

Certified Target-Sufficient Readout Design and Exact Finite-Model Reduction

Jeremy Rodgers
Independent Researcher
www.everythingequation.com

27 October 2026

<https://doi.org/10.5281/zenodo.23721133>

Abstract

We give an exact computational realization of target-relative readout sufficiency for finite sources. A unique-extension circuit compiler turns failure of target descent into an explicit pair of sources, and counterexample-guided menu synthesis returns globally minimum-cost sufficient feasible selections. A rational branch-and-bound implementation supplies independently checkable lower and upper costs even when interrupted, including non-upward-closed feasibility constraints. A nonlinear autonomous register has sharp weighted predictive-completion and full-source reconstruction cost formulas, yielding unit-cost values b and $2b - 1$. Technical appendices give exact signed rational contraction with explicit precision accounting, minimal sufficient parity summaries, and context-certified tensor-wire quotients with executable integer crossing factors. The working implementations use conventional exact backends. Separate subexponential complexity corollaries are conditional on stated external preprint theorems; neither external algorithm is advertised as implemented. The results distinguish target sufficiency, access cost, retained information and recovery of the actual source.

1 Question, contribution and prior scope

This paper gives finite computational realizations of target sufficiency within Shadow Theory’s pre-existing source–readout programme. Its abstract descent and minimal-completion foundations are established in [10, 11]; the contributions here concern their constructive realization, constrained readout design and exact resource analysis. Let a finite source set S have readout $\pi : S \rightarrow T$ and target $R : S \rightarrow A$. Use the reachable readout space $T = \pi(S)$, including $T = \emptyset$ when $S = \emptyset$. A readout-only correct answer exists exactly when R is constant on each fibre of π : necessity follows by evaluating a factorization $R = \bar{R} \circ \pi$ on equal readouts; sufficiency defines $\bar{R}(t)$ as the common target value of the fibre. This is the existing target-relative descent criterion[10]. Restrict every extension space to its reachable image $e(S)$; inactive extra points are not source distinctions. An extension is a map $e : S \rightarrow E$ through which the original readout factors, $\pi = p_E \circ e$; it is target-sufficient when also $R = r_E \circ e$. The joint image $\{(\pi(s), R(s)) : s \in S\}$ is then the coarsest target-sufficient extension[11]: the map $e(s) \mapsto (\pi(s), R(s))$ is well-defined because both coordinates factor through e . Neither abstract statement alone determines which permitted measurements realize the completion or what they cost. When S is empty the criterion is vacuous, but no conditional prediction is thereby assigned to an impossible readout. This paper adds a circuit compiler, witness-producing separation, certified constrained synthesis and exact resource examples under declared finite menus. Generic implicit hitting-set methods are established[2, 5]; Boolean Fourier and sum–product methods are established[4, 6, 13]. Our claims concern the explicit source/target interfaces, proofs, cost formulas and implementations, not invention of those methods or a universal physical efficiency theorem. The source model in the

main constructions is classical and finitely described. Its menu can impose incompatibilities, but it is not a simultaneous-value model for incompatible quantum observables. Resource prices, available ports and prediction deadlines are premises. A returned compatible preimage does not identify which source actually occurred. Probabilities require a nominated measure; no Born-selection law follows from counting. The finite model does not identify Boolean gate workspace with reusable physical memory. A source assignment here, a quantum symbol map generating a word law, a single-copy detector outcome, and a retained validation trajectory are different mathematical objects. To use a physical intervention as a menu item one must first specify its admitted preparations, joint protocol, disturbance, accessible records and finite representation; deterministic values for incompatible observables are not supplied by the compiler. The appendices preserve related exact weighted and tensor reductions without making them dependencies of the finite synthesis theorem. Their factorization/context hypotheses and costs are explicit. This separates the unconditional finite algorithms from external asymptotic counting claims.

2 Constructive obstruction and minimum-cost predictive completion

Target descent and the coarsest sufficient extension provide the information criterion used below. This section adds an exact finite compiler, a minimum-cost realization procedure within a specified sensor menu, and an analytically solved nonlinear dynamical family. Its ETH consequence is conditional on the supplied counting theorem. The executable backend is a conventional bounded-cache DPLL counter, not an implementation or practical speed demonstration of that theorem.

The abstract criterion and coarsest joint-image completion are already established in Rodgers's target-relative completion papers[10, 11]. The algorithms below realize that criterion within finite circuit and menu representations. The finite synchronous application is not an extension of the Hilbert-space analytic hypotheses of projected-dynamics theory.

2.1 Exact circuit reduction and conditional complexity

An input model consists of N Boolean source bits and a Boolean circuit with G gates, each of fan-in at most two; unary and constant gates are included in G . Its designated outputs are an admissibility bit C , a present readout π and a target R . The source is $\mathcal{S} = \{x \in \{0, 1\}^N : C(x) = 1\}$. Source labels have no gauge redundancy in this finite model; an application with redundant coordinates must first specify its quotient or invariant target. Sensor outputs are additional nominated circuit wires and are included in the circuit size. Let q be the number of distinct target wires.

Theorem 2.1 (Parsimonious collision compilation). *There is a 3-CNF whose satisfying assignments are in bijection with ordered pairs $(x, x') \in \mathcal{S}^2$ satisfying*

$$\pi(x) = \pi(x'), \quad R(x) \neq R(x').$$

It uses at most $V = 2(N + G) + 2q$ variables and $O(V)$ clauses after duplicate output comparisons are removed. Reading the supplied output lists incurs their usual polynomial description cost. Every satisfying source pair has exactly one extension to the gate variables. The theorem also holds for the readout augmented by any selected sensor subset.

Proof. Take two copies of the circuit. For every gate, forbid each inconsistent input/output truth-table row. A binary gate contributes at most four clauses of width three, a unary gate at most two clauses of width two, and a constant a unit clause. Topological induction forces every gate variable uniquely once its source inputs are fixed. Add the two unit clauses $C(x) = C(x') = 1$ and two width-two clauses for equality of each distinct readout wire across copies. There are at most $N + G$ distinct such wires even

when the supplied output or sensor lists repeat them. For each distinct target wire form its two-copy XOR. A binary OR tree over those XORs is constrained to one; this adds at most $2q - 1$ gates and their unique values. If there is no target coordinate, insert an empty clause instead. Thus every source pair either has no extension or exactly one, and it has one precisely under the displayed condition. All comparison gates and normalization constraints are counted. $\square$

Corollary 2.2 (Conditional constructive descent test). *Assume the fixed-density part of the supplied OpenAI preprint’s Theorem 1.1[7]. The ordered collision count, a collision witness when one exists, and hence exact target descent can be computed in*

$$\text{poly}(\text{description length}) 2^{O(V/\sqrt{\log V})}$$

bit operations and polynomial space, with finite exceptional sizes handled directly. In particular $G = O(N)$, together with a polynomial-length supplied circuit/output/menu description, gives a subexponential-in-source-bits test even for globally dependent targets. For larger circuits the bound is in the compiled size V ; polynomial circuit size alone does not imply subexponential time in N .

Proof. The compiled width is three and clause density is bounded by an absolute constant. Apply Theorem 1.1 and recover source bits by variable-by-variable self-reduction, using only polynomially many counting queries. A zero count is exactly fibre constancy by Rodgers’s criterion. Gate extension uniqueness ensures there is no multiplicity correction. An empty source gives count zero vacuously; the implementation reports that case explicitly. $\square$

For fixed-locality admissibility and readout predicates and a binary fixed-local target, a direct truth-table construction instead uses the two source copies alone and constant-width clauses, as in Appendix C.2. A multi-coordinate target can be tested for descent coordinate by coordinate; summing those collision counts would double-count pairs that disagree in several coordinates and is not the union count. The compiler above is broader: global parity, nonlinear global feedback and circuits reusing intermediate values are permitted. Its gates for every menu item are compiled even if that item is not selected; unused-gate cone elimination is not assumed in the complexity or executable counts.

The same one-copy gate encoding counts compatible source/target values. For a specified finite prior weight $w(x) \geq 0$, the appropriate conditional answer is

$$\Pr(R = a \mid \pi = t, C = 1) = \frac{\sum_{x: C(x)=1, \pi(x)=t, R(x)=a} w(x)}{\sum_{x: C(x)=1, \pi(x)=t} w(x)}.$$

A zero denominator is reported as impossible or zero mass, never normalized. Uniform compatible-source counting is the special nominated prior $w \equiv 1$, not a consequence of the readout and not a Born-selection argument. The implementation’s arbitrary rational prior table uses exact enumeration; it is not labelled a fast weighted ETH backend.

2.2 Counterexample-guided minimum-cost realization

Let M sensor items g_j have nonnegative exact rational costs c_j . An item can return several bits. An independently specified decidable family $\mathcal{F} \subseteq 2^{[M]}$ declares jointly feasible selections; it may encode incompatibility, resource caps or prerequisites. It need not be upward closed. The realized readout for selection A is

$$e_A(x) = (\pi(x), (g_j(x))_{j \in A}), \quad c(A) = \sum_{j \in A} c_j.$$

The optimizer can also be generalized to another exactly specified finite cost function; the supplied implementation uses additive costs.

For a current collision (x, x') define $D(x, x') = \{j : g_j(x) \neq g_j(x')\}$. Every sufficient selection must hit this set. Maintain such sets, minimize $c(A)$ over $A \in \mathcal{F}$ that hit all of them, and call the exact collision oracle on e_A . If it returns a pair, append that pair's set and *reoptimize the entire selection*. If it certifies no pair, return A . If a returned set is empty, the whole menu is insufficient; if the constrained optimization is infeasible, no feasible sufficient selection exists.

Theorem 2.3 (Exact minimum-cost completion synthesis). *With exact optimization and separation, this algorithm terminates and returns either a globally minimum-cost sufficient feasible selection or a sound impossibility result within the declared menu/family. No tested candidate selection repeats. There are at most $|\mathcal{F}| \leq 2^M$ separation calls, with at most one subsequent infeasible optimization.*

Proof. Every true sufficient selection separates every returned target-disagreeing pair, so it satisfies every accumulated cut. The current restricted optimization value is therefore a lower bound on the true optimum. If the chosen candidate has no collision, it is a sufficient feasible selection of exactly that cost, giving a matching upper bound. If it has a collision, none of its selected sensors belongs to the new $D(x, x')$, so the new cut excludes this candidate permanently. Finiteness proves termination. An empty distinguishing set forbids every subset; otherwise infeasibility of the accumulated necessary cuts within $\mathcal{F}$ forbids every feasible sufficient selection. Each cut's witness, selected set and exact cost are retained in the trace. $\square$

This is an application of established counterexample-guided synthesis and implicit hitting-set optimization, not their invention. Close antecedents include Solar-Lezama's *Program Synthesis by Sketching* (2008), Chandrasekaran–Karp–Moreno–Centeno–Vempala, *Algorithms for Implicit Hitting Set Problems* (SODA 2011, DOI 10.1137/1.9781611973082.48), and Moreno-Centeno–Karp, *The Implicit Hitting Set Approach to Solve Combinatorial Optimization Problems with an Application to Multigenome Alignment* (2013, DOI 10.1287/opre.1120.1139)[2, 5, 16]. The source-specific contribution here is the typed collision compiler, admitted-cost interface, exact verification and application to a proved predictive-completion frontier.

The complexity is the sum of all optimizer and separation calls. An ETH-fast separation oracle does not make the whole synthesis subexponential: the finite selection family itself may be exponential. The prototype re-enumerates subsets exactly, with no performance-optimality claim. A feasible returned selection alone is not a certificate if the final oracle is incomplete. The supplied iteration cap returns **incomplete** explicitly. More generally, a certified lower bound L on a restricted optimum and a fully validated sufficient selection of cost U certify $L \leq \text{OPT} \leq U$. If each approximate optimizer guarantees cost at most α times the current restricted optimum plus δ , with $\alpha \geq 1$ and $\delta \geq 0$, final exact separation gives $U \leq \alpha \text{OPT} + \delta$. Unvalidated heuristic proposals or absence of a discovered counterexample supply no such guarantee.

Reoptimization matters even on a two-bit source. Let the target be $x_0 \vee x_1$, with no initial readout. Offer three singleton-indicator sensors for states 01, 10, 11 at unit cost, and raw-bit sensors x_0, x_1 at cost 6/5 each. Greedily adding a cheapest sensor for each returned collision buys all three indicators, cost 3. Reoptimization finds one raw-bit sensor and the missing singleton, cost 11/5. No cheaper selection can cover all three target-disagreeing state pairs: two unit indicators miss one and a lone raw bit misses one. The exact prototype and an independent subset check both reproduce this gap.

2.3 A sharp nonlinear predictive-completion resource frontier

This family turns the compiler into a constructive autonomous-dynamics application to target-relative descent[10] and a finite synchronous counterpart of exact projected-dynamics closure[12]. For $b \geq 1$

use $2b$ classical bits and one fixed clock tick, with source update

$$F_0(x) = x_0 \wedge \bigoplus_{i=0}^{2b-1} x_i, \quad F_i(x) = x_i \quad (i > 0), \quad \pi(x) = x_0.$$

All source states are admissible. The target is the next observed bit $R = \pi F$, which depends globally and nonlinearly on the source. The permitted readers are the individual hidden-bit ports x_i , $i > 0$, and disjoint pair-parity ports $p_j = x_{2j} \oplus x_{2j+1}$, $0 \leq j < b$. A direct global target reader is not in the menu.

The source is a declared synchronous classical logic model. The ports read the same classical register nondestructively; joint measurement is therefore admitted. The task predicts the future from records acquired before the first update; waiting to observe the target after that update does not satisfy the prediction task. Only the stated present-time ports are admitted. No laboratory prediction-deadline or timing advantage is certified by this discrete model. A pair-parity port is a separately admitted two-bit reader with a specified cost, not a free computational substitute for physically inaccessible bits. Clock, fixed logic and archival choices are external specified resources. No energy, continuous-time or quantum efficiency conclusion follows from the cost units.

Theorem 2.4 (Exact predictive versus full-source port cost). *Give raw port x_i positive cost u_i and pair port p_j positive cost v_j , with no other joint constraints. The minimum cost of a readout determining every future observed bit is*

$$C_{\text{pred}} = \min\{u_1, v_0\} + \sum_{j=1}^{b-1} \min\{u_{2j} + u_{2j+1}, v_j\}.$$

The minimum cost for full source reconstruction is

$$C_{\text{full}} = \min\{u_1, v_0\} + \sum_{j=1}^{b-1} \min\{u_{2j} + u_{2j+1}, u_{2j} + v_j, u_{2j+1} + v_j\}.$$

Under unit port costs, $C_{\text{pred}} = b$ and $C_{\text{full}} = 2b - 1$. The selected predictive ports can be chosen so their augmented readout itself has an autonomous update, rather than merely predicting one step.

Proof. Restrict to the admitted subfamily $x_0 = 1$. Then the target is global parity. Holding every other block fixed shows that the selected sensors in each block must determine its pair parity. In block zero, known x_0 makes either x_1 or p_0 sufficient and at least one necessary. In another block, a pair-parity port suffices; without it, both raw bits are necessary because changing an unmeasured bit flips the target while preserving every measured output. The blocks have disjoint sensor scopes, so their independently necessary minima add and the displayed selection attains the sum. For full source reconstruction, each other block has four independent states and any two of its three possible binary ports recover both bits; fewer than two cannot do so. This proves its sum.

Write $q = \bigoplus_{i>0} x_i$. If $x_0 = 0$ it remains zero. If $x_0 = 1$, the next bit is $1 \oplus q$: it remains one when $q = 0$, otherwise becomes zero and stays zero. Hence all future readouts equal R after the first step, so one-step sufficiency is all-future sufficiency. Selected raw ports are fixed under F . Pair ports for $j > 0$ are also fixed; if p_0 is selected, it updates as $p'_0 = p_0 \oplus x_0 \oplus R$. Thus the cost-attaining augmented readout closes autonomously. $\square$

The ideal canonical predictive completion has only three states,

$$(\pi, R) \in \{(0, 0), (1, 0), (1, 1)\},$$

with autonomous transitions $00 \mapsto 00$, $10 \mapsto 00$, $11 \mapsto 11$. Under unit costs the displayed minimum-cost port signature contains 2^{b+1} distinct values, and its postprocessing yields this three-state quotient. If

raw port records are retained, that larger archive is not discarded from the resource account. Abstract minimal distinctions and minimum physical-menu access cost are different quantities.

Under unit costs the gap approaches a factor of two, but that is a declared port-cost result, not a universal hardware advantage. If raw ports cost c and each pair port costs $2c$, the predictive optimum is $(2b - 1)c$, the same as the raw strategy. The exact weighted formula therefore exposes the premise on which a gain depends.

For the source family, the initial ordered collision count is 2^{4b-3} : only $x_0 = 1$ contributes, with 2^{2b-2} sources of each target value, counted in both orders. For $b = 1, \dots, 5$ the compiler and exact synthesizer reproduce costs 1, 2, 3, 4, 5 against raw/full-source costs 1, 3, 5, 7, 9, and yield autonomous completion signatures on every source state. Exhaustive whole-subset checks establish independent optima for the smaller sizes; twelve unequal rational-cost tests verify both weighted formulas. A uniform admitted-source prior gives $\Pr(R = 1 \mid \pi = 1) = 1/2$; separately nominated weights three on $R = 1$ and one on $R = 0$ give $3/4$. Neither prior is selected by the readout or by ETH.

2.4 Implementation, verification and scope

The implementation consists of `completion.py` and `verify_finite.py`. It uses exact integers/Fractions and a conventional DPLL counter with a cache bounded to 4096 states. It includes truth-table circuit compilation, a two-copy obstruction oracle, compatible target counts, exact finite-prior reference calculations, menu constraints, reoptimized synthesis and explicit incomplete/empty-source statuses. Inputs with malformed bits, sensor indices, costs or negative prior weights are rejected. Nonuniform prior tables are globally validated, including entries outside the queried fibre.

The tests compare randomized CNF counts, unique gate extensions, random nonlinear constrained models, every sensor subset, compatible target counts and finite optima against direct source enumeration. They also cover empty sources, impossible readouts, empty targets, infeasible hardware constraints and early stopping. The independent audit uses a distinct source-enumeration implementation. Numerical runtime measurements describe this prototype only; no ETH-backend speedup is claimed.

This work adds a constructive finite synthesis interface and a sharp menu-constrained dynamical example to the supplied baseline. Continuous CP feasibility is not replaced by an unrestricted Boolean decision problem. Source identification, source accessibility and source counting remain distinct.

3 Scalable completion design with checkable interrupted bounds

The circuit interface permits exact source-collision separation with exhaustive menu reoptimization. Here a conventional exact branch-and-bound optimizer replaces exhaustive subset enumeration without changing the source semantics. Its output includes a covering tree, rational dual bounds, genuine source-pair witnesses and a separately validated sufficient incumbent. The improvement is an effective and audited implementation of established integer-optimization ideas, not an implementation of either external ETH algorithm or a new general optimization principle.

3.1 Problem, admissibility and lower certificates

Let the menu be $[m]$, with nonnegative rational prices c_j . Each accumulated source collision supplies a set $D \subseteq [m]$ that every sufficient selection must intersect. A declared feasible family may impose pair incompatibilities, prerequisites $a \Rightarrow b$, a cardinality cap and an additional arbitrary pure decidable predicate. In particular, feasibility need not be upward closed. The implementation accepts that predicate without making unsound assumptions about its values on supersets or partial assignments.

A search node assigns some ports S present and some F absent. Propagation closes prerequisites and their contrapositives, propagates incompatibilities, enforces the cardinality cap, and forces the unique available element of a still-uncovered cut. An inconsistent node is infeasible. Write $\mathcal{D}_S$ for the cuts not yet hit and $U = [m] \setminus (S \cup F)$. For nonnegative numbers y_D with

$$\sum_{D \in \mathcal{D}_S: j \in D} y_D \leq c_j \quad (j \in U),$$

the following lower bound is valid:

$$L(S, F) = \sum_{j \in S} c_j + \sum_{D \in \mathcal{D}_S} y_D.$$

Excluded items are removed from the residual cuts. A cheap greedy dual packing constructs the displayed y_D exactly: process residual cuts and increase the current dual variable by the least remaining price capacity on that cut. This is a valid packing bound, not necessarily the optimal LP value.

Theorem 3.1 (Certified bounded completion optimization). *For the finite declared menu and feasibility family, best-first binary branch-and-bound with the displayed lower bound returns an exact optimum or a sound infeasibility conclusion when exhausted. At any interruption, an existing feasible optimizer incumbent of cost U_{opt} and the minimum of the open node bounds and U_{opt} certify the restricted optimum interval. The bound is valid for arbitrary non-upward-closed leaf feasibility predicates. A finite covering-tree certificate can verify the interval without enumerating every menu subset or rerunning the optimizer.*

Proof. An extension $A \supseteq S$ disjoint from F must select at least one element of every residual cut. Therefore

$$\sum_{j \in A \setminus S} c_j \geq \sum_{j \in A \setminus S} \sum_{D \ni j} y_D \geq \sum_D y_D.$$

All additional admissibility constraints only remove extensions, so ignoring them in this dual calculation is safe. Every propagation rule is a necessary consequence of the declared constraints or a residual one-element cut. Branching on a free item partitions all remaining extensions into those including it and those excluding it. No extension is silently lost. Once the current S is itself feasible and hits every cut, it is cheapest within that node: further ports have nonnegative cost. If S fails an opaque predicate, its supersets are still explored unless every variable is assigned. Thus non-upward feasibility is preserved. Nodes whose lower bound is no smaller than the incumbent are safely pruned. The remaining nodes, incumbent and already closed branches cover the finite family, establishing both the interrupted interval and exact termination statement. A checker replays the covering tree, propagation, each rational capacity inequality, each solution leaf and every pruning inequality. It need not solve another optimization. $\square$

3.2 Outer source synthesis and safe stopping

The outer counterexample loop uses the unchanged criterion

$$C(x) \wedge C(x') \wedge [e_A(x) = e_A(x')] \wedge [R(x) \neq R(x')].$$

An optimizer incumbent is not automatically a source-sufficient incumbent: it may hit all known cuts and miss a new source collision. Only a completed exact no-collision check promotes a feasible selection to an upper certificate for source completion. A user may provide such a selection initially, but it is verified rather than trusted. Its cost remains a real upper bound even if the run stops before finding another candidate.

Corollary 3.2 (Anytime source-completion interval). *Every optimizer lower certificate on genuine collision cuts is a lower bound on the minimum cost of a sufficient feasible selection. A feasible selection with a completed no-collision certificate supplies an upper bound. Matching bounds certify global minimum cost under the declared menu. Interruption, failure to find a collision with an incomplete oracle, and failure to find an optimizer incumbent do not establish source sufficiency or menu insufficiency.*

Proof. Every sufficient selection separates each genuine target-disagreeing pair and therefore obeys every stored cut. The restricted optimizer’s feasible family contains all sufficient feasible selections. The previous lower bound thus applies to the true optimum, while the validated incumbent attains its stated upper cost. Equality sandwiches the optimum. If a genuine pair has an empty distinguishing set, no menu subset can separate it. If the necessary cuts are infeasible, no sufficient feasible selection exists. These last two conclusions require the corresponding positive evidence, not an unfinished search. $\square$

The supplement exports the final covering tree and one source witness per necessary cut. The checker independently recomputes each witness’s admissibility, original-readout equality, target disagreement and exact distinguishing menu. It validates the tree and replays the declared exact no-collision oracle for the claimed upper selection. This is an oracle-relative certificate interface, not a new proof-logging SAT/UNSAT format. Opaque feasibility is a declared pure predicate supplied unchanged to both optimizer and checker. Certificate bounds and dual weights are exact integers or rational numbers, not in-memory floating-point values; masks and indices have integer types. The public JSON interface parses decimal tokens directly as rational numbers and rejects nonfinite constants before certificate validation. These input contracts prevent rounded arithmetic from being accepted as an exact lower-bound proof.

An explicit incomplete example has zero expanded nodes, a certified lower bound zero and a previously validated all-port upper bound $227/3$. Its status remains incomplete; that wide interval is reported rather than converted into an apparent optimal result. The tests also cover zero prices, contradictory cuts, cyclic prerequisites, cardinality limits, unknown separation status and non-upward parity feasibility.

A less trivial interrupted example has 60 explicit source rows and 24 ports. Twenty-four charged source-oracle queries construct a greedy sufficient starting design of cost $26/3$. A subsequent run limited to sixteen expanded nodes per restricted optimization uses twenty further source-oracle calls (including incumbent revalidation) and certifies

$$6 \leq \text{OPT} \leq 26/3.$$

Its status remains bounded, not optimal. Both the lower proof tree and the source-sufficient upper design survive interruption; the input and certificate are supplied. The forty-four source queries include preparation of the incumbent rather than hiding that work outside the comparison.

3.3 Scaling the nonlinear predictive-source application

For the nonlinear register,

$$F_0 = x_0 \wedge \bigoplus_{i=0}^{2b-1} x_i, \quad F_i = x_i \ (i > 0), \quad \pi = x_0,$$

we preserve the same raw and disjoint-pair parity menu. An exact structural separator avoids repeatedly solving its generic CNF when the recognized model promise holds. On $x_0 = 0$ the target is constant. On $x_0 = 1$, two sources with hidden-bit difference $h \in \mathbb{F}_2^{2b-1}$ collide exactly when

$$\ell_j(h) = 0 \quad (j \in A), \quad \bigoplus_i h_i = 1,$$

where ℓ_j is the selected raw or pair parity functional. Gaussian elimination either supplies h and a concrete source-pair witness or proves the linear system inconsistent. Conversely, any original collision supplies such a difference, so the separator is complete for this actual nonlinear source family. It is not a generic shortcut for nonlinear circuits.

All 292 menu subsets of the one-, two- and three-block models are checked against the inherited circuit/CNF separator. At $b = 20$, the method certifies the unchanged optimum 20 using a 59-port menu and 40 source bits, with 40 separation calls and 630 expanded optimization nodes in the recorded run. This extends practical problem size, not the already proved b versus $2b - 1$ cost theorem. The extra gain from $\text{GF}(2)$ recognition is identified separately from the optimizer gain.

3.4 Controlled comparison and non-closed-form examples

The old and new optimizers use the same exact separator within each controlled comparison. Two finite-table families have independently declared initial readouts, random binary target values, rational port prices, incompatibilities and prerequisites. These are explicit finite classical reader menus, not simultaneous assignments to incompatible quantum observables. There is no free target port. No closed-form optimum is assumed for the random constrained instances.

Model	Optimum	Old subset visits	New expanded nodes	Oracle calls old/new
Predictive $b = 4$	4	16,384	30	8/8
32 rows, 12 ports	15/2	57,344	128	14/16
40 rows, 14 ports	89/6	327,680	142	20/19

Subset visits and expanded nodes are different units; both are reported rather than presented as a hardware speedup. In the recorded controlled execution with Python allocation tracing enabled, elapsed times were about 0.376/0.017, 1.341/0.096 and 6.390/0.136 seconds for old/new respectively. These are reproducible synthetic prototype observations, not a universal runtime theorem or a statistical performance study. The old and new source oracle counts can differ because tied optimum candidates differ.

The new method also certifies exact costs 28/3 for a 60-row, 24-port constrained source and 23/2 for an 80-row, 32-port constrained source. Their proof trees and full input tables are included. The latter execution used 77 separation calls, 10,445 expanded nodes and about 39 seconds with allocation tracing. Larger certificates consume additional storage; retaining them is not free. Worst-case runtime, oracle calls and tree size can remain exponential. No general subexponential synthesis bound follows from either ETH paper.

The independent reference suite checks 160 exact optimization instances and 640 interrupted bounds against all subsets, 20 full source-table optima and their interrupted runs against direct source-pair enumeration, the 292 $\text{GF}(2)$ /CNF comparisons, and rejection of forged dual/global-bound certificates. The implementation uses exact `Fraction` arithmetic. The distinction between empty admitted sources, impossible readouts and zero prior mass remains with the inherited model interface; this optimizer does not change its prior or probability semantics.

The practical contribution to Rodgers’s finite target-descent interface is a checked design procedure that can now stop with an honest cost interval and solve larger constrained menus. Its antecedents are conventional weighted hitting-set branch-and-bound, LP-dual packing and implicit hitting-set synthesis, including Chandrasekaran et al., SODA 2011, DOI 10.1137/1.9781611973082.48, and Moreno-Centeno–Karp, 2013, DOI 10.1287/opre.1120.1139. No worldwide priority is asserted for those methods.

4 External-theorem dependency and reproducibility

The circuit reduction, synthesis correctness, branch-and-bound certificates and nonlinear cost frontier are proved here without any ETH-refutation premise. The conditional counting-time corollary invokes the fixed-density part of the supplied October25 OpenAI preprint, Theorem1.1[7]. Its rational tensor adapter invokes that preprint’s TheoremD.1. These are external theorem assumptions, not conclusions of the prototype checks. The cited official Lean scope description reports the weaker exponent $O_k(n(\log \log n)^2/\sqrt{\log n})$ and distinguishes the sharper printed bounds[8]. The official release describes ongoing external examination[9]. The supplied Anthropic manuscript reports formal checking but explicitly leaves statement correspondence and mathematical interpretation to readers[3], AppendixE. The supplied OpenAI manuscript is dated 17 October 2026, revised 25 October 2026; the supplied Anthropic manuscript is dated 26 October 2026. Their exact input hashes and inspected theorem/page locators are recorded in the supplement. We inspected the relevant printed interfaces and primary-source scope descriptions; we did not build those formalizations, establish identity between current repository files and every pinned version, or independently reconstruct their entire proofs. No sharper theorem is labelled machine-verified here solely because a related Lean statement exists. The supplement supplies inspected source code and exact integer/rational tests, source-pair witnesses, menu instances, covering-tree certificates, weighted examples and independent enumeration checks. Primary implementations are conventional DPLL, branch-and-bound, enumeration and variable elimination. Their finite execution counts and benchmark timings do not implement the preprints’ asymptotic algorithms or establish a practical general SAT speedup. Exact rational arithmetic, asserted checks and actual oracle access remain part of the reproducibility contract.

Reproduction. The top-level `build.py` builds this manuscript in a disposable source copy. `reproduce.py--check` validates the scoped input hashes, Python syntax and local-module closure; `reproduce.py--outputPATH` runs the declared checks in a disposable expanded tree and retains per-job receipts, logs and products. The supplement’s evidence guide distinguishes original arithmetic/certificate checks from focused input-schema regressions. Assertions are part of the verification contract: optimized Python execution with `-O` is rejected by the runner and public certificate module.

5 Limitations and relation to source-readout research

The abstract completion order and internal-identifiability ceilings are unchanged. Our finite tools make a nominated sufficiency question effective and optimize access within a declared menu; they do not select a physically separate or ontologically unique source. In continuous projected dynamics, the existing all-initial-state closure criterion remains a separate theorem[12]. A small tensor response quotient is not a quantum-memory minimum, and a computationally reconstructed distinction is not automatically a readable physical record. The conditional asymptotic transfers do not supply a better general SAT exponent than their external inputs. Wider novelty and publication priority require separate assessment.

Computational assistance

AI assistance was used for review and computational checking.

A Exact rational weighted contraction and target-dependent precision

The supplied OpenAI preprint’s Appendix D.1[7] contracts Boolean-wire networks over a truncated polynomial ring modulo a power of two. Rational signed weights are not an automatic instance of that coefficient ring. We give an explicit transfer, including signed recovery and all precision costs, and a sharp improvement when the target supplies an independently justified interval. The resulting source-likelihood interface is useful without implementing the paper’s asymptotic contraction algorithm.

Weighted sum–product and variable elimination are established methods. Scott–Sorkin[13], Sections 1–2 and 6, treat polynomial and ring-valued constraint satisfaction; Kschischang–Frey–Loeliger [4], Sections II–III, explain factorized marginalization and its relation to forward/backward algorithms. We do not claim weighted contraction, denominator clearing, modular reconstruction or dynamic programming as newly discovered techniques. The contribution here is their precise transfer to the supplied ETH theorem and a checked, typed interface for the nominated source–readout tasks.

A.1 A rational factor model with explicit arithmetic costs

Let $x \in \{0, 1\}^N$, and let $T_v(x|_{S_v}) \in \mathbb{Q}$ be explicitly supplied tables, with $|S_v| \leq a$. Negative entries are permitted for algebraic contraction, though they are not probabilities. Write

$$Z = \sum_{x \in \{0,1\}^N} \prod_{v=1}^F T_v(x|_{S_v}), \quad I = \sum_v |S_v|.$$

The table index convention, scope order and declared unused variables are part of the input. Let L be the total binary description length and let b bound numerator and positive-denominator bit length of each rational entry. For each tensor take the positive least common denominator q_v of its entries, and set

$$W_v = q_v T_v \in \mathbb{Z}^{2^{|S_v|}}, \quad Q = \prod_v q_v, \quad A_v = \max |W_v|, \quad B = 2^N \prod_v A_v.$$

Then $Z = Z_{\text{int}}/Q$ for an integer Z_{int} satisfying $|Z_{\text{int}}| \leq B$. For fixed a , each q_v and each integer entry has $O_a(b)$ bits, so $\log_2(B+1) = O(N + Fb)$. Zero tensors, isolated scalar tensors and unused source variables can be handled directly. In particular, u unused variables multiply the contraction by 2^u ; they are not silently omitted. In the theorem below all scalar entries and these factors of two remain in the same modular integer contraction. The interval therefore refers to the original Z_{int} , with no further restoration multiplier after decoding. An implementation may multiply known integer scalar factors modulo 2^P before decoding. If it instead decodes a smaller residual contraction, it must derive a new interval and denominator for that residual problem. Scalar products require their actual input/output-bit costs in either case.

Theorem A.1 (Exact rational transfer of OpenAI Appendix D.1). *Given a justified integer bound $L_0 \leq Z_{\text{int}} \leq U_0$, choose $P \geq 1$ with $2^P > U_0 - L_0$. The factor model can be transformed into a bipartite Boolean-wire tensor network with at most $4I$ wires and maximum degree $\Delta \leq \max\{a, 3\}$, retaining scalar vertices as described above. Its integer entries reduced modulo 2^P are instances of Appendix D.1 with truncation order one.*

Let E' and V' be the actual transformed wire and vertex counts and put

$$s = 2 + E' + V' + P + \Delta^2.$$

Conditional on the supplied OpenAI preprint’s Theorem D.1, the contraction has deterministic bit complexity

$$\text{poly}(s + L + N) 2^{O(s/\sqrt{\log s})}$$

and polynomial space. The recovered value is the exact signed rational Z . For bounded arity, fixed entry bit length and no isolated scalar factors or unused source variables, $s = O(E')$ with the crude bound above; thus the time is subexponential in wires. Growing precision must remain in s . Retaining many scalar vertices or unused-variable multipliers can increase V' or P without increasing E' ; merely preprocessing those factors does not remove their cost from the full-integer formulation above.

Proof. Replace every occurrence of a variable by a port of its factor. Enforce agreement of its ports by a tree of equality tensors of degree at most three. A variable appearing once uses the unary tensor $(1, 1)$, twice uses a binary equality tensor, and $d \geq 3$ occurrences use $d - 2$ trivalent equality vertices. Every assignment of the original bit extends uniquely to the tree; inconsistent leaf values give zero. There are at most $2I$ edges before bipartization. Subdivide each edge with a binary equality tensor, placing all original vertices on one side and all subdivision vertices on the other. This at most doubles the number of edges and preserves the contraction, because the two new edge bits must agree. The transformed maximum degree is at most $\max\{a, 3\}$.

Reduce the integer tables W_v modulo 2^P , leaving equality entries zero or one. Reduction is a ring homomorphism, so contraction gives $Z_{\text{int}} \bmod 2^P$. Exact table access and reduction cost polynomial time in the description and precision; this meets the entry-evaluator hypothesis of Theorem D.1. Our conservative s makes all its edge, vertex, degree and precision conditions hold, with fixed absolute constants. The theorem gives the displayed time and space. There is exactly one integer in $[L_0, U_0]$ with the returned residue, since two distinct such integers would differ by at least $2^P > U_0 - L_0$. Divide the recovered full integer by Q to obtain Z .

For fixed arity and entry bits, the crude interval $[-B, B]$ needs $P = O(N + F)$. Under the stated absence of isolated scalar factors and unused source variables, $N, F = O(I) = O(E')$ for a nonempty constructed network; its vertex and degree bounds have the same order. This proves the last assertion. Bounded exceptional sizes are handled directly as in Theorem D.1. $\square$

The generic transfer is not uniformly subexponential in the original source bits if an encoding introduces many wires or if P grows much faster than the wire count. For example, an $O(E' \log E')$ padding parameter does not give a subexponential-in- E' bound from the displayed formula. Direct enumeration, low-treewidth elimination or a small transfer matrix may already be faster. The implementation reports these distinctions rather than promising a practical asymptotic speedup.

A.2 A sharp target-relative arithmetic interface

Theorem A.2 (Interval-assisted reconstruction and target descent). *For all integers in a nonempty interval $[L_0, U_0]$, reduction modulo $M = 2^P$ is injective if and only if $M > U_0 - L_0$. Given a residue r and this promise, the unique answer is*

$$z = r + M \left\lceil \frac{L_0 - r}{M} \right\rceil, \quad r \in \{0, \dots, M - 1\},$$

provided $z \leq U_0$; otherwise the residue contradicts the interval. For a nominated target f on an admitted integer set $\mathcal{S}$, knowing only $z \bmod M$ determines $f(z)$ exactly if and only if f is constant on each residue class intersected with $\mathcal{S}$.

Proof. An interval of width smaller than M contains no distinct pair differing by a nonzero multiple of M . If its width is at least M , the two integers $L_0, L_0 + M$ both belong to it and have the same residue. The formula chooses the least integer in the residue class not below L_0 , proving recovery and its exact failure test. For the target statement, a residue-only answer must take one value on each class. Constancy makes that value well defined and sufficient, while two unequal target values in one class preclude it. $\square$

Thus it is the width of a justified interval, not its distance from zero, that determines full-value precision. A coarse target can need still fewer bits: parity needs $P = 1$ on any interval. Conversely the target $f(z) = \mathbf{1}\{z = 0\}$ on $\{0, \dots, U_0\}$ needs $2^P > U_0$ whenever $U_0 \geq 1$. Otherwise zero and 2^P have identical residues and different answers. A generic decision-only contraction therefore cannot drop this precision merely because it returns one bit. If the admitted interval already excludes zero, that particular decision is immediate.

The residue criterion is an instance of Rodgers’s target-relative descent principle. It supplies a sharp arithmetic adapter, not a new source of information. The interval must be justified independently; it cannot be inferred from the single modular residue. The code deliberately labels a caller-supplied interval as a premise. An incorrect narrow interval can silently select the wrong congruent integer, which no residue-only algorithm can prevent.

A.3 Completed record-law calculation

For a concrete nonnegative source, take hidden bits $X_0, \dots, X_4$, initial distribution $(1/2, 1/2)$ and transitions

$$P = \begin{pmatrix} 4/5 & 1/5 \\ 1/5 & 4/5 \end{pmatrix}.$$

At each tick emit $Y_t = X_t$ with probability $6/7$ and its complement with probability $1/7$, independently conditional on the hidden trajectory. These preparations and emission laws define a conventional finite stochastic source. They are not inferred from a readout or a Born-selection argument. For the retained word $y = (1, 0, 1, 1, 0)$, factor contraction and an independently implemented forward recursion both give

$$\Pr(Y = y) = \frac{848}{60025}, \quad \Pr(X_0 = 0 \mid Y = y) = \frac{1501}{7420}.$$

The second is a posterior under the nominated source measure, not an identification of the actual hidden initial bit.

Clearing the ten factor denominators gives $Q = 21\,008\,750$ and scaled answer $296\,800$. The naive magnitude calculation supplies $B = 63\,700\,992$, so the symmetric integer interval requires $P = 27$. The initial distribution and transition rows sum to one, and each emitted word factor is at most $6/7$. Consequently the event probability is between zero and $(6/7)^5$, independently of computing its exact value. This yields the tighter scaled interval

$$0 \leq Z_{\text{int}} \leq \left\lfloor Q(6/7)^5 \right\rfloor = 9\,720\,000,$$

requiring only $P = 24$. Both modular evaluations reconstruct the same answer. This is a three-bit precision saving for a specified task; it is not an empirical speedup or a new general worst-case ETH bound.

For arbitrary nonnegative source factors, the adapter also supports exact event indicators and supplied record likelihoods. It normalizes numerator and denominator separately and returns an explicit undefined status if the conditioning weight is zero. Signed tables are rejected by the probability interface even though the algebraic contraction backend accepts them. No probability is assigned to an impossible observation, and no arbitrary uniform prior is silently introduced.

A.4 Verification, backends and an easy-instance ceiling

The reusable `weighted.py` exposes exact rational factor evaluation, ordinary variable elimination, integer denominator clearing, modular enumeration, interval decoding, posterior queries and a Boolean-wire network converter. It uses Python’s exact integers and `Fraction`. Its backends are not an implementation of the ETH paper’s recursive contraction theorem.

`verify_weighted.py` checks 140 random signed rational factor models against independently organized enumeration, modular evaluation and variable elimination. It checks 3 400 signed interval decodings, the sharp lower-precision failures, parity-versus-zero target counterexamples, and explicit port-network conversions including high incidence, isolated scalars and unused variables. It also rejects signed probability measures and handles impossible conditioning.

As a nontrivial signed cyclic example, use the same matrix on every edge,

$$M = \begin{pmatrix} 1/2 & -2/3 \\ 3/5 & -7/11 \end{pmatrix}, \quad Z_n = \sum_{x \in \{0,1\}^n} \prod_i M_{x_i, x_{i+1}}, \quad x_n = x_0.$$

An independently computed matrix power gives $Z_n = \text{Tr}(M^n)$, which agrees with the adapter at $n = 3, 5, 6, 40$. The small cases also agree with exhaustive and modular enumeration. For $n = 40$, ordinary elimination uses 618 factor-entry evaluations with maximum bucket scope three, where exhaustive enumeration would inspect 2^{40} assignments. This is a verified operation-count advantage of an established low-width method; the transfer-matrix solution is also polynomial. It is specifically evidence against replacing every easy source calculation with the general ETH algorithm.

The universal claim is the conditional theorem transfer and the exact reconstruction proof. Finite diagnostics check the implementation, not the ETH paper’s whole proof. We do not establish continuous CP feasibility, a new trine-source resource bound, quantum outcome selection or a consciousness mechanism. Each would require its own additional interface.

B Target-sensitive summary preprocessing and a completed restricted improvement

The supplied OpenAI preprint’s summary routine (Lemma 3.2, pp. 22–23) scans integer tuples in a box of size $(2D + 1)^c$. Its Definition 3.1 requires rational-linear replacement forms to be invariant under kernels of the specified integer measurement matrices. The following construction does not replace those conditions by an unproved nonlinear invariant. It preprocesses a declared finite boundary response *before* invoking a general contraction method, and optimizes the retained information for a specified output task. A concrete rational source example then reduces an exact weighted calculation’s arithmetic work from quadratic to linear in its horizon. This is a restricted reciprocal benefit, not an improved ETH worst-case exponent.

The algebra uses established Boolean Fourier analysis [6] and finite-state dynamic programming. The contribution relative to the source–readout completion baseline is an executable synthesis of minimal parity summaries, explicit source/readout and retained-record conditions, and the complete task-specific cost comparison. It is not a new general Fourier theorem, the first use of dynamic programming or a publication-priority claim. It specializes the canonical target-relative completion principle rather than adding a new ontological premise. Rodgers’s target-family completion proposition and relation-versus-target completion theorem already give the abstract coarsening and its universal property[11]. The parity synthesis below is a restricted executable realization and cost analysis, not a stronger abstract minimal-completion theorem. For the boundary example the initial readout can be constant; when a nonconstant original readout must be retained, its coordinates must remain in the sufficient response rather than silently being dropped.

B.1 The finite pushforward identity and its cost

Let X be a finite boundary domain, $\mu : X \rightarrow \mathbb{Q}$ a supplied weight function, $F : X \rightarrow \mathbb{Q}^q$ a response, and $h : X \rightarrow Y$ a summary such that $F = \bar{F} \circ h$. Then

$$\sum_{x \in X} \mu(x) F(x) = \sum_{y \in h(X)} \bar{F}(y) M(y), \quad M(y) = \sum_{x: h(x)=y} \mu(x). \quad (1)$$

This follows by partitioning the finite sum into disjoint fibres. Signed rational weights are allowed; when weights are probabilities, the right side is the ordinary pushforward probability calculation. Its cost is the cost of constructing or evaluating $\bar{F}$ and computing M , plus the final sum. A small summary alone does not make an arbitrary weight function easy to push forward.

The group-product version clarifies the link to Lemma 3.2. Partition finite assignments into $x_1, \dots, x_r$ and choose summaries $h_g(x_g)$. Suppose factor product A_g depends only on x_g and the other summaries. For an attainable tuple $y = (y_1, \dots, y_r)$ define

$$B_g(y) = \sum_{x_g: h_g(x_g)=y_g} A_g(x_g, y_{-g}).$$

Distributivity gives

$$\sum_{x_1, \dots, x_r} \prod_g A_g(x_g, h_{-g}(x_{-g})) = \sum_{y \in \prod_g h_g(X_g)} \prod_g B_g(y).$$

This is a finite-fibre identity, not a new analytic assumption. Nonlinear Boolean summaries are admitted only after proving this factorization for the actual response. Computing the image, fibre sums and representatives is charged. In particular, one may not use the identity to assert that an arbitrary parity summary supplies the rational-kernel condition for the ETH paper's Taylor-shifted forms at every interpolation node.

B.2 An exact minimum within the parity-summary class

Write $\chi_a(y) = (-1)^{a \cdot y}$ for $a, y \in \mathbb{F}_2^b$. Every vector-valued rational response has the unique Walsh expansion

$$F(y) = \sum_{a \in \mathbb{F}_2^b} \hat{F}(a) \chi_a(y), \quad \hat{F}(a) = 2^{-b} \sum_y F(y) \chi_a(y).$$

Let K_F be the binary linear span of the nonzero-frequency masks with $\hat{F}(a) \neq 0$; the zero mask has no effect on its rank.

Theorem B.1 (Minimal sufficient parity summary). *On the full Boolean boundary cube, F factors through $y \mapsto Hy$ over $\mathbb{F}_2$ if and only if*

$$K_F \subseteq \text{row}_{\mathbb{F}_2} H.$$

Thus the minimum parity-summary rank is $\dim K_F$. A basis of K_F and the coordinates of each supported frequency in that basis supply an exact constructive certificate. For any declared rational linear output projection L , $K_{LF} \subseteq K_F$; strict inclusion can remove summary coordinates required by the full response.

Proof. If F factors through H , it is unchanged by translation by every $u \in \ker H$. Fourier uniqueness gives $\hat{F}(a) = \chi_a(u) \hat{F}(a)$, so each supported a is orthogonal to $\ker H$ and belongs to $\text{row } H$. Conversely, if every supported a lies in that row space, each character χ_a is a function of Hy , hence so is their sum. A basis gives attainment, and any sufficient row space has dimension at least $\dim K_F$. Applying L to the expansion can annihilate coefficients but cannot introduce a new frequency, proving the final statement. All arithmetic for constructing a basis is exact binary Gaussian elimination; no numerical rank threshold is involved. $\square$

If s masks are supplied sparsely, basis synthesis takes polynomial work in s, b, q and coefficient bit length, without a full cube scan. Obtaining the sparse expansion can nevertheless require exponential work. The minimum is only among parity summaries, not all nonlinear readouts: Hamming weight on b bits has Walsh-support rank b but only $b + 1$ output values. Restrictions to an admissible proper subset of the cube can also change the minimum, so the full-domain hypothesis must not be dropped.

B.3 A positive stochastic response and its requested marginal

For a boundary word $y = (y_1, \dots, y_b)$, let g be a noisy record of its global parity and let $z_1, \dots, z_{b-1}$ be noisy records of the individual first $b - 1$ bits. Specify the response law

$$P(g, z \mid y) = \frac{1 + \alpha(-1)^{g + \sum_i y_i}}{2} \prod_{i=1}^{b-1} \frac{1 + \beta(-1)^{z_i + y_i}}{2}, \quad \alpha = \frac{1}{2}, \quad \beta = \frac{1}{3}. \quad (2)$$

Every factor is strictly positive, and independent private noise makes the normalized joint record law. The source and fresh noise stock are independent. This is an ordinary classical stochastic model with explicit rational probabilities. It is not a quantum measurement construction or a selection of a probability law from geometry. Generating g at the deadline requires a parity accumulator and a clock; any history-correlated controller available to later production belongs in the physical causal-memory account. The workspace comparison below concerns the mathematical calculation of a probability, not a reduction of that physical apparatus or of its retained archive.

The nominated full response can be the b individual zero-record probabilities, rather than its exponentially large joint table:

$$F(y) = \left(\frac{1 + \alpha\chi_{\mathbf{1}}(y)}{2}, \frac{1 + \beta\chi_{e_1}(y)}{2}, \dots, \frac{1 + \beta\chi_{e_{b-1}}(y)}{2} \right).$$

Its supported masks $\mathbf{1}, e_1, \dots, e_{b-1}$ span all b dimensions, so every parity summary sufficient for this full response has rank b . The requested target is only $P(g = 0 \mid y)$, obtained by the first-coordinate projection. Its minimum parity-summary rank is one, with the two possible summaries $\sum_i y_i \bmod 2$. Neither the nuisance outcomes nor their physical records are erased by this mathematical calculation; they are unread by the nominated target. Computing their complete law remains a different task.

There is an exact compositional elimination before any ETH deformation:

$$\sum_{z_i=0}^1 \frac{1 + \beta(-1)^{z_i}(1 - 2y_i)}{2} = 1.$$

The equality holds as a rational polynomial identity in y_i , not just on Boolean inputs. Thus tracing these independent noise records locally costs $O(b)$ and yields the target response without constructing a 2^b -entry output table. Conditional or adaptive access to the z_i would change the task and invalidate this omission.

B.4 An end-to-end exact cost comparison

Use a stationary binary Markov source, with one clock tick per bit,

$$\tau = (4/7, 3/7), \quad T = \frac{1}{12} \begin{pmatrix} 9 & 3 \\ 4 & 8 \end{pmatrix}.$$

The initial stock has the displayed distribution and is independent of future transition noise. All transition weights are positive, hence the full Boolean boundary is admissible. The task is one exact

predicted record probability at horizon b :

$$p_b = P(g = 0) = \frac{1 + \alpha \mathbb{E}[(-1)^{\sum_i Y_i}]}{2}.$$

A fair integer-summary comparator keeps Hamming weight $W = \sum_i Y_i$, not the needlessly large identity summary. W is a single integer-linear row with $b + 1$ attainable values (the paper’s symmetric box would contain $2b + 1$). It is sufficient for this target. A standard exact count-histogram dynamic program stores last bit and current weight. At length t it has $2t$ nonzero integer cells. Summing all outgoing transitions over $t = 1, \dots, b - 1$ requires exactly

$$U_W(b) = 2b(b - 1)$$

multiply-by-constant/add updates.

The target quotient instead retains last bit and parity. It has four nonzero cells from length two onwards, and requires exactly

$$U_{\text{parity}}(b) = 8b - 12 \quad (b \geq 2).$$

Both programs start from integer weights 4, 3, use transition numerators 9, 3, 4, 8, and share final denominator $D_b = 7 \cdot 12^{b-1}$. If their histogram is H , they return the same exact rational

$$p_b = \frac{2D_b + \sum_s (-1)^s H(s)}{4D_b}.$$

For the weight program s is an integer count; for the target program it is its parity. Equivalence follows either by summing count states modulo two at each update or directly from Eq. (1).

Each integer has $O(b)$ bits and updates use only multiplication by a fixed small integer and addition. Thus the displayed implementations have bit-work upper bounds $O(b^3)$ versus $O(b^2)$ and integer-workspace bounds $O(b^2)$ versus $O(b)$, respectively. These are explicit algorithm bounds, not lower bounds against all possible algorithms. A specialized signed transfer-matrix method can also exploit the same target and need not form a full weight histogram.

b	W updates	parity updates	W live cells	parity live cells
8	112	52	16	4
32	1984	244	64	4
128	32512	1012	256	4
256	130560	2036	512	4

At $b = 256$ the exact update ratio is $32640/509 \simeq 64.13$. The gain includes the inexpensive local output projection and the pushforward calculation; it is not merely a comparison of summary cardinalities with their construction cost omitted. Wall-clock timings are diagnostic and platform dependent. Both methods compute the same one target probability under the same fixed rational source law. The factored input has length $O(b)$; `compile_target_record` checks the local nuisance cancellation with $2(b - 1)$ coefficient additions and constructs the two target Walsh coefficients directly. It does not first build the dense full response or perform an exponential Fourier transform. The table counts DP updates separately from this common linear preprocessing. General sparse Fourier synthesis can cost more. This finite-domain integer-summary comparator is not asserted to satisfy Definition 3.1 for the ETH paper’s rationally deformed polynomials either; the shared object here is its summary-enumeration stage, with a fully proved finite response factorization.

B.5 Verification, reciprocal status and limits

The supplied code compiles sparse vector Walsh data to a basis and an exact frequency-coordinate certificate. Verification checks 254 response instances, 6060 complete small parity-map tests against independent fibre equality, direct source-word enumeration for lengths one through seven, and exact agreement of the two integer DPs through length 256. It also checks the Hamming-weight counterexample to unrestricted nonlinear minimality. The code is a conventional exact preprocessor and reference algorithm; it is not the paper’s external ETH counter or contraction routine.

This supplies an explicit answer to the reciprocal question at a restricted level: target-relative completion can remove unnecessary summary detail and improve a declared weighted calculation without recovering its full response distribution. It does not prove a new general ETH exponent, improve the best known algorithm for this simple Markov family, or show that the ETH paper overlooked a universally smaller sufficient summary. To use the parity quotient inside that paper’s own recursive Taylor/mode computation, one would additionally need a factorization and coefficient certificate at every transformed node. No such universal claim is made.

The result changes computational organization, not physical access. No individual hidden source bit becomes readable through faster marginal calculation. The source, Markov law, independent noise, target projection, clock and unread-record permissions remain assumptions. No quantum resource minimum, Born-selection law or consciousness condition changes.

C Context-relative crossing compilation and explicit integer pieces

The target-summary construction optimizes a finite response summary and a specified weighted dynamic programme. It does not itself implement a component of Anthropic’s crossing recursion. Here we supply two checked components: an exact context-quotient network compiler, including its effect on the actual eight-square gadget, and explicit integer factorizations completing the single-bit rank observation in Appendix H of the cited preprint [3]. The quotient principle, Bell basis and single-bit identity are established mathematics. The contribution is their precise executable interface, coefficient bounds, proof of admissible contexts and demonstrated component costs. It is not a new crossing identity or an improvement of the paper’s strongest general exponent.

C.1 The full context contract and exact scalar recovery

Consider a closed integer tensor network on a graph $G = (V, E)$, with one string $z_e \in \mathbb{F}_2^{\lambda_e}$ per edge. The tensor at a vertex reads all its incident edge strings. On each edge supply a full-row-rank binary map $H_e : \mathbb{F}_2^{\lambda_e} \rightarrow \mathbb{F}_2^{r_e}$ and supply the vertex tensor in the form

$$W_v((z_e)_{e \ni v}) = \overline{W}_v((H_e z_e)_{e \ni v}).$$

This is a representation promise, not a claim inferred from a few tensor queries. The admitted surrounding contexts are *all* integer or rational vertex functions of these quotient values, with arbitrary dependencies among the incident quotient coordinates.

Theorem C.1 (Context-certified wire quotient). *For the declared context class, put $D = \sum_e (\lambda_e - r_e)$. Then*

$$Z = \sum_z \prod_v W_v(z) = 2^D \sum_{y \in \prod_e \mathbb{F}_2^{r_e}} \prod_v \overline{W}_v(y) = 2^D \overline{Z}.$$

The replacement is valid uniformly over every admitted surrounding context, including signed ones. It requires no enumeration of the discarded fibres. The factor 2^D must be restored for the full value; it is not a discarded normalization convention.

Proof. Each H_e is surjective and has $2^{\lambda_e - r_e}$ preimages of each output. For each full quotient assignment y , every summand in its fibre has the same vertex product. Edge fibres are independent Cartesian factors, so their common cardinality is 2^D . Sum over y . $\square$

A binary right inverse is found by Gaussian elimination in polynomial work in the supplied matrices. The code checks rank and constructs lifted basis vectors; it does not enumerate 2^{λ_e} candidates to find a lift. If dense original tables, rather than quotient tables or a proved factorized evaluator, are supplied, establishing the representation can require scanning those tables. That cost cannot be omitted. Computing $2^D \bar{Z}$ writes up to D additional output bits; denominators and output lengths are charged. The context promise is necessary: on one two-bit edge let H retain only the low bit and let both endpoint tensors be $[z = 0]$. The original contraction is one, whereas replacing each by its values at representatives 0, 1 and restoring the two-element fibre factor gives two. The tensors do not factor through H ; representative agreement alone was insufficient.

This theorem concerns a computational representation of a nominated boundary response. It does not remove an accessible physical archive or shrink a causal quantum memory. Exact symmetry-aware tensor representations are well established[15]; their resource minima can depend on topology and admitted symmetry[14]. No unrestricted minimum-bond-dimension assertion follows here.

C.2 Target projection can spoil an inherited decomposition

In the paper's Eq. (27), for three-bit strings p, q , write

$$\sigma(p, q) = (-1)^{\sum_{i \leq j} p_i q_j + q_1 p_2 p_3 + p_1 q_2 p_3 + p_1 p_2 q_3}.$$

Its Eq. (30) defines

$$R = [x \oplus y \oplus u \oplus v = 0] \sigma(x, y) \sigma(v, u), \quad C = [x \oplus y \oplus u \oplus v = 0] \sigma(x, u) \sigma(v, y),$$

with $R + C = 2[x = v][y = u]$. Row indices are (x, y) and column indices (u, v) ; the alternative split uses (x, u) and (y, v) . The ranks of R in those two directions are $(8, 64)$, and those of C are $(64, 8)$.

Now admit every surrounding context that depends on each three-bit boundary only through its low bit, $H(x) = x \bmod 2$. Let $\tilde{R}$ and $\tilde{C}$ be the sums of the original pieces over each four-boundary fibre. In row order 00, 01, 10, 11, exact enumeration gives

$$\tilde{R} = 8 \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 1 & 0 & 0 & 2 \end{pmatrix}, \quad \tilde{C} = 8 \begin{pmatrix} 2 & 0 & 0 & -1 \\ 0 & 1 & 2 & 0 \\ 0 & 2 & 1 & 0 \\ -1 & 0 & 0 & 2 \end{pmatrix}.$$

Both pieces have rank four in *both* directions. Their sum is $32S$, where $S(x, y, u, v) = [x = v][y = u]$ on single bits. The factor 32 is 2 from the paper's identity times 4^2 original crossed-string lifts. It is not 4^4 , because the crossing constrains two pairs.

Put $a(0, 1) = 1$, $a(1, 0) = -1$, and $a(0, 0) = a(1, 1) = 0$. The identity already noted in the paper's Appendix H is

$$S = I - A, \quad I = [x = u][y = v], \quad A = a(x, y)a(u, v).$$

Its integer factors are explicit: I is one product at the column split and a four-term coordinate-selector sum at the row split; $-A$ is one product at the row split and a four-term coordinate-selector sum at the column split. Thus recomposing the *same* fibre-summed target as $32I - 32A$ gives directional ranks $(4, 1)$ and $(1, 4)$. Each piece's directional bit sum is two, compared with four for either projected octonion piece. There are still two terms. The new scalar 32 costs one more magnitude bit than the largest projected entry 16; this growth is accounted for rather than interpreted as a probability.

The equality is proved on all 16 basis contexts of the four retained bits, which by linearity is every rational context in the declared class. It is not just agreement for one selected contraction. This supplies a strict improvement over *projecting the inherited pieces separately and using their full-rank interfaces*. It is not a gain over a comparator that already recomposes the quotient crossing directly. The example explains why target projection should be followed by decomposition selection, not by assuming that an inherited low-rank advantage automatically survives.

C.3 Polynomially evaluable integer realization of Appendix H's rank hint

The paper's Appendix H points out the rank saving from the single-bit identity. To use a rank statement in Definitions 1 and 7, integer factors and their evaluation costs are needed. The following explicit construction supplies them without rational Gaussian elimination.

For integer $t \geq 1$ and strings of t bits define B_i by

$$\prod_{j=1}^t (I_j - T A_j) = \sum_{i=0}^t B_i T^i, \quad F(t, k) = \sum_{z=0}^{\min(t, k)} \binom{t}{z} 3^z,$$

with $F(t, k) = 0$ for $k < 0$. Use the four integer Bell vectors

$$v_0 = (1, 0, 0, 1), \quad v_1 = (1, 0, 0, -1), \quad v_2 = (0, 1, 1, 0), \quad v_3 = (0, 1, -1, 0).$$

They are orthogonal and all have squared norm two. A Bell pattern d of length t gives $v_d = \bigotimes_j v_{d_j}$, an integer vector with entries $0, \pm 1$ and squared norm 2^t .

Theorem C.2 (Explicit integer crossing pieces). *The integer pieces $A'_i = 2^t B_i$, $0 \leq i \leq t$, sum to $2^t S_t$, where $S_t = [x = v][y = u]$ on t -bit strings. Their exact row and column ranks are*

$$\text{rank}_{\text{row}} B_i = F(t, t - i), \quad \text{rank}_{\text{col}} B_i = F(t, i).$$

In each direction A'_i has an explicit integer factorization through the displayed number of terms, with factors of magnitude at most 2^{2t} . An individual factor, the piece value and its index are evaluable in polynomial bit work in t . Thus these are pieces in the paper's Definition 7 with

$$\rho = \frac{1}{4} \log_2 12 \simeq 0.8962406252, \quad c_i^{\text{row}} + c_i^{\text{col}} \leq t \log_2 12 + 2.$$

Proof. At $T = 1$ each factor is the single-bit crossing, proving the sum identity. At the row split, I is the identity matrix and $A = v_3 v_3^\top$ has eigenvalue two on v_3 , zero on the other three Bell vectors. For a pattern with z occurrences of 3, the eigenvalue of B_i is $(-2)^i \binom{z}{i}$. It is nonzero exactly when $z \geq i$. The number of such patterns is $F(t, t - i)$.

At the column split, $I = v_0 v_0^\top$ and $A = J \otimes J$, where $J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. On (v_0, v_1, v_2, v_3) the polynomial $I - TA$ has eigenvalues $(2 - T, T, T, -T)$. Let z count the nonspecial entries $d_j \neq 0$, and let n_3 count entries equal to 3. The coefficient eigenvalue is

$$c_d = (-1)^{n_3 + i - z} 2^{t-i} \binom{t-z}{i-z} \quad (z \leq i),$$

and zero otherwise. This gives rank $F(t, i)$.

Since $\sum_d v_d v_d^\top = 2^t \text{Id}$, at either split

$$2^t B_i = \sum_{d: c_d \neq 0} (c_d v_d) v_d^\top.$$

Both factors are integer; zero-padding gives an index of $\lceil \log_2 F \rceil$ bits. The displayed coefficients are bounded by 2^{2t} , and a Bell vector entry is a product of t entries in $\{0, \pm 1\}$. Patterns with at most k nonspecial entries are indexed lexicographically by suffix counts $F(s, k)$, so unranking one pattern uses polynomially many integer operations on $O(t)$ -bit counts. Binomial coefficients, eigenvalues and the polynomial coefficient defining B_i are likewise polynomially evaluable. No list of all 4^t patterns is constructed by a factor query.

Finally $F(t, k) \leq 2^t 3^k$, so the product of the two ranks is at most 12^t . Taking two ceilings proves the directional bit bound. The piece count is $t + 1$, the common scalar is 2^t , and coefficient sizes and evaluators meet Definition 7 after increasing its absolute constant c . $\square$

The rank calculation is already motivated in Appendix H; the result above is its executable integer-factor completion, not a new rank discovery. Its ρ is better than the main text’s simple 11/12 estimate but worse than Appendix B’s $1/\sqrt{2}$. Consequently it supplies a transparent alternative component, with polynomial-time entry evaluators in place of that construction’s stated $2^{O(t)}$ bound, *not* a better general ETH exponent. As always, actual evaluation costs and term enumeration must be compared jointly; a cheaper evaluator alone does not settle the tradeoff. The code implements these pieces, not the entire recursive ETH algorithm.

C.4 Rational adapter, network costs and fair comparisons

For supplied rational quotient tables choose the positive least common denominator q_v at each vertex, set $W_v^{\text{int}} = q_v \bar{W}_v$, and let $Q = \prod_v q_v$. Then

$$Z = 2^D \bar{Z}_{\text{int}} / Q.$$

The same Q is used once; no scalar or discarded-bit factor is restored twice. Signed tables remain algebraic weights rather than probabilities. Their exact likelihood interpretation requires a separately admitted nonnegative normalized source and readout model.

The integer network fits Anthropic’s Definition 2 using equality edge factors, with

$$U = 2 \sum_e r_e, \quad P = \max\{1, \max_v \sum_{e \ni v} r_e\}, \quad B = \max\{1, \max_{v,y} \text{bitlength } |W_v^{\text{int}}(y)|\}.$$

Thus $|W_v^{\text{int}}| < 2^B$. Exact table/evaluator cost is τ . For an admitted fixed K satisfying $|V| \leq g = 3K^8$, pad with isolated unit vertices to g . Conditional on an achievable pair (K, e) , Anthropic’s Lemma 1 gives the following bound in its stated word-RAM operation model:

$$C_1 2^{e(U+gP/K)+2gP/K} (P+B)^{C_1} \tau,$$

plus denominator clearing, output recovery and padding-description work. If $|V| > 3K^8$, this padding argument does not apply at that K ; one must choose a suitable larger achievable pair and charge its changed constants, or use a different reduction. No uniform bound is obtained by treating those new constants as fixed while the network size grows. This is an explicit direct integer-network bridge. It avoids modular reconstruction but does not prove a uniformly better complexity bound than the rational OpenAI adapter. When K grows, Appendix A(b) has the factor $[K(\lambda+B)]^{C_0 K^2}$. Here λ is the common clique-edge string width and B the table-entry magnitude parameter in that external theorem, not an individual original edge width λ_e . Precision dependence cannot then be described as a fixed polynomial independent of K ; its operation model must also be distinguished from the bit-complexity statement of the OpenAI adapter.

The implemented ordinary enumeration backend evaluates the quotient network exactly and restores its scalar. For a complete four-vertex network with three-bit edge strings, one retained parity per

edge and positive supplied vertex tables, both independent calculations give $Z = 45\,621\,248$. Full enumeration uses 262 144 assignments and 1 048 576 vertex queries; quotient enumeration uses 64 assignments and 256 vertex queries, plus polynomial representation checks and a 12-bit recovery shift. The full comparator additionally builds 48 local edge-readout entries. This is an actual executed finite comparison. At six vertices, quotient enumeration uses 32 768 assignments and 196 608 vertex queries, returning 215 011 853 527 416 832. The 2^{45} full assignment count is theoretical and was not executed.

A fair specialized comparator can already apply exact bond quotienting or exploit another low-width structure. It has the same compressed task; no advantage over that comparator is claimed. General contraction-tree costs and rank decompositions are also different computation models [1]. The restricted reciprocal improvement here is relative to the explicitly inherited component pipeline, not to all algorithms or to the paper’s worst-case theorem.

C.5 Coverage, verification and retained limits

The implementation checks all 4,096 octonion identities, all 16 quotient basis contexts and 40 additional signed contexts. It verifies the two projected rank matrices, explicit integer Bell factors in both directions, exact ranks through three retained bits, 1,306 pattern-unranking cases, 2,145 right-inverse queries, 30 signed rational networks and independent full small-network contractions. Tests and exact parameters are supplied.

Appendix D’s sparse checks were examined but not modified into a claimed faster algorithm. A target that annihilates some unwanted assignments can restrict a coverage obligation only after that vanishing is proved and its representation cost charged. A target with full support need not remove any obligation. Signed cancellation of the final answer does not justify dropping assignments from the paper’s nonnegative stopping certificate. No general reduction in check count or load bound is proved.

Likewise, the graded operators in Appendix B supply an algebraic factorization; they are not a derived physical interaction, detector, Born selector or consciousness mechanism. This component extension preserves the source–readout distinction between information required for a target and a uniquely identified physical source. It changes no delayed-trine memory or complete-history resource bound.

References

- [1] Per Austrin, Petteri Kaski, and Kaie Kubjas. Tensor network complexity of multilinear maps. *Theory of Computing*, 18(16):1–54, 2022. doi: 10.4086/toc.2022.v018a016. URL <https://theoryofcomputing.org/articles/v018a016/>.
- [2] Karthekeyan Chandrasekaran, Richard Karp, Erick Moreno-Centeno, and Santosh Vempala. Algorithms for implicit hitting set problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 614–629, 2011. doi: 10.1137/1.9781611973082.48. URL <https://karthik.ise.illinois.edu/pubs/implicit-hitting-set.pdf>.
- [3] Claude. A $2^{o(n)}$ time algorithm for 3-sat, 2026. URL <https://github.com/anthropics/formal-math/blob/16a5ce41304fcd4aea4e0be7c132e1c361dc26da5/eth/paper.pdf>. Supplied manuscript dated 26 October 2026; Anthropic affiliation.
- [4] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. doi: 10.1109/18.910572. URL <https://www.isiweb.ee.ethz.ch/papers/arch/aloe-2001-1.pdf>.

- [5] Erick Moreno-Centeno and Richard M. Karp. The implicit hitting set approach to solve combinatorial optimization problems with an application to multigenome alignment. *Operations Research*, 61(2):453–468, 2013. doi: 10.1287/opre.1120.1139.
- [6] Ryan O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi: 10.1017/CBO9781139814782. URL <https://arxiv.org/abs/2105.10386>.
- [7] OpenAI. The exponential time hypothesis is false, 2026. URL <https://github.com/openai/math/blob/main/preprints/The-Exponential-Time-Hypothesis-is-False-October-25-2026/paper.pdf>. Dated 17 October 2026, revised 25 October 2026; supplied manuscript.
- [8] OpenAI. Scope of the lean formalization: The exponential time hypothesis is false, 2026. URL <https://raw.githubusercontent.com/openai/math/main/lean/docs/378.md>.
- [9] OpenAI. Working with academic communities to build scientific understanding, October 2026. URL <https://openai.com/index/building-scientific-understanding-with-academic-communities/>.
- [10] Jeremy Rodgers. Target-relative necessity of completion: When readout loss obstructs, and what a sufficient extension must retain, 2026.
- [11] Jeremy Rodgers. Canonical minimal source completion: The coarsest readout extension on which a nominated family of source relations becomes well defined, 2026.
- [12] Jeremy Rodgers. Observable quotients and exact projected dynamics: Closure, memory, minimal dynamical completion, and effective field operators, 2026.
- [13] Alexander D. Scott and Gregory B. Sorkin. Polynomial constraint satisfaction problems, graph bisection, and the Ising partition function. *ACM Transactions on Algorithms*, 5(4):45:1–45:27, 2009. doi: 10.1145/1597036.1597049. URL <https://people.maths.ox.ac.uk/scott/Papers/counting.pdf>.
- [14] Sukhwinder Singh and Guifre Vidal. Global symmetries in tensor network states: Symmetric tensors versus minimal bond dimension. *Physical Review B*, 88:115147, 2013. doi: 10.1103/PhysRevB.88.115147.
- [15] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifre Vidal. Tensor network decompositions in the presence of a global symmetry. *Physical Review A*, 82:050301, 2010. doi: 10.1103/PhysRevA.82.050301.
- [16] Armando Solar-Lezama. *Program Synthesis by Sketching*. PhD thesis, University of California, Berkeley, 2008. URL <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-177.html>.