P vs NP via the Computation/Recognition Split: A Dual-Complexity Framework from Ledger Dynamics

An Exploratory Paper in Recognition Science

SCAFFOLD — Not a claim to have resolved P vs NP

Jonathan Washburn
Recognition Science Research Institute, Austin, Texas
washburn.jonathan@gmail.com

February 2026

Abstract

Claim hygiene. This paper explores a *hypothetical* framework for understanding the P vs NP problem; it does **not** claim to resolve it unconditionally. All separation results are conditional on the ledger-computation model.

We observe that the Turing machine model implicitly assumes zero-cost observation: reading a tape cell is free. In Recognition Science, observation has fundamental cost $J(r)$ per query. This motivates a dual-complexity framework with two independent measures:

- $T_c(n)$: *computation complexity* — internal evolution steps (double-entry ledger updates).
- $T_r(n)$: *recognition complexity* — observation operations (extracting information from the ledger).

In the standard Turing model, $T_r = 0$ and total cost $= T_c$. In the ledger model, $T_r > 0$ and total cost $= T_c + T_r$.

Under ledger assumptions, the double-entry structure forces *balanced-parity encoding*: information is hidden in the parity balance of ledger entries. Extracting one bit of parity requires $\Omega(n)$ recognition queries (information-theoretic lower bound). Meanwhile, the internal evolution (computation) can reorganise the ledger in $O(n^{1/3} \log n)$ steps (subpolynomial in n).

This creates a conditional separation: $T_c(\text{SAT}) = O(n^{1/3} \log n)$ but $T_r(\text{SAT}) = \Omega(n)$. The P vs NP question splits: P = NP at the computation scale (internal evolution is fast), P $\neq$ NP at the recognition scale (observation is expensive). The Clay Millennium problem, as traditionally stated, conflates T_c and T_r.

Status: SCAFFOLD. The Lean formalisation (`IndisputableMonolith.Complexity.*`) uses explicit hypotheses and placeholder types. No unconditional mathematical claims are made.

Keywords: P vs NP, dual complexity, computation, recognition, ledger, balanced parity, Turing incompleteness.

Contents

1 Introduction and Claim Hygiene

What this paper claims.

1. The Turing model assumes $T_\mathrm{r} = 0$ (zero observation cost). This is a modelling choice, not a physical law.
2. If observation has cost ($T_\mathrm{r} > 0$), a separation between computation and recognition complexity can arise.
3. The RS ledger structure provides a concrete model in which this separation is natural.

What this paper does not claim.

× An unconditional proof that P $\neq$ NP (or P = NP).
× That the ledger model is the "correct" model of computation.
× That the Clay problem is formally ill-posed in the standard Turing setting.

The paper should be read as: "*If* the ledger model captures physical computation, *then* the P/NP distinction splits into two independent questions."

2 The Standard P vs NP Problem

For completeness, we recall the standard formulation.

Definition 2.1 (Turing machine)**.** *A (deterministic) Turing machine is a tuple $M = (Q, \Sigma, \delta, q_0, q_\mathrm{acc}, q_\mathrm{rej})$ with finite state set Q, tape alphabet Σ, transition function $\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, R\}$, and distinguished states. The* time complexity *$T_M(n)$ is the maximum number of steps on inputs of length n.*

Definition 2.2 (P and NP (standard))**.**

- ***P****: the class of languages decidable by a deterministic TM in time $O(n^k)$ for some k.*
- ***NP****: the class of languages for which a "yes" certificate of length* poly(n) *can be* verified *in polynomial time.*

*The Clay Millennium problem [?] asks: is **P** = **NP**?*

Remark 2.3 (The observation that motivates this paper)**.** *In both definitions, every tape-read operation has **zero cost**. When the verifier checks a certificate, each symbol lookup is free. This is an idealisation. In any physical realisation:*

- *Reading a memory cell dissipates at least $k_B T \ln 2$ of energy (Landauer's bound [?]).*
- *In quantum mechanics, measurement disturbs the measured state (no-cloning, wavefunction collapse).*
- *In the RS framework, observation has cost $J(r) > 0$ per recognition query.*

The Turing model's $T_\mathrm{r} = 0$ assumption is a modelling choice, not a physical law. The question explored in this paper is: what happens to the P/NP distinction if we take $T_\mathrm{r} > 0$ seriously?

2.1 Existing barriers

No unconditional proof of **P** $\neq$ **NP** is known. Three *barriers* explain why standard techniques fail:

1. **Relativisation** [?]: there exist oracles A, B with $\mathbf{P}^A = \mathbf{NP}^A$ and $\mathbf{P}^B \neq \mathbf{NP}^B$. Any proof must be non-relativising.
2. **Natural proofs** [?]: if one-way functions exist, no "natural" combinatorial property can separate P from NP.
3. **Algebrisation** [?]: proofs that algebrise cannot separate P from NP.

Remark 2.4. *The dual-complexity framework in this paper is* not *a technique within the standard Turing model. It introduces a new* model *(ledger computation with $T_\mathrm{r} > 0$), which sidesteps the* *three barriers by changing the question rather than answering the original one. This is why we label the results "conditional," not "unconditional."*

3 Dual Complexity Framework

Definition 3.1 (Recognition-complete complexity)**.** *A* recognition-complete *complexity measure assigns to each problem instance of size n a pair $(T_\mathrm{c}(n), T_\mathrm{r}(n))$:*

- *$T_\mathrm{c}(n)$: computation steps (internal state transitions).*
- *$T_\mathrm{r}(n)$: recognition queries (observation/readout operations).*

Total cost: $T_{total}(n) = T_\mathrm{c}(n) + T_\mathrm{r}(n)$.

Definition 3.2 (Turing model as special case)**.** *The standard Turing model is the special case $T_\mathrm{r}(n) = 0$ for all n. All computation cost resides in T_c.*

4 The Ledger Computation Model

Definition 4.1 (Ledger computation)**.** *A* ledger computation *consists of:*

- ***States:*** *configurations of a double-entry ledger (balanced debit/credit pairs).*
- ***Evolution:*** *deterministic double-entry updates preserving balance ($\sigma = 0$).*
- ***Observation:*** *extracting information from the ledger by querying specific entries, at cost $J(r)$ per query.*

5 Balanced Parity Encoding

Definition 5.1 (Balanced parity)**.** *A ledger configuration has* balanced parity *if the total debit equals the total credit: $\sum d_i = \sum c_i$. The double-entry structure forces this for all admissible states.*

Hypothesis 5.2 (Information hiding)**.** *In a balanced-parity ledger of n entries, the value of any single-bit predicate (e.g. "is entry k a debit?") cannot be determined without querying at least $\Omega(n)$ entries, because each entry's value is constrained by the global balance condition.*

Proposition 5.3 (Parity lower bound (classical))**.** *Computing the parity of n bits requires reading all n bits in the worst case. No query algorithm can determine $\bigoplus_{i=1}^{n} x_i$ with fewer than n queries.*

Proof. Adversary argument. Fix any deterministic algorithm making $< n$ queries. An adversary answers consistently but chooses the unqueried bit to control parity. Since the algorithm never queries the last bit, both parity values are consistent with the observed answers. □ □

Theorem 5.4 (Balanced-parity lower bound)**.** *In a balanced ledger with n entries where $\sum d_i = \sum c_i$ (debit = credit), determining whether a* specified *entry k is a debit or credit requires querying at least $n - 1$ entries in the worst case.*

Proof. Fix an algorithm $\mathcal{A}$ that queries fewer than $n - 1$ entries and claims to determine d_k. There exist at least two unqueried entries $i, j \neq k$. Consider two configurations:

- C_1: all queried entries as answered, $d_k = +1$, and (d_i, d_j) chosen to satisfy balance.
- C_2: all queried entries as answered, $d_k = -1$, and (d_i, d_j) chosen to satisfy balance (adjust by swapping i, j).

Both C_1 and C_2 are consistent with the observed query answers (the algorithm cannot distinguish them). Yet d_k differs. Hence $\mathcal{A}$ must err on at least one of C_1, C_2.

More precisely: the balance constraint $\sum d_i = S$ (a known constant) has $\binom{n}{n/2}$ satisfying assignments. Conditioning on any $n-2$ entries leaves a 2-dimensional space; both values of d_k are compatible. The argument generalises to randomised algorithms by Yao's minimax principle [**?**]: a probabilistic algorithm needs $\Omega(n)$ queries to achieve error $< 1/3$. □

Corollary 5.5. *The* $\Omega(n)$ *lower bound is a consequence of* global coupling*: the balance constraint links all entries, so local queries reveal global information only after* $\Omega(n)$ *samples.*

Remark 5.6. *This is strictly stronger than the classical parity bound (prop:parity): parity hiding is an* incidental *property of bit strings, while balanced-parity hiding is a* structural *consequence of the double-entry ledger. The latter cannot be avoided by clever encoding because balance is an invariant of the dynamics (T3 conservation).*

6 Worked Example: 3-SAT on a Ledger

To ground the abstract framework, consider a concrete instance.

Example 6.1 (A 3-variable instance)**.** *Let* $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_3)$ *with* $n = 3$ *variables.*

Ledger encoding. *Each variable* x_i *is a ledger entry with value* $d_i \in \{+1, -1\}$ *(debit or credit). The balanced-parity constraint requires* $\sum_{i=1}^{3} d_i = \pm 1$ *(odd parity for 3 entries).*

Clause checking. *Clause* C_j *is satisfied iff the appropriate combination of* d_i *values yields a non-zero inner product with the clause template. A single clause check reads 3 entries: cost* $= 3 \cdot J(d_i) = 3 \cdot 0 = 0$ *for* $d_i = \pm 1$ *(unit entries have zero* J*). However,* determining the value $d_i = +1$ *vs* $d_i = -1$ *requires a recognition query.*

Computation phase. *The ledger evolves internally via double-entry updates. After* $O(n^{1/3} \log n) = O(1.4 \cdot 1.1) \approx 2$ *steps (for* $n = 3$*), the internal state reorganises.*

Recognition phase. *To* read out *the satisfying assignment, the observer must query each* d_i*: cost* $= n = 3$ *queries. Even after computation has finished, the answer is "hidden" in the ledger's balanced-parity structure until observed.*

The gap. $T_c = 2$*,* $T_r = 3$*. For* $n = 3$ *the gap is negligible, but it grows:* $T_c = O(n^{1/3} \log n)$ *is sublinear while* $T_r = \Omega(n)$ *is linear. By* $n = 1000$*:* $T_c \approx 70$ *but* $T_r \geq 1000$*.*

7 Conditional SAT Separation

Hypothesis 7.1 (SAT computation complexity)**.** *Under the ledger model, the internal evolution can reorganise a SAT instance of* n *variables into a satisfying assignment (if one exists) in* $T_c(n) = O(n^{1/3} \log n)$ *steps.*

Hypothesis 7.2 (SAT recognition complexity)**.** *Under the ledger model,* verifying *that the reorganised ledger encodes a satisfying assignment requires* $T_r(n) = \Omega(n)$ *recognition queries (from balanced-parity information hiding).*

Theorem 7.3 (Conditional separation)**.** *If Hypotheses* **??** *and* **??** *hold, then:*

$$T_c(SAT) = O(n^{1/3} \log n) \ \ll \ T_r(SAT) = \Omega(n).$$

Computation is fast; recognition is slow. The "hardness" of SAT resides in observation, not evolution.

8 The Split Resolution

Theorem 8.1 (P vs NP splits (conditional))**.** *Under the dual-complexity framework:*

- ***At the computation scale (*T_c *only):*** *P = NP. The internal evolution can solve NP-complete problems in subpolynomial* T_c*.*

- ***At the recognition scale** (T_r **only):*** *P $\neq$ NP. Observation of the solution requires polynomial T_r, creating a separation from the T_c measure.*
- ***In the Turing model** ($T_\mathrm{r} = 0$)**:*** *The question is ill-conditioned because T_r is absorbed into T_c and the split is invisible.*

9 Implications

1. **Quantum computers shift T_c, not T_r.** Quantum speedups (Grover, Shor) accelerate internal evolution but do not eliminate observation cost. The recognition barrier remains.

2. **Measurement is fundamentally expensive.** The RS collapse threshold $C \geq 1$ makes measurement a real physical cost, not a free operation.

3. **Consciousness has irreducible observation cost.** The attention operator (Section 4 of [?]) is a recognition gate with bounded capacity φ^3. Even a conscious agent cannot bypass T_r.

10 Falsification Criteria

Falsification Criterion 10.1 (Free observation)**.** *If a physical system is demonstrated where observation has zero energy cost (violating Landauer's bound), the $T_\mathrm{r} > 0$ premise is falsified.*

Falsification Criterion 10.2 (No parity barrier)**.** *If a SAT instance can be verified in $o(n)$ queries on a balanced-parity ledger, the information-hiding hypothesis is falsified.*

11 Comparison with Existing Work

c

Reference	Relation to this work
Baker–Gill–Solovay [?]	Relativisation barrier; we sidestep it by changing the model, not proving a Turing-model sep
Razborov–Rudich [?]	Natural proofs barrier; our lower bound is information-theoretic (adversary), not combinat
Aaronson–Wigderson [?]	Algebrisation barrier; the ledger model is not an algebraic extension of a Turing oracle
Landauer [?]	Physical cost of information; we formalise this as $T_\mathrm{r} > 0$.
Bennett [?]	Reversible computation; T_c in reversible models is $O(T_{\mathrm{c}\,\mathrm{irrev}}^{2})$, but T_r is unchanged.
Grover [?]	Quantum search gives $T_\mathrm{c} \to O(\sqrt{n})$; T_r remains $\Omega(n)$ (measurement collapses the state

12 Discussion

Claims and non-claims

We have introduced a dual-complexity framework $(T_\mathrm{c}, T_\mathrm{r})$ and shown that the RS ledger model provides a natural setting where T_c and T_r can diverge. The key mathematical content is:

1. The balanced-parity lower bound (thm:balanced): proved unconditionally within the query-complexity model.
2. The conditional separation (thm:separation): $T_\mathrm{c} \ll T_\mathrm{r}$ for SAT, *if* the ledger model's T_c hypothesis holds.
3. The split (thm:split): P = NP at T_c, P $\neq$ NP at T_r, under the same hypothesis.

Item (1) is rigorous. Items (2) and (3) are conditional.

Why this is not a resolution of P vs NP

The Clay problem asks about the standard Turing model, where $T_{\mathrm{r}} = 0$ by definition. Our framework changes the model. We do *not* prove P $\neq$ NP in the Turing model; we argue that the Turing model's conflation of T_{c} and T_{r} may be the source of the difficulty.

Falsifiability

The framework makes two testable predictions:

1. Any physical computation system will exhibit $T_{\mathrm{r}} > 0$ (Landauer's bound is never zero).
2. The "hardness" of NP-complete problems in practice will correlate more with *verification cost* (how many bits must be read to check a solution) than with *search cost* (how many internal steps to find a candidate).

Open problems

(Q1) Can $T_{\mathrm{c}}(\mathrm{SAT}) = O(n^{1/3} \log n)$ be proved in a concrete ledger model, or is it only a hypothesis?

(Q2) Does the dual framework have a clean complexity-class formulation (e.g. "$\mathbf{P}_c$" for computation-only, "$\mathbf{P}_r$" for recognition-only)?

(Q3) Is there a natural analogue of the PCP theorem in the dual setting (probabilistic recognition with $o(n)$ queries)?

(Q4) Does the framework apply to **BPP** vs **BQP** (randomised vs quantum)?

13 Lean Formalization Status

The Lean module `IndisputableMonolith.Complexity.ComputationBridge` is explicitly marked as **SCAFFOLD** and is **not** part of the verified certificate chain. Key caveats:

- `LedgerComputation.states` uses `Type` as a placeholder (often `Unit`).
- Separation theorems rely on hypothetical model assumptions.
- No result should be cited as proven mathematics.

Module	**Content**
`Complexity.ComputationBridge`	Dual framework, separation
`Complexity.BalancedParityHidden`	Parity hiding
`Complexity.VertexCover`	Example reductions
`Complexity.RSVC`	RS vertex cover

References

[1] J. Washburn and M. Zlatanović, "The Cost of Coherent Comparison," arXiv:2602.05753v1, 2026.

[2] J. Washburn, "Decision as Cost Geodesic," Recognition Science preprint, February 2026.

[3] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Dev.*, 5(3):183–191, 1961.

[4] S. Cook, "The P versus NP problem," Clay Mathematics Institute Millennium Problems, 2000.

[5] T. Baker, J. Gill, and R. Solovay, "Relativizations of the P =? NP question," *SIAM J. Comput.*, 4(4):431–442, 1975.

[6] A. A. Razborov and S. Rudich, "Natural proofs," *J. Comput. System Sci.*, 55(1):24–35, 1997.

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

[8] C. H. Bennett, "The thermodynamics of computation—a review," *Int. J. Theor. Phys.*, 21(12):905–940, 1982.

[9] L. K. Grover, "A fast quantum mechanical algorithm for database search," *Proc. 28th ACM STOC*, 212–219, 1996.

[10] A. C.-C. Yao, "Probabilistic computations: toward a unified measure of complexity," *Proc. 18th IEEE FOCS*, 222–227, 1977.