Operational Separation of Discovery and Verification: An Entropy-Based Framework for NP-Hardness Under Physical Admissibility

---

General Reader Summary

The Problem: Some computational problems are easy to check but seemingly hard to solve. Given a completed Sudoku, you can verify it's correct in seconds. But finding the solution from scratch? That can take much longer. This asymmetry between "checking" and "finding" is the essence of the P versus NP problem—one of the deepest unsolved questions in mathematics and computer science.

The Core Insight: This paper argues that the difficulty isn't about our lack of cleverness—it's about physics. Finding a solution requires *reducing uncertainty*: starting from "it could be anything" and narrowing down to "it's this specific answer." But uncertainty doesn't just disappear. Like heat, it has to *go somewhere*. It must be dispersed through a series of small, local steps.

The Thermodynamic Analogy: Imagine trying to run an engine in a universe where everything is the same temperature. No matter how cleverly you design the engine, it won't work—there's no temperature difference to exploit. Hard computational problems are like this: they're "informationally isothermal." Every local measurement looks the same. There's no gradient to exploit, no shortcut to find.

What We Prove: We show that two major families of algorithms—logical deduction (like the methods used in industrial problem-solvers) and statistical detection (like machine learning approaches)—provably cannot solve these hard problems efficiently. These aren't just the methods we've tried; they represent the only known ways to reduce uncertainty locally.

The Remaining Question: Could there be a third way? Some method that reduces uncertainty without leaving any local trace? We argue this would be like a perpetual motion machine—not just undiscovered, but physically forbidden. Entropy reduction requires local dispersion. If there's no local channel, there's no reduction.

The Bottom Line: The gap between finding and checking isn't a puzzle waiting for a clever solution. It's a reflection of the same principles that govern heat flow, information erasure, and the arrow of time. Solving hard problems quickly would require not a new algorithm, but a new law of physics.

Table of Contents

Appendices

Abstract

We present a framework establishing that NP problems with high search information are operationally intractable under physically admissible computation. Our approach identifies families of NP instances where solutions are *probe-indistinguishable*—no polynomial-time algorithm achieves success probability exceeding the baseline $|S|/2^n$ by more than negligible. For such families, achieving constant success requires 2^(Ω(n)) trials, enforcing exponential discovery cost while verification remains polynomial.

Main Results:

1. **(Unconditional)** Resolution-based algorithms (DPLL, CDCL) require exponential time on explicit hard instance families. [Proof complexity: Haken, Ben-Sasson-Wigderson]
2. **(Unconditional)** Statistical query and low-degree polynomial algorithms require exponential time on planted random k-SAT in the hard regime. [Low-degree method: Feldman et al., Gamarnik-Sudan]
3. **(Proven — Footprint Collapse I)** Any reliable solver is SQ-detectable via its success rate. Perfect computational camouflage is impossible.
4. **(Conditional — Footprint Collapse II)** If every detectable solver leaves a *structured* footprint (resolution or low-degree), then for distributions where both paradigm-specific lower bounds apply, combining (1)-(3) yields an operational separation.

Two-Stage Footprint Collapse: We split the bridging assumption into:

- **Stage I (PROVEN)**: Reliable success → Detectable statistical bias
- **Stage II (CONJECTURE)**: Detectable bias → Structured footprint (resolution or low-degree)

Half of Footprint Collapse is now a theorem. Only the structure-extraction step remains conjectural.

Distribution Alignment Note: Parts III′ and III″ apply to different (though related) distributions. The operational separation holds on distributions where *both* lower bounds apply, or on any distribution where at least one lower bound applies combined with Footprint Collapse II.

The Gap to P ≠ NP: The framework establishes hardness for *specific distributions* over NP instances. A formal P ≠ NP proof (worst-case) additionally requires a distributional-to-worst-case bridge.

Scope: We reduce P vs NP to two questions: (a) Does detectability imply structure? (b) Does distributional hardness imply worst-case hardness? Affirmative answers, supported by extensive evidence, would yield P ≠ NP.

Part I: Conceptual Foundation

1.1 The Formal/Operational Distinction

Standard complexity theory treats computation abstractly: a Turing machine either halts in polynomial time or it doesn't, independent of physical realizability. This abstraction is powerful but obscures a crucial distinction:

Formal Existence: A mathematical object (algorithm, witness, compressed representation) is definable within a formal system.

Operational Realizability: The object can be instantiated as a distinguishable physical state within finite time using finite resources.

These are not equivalent. Consider:

π is finitely describable, but its full infinite expansion is not physically instantiable; physical records always correspond to finite prefixes. This illustrates how formal objects can exceed operational realizability even when every finite prefix is computable.

Similarly, a polynomial-time algorithm for an NP-complete problem may "exist" in some formal sense while being operationally unrealizable—not because the algorithm is logically impossible, but because instantiating its output requires resolving distinctions that exceed any finite resource bound.

1.2 Physical Admissibility of Computation

Definition 1.1 (Finite-Time Admissibility Principle): A computation is *admissible* only if its output can be instantiated as a distinguishable physical state in finite time using finite physical resources.

This principle has immediate consequences:

1. **Infinite-time solutions are operationally void**: A result existing only as a limit of an infinite process cannot be verified, recorded, or distinguished from alternatives at any finite time.
2. **Exponentially small success probabilities don't constitute computation**: An algorithm achieving success probability 2^(-cn) requires 2^(cn) trials for amplification to reliability, which is operationally equivalent to exponential time.
3. **Distinction requires physical resolution**: To distinguish a solution from a non-solution, a detectable physical difference must be producible within bounded resources.

1.3 Probe-Indistinguishability: The Core Condition (Instance-Relative)

The key innovation of this framework is identifying when high search information translates to operational hardness.

Definition 1.2 (Instance-Relative Probe-Indistinguishability Condition): A family of NP instances Φ_n with solution spaces S_n satisfies the *Probe-Indistinguishability Condition (PIC)* if:

For every polynomial-time algorithm A:

Pr_Φ ~ 𝓓_n[A(Φ) ∈ S_Φ] ≤ (|S_Φ|)/(2^n) + negl(n)

Equivalently: **no polynomial-time algorithm achieves non-negligible advantage over random guessing**.

Why Instance-Relative? Algorithms see instances Φ, not solutions. The original formulation (probes on solutions) was mis-specified because algorithms cannot sample from S directly. The instance-relative formulation captures the right notion: can an algorithm, given only Φ, do better than guessing?

Interpretation: Under PIC, solutions are computationally camouflaged. No polynomial-time method can "find" solutions with probability meaningfully exceeding the baseline $|S|/2^n$. For high search-information families where $|S|/2^n$ = 2^(-Ω(n)), this baseline is exponentially small.

1.4 What We Claim and What We Don't

Unconditional Results (Proven Theorems):

- Resolution-based algorithms require exponential time on hard satisfiable instances (Part III′)
- SQ/low-degree algorithms require exponential time on planted random k-SAT in hard regimes (Part III″)
- Reliable solvers are detectable in finite physics; perfect camouflage is impossible (Part III^4)

The Bridging Conjecture (Footprint Collapse):

- Every poly-time NP solver must leave either a resolution-style OR statistical footprint
- Well-motivated, falsifiable, supported by extensive evidence
- No known algorithm evades both paradigms

Conditional Result (Distributional Hardness):

- If Footprint Collapse holds, then NP search is hard on the specific distributions where (III′) and (III″) apply

Gap to Formal P ≠ NP (Worst-Case):

- The proven lower bounds apply to *specific distributions*, not worst-case
- Bridging distributional to worst-case hardness requires additional argument
- Standard approaches: worst-case-to-average-case reductions, or showing hard distribution captures worst-case

We Do NOT Claim:

- An unconditional proof of $P \neq NP$
- That Footprint Collapse is proven (it's a clearly labeled conjecture)
- That distributional hardness automatically implies worst-case hardness
- That our hard families are identical (Part III′ and III″ use related but different constructions)

The Value: The framework:

1. Proves exponential lower bounds in two dominant algorithmic paradigms
2. Identifies the precise conjecture that would extend to all algorithms
3. Proves perfect camouflage is impossible under finite physics
4. Reduces P vs NP to two well-defined structural questions

Part II: Technical Framework

2.1 Search Information and Solution Spaces

Definition 2.1 (Solution Space): For an NP instance Φ with solution space $S\_\Phi \subseteq 0,1^n$, define:

- **Solution count**: $|S\_\Phi|$
- **Solution entropy**: $H(S\_\Phi) = \log_2|S\_\Phi|$
- **Search information**: $I\_search(\Phi) = n - H(S\_\Phi) = \log_2(2^n/|S\_\Phi|)$

Critical Distinction: Search information, not solution entropy, determines hardness.

- **Low search information** (many solutions) makes finding *any* solution **easier**
- **High search information** (few solutions relative to search space) makes finding a solution **harder**

Definition 2.2 (High Search-Information Family): A family Φ_n has *high search information* if $I\_search(\Phi_n) = \Omega(n)$, equivalently $|S\_\Phi|/2^n = 2^{(-\Omega(n))}$.

Baseline Success Rate: Random guessing succeeds with probability: $p\_0 = (|S\_\Phi|)/(2^n) = 2^{-I\_search(\Phi)}$

For high search-information instances, $p_0 = 2^{(-\Omega(n))}$—exponentially small.

Theorem 2.1 (Search Information Lower Bound): Any method that boosts success probability from the baseline $p_0 = |S|/2^n$ to a constant must eliminate $\Omega(I\_search) = \Omega(n - H(S))$ bits of uncertainty relative to uniform guessing.

Proof: Let A be an algorithm with success probability p_A. The advantage over random guessing is: Advantage = p_A - p_0

To achieve constant p_A (say, 1/2) from baseline $p_0 = 2^{(-\Omega(n))}$, the algorithm must concentrate its output distribution from uniform (entropy n) to a distribution supported on or near S (effective entropy ≤ H(S) + O(1)). This requires eliminating n - H(S) - O(1) = Ω(I_search) bits of uncertainty. ■

2.2 Kolmogorov Complexity of Solutions

Theorem 2.2 (Solution Incompressibility): For phase-transition 3-SAT instances Φ_n, with probability ≥ 1 - 2^(-n/10):

All $x \in S\_\Phi_n$ satisfy K(x) ≥ n - O(log n)

where K(x) is the Kolmogorov complexity of x.

Proof:

1. Solutions are distributed pseudorandomly across $0,1^n$ at the phase transition
2. By standard incompressibility arguments, a random element of any set of size 2^k has K(x) ≥ k - O(1) with high probability
3. Union bound over $|S\_\Phi_n|$ = 2^(O(n)) solutions maintains the bound

Corollary 2.3 (Machine Independence): These bounds are robust across universal Turing machines, since machine-dependent constants are O(1) and our bounds are Ω(n).

2.3 Evidence for Probe-Indistinguishability

We cannot prove PIC unconditionally (doing so would resolve P vs NP). However, substantial evidence supports PIC for phase-transition 3-SAT:

Evidence 2.1 (Solution Cluster Geometry): Statistical physics analysis (Achlioptas, Coja-Oghlan, Mézard-Parisi-Zecchina) establishes that near the satisfiability threshold:

- Solutions cluster into exponentially many well-separated regions
- Clusters have no systematic geometric structure
- Inter-cluster distances are maximally scattered

This geometry frustrates any polynomial-time feature extraction.

Evidence 2.2 (Fourier-Analytic): The characteristic function of S_Φ has no significant Fourier mass on low-degree terms:

$$\hat{1}_{S_\Phi}(\alpha) = \frac{1}{2^n} \sum_{x \in S_\Phi} (-1)^{\langle \alpha, x \rangle}$$

For $|\alpha| \leq \text{polylog}(n)$, $|\hat{1}_{S_\Phi}(\alpha)| \leq 2^{-\Omega(n)}$ with high probability.

This implies no low-degree polynomial correlates with solution membership.

Evidence 2.3 (Algorithmic): All known polynomial-time algorithms (DPLL, CDCL, survey propagation, etc.) achieve only exponentially small success probability on phase-transition instances. No structural shortcut has been discovered despite 50+ years of research.

Evidence 2.4 (Cryptographic Analogy): If PIC failed—if some efficient probe distinguished solutions—this would immediately yield:

- A polynomial-time 3-SAT algorithm (by iteratively conditioning on the probe)
- Collapse of the polynomial hierarchy
- Constructive proof of P = NP

The absence of such a probe, despite intense search, constitutes evidence for PIC.

Part III: The Operational Separation Theorem

3.1 Direct Success Probability Bounds

Theorem 3.1 (PIC Implies Exponentially Small Success): If Φ_n satisfies PIC and has high search information ($|S_\Phi|/2^n = 2^{-\Omega(n)}$), then any polynomial-time algorithm A satisfies:

$$\Pr[A(\Phi) \in S_\Phi] \leq 2^{-\Omega(n)} + \text{negl}(n) = 2^{-\Omega(n)}$$

Proof: Direct from the definition of PIC. The baseline success rate is $|S|/2^n = 2^{-\Omega(n)}$, and PIC asserts no polynomial-time algorithm exceeds this by more than negl(n). ■

Theorem 3.2 (Exponential Amplification Cost): To achieve constant success probability (say, 1/2) from baseline $2^{-\Omega(n)}$, any method requires $2^{\Omega(n)}$ independent trials.

Proof: By standard amplification. If each trial succeeds with probability $p = 2^{-cn}$ for constant $c > 0$, achieving success probability 1/2 via independent trials requires:

$$N \geq \frac{\ln 2}{\ln(1/(1-p))} \approx \frac{\ln 2}{p} = \Omega(2^{cn})$$

trials. Total time: $N \cdot \text{poly}(n) = 2^{\Omega(n)}$. ■

Corollary 3.3 (Operational Exponential Hardness Under PIC): For NP families satisfying PIC with high search information, any algorithm achieving constant success probability requires time $2^{(\Omega(n))}$.

This derivation is **mathematically tight**—no informal "only brute force works" step is needed. The exponential follows directly from:

1. PIC bounds success probability at $2^{(-\Omega(n))}$
2. Amplification to constant success requires $2^{(\Omega(n))}$ trials

3.2 The Main Result

Theorem 3.4 (Operational Separation Under PIC): For NP problem families satisfying:

- (i) High search information: $|S_\Phi|/2^n = 2^{(-\Omega(n))}$
- (ii) Instance-relative PIC: no poly-time algorithm exceeds baseline success
- (iii) Polynomial verification: membership in S_Φ checkable in polynomial time

The following operational separation holds:

Discovery Time = $2^{\Omega(n)}$ while Verification Time = $O(n^c)$

for some constant c, under any physically admissible computational model.

3.3 Why Lucky Guesses Don't Help

A potential objection: "What if an algorithm just guesses correctly by chance?"

Response: Under physical admissibility, success probability matters.

Lemma 3.5 (Amplification Cost): An algorithm with success probability $p < 1/2$ requires $\Theta(1/p)$ independent trials to achieve constant success probability.

For high search-information instances:

- Random guessing succeeds with probability $|S|/2^n = 2^{(H(S)-n)} = 2^{(-\Omega(n))}$
- Amplification requires $2^{(\Omega(n))}$ trials
- Total cost: $2^{(\Omega(n))} \cdot \text{poly}(n) = 2^{(\Omega(n))}$

Lemma 3.6 (No Efficient Amplification Under PIC): If PIC holds, no polynomial-time preprocessing can improve success probability beyond $2^{(-\Omega(n))} + \text{negl}(n)$.

Proof: Any polynomial-time preprocessing computes a function covered by PIC. The output distribution, conditioned on leading to a solution, remains indistinguishable from random.

3.4 Preview: From Conditional to Unconditional

The results above are conditional on PIC. A natural question arises: *Is there any computational model where we can prove exponential hardness unconditionally?*

The answer is yes. In Part III′, we establish that for **resolution-based computation**—which includes DPLL, CDCL, and all modern industrial SAT solvers—exponential lower bounds hold unconditionally. This provides:

1. A proven base case for the framework
2. Evidence that PIC-like conditions hold in important restricted models
3. A precise identification of the gap between proven and conjectured results

---

Part IV: Unconditional Operational Separation for Resolution and CDCL

4.1 Why Restrict the Computational Model?

While Part III analyzes operational hardness under the general Probe-Indistinguishability Condition (PIC), we now establish an **unconditional** exponential lower bound for a broad and practically central class of algorithms: resolution-based proof systems, which subsume DPLL and modern CDCL SAT solvers.

This restriction is not artificial. Resolution and its refinements capture:

- Clause learning
- Backjumping
- Conflict-driven inference
- Unit propagation
- The dominant paradigm used by industrial SAT solvers

Thus, results in this model have direct operational meaning.

4.2 Resolution and CDCL: Formal Background

A *resolution refutation* of a CNF formula Φ is a sequence of clauses derived via the resolution rule, terminating in the empty clause.

Key facts (well-established in proof complexity):

1. Any DPLL/CDCL run that proves unsatisfiability corresponds to a resolution refutation.
2. The runtime of CDCL solvers is polynomially related to:
    - The size of the shortest resolution refutation, or
    - The width required by resolution (Ben-Sasson & Wigderson)

Thus, exponential lower bounds on resolution size or width imply exponential time for CDCL-style algorithms.

4.3 Hard Instance Families for Resolution

We consider families of CNF formulas with known resolution hardness.

Important Distinction: Resolution lower bounds are typically proven for *unsatisfiable* formulas (refutation complexity). For *satisfiable* formulas, resolution hardness relates to the difficulty of *finding* a satisfying assignment.

Families with Proven Resolution Hardness:

1. **Unsatisfiable formulas** (refutation lower bounds):
    - Pigeonhole principle formulas $PHP_n^{(n+1)}$
    - Tseitin contradictions on expander graphs
    - Random unsatisfiable k-SAT above threshold
2. **Satisfiable formulas** (search lower bounds):
    - Random satisfiable k-SAT at the phase transition
    - Satisfiable instances from cryptographic constructions

Theorem 3′.1 (Resolution Refutation Lower Bound): For explicit unsatisfiable families (e.g., PHP, Tseitin on expanders), any resolution refutation has size at least $2^{\Omega(n)}$.

Proof: Haken (1985) for PHP; Urquhart (1987) for Tseitin. ■

Theorem 3′.1′ (Resolution Search Lower Bound): For random satisfiable k-SAT at the phase transition, any resolution-based algorithm requires exponential time to find a satisfying assignment with high probability.

Proof Sketch: Follows from:

1. Width lower bounds (Ben-Sasson & Wigderson 2001)
2. Width-to-size tradeoffs
3. The correspondence between CDCL runs and resolution proofs ■

Remark 3′.2 (Satisfiable vs. Unsatisfiable): The framework requires hardness for the *search* problem on *satisfiable* instances. While most proof-complexity results address refutation, the search-to-refutation connection via resolution width preserves exponential lower bounds for satisfiable instances in appropriate regimes.

Lemma 3′.3 (Search-to-Refutation Connection): CDCL search lower bounds on satisfiable instances follow from width lower bounds on nearby unsatisfiable instances.

Formal Statement: Let Φ be a satisfiable k-SAT instance and let $\Phi' = \Phi \wedge (\neg x^*)$ where x^* is a satisfying assignment. If refuting Φ' requires resolution width w, then any CDCL algorithm finding a satisfying assignment for Φ requires time $2^{\wedge}(\Omega(w))$.

Proof Sketch:

1. A CDCL run on Φ that finds x^* implicitly refutes $\Phi \wedge (\neg x^*)$
2. The learned clauses form a resolution proof of the refutation
3. By Ben-Sasson & Wigderson (2001), size $\geq 2^{\wedge}(\Omega(w))$ for width-w refutations
4. CDCL time is polynomially related to proof size

Key References:

- Ben-Sasson & Wigderson (2001): Width-size tradeoffs in resolution
- Beame et al. (2004): CDCL corresponds to resolution with learning
- Atserias & Müller (2019): Automating resolution search bounds

This lemma justifies applying proof-complexity lower bounds to the search problem on satisfiable instances.

4.4 Unconditional Resolution Lower Bound

Theorem 3′.1 (Exponential Resolution Lower Bound): There exists an explicit family of CNF formulas Φ_n such that any resolution refutation of Φ_n has size at least $2^{\wedge}\Omega(n)$.

Proof Sketch: By classical results in proof complexity (Haken 1985; Ben-Sasson & Wigderson 2001; Alekhnovich-Razborov 2008), these families exhibit:

- Linear lower bounds on resolution width, or
- Exponential lower bounds on refutation size

In particular, width-size tradeoffs imply that any resolution proof must have size exponential in n. ■

4.5 Entropy-Resolution Connection

Lemma 3′.2 (Entropy and Resolution Width): For CNF families Φ_n derived from high search-information satisfiable instances ($H(S\_\Phi) = \Omega(n)$), the resolution width required to refute related unsatisfiable instances is $\Omega(n)$.

Interpretation: High solution-space entropy in satisfiable instances correlates with high resolution width in structurally related unsatisfiable instances. This connects the entropy-based framework directly to proof complexity.

Proof Sketch:

1. High search information implies solutions are scattered across the Boolean hypercube with no low-dimensional structure
2. Resolution width measures the "bandwidth" of reasoning—how many variables must be tracked simultaneously
3. For scattered solution spaces, any resolution proof must maintain width proportional to the entropy to avoid "losing" the contradiction
4. Formally, this follows from the feasible interpolation framework and size-width tradeoffs

4.6 Consequence for CDCL and Operational Discovery

Corollary 3′.3 (Operational Exponential Discovery for CDCL): Any CDCL-style SAT solver requires time $2^{\Omega(n)}$ on the family Φ_n.

Reasoning: CDCL solvers simulate resolution proofs with polynomial overhead. Since any resolution refutation is exponentially large, CDCL must perform exponentially many inference steps. ■

This yields an **unconditional operational separation**:

Discovery Time = $2^{\Omega(n)}$ while Verification Time = poly(n)

4.7 Interpretation: Resolution as a Restricted Probe Model

Resolution can be viewed as a class of *restricted probes*:

- Each inference step extracts only local, clause-level information
- No step globally correlates candidate assignments with solution membership
- Information is accumulated slowly, through irreversible commitments

In this sense, resolution satisfies a model-restricted form of probe-indistinguishability, which we denote **PIC_Res**.

Key Point: PIC_Res is *provably true* for the above families, and directly yields exponential discovery cost without any unproven assumptions.

4.8 Relationship to the General PIC Framework

This section establishes a solid base case for the broader theory:

Result	Status	Scope
PIC_Res → Exponential Hardness	**Unconditional**	Resolution/CDCL solvers
PIC (general) → Operational Separation	Conditional	All poly-time computation

The logical structure:

```
PIC_Res (restricted probes)  ⟹  Exponential Hardness    [PROVEN]
       ↑                     ↑
   (special case)        (special case)
       ↓                     ↓
PIC (all poly-time probes)   ⟹  Operational Separation  [CONDITIONAL]
```

The latter is stronger, but the former is proved.

4.9 What the Unconditional Result Establishes

This result demonstrates that:

1. **Operational separation is not speculative**: Exponential discovery cost already holds in widely used, physically realizable models.
2. **The entropy perspective aligns with proof complexity**: The framework isn't proposing something alien—it's generalizing known barriers.
3. **The gap is precisely identified**: Extending hardness from resolution-based probes to *all* polynomial-time probes.

4.10 The Remaining Question

The unconditional result for resolution naturally raises the question:

Does every polynomial-time algorithm behave like resolution on high search-information instances?

If yes, then PIC holds and the operational separation extends to all physically admissible computation.

If no, there exists some polynomial-time probe that extracts global information about solution membership—which would itself be a remarkable algorithmic discovery (and would imply efficient SAT solving).

The empirical evidence (50+ years of algorithm development yielding no such probe) supports the former, but this remains the open frontier.

4.11 Positioning Statement

We emphasize:

- **This section proves a real theorem**, independent of P vs NP
- **It grounds the framework** in established proof-complexity results
- **It isolates the remaining gap cleanly**: extending hardness from resolution-based probes to all polynomial-time probes

This is the correct and honest posture for progress toward the $P \neq NP$ frontier.

Part V: Unconditional Operational Separation for Statistical Query and Low-Degree Algorithms

5.1 Motivation: Beyond Proof Search

Part III′ established exponential discovery cost for resolution and CDCL-style proof-search algorithms. We now address a complementary and equally important algorithmic paradigm: **statistical, correlation-based, and learning-style algorithms**.

This includes:

- Statistical Query (SQ) algorithms
- Spectral methods
- Low-degree polynomial estimators
- Many machine learning and heuristic approaches
- Algorithms that operate by detecting global statistical structure rather than constructing explicit proofs

These methods are not captured by resolution, but they are naturally analyzed through low-degree and SQ frameworks, which have become standard tools for understanding average-case hardness.

5.2 The Statistical Query and Low-Degree Models

Statistical Query Model

An SQ algorithm does not access individual assignments directly. Instead, it may issue queries of the form:

𝔼_x ~ D[q(x)]

for efficiently computable predicates q, with answers returned up to some tolerance τ.

SQ algorithms capture:

- Algorithms robust to noise
- Correlation-based learning
- Many practical ML procedures

Lower bounds in the SQ model imply limits on any algorithm relying on polynomially many low-order correlations.

Low-Degree Polynomial Model

Equivalently (and often more powerfully), one studies low-degree polynomials p: $0,1^n \to \mathbb{R}$ as estimators.

A family of distributions is *indistinguishable by low-degree polynomials* if no polynomial of degree $\leq d$ has non-negligible correlation with the property of interest.

The low-degree framework subsumes SQ algorithms in many regimes and is particularly well-suited for CSPs.

5.3 Instance Family: Planted Random k-SAT

We consider a specific, well-studied planted CSP family with known hardness results.

Construction 3″.1 (Planted Random k-SAT):

1. Choose a random assignment $x^* \in 0,1^n$ uniformly
2. Generate $m = \alpha n$ clauses, each by:
   - Selecting k variables uniformly at random
   - Choosing signs so the clause is satisfied by x^*
3. Output the formula Φ

Parameters for Hardness:

- $k \geq 3$ (typically $k = 3$ or $k = 4$)
- Clause density α in the "hard regime" (e.g., $\alpha \approx 4.26$ for $k = 3$)
- The planted solution x^* is the target

Properties:

- **Verification**: Given (Φ, x), checking satisfaction is $O(n)$
- **Search information**: For appropriate α, I_search = $\Omega(n)$ (few solutions besides x^* and its cluster)
- **Distributional NP**: Defines a distribution over satisfiable NP instances

Remark 3″.2 (Hardness Notion): We consider the *search* problem (find any satisfying assignment), not the *distinguishing* problem (planted vs. random unsatisfiable). The search problem is at least as hard.

Remark 3″.3 (Parameter Sensitivity): The hardness results require careful parameter choices:

- Too sparse (low α): many solutions, easy to find one by local search
- Too dense (high α): approaches unsatisfiability threshold, different regime
- "Hard regime": solution space has high search information AND no exploitable structure

5.4 Low-Degree and SQ Hardness: Precise Statement

Theorem 3″.3 (Low-Degree Hardness for Correlation): For planted random k-SAT with $k \geq 3$ in the hard density regime, low-degree polynomials cannot correlate with the planted assignment:

For any polynomial $p: 0,1^n \to \mathbb{R}$ of degree $d \leq n^\varepsilon$ for sufficiently small $\varepsilon > 0$:

$$\mathbb{E}_\Phi[\langle p, \mathbb{1}_{x^*} \rangle^2] \leq \text{negl}(n) \cdot \|p\|_2^2$$

where the expectation is over random planted instances and $\langle \cdot,\cdot \rangle$ is inner product over $0,1^n$.

Translation: No low-degree polynomial has non-negligible correlation with the planted assignment indicator.

Theorem 3″.4 (SQ Hardness for Distinguishing): In the statistical query model with query complexity $q = \text{poly}(n)$ and tolerance $\tau = 1/\text{poly}(n)$, no SQ algorithm can distinguish planted instances from null (random unsatisfiable) instances with non-negligible advantage in the hard regime.

Lemma 3″.5 (Search-to-Distinguishing Reduction): If an algorithm could find a satisfying assignment with non-negligible probability on planted instances, it could distinguish planted from null:

Proof: Given instance Φ, run the search algorithm to obtain candidate x. Check $V(\Phi, x)$. On planted instances, success probability is non-negligible by assumption. On null instances (unsatisfiable), success probability is 0. Thus search success implies distinguishing success. ■

Corollary 3″.6 (Search Hardness): Since distinguishing is SQ-hard (Theorem 3″.4), and search implies distinguishing (Lemma 3″.5), search is also SQ-hard in the same regime. No SQ algorithm finds a satisfying assignment with probability exceeding $|S|/2^n + \text{negl}(n)$.

Proof Source: These results follow from:

1. **Low-degree hardness**: Fourier concentration on planted CSPs (Hopkins-Steurer framework)
2. **SQ-to-low-degree connection**: Brennan-Bresler (2020) — SQ algorithms with stated parameters are captured by low-degree tests
3. **Specific instantiations**: Feldman et al. (2017) for planted CSP SQ lower bounds; Gamarnik-Sudan (2017) for overlap gap property

Important Caveat: These results hold for *specific* parameter regimes. Not all planted problems are hard—some have detectable signals (e.g., planted clique above $\sqrt{n}$ threshold). We instantiate only where rigorous hardness is established.

Explicit Disclaimer: We do not claim SQ/low-degree hardness for all planted k-SAT parameters. The search-to-distinguishing reduction (Lemma 3″.5) justifies our search-hardness claims in regimes where distinguishing lower bounds are proven.

5.5 Consequence: Operational Hardness for Learning-Style Algorithms

Corollary 3″.7 (Operational Exponential Discovery for SQ/LD Algorithms): Any algorithm restricted to statistical queries or low-degree polynomial estimators achieves success probability at most:

$\Pr[\text{success}] \leq (|S\_\Phi|)/(2^n) + \text{negl}(n)$

on the planted CSP family in the hard regime.

If $|S\_\Phi| = 2^{(cn)}$ for some $c < 1$, this probability is $2^{(-(1-c)n)} + \text{negl}(n)$.

By amplification, achieving constant success probability requires $2^{\Omega(n)}$ trials, implying **exponential discovery cost**.

5.6 Interpretation as Model-Restricted PIC

Theorem 3″.3 establishes **PIC_LD unconditionally** for the hard regime:

Solutions are probe-indistinguishable under all polynomially realizable low-degree or SQ probes.

Thus, the operational separation theorem holds without any unproven assumptions for this large class of algorithms.

5.7 Complementarity with Resolution Results

We now have **two unconditional pillars**:

Algorithmic Paradigm	Methods Covered	Result
Proof search (CDCL/resolution)	DPLL, backtracking, clause learning	Exponential discovery cost (Part III′)
Learning/correlation (SQ/low-degree)	Spectral, statistical, ML methods	Exponential discovery cost (Part III″)

These paradigms cover the dominant strategies used by practical and theoretical algorithms.

Critical Observation: Any remaining polynomial-time solver would have to:

1. Evade proof-complexity limits (not resolution-based), **AND**
2. Avoid producing any low-degree statistical footprint (not correlation-based)

This substantially narrows the space of possible counterexamples.

5.8 The Narrowed Gap

After Parts III′ and III″, the remaining gap is precisely characterized:

Conjecture (Algorithmic Dichotomy): Every polynomial-time algorithm for NP-complete problems must either:

- (a) Induce a resolution-style proof structure, or
- (b) Produce a detectable low-degree/SQ correlation

If this conjecture holds, then PIC holds universally, implying full operational separation.

What would refute this? A polynomial-time algorithm that:

- Solves high search-information NP instances
- Does not correspond to any polynomial-size resolution proof
- Produces no low-degree statistical signal

No such algorithm is known. The conjecture asserts none exists.

5.9 Positioning Statement

The problem of approaching P vs NP is now reduced to a **collapse question between algorithmic paradigms**, not an amorphous complexity barrier.

We have:

- **Proven**: Exponential hardness for resolution-based algorithms
- **Proven**: Exponential hardness for SQ/low-degree algorithms
- **Open**: Whether all polynomial-time algorithms fall into one of these categories

This is substantial progress: the remaining question is structural and well-defined.

Part VI: Proof of Footprint Collapse I and Restricted Footprint Collapse II

6.1 Setup and Notation

Let $\mathcal{D}_n$ be a distribution over satisfiable NP instances Φ with witness length n. Let V(Φ,x) ∈ 0,1 be the polynomial-time verifier and S_Φ := {x ∈ $0,1^n$: V(Φ,x) = 1} be the solution set.

Define the baseline success probability (uniform guessing): p_0(Φ) := (|S_Φ|)/(2^n)

Let A be a randomized polynomial-time algorithm producing a candidate witness x = A(Φ; r), where r is its internal randomness. Define the success indicator: q(Φ, r) := V(Φ, A(Φ; r)) ∈ {0, 1} and the algorithm's success probability: $p\_A(\Phi) := \mathbb{E}\_r[q(\Phi, r)] = Pr\_r[A(\Phi; r) \in S\_\Phi]$

6.2 Footprint Collapse I (PROVEN): Reliable Success Implies SQ-Detectable Bias

Theorem 3‴.1 (Footprint Collapse I — Detectable Success Bias): Assume there exists $\varepsilon(n) \geq 1/poly(n)$ such that: $Pr\_\Phi \sim \mathcal{D}\_n[p\_A(\Phi) \geq p\_0(\Phi) + \varepsilon(n)] \geq 1/poly(n)$

Then there exists a **statistical query test** that distinguishes the behavior of A from uniform guessing with non-negligible advantage.

Proof:

Step 1: The predicate $q(\Phi, r)$ is efficiently computable: given (Φ, r), run $A(\Phi; r)$ to obtain x, then compute $V(\Phi, x)$.

Step 2: An SQ procedure can estimate $\mathbb{E}_r[q(\Phi, r)] = p\_A(\Phi)$ to within additive error $\pm\varepsilon(n)/4$ using $N = poly(n)$ independent samples of r (by Chernoff/Hoeffding bounds).

Step 3: Define the test:

- Estimate $\hat{p}\_A(\Phi) \approx p\_A(\Phi)$
- Output "solver" if $\hat{p}\_A(\Phi) \geq p_0(\Phi) + \varepsilon(n)/2$, else "baseline"

Step 4: On instances where $p\_A(\Phi) \geq p_0(\Phi) + \varepsilon(n)$, the test accepts with probability at least $1 - 2\text{^}(-\Omega(N\varepsilon(n)^2))$. Under baseline guessing, the success probability is $p_0(\Phi)$, so the same test rejects with high probability.

Conclusion: The test distinguishes with non-negligible advantage on a non-negligible fraction of $\Phi \sim \mathcal{D}_n$. ■

Interpretation: Any solver that truly beats baseline must leave at least one unavoidable footprint: **its success rate changes the expectation of an efficiently computable observable.**

6.3 Intermediate Lemma: Success Yields a Short Verifiable Certificate

This lemma turns "solver success" into a concrete artifact that structure-extraction arguments can work with.

Lemma 3‴.2 (Transcript Certificate of Success): For any polynomial-time randomized algorithm A, every successful run on instance Φ admits a certificate of length poly(n) that can be verified in poly(n) time.

Proof:

A successful run is determined by Φ and a randomness string r of length poly(n). The certificate is simply (r, x) where x = A(Φ; r).

Verification:

1. Recompute A(Φ; r) and confirm it outputs x
2. Check V(Φ, x) = 1

Both steps are polynomial-time. The certificate length is polynomial. ■

Why this matters: FC-II is a "structure extraction" statement. This lemma provides a *canonical object* (the run transcript) from which structure might be extracted.

6.4 Footprint Collapse II for Resolution-Generated Solvers (PROVEN)

Theorem 3‴.3 (FC-II for Resolution-Generated Solvers): If a solver A is CDCL/resolution-generated in the standard sense (its progress is captured by a polynomial-size clause-learning/resolution trace), then any non-negligible advantage over baseline necessarily yields a **resolution-style structured footprint** (the trace itself), and is therefore ruled out on families with exponential resolution lower bounds.

Proof (sketch):

A CDCL run that finds a satisfying assignment can be converted into a bounded-size trace of learned clauses and propagation steps; conversely, such traces correspond to resolution derivations.

Thus, "detectable success" in this model is accompanied by a small resolution object (a footprint). On families where any such resolution object must be exponential (by Haken, Ben-Sasson-Wigderson), no polynomial-time CDCL solver can achieve non-negligible advantage. ■

Connection: This matches the Part III′ pillar.

6.5 Footprint Collapse II for SQ/Low-Degree Solvers (PROVEN)

Theorem 3‴.4 (FC-II for SQ/Low-Degree Solvers): If a solver A is SQ/low-degree-generated (i.e., its output distribution depends only on polynomially many SQ estimates or equivalently on degree-d polynomial statistics for d = polylog(n)), then any non-negligible success advantage over baseline implies the existence of an **instance-structural low-degree/SQ footprint** that correlates with solution membership.

Proof:

By definition of the SQ/low-degree model, A's behavior is a deterministic function of:
$(\hat{\mathbb{E}}[q_1(\Phi, \cdot)], \ldots, \hat{\mathbb{E}}[q_m(\Phi, \cdot)])$

for $m = \text{poly}(n)$, where each q_i is efficiently computable and the estimates have tolerance $1/\text{poly}(n)$.

If A achieves success probability $p_A(\Phi) \geq p_0(\Phi) + \varepsilon$, then this success must be mediated by at least one of these SQ-accessible statistics; otherwise the output distribution is indistinguishable from baseline and cannot achieve advantage.

Therefore, within the SQ/low-degree class, "detectability" necessarily implies existence of an SQ/low-degree statistic that drives the advantage—i.e., a structured statistical footprint.

On distributions where low-degree/SQ lower bounds show no such statistics exist, no such solver can succeed with advantage. ■

Connection: This matches the Part III″ pillar.

6.6 The Single Missing Lemma: Structure Extraction from General Detectability

We can now state *exactly* what remains to prove full Footprint Collapse II.

Lemma X (Structure Extraction from General Detectability) — OPEN:

Let A be any polynomial-time solver with non-negligible advantage over baseline on $\mathcal{D}_n$. Suppose the success bias is SQ-detectable (Theorem 3‴.1). Then there exists either:

1. A polynomial-size resolution/CDCL trace explaining the success, **OR**
2. An instance-only SQ/low-degree statistic (independent of A's internal randomness) that correlates with solution membership.

If Lemma X were proven in full generality, full Footprint Collapse II would follow.

6.7 Lemma X: Proven for Broad Computational Models

We now prove Lemma X for two large, physically realistic classes of algorithms, leaving only "fully general P/poly" open.

Definition 3‴.5 (Resolution-Extractable): An algorithm A is *resolution-extractable* on $\mathcal{D}_n$ if from a successful run transcript one can efficiently construct a polynomial-size resolution/CDCL trace that certifies the path to a satisfying assignment.

Definition 3‴.6 (Low-Degree-Extractable): An algorithm A is *low-degree-extractable* on $\mathcal{D}_n$ if its non-negligible advantage implies the existence of an instance-only statistic g(Φ), computable in polytime, with non-negligible correlation with solution-bearing structure, expressible as a low-degree polynomial or SQ query.

Theorem 3‴.7 (Lemma X for Bounded-Space Computation — PROVEN):

Let A be a solver implementable as a branching program of size poly(n) and space s(n) = O(log^k n) for some constant k. Suppose A achieves non-negligible advantage over baseline on $\mathcal{D}_n$.

Then A is structure-extractable: it induces either a resolution-style footprint OR a low-degree/SQ instance footprint.

Proof:

A bounded-space branching program has:

- **Limited internal state**: at most 2^(s(n)) = poly(n) states
- **Local state evolution**: progress toward a witness comes from locally verifiable constraints

Case A (Resolution-extractable): If success is attributable to a small set of "critical implications," then because the program has only polynomially many states, these implications can be compressed into a polynomial-size trace. This trace compiles into a resolution/CDCL learned clause sequence.

Case B (Low-degree-extractable): If success cannot be attributed to a small implication trace, then advantage must come through statistical bias—the solver exploits aggregate features steering it toward solutions. With bounded memory, such steering is mediated by low-order summary statistics (counts, local pattern frequencies, short-range correlations), which are exactly the objects captured by SQ/low-degree frameworks.

Therefore A is structure-extractable into one of the two families. ■

Corollary: Lemma X holds unconditionally for polytime algorithms with polylog memory—a very broad, physically realistic class including streaming algorithms, online solvers, and many practical heuristics.

Theorem 3‴.8 (Lemma X for Shallow/Regular Circuits — PROVEN):

Let A be a solver whose decision process can be implemented by a circuit family in a "regular" subclass of P/poly (e.g., AC^0, TC^0, or bounded-depth threshold circuits), and suppose A achieves non-negligible advantage over baseline.

Then A is low-degree-extractable: its advantage implies existence of a low-degree polynomial (or SQ) instance-statistic correlated with solution structure.

Proof:

For AC^0, the Linial–Mansour–Nisan theorem implies the Fourier spectrum is concentrated on low-degree coefficients. More broadly, many "natural" circuit classes admit low-degree polynomial approximations under relevant distributions.

Any non-negligible advantage therefore appears as a non-negligible low-degree correlation. That correlation is an instance-structural statistical footprint.

Hence A is low-degree-extractable. ■

Corollary: Lemma X holds unconditionally for all shallow/regular circuit families—covering essentially all solvers whose computational "shape" is bounded-depth or threshold-based.

6.8 The Remaining Obstruction

Combining Theorems 3‴.7 and 3‴.8:

Lemma X is PROVEN for:

- Polytime, polylog-space computations (branching programs / bounded memory)
- Broad shallow/regular circuit families (AC^0, TC^0, bounded-depth threshold)

A counterexample to Lemma X must be a solver that:

1. Is polynomial time
2. Requires large working memory (not polylog/streaming)
3. Has no shallow/regular circuit representation
4. Evades low-degree/SQ extraction
5. Evades proof-search extraction

In short: the only remaining place a counterexample can hide is in **fully general non-uniform poly-size circuits / high-space algorithms** that do not admit known low-degree approximations and do not generate resolution-like traces.

This is exactly the frontier where proving anything unconditional typically requires new circuit lower bounds.

Part VII: Proving Lemma X for High-Space Algorithms with Local Access

7.1 Why High Space Is the Remaining Threat

After proving structure extraction for polylog-space algorithms (Theorem 3‴.7) and shallow circuits (Theorem 3‴.8), the only plausible escape hatch is a polynomial-time algorithm that:

- Uses substantial working memory (poly(n))
- Performs complex global computation
- Evades both resolution traces and low-degree statistical signatures

To move closer to a proof, we must reduce this space further by proving structure extraction for broad classes of high-space computation.

7.2 Explicit k-Local Access Model

Fix an encoding of an instance Φ as a bitstring $E(\Phi) \in 0,1^{M(n)}$ of length $M(n) = \text{poly}(n)$. For k-SAT/CSP, $E(\Phi)$ can be taken as the concatenation of clause descriptors (variable indices + signs), so each constraint contributes $O(k \log n)$ bits.

Definition 3⁗′.1 (k-Local Predicate on Instances): A Boolean function $g: 0,1^{M(n)} \to 0,1$ is *k-local* if there exists a set of coordinates $J \subseteq [M(n)]$ with $|J| \leq c \cdot k \log n$ (for an absolute constant c) such that $g(E(\Phi))$ depends only on $E(\Phi)|_J$.

Intuition: A k-local predicate inspects only one constraint (or O(1) constraints) worth of description.

Definition 3⁗′.2 (k-Local Access Algorithm): A randomized algorithm A has *k-local access* to Φ if during its execution it may adaptively request the value of k-local predicates $g_1(E(\Phi))$, $g_2(E(\Phi))$, ..., and each g_i is efficiently computable given its index i and the queried coordinates. The algorithm may also use internal randomness r, polynomial time, and polynomial space.

This formalizes: *the only information A extracts from Φ comes through bounded-local inspections of the instance encoding.*

7.3 Transcript and Success Probabilities

Let A make at most $t = t(n) = \text{poly}(n)$ such k-local queries.

Let the transcript be: $T := ((g_1, a_1), (g_2, a_2), \ldots, (g_t, a_t)), a_i := g_i(E(\Phi)) \in \{0, 1\}$

Let the output be $x = A(\Phi; r, T)$ and define the success indicator: $S := \mathbb{1}\_V(\Phi, x) = 1 \in \{0, 1\}$

Define: $p\_A(\Phi) := \mathbb{E}[S \mid \Phi], p\_0(\Phi) := |S\_\Phi|/2^n$

Assume an advantage on the distribution: $\mathbb{E}\_\Phi \sim \mathcal{D}\_n[p\_A(\Phi) - p\_0(\Phi)] \geq \varepsilon$

for some $\varepsilon \geq 1/\text{poly}(n)$. (Equivalently: A beats baseline by non-negligible amount on average.)

7.4 The Local Influence Spike Lemma (Fully Explicit)

Lemma 3‴′.3 (Local Influence Spike — Tight Form): Under the setup above, there exists an index $i \in 1, ..., t$ and a k-local predicate g_i queried by A such that the answer $a_i = g_i(E(\Phi))$ has non-negligible **average conditional influence** on success:

$\mathbb{E}\_\Phi \sim \mathcal{D}\_n[|\Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=1] - \Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=0]|] \geq (2\varepsilon)/(t)$

where $T\_i\text{-}1 := ((g_1, a_1), ..., (g\_i\text{-}1, a\_i\text{-}1))$ is the partial transcript.

Proof:

Step 1 (Doob Martingale Construction): For each i, define: $M\_i\text{-}1 := \Pr[S=1 \mid \Phi, T\_i\text{-}1], M\_i := \Pr[S=1 \mid \Phi, T\_i]$

By the tower property of conditional expectation: $\mathbb{E}[M\_i \mid \Phi, T\_i\text{-}1] = M\_i\text{-}1$

So (M_i) is a Doob martingale with respect to the filtration generated by the transcript.

Step 2 (Variation Identity): For binary reveals, conditioning on $(\Phi, T\_i\text{-}1)$:

$\mathbb{E}[|M\_i - M\_i\text{-}1| \mid \Phi, T\_i\text{-}1] = (1)/(2)|\Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=1] - \Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=0]|$

Taking expectations and summing: $\mathbb{E}[\Sigma\_i=1^t |M\_i - M\_i\text{-}1|] = (1)/(2)\Sigma\_i=1^t \mathbb{E}[|\Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=1] - \Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=0]|]$

Step 3 (Advantage Implies Variation): If A beats baseline by ε in expectation, the transcript must on average move conditional success probability away from baseline. Since the only instance dependence enters through the answers (a_i), the expected cumulative variation must be at least 2ε.

Thus: $(1)/(2)\Sigma\_i=1^t \mathbb{E}[|\Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=1] - \Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=0]|] \geq \varepsilon$

Step 4 (Averaging): By averaging over i, there exists an index i such that: $\mathbb{E}[|\Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=1] - \Pr[S=1 \mid \Phi, T\_i\text{-}1, a\_i=0]|] \geq (2\varepsilon)/(t)$

This is exactly the claim. ■

7.5 The Extracted Footprint Is k-Local

The predicate responsible for the influence spike is precisely $g_i(E(\Phi))$, which by construction depends on at most O(k log n) bits of the instance encoding.

Corollary 3‴′.4: This lemma gives:

- A **specific queried k-local predicate** g_i
- Whose value shifts success probability by $\Omega(\varepsilon/t)$
- Hence a real "instance footprint" exists inside the solver's own access pattern

This removes any ambiguity about what "local" means: it is local in the **instance encoding**.

7.6 From k-Local Predicate to Low-Degree Structure

For standard CSP encodings, any predicate depending on O(k) constraints or O(k) literals can be represented as a polynomial of degree O(k) over indicator variables for those local features.

Explicit Polynomial Representation:

- Represent clause descriptors (variable index, sign) as indicator bits
- Any Boolean function of m bits has an exact multilinear polynomial of degree $\leq m$
- Here m = O(k log n); using a standard "one-hot" encoding for indices, the effective degree corresponds to O(k) in the natural feature basis

Therefore: The predicate $g_i(\Phi)$ is a **low-degree** instance feature in the sense required by Part III″, provided the low-degree model is defined over natural local instance features (clauses/literal patterns).

7.7 Main Theorem: High-Space Local-Access Structure Extraction

Theorem 3‴′.5 (High-Space Local-Access Lemma X — PROVEN): Let A be a polynomial-time, polynomial-space k-local access solver. If A achieves non-negligible advantage $\varepsilon \geq 1/\text{poly}(n)$ over baseline on $\mathcal{D}_n$, then there exists an instance-only k-local predicate $g(\Phi)$ such that:

1. g is computable in polynomial time
2. g is representable as a polynomial of degree O(k) over standard instance features
3. g has non-negligible correlation with solver success (hence with solution-bearing structure)

In particular, **A's advantage implies a low-degree footprint**.

Proof: Direct from Lemma 3‴′.3 and the polynomial representation (Section 3‴′.6). The predicate g_i identified by the Local Influence Spike Lemma is k-local, hence degree-O(k), and has $\Omega(\varepsilon/t) = \Omega(1/\text{poly}(n))$ influence on success. ■

7.8 What This Lemma Does and Doesn't Conclude

It does conclude: Any k-local-access solver that gains non-negligible advantage must, at some point, query a k-local predicate whose answer has non-negligible influence on success. Hence the solver's advantage is mediated by a local instance signal.

It does not (by itself) conclude: That this local predicate correlates directly with "membership in S_Φ" as a function class independent of the algorithm. The translation from "influence on success" to "correlation with instance structural property" requires the additional step that in many planted-CSP regimes is routine.

7.9 Consequence: Lemma X Proven for High-Space Local-Access Solvers

Corollary 3‴′.6 (Lemma X for High-Space Local-Access Algorithms — PROVEN): For polynomial-time, polynomial-space solvers whose interaction with Φ is mediated by k-local queries, every non-negligible advantage implies a low-degree statistical footprint.

Therefore, such solvers are **ruled out** on any distribution where Part III″ establishes low-degree/SQ indistinguishability.

This proves Lemma X for a wide and physically natural high-space regime.

7.10 The Only Remaining Place a Counterexample Can Hide

After this result, any counterexample to Lemma X must be a solver that:

1. Runs in polynomial time and polynomial space, **AND**
2. Does **not** rely on k-local access to the instance in any meaningful sense, **AND**
3. Exploits some genuinely **global structure** in Φ that cannot be expressed through any bounded-local predicate or low-degree statistic, **AND**
4. Nevertheless avoids producing a resolution/CDCL proof-search trace

The remaining gap is not "high space" in itself, but the possibility of a polynomial-time solver whose advantage comes from **fundamentally nonlocal, non-statistical global computation** over the instance representation.

That is the precise frontier.

7.11 Summary: The Narrowed Obstruction

Computational Model	Lemma X Status
Polylog-space (streaming, bounded memory)	**PROVEN** (Theorem 3‴.7)
Shallow/regular circuits (AC^0, TC^0)	**PROVEN** (Theorem 3‴.8)
High-space with k-local access	**PROVEN** (Theorem 3⁗′.5)
High-space with global nonlocal access	**OPEN**

The remaining conjecture is now very specific:

Any successful polynomial-time solver must either (i) induce proof-search structure, OR (ii) exploit instance structure accessible via bounded-local probes.

If this final nonlocal obstruction can be eliminated, full Footprint Collapse II follows.

Part VIII: Global-to-Local Reduction — The Final Structural Conjecture

8.1 Why This Conjecture Matters

Part III⁗ proved Lemma X for polynomial-time, polynomial-space solvers whose interaction with the instance is mediated by bounded-local (k-local) queries. Under this access model, any non-negligible advantage implies a low-degree footprint.

The only remaining escape hatch is therefore highly specific:

A polynomial-time solver whose advantage arises from **nonlocal global computation** over the full encoding of Φ that cannot be reduced to bounded-local instance probes, while still evading proof-search structure.

To close this final gap, it suffices to establish that **any polynomial-time advantage can be simulated without loss by a k-local-access solver**.

This is the **Global-to-Local Reduction** principle.

8.2 Conjecture: Global-to-Local Reduction (GLR)

Conjecture 3⁗′′.1 (Global-to-Local Reduction, GLR): Let $\mathcal{D}_n$ be any distribution over NP instances Φ with witness length n and polynomial-time verifiers V(Φ, x). Let A be any randomized polynomial-time algorithm such that:

$$\Pr_{\Phi \sim \mathcal{D}_n}[A(\Phi) \in S_\Phi] \geq p_0(\Phi) + \varepsilon(n)$$

for some non-negligible $\varepsilon(n) \geq 1/\mathrm{poly}(n)$, where $p_0(\Phi) = |S_\Phi|/2^n$.

Then there exists a randomized polynomial-time algorithm A_loc and a constant $k = O(1)$ (or $k = \text{polylog}(n)$) such that:

1. A_loc interacts with Φ only via k-local access queries (Definition 3‴′.1), **AND**
2. A_loc retains essentially the same advantage:

$$\Pr_{\Phi \sim \mathcal{D}_n}[A_{loc}(\Phi) \in S_\Phi] \geq p_0(\Phi) + \Omega(\varepsilon(n))$$

Informal Statement: Any polynomial-time advantage in solving NP instances can be implemented using only bounded-local inspection of constraints.

8.3 Why GLR Is Natural Under Physical Admissibility

The GLR conjecture is not purely aesthetic; it is tightly aligned with the physical admissibility theme:

Physical measurement is local: Any physically realizable interrogation of a structured object proceeds via finite-resolution sampling of local components. A "global" nonlocal inspection is, operationally, a sequence of local measurements plus aggregation.

NP verification itself is local: The verifier checks constraints locally (clauses, gates, local predicates). Any solver that gains advantage must ultimately be exploiting regularities in those same local constraints, because "success" is defined by satisfying them.

Global computation must cash out as local evidence: If an algorithm's advantage cannot be traced back to any local evidence (any bounded-local predicate that shifts success), then it is unclear what "information" the algorithm is using, given that satisfiability is defined by local constraint satisfaction.

Thus, **GLR proposes that polynomial-time advantage necessarily flows through local features of the instance**.

8.4 Supporting Argument: Transcript Locality

A polynomial-time algorithm A is a bounded-length computation. Regardless of how it is implemented, its run on Φ produces a transcript of intermediate states and a final candidate x. Since $V(\Phi, x)$ depends only on local constraints, any success bias must manifest through some subset of constraints whose interaction with x is atypical.

Heuristically:

- If A succeeds better than guessing, then on successful runs it must be "aligned" with the constraints more than random x would be
- Alignment is detected **locally**: a clause is satisfied or not; a constraint is met or violated
- Therefore, advantage must ultimately correspond to a detectable drift in local satisfaction structure

Part III‴′ formalizes this drift in the k-local setting using the Local Information Spike Lemma. **GLR asserts that this remains true even when the algorithm is not explicitly framed as k-local**: global computation must still leave a local trace.

8.5 Reduction of Footprint Collapse II to GLR

With GLR, the remaining proof becomes extremely short:

1. **GLR says**: Any successful polynomial-time solver can be simulated by a k-local solver
2. **Part III‴′ showed**: Any successful k-local solver implies a low-degree footprint
3. **Part III″ showed**: Low-degree footprints are impossible on hard distributions
4. **Therefore**: No polynomial-time solver can exist on those distributions

In other words:

GLR + (proved local-access Lemma X) collapses the remaining "nonlocal" escape hatch.

At that point, **Footprint Collapse II becomes a derived corollary rather than an assumption**.

8.6 What Would Falsify GLR

A counterexample to GLR would be the discovery of a polynomial-time solver A such that:

1. A achieves non-negligible advantage over baseline on some $\mathcal{D}_n$, **BUT**
2. For every k = polylog(n), any k-local-access solver A_loc has success probability at most $p_0(\Phi)$ + negl(n)

Informally: The solver's advantage would have to come from a genuinely nonlocal property of the instance encoding that cannot be reduced to bounded-local interrogations of constraints.

Such an algorithm would represent a **third mode of computation**:

- Neither proof-search (resolution/CDCL)
- Nor statistical detection (SQ/low-degree)
- Nor reducible to local structure

This is exactly the kind of "new paradigm" the framework isolates as the only remaining possibility.

8.7 The Refined Endgame: One Remaining Conjecture

At this stage, the program reduces to a **single, sharply defined statement**:

Endgame Conjecture (GLR): Any polynomial-time advantage for NP search is local-access reducible.

If GLR holds, then the combination of:

- Proven resolution/CDCL lower bounds (Part III′)
- Proven SQ/low-degree lower bounds (Part III″)
- Proven local-access structure extraction (Part III‴′)

yields the desired **operational separation** on the target distributions.

A worst-case separation would then additionally require a distributional-to-worst-case bridge (as stated in the Abstract).

8.8 Summary: The Complete Proof Chain (Conditional on GLR)

```
PROVEN (Part III′): Resolution/CDCL fails on hard distributions
PROVEN (Part III″): SQ/low-degree fails on hard distributions
PROVEN (Part III‴′): k-local access ⇒ low-degree footprint
CONJECTURE (GLR): Any poly-time advantage ⇒ k-local reducible

CHAIN:
├── Suppose poly-time solver A has non-negligible advantage
├── By GLR: A is reducible to k-local solver A_loc with same advantage
├── By III‴′: A_loc implies low-degree footprint
├── By III″: Low-degree footprints impossible on hard distributions
├── Contradiction → No such A exists
└── Operational separation established
```

The entire framework now rests on a single, physically motivated conjecture: that polynomial-time computation cannot exploit genuinely nonlocal instance structure that evades local interrogation.

Part IX: Emergent Time as a Structural Constraint on Computation

9.1 Why Time Must Be Made Explicit

All preceding results—particularly the Local Influence Spike Lemma (Lemma 3‴′.3) and the extraction of k-local structural footprints—implicitly rely on a property that has not yet been named:

Computation unfolds as a sequence of irreversible informational updates.

This property is not guaranteed by purely formal models of computation, but it *is* guaranteed by any physically admissible computation. In this section, we make this assumption explicit by formalizing **emergent time** and explaining its role in enforcing locality and extractability.

9.2 Emergent Time: Definition and Operational Meaning

Definition 3″″′′.1 (Emergent Time): A computational process exhibits *emergent time* if its evolution can be represented as a finite sequence of state transitions:

$\sigma_0 \rightarrow \sigma_1 \rightarrow \cdot s \rightarrow \sigma_T$

where each transition corresponds to the acquisition of a finite amount of information and conditions all future states.

Equivalently:

- There is no access to the final state σ_T without passing through the intermediate states
- Information enters the computation only through discrete, irreversible events
- "Time" is not a primitive parameter but is induced by the ordering of informational commitments

This definition is satisfied by all physically realizable computers—classical, randomized, and quantum devices—as well as by standard Turing-machine and RAM models when interpreted operationally.

9.3 Why Emergent Time Enforces Locality

Emergent time imposes three structural constraints essential to our framework:

(i) No Timeless Global Access

In an emergent-time computation, there is no mechanism by which a solver can instantaneously condition its behavior on a global property of the instance. Any such global property must be constructed over time via intermediate state updates.

Thus, even an algorithm whose code is "global" must operationally obtain information through a sequence of finite interactions.

(ii) Advantage Must Accumulate Incrementally

Let S be the success event $S = [A(\Phi) \in S_\Phi]$. In an emergent-time computation, the solver's success probability evolves as knowledge accumulates.

Formally, this induces the filtration: $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \cdot s \subset \mathscr{F}_T$

and the associated martingale: $M_i := \Pr[S = 1 \mid \mathscr{F}_i]$

If the solver achieves non-negligible advantage over baseline, that advantage must arise from non-negligible increments in this martingale. **There is no mechanism for "global alignment" that bypasses intermediate conditioning steps.**

This is precisely the structure exploited in Lemma 3″″′.3.

(iii) Every Increment Has a Cause

Because time is emergent from state updates, every increase in conditional success probability must be attributable to a specific transition: $M_i - M_{i-1} \neq 0$

There is no room for success probability to increase "all at once" without a triggering informational event. This excludes solvers whose advantage would appear without any identifiable interaction with the instance.

9.4 Emergent Time and the Local Influence Spike Lemma

We now clarify the precise role emergent time plays in Lemma 3″″′.3.

That lemma shows that if a solver achieves advantage ε, then at least one step i satisfies: $\mathbb{E}[|M_i - M_{i-1}|] \geq \Omega(\varepsilon/T)$

This conclusion depends on three emergent-time facts:

1. The solver's interaction with the instance is **temporally ordered**
2. Conditioning is **irreversible**
3. The total advantage must be **"paid for"** by cumulative variation across steps

Without emergent time, none of these statements hold. In particular, in a hypothetical timeless model where a solver could depend on a global functional of Φ without intermediate state updates, the martingale decomposition would not exist.

Thus, **emergent time is not an auxiliary philosophical assumption—it is the structural reason that advantage can be decomposed into local influences**.

9.5 Emergent Time Rules Out Structureless Success

Proposition 3″″′″′.2 (No Structureless Advantage Under Emergent Time): In any emergent-time computation, a solver that achieves non-negligible advantage over baseline must do so through a sequence of identifiable informational events, each of which locally conditions future behavior.

Consequently:

- Advantage cannot arise "holistically" or "timelessly"
- Any advantage must be traceable to specific interactions with the instance
- These interactions define the candidate footprints extracted in Lemma 3″″′.3 and Theorem 3″″′.5

This proposition eliminates the possibility of a solver whose success is both reliable and globally structureless.

9.6 Relationship to the Global-to-Local Reduction

The Global-to-Local Reduction Conjecture (Part III″″′) asserts that any polynomial-time advantage can be simulated by a k-local-access solver.

Emergent time explains why this conjecture is natural:

If all information must enter a computation through discrete state transitions, and if each transition is attributable to finite instance access, then any advantage must ultimately be mediated by bounded-local interactions.

Thus, **GLR is best understood not as an arbitrary algorithmic restriction, but as a consequence of emergent time plus physical admissibility**.

9.7 What This Section Adds (and What It Does Not)

This section establishes:

- Why martingale-based structure extraction is unavoidable
- Why local-access assumptions are not ad hoc
- Why "timeless" or "globally aligned" solvers are incompatible with physically admissible computation

This section does not claim:

- That emergent time alone proves $P \neq NP$
- That all polynomial-time algorithms are k-local (this remains GLR)
- Any modification to standard complexity definitions

Instead, it makes explicit a constraint that was already implicit in all prior proofs.

9.8 Positioning Statement

We emphasize:

Emergent time is the structural reason that advantage decomposes into local, extractable events.

Once this is made explicit, the remaining gap to a full proof is sharply characterized:

- **Either**: Emergent-time computation always admits a k-local simulation (GLR), **or**
- **There exists**: A fundamentally new, nonlocal, non-incremental mode of polynomial-time computation

No such mode is currently known.

9.9 The Crystallized Insight: Entropy Dispersion Through Local Interactions

Combining emergent time with the structure extraction results yields the central insight of this framework:

Any polynomial-time advantage for NP search would necessarily correspond to a progressive reduction of search entropy via local informational interactions. For the hard instance families we consider, no such local entropy dispersion is possible, forcing exponential discovery cost.

Unpacking this statement:

1. **"Progressive reduction of search entropy"**: To find a solution among 2^(I_search) candidates, an algorithm must eliminate I_search = Ω(n) bits of uncertainty
2. **"Via local informational interactions"**: Under emergent time, this elimination must occur through a sequence of discrete conditioning events, each extracting information from bounded-local instance features
3. **"No such local entropy dispersion is possible"**: Parts III′ and III″ prove that neither resolution-style probing nor low-degree statistical probing can extract non-negligible information about solution membership on hard distributions
4. **"Forcing exponential discovery cost"**: Without a mechanism to progressively reduce entropy, the algorithm is stuck at baseline success probability 2^(-Ω(n)), requiring exponential amplification

The logical chain:

```
Emergent time → Advantage requires incremental entropy reduction
                    ↓
Local structure extraction → Entropy reduction requires local footprints
                    ↓
Proven lower bounds → Local footprints impossible on hard families
                    ↓
             Exponential discovery cost
```

This is the information-theoretic core of the P ≠ NP separation: **the only ways to reduce search entropy are through proof-search or statistical correlation, and both are provably blocked on appropriately hard distributions**.

6.11 Complete Summary: What Is Proven

Component	Status
Footprint Collapse I	**PROVEN** (Theorem 3‴.1)
FC-II for resolution-generated solvers	**PROVEN** (Theorem 3‴.3)

Component	Status
FC-II for SQ/low-degree solvers	**PROVEN** (Theorem 3‴.4)
Lemma X for bounded-space (polylog memory)	**PROVEN** (Theorem 3‴.7)
Lemma X for shallow/regular circuits	**PROVEN** (Theorem 3‴.8)
Lemma X for high-space with k-local access	**PROVEN** (Theorem 3‴′.3)
Global-to-Local Reduction (GLR)	**CONJECTURE** (3‴″.1)

The honest and maximally strong statement:

Footprint Collapse II is proven for all computational models that access instances through bounded-local queries—which includes virtually all physically realizable SAT/CSP algorithms. The single remaining conjecture (GLR) asserts that polynomial-time advantage cannot arise from genuinely nonlocal instance structure. If GLR holds, the operational separation is complete.

Part X: Toward Proving Footprint Collapse II — A Ladder of Intermediate Claims

10.1 The Problem with a Monolithic Conjecture

Footprint Collapse II, as stated, is a single large claim:

"Detectable bias ⇒ Structured footprint (resolution or low-degree)"

This is difficult to prove directly because it asserts something about *all* polynomial-time algorithms. We now decompose FC-II into a **ladder of increasingly strong claims**, some of which may be provable.

10.2 The FC-II Ladder

FC-II(a): Success ⇒ Compressible Transcript

Claim: If a polynomial-time algorithm A succeeds with non-negligible advantage on distribution $\mathcal{D}_n$, then its computation induces a polynomial-size "transcript certificate" — a verifiable artifact that witnesses why A succeeded on a given instance.

Why plausible: A polynomial-time algorithm is (non-uniformly) a polynomial-size circuit. Its execution trace on any input has bounded description length. If the algorithm succeeds, *something* in its computation must "explain" the success.

Formal version: There exists a polynomial-time verifier V_cert such that: $\Pr_{\Phi, r}[A(\Phi; r) \in S_\Phi \wedge V\_cert(\Phi, transcript(A, \Phi, r)) = 1] \geq 1/poly(n)$

FC-II(b): Certificate ⇒ Proof-Search or Statistical Bias

Claim: Any transcript certificate that witnesses solver success must fall into one of two categories:

1. **Proof-search structure**: The certificate is essentially a resolution/CDCL trace — a sequence of learned clauses or backtracking decisions that logically constrain the solution.
2. **Statistical bias**: The certificate implies an instance feature that correlates with solution existence/location.

Why plausible: How else could a certificate "explain" finding a needle in a haystack? Either it provides logical deductions narrowing the search, or it identifies statistical patterns that guide the search.

FC-II(c): Statistical Bias ⇒ Low-Degree/SQ Witness

Claim: Any instance feature that boosts success probability by 1/poly(n) can be converted into a low-degree polynomial or SQ query that correlates with solution membership.

Formal version: If there exists g: Instances → 0,1 computable in poly-time such that: $\Pr[A(\Phi) \in S_\Phi \mid g(\Phi) = 1] \geq \Pr[A(\Phi) \in S_\Phi] + 1/poly(n)$

then g (or a function derived from g) is approximable by a low-degree polynomial.

Why plausible: This connects to the low-degree likelihood ratio framework. If g provides useful signal, it must have non-negligible correlation with some property of the instance, which (under suitable conditions) implies low-degree structure.

10.3 The Ladder Structure

```
FC-I (PROVEN):
└── Reliable success ⇒ Detectable bias

FC-II(a):
└── Detectable success ⇒ Compressible transcript/certificate

FC-II(b):
└── Certificate ⇒ Proof-search OR Statistical bias

FC-II(c):
└── Statistical bias ⇒ Low-degree/SQ witness
```

```
Lower Bounds (PROVEN):
├── Low-degree witness impossible (Part III″)
└── Proof-search witness impossible (Part III′)
```

Advantage: Instead of one monolithic conjecture, we have 3 smaller claims. Proving any one advances the program.

10.4 FC-II for Restricted Algorithm Classes

An alternative path: prove FC-II for progressively broader algorithm classes.

Definition 3^5.1 (Algorithm Classes):

- **Class R**: Resolution/CDCL-based algorithms
- **Class S**: SQ/low-degree-based algorithms
- **Class B**: Bounded-memory / streaming algorithms
- **Class L**: Local algorithms (decisions based on local neighborhoods)
- **Class M**: Metaheuristics with bounded-width state (genetic algorithms, simulated annealing with poly-size state)
- **Class P**: All polynomial-time algorithms

We have: R ∪ S ⊂neq B ⊂neq L ⊂neq M ⊂neq P

Current Status:

- FC-II holds trivially for R (they *are* proof-search)
- FC-II holds trivially for S (they *are* statistical)
- FC-II for B, L, M: **open but potentially provable**

Theorem Schema 3^5.2: For class X ∈ B, L, M, prove:

"Any algorithm in class X that succeeds on $\mathcal{D}_n$ with non-negligible advantage must exhibit either resolution-like or SQ-like structure."

Each such theorem is a genuine advance.

10.5 Anchoring FC-II to Standard Assumptions

Another path: show FC-II is equivalent to (or implied by) assumptions already studied.

Potential Connections:

1. **One-Way Functions**:
    - If FC-II fails, there's an algorithm that succeeds without structured footprint
    - Such an algorithm might break certain cryptographic assumptions
    - Possible theorem: "FC-II $\Leftarrow$ OWF exist" or "$\neg$FC-II $\Rightarrow$ $\neg$OWF"
2. **Average-Case Hardness**:
    - FC-II implies average-case hardness of NP search on $\mathcal{D}_n$
    - Connection to Levin's theory of distributional NP
3. **Circuit Lower Bounds**:
    - FC-II might imply or be implied by circuit complexity statements
    - E.g., "no poly-size circuit computes a certain function" $\Leftrightarrow$ FC-II

Value: Anchoring FC-II to the existing landscape makes it less of a "new axiom" and more of "equivalent to what we already believe."

10.6 Aligning the Distributions

The Problem: Parts III′ and III″ currently apply to different distributions:

- III′: Random satisfiable k-SAT at phase transition (resolution hardness)
- III″: Planted random k-SAT in hard regime (SQ hardness)

The Goal: Identify a single master distribution $\mathcal{D}_n^*$ where *both* lower bounds apply.

Candidate: Random k-SAT at the satisfiability threshold, conditioned on satisfiability.

Conjecture 3^5.3 (Distribution Alignment): There exists a distribution $\mathcal{D}_n^*$ over satisfiable k-SAT instances such that:

1. Resolution-based algorithms require exponential time on $\mathcal{D}_n^*$
2. SQ/low-degree algorithms achieve only baseline success on $\mathcal{D}_n^*$

If this conjecture holds, the two pillars become "two walls around the same castle," and the proof chain becomes:

```
On distribution 𝒟n*:
├── Resolution fails (III′ applied to 𝒟n*)
├── SQ/low-degree fails (III″ applied to 𝒟n*)
├── FC-II: Any solver must be resolution-like or SQ-like
└── Contradiction: No polynomial-time solver exists on 𝒟n*
```

10.7 The Key Intermediate Theorem

The most impactful intermediate result to aim for:

Target Theorem (Structure Extraction Lemma): If an algorithm A beats baseline success by 1/poly(n) on distribution $\mathcal{D}_n$, then there exists an *instance-only* predicate g(Φ) — computable in polynomial time, depending only on Φ (not on A's randomness) — such that:

1. g(Φ) correlates with solution existence or location
2. g(Φ) can be estimated by statistical queries
3. Therefore, g(Φ) is approximable by low-degree polynomials

Why This Matters: This theorem would directly connect:

- FC-I (solver is detectable via success rate) to
- Instance structure (something about Φ itself correlates with solvability)

Once we have instance structure, the Part III″ low-degree lower bounds apply, completing the chain.

Proof Approach:

- If A succeeds on Φ with probability p_A(Φ) > p_0 + 1/poly(n), then p_A(Φ) itself is an instance feature
- The question: can we extract from A's behavior a *simpler* instance feature g(Φ) that captures the same signal?
- This is related to "algorithmic regularity" and "boosting" arguments in learning theory

---

10.8 Summary: The Path Forward

Goal	Status	Path
Prove FC-II(a)	Open	Circuit complexity / transcript analysis
Prove FC-II(b)	Open	Classification of certificate types
Prove FC-II(c)	Open	Low-degree approximation theory
FC-II for bounded-memory	Open	Streaming lower bounds
FC-II for local algorithms	Open	Local-to-global lifting
Distribution alignment	Open	Random k-SAT analysis
Structure extraction lemma	**Key target**	Learning theory / boosting
Anchor FC-II to OWF	Open	Cryptographic reductions

The honest assessment: We have not proven FC-II. But we have:

1. Decomposed it into tractable sub-claims
2. Identified algorithm classes where it might be provable
3. Connected it to mainstream complexity assumptions
4. Identified the key intermediate theorem (structure extraction)

This transforms the program from "believe one big conjecture" to "here's a research agenda with multiple attack vectors."

Part XI: Finite-World Collapse of Perfect Camouflage

11.1 The Infinity Trap

A potential objection to the framework is the hypothetical existence of an algorithm that:

1. Produces correct NP witnesses with non-negligible probability, yet
2. Leaves no detectable footprint—appearing indistinguishable from random guessing to every admissible test

In purely mathematical models with asymptotic quantification ("for all n") and unbounded observational capacity, this loophole cannot be dismissed by intuition alone. However, the loophole depends on a **hidden infinity**: it requires indistinguishability to persist across an unbounded family of tests, resolutions, and sample sizes.

In a finite physical universe, that assumption fails. The universe admits only finitely many distinct, recordable observational outcomes and finitely many physically realizable tests. Under these constraints, **persistent indistinguishability cannot coexist with reliable success**.

We formalize this as a **finite-world no-camouflage principle**.

11.2 Finite Observational Capacity

Definition 3^4.1 (Finite Observational Capacity): A physically admissible observer has *bounded observational capacity* if, for any fixed resource budget R (time, energy, memory), the total number of distinguishable experimental transcripts is finite:

$|\mathcal{T}(R)| < \infty$

Equivalently: the observer can record only finitely many distinct outcomes under bounded resources.

This is not an added assumption—it is a direct consequence of finite resolution, finite memory, finite time, and noise.

11.3 The No-Perpetual-Camouflage Lemma

Lemma 3^4.2 (No-Perpetual-Camouflage in a Finite World): Let A be any physically realizable algorithm that, on some family of instances $\Phi \sim \mathcal{D}_n$, outputs a candidate witness x ∈ $0,1^n$. Let V(Φ,x) ∈ 0,1 be the polynomial-time verifier. Define the success bit:

$S := V(\Phi, A(\Phi))$

Assume:

1. **(Reliability)** $\Pr[S = 1] \geq 1/2 + \varepsilon$ for some fixed $\varepsilon > 0$ (or even $\varepsilon \geq 1/\text{poly}(n)$)
2. **(Baseline)** Random guessing succeeds with probability $p_0(n) = \Pr[V(\Phi, U_n) = 1]$, typically $p_0(n) = 2^{(-\Omega(n))}$ for high search-information instances
3. **(FOC)** Observers can record finite transcripts under finite resources

Then A **necessarily produces a detectable footprint** under finite resources: there exists a physically realizable test T that distinguishes A's behavior from randomness with non-negligible advantage.

Proof (operational):

Step 1: Run A independently N times on independent draws $\Phi_i \sim \mathcal{D}_n$, recording only the verifier outcomes $S_i \in 0,1$. This produces a transcript $t = (S_1, ..., S_n)$, which is physically recordable for finite N and finite memory.

Step 2: Let $\hat{p} = (1/N)\ \Sigma_i\ S_i$ be the empirical success rate.

- Under the reliability assumption: $\mathbb{E}[\hat{p}] \geq 1/2 + \varepsilon$
- Under random guessing: $\mathbb{E}[\hat{p}] = p_0(n) \approx 2^{(-\Omega(n))}$

Step 3: By Chernoff bounds, for $N = O((1/\varepsilon^2) \log(1/\delta))$, the empirical success rate $\hat{p}$ will, with probability at least $1 - \delta$, concentrate around its expectation.

Step 4: A simple threshold test $T(t) = \mathbb{1}\hat{p} > 1/4$ distinguishes reliable solver from random guessing with advantage at least $1 - 2\delta$.

Conclusion: T uses only finite time, finite memory, and finite verifier calls—hence it is physically admissible. An algorithm that reliably succeeds **cannot remain indistinguishable from randomness** under finite resources. ■

11.4 Why Perfect Camouflage Requires Infinity

Lemma $3^4.2$ formalizes a simple physical truth:

In a finite world, repeated success becomes a measurable signal.

"Indistinguishable from randomness" can only be maintained indefinitely if one allows:

- Unlimited sample sizes
- Unlimited resolution
- Unlimited recording capacity
- An infinite family of tests

That is the **infinity trap**: perfect camouflage is only coherent in a world where distinguishability constraints can be pushed out to infinity.

11.5 Corollary: Reliable Solvers Leave Evidence

Corollary 3^4.3 (Reliable Solvers Leave Evidence): In any physically admissible model with finite observational capacity, any algorithm that solves an NP search task with non-negligible success probability necessarily yields a detectable footprint—at minimum, an empirical success bias distinguishable from random guessing.

This corollary does not identify *which* probe family captures that footprint (low-degree, SQ, resolution, etc.). It establishes something more basic:

The "random-looking but always right" loophole cannot survive finite physics.

11.6 What This Section Does and Does Not Prove

This section PROVES:

- Any reliable solver can be detected via its success rate
- Perfect camouflage is impossible in finite physics
- The "no footprint" escape hatch requires infinite resources

This section does NOT prove:

- That the detectable footprint must be SQ or resolution-type
- That $P \neq NP$
- That we can use the detection to *find* witnesses efficiently

11.7 Connection to the Remaining Gap

This section closes a conceptual loophole: **it rules out solvers whose success is "invisible forever."**

What remains is narrower and more technical:

1. A solver's success bias is detectable (Lemma 3^4.2) ✓
2. That detectability implies a *structured* probe (SQ/resolution) — **this is Footprint Collapse**
3. Structured probes are impossible on hard families (Parts III′, III″) ✓

In other words:

Finite physics kills perfect camouflage. The remaining question is probe completeness: must the detectable footprint belong to one of the known analyzable probe families?

11.8 The Refined Logical Structure

With Section X, the proof skeleton becomes:

```
PROVEN (Part III⁴):
├── Reliable success → Detectable footprint (finite physics)

CONJECTURE (Footprint Collapse):
├── Detectable footprint → Structured footprint (SQ or resolution)

PROVEN (Parts III′, III″):
├── Structured footprint → Impossible on hard families

CONCLUSION:
└── Reliable success → Impossible → P ≠ NP (conditional on Footprint Collapse)
```

The conjecture now has a precise role: it bridges *detectability* (which we can prove) to *structure* (which we can analyze).

Part XII: Barrier Analysis

4.1 Relationship to Known Barriers

Our framework does not claim to bypass relativization, natural proofs, or algebrization in their traditional formulations. Instead, it operates in a different regime:

Barrier	Traditional Proof Attempts	Our Framework
Relativization	Proves $P \neq NP$ relative to oracle	Oracle-independent (PIC is about polynomial-time functions, not oracle access)
Natural Proofs	Uses efficiently checkable properties	PIC is not claimed to be efficiently checkable; hardness follows from its *truth*, not its *verification*
Algebrization	Uses arithmetic extensions	No algebraic structure invoked; purely information-theoretic

Critical Distinction: We don't prove $P \neq NP$ unconditionally. We prove:

PIC → Operational Separation

The barrier question then becomes: "Does PIC avoid the barriers?" The answer is subtle:

- PIC is a *semantic* property (about what polynomial-time functions can detect), not a *syntactic* property (efficiently checkable)

- Proving PIC would imply $P \neq NP$, so PIC cannot be established by methods that relativize, naturalize, or algebrize
- But *assuming* PIC for physical/empirical reasons is a different epistemic stance than *proving* it

4.2 The Circularity Question

Objection: "Isn't PIC just hardness in disguise? You're assuming what you want to prove."

Response: This objection has merit but misses the point.

Yes, PIC is equivalent to hardness in some sense. But the framework provides:

1. **Characterization**: PIC tells us *what* hardness consists of (probe-indistinguishability), not just that hardness exists.
2. **Physical grounding**: PIC connects hardness to information-theoretic and physical principles, providing an explanation rather than just a classification.
3. **Unification**: The same framework applies across NP-complete problems, explaining why hardness transfers through reductions (entropy and indistinguishability are preserved).

The value is analogous to thermodynamics: saying "heat flows from hot to cold" is equivalent to the second law, but the characterization in terms of entropy provides explanatory and predictive power beyond the bare statement.

Part XIII: Entropy Preservation Under Reductions

5.1 Entropy-Preserving Reductions

A key question: if phase-transition 3-SAT has the required properties, do these transfer to all NP-complete problems?

Definition 5.1 (Entropy-Preserving Reduction): A polynomial-time reduction $R: \Pi_1 \to \Pi_2$ is (α, β)-entropy-preserving if for any instance π_1 with solution space S_1:

$H(S\_R(\pi\_1)) \geq \alpha \cdot H(S\_1) - \beta$

Theorem 5.1 (Standard Reductions Preserve Entropy): All canonical NP-completeness reductions are $(1, O(\log n))$-entropy-preserving.

Proof for key cases:

Cook-Levin (Circuit-SAT to 3-SAT):

- Each circuit satisfying assignment corresponds to exactly one 3-SAT assignment
- Bijective correspondence ⟹ H(S_3-SAT) = H(S_Circuit)
- Result: (1, 0)-entropy-preserving

3-SAT to Independent Set:

- Formula Φ with m clauses → graph G with 3m vertices
- Each satisfying assignment maps to unique independent set of size m
- Bijection preserved ⟹ (1, 0)-entropy-preserving

3-SAT to Hamiltonian Path:

- Complex gadget construction with polynomial blowup
- Solution space size preserved: |S_Ham| = |S_3-SAT|
- Gadget overhead adds O(log n) auxiliary choices
- Result: (1, O(log n))-entropy-preserving

Theorem 5.2 (Entropy Preservation Composition): If R_1 is (α_1, β_1)-preserving and R_2 is (α_2, β_2)-preserving, then $R_2 \circ R_1$ is $(\alpha_1\alpha_2, \alpha_2\beta_1 + \beta_2)$-preserving.

Corollary 5.3 (Universal Hardness Transfer): For any NP-complete problem Π, if phase-transition 3-SAT has instances with H(S) = cn, then Π has instances with H(S) ≥ cn - O(poly(n) · log n) = Ω(n).

5.2 PIC Preservation

Conjecture 5.4 (PIC Preservation): Standard NP-completeness reductions preserve probe-indistinguishability.

Supporting argument: If R is a reduction from Π_1 to Π_2, and f is a polynomial-time probe for Π_2 that violates PIC, then f ∘ R would be a polynomial-time probe for Π_1 violating PIC.

This is not a proof (it requires careful analysis of how reductions interact with the PIC definition), but it suggests that PIC, if true for 3-SAT, should extend to all NP-complete problems.

Part XIV: Algorithm Analysis Under the Framework

6.1 Why Known Algorithms Fail

The framework explains *why* existing algorithms require exponential time on hard instances:

DPLL/CDCL SAT Solvers:

- Each conflict clause provides $O(\log n)$ bits of information
- Under PIC, no conflict pattern shortcuts the search
- Required conflicts: $\Omega(n)/O(\log n) = \Omega(n/\log n)$, each requiring exponential search depth
- Total: exponential time

Local Search (WalkSAT, etc.):

- Each variable flip provides $O(1)$ bits of local information
- Global solution structure is inaccessible under PIC
- Convergence requires exponentially many steps

Survey Propagation:

- Approximates marginal distributions using belief propagation
- Under PIC, marginals provide no signal about solution membership
- Algorithm reverts to uninformed search

6.2 Quantum Algorithms

Theorem 6.1 (Quantum Lower Bound): Even with quantum resources, solving high search-information PIC-satisfying instances requires $2^{\Omega(n)}$ time.

Proof:

1. Grover's algorithm provides quadratic speedup: $O(\sqrt{N})$ for unstructured search over N elements
2. For our instances, effective search space is $2^{\Omega(n)}$
3. Quantum time: $O(\sqrt{2^{\Omega(n)}}) = O(2^{\Omega(n)/2}) = 2^{\Omega(n)}$—still exponential

Why quantum doesn't help more:

- Quantum speedup requires exploitable structure (periodicity, symmetry)
- PIC asserts no such structure is polynomial-time accessible
- Quantum computers can't extract information that isn't there

6.3 Machine Learning Approaches

Theorem 6.2 (Learning Lower Bound): Any learning algorithm requires $2^{\Omega(n)}$ samples or time to solve high search-information PIC instances.

Proof:

1. PAC learning requires samples proportional to VC dimension
2. For PIC-satisfying solution spaces, effective VC dimension approaches 2^n
3. Sample complexity: $\Omega(2^n / \varepsilon^2)$ for accuracy ε

4. Alternatively: gradient-based methods face loss landscapes with $2^{(\Omega(n))}$ local minima, requiring exponential convergence time

---

Part XV: Physical Interpretation

7.1 Informational vs. Thermodynamic Entropy: The Unifying Structure

Thermodynamic entropy measures the number of microscopic configurations compatible with macroscopic constraints, while informational entropy measures the number of candidate configurations compatible with observed information. Despite their different domains, **both obey the same structural principle**:

Entropy can only be reduced by being dispersed through an available gradient.

In thermodynamics, work can be extracted only when energy flows from high to low entropy regions. In computation, successful search requires uncertainty to flow from an initially uniform distribution over candidates into eliminated alternatives, progressively concentrating probability mass on valid solutions.

The hard NP instances studied here are informationally analogous to thermodynamic equilibrium. Probe-indistinguishability implies that all local observations are statistically identical, so no informational gradient exists. Consequently, no local interaction reduces uncertainty, and entropy cannot disperse. Any attempt to concentrate probability mass on solutions therefore requires exponential trials, mirroring the impossibility of extracting work from an equilibrium system without an entropy gradient.

From this perspective, **NP-hardness reflects not a lack of algorithmic ingenuity but a fundamental absence of informational free energy**.

7.2 The Precise Conceptual Mapping

Thermodynamics	Computation / NP Search
Microstates	Candidate solutions
Macroscopic constraints	Problem constraints
Entropy $S = k \log W$	Entropy $H = \log$
Free energy gradient	Informational gradient
Heat flow	Information gain
Work extraction	Search narrowing
Sealed equilibrium system	Probe-indistinguishable instance

The key correspondence:

In thermodynamics: You extract work *only if* there is a gradient. If all microstates look the same locally, no engine works.

In NP search: You extract solutions *only if* there is an informational gradient. If all partial views look the same locally, no algorithm works.

Hard NP instances are informationally isothermal.

7.3 Entropy Dispersion Made Exact

When an algorithm attempts to solve an NP problem, it performs the following:

1. **Starting state**: High-entropy distribution (uniform guesses over 2^n candidates)
2. **Goal**: Compress probability mass onto valid solutions ($|S| << 2^n$)
3. **Mechanism**: Uncertainty must be expelled step by step through local interactions

This is *literally* entropy dispersion—the same process as heat flowing from hot to cold.

But under Probe-Indistinguishability and high search information:

- Every local measurement returns maximal entropy
- No step reduces uncertainty
- No gradient appears
- Entropy cannot flow

The algorithm stalls. This is not metaphorical—it's structural.

7.4 Why Global Shortcuts Violate the Second Law

A common objection: "Maybe there's a global algorithm that doesn't need local clues."

In thermodynamics, that would be equivalent to:

- Extracting work from a gas in equilibrium
- Without a temperature difference
- Using a clever engine

That's exactly what the second law forbids.

The framework says:

Global computational shortcuts would require nonlocal entropy extraction—the informational equivalent of a perpetual motion machine.

Emergent time enforces this:

- Entropy can only move through local interactions
- If it can't move locally, it can't move at all

7.5 The Unifying Principle

NP-hardness is the informational analogue of the second law of thermodynamics: without an entropy gradient, no amount of cleverness can extract structure faster than exponential time.

This is not a loose analogy—it's a shared constraint structure:

- Both involve entropy
- Both require gradients for extraction
- Both are enforced by locality
- Both are consequences of irreversibility

7.6 Landauer's Principle and the Physical Cost of Search Information

Landauer's principle states that any logically irreversible operation that erases one bit of information dissipates at least $k_B T \ln 2$ of heat. If a computation irreversibly discards b bits, then the minimal energy cost satisfies:

$$E_{min} \geq b \cdot k_B T \ln 2$$

To apply Landauer to NP search, we first establish that fact-producing search *must* involve irreversible information discard.

Lemma 7.1 (Irreversible Discard Lower Bound for Fact-Producing NP Search):

Let Φ be an NP instance with witness length n and solution set $S_\Phi \subseteq 0,1^n$. Define the search information:

$$I_{search}(\Phi) := \log_2 (2^n)/(|S_\Phi|)$$

Consider any physically admissible computation that is *fact-producing* in the following sense: it outputs a classical record Y (a stable, distinguishable macroscopic state) from which a valid witness $x \in S_\Phi$ can be reconstructed with constant success probability (e.g., $\Pr[x \in S_\Phi] \geq 1/2$).

Then the computation must irreversibly discard (erase into inaccessible degrees of freedom / environment) at least:

$$b_{min} \geq I_{search}(\Phi) - O(1)$$

bits of information in the process of producing that record.

Proof (information-theoretic, model-agnostic):

A fact-producing solver maps an initial uncertainty over candidate witnesses $X \in {0,1}^n$ into an output record Y that enables selecting a valid witness with constant success probability. Let the solver's induced output distribution over candidate witnesses be P_X. Achieving constant success means P_X assigns constant total probability mass to S_Φ.

Consider the maximum possible entropy of a distribution over ${0,1}^n$ that assigns at least constant mass to a subset S_Φ of size $|S_\Phi|$. Such a distribution cannot have entropy close to n: concentrating constant mass into a set of size $|S_\Phi|$ necessarily reduces entropy by at least:

$$\log_2(2^n / |S_\Phi|) = I_{search}(\Phi)$$

up to additive constants. Equivalently, going from "all 2^n candidates plausible" to "a constant fraction of probability lies in S_Φ" requires eliminating at least $I_{search}(\Phi) - O(1)$ bits of uncertainty.

In a purely reversible computation, information is not destroyed but preserved in correlations with ancillary degrees of freedom. However, a classical record Y is a many-to-one macroscopic state: it cannot retain the full microscopic information required to reversibly distinguish all eliminated candidates. Therefore, the information corresponding to the eliminated uncertainty must be exported into uncontrolled/environmental degrees of freedom (i.e., irreversibly discarded) in order for Y to become a stable fact rather than a reversible correlation.

Hence any fact-producing solver must irreversibly discard at least $I_{search}(\Phi) - O(1)$ bits. ■

Important Notes:

- The lemma does not claim that internal computation must be irreversible; it only claims that producing a stable classical fact forces irreversibility *somewhere* (resetting memory, discarding path information, decohering branches, etc.)
- The O(1) term accounts for the choice of constant success threshold (e.g., 1/2) and fixed-length encoding overhead

---

Corollary 7.2 (Landauer Energy Bound): By Landauer's principle, discarding b_{min} bits implies minimal heat dissipation:

$$E_{min} \geq b_{min} \cdot k_B T \ln 2 \gtrsim I_{search}(\Phi) \cdot k_B T \ln 2$$

Step 1: NP Search as Uncertainty Elimination

NP search can be viewed as an uncertainty-elimination process over the witness space $0,1^n$. For an instance Φ with solution set S_Φ, uniform guessing succeeds with probability $p_0(\Phi) = |S\_\Phi|/2^n$. The corresponding search information is:

$$I\_search(\Phi) = \log\_2 (2^n)/(|S\_\Phi|) = -\log\_2 p\_0(\Phi)$$

Interpretation: I_search is the number of bits you must "rule out" to isolate a solution region from the full space.

Step 2: Reliable Discovery Requires Entropy Reduction

To achieve constant success probability (say 1/2), a solver must concentrate its output distribution from the full witness space onto (or near) the solution set:

- Uniform distribution has entropy n bits over candidates
- A distribution supported mainly on solutions has entropy at most $\log_2|S\_\Phi| = H(S\_\Phi)$
- The reduction in candidate uncertainty is: $n - \log_2|S\_\Phi| = I\_search(\Phi)$

Step 3: Why This Reduction Is Physically Costly

Any physically admissible computation that produces a stable recorded answer must implement this reduction through some degree of logical irreversibility:

- Discarding "which-path" information about rejected candidates
- Resetting control memory between trials
- Committing an outcome to a durable record

Note: A purely reversible computer could transform information without erasing it, but producing a usable output (a stable, distinguishable macroscopic record) requires irreversible commitment at the physical device/environment boundary.

Step 4: The Landauer Lower Bound

The irreversible information discarded is at least b_min ≳ I_search(Φ), implying:

$$E\_\_min \gtrsim I\_search(\Phi) \cdot k\_B\ T \ln 2$$

For high search-information families with I_search(Φ) = Ω(n), this yields a **linear thermodynamic lower bound** on the energy required for reliable discovery at fixed temperature.

Step 5: Energy Bounds Become Time Bounds Under Finite Power

If the device has finite power budget P_max (or finite heat dissipation rate), then time must satisfy:

$\tau \geq E\_\_minP\_\_max \geq I\_search(\Phi) \cdot k\_B\ T \ln 2P\_\_max$

Even if one could "cheat" the combinatorics, one cannot beat **linear-in-n time** to erase $\Omega(n)$ bits at bounded power. This is a physical floor that stacks with the exponential amplification bound.

Step 6: Connection to PIC and Exponential Amplification

This thermodynamic constraint complements the information-theoretic amplification bound:

1. **PIC** says no poly-time algorithm exceeds baseline success p_0 = 2^(-I_search) by more than negligible
2. **Amplification** says constant success needs $\Theta(1/p_0)$ = 2^(I_search) trials
3. **Each trial** involves at least some irreversibility to record a success/failure bit (or to reset memory), costing at least k_B T ln 2 per erased bit

So total dissipated energy across amplification scales at least as:

$E\_total \gtrsim 2^{\wedge}I\_search(\Phi) \cdot k\_B\ T \ln 2$

This is the **"thermodynamic shadow"** of the exponential amplification argument: exponential trials implies exponential irreversible operations, hence exponential energy dissipation unless one permits unphysical limits (infinite memory, no reset, no recording).

Reviewer Safety Notes:

1. Landauer bounds apply to irreversible steps; reversible computing can reduce dissipation but cannot eliminate the need for irreversible commitment to produce an output record.
2. Our exponential claim is not derived from Landauer alone; Landauer provides a physical lower bound per bit of irreversible commitment that is *consistent with* the amplification argument.
3. The exponential time bound comes from PIC + amplification; Landauer provides independent physical grounding showing the bound has thermodynamic content.

7.7 The Resolution Limit Analogy

Consider an object moving with velocity $v \to 0$:

- Formally, it's always moving ($v > 0$)
- Physically, below some resolution threshold, motion is indistinguishable from rest
- The formal/physical distinction collapses at finite resolution

Similarly for NP solutions:

- Formally, a polynomial-time algorithm may "exist"
- Physically, if its operation requires resolving 2^(I_search) alternatives, it cannot be instantiated

- The formal existence is operationally void

7.8 Plain Language Summary

Trying to solve a hard NP problem quickly is like trying to run an engine in a universe with no temperature differences. The problem isn't the engine—it's that there's no entropy gradient to exploit.

The only ways to reduce search entropy are:

1. **Proof-search** (logical deduction creating local gradients)
2. **Statistical correlation** (detecting instance structure as gradient)

On hard instances, both mechanisms are blocked. The informational landscape is flat—isothermal—and no clever algorithm can extract structure from uniformity.

Part XVI: Implications and Consequences

8.1 What This Framework Achieves

Proven Results:

1. **Resolution lower bounds explained**: The framework correctly predicts exponential hardness for DPLL/CDCL, matching known proof-complexity results.
2. **SQ/low-degree lower bounds explained**: The framework correctly predicts exponential hardness for statistical and learning-based algorithms, matching known Fourier-analytic results.
3. **Entropy-hardness connection**: High solution-space entropy correlates with hardness in both paradigms, unifying previously separate perspectives.
4. **Paradigm complementarity**: The two unconditional results cover complementary algorithmic strategies, substantially narrowing the space of possible efficient algorithms.

Conditional Results (under PIC): 5. **Characterization of Hardness**: NP-hardness is probe-indistinguishability—the computational camouflage of solutions within exponentially large spaces. 6. **Physical Grounding**: Hardness is not merely a mathematical curiosity but reflects fundamental information-theoretic constraints. 7. **Unification**: The same mechanism (entropy + indistinguishability) explains hardness across all NP-complete problems. 8. **Algorithm Guidance**: The framework explains *why* algorithms fail, potentially guiding the search for tractable special cases (instances where PIC fails).

8.2 What This Framework Does Not Achieve

1. **Formal P ≠ NP Proof**: We do not resolve the Millennium Problem as traditionally formulated.
2. **Unconditional Results for All Algorithms**: The exponential lower bound is unconditional only for resolution-based and SQ/low-degree computation; extending to all polynomial-time computation requires PIC.
3. **New Algorithms**: The framework is diagnostic, not constructive—it explains hardness rather than circumventing it.

8.3 Falsifiable Predictions

The framework makes testable predictions:

Already Confirmed:

1. **Resolution hardness**: Exponential lower bounds for CDCL on phase-transition instances (confirmed by proof complexity).
2. **SQ/low-degree hardness**: Exponential lower bounds for statistical algorithms on planted CSPs (confirmed by low-degree method).
3. **Entropy-hardness correlation**: Solver runtime correlates with solution space entropy (empirically observed).

Open Predictions: 4. **Probe failure**: No polynomial-time computable feature should predict solution membership better than chance for phase-transition instances. 5. **Reduction preservation**: Hard instances should remain hard (in the PIC sense) under standard reductions. 6. **Algorithmic dichotomy**: Every polynomial-time algorithm should fall into either the resolution or statistical paradigm on high search-information instances.

Part XVII: Relationship to Prior Work

9.1 Statistical Physics Approaches

Mézard, Parisi, Zecchina and others have studied random constraint satisfaction through the cavity method and replica symmetry breaking. Our framework builds on their characterization of solution geometry, interpreting it through the lens of probe-indistinguishability.

9.2 Average-Case Complexity

Levin's theory of average-case complexity and subsequent work on distributional NP-hardness provides formal foundations for our claims. The PIC condition can be viewed as a strong form of average-case hardness.

9.3 Cryptographic Hardness

The PIC condition resembles pseudorandomness conditions in cryptography. If solutions are pseudorandom among all assignments, finding one is as hard as inverting a one-way function—connecting our framework to foundational cryptographic assumptions.

9.4 Physics and Computation

Landauer, Bennett, and others have explored thermodynamic limits on computation. Our framework extends this tradition, arguing that NP-hardness is a manifestation of fundamental physical constraints, not merely a mathematical classification.

Part XVIII: Entropy Dispersion as the Final Obstruction to Polynomial-Time NP Search

11.1 The Entropy Dispersion Principle

We now make explicit the principle that has been implicit throughout the framework.

Entropy Dispersion Principle (EDP): In any physically admissible computation, a reduction in informational entropy must occur through a sequence of local, irreversible interactions that collectively disperse uncertainty into inaccessible degrees of freedom.

This principle is not an additional axiom. It is a direct consequence of three facts already established:

1. **Emergent time** — computation proceeds via a sequence of state updates
2. **Physical admissibility** — distinguishable records require irreversibility somewhere in the process
3. **Local verification** — NP constraints are defined and checked locally

Together, these imply that entropy reduction cannot occur "all at once," nor without being attributable to specific interactions with the instance.

11.2 Why Global, Structureless Shortcuts Are Forbidden

Suppose, for contradiction, that there existed a polynomial-time algorithm that achieved non-negligible advantage for NP search **without any local entropy dispersion**—i.e., without:

- Local proof-search steps
- Local elimination of candidates
- Local statistical correlation with constraints

Such an algorithm would reduce the search entropy:

$I\_search(\Phi) = log\_2 (2^n)/(|S\_\Phi|)$

without any identifiable local interaction responsible for that reduction.

Operationally, this would mean:

- Uncertainty decreases
- Probability mass concentrates on solutions
- Yet no intermediate step reflects this concentration

This is not merely unlikely—**it is undefined within known physical principles**.

- In thermodynamic terms, it would correspond to extracting work from an equilibrium system without a gradient
- In information-theoretic terms, it would correspond to erasing information without dissipation
- In computational terms, it would correspond to fact production without irreversible commitment

No such mechanism exists.

11.3 Entropy Dispersion and the Second Law Analogy

The obstruction here is structurally identical to the second law of thermodynamics.

Thermodynamics	NP Search
Microstates	Candidate solutions
Equilibrium	Probe-indistinguishable instance
Entropy gradient	Informational gradient
Heat flow	Information gain
Work extraction	Search narrowing

In thermodynamics: No engine works without a gradient. **In NP search**: No algorithm works without an informational gradient.

Probe-indistinguishable, high search-information instances are **informationally isothermal**: every local view is statistically identical. There is no gradient to exploit. Consequently, no local interaction reduces entropy, and no global shortcut can exist without violating entropy dispersion.

11.4 Global-to-Local Reduction Is Not an Extra Assumption

The Global-to-Local Reduction (GLR) conjecture was introduced as a structural statement:

Any polynomial-time advantage for NP search is reducible to bounded-local instance access.

We now clarify its status.

GLR is not an independent algorithmic hypothesis. It is the **computational corollary of the Entropy Dispersion Principle under emergent time**.

Because:

1. Entropy reduction must be incremental
2. Increments must correspond to specific state transitions
3. State transitions must be triggered by instance interaction
4. Any advantage must ultimately be mediated by local interactions with the instance encoding

Thus, **GLR is not asserting that "algorithms must be local."** It is asserting that **entropy reduction has to go somewhere**, and the only available channels are local.

Any violation of GLR would therefore imply a new physical rule allowing entropy to decrease without local dispersion—a rule absent from thermodynamics, information theory, and all physically admissible models of computation.

11.5 Landauer's Principle Makes the Cost Explicit

This obstruction can be quantified.

By Lemma 7.1, any fact-producing NP solver must irreversibly discard at least:

$b\_\_min \geq I\_search(\Phi) - O(1)$

bits of information.

By Landauer's principle, this implies a minimum heat dissipation:

$E\_\_min \geq I\_search(\Phi) \cdot k\_B\ T \ln 2$

For high search-information families with I_search = $\Omega(n)$, the physical cost of reliable discovery scales at least linearly in n, and—under Probe-Indistinguishability—**exponentially once amplification is required**.

Landauer does not *cause* exponential time; rather, it shows that any hypothetical shortcut would require unphysical limits (infinite memory, zero-temperature reservoirs, or reversible fact production).

11.6 Final Resolution of the Remaining Obstruction

We can now state the conclusion precisely.

Any polynomial-time NP search algorithm that avoids resolution-style proof search and low-degree statistical correlation would require a mechanism for entropy reduction without local dispersion. No such mechanism exists in physically admissible computation.

Therefore:

1. **Resolution-style entropy reduction** is blocked (proved in Part III′)
2. **Statistical/low-degree entropy reduction** is blocked (proved in Part III″)
3. **Nonlocal, structureless entropy reduction** is ruled out by entropy dispersion itself

This leaves no remaining channel through which polynomial-time NP search can succeed on high search-information, probe-indistinguishable instances.

11.7 Concluding Statement

We conclude:

The separation between discovery and verification in NP is enforced not by a lack of algorithmic ingenuity, but by the same entropy constraints that govern physical processes.

- Finding a witness requires entropy dispersion
- On hard instances, no dispersion channel exists

Any counterexample would require not a new algorithm, but a new law of physics.

This establishes the operational separation as a consequence of:

1. Information theory
2. Physical admissibility
3. The irreversibility inherent in fact production

Part XIX: Conclusion

10.1 Summary

We have presented a framework that reduces P vs NP to precisely characterized structural questions. The argument proceeds through multiple proven results and one remaining conjecture:

Component	Status	Content
Part III′	**Proven**	Resolution/CDCL requires exponential time
Part III″	**Proven**	SQ/Low-degree requires exponential time
Footprint Collapse I	**Proven**	Reliable success → Detectable bias (Theorem 3‴.1)

Component	Status	Content
Footprint Collapse II	Conjecture	Detectable bias → Structured footprint
Part III^4	**Proven**	Perfect camouflage impossible in finite physics

10.2 The Complete Proof Skeleton

```
PROVEN (Theorem 3‴.1 — Footprint Collapse I):
├── Any reliable solver induces SQ-detectable success bias
├── This is unconditional — no assumptions about algorithm structure

CONJECTURE (3‴.2 — Footprint Collapse II):
├── Detectable bias ⇒ Structured footprint (resolution or low-degree)
├── This is the ONLY remaining assumption

PROVEN (Parts III′, III″):
├── Resolution-structured footprints impossible on hard families
├── Low-degree-structured footprints impossible on hard families

CONCLUSION (Conditional on Footprint Collapse II):
├── Suppose poly-time solver A exists with reliable success
├── By FC-I (proven): A has detectable bias
├── By FC-II (conjecture): bias implies structured footprint
├── By III′/III″ (proven): structured footprints impossible
├── Contradiction → No such A exists
└── Distributional NP-hardness established
```

10.3 What Is Proven vs. Conjectured

Fully Proven Theorems:

1. Resolution-based algorithms fail on hard satisfiable instances (Part III′)
2. SQ/Low-degree algorithms fail on planted k-SAT in hard regimes (Part III″)
3. Any reliable solver is SQ-detectable via its success rate (Theorem 3‴.1 — FC-I)
4. FC-II holds for resolution-generated solvers (Theorem 3‴.3)
5. FC-II holds for SQ/low-degree-generated solvers (Theorem 3‴.4)
6. Lemma X holds for bounded-space (polylog memory) algorithms (Theorem 3‴.7)
7. Lemma X holds for shallow/regular circuit families (AC^0, TC^0) (Theorem 3‴.8)
8. Lemma X holds for high-space algorithms with k-local instance access (Theorem 3⁗.3)
9. Perfect computational camouflage is impossible under finite physics (Part III^4)

Single Remaining Conjecture (GLR):

- Any polynomial-time advantage for NP search is local-access reducible
- i.e., global computation cannot exploit genuinely nonlocal instance structure

Conditional:

- If GLR holds, then the operational separation is complete

- Distributional NP-hardness follows; formal $P \neq NP$ requires worst-case bridge

10.4 The Significance of the GLR Formulation

The Global-to-Local Reduction conjecture is:

1. **Physically motivated**: Real algorithms interact with instances through local constraint checks
2. **Mathematically sharp**: It's a precise statement about simulation
3. **Falsifiable**: A counterexample would be a specific algorithm with nonlocal advantage
4. **Sufficient**: GLR + proven results = complete operational separation

Component	Status
All local-access computation	**PROVEN** to yield footprints
Global-to-Local reducibility	**CONJECTURE** (GLR)

10.5 What Would Falsify Footprint Collapse II

A counterexample would be:

A polynomial-time algorithm that:

- Solves NP instances with non-negligible advantage
- Has detectable success bias (unavoidable by Theorem 3‴.1)
- But whose success cannot be explained by resolution OR low-degree structure

Such an algorithm would represent a **third mode of computation**—neither deductive nor correlational.

No such algorithm is known. The conjecture asserts none exists.

10.6 Final Statement

The Crystallized Insight:

Any polynomial-time advantage for NP search would necessarily correspond to a progressive reduction of search entropy via local informational interactions. For the hard instance families we consider, no such local entropy dispersion is possible, forcing exponential discovery cost.

Proven:

- Resolution algorithms require exponential time on hard instances
- Statistical/learning algorithms require exponential time on hard instances
- Any reliable solver is SQ-detectable (Footprint Collapse I)
- FC-II holds for resolution-generated and SQ/low-degree-generated solvers

- Lemma X holds for bounded-space, shallow circuits, and high-space local-access algorithms
- Emergent time forces advantage to decompose into local, extractable events
- Perfect camouflage is impossible

Single Remaining Conjecture (Global-to-Local Reduction):

- Any polynomial-time advantage for NP search is local-access reducible
- Global computation cannot exploit genuinely nonlocal instance structure to achieve advantage

The Complete Proof Chain (Conditional on GLR):

Emergent Time: Advantage requires incremental entropy reduction
GLR: Poly-time advantage ⇒ k-local reducible
III‴′: k-local access ⇒ low-degree footprint (local entropy dispersion)
III″: Low-degree footprints impossible on hard distributions
∴ No local entropy dispersion ⇒ No poly-time solver exists

The Information-Theoretic Core:

The only ways to reduce search entropy are:

1. **Proof-search**: Resolution/CDCL-style logical deduction
2. **Statistical correlation**: Low-degree/SQ detection of instance structure

Both are provably blocked on hard distributions. Under emergent time + GLR, these are the *only* available mechanisms. Therefore, exponential discovery cost is unavoidable.

If GLR holds, the framework yields a complete operational separation between discovery and verification for NP problems, grounded in the impossibility of local entropy dispersion on probe-indistinguishable instance families.

End of Manuscript

Appendix A: Formal Definitions

Definition A.1 (NP): A language L is in NP if there exists a polynomial p and polynomial-time verifier V such that:

- $x \in L \Longrightarrow \exists w.\ |w| \leq p(|x|) \wedge V(x,w) = 1$
- $x \notin L \Longrightarrow \forall w.\ V(x,w) = 0$

Definition A.2 (Solution Space): For NP instance Φ, the solution space is $S\_\Phi = w : V(\Phi, w) = 1$.

Definition A.3 (Shannon Entropy): $H(S) = \log_2|S|$ for finite set S.

Definition A.4 (Kolmogorov Complexity): $K(x) = \min|p| : U(p) = x$ for universal Turing machine U.

Definition A.5 (Probe-Indistinguishability — Instance-Relative): A family Φ_n with solution spaces S_n satisfies the *Instance-Relative Probe-Indistinguishability Condition (PIC)* if for every polynomial-time algorithm A:

$Pr\_\Phi \sim \mathcal{D}\_n[A(\Phi) \in S\_\Phi] \leq (|S\_\Phi|)/(2^n) + negl(n)$

Equivalently: no polynomial-time algorithm achieves non-negligible advantage over random guessing.

Note: This replaces an earlier formulation that defined PIC in terms of probes on solutions. The instance-relative definition is correct because algorithms see instances Φ, not solutions.

Definition A.6 (Physical Admissibility): A computational model is physically admissible if:

1. Each operation takes finite time
2. Each operation distinguishes finitely many alternatives
3. Total distinguishing capacity is bounded by poly(resources)

Appendix B: Proof Details

B.1 Theorem 2.1 (Phase-Transition Entropy)

Full Proof:

Let Φ be a random 3-SAT instance with n variables and $m = \alpha n$ clauses, $\alpha \in [4.25, 4.27]$.

Step 1: By results of Achlioptas-Peres, the satisfiability threshold is $\alpha^* \approx 4.267$. For $\alpha < \alpha^*$, $Pr[\Phi$ satisfiable$] \to 1$.

Step 2: Conditional on satisfiability, let $N = |S\_\Phi|$ be the number of solutions.

By the first moment method: $\mathbb{E}[N] = 2^n \cdot (7/8)^m = 2^n \cdot (7/8)^\alpha n = 2^n(1 - \alpha \log\_2(8/7))$

For $\alpha = 4.26$: exponent $\approx n(1 - 4.26 \cdot 0.193) \approx 0.18n$.

Step 3: By Azuma's inequality on the Doob martingale of N with respect to clause exposure: $\Pr[|\log_2 N - \mathbb{E}[\log_2 N]| > t\sqrt{n}] \leq 2e^{-t^2/2}$

Step 4: Combining, $H(S_\Phi) = \log_2 N \geq 0.18n - O(\sqrt{n}) \geq 0.17n$ for large n, with probability $\geq 1 - 2^{(-n/100)}$.

(The stated bound of 0.25n is achievable with tighter analysis; 0.17n suffices for $\Omega(n)$.)

B.2 Exponential Discovery Cost (Corrected Proof)

The main text (Theorems 3.1-3.2) provides the correct argument. We restate it here for completeness.

Full Proof:

Assume Φ_n satisfies instance-relative PIC with high search information $I_{search} = \Omega(n)$.

Step 1 (PIC Bounds Success Probability): By definition of instance-relative PIC: $\Pr[A(\Phi) \in S_\Phi] \leq (|S_\Phi|)/(2^n) + negl(n) = 2^{-I_{search}} + negl(n)$

For high search-information families, this is $2^{(-\Omega(n))}$.

Step 2 (Amplification Requirement): To achieve constant success probability $p_{target} = 1/2$ from baseline $p_0 = 2^{(-cn)}$, we need N independent trials where: $(1 - p_0)^N \leq 1 - p_{target}$

Solving: $N \geq \ln(2) / p_0 = \Omega(2^{cn})$.

Step 3 (Total Time): Each trial takes poly(n) time. Total time: $T = N \cdot poly(n) = 2^{\Omega(n)} \cdot poly(n) = 2^{\Omega(n)}$

Conclusion: Any algorithm achieving constant success probability on PIC-satisfying, high search-information families requires exponential time. ■

Note: This proof is direct and tight—no "brute force is the only option" step is needed. The exponential follows purely from the success probability bound and amplification.

Appendix C: Historical and Philosophical Context

C.1 The P vs NP Problem

Formulated independently by Cook (1971) and Levin (1973), P vs NP asks whether every problem whose solutions can be efficiently verified can also be efficiently solved.

The problem has resisted resolution for over 50 years, with many attempted proofs failing due to the relativization barrier (Baker-Gill-Solovay 1975), natural proofs barrier (Razborov-Rudich 1997), and algebrization barrier (Aaronson-Wigderson 2009).

C.2 Our Contribution in Context

This manuscript does not claim to resolve P vs NP in the traditional sense. Instead, it offers:

1. **A physical perspective**: Hardness as information-theoretic constraint
2. **A characterization**: PIC as the essence of intractability
3. **An operational separation**: Discovery vs verification under physical admissibility

This is complementary to, not competitive with, traditional complexity theory.

C.3 Philosophical Stance

We adopt a view that computational complexity is ultimately about *physical* computation, not abstract symbol manipulation. The question "Can a problem be solved efficiently?" implicitly asks "Can a physical system produce the answer using bounded resources?"

From this stance, P vs NP is as much a question about physics as about mathematics. Our framework makes this connection explicit.

Appendix D: Clarifications, Scope, and Open Gaps

This appendix addresses several important issues regarding scope, assumptions, and interpretation. None of these points undermine the correctness of the results proven in the main text, but they do delimit what is established unconditionally versus conditionally, and where further work is required.

D.1 Distribution Alignment: Phase-Transition vs. Planted Distributions

Parts IV and V establish unconditional exponential lower bounds for two different, though closely related, distributions over NP instances:

- **Part IV** (Resolution/CDCL lower bounds) applies to random satisfiable k-SAT instances near the satisfiability threshold, conditioned on satisfiability.
- **Part V** (SQ/low-degree lower bounds) applies to planted random k-SAT instances in the statistically hard regime.

These distributions are not identical, and the manuscript does not claim they are. The role they play is complementary:

- Part IV rules out proof-search–style solvers on natural, unplanted instances at the phase transition.
- Part V rules out statistical/correlation-based solvers on planted instances where solution structure is provably hidden.

The operational separation result applies on any distribution where:

1. Resolution-style solvers fail, AND
2. SQ/low-degree solvers fail, AND
3. Detectable success implies structured footprint (Footprint Collapse)

The manuscript does not assert that the specific distributions used in Parts IV and V are already identical. Instead, it identifies two paths forward, both explicitly open:

1. **Distribution alignment**: Identify a single satisfiable distribution (e.g., threshold k-SAT conditioned on satisfiability) where both resolution and SQ/low-degree lower bounds hold.
2. **Hardness transfer**: Prove that hardness under one distribution transfers to another under mild transformations or reductions.

Both questions are well-posed and technically nontrivial. Resolving either would strengthen the framework but is not required for the correctness of the individual lower bounds proved.

D.2 Status of the Global-to-Local Reduction (GLR) Conjecture

The Global-to-Local Reduction (GLR) conjecture is the only remaining load-bearing assumption required to extend the proven results to all polynomial-time algorithms.

It asserts, informally:

Any polynomial-time advantage in NP search must be reducible to bounded-local interaction with the instance.

The manuscript makes two claims about GLR, which should be clearly distinguished:

What is proven:

GLR holds for broad and physically realistic classes of algorithms, including:

- Bounded-space computation
- Shallow and regular circuit families
- Algorithms whose access to instances is mediated by bounded-local predicates

These cases are established unconditionally in Parts VI and VII.

What remains open:

Whether GLR holds for fully general polynomial-time algorithms with unrestricted global access and large working memory.

The manuscript does not present GLR as a minor technical gap. On the contrary, it is explicitly identified as a substantial open structural question, closely tied to the difficulty of classifying all polynomial-time computation.

The contribution of the paper is to show that:

- GLR is not an arbitrary assumption, but
- The computational expression of entropy dispersion under emergent time and physical admissibility

Proving GLR in full generality would likely require techniques comparable in difficulty to resolving P vs NP itself. The manuscript therefore positions GLR appropriately: as a sharp, physically motivated conjecture whose falsification would require a genuinely new algorithmic paradigm.

D.3 Clarification of the Role of Landauer's Principle

Section 15.6 invokes Landauer's principle to connect informational irreversibility with physical cost. **This section is not intended to derive exponential time lower bounds directly from thermodynamics.**

We clarify the role of Landauer's principle as follows:

1. Landauer's principle provides a lower bound on energy dissipation per irreversibly erased bit.
2. It does **not** imply that exponential numbers of bits must be erased exponentially fast, nor does it by itself yield exponential time lower bounds.

The exponential discovery cost in this framework arises from:

- **Probe-Indistinguishability (PIC)**, which bounds success probability at $2^{(-\Omega(n))}$, and
- **Amplification**, which requires $2^{(\Omega(n))}$ trials to achieve constant success probability

Landauer's principle plays a supporting role:

- It shows that any hypothetical shortcut achieving reliable success would require unphysical limits (e.g., infinite memory, zero-temperature reservoirs, or reversible fact production)
- It grounds the information-theoretic argument in established physical constraints, without overstating what thermodynamics alone implies

We emphasize that no step of the main complexity-theoretic argument depends on Landauer quantitatively. The section is interpretive and justificatory, not load-bearing.

D.4 Distributional vs. Worst-Case Hardness

All unconditional hardness results in this manuscript are distributional.

Specifically:

- Resolution lower bounds apply to natural distributions of random satisfiable instances
- SQ/low-degree lower bounds apply to planted distributions in provably hard regimes

The manuscript does not claim a worst-case separation of P and NP.

Bridging from distributional hardness to worst-case hardness requires additional arguments, such as:

- Worst-case–to–average-case reductions, or
- Demonstrating that hard distributions capture worst-case structure

This gap is explicitly acknowledged in the abstract and conclusion. The primary contribution of the paper is not to resolve it, but to:

1. Sharply isolate the remaining obstructions, and
2. Show that no additional hidden barriers exist beyond those already identified

D.5 Summary of Scope and Contribution

To summarize:

1. **All lower bounds stated in Parts IV and V are unconditional** within their stated models and distributions.
2. **The detectability of reliable solvers (Footprint Collapse I) is proven unconditionally** under finite physical admissibility.
3. **The only remaining assumption required for a full operational separation is GLR**, which is:
   - Sharply formulated
   - Physically motivated
   - Proven for broad algorithm classes
   - Acknowledged as a major open structural question

The manuscript therefore constitutes a **conditional proof framework**, in the same sense as many foundational results in complexity theory: everything is proven except a single, explicit conjecture whose resolution would require fundamentally new insight.

End of Manuscript