), \tau_{4,2}(1), \tau_{5,2}(1), \tau_{2,1}(-1).$$

C.1.2 Qudit SUM gate

On computational basis labels in $\mathbb{Z}_d$,

$$\mathrm{SUM}_{s\to t}^{(a)}|x_s, x_t\rangle = |x_s, x_t + ax_s \bmod d\rangle.$$

Its classical linear action is the transvection $I + aE_{t,s}$ over $\mathbb{Z}_d$.

C.1.3 Binary symplectic form

Pauli exponents are ordered as

$$(x_0, \ldots, x_{n-1}, z_0, \ldots, z_{n-1}).$$

The symplectic form is

$$J = \begin{pmatrix} 0 & I_n \\ I_n & 0 \end{pmatrix}$$

over $\mathbb{F}_2$. A matrix M is symplectic when

$$M^T J M = J.$$

C.1.4 Clifford generator matrices

For qubit i, the Hadamard swaps x_i and z_i:

$$H_i : e_i \leftrightarrow e_{n+i}.$$

The phase gate has

$$z_i \leftarrow z_i + x_i.$$

For CNOT with control c and target t,

$$x_t \leftarrow x_t + x_c, \qquad z_c \leftarrow z_c + z_t.$$

For CZ on i, j,

$$z_i \leftarrow z_i + x_j, \qquad z_j \leftarrow z_j + x_i.$$

SWAP exchanges both x and z coordinates of the two qubits.

The parabolic subgroup used in the census is generated by S_i, CZ_{ij}, and $\mathrm{CX}_{i\to j}$. It stabilizes the Z Lagrangian

$$L_Z = \mathrm{span}(e_n, \ldots, e_{2n-1}).$$

C.1.5 Two-qubit Clifford witness

The target symplectic matrix is

$$M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

A restricted path is

$$\mathrm{CX}_{0\to1},\ S_1,\ \mathrm{CX}_{1\to0},\ \mathrm{CX}_{0\to1},$$

with three charged entangling gates. An ambient path is

$$H_1,\ \mathrm{CX}_{1\to0},\ \mathrm{CX}_{0\to1},\ H_0,\ S_1,\ S_0,$$

with two charged entangling gates when H, S are free.

C.1.6 Three-qubit Clifford witness

The target symplectic matrix is

$$M_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

A restricted total-count path is

$$\mathrm{CX}_{0\to2},\ \mathrm{CZ}_{1,2},\ \mathrm{CX}_{2\to0},\ \mathrm{CX}_{0\to2},\ \mathrm{CX}_{0\to1},\ \mathrm{CX}_{1\to0},\ \mathrm{CX}_{0\to1},$$

while an ambient path is

$$H_1,\ \mathrm{CX}_{1\to2},\ \mathrm{CX}_{2\to1},\ \mathrm{CX}_{2\to0},\ \mathrm{CX}_{0\to2},\ H_0.$$

C.1.7 Classical reversible gates

For a bit string $x \in \mathbb{F}_2^n$:

$$X_i : x_i \leftarrow x_i + 1,$$

$$\mathrm{CX}_{c\to t} : x_t \leftarrow x_t + x_c,$$

$$\mathrm{TOF}_{a,b\to t} : x_t \leftarrow x_t + x_a x_b.$$

All other coordinates are unchanged. The first two generate $\mathrm{AGL}(n, 2)$. On three bits, all three generate the full permutation group S_8. For $n \geq 4$ without ancillas they are all even permutations, so the generated group is contained in A_{2^n} and is not the full symmetric group.

C.1.8 Bosonic symplectic embedding

For $A \in \mathrm{GL}(n, \mathbb{R})$,

$$\iota(A) = \mathrm{diag}(A, A^{-T}) \in \mathrm{Sp}(2n, \mathbb{R}).$$

This convention orders quadratures as $(q_1, \ldots, q_n, p_1, \ldots, p_n)$.

C.1.9 Pauli rotations

For a Hermitian Pauli P,

$$R_P(\theta) = e^{-i\theta P/2}.$$

Clifford conjugation gives

$$CR_P(\theta)C^\dagger = R_{CPC^\dagger}(\theta).$$

The sign of $CPC^\dagger$ can be absorbed by changing θ to $-\theta$.

C.1.10 Tensor and direct-sum conventions

For linear maps, $A \oplus B$ denotes a block-diagonal direct sum. The Kronecker product $A \otimes B$ uses lexicographic ordering. In quantum chapters, $U \otimes V$ denotes the physical tensor product on disjoint systems. These operations have different dimension and cost normalizations and are not interchanged silently.

C.2 Finite census tables and witnesses

C.2.1 Status key

Every table in this appendix is generated from exact finite output in the accompanying JSON files. "Restricted size" and "ambient size" refer to the enumerated quotient groups, not necessarily the full unitary groups including phases and translations.

C.2.2 Clifford census summary

Qubits	Metric	SWAP	Restricted size	Ambient size	diam_R	diam_A	Defects
1	total	no	2	6	1	3	0
1	entangling	no	2	6	0	0	0
2	total	no	48	720	5	8	0
2	entangling	no	48	720	3	3	4
3	total	no	10,752	1,451,520	8	12	24
3	total	yes	10,752	1,451,520	7	11	0
3	entangling	no	10,752	1,451,520	6	6	1,624

C.2.3 Clifford defect distributions

Model	Distance pair	Count
Two qubits, entangling	$3 \to 2$	4
Three qubits, total	$7 \to 6$	6
Three qubits, total	$8 \to 7$	18
Three qubits, entangling	$3 \to 2$	24
Three qubits, entangling	$4 \to 3$	432
Three qubits, entangling	$5 \to 4$	1,056
Three qubits, entangling	$6 \to 4$	48
Three qubits, entangling	$6 \to 5$	64

C.2.4 Affine-versus-Toffoli census

Bits	Metric	Restricted size	Ambient size	Restricted diameter	Defects
3	total	1,344	40,320	7	0
3	entangling	1,344	40,320	6	0

C.2.5 Linear atom modular checks

Modulus	Restricted word length	Ambient word length	Both match target
2	7	6	yes
3	7	6	yes

5	7	6	yes
8	7	6	yes

These checks confirm the symbolic words. Exact optimality over general rings is proved separately in the core chapter.

C.2.6 Core binary six-dimensional census

For $U_6(\mathbb{F}_2)$ there are $2^{15} = 32{,}768$ targets. The complete comparison is:

Restricted distance	Ambient distance	Number of targets
0	0	1
1	1	15
2	2	125
3	3	720
4	4	2,914
5	5	7,851
6	6	12,656
7	6	20
7	7	7,926
8	8	540

The twenty strict defects are exactly the analytic three-cross, one-phase motifs.

C.2.7 Endpoint Green coefficients for the canonical atom

Restricted word counts for degrees zero through ten are

$$0, 0, 0, 0, 0, 0, 0, 543, 25080, 510703, 10824940.$$

Ambient word counts for degrees zero through eight are

$$0, 0, 0, 0, 0, 0, 2, 776, 39454.$$

Thus first arrival occurs at seven and six, respectively.

C.2.8 Machine-readable locations

The release contains:

- `results/atlas_results.json`,
- `results/clifford_n3_total.json`,
- `results/clifford_n3_total_swap.json`,
- `results/clifford_n3_entangling.json`,
- the complete unitriangular supplement under `core_extract/supplement/`.

The JSON schema records model, metric, group sizes, diameters, defect distributions, and selected witnesses.

C.3 Python and C++ interface

C.3.1 Package overview

The Python package `catalytic_gate_atlas` exports exact primitives for the linear, Clifford, and classical reversible experiments. It requires Python 3.10 or later and has no runtime dependencies.

C.3.2 Installation

```
cd code
python -m venv .venv
. .venv/bin/activate
python -m pip install -e . --no-build-isolation
python -m pytest tests
```

On Windows PowerShell, activation is

```
.\.venv\Scripts\Activate.ps1
```

C.3.3 Top-level API

```
from catalytic_gate_atlas import (
   RowGate,
   canonical_ambient_word,
   canonical_restricted_word,
   canonical_six_target,
   snapshot_target,
   verify_canonical_atom,
   exact_clifford_census,
   exact_affine_toffoli_census,
   gate_word_to_openqasm,
)
```

C.3.4 Linear atom

```
from catalytic_gate_atlas import verify_canonical_atom

for modulus in (2, 3, 5, 8):
   print(verify_canonical_atom(modulus))
```

The function evaluates the displayed restricted and ambient words over $\mathbb{Z}/m\mathbb{Z}$ and compares both endpoints with the canonical target.

C.3.5 Snapshot targets

```
from catalytic_gate_atlas import snapshot_target

# u=4 source updates; snapshots at phases 0, 2, and 4.
A = snapshot_target(4, phases=(0, 2, 4), modulus=2)
for row in A:
   print(row)
```

The generator checks phase order and produces the exact lower-unitriangular target matrix used in the symbolic family.

C.3.6 Clifford census

```
from catalytic_gate_atlas import exact_clifford_census

print(exact_clifford_census(2, entangling_cost=True))
print(exact_clifford_census(2, entangling_cost=False))
```

The optional three-qubit Python census is slower. The release includes an optimized C++ program for the full $1{,}451{,}520$-state group.

C.3.7 Affine-versus-Toffoli census

```
from catalytic_gate_atlas import exact_affine_toffoli_census

print(exact_affine_toffoli_census(3, entangling_cost=False))
print(exact_affine_toffoli_census(3, entangling_cost=True))
```

The complete ambient enumeration is intentionally limited to three bits.

C.3.8 OpenQASM witness export

```
from catalytic_gate_atlas import gate_word_to_openqasm

word = ["H1", "CX12", "CX21", "CX20", "CX02", "H0"]
print(gate_word_to_openqasm(word, 3, title="ambient witness"))
```

The exporter supports the named H, S, X, CX, CZ, and SWAP gates used by the finite atlas and writes OpenQASM 3. The supplied files in `examples/` encode the two- and three-qubit restricted and ambient witness words. The Clifford census is performed modulo global phase and Pauli translations, so equality of an exported pair refers to its binary symplectic action unless the ignored Pauli and phase data are separately reconstructed.

```
PYTHONPATH=. python scripts/export_qasm_examples.py
```

C.3.9 Search primitives

```
from catalytic_gate_atlas.search import (
   NamedGenerator,
   breadth_first_distances,
   breadth_first_with_parents,
   zero_one_distances,
   zero_one_with_parents,
   dijkstra_distances,
   reconstruct_path,
   defect_distribution,
)
```

Every state must be immutable and hashable. The caller supplies a multiplication function with the convention

$$\texttt{multiply(generator, state)}.$$

C.3.10 Binary matrix primitives

The module `algebra.py` provides:

- `gf2_identity`,
- `gf2_mul`,
- `gf2_transpose`,
- `gf2_rank`,
- dense modular matrix multiplication,
- exact permutation composition and inversion.

C.3.11 C++ verifier

```
g++ -O3 -std=c++20 scripts/clifford_census.cpp -o scripts/clifford_census
./scripts/clifford_census --n 3 --metric total
./scripts/clifford_census --n 3 --metric total --swap
./scripts/clifford_census --n 3 --metric entangling
```

The executable can write JSON using the `--output` option.

C.3.12 Figure generation

```
python scripts/make_monograph_figures.py
```

All new monograph figures are generated by Matplotlib without external data. The established unitriangular figures are preserved from the Paper I source archive.

C.3.13 Adding a new finite gate model

A new module should implement:

1. a canonical immutable state representation;
2. exact multiplication;
3. named restricted and ambient generators with nonnegative costs;
4. an exact target-membership test;
5. a census function returning the standard JSON fields;
6. smoke tests for group size and witness evaluation.

C.3.14 Standard result schema

A census result should include:

```
{
  "model": "descriptive-name",
  "size": n,
  "metric": "total-or-weighted",
  "restricted_group_size": ...,
  "ambient_group_size": ...,
  "restricted_diameter": ...,
  "ambient_diameter": ...,
  "defect_count": ...,
  "defect_distribution": {...},
```

```
  "maximum_defect": ...,
  "witness": {...},
  "status": "complete finite enumeration"
}
```

A partial search must not use the status string "complete finite enumeration."

C.3.15 Error handling

Several entry points validate dimensions, distinct controls, matrix shape, and cost restrictions. The complete affine census rejects widths above three, and the C++ Clifford executable accepts only one through three qubits. Low-level helpers assume well-formed indices and states, so callers should not treat every helper as a fully validated public parser. The search code does not silently truncate its reachable state space.

C.4 Conjecture and obligation ledger

C.4.1 How to read the ledger

The ledger separates established statements, complete computations, partial checks, and open conjectures. "Open" means no proof is claimed in this monograph. "Computed" refers only to the stated finite model.

ID	Statement	Status	Decisive next step
L1	Canonical six-row transvection target has lengths 7 and 6 over nonzero commutative rings.	Proved	Independent external proof audit.
L2	Dimensions below six are rigid over every coefficient field or ring.	Open	Symbolic bounded-skeleton elimination for $n \leq 5$.
L3	Maximum unitriangular defect is $\Theta(n \log n)$.	Open	Matching upper bound or larger interacting-catalyst family.
L4	Multiplicative unitriangular distortion is bounded by $3/2$.	Open	Normal form beyond one clean carrier.
L5	Canonical atom has lower border length seven.	Open	Eliminate six-gate word-map components algebraically.
L6	A one-copy rigid target exhibits tensor activation.	Open	Exact tensor-power search or asymptotic construction.
C1	Two-qubit parabolic Clifford entangling model has four $3 \to 2$ defects in one free-phase orbit.	Analytic costs and orbit; finite classification	Complete: Theorem 10.5 plus the exhaustive census.
C2	Three-qubit swap-free total Clifford model has twenty-four defects.	Computed	Analytic motif classification.
C3	Unit-cost SWAP eliminates all three-qubit total defects.	Computed	Determine whether rigidity persists for larger n.
C4	Parabolic-versus-full Clifford defect is unbounded.	Open	Construct and certify an amplifiable atom.

N1	Three-bit affine targets receive no Toffoli-assisted saving in the two tested metrics.	Computed	Independent implementation and formal certificate.
N2	Some finite affine target is shortened by temporary nonlinearity.	Open	Complete four-bit normal-form search.
P1	A diagonal target has a Pareto improvement from non-diagonal Clifford escape.	Open	Exact small-width multiobjective atlas.
P2	Phase-polynomial T-count displays tensor activation.	Open	Joint synthesis of repeated phase targets.
G1	A number-conserving Gaussian target is shortened by Bogoliubov escape.	Open	Exact low-mode orthogonal/symplectic search.
G2	A nearest-neighbor matchgate target is shortened by restored mode permutation.	Open	Architecture-weighted routing theorem.
S1	A constant-size symmetry-breaking excursion can save unbounded cost.	Open	Explicit sector-coupled family and lower bound.
A1	A fixed number of dirty ancillas can yield unbounded saving in a non-linear quantum gate model.	Open	Family with all-input restoration proof.
R1	A nonsimply laced root system has catalytic ratio exceeding $3/2$.	Open	Low-rank Chevalley word search and restricted lower bound.
H1	Typed Green kernels yield non-enumerative distance lower bounds.	Open	Representation-theoretic coefficient inequalities.
H2	Finite-degree transition moments suffice to distinguish all Latin squares of fixed order.	Open	Moment hierarchy and counterexample search.
Q1	The order-six quantum-Latin product realization is empty while the entangled realization is nonempty.	Established externally	See Ball–Simoens and cited entangled constructions.

C.4.2 Negative results worth preserving

The following failed or negative conclusions are part of the research record.

- Comparisons involving the separate hard Latin line-spread computation remain pending until its input representatives and verifier are supplied.
- Abstract row-column-symbol eigenvalues do not distinguish order-six Latin main classes.
- No classification of the two hard classes by a projective spread operator is certified by the present archive.
- Three-bit Toffoli assistance does not shorten affine targets in the tested costs.
- Adding unit-cost SWAP removes the three-qubit total-count Clifford defects in the declared quotient model.

These results prevent overgeneralization and sharpen the next search spaces.

C.4.3 Publication obligations

Before any skeleton chapter is separated into a theorem paper, complete:

1. targeted literature review,
2. independent verification of all exact computations,
3. a formal statement of equivalence conventions,
4. proof or certificate for every lower bound,
5. archive of source, expected outputs, and manifests,
6. explicit discussion of what the model omits.

APPENDIX D

The formal circuit model

The purpose and evidence contract

This companion explains the actual Lean source behind the Circuit Obstructions main paper. It supplies the computational model, the meaning of its predicates, the route from legal gates to the joint lower bound, and all 115 indexed theorem types with exact source locations. The written arguments are explained for a mathematical reader; the source proof terms remain authoritative.

The formal source has 115 indexed theorem declarations across eight modules.

> **Current formal verification**
>
> Fresh source compilation, axiom inspection, and replay of all eight modules passed. All eight deliberately invalid or forbidden controls were rejected. The source package preserves the complete current receipt and logs alongside historical evidence.

The formal subset consists of C01, the maintained NCT sharing-degree dependency from F3, and the numerical lower-bound part of C02. C02's evaluation-span dual, affine-input invariance and general monotone ideal, together with C03–C08, remain outside this block. A theorem-name count is not a count of independent discoveries. Forty declarations were inherited and 75 added in Block 02.

Part	Reader's task
Explanatory body	Check the hypotheses and how the predicates represent Boolean mathematics
Proof dependencies	Follow legal reversibility, cut coverage, shared degree, balancing and affine compilation
Evidence chapters	Distinguish written reasoning, exact finite checks, text scans and archived kernel checking
Declaration register	Inspect each complete elaborated type, file, line, block and axiom set

The guide does not supply a new toolchain, an independent checker or independent human review. It preserves the original rights record and the original release files. There is no new publication-novelty or unrestricted hardness claim.

The formal endpoint

Let c be a finite legal classical NOT/CNOT/Toffoli circuit. For n input bits and a zero-initialized ancillas, assume for every x and both initial target values b that

$$\mathrm{run}(c)(x, 0^a, b) = (x, 0^a, b \oplus f(x)).$$

Write $T = \mathrm{tc}(c)$ for its number of Toffoli gates. The Block 02 joint endpoint is

Degree and complete-cover bound

$$\boxed{\mathrm{DegreeLE}(T+1, f) \quad \text{and} \quad 2\beta^*(f) \le T + 3.}$$

`CircuitObstructions.restored_degree_and_cover_bound`

The first conclusion is an explicit certificate that f has a Boolean polynomial representation of degree at most $T + 1$. The second is the affine-optimized complete safe-cover lower bound, equivalently $T \ge \max\{0, 2\beta^*(f) - 3\}$.

This applies even when the output controls intermediate gates, the inputs are modified temporarily, and workspace is dirty internally. It requires restoration only at the boundary. It makes no restriction to clean intermediate representations.

What changed

Block 01 proved that discrepancies between a prefix and an inverse suffix form a complete safe cover. Block 02 supplies the missing degree dependency, proves that the cover parameters exist, constructs the balanced cut, and implements affine-output adjustment using legal zero-Toffoli circuits.

What did not change

The mathematical scope remains the eight maintained claim groups. C02 is still partial as a group: its dual-certificate theorem is not included. The formal source also proves

$$\beta(f) \le \beta^*(f) \le n.$$

This ceiling prevents the degree parameter alone from producing a superpolynomial lower bound. No new best-known bound or publication-novelty claim is made.

Where the proof lives

Every proof term is in `formalization/lean/CircuitObstructions/`. The source inventory, archived 21 September elaboration log, and archived axiom report identify the exact statements. The register in this document lists all 115 maintained theorem declarations, including the 40 inherited declarations. Imported standard-library theorems are not counted.

Model and definitions

THE ASSUMPTIONS ARE PART OF THE THEOREM

A general register state is a Boolean function $W \to \text{Bool}$ on any wire type with decidable equality. C01 does not assume that W is finite. The canonical degree model has a finite wire set with n input wires, a ancillas, and one target. A NOT complements a bit. A CNOT adds one control to a distinct target. A Toffoli adds the product of two distinct controls to a third distinct wire. Gate constructors carry the required inequalities.

All arithmetic is over $\mathbb{F}_2$, implemented using Boolean XOR and AND. Each gate is its own inverse. Circuit reversal, restoration, and every cut are inherited from the unchanged Block 01 source.

Quantity	Meaning in the formal source
T	Toffoli count. NOT and CNOT have cost zero in this count only.
DegreeLE(d, f)	f is an XOR of finitely many monomials, each containing at most d variable occurrences.
Safe detector	A Boolean h with $h(x) = 1 \Rightarrow f(x) = 1$.
Complete coverage	Every positive v has a safe detector with $h(v) = 1$.
$\beta(f)$	Least degree bound sufficient for complete safe coverage.
$\beta^*(f)$	Least upper bound on $\beta(f \oplus g)$ over all degree-one g.

Polynomial certificates are not postulated degree oracles

The inductive certificate has three constructors: zero, a bounded-length monomial, and XOR of two certified functions. It is tied to the concrete evaluator

$$\text{polyEval}(p, x) = \bigoplus_{L \in p} \prod_{i \in L} x_i$$

by a proved equivalence. A certificate yields an explicit finite monomial list, and such a list yields a certificate.

Repeated variables are allowed in the list. Boolean idempotence reduces their repetitions without increasing degree. This is the representation form of the usual Boolean degree bound. The formal source does not separately implement canonical ANF normalization or claim that extracting every homogeneous grade is free.

Boundary of the model

There are no quantum phases, measurements, or discarded work registers. Inputs and zero ancillas restore for every logical input and both target values. Standard classical choice defines least semantic bounds, but supplies no polynomial-time procedure for finding their witnesses.

Why sharing does not break the degree bound

A STRONGER INVARIANT THAN SINGLE-REGISTER DEGREE

For a current state $s(x)$ after t Toffolis, the invariant is

$$\forall L, \qquad \mathrm{DegreeLE}\left(t + |L|,\ x \mapsto \prod_{i \in L} s_i(x)\right).$$

L may be any finite list, including repeated registers. Initially each coordinate is a variable or a constant, so a product of r coordinates has degree at most r.

Suppose the next gate toggles target u by the product of a control list of length q. A product that omits u is unchanged. If u occurs, idempotence removes all its repetitions. With R the remaining list, the updated product is

$$\left(s_u \prod_{i \in R} s_i\right) \oplus \left(\prod_{j \in \mathrm{controls}} s_j \prod_{i \in R} s_i\right).$$

The two old-state products have certified degrees at most

$$t + 1 + |R| \qquad \text{and} \qquad t + q + |R|.$$

Since $|R| + 1 \le |L|$, the invariant's budget rises by at most $\max\{q - 1, 0\}$: zero for NOT and CNOT, one for Toffoli.

Induction along the actual gate list gives

$$\mathrm{DegreeLE}\left(T + r,\ x \mapsto \prod_{i \in L} \mathrm{run}(c)(s(x))_i\right), \qquad r = |L|.$$

With $r = 1$, every register has degree at most $T + 1$.

> **Why this handles a DAG rather than just a formula**
>
> The proof never unfolds a reused intermediate value into duplicated computation. It estimates arbitrary products of the current registers in one shared state. The budget increases once per actual Toffoli, including when its controls share earlier work or its target participates in feedback.

The statements are `workspaceDegree_run` and `canonical_run_coordinate_degree`. This formalizes the maintained NCT specialization of F3. A separate generic XOR–AND circuit syntax and simulation theorem are not added here.

From the cut to a numerical lower bound

EXISTENCE, BALANCE, AND COMPLETE COVERAGE

Write $c = p +\!+ q$. At this cut, Block 01 defines

$$H_i(x) = \mathsf{run}(p)(x, 0^a, 0)_i \oplus \mathsf{run}(q^{-1})(x, 0^a, 0)_i$$

and proves

$$H_i \le f, \qquad f(x) = 1 \iff \exists i\ H_i(x) = 1.$$

Reversal does not change Toffoli count. The new coordinate degree theorem therefore gives

$$\deg(H_i) \le \max\{\mathsf{tc}(p), \mathsf{tc}(q)\} + 1.$$

The least safe degree exists

For a positive v, take the point indicator

$$\delta_v(x) = \prod_{i=1}^{n} \begin{cases} x_i, & v_i = 1, \\ 1 + x_i, & v_i = 0. \end{cases}$$

It is safe, detects v, and has a degree-n certificate. Hence complete safe coverage exists at degree n for every function. Strong induction proves the existence of a least admissible natural degree. The definition `beta` chooses this bound and its specification and minimality are proved.

The balanced cut is constructed

Because each gate has cost zero or one, a prefix can realize every Toffoli count from zero to T. The source proves this fact and extracts a cut with

$$\max\{\mathsf{tc}(p), \mathsf{tc}(q)\} \le \lceil T/2 \rceil .$$

The complete safe cut cover then proves

$$\beta(f) \le \lceil T/2 \rceil + 1, \qquad \boxed{2\beta(f) \le T + 3.}$$

Natural-number arithmetic retains the floor/ceiling and small-T cases. No balance assumption is added to the final theorem.

Affine output adjustment is an actual circuit

THE OPTIMIZATION DOES NOT ASSUME A CLEAN NORMAL FORM

For every degree-one Boolean function g, the source constructs a restored implementation with zero Toffolis.

A constant is computed by the empty circuit or target NOT. An input variable uses a legal input-to-target CNOT. XOR is implemented by concatenating restored circuits. Induction on a degree-one certificate gives the general construction.

If c restores f and d restores g, their concatenation restores $f \oplus g$. For the affine implementation,

$$\mathrm{tc}(c +\!\!+ d) = \mathrm{tc}(c).$$

The unoptimized bound applies to every affine g at the same T. Define $\beta^*(f)$ as the least upper bound of those $\beta(f \oplus g)$ values. Existence follows from their common upper bound n.

Therefore

$$\boxed{2\beta^*(f) \le T + 3, \qquad \beta(f) \le \beta^*(f) \le n.}$$

The final output at initial target zero is exactly f, so the forward coordinate theorem also gives DegreeLE$(T + 1, f)$. The joint endpoint combines these two proved conclusions.

Nonvacuity: two-input AND

Every degree-one function on two inputs has zero XOR over the four corners of the input square. AND has corner XOR one. Thus AND has no degree-one certificate.

A degree-one complete safe cover for AND would include a detector of its sole accepting input. Safety forces that detector to be AND itself, contradicting the corner-parity lemma. The degree-two monomial is a valid detector. The source proves

$$\beta(\mathrm{AND}_2) = \beta^*(\mathrm{AND}_2) = 2$$

and therefore every restored implementation of AND_2 uses at least one Toffoli.

This is a theorem about all implementations in the model, not an exhaustive search through circuits. It does not formalize the general k-input monomial optimum.

Verification and proof boundaries

UNIVERSAL PROOFS AND FINITE CONTROLS REMAIN DISTINCT

Module	Count	Formal role
Gates	16	Inherited legal primitives and reversal
CutComparison	13	Inherited complete cut cover
Examples	11	Inherited dirty and output-active controls
Degree	19	Polynomial certificates and closure
MultiplicativeDegree	12	Degree invariant with sharing
SafeDegree	22	Least complete degree and balanced cuts
AffineOutput	14	Affine compiler and optimized bound
DegreeChecks	8	AND2 exclusion and exact parameters
Total	115	75 new declarations, 40 inherited

The current audit rebuilt all source in a fresh project directory, treated warnings as errors, printed each declaration's type and axiom dependencies, and replayed all eight modules through the bundled checker. Its eight deliberately invalid or forbidden controls were rejected.

Only `propext`, `Classical.choice`, and `Quot.sound` occur in the archived audited dependencies. No custom axiom, admitted proof, native-decision oracle, or kernel bypass is used. Lean 4.34.0-rc2 and its bundled Std are pinned. This checker shares the toolchain's trust boundary and is not an independent implementation or human review.

The separate Python checker, freshly reproduced in this edition, uses exact truth tables and integer Möbius transforms. It covers all 278 functions on zero through three inputs; all words over the documented 12-gate three-wire alphabet through length three and the 28-gate four-wire alphabet through length two (2,698 words); 28,088 register-product checks; and 251 restored words. The universal theorems do not follow by extrapolation from those finite tests.

What remains outside the formal block

C02's evaluation-span dual, affine input invariance, and general monotone ideal are not formalized. C03–C08 retain their previous written and finite-check status. The original classical/reversible circuit-size comparison and general exact monomial-cost theorem are not newly formalized. The active claim register controls interpretation.

Reading the formal model literally

QUANTIFIERS AND ENCODING CARRY THE HYPOTHESES

The generic cut theory uses a logical-input type X, a wire type W with decidable equality, and a boundary structure. Its essential fields are an encoding $\mathrm{encode} : X \to \mathrm{Bool} \to (W \to \mathrm{Bool})$, a target wire, and the proof

$$\mathrm{encode}(x, b)(\mathrm{target}) = b.$$

This target equation separates the two target branches for each fixed input. It excludes an encoding that silently ignores its supplied target bit. The source proves `Boundary.target_separated`; it does not assume an injective encoding of every logical-input value as an additional hypothesis.

The restoration predicate is

$$\forall x\, b, \quad \mathrm{run}(c)(\mathrm{encode}(x, b)) = \mathrm{encode}(x, b + f(x)).$$

It is a hypothesis of the conditional theorem. The code does not assert that every gate list realizes a restored oracle. The canonical encoding sets each input wire to its input bit, each ancillary wire to zero, and the target to b. Its wire type is

```
Sum (Fin n) (Sum (Fin a) Unit)
```

which has exactly $n + a + 1$ wires.

In the generic C01 theorem, completeness is the existential statement

$$f(x) = 1 \quad \Longleftrightarrow \quad \exists i : W,\ H_i(x) = 1.$$

No finite disjunction is needed there. Specializing to the canonical finite wires yields the finite complete family used in the main paper. Completeness does not say that the XOR of all packets reconstructs f; detecting packets can cancel under XOR. It states existence of a detecting coordinate at every positive input.

No clean-intermediate condition occurs in these predicates. Legal gates may temporarily alter inputs, dirty scratch and use the target as a control. The source examples exercise those allowances. Its compute/copy/uncompute AND example uses two Toffolis and is a restoration witness, whereas direct AND uses one Toffoli. Neither example is an unrestricted circuit-search result.

Certificates, semantic minima and natural arithmetic

WHAT THE FORMAL OBJECTS DO AND DO NOT COMPUTE

`DegreeLE d f` is an inductive representation predicate. Its constructors build zero, a monomial represented by a variable-occurrence list of length at most d, and XOR of certified functions. The polynomial evaluator XORs the monomial-list values. The source proves the equivalence between a degree certificate and such a finite representation.

Repeated indices are permitted. Boolean idempotence reduces their repetitions, so the representation gives the usual multilinear degree bound. The formal block supplies no separate canonical ANF normalization algorithm and no theorem that extracting homogeneous grades is computationally free. Zero has a degree-zero certificate. The Python degree helper's -1 convention for zero is a mathematical convention outside the natural-valued formal bound.

`CompleteSafeDegreeLE f d` says that every positive point has some degree-d safe detector. It does not require one detector to cover all positives, and it places no efficiency bound on finding those detectors. Point indicators show that $d = n$ suffices. A least-bound existence theorem then permits the noncomputable definition `beta`. Its specification supplies complete coverage; its minimality compares it with every sufficient natural bound.

`AffineSafeDegreeLE f d` universally quantifies every degree-one output function g and requires $\beta(f + g) \leq d$. The definition `betaStar` chooses its least common upper bound. There are finitely many Boolean functions on n inputs, so this least bound equals the main paper's maximum over affine output masks. It is a maximum over mask values, not a minimum over masks.

The source proves $\beta(f) \leq \beta^*(f) \leq n$. The zero mask supplies the first inequality. The point-indicator bound supplies the second. Classical choice in the definitions is not a polynomial-time algorithm for computing these parameters.

All counters and degree bounds in this block are natural numbers. In `2 * betaStar f - 3`, subtraction saturates at zero. Thus

$$2\beta^*(f) - 3 \leq T$$

in the formal natural-number statement is the integer inequality $\max\{0, 2\beta^*(f) - 3\} \leq T$. The nonsubtractive endpoint $2\beta^*(f) \leq T + 3$ avoids ambiguity. Rounding at odd T is retained; the balanced cut is chosen by Toffoli count, not by half the gate-list length.

The proof dependency route

A SOURCE MAP RATHER THAN A NEW AXIOM

Stage	Source role and conclusion
Legal gates and reversal	Distinct-wire constructor proofs; each gate is self-inverse; reverse gate lists invert the run
Separated boundary	The two initial target branches remain distinct after every prefix
Cut transport	Inverse suffix meets the appropriate forward target branch; $H_i = fD_i$ gives safety and pointwise completeness
Polynomial certificates	Monomial evaluation and XOR/product closure connect concrete lists to degree bounds
Shared register products	Products of r current registers have budget $t + r$ after t Toffolis; repeats and feedback are allowed
Balanced cut	An attainable prefix cost gives both sides cost at most $\lceil T/2 \rceil$; the complete family gives the beta bound
Affine-output circuit	Degree-one certificates compile to restored NOT/CNOT lists, giving zero extra Toffolis for every mask
Joint endpoint	The forward target at initial zero supplies f; degree and optimized-cover conclusions are conjoined

The strongest degree invariant concerns products of actual current registers. It does not duplicate the histories of shared controls. This is why its coordinate consequence applies to DAG reuse, temporary input modification and target feedback in this gate model. The general XOR–AND syntax argument in the main paper remains a separate written proof.

The general run theorem needs degree-one certificates for the initial register functions. The canonical boundary supplies these certificates; restoration is used later to identify the final target with f. The affine existence theorem constructs gate lists inside its proof. It does not expose a separate executable compiler function or assert minimum total gate count.

The balanced cut and least-degree facts are established dependencies. They are not extra assumptions added to `restored_degree_and_cover_bound`. That endpoint assumes a legal gate list and restoration in the canonical boundary model. The register prints its full type as L107.

Formal and computational evidence

FRESH SOURCE CHECKING AND BOUNDED EXACT REPLAY

The current pinned Lean replay passed for all 115 indexed theorem declarations and eight modules. The permitted axiom union is `propext`, `Classical.choice`, and `Quot.sound`. No project axiom or admitted proof occurs in the maintained source.

The complete exact scientific replay also passed. Its finite domains are recorded in the claim-coverage appendix and in the emitted run manifest. The degree-profile checker counts 278 functions on zero through three inputs. The small-oracle construction suite counts 276 functions on one through three inputs. The two zero-arity functions explain the difference. Those totals are not added together as disjoint evidence.

The package retains historical failures as historical records. Current source compilation, kernel replay, finite computation, source integrity, and PDF rendering each establish a different kind of fact.

Negative controls and the trust boundary

A REJECTION HAS A PRECISE ROLE

The current audit deliberately tested the following controls:

Control	Why rejection matters
Illegal repeated wire	Legal gate constructors require distinct controls and target
Target-only cut reconstruction	A discrepancy can move to a data wire; completeness needs the entire coordinate family
False theorem	A false statement is not accepted merely because it has a theorem wrapper
Placeholder guard	The project rejects admitted or placeholder source constructs
False monomial degree	A two-occurrence monomial cannot satisfy the supplied length-at-most-one premise
False affine AND claim	The supplied degree-two proof has the wrong type for a degree-one target
False beta value	Complete safe detection for AND2 cannot have degree one
False betaStar bound	Optimization cannot erase the AND2 obstruction

All eight controls were freshly rejected. They help check that the audit detects these specified errors. They are not additional positive mathematical results or an exhaustive attack on every possible checker defect. The separate positive four-corner theorem proves the semantic impossibility of an affine AND2 certificate; it is not the mechanism of that particular wrong-type negative test.

The bundled checker shares the Lean toolchain's trust boundary. It is not an independently implemented proof checker. The restored runtime archive was checked against the recorded combined hash. The runtime remains a separately supplied dependency.

Historical development failures, rejected negative controls, and successful current checks are retained as separate events in the evidence package.

Reproduce and inspect the authoritative files

COMMANDS AND SOURCE IDENTITY

From `source/komorebi-circuit-obstructions-v1.2.0`, run

```
python scripts/check_integrity.py
python scripts/verify.py --all
python scripts/verify_lean.py --toolchain /path/to/toolchain
```

The integrity check needs Python; the full scientific replay also needs a C++17 compiler. The last command needs the pinned Lean toolchain supplied separately. The verification scripts do not download it. A missing toolchain fails before proof checking. The current completion pass restored the pinned runtime and successfully executed this verification.

From the publication-set root, regenerate the exact declaration appendix and the explanatory guide source with

```
python scripts/build_lean_register.py
python scripts/build_lean_guide.py
python scripts/build_documents.py
python scripts/verify_publication.py
```

The register generator verifies all 115 source hashes and declaration lines against the archived catalog, then copies each complete elaborated type. The guide generator preserves the archived explanatory proof route and adds this edition's evidence distinctions. Neither script compiles Lean.

The unmodified source ZIP's SHA-256 is

```
88628c8274660bd20e11fec83249fdc04960403b9d64d1e5068a476b8c025936
```

The complete publication package preserves that ZIP, the repository bundle, original companions and current evidence. Its new manifest binds the revised papers, generators and figure assets. Hash consistency is not an authorship certificate or mathematical proof.

The declaration register follows. It includes the 40 inherited results and 75 Block 02 additions. The ten non-ASCII logical glyphs in its exact type text are rendered as mathematical glyphs for readability. The unchanged UTF-8 sidecar is `manuscripts/lean_statements_utf8.txt`; the original authoritative catalog is `reference/lean-block02/declaration_catalog.json` in the source release. Imported library declarations and explanatory prose are not counted as new project theorems.

Complete declaration register

The following 115 types are transcribed from the archived Block 02 declaration catalog. Unicode logical operators are rendered as mathematical glyphs; the complete original UTF-8 statements are also supplied. Source files and line locators match the preserved source hashes. The current replay rechecked the source declarations. The transcribed types remain tied to the supplied declaration catalog.

Gates: 16 declarations

L001 `write_same`
Source. `CircuitObstructions/Gates.lean:17`
Block. 01 **Axioms.** propext

```
CircuitObstructions.write_same {W : Type} [DecidableEq W] (s : CircuitObstructions.State W) (t : W) (b : Bool) :
  CircuitObstructions.write s t b t = b
```

L002 `write_other`
Source. `CircuitObstructions/Gates.lean:20`
Block. 01 **Axioms.** propext

```
CircuitObstructions.write_other {W : Type} [DecidableEq W] (s : CircuitObstructions.State W) (t i : W) (b : Bool)
  (h : i ≠t) : CircuitObstructions.write s t b i = s i
```

L003 `xor_cancel_right`
Source. `CircuitObstructions/Gates.lean:23`
Block. 01 **Axioms.** none

```
CircuitObstructions.xor_cancel_right (a b : Bool) : (a ^^ b ^^ b) = a
```

L004 `xor_cancel_left`
Source. `CircuitObstructions/Gates.lean:26`
Block. 01 **Axioms.** none

```
CircuitObstructions.xor_cancel_left (a b : Bool) : (a ^^ (a ^^ b)) = b
```

L005 `xor_commute`
Source. `CircuitObstructions/Gates.lean:29`
Block. 01 **Axioms.** none

```
CircuitObstructions.xor_commute (a b : Bool) : (a ^^ b) = (b ^^ a)
```

L006 `xor_true_iff_ne`
Source. `CircuitObstructions/Gates.lean:32`
Block. 01 **Axioms.** none

```
CircuitObstructions.xor_true_iff_ne (a b : Bool) : (a ^^ b) = true ↔a ≠b
```

L007 `eval_involutive`
Source. `CircuitObstructions/Gates.lean:48`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.Gate.eval_involutive {W : Type} [DecidableEq W] (g : CircuitObstructions.Gate W)
  (s : CircuitObstructions.State W) : g.eval (g.eval s) = s
```

L008 `eval_injective`
Source. `CircuitObstructions/Gates.lean:67`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.Gate.eval_injective {W : Type} [DecidableEq W] (g : CircuitObstructions.Gate W)
 {s t : CircuitObstructions.State W} (h : g.eval s = g.eval t) : s = t
```

L009 `run_nil`
Source. `CircuitObstructions/Gates.lean:78`
Block. 01 **Axioms.** none

```
CircuitObstructions.run_nil {W : Type} [DecidableEq W] (s : CircuitObstructions.State W) :
 CircuitObstructions.run [] s = s
```

L010 `run_cons`
Source. `CircuitObstructions/Gates.lean:81`
Block. 01 **Axioms.** none

```
CircuitObstructions.run_cons {W : Type} [DecidableEq W] (g : CircuitObstructions.Gate W)
 (gs : CircuitObstructions.Circuit W) (s : CircuitObstructions.State W) :
 CircuitObstructions.run (g :: gs) s = CircuitObstructions.run gs (g.eval s)
```

L011 `run_append`
Source. `CircuitObstructions/Gates.lean:84`
Block. 01 **Axioms.** none

```
CircuitObstructions.run_append {W : Type} [DecidableEq W] (p q : CircuitObstructions.Circuit W)
 (s : CircuitObstructions.State W) :
 CircuitObstructions.run (p ++ q) s = CircuitObstructions.run q (CircuitObstructions.run p s)
```

L012 `run_reverse_run`
Source. `CircuitObstructions/Gates.lean:90`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.run_reverse_run {W : Type} [DecidableEq W] (c : CircuitObstructions.Circuit W)
 (s : CircuitObstructions.State W) : CircuitObstructions.run (List.reverse c) (CircuitObstructions.run c s) = s
```

L013 `run_run_reverse`
Source. `CircuitObstructions/Gates.lean:98`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.run_run_reverse {W : Type} [DecidableEq W] (c : CircuitObstructions.Circuit W)
 (s : CircuitObstructions.State W) : CircuitObstructions.run c (CircuitObstructions.run (List.reverse c) s) = s
```

L014 `run_injective`
Source. `CircuitObstructions/Gates.lean:102`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.run_injective {W : Type} [DecidableEq W] (c : CircuitObstructions.Circuit W)
 {s t : CircuitObstructions.State W} (h : CircuitObstructions.run c s = CircuitObstructions.run c t) : s = t
```

L015 `toffoliCount_append`
Source. `CircuitObstructions/Gates.lean:114`
Block. 01 **Axioms.** propext

```
CircuitObstructions.toffoliCount_append {W : Type} (p q : CircuitObstructions.Circuit W) :
 CircuitObstructions.toffoliCount (p ++ q) = CircuitObstructions.toffoliCount p + CircuitObstructions.toffoliCount q
```

L016 `toffoliCount_reverse`
Source. `CircuitObstructions/Gates.lean:118`
Block. 01 **Axioms.** propext

```
CircuitObstructions.toffoliCount_reverse {W : Type} (c : CircuitObstructions.Circuit W) :
 CircuitObstructions.toffoliCount (List.reverse c) = CircuitObstructions.toffoliCount c
```

CutComparison: 13 declarations

L017 `target_separated`
Source. `CircuitObstructions/CutComparison.lean:17`
Block. 01 **Axioms.** propext

```
CircuitObstructions.Boundary.target_separated {X W : Type} (e : CircuitObstructions.Boundary X W) (x : X) :
 e.encode x false ≠e.encode x true
```

L018 `canonical_input`
Source. `CircuitObstructions/CutComparison.lean:39`
Block. 01 **Axioms.** none

```
CircuitObstructions.canonical_input (n a : Nat) (x : Fin n →Bool) (b : Bool) (i : Fin n) :
 (CircuitObstructions.canonicalBoundary n a).encode x b (Sum.inl i) = x i
```

L019 `canonical_ancilla`
Source. `CircuitObstructions/CutComparison.lean:42`
Block. 01 **Axioms.** none

```
CircuitObstructions.canonical_ancilla (n a : Nat) (x : Fin n →Bool) (b : Bool) (i : Fin a) :
 (CircuitObstructions.canonicalBoundary n a).encode x b (Sum.inr (Sum.inl i)) = false
```

L020 `canonical_target`
Source. `CircuitObstructions/CutComparison.lean:45`
Block. 01 **Axioms.** none

```
CircuitObstructions.canonical_target (n a : Nat) (x : Fin n →Bool) (b : Bool) :
 (CircuitObstructions.canonicalBoundary n a).encode x b (Sum.inr (Sum.inr ())) = b
```

L021 `inverse_suffix_boundary`
Source. `CircuitObstructions/CutComparison.lean:49`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.inverse_suffix_boundary {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p q : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e (p ++ q) f) (x : X)
 (b : Bool) :
 CircuitObstructions.run (List.reverse q) (e.encode x b) = CircuitObstructions.run p (e.encode x (b ^^ f x))
```

L022 `direction_nonzero`
Source. `CircuitObstructions/CutComparison.lean:67`
Block. 01 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.direction_nonzero {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p : CircuitObstructions.Circuit W) (x : X) : ∃i, CircuitObstructions.direction e p x i = true
```

L023 `cut_transport_identity`
Source. `CircuitObstructions/CutComparison.lean:82`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.cut_transport_identity {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p q : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e (p ++ q) f) (x : X)
 (i : W) : CircuitObstructions.packet e p q x i = (f x && CircuitObstructions.direction e p x i)
```

L024 `cut_packet_safe`
Source. `CircuitObstructions/CutComparison.lean:91`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.cut_packet_safe {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p q : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e (p ++ q) f) (x : X)
 (i : W) : CircuitObstructions.packet e p q x i = true →f x = true
```

L025 `cut_cover_complete`
Source. `CircuitObstructions/CutComparison.lean:99`
Block. 01 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.cut_cover_complete {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p q : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e (p ++ q) f) (x : X) :
 f x = true ↔∃i, CircuitObstructions.packet e p q x i = true
```

L026 `complete_safe_cover`
Source. `CircuitObstructions/CutComparison.lean:111`
Block. 01 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.complete_safe_cover {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p q : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e (p ++ q) f) :∀
 ( (x : X) (i : W), CircuitObstructions.packet e p q x i = true →f x = true) ∧∀
   (x : X), f x = true ↔∃i, CircuitObstructions.packet e p q x i = true
```

L027 `cut_states_equal_iff_reject`
Source. `CircuitObstructions/CutComparison.lean:119`
Block. 01 **Axioms.** propext, Quot.sound

```
CircuitObstructions.cut_states_equal_iff_reject {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (p q : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e (p ++ q) f) (x : X) :
 CircuitObstructions.run p (e.encode x false) = CircuitObstructions.run (List.reverse q) (e.encode x false) ↔
  f x = false
```

L028 `every_index_cut`
Source. `CircuitObstructions/CutComparison.lean:131`
Block. 01 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.every_index_cut {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (c : CircuitObstructions.Circuit W) (f : X →Bool) (hR : CircuitObstructions.Restores e c f) (k : Nat) (x : X) :
 f x = true ↔∃i, CircuitObstructions.packet e (List.take k c) (List.drop k c) x i = true
```

L029 `canonical_complete_safe_cover`
Source. `CircuitObstructions/CutComparison.lean:139`
Block. 01 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.canonical_complete_safe_cover (n a : Nat)
 (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : (Fin n →Bool) →Bool)
 (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) (k : Nat) :∀
 ( (x : Fin n →Bool) (i : CircuitObstructions.CanonicalWire n a),
    CircuitObstructions.packet (CircuitObstructions.canonicalBoundary n a) (List.take k c) (List.drop k c) x i =
       true →
     f x = true) ∧∀
   (x : Fin n →Bool),
    f x = true ↔∃
      i,
       CircuitObstructions.packet (CircuitObstructions.canonicalBoundary n a) (List.take k c) (List.drop k c) x i =
         true
```

Examples: 11 declarations

L030 `andOracle_restores`
Source. `CircuitObstructions/Examples.lean:27`
Block. 01 **Axioms.** Quot.sound

```
CircuitObstructions.Examples.andOracle_restores :
 CircuitObstructions.Restores CircuitObstructions.Examples.andBoundary CircuitObstructions.Examples.andOracle fun x =>
  x.fst && x.snd
```

L031 `andOracle_toffoli_count`
Source. `CircuitObstructions/Examples.lean:33`
Block. 01 **Axioms.** propext

```
CircuitObstructions.Examples.andOracle_toffoli_count :
 CircuitObstructions.toffoliCount CircuitObstructions.Examples.andOracle = 2
```

L032 `and_scratch_can_be_dirty`
Source. `CircuitObstructions/Examples.lean:36`
Block. 01 **Axioms.** none

```
CircuitObstructions.Examples.and_scratch_can_be_dirty :
 CircuitObstructions.run CircuitObstructions.Examples.andCompute
    (CircuitObstructions.Examples.andBoundary.encode (true, true) false)
    CircuitObstructions.Examples.WorkWire.scratch =
  true
```

L033 `andOracle_every_cut`
Source. `CircuitObstructions/Examples.lean:40`
Block. 01 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.Examples.andOracle_every_cut (k : Nat) (x : Bool ×Bool) :
 (x.fst && x.snd) = true ↔∃
   i,
    CircuitObstructions.packet CircuitObstructions.Examples.andBoundary
       (List.take k CircuitObstructions.Examples.andOracle) (List.drop k CircuitObstructions.Examples.andOracle) x
       i =
     true
```

L034 `outputActiveOracle_restores`
Source. `CircuitObstructions/Examples.lean:66`
Block. 01 **Axioms.** Quot.sound

```
CircuitObstructions.Examples.outputActiveOracle_restores :
 CircuitObstructions.Restores CircuitObstructions.Examples.swapBoundary CircuitObstructions.Examples.outputActiveOracle
   fun x => true
```

L035 `outputActiveOracle_length`
Source. `CircuitObstructions/Examples.lean:71`
Block. 01 **Axioms.** none

```
CircuitObstructions.Examples.outputActiveOracle_length : List.length CircuitObstructions.Examples.outputActiveOracle = 7
```

L036 `outputActiveOracle_toffoli_count`
Source. `CircuitObstructions/Examples.lean:73`
Block. 01 **Axioms.** propext

```
CircuitObstructions.Examples.outputActiveOracle_toffoli_count :
 CircuitObstructions.toffoliCount CircuitObstructions.Examples.outputActiveOracle = 0
```

L037 `moved_discrepancy_data`
Source. `CircuitObstructions/Examples.lean:76`
Block. 01 **Axioms.** none

```
CircuitObstructions.Examples.moved_discrepancy_data (x : Bool) :
 CircuitObstructions.packet CircuitObstructions.Examples.swapBoundary CircuitObstructions.Examples.swapCircuit
    CircuitObstructions.Examples.swapSuffix x CircuitObstructions.Examples.SwapWire.data =
   true
```

L038 `moved_discrepancy_target`
Source. `CircuitObstructions/Examples.lean:80`
Block. 01 **Axioms.** none

```
CircuitObstructions.Examples.moved_discrepancy_target (x : Bool) :
 CircuitObstructions.packet CircuitObstructions.Examples.swapBoundary CircuitObstructions.Examples.swapCircuit
    CircuitObstructions.Examples.swapSuffix x CircuitObstructions.Examples.SwapWire.target =
   false
```

L039 `target_only_reconstruction_is_false`
Source. `CircuitObstructions/Examples.lean:85`
Block. 01 **Axioms.** none

```
CircuitObstructions.Examples.target_only_reconstruction_is_false :
 ¬∀(x : Bool),
    CircuitObstructions.packet CircuitObstructions.Examples.swapBoundary CircuitObstructions.Examples.swapCircuit
       CircuitObstructions.Examples.swapSuffix x CircuitObstructions.Examples.SwapWire.target =
     true
```

L040 `degenerate_encoding_control`
Source. `CircuitObstructions/Examples.lean:93`
Block. 01 **Axioms.** none

```
CircuitObstructions.Examples.degenerate_encoding_control :
 have bad := fun x x_1 => false;∀
 ( (y : Bool), CircuitObstructions.run [] (bad y) = bad (y ^^ true)) ∧∀
   (i : CircuitObstructions.Examples.SwapWire), (bad false i ^^ bad false i) = false
```

Degree: 19 declarations

L041 `bprod_nil`
Source. CircuitObstructions/Degree.lean:19
Block. 02 **Axioms.** none

```
CircuitObstructions.bprod_nil {W : Type} (s : CircuitObstructions.State W) : CircuitObstructions.bprod [] s = true
```

L042 `bprod_cons`
Source. CircuitObstructions/Degree.lean:20
Block. 02 **Axioms.** none

```
CircuitObstructions.bprod_cons {W : Type} (i : W) (is : List W) (s : CircuitObstructions.State W) :
 CircuitObstructions.bprod (i :: is) s = (s i && CircuitObstructions.bprod is s)
```

L043 `bprod_append`
Source. CircuitObstructions/Degree.lean:23
Block. 02 **Axioms.** propext

```
CircuitObstructions.bprod_append {W : Type} (l k : List W) (s : CircuitObstructions.State W) :
 CircuitObstructions.bprod (l ++ k) s = (CircuitObstructions.bprod l s && CircuitObstructions.bprod k s)
```

L044 `xor_and_right`
Source. CircuitObstructions/Degree.lean:29
Block. 02 **Axioms.** none

```
CircuitObstructions.xor_and_right (a b c : Bool) : ((a ^^ b) && c) = (a && c ^^ b && c)
```

L045 `and_xor_right`
Source. CircuitObstructions/Degree.lean:33
Block. 02 **Axioms.** none

```
CircuitObstructions.and_xor_right (a b c : Bool) : (a && (b ^^ c)) = (a && b ^^ a && c)
```

L046 `not_as_xor_one`
Source. CircuitObstructions/Degree.lean:37
Block. 02 **Axioms.** none

```
CircuitObstructions.not_as_xor_one (a : Bool) : (!a) = (a ^^ true)
```

L047 `mono`
Source. CircuitObstructions/Degree.lean:47
Block. 02 **Axioms.** none

```
CircuitObstructions.DegreeLE.mono {n d e : Nat} {f : CircuitObstructions.BFun n} (h : CircuitObstructions.DegreeLE d f)
 (hde : d ≤ e) : CircuitObstructions.DegreeLE e f
```

L048 `one`
Source. CircuitObstructions/Degree.lean:54
Block. 02 **Axioms.** none

```
CircuitObstructions.DegreeLE.one (n d : Nat) : CircuitObstructions.DegreeLE d fun x => true
```

L049 `constant`
Source. CircuitObstructions/Degree.lean:57
Block. 02 **Axioms.** none

```
CircuitObstructions.DegreeLE.constant (n d : Nat) (b : Bool) : CircuitObstructions.DegreeLE d fun x => b
```

L050 `variable`
Source. `CircuitObstructions/Degree.lean:63`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.DegreeLE.variable {n : Nat} (i : Fin n) : CircuitObstructions.DegreeLE 1 fun x => x i
```

L051 `xor`
Source. `CircuitObstructions/Degree.lean:67`
Block. 02 **Axioms.** propext

```
CircuitObstructions.DegreeLE.xor {n d e : Nat} {f g : CircuitObstructions.BFun n}
  (hf : CircuitObstructions.DegreeLE d f) (hg : CircuitObstructions.DegreeLE e g) :
  CircuitObstructions.DegreeLE (max d e) fun x => f x ^^ g x
```

L052 `not`
Source. `CircuitObstructions/Degree.lean:72`
Block. 02 **Axioms.** Quot.sound

```
CircuitObstructions.DegreeLE.not {n d : Nat} {f : CircuitObstructions.BFun n} (hf : CircuitObstructions.DegreeLE d f) :
  CircuitObstructions.DegreeLE d fun x => !f x
```

L053 `monomial_mul`
Source. `CircuitObstructions/Degree.lean:76`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.DegreeLE.monomial_mul {n d e : Nat} (l : List (Fin n)) (hl : l.length ≤d)
  {g : CircuitObstructions.BFun n} (hg : CircuitObstructions.DegreeLE e g) :
  CircuitObstructions.DegreeLE (d + e) fun x => CircuitObstructions.bprod l x && g x
```

L054 `mul`
Source. `CircuitObstructions/Degree.lean:87`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.DegreeLE.mul {n d e : Nat} {f g : CircuitObstructions.BFun n}
  (hf : CircuitObstructions.DegreeLE d f) (hg : CircuitObstructions.DegreeLE e g) :
  CircuitObstructions.DegreeLE (d + e) fun x => f x && g x
```

L055 `product`
Source. `CircuitObstructions/Degree.lean:96`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.DegreeLE.product {n : Nat} {W : Type} (l : List W)
  (s : (Fin n →Bool) →CircuitObstructions.State W) (h : ∀(i : W), CircuitObstructions.DegreeLE 1 fun x => s x i) :
  CircuitObstructions.DegreeLE l.length fun x => CircuitObstructions.bprod l (s x)
```

L056 `polyEval_append`
Source. `CircuitObstructions/Degree.lean:109`
Block. 02 **Axioms.** propext

```
CircuitObstructions.polyEval_append {n : Nat} (p q : List (List (Fin n))) (x : Fin n →Bool) :
  CircuitObstructions.polyEval (p ++ q) x = (CircuitObstructions.polyEval p x ^^ CircuitObstructions.polyEval q x)
```

L057 `polynomial_witness`
Source. `CircuitObstructions/Degree.lean:115`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.DegreeLE.polynomial_witness {n d : Nat} {f : CircuitObstructions.BFun n}
  (h : CircuitObstructions.DegreeLE d f) :∃
   p, ∀( (l : List (Fin n)), l ∈p →l.length ≤d) ∧CircuitObstructions.polyEval p = f
```

L058 `polynomial_degree_certificate`
Source. `CircuitObstructions/Degree.lean:134`
Block. 02 **Axioms.** propext

```
CircuitObstructions.polynomial_degree_certificate {n d : Nat} (p : List (List (Fin n)))
  (hp : ∀(l : List (Fin n)), l ∈p →l.length ≤d) : CircuitObstructions.DegreeLE d (CircuitObstructions.polyEval p)
```

L059 `degreeLE_iff_polynomial`
Source. `CircuitObstructions/Degree.lean:142`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.degreeLE_iff_polynomial {n d : Nat} (f : CircuitObstructions.BFun n) :
  CircuitObstructions.DegreeLE d f ↔∃
    p, ∀( (l : List (Fin n)), l ∈p →l.length ≤d) ∧CircuitObstructions.polyEval p = f
```

MultiplicativeDegree: 12 declarations

L060 `withoutWire_length_le`
Source. `CircuitObstructions/MultiplicativeDegree.lean:17`
Block. 02 **Axioms.** propext

```
CircuitObstructions.withoutWire_length_le {W : Type} [DecidableEq W] (l : List W) (t : W) :
  (CircuitObstructions.withoutWire l t).length ≤l.length
```

L061 `withoutWire_length_lt`
Source. `CircuitObstructions/MultiplicativeDegree.lean:21`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.withoutWire_length_lt {W : Type} [DecidableEq W] (l : List W) (t : W) (ht : t ∈l) :
  (CircuitObstructions.withoutWire l t).length + 1 ≤l.length
```

L062 `bprod_write`
Source. `CircuitObstructions/MultiplicativeDegree.lean:33`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.bprod_write {W : Type} [DecidableEq W] (l : List W) (s : CircuitObstructions.State W) (t : W)
  (b : Bool) :
  CircuitObstructions.bprod l (CircuitObstructions.write s t b) =
    ((if t ∈l then b else true) && CircuitObstructions.bprod (CircuitObstructions.withoutWire l t) s)
```

L063 `bprod_write_absent`
Source. `CircuitObstructions/MultiplicativeDegree.lean:48`
Block. 02 **Axioms.** propext

```
CircuitObstructions.bprod_write_absent {W : Type} [DecidableEq W] (l : List W) (s : CircuitObstructions.State W) (t : W)
  (b : Bool) (ht : ¬t ∈l) :
  CircuitObstructions.bprod l (CircuitObstructions.write s t b) = CircuitObstructions.bprod l s
```

L064 `workspaceDegree_initial`
Source. `CircuitObstructions/MultiplicativeDegree.lean:62`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.workspaceDegree_initial {n : Nat} {W : Type} (s : (Fin n →Bool) →CircuitObstructions.State W)
  (h : ∀(i : W), CircuitObstructions.DegreeLE 1 fun x => s x i) : CircuitObstructions.WorkspaceDegree s 0
```

L065 `workspaceDegree_toggle`
Source. `CircuitObstructions/MultiplicativeDegree.lean:73`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.workspaceDegree_toggle {n : Nat} {W : Type} [DecidableEq W]
  (s : (Fin n →Bool) →CircuitObstructions.State W) (t : W) (controls : List W) (k : Nat)
  (h : CircuitObstructions.WorkspaceDegree s k) :
  CircuitObstructions.WorkspaceDegree (CircuitObstructions.toggleProduct s t controls) (k + (controls.length - 1))
```

L066 `workspaceDegree_gate`
Source. `CircuitObstructions/MultiplicativeDegree.lean:104`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.workspaceDegree_gate {n : Nat} {W : Type} [DecidableEq W]
  (s : (Fin n →Bool) →CircuitObstructions.State W) (g : CircuitObstructions.Gate W) (k : Nat)
  (h : CircuitObstructions.WorkspaceDegree s k) :
  CircuitObstructions.WorkspaceDegree (fun x => g.eval (s x)) (k + g.toffoliCost)
```

L067 `workspaceDegree_run`
Source. `CircuitObstructions/MultiplicativeDegree.lean:128`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.workspaceDegree_run {n : Nat} {W : Type} [DecidableEq W] (c : CircuitObstructions.Circuit W)
  (s : (Fin n →Bool) →CircuitObstructions.State W) (k : Nat) (h : CircuitObstructions.WorkspaceDegree s k) :
  CircuitObstructions.WorkspaceDegree (fun x => CircuitObstructions.run c (s x))
    (k + CircuitObstructions.toffoliCount c)
```

L068 `run_coordinate_degree`
Source. `CircuitObstructions/MultiplicativeDegree.lean:139`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.run_coordinate_degree {n : Nat} {W : Type} [DecidableEq W] (c : CircuitObstructions.Circuit W)
  (s : (Fin n →Bool) →CircuitObstructions.State W) (hs : ∀(i : W), CircuitObstructions.DegreeLE 1 fun x => s x i)
  (i : W) :
  CircuitObstructions.DegreeLE (CircuitObstructions.toffoliCount c + 1) fun x => CircuitObstructions.run c (s x) i
```

L069 `canonical_boundary_degree`
Source. `CircuitObstructions/MultiplicativeDegree.lean:146`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.canonical_boundary_degree (n a : Nat) (b : Bool) (i : CircuitObstructions.CanonicalWire n a) :
  CircuitObstructions.DegreeLE 1 fun x => (CircuitObstructions.canonicalBoundary n a).encode x b i
```

L070 `canonical_run_coordinate_degree`
Source. `CircuitObstructions/MultiplicativeDegree.lean:155`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.canonical_run_coordinate_degree (n a : Nat)
  (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (b : Bool)
  (i : CircuitObstructions.CanonicalWire n a) :
  CircuitObstructions.DegreeLE (CircuitObstructions.toffoliCount c + 1) fun x =>
    CircuitObstructions.run c ((CircuitObstructions.canonicalBoundary n a).encode x b) i
```

L071 `canonical_cut_packet_degree`
Source. `CircuitObstructions/MultiplicativeDegree.lean:161`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.canonical_cut_packet_degree (n a : Nat)
  (p q : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a))
  (i : CircuitObstructions.CanonicalWire n a) :
  CircuitObstructions.DegreeLE (max (CircuitObstructions.toffoliCount p) (CircuitObstructions.toffoliCount q) + 1)
    fun x => CircuitObstructions.packet (CircuitObstructions.canonicalBoundary n a) p q x i
```

SafeDegree: 22 declarations

L072 `mono`
Source. `CircuitObstructions/SafeDegree.lean:16`
Block. 02 **Axioms.** none

```
CircuitObstructions.CompleteSafeDegreeLE.mono {n d e : Nat} {f : CircuitObstructions.BFun n}
  (h : CircuitObstructions.CompleteSafeDegreeLE f d) (hde : d ≤e) : CircuitObstructions.CompleteSafeDegreeLE f e
```

L073 `bprod_true_iff`
Source. `CircuitObstructions/SafeDegree.lean:22`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.bprod_true_iff {W : Type} (l : List W) (s : CircuitObstructions.State W) :
  CircuitObstructions.bprod l s = true ↔∀(i : W), i ∈l →s i = true
```

L074 `pointIndicator_true_iff`
Source. `CircuitObstructions/SafeDegree.lean:31`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.pointIndicator_true_iff {n : Nat} (v x : Fin n →Bool) :
  CircuitObstructions.pointIndicator v x = true ↔x = v
```

L075 `pointIndicator_degree`
Source. `CircuitObstructions/SafeDegree.lean:44`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.pointIndicator_degree {n : Nat} (v : Fin n →Bool) :
  CircuitObstructions.DegreeLE n (CircuitObstructions.pointIndicator v)
```

L076 `completeSafeDegree_le_inputs`
Source. `CircuitObstructions/SafeDegree.lean:56`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.completeSafeDegree_le_inputs {n : Nat} (f : CircuitObstructions.BFun n) :
  CircuitObstructions.CompleteSafeDegreeLE f n
```

L077 `exists_least_degree`
Source. `CircuitObstructions/SafeDegree.lean:65`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.exists_least_degree (P : Nat →Prop) (n : Nat) (hn : P n) : ∃r, P r ∧∀(k : Nat), P k →r ≤k
```

L078 `beta_exists`
Source. `CircuitObstructions/SafeDegree.lean:83`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_exists {n : Nat} (f : CircuitObstructions.BFun n) :∃
  r, CircuitObstructions.CompleteSafeDegreeLE f r ∧∀(k : Nat), CircuitObstructions.CompleteSafeDegreeLE f k →r ≤k
```

L079 `beta_spec`
Source. `CircuitObstructions/SafeDegree.lean:90`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_spec {n : Nat} (f : CircuitObstructions.BFun n) :
  CircuitObstructions.CompleteSafeDegreeLE f (CircuitObstructions.beta f)
```

L080 `beta_le_of_cover`
Source. `CircuitObstructions/SafeDegree.lean:93`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_le_of_cover {n d : Nat} {f : CircuitObstructions.BFun n}
  (h : CircuitObstructions.CompleteSafeDegreeLE f d) : CircuitObstructions.beta f ≤d
```

L081 `beta_le_iff`
Source. `CircuitObstructions/SafeDegree.lean:97`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_le_iff {n d : Nat} (f : CircuitObstructions.BFun n) :
  CircuitObstructions.beta f ≤d ↔CircuitObstructions.CompleteSafeDegreeLE f d
```

L082 `beta_le_inputs`
Source. `CircuitObstructions/SafeDegree.lean:103`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_le_inputs {n : Nat} (f : CircuitObstructions.BFun n) : CircuitObstructions.beta f ≤n
```

L083 `beta_zero`
Source. `CircuitObstructions/SafeDegree.lean:106`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_zero (n : Nat) : (CircuitObstructions.beta fun x => false) = 0
```

L084 `canonical_cut_completeSafeDegree`
Source. `CircuitObstructions/SafeDegree.lean:113`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.canonical_cut_completeSafeDegree (n a : Nat)
  (p q : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
  (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) (p ++ q) f) :
  CircuitObstructions.CompleteSafeDegreeLE f
    (max (CircuitObstructions.toffoliCount p) (CircuitObstructions.toffoliCount q) + 1)
```

L085 `toffoliCost_le_one`
Source. `CircuitObstructions/SafeDegree.lean:124`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.Gate.toffoliCost_le_one {W : Type} (g : CircuitObstructions.Gate W) : g.toffoliCost ≤1
```

L086 `toffoliCount_nil`
Source. `CircuitObstructions/SafeDegree.lean:127`
Block. 02 **Axioms.** propext

```
CircuitObstructions.toffoliCount_nil {W : Type} : CircuitObstructions.toffoliCount [] = 0
```

L087 `toffoliCount_cons`
Source. `CircuitObstructions/SafeDegree.lean:129`
Block. 02 **Axioms.** propext

```
CircuitObstructions.toffoliCount_cons {W : Type} (g : CircuitObstructions.Gate W) (c : CircuitObstructions.Circuit W) :
  CircuitObstructions.toffoliCount (g :: c) = g.toffoliCost + CircuitObstructions.toffoliCount c
```

L088 `exists_cut_with_toffoliCount`
Source. `CircuitObstructions/SafeDegree.lean:133`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.exists_cut_with_toffoliCount {W : Type} (c : CircuitObstructions.Circuit W) (k : Nat)
  (hk : k ≤CircuitObstructions.toffoliCount c) : ∃p q, c = p ++ q ∧CircuitObstructions.toffoliCount p = k
```

L089 `exists_balanced_toffoli_cut`
Source. `CircuitObstructions/SafeDegree.lean:155`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.exists_balanced_toffoli_cut {W : Type} (c : CircuitObstructions.Circuit W) :∃
  p q,
   c = p ++ q ∧
    max (CircuitObstructions.toffoliCount p) (CircuitObstructions.toffoliCount q) ≤
     (CircuitObstructions.toffoliCount c + 1) / 2
```

L090 `canonical_restored_completeSafeDegree`
Source. `CircuitObstructions/SafeDegree.lean:165`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.canonical_restored_completeSafeDegree (n a : Nat)
  (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
  (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
  CircuitObstructions.CompleteSafeDegreeLE f ((CircuitObstructions.toffoliCount c + 1) / 2 + 1)
```

L091 `beta_le_half_toffoliCount`
Source. `CircuitObstructions/SafeDegree.lean:173`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_le_half_toffoliCount (n a : Nat)
  (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
  (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
  CircuitObstructions.beta f ≤(CircuitObstructions.toffoliCount c + 1) / 2 + 1
```

L092 `beta_restoration_bound`
Source. `CircuitObstructions/SafeDegree.lean:179`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_restoration_bound (n a : Nat)
  (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
  (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
  2 * CircuitObstructions.beta f ≤CircuitObstructions.toffoliCount c + 3
```

L093 `toffoliCount_ge_two_beta_sub_three`
Source. `CircuitObstructions/SafeDegree.lean:186`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.toffoliCount_ge_two_beta_sub_three (n a : Nat)
  (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
  (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
  2 * CircuitObstructions.beta f - 3 ≤CircuitObstructions.toffoliCount c
```

AffineOutput: 14 declarations

L094 `restores_append`
Source. `CircuitObstructions/AffineOutput.lean:12`
Block. 02 **Axioms.** propext

```
CircuitObstructions.restores_append {X W : Type} [DecidableEq W] (e : CircuitObstructions.Boundary X W)
 (c d : CircuitObstructions.Circuit W) (f g : X →Bool) (hc : CircuitObstructions.Restores e c f)
 (hd : CircuitObstructions.Restores e d g) : CircuitObstructions.Restores e (c ++ d) fun x => f x ^^ g x
```

L095 `canonical_target_not`
Source. `CircuitObstructions/AffineOutput.lean:20`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.canonical_target_not (n a : Nat) (x : Fin n →Bool) (b : Bool) :
 (CircuitObstructions.Gate.not (CircuitObstructions.canonicalBoundary n a).target).eval
    ((CircuitObstructions.canonicalBoundary n a).encode x b) =
  (CircuitObstructions.canonicalBoundary n a).encode x (b ^^ true)
```

L096 `canonical_input_to_target`
Source. `CircuitObstructions/AffineOutput.lean:34`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.canonical_input_to_target (n a : Nat) (i : Fin n) (x : Fin n →Bool) (b : Bool) :
 (CircuitObstructions.inputToTarget n a i).eval ((CircuitObstructions.canonicalBoundary n a).encode x b) =
  (CircuitObstructions.canonicalBoundary n a).encode x (b ^^ x i)
```

L097 `affine_restored_exists`
Source. `CircuitObstructions/AffineOutput.lean:45`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.affine_restored_exists (n a : Nat) (g : CircuitObstructions.BFun n)
 (hg : CircuitObstructions.DegreeLE 1 g) :∃
 d,
  CircuitObstructions.toffoliCount d = 0 ∧
   CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) d g
```

L098 `betaStar_exists`
Source. `CircuitObstructions/AffineOutput.lean:76`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_exists {n : Nat} (f : CircuitObstructions.BFun n) :∃
  r, CircuitObstructions.AffineSafeDegreeLE f r ∧∀(k : Nat), CircuitObstructions.AffineSafeDegreeLE f k →r ≤k
```

L099 `betaStar_spec`
Source. `CircuitObstructions/AffineOutput.lean:85`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_spec {n : Nat} (f : CircuitObstructions.BFun n) :
 CircuitObstructions.AffineSafeDegreeLE f (CircuitObstructions.betaStar f)
```

L100 `betaStar_le_of_bound`
Source. `CircuitObstructions/AffineOutput.lean:88`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_le_of_bound {n d : Nat} {f : CircuitObstructions.BFun n}
 (h : CircuitObstructions.AffineSafeDegreeLE f d) : CircuitObstructions.betaStar f ≤d
```

L101 `betaStar_le_inputs`
Source. `CircuitObstructions/AffineOutput.lean:92`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_le_inputs {n : Nat} (f : CircuitObstructions.BFun n) : CircuitObstructions.betaStar f ≤n
```

L102 `beta_le_betaStar`
Source. `CircuitObstructions/AffineOutput.lean:97`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_le_betaStar {n : Nat} (f : CircuitObstructions.BFun n) :
 CircuitObstructions.beta f ≤CircuitObstructions.betaStar f
```

L103 `betaStar_le_half_toffoliCount`
Source. `CircuitObstructions/AffineOutput.lean:101`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_le_half_toffoliCount (n a : Nat)
 (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
 (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
 CircuitObstructions.betaStar f ≤(CircuitObstructions.toffoliCount c + 1) / 2 + 1
```

L104 `betaStar_restoration_bound`
Source. `CircuitObstructions/AffineOutput.lean:112`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_restoration_bound (n a : Nat)
 (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
 (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
 2 * CircuitObstructions.betaStar f ≤CircuitObstructions.toffoliCount c + 3
```

L105 `toffoliCount_ge_two_betaStar_sub_three`
Source. `CircuitObstructions/AffineOutput.lean:119`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.toffoliCount_ge_two_betaStar_sub_three (n a : Nat)
 (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
 (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
 2 * CircuitObstructions.betaStar f - 3 ≤CircuitObstructions.toffoliCount c
```

L106 `canonical_restored_output_degree`
Source. `CircuitObstructions/AffineOutput.lean:126`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.canonical_restored_output_degree (n a : Nat)
 (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
 (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
 CircuitObstructions.DegreeLE (CircuitObstructions.toffoliCount c + 1) f
```

L107 `restored_degree_and_cover_bound`
Source. `CircuitObstructions/AffineOutput.lean:140`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.restored_degree_and_cover_bound (n a : Nat)
 (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire n a)) (f : CircuitObstructions.BFun n)
 (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary n a) c f) :
 CircuitObstructions.DegreeLE (CircuitObstructions.toffoliCount c + 1) f ∧
  2 * CircuitObstructions.betaStar f ≤CircuitObstructions.toffoliCount c + 3
```

DegreeChecks: 8 declarations

L108 `squareParity_add`
Source. `CircuitObstructions/DegreeChecks.lean:17`
Block. 02 **Axioms.** propext

```
CircuitObstructions.squareParity_add (f g : CircuitObstructions.BFun 2) :
  (CircuitObstructions.squareParity fun x => f x ^^ g x) =
    (CircuitObstructions.squareParity f ^^ CircuitObstructions.squareParity g)
```

L109 `affine_squareParity_zero`
Source. `CircuitObstructions/DegreeChecks.lean:22`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.affine_squareParity_zero (f : CircuitObstructions.BFun 2) (h : CircuitObstructions.DegreeLE 1 f) :
  CircuitObstructions.squareParity f = false
```

L110 `andTwo_degree_two`
Source. `CircuitObstructions/DegreeChecks.lean:39`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.andTwo_degree_two : CircuitObstructions.DegreeLE 2 CircuitObstructions.andTwo
```

L111 `andTwo_not_affine`
Source. `CircuitObstructions/DegreeChecks.lean:43`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.andTwo_not_affine : ¬CircuitObstructions.DegreeLE 1 CircuitObstructions.andTwo
```

L112 `andTwo_no_degree_one_safe_cover`
Source. `CircuitObstructions/DegreeChecks.lean:49`
Block. 02 **Axioms.** propext, Quot.sound

```
CircuitObstructions.andTwo_no_degree_one_safe_cover :
  ¬CircuitObstructions.CompleteSafeDegreeLE CircuitObstructions.andTwo 1
```

L113 `beta_andTwo`
Source. `CircuitObstructions/DegreeChecks.lean:64`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.beta_andTwo : CircuitObstructions.beta CircuitObstructions.andTwo = 2
```

L114 `betaStar_andTwo`
Source. `CircuitObstructions/DegreeChecks.lean:72`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.betaStar_andTwo : CircuitObstructions.betaStar CircuitObstructions.andTwo = 2
```

L115 `restored_andTwo_needs_toffoli`
Source. `CircuitObstructions/DegreeChecks.lean:78`
Block. 02 **Axioms.** propext, Classical.choice, Quot.sound

```
CircuitObstructions.restored_andTwo_needs_toffoli (a : Nat)
  (c : CircuitObstructions.Circuit (CircuitObstructions.CanonicalWire 2 a))
  (hR : CircuitObstructions.Restores (CircuitObstructions.canonicalBoundary 2 a) c CircuitObstructions.andTwo) :
  1 ≤CircuitObstructions.toffoliCount c
```

APPENDIX E

Claim coverage and theorem index

The source register has eight maintained claim groups. This reference expands the maintained arguments into individually numbered statements and adds the later register-product invariant. Prose statement counts and formal lemma counts have different meanings. The Lean guide lists all 115 formal declarations; the original theorem companion and its historical theorem map remain preserved.

E.1 Coverage of the maintained register

ID	Maintained subject	Results in this reference	Source location
C01	Exact cut-comparison cover	Theorem 14.1	Foundations F2
C02	Complete-cover bound and dual certificates	Results 15.1–15.7	Theory §1; Foundations F4
C03	Unbalanced-OR calibration	Results 16.1–16.4	Theory §2
C04	Optimal small restored oracles	Theorem 16.5	Theory §2
C05	Projection cost and safety	Results 17.1–17.5	Theory §3
C06	Faithful flat gluing dimension	Theorem 18.1; Corollary 18.2	Theory §4
C07	Weighted witness fingerprint	Results 19.1–19.5	Theory §5
C08	Overlap obstruction	Proposition 20.1; Counterexample 20.2	Theory §5

The foundational simulation and degree results, Theorem 13.1 and Lemma 13.2, support the active claims. The conditional status of Proposition 19.5 is unchanged from C07. Counterexamples are deliberately indexed as counterexamples, rather than presented as successful universal compression theorems.

E.2 Dependency structure

The main proof dependencies are:

$$\begin{gathered}\text{restored oracle convention} \implies \text{cut comparison},\\ \text{cut comparison} + \text{degree per multiplication} \implies \text{complete-cover lower bound},\\ \text{safe ideal} + \text{finite linear duality} \implies \text{evaluation-span certificates},\\ \text{monotone ideal} + \text{affine input/output changes} \implies \text{calibration formulas},\\ \text{affine restriction} + \text{quadratic polar rank} \implies \text{projection-cost lower bound},\\ \text{orthogonal idempotents} \implies \text{gluing dimension},\\ \text{Boolean coefficient identity} + \text{root bound} \implies \text{conditional fingerprint test}.\end{gathered}$$

The gluing-dimension theorem and fingerprint construction are separate from the circuit-cut theorem. No proof step identifies algebra dimension with unrestricted circuit size.

E.3 Complete numbered-result index

No.	Result	Scope	Page
13.1	Classical/restored simulation comparison	Base	220
13.2	Degree per multiplication	Base	220
13.3	Register-product degree invariant	Base / formal	221
14.1	Cut-comparison identity and complete coverage	C01	222
15.1	Basic size bounds	C02	223
15.2	Affine input invariance	C02	224
15.3	Affine-optimized complete-cover bound	C02	224
15.4	Dual certificate for failure of safe detection	C02	225
15.5	Finite support of a dual witness	C02	225
15.6	Safe monomial ideal for a monotone function	C02	225
15.7	Exact monomial oracle cost	C02	226
16.1	Raw calibration parameters	C03	227
16.2	Affine optimization of the one-packet parameter	C03	227
16.3	Affine optimization of complete coverage	C03	228
16.4	Restored cost interval for the calibration family	C03	228
16.5	Exact small restored-oracle calibration	C04	229
17.1	Affine subspace after multiplication restrictions	C05	230
17.2	Exact cost of a nondegenerate paired quadratic	C05	230
17.3	Exact nonlinear cost of the whole function	C05	231
17.4	Exact cost of the extracted grade	C05	231
17.5	Projection does not preserve safe support	C05	231
18.1	Dimension of a faithful multiplicative summary	C06	233
18.2	Scalar unital summaries are evaluations	C06	233
19.1	Exact witness fingerprint	C07	235
19.2	Top coefficient as a Boolean-cube sum	C07	235
19.3	Weighted top-coefficient identity	C07	235
19.4	Finite-field root bound	C07	236
19.5	Conditional randomized satisfiability test	C07	236
20.1	Disjoint-variable factorization	C08	237
20.2	Shared-variable multiplication fails	C08	237

E.4 Baseline domains reproduced by the full run

The exact mathematical outputs below were compared with the preserved baseline. The historical runner also reproduced its seven scientific output records. These finite ranges remain part of the evidence even where a universal written theorem accompanies them.

Evidence area	Executed domain in the baseline	Limit of the evidence
Small functions	All 276 functions on one, two, or three inputs; 4,240 scalar input/target cases; 2,344 cuts	Exact constructions plus proved lower bounds, not exhaustive circuit search
Four-input profiles	All 65,536 truth tables; all 168 monotone functions	Independent evaluation-span method checks only the declared 765 cases
Unbalanced-OR family	Designated truth tables through nine inputs; all affine masks through eight	No exhaustive ninth-input affine-mask claim

Evidence area	Executed domain in the baseline	Limit of the evidence
Projection coefficients	693,965 pair coefficients and nine exact rank cases, through 1,022 inputs	Full truth tables only at $m = 2, 3, 4$
Projection oracle checks	65,808 logical-input/target cases across the two constructed oracle families in the stated small domains	Large entries use symbolic formulas, exact rank, and construction counts
Weighted fingerprints	1,052,736 verifier/field-point evaluations	Enumerated verifiers have at most three witness bits
Independent coefficient extraction	20,544 cases	All field points for $m \leq 2$; restricted coordinate grid at $m = 3$
Disjoint product rule	4,096 exact controls	Does not extend the rule to shared variables
Scalar algebra maps	66,048 linear maps over the three specified witness/field domains	Controls the scalar special case; general dimension theorem has a written proof

The field used in the fingerprint checker is

$$\mathrm{GF}(16) = \mathbb{F}_2[t]/(t^4 + t + 1).$$

All mathematical arithmetic in these tests is Boolean, exact integer, or finite-field arithmetic. Runtime is not presented as a complexity benchmark. Equality of checksums certifies matching bytes, not mathematical truth.

E.5 Excluded inferences

The following boundaries are retained from the release's failure register. They are part of the closed scope, not a promise to finish every exploratory route.

Semantic elimination curvature. Exact existential eliminations commute. A nonzero semantic commutator cannot be asserted for those exact operators.

Expanded-state counting. Exponentially many residuals or obstruction descriptions can arise from a small circuit. The number of expanded objects is not their shared construction cost.

Small-case equality. Agreement of affine-optimized one-packet and complete-cover degrees through four inputs does not extend to all sizes. Chapter 16 gives an exact separating family.

Free homogeneous extraction. A projected grade may have greater nonlinear cost than the whole and can fail to remain safe. Chapter 17 keeps the input length and total affine cost explicit.

Unrestricted gluing. The exponential dimension theorem applies to a fixed faithful linear multiplicative map. It does not exclude succinct, nonlinear, or context-dependent algorithms.

Context-free trace multiplication. The disjoint-variable product rule fails at shared witness variables. A short expression containing an extraction operator is not an efficient implementation of that extraction.

APPENDIX F

Verification and proof boundaries

The arithmetic bridge and restored-circuit model have executable Lean companions. Their coverage is explicit: the arithmetic package formalizes the fixed twelve-dimensional realization, while the circuit package formalizes legal reversible gates, cut coverage, the degree invariant with sharing, and the numerical complete-cover bound.

Component	Formal scope	Current replay
Arithmetic bridge	96 named theorems in five modules	PASS: fresh build, axiom inspection, five kernel replays.
Circuit obstructions	115 indexed theorems in eight modules	PASS: fresh build, axiom inspection, eight kernel replays, eight rejected controls.

A compiler pass checks elaboration of the source. Axiom inspection records its logical dependencies. Kernel replay rechecks the emitted declarations with Lean's kernel. Rejected controls test that the checking procedure detects specified invalid conclusions. These operations do not formalize the remaining prose merely because it appears in the same volume.

The general-modulus arithmetic theory, symmetry simulation theorems, catalytic gate theory, and the remaining circuit claims retain their stated written or finite computational evidence. The open research programs remain listed in their own chapters and appendices. This volume does not assert unrestricted quantum advantage, a general circuit lower bound, or complete Lean coverage of all its results.

The symmetry treatment also supplies an adaptive discarded-state certificate. Its written proof bounds the error by the sum of discarded norms on the computed trajectory. That result does not itself prove that the required packets or norm calculations remain polynomial in size.

Glossary

Ambient realization
A representation or implementation in a larger permitted model, with its maps and costs specified.

Arithmetic selector
A diagonal choice of Fourier modes, transported to a signal projector by a unitary basis change.

Complete safe cover
A family of detectors supported on the positive inputs and detecting every positive input.

Gate set defect
The difference between minimum restricted and ambient implementation costs under a fixed gate alphabet and resource convention.

Residual
The component not retained by a specified projection or coarse representation. Its recovery requires additional data.

Restoration
The requirement that logical inputs and workspace return to their prescribed boundary values after computation.

Term	Meaning
Active packet	A symmetry sector or representation component reached by the input and gates.
Approximate packet	A subspace or representation that is not exactly invariant but has certified leakage.
Compressed packet	A packet whose states are represented by a compact classical data structure rather than explicit amplitudes.
Coherence block	$P_\alpha \rho P_\beta$ with possibly distinct left and right packet labels.
Equitable partition	A graph partition whose cells have uniform neighbor counts, yielding an invariant quotient subspace.
Global interference	Dependence of the output on coherent phase relations among many contributions or packets.
Invariant packet	A subspace mapped into itself by every gate in the declared circuit decomposition.
Leakage	Amplitude or probability transferred outside a proposed active representation.
Multiplicity space	Repeated-copy factor accompanying an irreducible representation in a group decomposition.
Orbit state	Normalized uniform superposition over one group orbit of basis states or graph vertices.
Packet transfer graph	Directed graph recording nonzero blocks $P_\alpha G P_\beta$.
Symmetry collapse	Exact or controlled replacement of a quantum evolution by a polynomial classical representation using symmetry or related compression.
Uniform circuit family	Circuits generated from the input by a deterministic polynomial-time procedure.

Sources and references

Source lineage

The four primary literature entries are provided in the accompanying reference fragment. Ramanujan's work supplies the arithmetic lineage. Vaidyanathan's papers establish the signal-processing setting of Ramanujan spaces and filter banks. The internal Komorebi lineage is *Stratified Universality for Resolved Representations*, v1.16.0, Proposition 21.11, printed page 189, with Boundary 21.12 on the distinction between character orders and coefficient-residue lanes.

The local authoritative source set for the finite example consists of `Worked_Example.md`, the original reproducibility verifier, the earlier four-module checked source and proof-boundary document, and the later five-module checked source with its verification records. Their exact filenames, source hashes, and evidence classification are listed in the release provenance and the Lean guide. The traditional paper supplies self-contained mathematical proofs, while those companion records make the implementation and formal scope inspectable.

Bibliography

[1] S. Ramanujan, *On certain trigonometrical sums and their applications in the theory of numbers*, Transactions of the Cambridge Philosophical Society **22**, no. 13 (1918), 259–276. Original-paper scan index: https://www.imsc.res.in/~rao/ramanujan/CamUnivCpapers/Cpaper21/page1.htm.

[2] P. P. Vaidyanathan, *Ramanujan-sum expansions for finite duration (FIR) sequences*, 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4933–4937. https://doi.org/10.1109/ICASSP.2014.6854540. Author manuscript: https://systems.caltech.edu/dsp/PPVSquare.pdf.

[3] P. P. Vaidyanathan, *Ramanujan Sums in the Context of Signal Processing—Part I: Fundamentals*, IEEE Transactions on Signal Processing **62**(16) (2014), 4145–4157. https://doi.org/10.1109/TSP.2014.2331617.

[4] P. P. Vaidyanathan, *Ramanujan Sums in the Context of Signal Processing—Part II: FIR Representations and Applications*, IEEE Transactions on Signal Processing **62**(16) (2014), 4158–4172. https://doi.org/10.1109/TSP.2014.2331624.

[5] Komorebi P., *Stratified Universality for Resolved Representations*, v1.16.0, Block 16, internal manuscript, 2026. Proposition 21.11 and Boundary 21.12, printed pp. 189–190. Preserved source: `Komorebi_Stratified_Universality_Foundations_v1.16.0_Block16.pdf`.

[6] S. Aaronson and D. Gottesman, *Improved Simulation of Stabilizer Circuits*, Physical Review A 70 (2004), 052328.

[7] S. Aaronson and A. Ambainis, *Forrelation: A Problem that Optimally Separates Quantum from Classical Computing*, SIAM Journal on Computing 47 (2018), 982–1038.

[8] S. Ball and R. Simoens, *Thirty-six quantum officers are entangled*, arXiv:2603.02334, 2026.

[9] E. Bernstein and U. Vazirani, *Quantum Complexity Theory*, SIAM Journal on Computing 26 (1997), 1411–1473.

[10] R. Jozsa and A. Miyake, *Matchgates and Classical Simulation of Quantum Circuits*, Proceedings of the Royal Society A 464 (2008), 3089–3106.

[11] B. Musto and J. Vicary, *Quantum Latin Squares and Unitary Error Bases*, Quantum Information and Computation 16 (2016), 1318–1332.

[12] S. A. Rather, A. Burchardt, W. Bruzda, G. Rajchel-Mieldzioć, A. Lakshminarayan, and K. Życzkowski, *Thirty-six Entangled Officers of Euler: Quantum Solution to a Classically Impossible Problem*, Physical Review Letters 128 (2022), 080507.

[13] Standard Schur–Weyl duality and representation-theoretic decomposition results are used in their finite-dimensional form.

[14] P. W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM Journal on Computing 26 (1997), 1484–1509.

[15] D. Simon, *On the Power of Quantum Computation*, SIAM Journal on Computing 26 (1997), 1474–1483.

[16] L. G. Valiant, *Quantum Circuits that Can Be Simulated Classically in Polynomial Time*, SIAM Journal on Computing 31 (2002), 1229–1254.

[17] K. P., *Quantum Transfer Blocks and Quantum Polyphase Transfer Calculus*, project research records, 2026.

[18] A. Acuaviva, A. Acuaviva, and P. Acuaviva, *CNOT-Distance is NP-complete under all-to-all connectivity*, arXiv:2608.03825, 2026.

[19] M. Amy and M. Mosca, *T-count optimization and Reed–Muller codes*, IEEE Trans. Inform. Theory **65** (2019), 4771–4784.

[20] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, *A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **32** (2013), 818–830.

[21] M. Bataille, *Quantum circuits of CNOT gates: optimization and entanglement*, Quantum Inf. Process. **21** (2022), Article 269.

[22] S. Bravyi and A. Kitaev, *Universal quantum computation with ideal Clifford gates and noisy ancillas*, Phys. Rev. A **71** (2005), 022316.

[23] A. Bu, E. Fan, and R. Joo, *Minimum synthesis cost of CNOT circuits*, Quantum Inf. Process. **24** (2025), Article 208.

[24] R. W. Carter, *Simple Groups of Lie Type*, Wiley, 1972.

[25] J. E. Christensen, S. F. Jørgensen, A. Pavlogiannis, and J. van de Pol, *On exact sizes of minimal CNOT circuits*, in *Reversible Computation 2025*, Lecture Notes in Comput. Sci. **15716**, Springer, 2025, pp. 71–88.

[26] J. Dehaene and B. De Moor, *Clifford group, stabilizer states, and linear and quadratic operations over* $\mathrm{GF}(2)$, Phys. Rev. A **68** (2003), 042318.

[27] S. Gong and A. Yu, *Explicit matrices over $\mathbb{Z}_2$ with CNOT and row complexity $4n - o(n)$ and local logic gates*, arXiv:2607.28598, 2026.

[28] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. thesis, California Institute of Technology, 1997.

[29] T. Goubault de Brugière, M. Baboulin, B. Valiron, S. Martiel, and C. Allouche, *Gaussian elimination versus greedy methods for the synthesis of linear reversible circuits*, ACM Trans. Quantum Comput. **2** (2021), Article 11.

[30] R. Hartshorne, *Algebraic Geometry*, Graduate Texts in Mathematics, vol. 52, Springer, 1977.

[31] J. E. Humphreys, *Linear Algebraic Groups*, Graduate Texts in Mathematics, vol. 21, Springer, 1975.

[32] J. Jiang, X. Sun, S.-H. Teng, B. Wu, K. Wu, and J. Zhang, *Optimal space-depth trade-off of CNOT circuits in quantum logic synthesis*, in *Proceedings of SODA 2020*, SIAM, 2020, pp. 213–229.

[33] S. F. Jørgensen, *Lower bounds for the CNOT-complexity of linear reversible operators*, arXiv:2607.22248, 2026.

[34] V. Kliuchnikov, D. Maslov, and M. Mosca, *Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates*, Quantum Inf. Comput. **13** (2013), 607–630.

[35] J. M. Landsberg, *Tensors: Geometry and Applications*, Graduate Studies in Mathematics, vol. 128, American Mathematical Society, 2012.

[36] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th anniversary ed., Cambridge University Press, 2010.

[37] J. Oxley, *Matroid Theory*, 2nd ed., Oxford Graduate Texts in Mathematics, vol. 21, Oxford University Press, 2011.

[38] K. N. Patel, I. L. Markov, and J. P. Hayes, *Optimal synthesis of linear reversible circuits*, Quantum Inf. Comput. **8** (2008), 282–294.

[39] S. A. Rather, N. Ramadas, V. Kodiyalam, and A. Lakshminarayan, *Absolutely maximally entangled state equivalence and the construction of infinite quantum solutions to the problem of 36 officers of Euler*, Phys. Rev. A **108** (2023), 032412.

[40] M. Saeedi and I. L. Markov, *Synthesis and optimization of reversible circuits—a survey*, ACM Comput. Surv. **45** (2013), Article 21.

[41] P. Selinger, *Quantum circuits of T-depth one*, Phys. Rev. A **87** (2013), 042302.

[42] J.-P. Serre, *Linear Representations of Finite Groups*, Graduate Texts in Mathematics, vol. 42, Springer, 1977.

[43] I. Shaik and J. van de Pol, *CNOT-optimal Clifford synthesis as SAT*, in *28th International Conference on Theory and Applications of Satisfiability Testing*, LIPIcs **341** (2025), Article 28.

[44] M. Webster, S. Koutsioumpas, and D. E. Browne, *Heuristic and optimal synthesis of CNOT and Clifford circuits*, arXiv:2503.14660, 2025.

[45] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, *Gaussian quantum information*, Rev. Mod. Phys. **84** (2012), 621–669.

[46] Komorebi P. *Foundations: restored computation and complete discrepancies*. `docs/FOUNDATIONS.md`, source release v1.2.0, September 21, 2026. Supplied repository source.

[47] Komorebi P. *Complete discrepancy covers, grade extraction, and exact gluing*. `docs/THEORY.md`, source release v1.2.0, September 21, 2026. Supplied repository manuscript.

[48] Komorebi P. *Claim register*. `docs/CLAIMS.md` and `docs/claims.json`, source release v1.2.0. Authoritative active groups C01–C08.

[49] Komorebi P. *Proof and computation boundary*. `PROOF_BOUNDARY.md`, source release v1.2.0. Controlling model and evidence qualifications.

[50] Komorebi P. *Release audit: 1.2.0*. `evidence/lean-block02/RELEASE_AUDIT.md`, September 21, 2026. Scoped archived formal verification. This edition also supplies fresh computational and document validation records.

[51] Komorebi P. *Reproduction contract*. `docs/REPRODUCIBILITY.md`, source release v1.2.0. Frozen finite results: `evidence/baseline/completeness.json`, `projection.json`, and `fingerprint.json` in the same baseline directory.

[52] Mathias Soeken. *Determining the Multiplicative Complexity of Boolean Functions using SAT*. arXiv:2005.01778v1, 2020. Background XOR–AND model and degree-bound context.

https://arxiv.org/abs/2005.01778v1

[53] Joan Boyar and Magnus Gausdal Find. *Multiplicative Complexity of Vector Valued Boolean Functions*. arXiv:1407.6169v3, 2018 revision of the 2014 preprint. Background affine-restriction and quadratic-complexity context.

https://arxiv.org/abs/1407.6169v3

[54] Chris Beck and Yuan Li. *Represent MOD function by low degree polynomial with unbounded one-sided error*. arXiv:1304.0713v1, 2013. Background one-sided polynomial-immunity terminology.

https://arxiv.org/abs/1304.0713v1