ADVERSARIAL PROOF ENGINEERING FOR THE P VERSUS NP PROBLEM: AUDITED LEMMAS, COUNTEREXAMPLES, AND AN EXPLICIT RESEARCH FRONTIER

FLOYD KORZAN

Abstract. We report an adversarial, reproducible proof-engineering investigation of the P versus NP problem. The investigation does *not* resolve the problem. Its contribution is a formally separated collection of proved conditional bridges, exact obstructions, and open dependencies. We prove that a logarithmic structural backdoor to 2-SAT, if universal, would yield a polynomial-time algorithm for 3-SAT, and that constant-fraction shrinkage in both branches of unit-propagating DPLL would do the same. We then refute both universal premises by explicit infinite families, with small machine-checked certificates. A counting construction gives a decidable language outside $\mathsf{P/poly}$, but its membership in NP remains unproved; a clocked diagonalization leaves P but fails the uniform-exponent requirement for NP verification; and a naive universal Cook–Reckhow proof system is unsound. Eleven routes are compared against relativization, Natural Proofs, algebrization, locality, uniformity, and quantifier risks. The surviving frontier is an exact Gap-MCSP hardness-magnification antecedent, preceded by a fully finite diagnostic that is expressly not extrapolated. All claims are classified, their dependencies are exposed, and no open edge is used to infer either $\mathsf{P} = \mathsf{NP}$ or $\mathsf{P} \neq \mathsf{NP}$.

1. Introduction

The question whether every language whose positive instances admit polynomially checkable witnesses also admits a deterministic polynomial-time decision algorithm is the central open problem of complexity theory. In the standard notation, it asks whether

$$\mathsf{P} \stackrel{?}{=} \mathsf{NP}.$$

As of the date of this manuscript, the Clay Mathematics Institute continues to list the problem as unsolved [4]. This status has a concrete logical consequence for any new investigation: an interesting restricted lower bound,

Date: August 2, 2026.

2020 *Mathematics Subject Classification.* Primary 68Q15; Secondary 68Q17, 03F20, 68Q25.

Key words and phrases. P versus NP, circuit complexity, SAT algorithms, MCSP, proof complexity, barriers, counterexamples.

a finite computation, or a conditional theorem cannot be presented as a solution.

The aim here is therefore narrower and auditable. We run a portfolio of plausible proof routes, force each route to state its exact final implication, identify its first missing lemma, and attack that lemma before building more machinery. When a lemma fails, the counterexample and the valid conditional residue are retained. When it survives finite tests, it remains a conjecture until an asymptotic proof is supplied. This produces negative information, but it is useful negative information: it prevents the same invalid quantifier swaps, model transfers, and finite-to-asymptotic leaps from being hidden in more elaborate notation.

The principal findings are as follows.

(1) Two exact structural hypotheses for 3-SAT imply polynomial-time algorithms. Both hypotheses fail on elementary infinite families.
(2) Circuit counting defines a language $H \notin \mathsf{P/poly}$. The missing step is not hardness but an NP verifier for the selected truth-table bits.
(3) Direct clocked diagonalization gives a decidable language outside P, but the obvious tableau verifier has an input-dependent exponent.
(4) Enumerating polynomial-time transducers does not construct a universal propositional proof system, because the enumeration also contains unsound transducers.
(5) After a predeclared five-attempt reassessment, Gap-MCSP hardness magnification is promoted as the primary frontier. Its known implication is useful only after its precise promise, circuit model, and parameters are fixed.

Every item above is substantially weaker than a resolution. Section 2 fixes the computational and evidentiary models. Section 3 describes the adversarial method and barrier audit. Sections 4–7 give the proofs and exact failure points. Section 8 states the promoted frontier and its finite precursor. The final sections reconstruct the dependency cut and the reproducibility protocol.

2. Models and integrity boundary

Languages are subsets of $\{0,1\}^*$. Inputs are finite binary strings, and their length is the number of encoded bits. The reference uniform model is a deterministic multitape Turing machine. Integers are encoded in binary, and arithmetic complexity is measured in bit operations; no unit-cost arithmetic on unbounded integers is used.

Definition 2.1. A language L is in P if there exist a deterministic Turing machine M and constants $c, k \in \mathbb{N}$ such that M decides L and halts on every input x within $c(|x|+1)^k$ steps.

Definition 2.2. A language L is in NP if there exist a polynomial p and a deterministic polynomial-time predicate V such that, for every x,

$$x \in L \iff \exists y \in \{0,1\}^* \ (|y| \le p(|x|) \ \wedge \ V(x,y) = 1).$$

Crucially, one verifier and one fixed polynomial must work for all inputs.

For nonuniform arguments, $\mathsf{P/poly}$ denotes languages having polynomial-size Boolean circuit families over a fixed finite fan-in basis. Changing the basis or adding constants changes size by at most a constant factor and does not affect the conclusions below.

2.1. **Status vocabulary.** Each substantive assertion receives one of the following labels.

Label	Meaning
KNOWN THEOREM	Imported result with a checked statement and source.
PROVED HERE	A complete proof appears in this manuscript.
CONDITIONAL	The implication is proved, but at least one displayed hypothesis remains open.
OPEN OBLIGATION	A precise missing edge; no proof is claimed.
DISPROVED	An explicit counterexample or logical contradiction is given.
FINITE ONLY	Exhaustive computation establishes only the stated bounded instance.

The main integrity rule is simple: a conclusion may use only proved-here and verified known-theorem edges. Conditional, open, and finite-only statements cannot discharge an asymptotic dependency.

3. Methodology and route tournament

3.1. **Adversarial loop.** For each route we use the same six-step loop.

(1) State the endpoint exactly: $\mathsf{P} = \mathsf{NP}$ or $\mathsf{P} \neq \mathsf{NP}$.
(2) Draw a dependency chain from a proposed lemma to that endpoint.
(3) Mark the first edge not already proved or sourced.
(4) Test its smallest nontrivial finite form and search for structured counterexamples.
(5) If the edge survives, separate the finite observation from the required asymptotic statement; if it fails, prove the counterexample family.
(6) Audit quantifiers, encodings, uniformity, model transfers, and known barriers before changing the claim status.

This organization deliberately separates four layers that are often conflated: a proof idea, exploratory scratch work, a proof with all quantifiers visible, and the moral that determines whether the route should continue.

3.2. **Portfolio scoring.** Eleven routes were scored from 0 to 10 on logical sufficiency (S), proximity (D), barrier survivability (B), falsifiability (F), experimental value (X), formalizability (L), novelty potential (N), and quantifier safety (Q). Higher is favorable in every coordinate; totals are triage scores, not probabilities.

TABLE 1. Initial route tournament.

Route	S	D	B	F	X	L	N	Q	Total
R1 Structural 3-SAT decomposition	10	7	4	10	10	9	3	8	61
R2 Counting plus NP-explicitness	10	7	2	6	6	7	4	3	45
R3 Resource-bounded diagonalization	10	6	1	7	5	8	3	2	42
R4 Non-natural circuit property	10	4	3	4	4	6	8	4	43
R5 Circuit-SAT algorithm to NP lower bound	8	3	5	6	8	6	7	3	46
R6 Gap-MCSP hardness magnification	9	6	4	6	7	7	7	4	50
R7 Optimal proof system plus lower bound	9	4	5	7	5	9	7	3	49
R8 Algebraic/GCT obstruction and transfer	6	2	5	5	8	7	9	2	44
R9 Communication lifting to circuit size	8	3	5	7	8	7	8	4	50
R10 Ordered descriptive complexity	10	4	6	5	6	8	8	3	50
R11 Determinantal SAT representation	10	4	2	10	10	9	8	8	61

R1 was selected first because its proposed lemmas had exact algorithmic consequences and could be killed cheaply if false. R6 was the backup because hardness magnification offers a short implication from a modest-looking lower bound to a major circuit separation, provided the parameters match exactly. R11 was retained as a methodologically different wildcard. The reassessment rule was fixed in advance: after five substantive attempts, promote R6 if R1 has accumulated two structurally different infinite obstructions. That condition was met in Section 4.

3.3. **Barrier audit.** The portfolio is not an assertion that familiar barriers are absolute impossibility theorems. It is a requirement to identify which proof features they rule out.

TABLE 2. Barrier exposure of the active ideas.

Barrier	Trigger	Audit consequence
Relativization [2]	Simulation, diagonalization, or reduction works unchanged relative to every oracle.	Any completion retaining R3's relativizing core needs a nonrelativizing step; its current open edge is NP membership.
Natural Proofs [9]	Under the stated pseudorandomness hypothesis, a property has a polynomial-truth-table-time subproperty, inverse- polynomial density, and usefulness against the target circuit class.	R2 and R4 must calculate constructivity, density, usefulness, and the conditional hardness premise rather than merely calling a function "explicit."
Algebrization [1]	Arithmetization plus oracle access preserves the argument.	Algebraic encodings require a specifically nonalgebrizing transfer, not just polynomial identities.
Locality in magnification [3]	A lower-bound method extends to circuits with specified oracle quantity q, fan-in ℓ, and adaptivity a.	R6 must test a proposed method against the particular oracle upper bound, including q, ℓ, a, placement, and total size.
Restricted-model gap	A theorem controls depth, monotonicity, formulas, or one proof system.	R5, R7, R8, and R9 must prove the missing model-transfer edge; it is never implicit.

Cook–Levin supplies the standard algorithmic endpoint [5]. Williams' algorithm-to-lower-bound theorem [10] demonstrates that circuit algorithms can yield major unrestricted lower bounds, but its NEXP conclusion is not silently replaced by an NP conclusion. These distinctions govern the proofs that follow.

4. Structural 3-SAT: valid bridges and fatal premises

Clauses are normalized sets of at most three noncomplementary literals. Let N be the bit length of an encoded formula, m its number of clauses, and n its number of active variables.

4.1. Strong backdoors to 2-SAT.

Definition 4.1. Let F be a 3-CNF and let B be a set of its variables.

(1) B is a *semantic strong* 2*-CNF backdoor* if every assignment to B simplifies F either to a contradiction or to a CNF of width at most two.
(2) B is a *structural strong* 2*-CNF backdoor* if B intersects the variable support of every width-three clause.

Here a contradictory syntactic residual means one containing an empty clause. Thus an assignment producing an empty clause satisfies the semantic condition even if an untouched width-three clause is also present.

The distinction matters for unsatisfiable formulas: a contradictory residual may collapse before an untouched width-three clause is inspected.

Lemma 4.2 (Structural characterization)**.** *Every structural strong* 2*-CNF backdoor is semantic. If F is satisfiable, then every semantic strong* 2*-CNF backdoor is structural.*

Proof. Suppose first that B intersects every width-three clause, and fix an assignment α to B. A width-three clause containing a true assigned literal disappears. Otherwise at least one false assigned literal is deleted, so the residual clause has width at most two. Clauses initially of width at most two do not grow. Hence every noncontradictory residual is a 2-CNF.

Conversely, suppose that F is satisfiable and B misses a width-three clause C. Restrict a satisfying assignment of F to B, obtaining α. Then $F \restriction \alpha$ is satisfiable and hence not contradictory. Because no variable of C was assigned, C remains a width-three clause. Thus the residual is not a 2-CNF, and B is not semantic. □

Theorem 4.3 (Conditional backdoor algorithm)**.** *Assume that a constant c exists such that every normalized* 3*-CNF F of encoding length N has a structural strong* 2*-CNF backdoor of size at most $c \log_2(N+1)$. Then* 3*-SAT is in* P*, and consequently* P = NP*.*

Proof. Set $k = \lceil c \log_2(N+1) \rceil$. Finding a structural backdoor of size at most k is a rank-three hitting-set problem. Choose an unhit width-three clause and branch on adding each of its at most three variables. The bounded search tree has at most $3^k = (N+1)^{c \log_2 3 + O(1/\log N)}$ leaves and polynomial work per node. The hypothesis guarantees that the search succeeds.

For the resulting set B, enumerate its at most $2^k \le 2(N+1)^c$ assignments. Lemma 4.2 reduces every noncontradictory residual to 2-SAT, which is decidable in polynomial time. The original formula is satisfiable exactly when some residual is satisfiable. The full running time is polynomial in N. The Cook–Levin theorem and the standard reduction to 3-SAT then give P = NP. □

The implication is valid; its premise is not.

Proposition 4.4 (Disjoint-clause obstruction)**.** *For $t \geq 1$, let*

$$D_t = \bigwedge_{j=0}^{t-1} (x_{3j+1} \vee x_{3j+2} \vee x_{3j+3}).$$

The minimum structural strong 2-CNF backdoor of D_t has size exactly t. The same is true semantically. Consequently, no universal $O(\log N)$ bound of the form assumed in Theorem 4.3 can hold.

Proof. The t width-three clause supports are pairwise disjoint. A structural backdoor must choose at least one variable from each support, so its size is at least t; one choice from every support gives size t. The formula is satisfiable, so Lemma 4.2 gives equality with the semantic minimum. Under a standard binary variable encoding, the formula length is $N = O(t \log(t+1))$. Since $t/\log N$ is unbounded, the claimed universal logarithmic bound fails. □

For the concrete constant-one claim $|B| \leq \lceil \log_2(m+1) \rceil$, the smallest member D_3 already has minimum backdoor size three while the proposed bound is two.

4.2. **Two-branch unit-propagation shrinkage.** Exhaustive unit propagation repeatedly satisfies unit clauses and deletes the negations of their literals. A residual is terminal if it is satisfied, contains an empty clause, or has at most a fixed constant number of variables and is solved exhaustively. A residual formula with n active variables has the δ-shrinkage property if some active variable v has the following behavior for both $b \in \{0,1\}$: after setting $v = b$ and applying unit propagation, the branch contradicts or has at most $(1-\delta)n$ active variables.

Theorem 4.5 (Conditional shrinkage algorithm)**.** *Fix $\delta \in (0,1)$. If a polynomial-time procedure finds a δ-shrinkage variable in every nonterminal residual 3-CNF, then 3-SAT is in* P*, and consequently* P = NP*.*

Proof. Branch on the returned variable and apply exhaustive unit propagation. If $T(n)$ is the maximum number of recursive nodes below a residual with n active variables, then

$$T(n) \leq 2T(\lfloor (1-\delta)n \rfloor) + 1.$$

Every root-to-leaf path has length at most

$$h = \left\lceil \log_{1/(1-\delta)} n \right\rceil + O(1).$$

The binary tree therefore has $O(2^h) = O(n^{\log_{1/(1-\delta)} 2})$ nodes. Each node performs polynomial work in the original input length. Exhaustive branching is sound and complete, so the resulting deterministic algorithm is polynomial. The final implication again follows from Cook–Levin. □

Proposition 4.6 (Connected and sign-balanced obstructions)**.** *No fixed $\delta > 0$ satisfies the universal premise of Theorem 4.5. The failure occurs both*

for connected positive clause chains and for connected CNFs in which every variable appears with both signs.

Proof. First consider

$$C_t = \bigwedge_{j=0}^{t-1} (x_{2j+1} \vee x_{2j+2} \vee x_{2j+3}),$$

where consecutive clauses share an endpoint variable. Setting any chosen variable false changes incident 3-clauses into 2-clauses and creates no unit clause. All $n-1$ other variables remain active. For fixed $\delta > 0$ and $n > 1/\delta$, one has $n-1 > (1-\delta)n$, so no variable shrinks both branches by the required fraction.

For a sign-balanced family, impose the parity constraints $x_{2j+1} \oplus x_{2j+2} \oplus x_{2j+3} = 0$ in a chain, encoding each constraint as

$$(x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z) \\ \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z).$$

Every variable occurs with both signs. Assigning any one variable turns each incident parity constraint into a pair of width-two clauses, never a unit clause. In both truth branches all $n-1$ other variables remain active. The same inequality defeats every fixed $\delta > 0$. □

At $\delta = 1/4$, five active variables are necessary because assignment always removes the selected variable and $n - 1 \leq 3n/4$ for $n \leq 4$. The positive formula

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_4 \vee x_5)$$

is therefore a smallest witness. A two-constraint parity chain supplies a five-variable sign-balanced witness.

Remark 4.7 (Moral of R1). Theorems 4.3 and 4.5 are genuine polynomial-time bridges, but Propositions 4.4 and 4.6 disprove their premises. The easy nature of some counterexample formulas is irrelevant: the formulas refute universal structural statements, not the tractability of their own instances. Two different infinite obstructions trigger the predeclared pivot from R1 to R6.

5. Counting hardness and the explicitness obligation

Counting proves that most Boolean functions require large circuits. The tempting additional move is to select the lexicographically first hard truth table at every length. The selection is computable, but it does not inherit an efficient verifier merely because it is uniquely defined.

Fix $m \geq 4$, put $T = 2^m$, and define

$$s_m = \left\lfloor \frac{T}{20m} \right\rfloor.$$

Circuits use fan-in-two $\{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}\}$ gates; a NOT gate may ignore its second named input. Size is gate count. Order the T-bit truth tables lexicographically, and let h_m be the first table not computed by a circuit of size at most s_m. For $m < 4$, set h_m to the all-zero table. Define

$$H = \{x \in \{0,1\}^* : h_{|x|}(x) = 1\},$$

where x denotes its lexicographic truth-table position.

Lemma 5.1 (Counting lemma)**.** *For every $m \geq 4$, the table h_m is well-defined and has circuit size greater than s_m.*

Proof. Write $s = s_m$. We intentionally overcount circuits. Give at most $s+1$ choices for the number of gates. In a topological description, give each gate three type choices and each of its two named inputs at most $m+s$ source-wire choices. Allowing future sources only enlarges the count. Give the output wire at most $m + s$ choices. The number C_m of described functions is at most

$$C_m \leq (s+1)(m+s)\big(3(m+s)^2\big)^s.$$

Because $m \geq 4$, we have $m + s \leq T$ and $s + 1 \leq T$. Hence

$$\begin{aligned}
\log_2 C_m &\leq \log_2(s+1) + \log_2(m+s) + s\big(\log_2 3 + 2\log_2(m+s)\big) \\
&< 2m + s(2m+2) \\
&\leq 2m + \frac{T}{20m}(2m+2) \\
&\leq \frac{T}{2} + \frac{T}{8} < T.
\end{aligned}$$

There are exactly 2^T Boolean functions on m variables, while fewer than 2^T are represented by the counted descriptions. A missing table exists, and the lexicographically first such table has size greater than s_m. □

Corollary 5.2. *The language H is decidable and $H \notin \mathsf{P/poly}$.*

Proof. On an input of length m, a finite procedure enumerates every circuit of size at most s_m, evaluates its complete truth table, and chooses the first absent table. Thus H is decidable, although this procedure has no useful polynomial-time bound.

Suppose $H \in \mathsf{P/poly}$. After a constant-factor basis conversion, some fixed polynomial $q(m)$ bounds circuits computing every sufficiently large length slice of H. Since $2^m/(20m)$ eventually exceeds $q(m)$ by an unbounded factor, $q(m) \leq s_m$ for all sufficiently large m. Such a circuit would compute h_m with at most s_m gates, contradicting Lemma 5.1. □

Open obligation 5.3 (NP-explicitness gap)**.** Prove $H \in \mathsf{NP}$, or construct another language in NP whose circuit lower bound follows with comparable rigor.

If Obligation 5.3 were discharged, Corollary 5.2 would give $\mathsf{NP} \not\subseteq \mathsf{P/poly}$, hence $\mathsf{P} \neq \mathsf{NP}$. The obvious verifier does not discharge it. Certifying that a proposed table is the lexicographically first missing table requires both

(1) a universal assertion that no size-s_m circuit computes the table, and
(2) evidence that every earlier table is computed by such a circuit.

The second list may have doubly exponentially many members relative to the input length m, and the first assertion has the wrong quantifier shape for a plain NP witness. No succinct certificate theorem is proved here.

Remark 5.4 (Moral of R2). The hard function is not the missing object. Efficient access to its selected bits is. "Lexicographically first" supplies canonicality, not NP-explicitness.

6. Resource-bounded diagonalization and uniform exponents

Let $(M_i, c_i, k_i)_{i\geq 1}$ effectively enumerate deterministic machines with positive integer clocks. Let D_i simulate M_i for at most $c_i(n+1)^{k_i}$ steps on length-n inputs and reject if the clock expires. Every polynomial-time language is decided by at least one D_i. Choose a canonical string $w_i = 1^i 0$; its range is decidable and its index is recoverable in linear time. Put

$$w_i \in L_{\mathrm{diag}} \quad \Longleftrightarrow \quad D_i(w_i) = 0,$$

rejecting noncanonical strings.

Proposition 6.1. *The language L_{diag} is decidable and does not belong to* P.

Proof. The relevant clocked simulation is finite, so the language is decidable. If some D_j decided L_{diag}, then on the canonical input w_j its acceptance bit would satisfy

$$D_j(w_j) = 1 \quad \Longleftrightarrow \quad D_j(w_j) = 0,$$

a contradiction. Since every polynomial-time decider is represented by a clocked machine, $L_{\mathrm{diag}} \notin \mathsf{P}$. □

The almost-correct next sentence is that a certificate can contain the full computation tableau of $D_i(w_i)$. The clock gives only the upper bound

$$\mathrm{time}_{D_i}(w_i) \leq c_i(|w_i|+1)^{k_i};$$

the exponent alone is not a runtime lower bound. Nevertheless, the literal full-history certificate scheme has no uniform polynomial bound. For every $d, r \geq 1$, the enumeration contains a syntactically distinct clocked machine $R_{d,r}$ that performs at least $(n+1)^d$ dummy steps on every length-n input and then rejects, with a clock large enough to finish. For fixed d, the distinct indices obtained as r varies give unary canonical strings of unbounded length. Given a proposed bound Cn^K, choose $d > K$ and then choose r so that the canonical string is sufficiently long. Its rejecting history has more than $C|w_i|^K$ configurations, while w_i is a positive instance of L_{diag}. Thus literal full histories are not uniformly polynomially bounded.

This refutes only the full-history certificate format; it does not exclude a different succinct NP certificate. The underlying quantifier hazard is

$$\forall i\ \exists k_i \quad \not\Rightarrow \quad \exists K\ \forall i.$$

Padding the same simulated input does not remove its input-dependent exponent; simulating a shorter input removes the self-reference needed for the diagonal contradiction. Thus $L_{\mathrm{diag}} \in \mathsf{NP}$ remains an open obligation, not a conclusion. Moreover, the simulation-and-flip step relativizes, exposing the route to the Baker–Gill–Solovay barrier [2].

Remark 6.2 (Moral of R3). Diagonalization can range over all polynomial exponents, or a single verifier can have one polynomial exponent. The full-tableau construction does not provide both. Any repair needs a uniform succinct verification mechanism and a nonrelativizing ingredient.

7. Proof complexity and unsound universalization

In the functional Cook–Reckhow formulation, a propositional proof system is a polynomial-time computable surjection $P : \{0,1\}^* \to \mathsf{TAUT}$ [6]. It is *polynomially bounded* if one polynomial bounds the shortest P-proof of every tautology in terms of the formula length.

Proposition 7.1 (Cook–Reckhow bridge)**.** *A polynomially bounded propositional proof system exists if and only if* $\mathsf{NP} = \mathsf{coNP}$.

Proof. Suppose first that a polynomially bounded system P exists. On input φ, an NP verifier guesses a polynomial-length string π, computes $P(\pi)$, and accepts when the result equals φ. Surjectivity and the range condition give completeness and soundness, so $\mathsf{TAUT} \in \mathsf{NP}$. Since TAUT is coNP-complete, $\mathsf{coNP} \subseteq \mathsf{NP}$; taking complements gives the reverse inclusion.

Conversely, suppose $\mathsf{NP} = \mathsf{coNP}$. Then $\mathsf{TAUT} \in \mathsf{NP}$, so a polynomial-time verifier $V(\varphi, y)$ has polynomially bounded witnesses for exactly the tautologies. Define a map on encodings (φ, y) that outputs φ when V accepts and a fixed tautology otherwise. This is a polynomial-time surjection onto TAUT, and every tautology has a polynomial-length preimage. □

A system U *p-simulates* P if there is a polynomial-time translation f such that $U(f(\pi)) = P(\pi)$ for every proof π; in particular, the translated proof has polynomial length. A system p-simulating every Cook–Reckhow system is called p-optimal.

Lemma 7.2 (Conditional optimal-system bridge)**.** *If a p-optimal proof system U exists and its shortest proofs are not bounded by any polynomial, then* $\mathsf{P} \neq \mathsf{NP}$.

Proof. If $\mathsf{P} = \mathsf{NP}$, then $\mathsf{NP} = \mathsf{coNP}$, and Proposition 7.1 supplies a polynomially bounded system P. Polynomial simulation of P by U composes the bound on P-proofs with polynomial translation overhead, yielding a polynomial bound on U-proofs. This contradicts the hypothesis. □

The direct naive construction enumerates clocked polynomial-time transducers T_i and defines $U(i, \pi) = T_i(\pi)$. It has two defects. First, because the clock exponent is encoded in the unbounded index i, direct universal

simulation has no one polynomial running-time bound in the combined input length. Unary padding or an explicitly checked bounded computation history can repair this runtime issue while giving polynomial overhead for each fixed T_i. Soundness still fails: the enumeration contains the transducer that always outputs the one-variable formula x. Since x is not a tautology, the repaired universal range is not contained in TAUT; it is not a proof system. Filtering for the semantic condition "every output is a tautology" simply restates a universal soundness obligation. Requiring a proof of soundness in one fixed theory covers only systems whose soundness that theory proves.

Remark 7.3 (Moral of R7). Lower bounds for Resolution or another fixed system do not imply $\mathsf{NP} \neq \mathsf{coNP}$. The unproved edges are the existence of a sound p-optimal system and a superpolynomial lower bound for that system. Enumerating programs does not settle the first edge.

8. The promoted Gap-MCSP frontier

The Minimum Circuit Size Problem (MCSP) receives the full truth table of a Boolean function and asks whether the function has a small circuit [7]. Its promise version separates two size thresholds. For $a < b$, write Gap-MCSP$[a, b]$ for the promise problem whose yes-instances have circuit size at most a and whose no-instances have circuit size greater than b; behavior inside the gap is unrestricted.

The exact implication used by this route is parameter-sensitive.

Theorem 8.1 (Hardness magnification; Oliveira–Pich–Santhanam)**.** *There is a universal constant $c \geq 1$ with the following property. Let $N = 2^n$. If, for some $\varepsilon > 0$ and all sufficiently small $\beta > 0$,*

$$\text{Gap-MCSP}\left[\frac{2^{\beta n}}{cn}, 2^{\beta n}\right] \notin \text{Circuit}[N^{1+\varepsilon}],$$

where Circuit$[s]$ *denotes nonuniform, fan-in-two, unbounded-depth Boolean circuits with at most s gates, then* $\mathsf{NP} \not\subseteq$ Circuit[poly].

This is an imported conditional theorem, not a proved lower bound [8]. Here Circuit[poly] $= \mathsf{P/poly}$, so its conclusion implies $\mathsf{P} \neq \mathsf{NP}$ because $\mathsf{P} \subseteq \mathsf{P/poly}$. The antecedent is the open edge. In particular, N is the truth-table length, not the number n of variables, and both gap thresholds and the quantifiers on β must be preserved.

Open obligation 8.2 (R6 asymptotic obligation)**.** Prove the lower bound in Theorem 8.1, with the same promise, thresholds, input-length convention, and circuit model. For any proposed method, preserve those parameters and prove that its lower-bound argument does not extend to the specific oracle-augmented circuits known to compute the promise problem.

The locality analysis of Chen, Hirahara, Oliveira, Pich, Rajgopal, and Santhanam parameterizes an oracle circuit by the number $q(N)$ of oracle gates, their maximum fan-in $\ell(N)$, and oracle adaptivity $a(N)$ along

an input-to-output path [3]. For the associated Gap-MCSP regime and constructive $s(n) \le 2^n/\operatorname{poly}(n)$, they record an oracle upper bound with $q = N\operatorname{poly}(s)$, $\ell = \operatorname{poly}(s)$, $a = \operatorname{poly}(s)$, and total size $N\operatorname{poly}(s)$. Thus a proposed lower-bound method cannot extend unchanged to that specific augmented class. This is not a theorem that Obligation 8.2 is impossible; locality is a parameterized property of the proposed method.

8.1. **A finite distributional diagnostic.** Before proposing an asymptotic invariant, we isolate a completely finite precursor. The circuit basis has binary AND and OR and unary NOT; every gate, including every NOT, costs one. There are no constants and no free negated input literals. DAG sharing, unrestricted fan-out, and output directly from a primary input are allowed. This standalone diagnostic is not asserted to be a literal fixed-parameter specialization of Theorem 8.1; no model-transfer lemma is used. Let $\mathcal{L}_{3,2} \subseteq \{0,1\}^8$ be the truth tables of three-variable functions computed by at most two gates in this model. Let $\mathcal{D}_{8,3}$ be the Boolean distinguishers on eight input bits computed by at most three gates in the same model. We ask whether a distribution μ supported on $\mathcal{L}_{3,2}$ can $1/4$-fool every $D \in \mathcal{D}_{8,3}$ relative to the uniform distribution U_8 on $\{0,1\}^8$:

$$\left| \Pr_{z\sim\mu}[D(z)=1] - \Pr_{z\sim U_8}[D(z)=1] \right| \le \frac{1}{4}. \tag{P0}$$

Finite experiment 8.3 (Exact finite P0 certificate)**.** Problem (P0) is feasible. In the counted-NOT model above,

$$|\mathcal{L}_{3,2}| = 40, \qquad |\mathcal{D}_{8,3}| = 6{,}794.$$

There is a rational distribution of support size 19 and denominator 352 whose largest distinguishing gap over all 6,794 distinguishers is exactly $1/4$.

Here is the complete nonzero support. A hexadecimal byte denotes the truth table whose bit a is the output on the assignment encoded by a, with x_0 the least-significant assignment bit.
Enumeration completeness. Topologically order the gates of a k-gate circuit, and write the order as

$$g_0, \ldots, g_{k-1}.$$

At layer j, every source is one of the primary inputs or one of $g_0, \ldots, g_{j-1}$. Enumerating all operations and all such source indices therefore visits every circuit; ordering the inputs of commutative gates loses no function. Conversely every visited syntax is a legal circuit. A second enumerator retained only tuples of available source truth tables after each layer. Because all continuations depend only on those values, this quotient enumerator is sound. The raw and quotient enumerators produced identical function sets at every depth. Deduplication gave the two cardinalities in Experiment 8.3.

TABLE 3. Exact distribution certifying finite P0 feasibility.

Truth table	Weight	At-most-two-gate witness
03	5/352	$\neg(x_1 \vee x_2)$
05	31/352	$\neg(x_0 \vee x_2)$
11	10/352	$\neg(x_0 \vee x_1)$
22	8/352	$x_0 \wedge \neg x_1$
30	69/352	$x_2 \wedge \neg x_1$
3f	10/352	$\neg(x_1 \wedge x_2)$
50	12/352	$x_2 \wedge \neg x_0$
55	3/352	$\neg x_0$
5f	45/352	$\neg(x_0 \wedge x_2)$
77	27/352	$\neg(x_0 \wedge x_1)$
88	21/352	$x_0 \wedge x_1$
a0	19/352	$x_0 \wedge x_2$
a8	1/352	$x_0 \wedge (x_1 \vee x_2)$
bb	18/352	$x_0 \vee \neg x_1$
cf	11/352	$x_1 \vee \neg x_2$
ea	3/352	$x_0 \vee (x_1 \wedge x_2)$
ee	33/352	$x_0 \vee x_1$
f0	10/352	x_2
fc	16/352	$x_1 \vee x_2$

Exact certificate check. The nonnegative numerators in Table 3 sum to 352. For a distinguisher D, let

$$A_D = \sum_{t \in \mathrm{supp}(\mu)} w_t D(t), \qquad B_D = |\{u \in \{0,1\}^8 : D(u) = 1\}|,$$

where w_t is the displayed numerator. Then

$$|\mathbb{E}_\mu D - \mathbb{E}_{U_8} D| = \left|\frac{A_D}{352} - \frac{B_D}{256}\right| = \frac{|8A_D - 11B_D|}{2816}.$$

Exact integer evaluation over all 6,794 deduplicated distinguishers found maximum numerator 704, hence maximum gap $704/2816 = 1/4$; 37 distinguishers are tight. For example, $D(y_0, \ldots, y_7) = y_4 \wedge \neg y_7$ accepts with probabilities $176/352 = 1/2$ and $64/256 = 1/4$ under μ and U_8, respectively. Floating-point linear programming was used only to discover the weights; the displayed certificate and every final inequality use exact integers. The displayed point is not claimed to minimize the largest gap: P0 asks only for feasibility at threshold $1/4$.

Regardless of its outcome, (P0) is a linear feasibility question over a finite rational polytope. A feasible rational distribution or an exact dual infeasibility certificate can settle only these fixed parameters. It cannot establish the asymptotic lower bound in Obligation 8.2, and it cannot show that a candidate lower-bound proof fails to extend to the relevant oracle-augmented circuit class.

9. Dependency reconstruction and audits

9.1. **The exact dependency cut.** Table 4 reconstructs every route developed beyond literature triage. The last column is the first edge that prevents a conclusion.

Table 4. No developed dependency chain reaches a P-versus-NP endpoint.

Route	Valid residue	First blocking edge
R1	Logarithmic structural backdoors and fixed-fraction two-branch shrinkage each imply $\mathsf{P} = \mathsf{NP}$.	Both universal premises are disproved by infinite families.
R2	The decidable language H lies outside $\mathsf{P/poly}$.	$H \in \mathsf{NP}$ is not proved.
R3	L_{diag} is decidable and outside P.	One fixed-polynomial NP verifier is not supplied.
R6	Theorem 8.1 reaches $\mathsf{NP} \not\subseteq \mathrm{Circuit}[\mathrm{poly}]$.	Its exact modest lower-bound antecedent is open.
R7	A superpolynomial lower bound for a p-optimal system would imply $\mathsf{P} \neq \mathsf{NP}$.	Sound p-optimality and the lower bound are both open; the naive universal map is unsound.
R11	A polynomial-size, polynomial-bit determinantal representation of SAT would give $\mathsf{SAT} \in \mathsf{P}$.	No cancellation-free, globally consistent representation is proved.

It follows only that the displayed research program currently has no complete path to either endpoint. This is a statement about the proved dependency graph, not an independence theorem about P versus NP.

9.2. **Quantifier, model, and complexity audit.** The following checks were applied line by line.

(1) *Uniformity.* Every P or NP claim uses one machine and one fixed exponent. This rejects the tableau step in Section 6.
(2) *Input length.* Formula algorithms are polynomial in encoded formula length. MCSP distinguishes n function variables from truth-table length $N = 2^n$.
(3) *Nonuniform endpoints.* A lower bound outside $\mathsf{P/poly}$ implies $\mathsf{P} \neq \mathsf{NP}$ only after the hard language is placed in NP.

(4) *Model transfer.* Restricted circuit or proof-system lower bounds are not promoted without an explicit simulation or completeness theorem.
(5) *Arithmetic cost.* Proposed determinant computations must have polynomial matrix dimension and polynomial-bit entries; those conditions are still absent in R11.
(6) *Computation.* Exhaustive searches prove finite claims only. The infinite backdoor and shrinkage refutations are justified by Propositions 4.4 and 4.6, not by extrapolation.

9.3. **Reproducible finite checks.** The accompanying Python programs perform the following deterministic checks. The SAT and repository checkers use the standard library. The P0 discovery step uses NumPy, SciPy, and HiGHS; its reconstructed certificate is then verified with exact integer arithmetic.

(1) Enumerate normalized positive 3-CNFs until the constant-one backdoor claim fails; independently compare structural hitting sets with exhaustive restrictions on satisfiable instances. The run examined 44,833 formulas before returning D_3.
(2) Exhaust the boundary for $\delta = 1/4$ shrinkage and return the five-variable positive and parity-chain witnesses. Three JSON certificates are checked against their DIMACS instances.
(3) Evaluate the exact logarithmic circuit-counting inequality for $4 \leq m \leq 24$ while keeping the symbolic proof of Lemma 5.1 separate.
(4) Enumerate and solve (P0) exactly at its fixed parameters, recording the model convention, truth tables, and a checkable certificate.

All experiments are deterministic and use no network service. Their role is counterexample discovery, finite certification, and regression checking. No experimental trend is used as a complexity lower bound.

10. Conclusion

This investigation has not proved $\mathsf{P} = \mathsf{NP}$ and has not proved $\mathsf{P} \neq \mathsf{NP}$. It has produced four kinds of durable information: exact conditional algorithms, explicit infinite counterexamples, independently checkable finite certificates, and a dependency graph whose open edges are visible.

The most important methodological result is that three seductive words—"first," "polynomial," and "universal"—each hid a different quantifier failure. The first hard truth table is not thereby NP-verifiable; each clocked machine being polynomial does not give one polynomial for their diagonal language; and an enumeration containing every sound proof system is not itself sound. The two structural SAT failures are more concrete: a valid algorithmic recurrence is powerless when its universal measure premise is false.

The next legitimate action is Obligation 8.2: preserve the exact hardness-magnification model and seek a lower-bound method proved not to extend to the relevant oracle-augmented class. The finite diagnostic (P0) is useful only

for falsifying the smallest version of such an idea. Until every open edge in a dependency chain is closed, the correct status is CONTINUE, not SOLVED.

REFERENCES

[1] S. Aaronson and A. Wigderson, *Algebrization: A new barrier in complexity theory*, ACM Trans. Comput. Theory **1** (2009), no. 1, Art. 2, 54 pp.

[2] T. Baker, J. Gill, and R. Solovay, *Relativizations of the* $\mathsf{P} = ?\mathsf{NP}$ *question*, SIAM J. Comput. **4** (1975), no. 4, 431–442.

[3] L. Chen, S. Hirahara, I. C. Oliveira, J. Pich, N. Rajgopal, and R. Santhanam, *Beyond Natural Proofs: Hardness magnification and locality*, J. ACM **69** (2022), no. 4, Art. 25, 49 pp.

[4] Clay Mathematics Institute, *P vs NP*, Millennium Prize Problems, https://www.claymath.org/millennium/p-vs-np/, accessed August 2, 2026.

[5] S. A. Cook, *The complexity of theorem-proving procedures*, Proc. Third Annual ACM Symposium on Theory of Computing (1971), 151–158.

[6] S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*, J. Symbolic Logic **44** (1979), no. 1, 36–50.

[7] V. Kabanets and J.-Y. Cai, *Circuit minimization problem*, Proc. 32nd Annual ACM Symposium on Theory of Computing (2000), 73–79.

[8] I. C. Oliveira, J. Pich, and R. Santhanam, *Hardness magnification near state-of-the-art lower bounds*, 34th Computational Complexity Conference, LIPIcs **137** (2019), Art. 27, 29 pp.

[9] A. A. Razborov and S. Rudich, *Natural Proofs*, J. Comput. System Sci. **55** (1997), no. 1, 24–35.

[10] R. Williams, *Improving exhaustive search implies superpolynomial lower bounds*, SIAM J. Comput. **42** (2013), no. 3, 1218–1244.

COLUMBIA UNIVERSITY