Observer Structure and Computational Complexity:

A Geometric Framework from Multi-Observer Coherence

Léon Fernando Vlegels
Independent Researcher

2025-12

Contents

Abstract

We do not resolve the P versus NP problem. We propose a geometric framework, grounded in the multi-observer coherence structure of Paper 27, in which the P vs. NP question becomes observer-dependent. The key observation: for a single observer, the identity $C \circ P = I$ (Collapse composed with Projection equals Identity) is trivially satisfied; finding and verifying are inverse operations on the same structure, so the distinction does not arise. The asymmetry emerges only when a second observer is introduced, one who verifies what another found. We formalise this using the three-layer decomposition $\alpha^{-1} = 4\pi^3 + \pi^2 + \pi \approx 137.036$: bulk operations (90.5%) correspond to single-observer computation; boundary operations (7.2%) correspond to the shared verification interface; edge phenomena (2.3%) correspond to worst-case hard instances at phase transitions. The framework offers a conceptual lens for understanding why the question has resisted resolution: standard complexity theory implicitly assumes multi-observer structure without making this assumption explicit. We offer predictions and connections to algorithm design, but make no claim to resolve the Clay Millennium Problem.

Scope. This paper is part of the corpus's application series. It translates the P versus NP problem into the framework's geometry and derives conditional results inside that translation. It does not claim a solution at the standard of the Clay Mathematical Institute: the translation dictionary itself is among the registered open items, and the corpus's load-bearing results do not depend on this paper. The registry (Paper 40) records the specific defects.

1 Introduction

1.1 The Problem

The P versus NP problem asks: if a solution can be verified quickly, can it also be found quickly? Formally, does P = NP, where P is the class of problems solvable in polynomial time and NP is the class verifiable in polynomial time?

The question has resisted resolution. We propose the resistance stems from a hidden assumption: that "finding" and "verifying" are operations performed by the same entity. When we make this assumption explicit and examine what happens with multiple observers, the question becomes observer-dependent.

The framework reframes the question in geometric and conceptual terms; it does not address P vs. NP as a formal statement about Turing machines and polynomial-time complexity classes (see the scope statement above).

1.2 The Second Observer

Paper 27 [1] establishes that the geometric structure of physical reality emerges from multi-observer coherence. The core result: for a single observer, the identity

$$C \circ P = I \tag{1}$$

is trivially satisfied. Any operator P (projection/computation) admits an inverse $C = P^{-1}$ (collapse/verification), and the composition is identity by construction.

With two observers $\mathcal{O}_A$ and $\mathcal{O}_B$, the situation changes. Now we require:

$$C_A \circ P_A = I \quad \text{(A's self-coherence)} \tag{2}$$

$$C_B \circ P_B = I \quad \text{(B's self-coherence)} \tag{3}$$

$$C_B \circ P_A = I \quad \text{(cross-coherence: B verifies what A finds)} \tag{4}$$

The third condition is the crucial constraint. It requires that what observer A projects can be verified by observer B. This is *not* automatically satisfied; it forces geometric structure on the shared space.

1.3 The Computational Translation

Translate to complexity theory:

- P_A: Machine A searches for a solution (the "finding" operation)
- C_B: Machine B verifies the solution (the "checking" operation)
- $C_B \circ P_A = I$: Verification confirms what search produced

The P vs NP question, as standardly posed, assumes this two-machine setup. One machine (or the "finder" aspect of a machine) produces a candidate; another (or the "verifier" aspect) checks it.

Principle 1.1 (Observer-Dependence of Complexity)**.** *The separation $P \neq NP$ is an artifact of multi-observer structure. For a single observer, $P = NP$ trivially.*

1.4 Structure of This Paper

Section 2 reviews the three-layer ontology. Section 3 formalizes the single-observer case where P = NP. Section 4 shows how the second observer breaks this symmetry. Section 5 maps complexity classes to layers. Section 6 addresses implications and predictions. Section 7 concludes.

2 The Three-Layer Ontology

2.1 The Fundamental Decomposition

From Papers 1 and 4 [2, 3], the fine-structure constant admits exact decomposition:

$$\alpha^{-1} = 4\pi^3 + \pi^2 + \pi = 137.036... \tag{5}$$

This reveals three layers:

Table 1: Three-layer decomposition of α^{-1}.

Layer	Contribution	Fraction	Geometric role
Bulk	$4\pi^3$	90.5%	Interior of B^4
Boundary	π^2	7.2%	Surface S^3
Edge	π	2.3%	Fiber S^1

2.2 Intersubjective Interpretation

From Paper 27, these layers have intersubjective meaning:

- **Bulk (90.5%)**: Private to each observer: internal states, computations, experiences inaccessible to others
- **Boundary (7.2%)**: Shared between observers; the common reality where agreement is possible
- **Edge (2.3%)**: Irreducibly individual; the perspective that cannot be reduced to shared structure

2.3 The Coherence Identity

The fundamental interaction identity:

$$C \circ P = I \tag{6}$$

For a single observer, this holds automatically. For multiple observers, it becomes a constraint requiring shared boundary structure.

3 Single-Observer Complexity: P = NP

3.1 The Trivial Case

Consider a single computational entity with no external verifier.

Proposition 3.1 (Single-Observer Equivalence)**.** *For a single observer, finding and verifying are the same operation. Hence $P = NP$.*

Argument. Let $\mathcal{O}$ be a single observer with projection P (search/computation) and collapse C (verification/measurement).

Self-coherence requires $C \circ P = I$. This means $C = P^{-1}$ wherever P is invertible.

For the observer, "finding a solution" and "verifying a solution" are inverse operations on the same state space. The time to find equals the time to verify (up to constant factors) because they traverse the same structure in opposite directions.

There is no separation because there is no second entity to impose independent constraints. □

Remark 3.2 (Formal status). Proposition 3.1 is a philosophical observation, not a result in formal complexity theory. The operators P and C are not defined as Turing machines or circuit families; the identity $C \circ P = I$ is not a statement about time complexity. Standard complexity theory defines P and NP in terms of worst-case time on a fixed computational model, independently of any notion of observer. The claim that "P = NP for a single observer" does not contradict the standard conjecture P $\neq$ NP, because the terms are being used in different senses. The proposition shows that in the single-observer framework *as defined here*, the find/verify distinction does not arise; it says nothing about the formal question of whether SATISFIABILITY is in P.

Remark 3.3 (TBS). The invocation of $C \circ P = I$ to conclude "P = NP for a single observer" conflates two distinct formal systems. In standard complexity theory, P and NP are defined via language membership and worst-case Turing-machine time; neither involves an operator composition. The framework here defines P, C as abstract maps on an observer's state space, a definition that has no straightforward translation into circuit complexity or Turing complexity. Until a formal dictionary is provided (specifying which Turing-machine time class corresponds to which operator composition in which observer configuration), the conclusion "P = NP (single-observer)" cannot be evaluated as a statement of complexity theory, true or false. The existing Remark 3.2 acknowledges this; the derivation of Theorem 3.1 should therefore not be called a proof.

3.2 Why Verification Seems Easier

Even for a single observer, verification "feels" easier. Why?

Proposition 3.4 (Verification as Collapse)**.** *Verification is collapse onto a boundary; search is navigation through bulk. Collapse is dimensionally lower than navigation.*

Verification asks: "Is this point inside or outside the valid region?" This is a boundary determination: projection onto the S^3 surface.

Search asks: "Where in configuration space is a valid point?" This is bulk navigation: movement through the B^4 interior.

Dimensionally, boundary operations are simpler than bulk operations (3D vs 4D). But this dimensional reduction is exactly compensated by the $C \circ P = I$ identity: the information lost in collapse equals the information gained in projection.

For a single observer, these balance perfectly. The appearance of asymmetry is perspectival.

4 Two Observers: Symmetry Breaking

4.1 The Cross-Coherence Constraint

Now introduce observer $\mathcal{O}_B$ who will verify what $\mathcal{O}_A$ finds.

Theorem 4.1 (Symmetry Breaking by Second Observer)**.** *The introduction of a second observer breaks the $P = NP$ symmetry.*

Proof. With two observers, we require:

$$C_B \circ P_A = I \tag{7}$$

This is cross-coherence: B's collapse must compose with A's projection to yield identity.

Unlike self-coherence ($C_A \circ P_A = I$), cross-coherence is *not* automatically satisfied. It requires:

1. A shared space where both can operate
2. Agreement on what counts as valid projection/collapse
3. Communication of the solution from A to B

The communication step is where asymmetry enters. A must encode the solution (polynomial in solution size). B must decode and verify (polynomial in solution size). But A's *search* for the solution is not constrained by what B can verify; A operates in private bulk.

From B's boundary perspective, A's bulk navigation appears to require exponential time because B cannot access A's private structure. The P $\neq$ NP separation is B's view of A's computation. □

4.2 The Frame Change Problem

A deeper issue: verification cannot be performed without first finding.

Principle 4.2 (Frame Transformation)**.** *Finding a solution transforms the observer's reference frame. Verification occurs in the post-finding frame, not the pre-finding frame.*

Before finding: the solution is unknown; the observer's state space includes all possibilities.

After finding: the solution is known; the observer's state space has collapsed to a specific branch.

Verification is performed *after* this collapse. The entity that verifies is not the same entity that searched; it has been transformed by the search. Comparing "search time" and "verify time" compares operations in different frames.

This is why the question "Can finding be as fast as verifying?" is malformed. Finding *includes* the frame transformation; verifying *assumes* it already happened.

4.3 Formalizing the Asymmetry

Definition 4.3 (Pre-Solution Frame)**.** The pre-solution frame $\mathcal{F}_0$ is the observer's state space before any solution is found. Dimension: full bulk ($\dim B^4 = 4$).

Definition 4.4 (Post-Solution Frame)**.** The post-solution frame $\mathcal{F}_1$ is the observer's state space after a solution is found. Dimension: collapsed to boundary ($\dim S^3 = 3$).

Theorem 4.5 (Frame-Dependent Complexity)**.** *(i) In $\mathcal{F}_0$: Search is 4-dimensional navigation*

(ii) In $\mathcal{F}_1$: Verification is 3-dimensional projection

(iii) The transition $\mathcal{F}_0 \to \mathcal{F}_1$ is the "finding" operation
Comparing search time (in $\mathcal{F}_0$) to verify time (in $\mathcal{F}_1$) compares incommensurable quantities.

5 Complexity Classes as Layers

5.1 The Correspondence

We now map complexity classes to ontological layers:

Definition 5.1 (Layer-Complexity Correspondence)**.**

$$\text{P} \longleftrightarrow \text{Bulk operations (single-observer)} \tag{8}$$

$$\text{NP-verification} \longleftrightarrow \text{Boundary operations (shared)} \tag{9}$$

$$\text{NP-hard (worst case)} \longleftrightarrow \text{Edge phenomena (singular)} \tag{10}$$

5.2 Interpretation

Bulk (P, 90.5%):
The interior where single-observer computation flows. Polynomial-time algorithms navigate bulk structure: they follow geodesics in configuration space. No exponential branching because the bulk is simply connected.

From the bulk perspective, finding and verifying coincide: they are inverse traversals of the same structure, so in the framework's single-observer reading the P = NP identification holds at the level of its own definitions.

Boundary (NP-verification, 7.2%):
The interface where observers must agree. Verification is a boundary operation: it checks whether a proposed solution lies inside or outside the valid region. This projection onto shared structure is inherently polynomial.

From the boundary perspective, verification is primary; search appears harder because it requires accessing another observer's bulk.

Edge (NP-hard, 2.3%):
The singular locus where worst cases live. Hard instances cluster at phase transitions, the edge where boundary meets boundary. Small input changes produce large output changes.

From the edge perspective, exponential blowup is visible and unavoidable.

5.3 Why Most Instances Are Easy

Empirical observation: most NP-complete instances are solved quickly. SAT solvers handle millions of variables. The "hard" instances are rare.

The layer fractions explain this:

- 90.5% of instances live in bulk → polynomial (P-like behavior)

- 7.2% live at boundary → polynomial verification dominates
- 2.3% live at edge → exponential worst case

Conjecture 5.2 (Hard Instance Measure)**.** *The fraction of genuinely hard instances for any NP-complete problem is approximately* $2.3\% = \pi/\alpha^{-1}$.

Remark 5.3 (TBS). The identification of the edge-layer fraction $\pi/\alpha^{-1} \approx 2.3\%$ with the fraction of hard NP-complete instances is asserted without proof. No polynomial-time reduction from a standard NP-complete problem (SAT, 3-SAT, Hamiltonian cycle, etc.) to the geometric edge layer is given; nor is there a proof that the edge layer has the same measure as the hard-instance measure for any specific problem distribution. The empirical 3-SAT phase-transition ratio (≈ 4.26) is cited suggestively ($4.26 \approx \alpha^{-1}/32$), but the factor 32 is unaccounted for and the coincidence is not derived. The conjecture is formally open and requires a separate theorem establishing the measure correspondence.

5.4 Phase Transitions

The edge layer corresponds to phase transitions in random instances:

Proposition 5.4 (Phase Transition as Edge)**.** *The SAT phase transition (clause-to-variable ratio* ≈ 4.26 *for 3-SAT) occurs at the edge layer, the boundary between satisfiable and unsatisfiable regions.*

Hard instances cluster here because:

1. Both SAT and UNSAT require global reasoning (no local shortcuts)
2. Small changes flip satisfiability (sensitivity)
3. The solver cannot distinguish which side without full search

This is precisely edge behavior: singular structure where small perturbations have large effects.

6 The Observer-Dependent Reading

6.1 Precise Statement

Theorem 6.1 (Observer-Dependent Reading of P vs NP)**.** *The P versus NP question admits different answers depending on observer structure:*

1. ***Single observer (bulk)****: $P = NP$. Finding and verifying are inverse operations; $C \circ P = I$ guarantees equivalence.*
2. ***Two observers (boundary)****: $P \subseteq NP$ with apparent separation. Cross-coherence constraints create asymmetry between finder and verifier.*
3. ***Edge observation****: $P \neq NP$ with exponential separation. Singular structure at phase transitions creates worst-case gaps.*

Argument. Single observer: Proposition 3.1.

Two observers: Theorem 4.1. The cross-coherence requirement $C_B \circ P_A = I$ forces A's bulk computation to be communicated through shared boundary, creating dimensional mismatch (4D search appears exponential from 3D verification perspective).

Edge: The 2.3% edge fraction corresponds to measure of instances at phase transitions. These exhibit exponential behavior because they require resolving boundary/boundary contact, navigation near singularities. □

Remark 6.2 (TBS). Theorem 6.1 does not resolve the formal Clay Millennium Problem and cannot do so within the framework as given. Parts (1)–(3) redescribe the framework's own definitions: "P = NP (single observer)" uses the framework's non-standard P, C operators; "P $\subseteq$ NP (boundary)" does not establish a formal complexity separation; "P $\neq$ NP (edge, 2.3%)" identifies worst-case behavior with a geometric layer but provides no diagonalisation or circuit-lower-bound argument. The theorem's proof cites Theorems 3.1 and 4.1, both of which are philosophical observations in the framework's own terms rather than formal complexity results. Resolving the Clay problem requires showing that any polynomial-time algorithm for an NP-complete language either exists or provably does not; no argument of this form appears in the paper.

6.2 Why Standard Proofs Fail

Standard approaches seek a layer-independent answer. But:

- Proofs of P = NP implicitly assume single-observer structure (bulk access)
- Proofs of P $\neq$ NP implicitly assume multi-observer structure (boundary separation)
- Neither can succeed because each is correct for its layer

Remark 6.3 (Barriers and observer structure). The framework suggests a possible interpretation of the known proof barriers (relativization, natural proofs, algebrization): each barrier may correspond to a technique that implicitly fixes observer structure. Relativization works relative to an oracle (an external verifier, i.e. a two-observer setting). Natural proofs require properties decidable by small circuits, a boundary-level check. Algebrization extends relativization to low-degree extensions. If this interpretation is correct, proofs that transcend all three barriers would need to operate in a regime that does not presuppose observer structure. This is a conjecture, not a theorem; we do not claim that no proof of P $\neq$ NP exists within standard axioms.

This is a speculative observation consistent with the framework. It would require a formal translation of "observer structure" into circuit complexity to become a theorem.

6.3 The Self-Reference Angle

Any system capable of formulating P vs NP rigorously is self-referential: it models computation while being a computation.

Proposition 6.4 (Self-Reference Implies Bulk Access)**.** *A self-referential computational system operates above the self-lensing threshold and has implicit bulk access.*

From Paper 3 [4], the self-lensing energy is:

$$E_{\text{self}} = 13.177... \tag{11}$$

Systems operating above this threshold can model themselves. Such systems implicitly operate in the bulk; they must, to contain their own model.

Corollary 6.5. *Any system capable of rigorously asking "$P \stackrel{?}{=} NP$" already operates in a regime where $P = NP$, but it may phrase the question from boundary perspective, obscuring this fact.*

7 Implications

7.1 For Complexity Theory

1. **Definitional refinement**: Complexity definitions should specify observer structure. "Polynomial time for whom?" matters.

2. **Average-case focus**: Since 90.5% of instances are bulk (easy), average-case complexity is more physically meaningful than worst-case.

3. **Verification-guided search**: The identity $C \circ P = I$ suggests verification structure can guide search. SAT solvers already exploit this (unit propagation, clause learning).

7.2 For Algorithm Design

1. **Exploit bulk structure**: Algorithms that navigate simply-connected bulk will be polynomial. Those that fight edge singularities will be exponential.

2. **Avoid phase transitions**: If possible, transform instances away from the 2.3% edge region.

3. **Single-observer simulation**: If you can simulate single-observer structure (treat find and verify as single unified operation), you access P = NP equivalence.

7.3 For Quantum Computing

Quantum computation operates at the boundary: superposition is neither definitely inside nor outside until measured.

Proposition 7.1 (Quantum Speedup Ceiling)**.** *Quantum computers provide at most polynomial speedup for NP problems (boundary access), not exponential (bulk access).*

This matches observations: Grover's algorithm gives quadratic speedup, not exponential. Shor's algorithm achieves exponential speedup for factoring, but factoring may not be NP-complete; it might have hidden bulk structure.

7.4 For Artificial Intelligence

AI systems increasingly operate as single observers: they find and verify within unified architectures.

Conjecture 7.2 (AI Complexity Advantage)**.** *Sufficiently integrated AI systems may achieve $P = NP$-like behavior for problems where the find/verify distinction is artifactual.*

This is not a claim that AI "solves" NP-hard problems. It is a claim that the find/verify separation may be less fundamental than assumed when observer structure is unified.

8 Predictions

1. **Hard instance fraction**: Approximately 2.3% of random NP-complete instances should be genuinely hard (requiring exponential time). This is testable empirically.

2. **Phase transition correlation**: The location of phase transitions (e.g., 3-SAT at ratio 4.26) should relate to geometric layer boundaries. The ratio $4.26 \approx \alpha^{-1}/32$ suggests connection.

3. **Quantum speedup limits**: No quantum algorithm should achieve exponential speedup for NP-complete problems generally (boundary access ceiling).

4. **Self-referential complexity**: Problems involving self-reference should show anomalous complexity behavior (implicit bulk access).

5. **Unified architectures**: Computational architectures that unify find/verify (no external verifier) should outperform separated architectures on NP-intermediate problems.

9 Conclusion

The P versus NP problem asks whether finding can be as fast as verifying. We have argued, within the framework's translation, that this question is observer-dependent.

For a single observer, $C \circ P = I$ guarantees P = NP: finding and verifying are inverse operations on the same structure. The separation arises only when a second observer introduces cross-coherence constraints: when one entity must verify what another found.

The three-layer decomposition $\alpha^{-1} = 4\pi^3 + \pi^2 + \pi$ maps naturally to complexity structure:

- Bulk (90.5%): P = NP (single-observer equivalence)
- Boundary (7.2%): P $\subseteq$ NP (verification-centric separation)
- Edge (2.3%): P $\neq$ NP (worst-case exponential gaps)

This paper does not prove P = NP or P $\neq$ NP. It proposes a conceptual reframing: the question, as standardly posed, implicitly assumes multi-observer verification structure, and different observer configurations yield different answers within the framework, all internally consistent, none settling the formal question.

Standard complexity theory assumes multi-observer verification, which is appropriate for cryptography, distributed computation, and proof checking. The framework suggests that making this assumption explicit may clarify why existing proof techniques face systematic barriers.

Three concrete predictions follow from the framework (Section 7) and are independently testable: the hard-instance fraction, phase transition geometry, and quantum speedup ceilings. These provide empirical traction regardless of the formal status of the observer-dependence thesis.

References

[1] L. F. Vlegels, *The Second Observer: Why Existence Requires Witness and How Intersubjectivity Generates Geometry*, This volume (2025).

[2] L. F. Vlegels, *The Perfect Stable Sphere: Deriving α^{-1} from (B^4, S^3) Geometry*, This volume (2025).

[3] L. F. Vlegels, *The Three-Layer Ontology of Physical Reality*, This volume (2025).

[4] L. F. Vlegels, *Mathematical Foundations of Geometric Fundamental Physics*, This volume (2025).

[5] L. F. Vlegels, *Rosetta Map of the Monad Identity and the Geometric Theory of Everything*, This volume (2025).

[6] S. A. Cook, "The complexity of theorem-proving procedures," *Proc. 3rd ACM Symposium on Theory of Computing*, 151–158 (1971).

[7] L. A. Levin, "Universal sequential search problems," *Problems of Information Transmission* **9**, 265–266 (1973).

[8] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press (2009).

[9] A. A. Razborov and S. Rudich, "Natural proofs," *J. Comput. Syst. Sci.* **55**, 24–35 (1997).

[10] S. Aaronson and A. Wigderson, "Algebrization: A new barrier in complexity theory," *ACM Trans. Comput. Theory* **1**, 2:1–2:54 (2009).