


# To XOR a Stone with Six Birds: Closure Diagnostics for Emergent Bits, Gates, and Booleanity

Ioannis Tsiokos 

March 9, 2026

## Abstract

Boolean logic is usually treated as a formal starting point. We ask instead when a substrate actually earns a logic layer. In the Six Birds emergence calculus—P1 operator rewrite, P2 gating/constraints, P3 protocol holonomy, P4 sectors/invariants, P5 packaging, and P6 accounting/audit—we define a logic layer as an audited closure of packaged predicates under induced dynamics. This yields two complementary views. In the feasibility view, propositions are definable feasible sets and connectives are their stable closure operations. In the operator view, bits are packaged equivalence classes and gates are macro-updates induced by the underlying dynamics. We implement quantitative diagnostics for both views: metastability, route mismatch, closure defect, gate error, conditional entropy, unretained input information, and entropy-production audits. Across controlled finite Markov laboratories we obtain four main results. First, parity/XOR-type variables are unusually robust coarse variables: the parity lens beats the median random binary partition across the full tested grid. Second, NOT, AND, and reversible XOR embeddings exhibit distinct closure regimes under noise, staging, and timescale stress. Third, when a reversible CNOT embedding is coarse-grained to an erased XOR output, closure degrades sharply and unretained input information increases by more than eightfold relative to the retained view. Fourth, a minimal unsupervised discovery pipeline recovers stable bits and reconstructs a NOT gate directly from transition structure. The result is a falsifiable criterion for when a system has logic at a layer—and for which logic it has—without treating logic as fundamental.

**Keywords:** emergent logic; coarse-graining; closure diagnostics; parity and XOR; reversible computation; information loss.

**Contribution summary.** This paper contributes a layer-relative definition of logic as audited closure, an operational diagnostic suite for detecting bits and gates in coarse-grained dynamics, and controlled demonstrations of parity robustness, reversible-versus-erased gate divergence, and unsupervised recovery of packaged bits and a NOT gate from transition structure.

## 1 Introduction

Many canonical discussions of physical computation and reversible logic begin from a prior vocabulary of bits, registers, and truth tables [1–4]. That question is important, but it already assumes the existence of a symbolic layer. Before implementation lies a harder prior question: when does a coarse-grained layer deserve to be described in logical terms at all? A system may admit labels that look like 0 and 1 without supporting stable predicates, reliable negation, or closed gate operators. In that case, logic has been projected onto the system rather than discovered in it.

This paper gives an operational answer. We work in the Six Birds emergence calculus [5], whose six primitives are P1 operator rewrite, P2 gating / constraints, P3 autonomous protocol holonomy,

P4 sectors / invariants, P5 packaging, and P6 accounting / audit. Our central claim is that logic is not a seventh primitive and not a hidden Platonic layer. It is a layer-relative achievement of these six ingredients. A system has a logic layer when packaged binary predicates become stable, when induced macro-operators close on those predicates, and when the reliability and loss of that closure can be audited. In one sentence: a logic layer is an audited closure of packaged predicates under induced dynamics.

That perspective shifts the usual starting point. In the feasibility view, propositions are not abstract atoms but definable feasible sets, and logical connectives are stable closure operations on those sets. In the operator view, bits are not fundamental carriers but packaged equivalence classes of microstates, and gates are macro-updates induced by the underlying dynamics. The two views are complementary. The first gives semantics to truth; the second tells us when those semantics are executable by a substrate. Together they replace the familiar question “how is logic implemented?” with the more basic question “when has logic emerged strongly enough to close?” [6–8]

The title emphasizes XOR for a reason. Parity-like variables aligned with a system’s invariant or sector structure are often especially natural coarse variables; in this paper parity is the controlled test case for that more general idea [8–10]. It survives cancellation of micro-detail, aligns naturally with invariant structure, and persists under coarse-graining in regimes where more brittle variables do not. That does not make XOR metaphysically privileged. It makes parity an empirical test case for the broader thesis of the paper: some logical forms are easier for substrates to stabilize than others. The right question is therefore not whether a substrate can be made to emulate a desired truth table in principle, but which logical distinctions it supports natively, robustly, and at what accounting cost.

Our emphasis is therefore diagnostic rather than merely implementational. We introduce a suite of quantitative tests for candidate logic layers: predicate stability, route mismatch, closure defect, gate error, conditional entropy, unretained input information, and entropy-production audits. These tests distinguish genuine closure from symbolic overinterpretation. They let us ask whether two routes to the same purported macro-description agree, whether a packaged bit persists long enough to matter, whether a gate is really a closed macro-operator rather than a protocol artifact, and whether information that appears to disappear has in fact been erased, hidden, or retained in side variables. This is also where the Six Birds grammar becomes empirically useful: packaging makes predicates possible, constraints give them meaning, protocol effects create route dependence, sectors stabilize them, rewrite repairs or redesigns closure, and accounting determines what can be maintained without cheating.

Our starting point is therefore closer to coarse-graining and audited closure than to implementation-first accounts of logic. Internally, the paper draws on the Six Birds foundations [5] and on the broader closure-and-substrate program developed in [11, 12]. Externally, the descent of dynamics to coarse variables is related to lumpability and effective Markov aggregation [9, 13–15], while the distinction between retained and erased descriptions connects directly to the classic literature on irreversibility and reversible computation [1, 2, 4].

The payoff is a controlled program rather than a metaphor. We build finite Markov laboratories in which NOT, AND, and reversible XOR embeddings can be realized with tunable noise, metastability, and coarse-graining. Within those laboratories we ask four concrete questions. First, which binary coarse variables close most robustly under the induced dynamics? Second, how do familiar gates behave as noise, staging barriers, and timescale stress are varied? Third, what changes when a reversible embedding is coarse-grained to an apparently simpler but information-losing gate? Fourth, can stable bits and gates be recovered from transition structure without declaring them by hand in advance?

This paper answers all four. Parity beats the median random binary partition across the full tested grid, making XOR the paper’s clearest instance of a substrate-native logical distinction. NOT, AND, and reversible XOR embeddings exhibit distinct closure regimes rather than a single uniform notion of Boolean success. When a reversible CNOT embedding is viewed through an erased XOR output, closure degrades sharply and unretained input information increases by more than an order-one factor, showing that reversible embedding and erased output are not the same logical object. Finally, a minimal discovery pipeline recovers stable packaged bits and reconstructs a NOT gate directly from transition structure, showing that the framework can find logic rather than merely redescribe it.

**Contributions.** This paper makes three contributions. First, it gives a layer-relative definition of logic in the Six Birds framework: a logic layer is an audited closure of packaged predicates under induced dynamics. Second, it introduces operational diagnostics that distinguish genuine logic layers from projected symbolic descriptions. Third, it demonstrates the framework in controlled finite laboratories, yielding a parity-robustness result, a reversible-versus-erased gate separation, and an unsupervised recovery result for both packaged bits and a NOT gate.

The paper is not a general survey of physical computation, nor a claim that Boolean logic is the only endpoint of coarse-graining. On the contrary, the viewpoint developed here is pluralistic: the logic a layer “has” depends on which closures it can stably support. In particular, complement, negation, and even the availability of binary truth carriers are empirical achievements rather than automatic givens. What we do show is that once closure, packaging, and audit are treated as first-class objects, logic becomes something a substrate can earn, lose, or support only partially.

The remainder of the paper develops this claim in stages. Section 2 formalizes the notion of a logic layer through packaging, definability, and induced closure. Section 3 introduces the diagnostics and the controlled finite laboratories used throughout. Section 4 shows that parity is an unusually robust coarse variable. Section 5 studies gate closure regimes for NOT, CNOT, and AND across noise, staging, and timescale stress. Section 6 compares reversible embedding to erased XOR and quantifies the resulting accounting differences. Section 7 shows that bits and a NOT gate can be recovered directly from transition structure. Section 8 closes by returning to the broader question of what it means for a layer to have a logic at all.

## 2 Logic Layers, Packaging, and Closure

We do not rederive the full Six Birds formalism here. For the present paper only the pieces needed to speak about logic are essential: a lens that fixes what is observable, a packaging step that stabilizes those observables into reusable carriers, an induced dynamics that may or may not close on the packaged variables, and an audit that measures persistence, mismatch, and loss. In Six Birds terms, P5 supplies the package, P2 supplies feasibility semantics, P4 supplies metastability or sector structure, P1 governs whether new effective operators can be introduced, P3 diagnoses route dependence, and P6 keeps the accounting honest. This section isolates those ingredients and turns them into a concrete criterion for what we will call a *logic layer* [5].

### 2.1 Lenses, definability, and packaged predicates

Let  $Z$  denote the microstate space and let

$$f : Z \rightarrow X$$

be a lens or coarse-graining map into a smaller label space  $X$ . The fibers of  $f$  specify which microstates the candidate layer refuses to distinguish. A binary predicate  $h : Z \rightarrow \{0, 1\}$  is *definable* at that lens when it is constant on fibers, equivalently when there exists a macro-predicate  $\tilde{h} : X \rightarrow \{0, 1\}$  such that

$$h = \tilde{h} \circ f.$$

We write  $\Sigma_f$  for the family of events or predicates definable from  $f$ .  $\Sigma_f$  is the expressive content of the layer: if a distinction is not definable there, then it is not available as a logical statement at that layer [6–8].

Packaging is the step that turns a mere definable distinction into a reusable symbol carrier. In the binary case, a package is a map

$$\Pi : Z \rightarrow \{0, 1\},$$

with many microstates assigned to the same label because they are treated as equivalent for the purposes of the layer. For a definable event  $A \in \Sigma_f$ , the associated packaged predicate is simply the indicator

$$\Pi_A(z) = \mathbf{1}[z \in A].$$

This is the point at which P5 becomes indispensable: truth values are not primitive objects but compressed labels of microstates. They remain meaningful only when P4 supplies enough staging, metastability, or sector structure for those labels to persist across the timescale on which the layer is supposed to operate [8, 9, 16].

## 2.2 Two complementary views of logic

The first view is semantic. In the *feasibility view*, propositions are definable feasible sets and logical connectives are stable closure operations on those sets. Once a universe of admissible states is fixed, the familiar connectives are forced:

$$A \wedge B = A \cap B, \quad A \vee B = A \cup B, \quad \neg A = Z \setminus A,$$

and, for parity,

$$A \oplus B = (A \setminus B) \cup (B \setminus A).$$

On this view, logic is not an extra ontology layered on top of physics. It is the algebra of feasible distinctions that a lens can express and a substrate can stabilize. P2 appears here as the semantics of admissibility: propositions are gates on what states are allowed, and contradictions are simply empty feasible sets.

The second view is operational. In the *operator view*, a packaged bit matters only if the substrate supports induced updates on it. Let  $F : Z \rightarrow Z$  denote a micro-update, or more generally let  $P$  be a Markov kernel on  $Z$ . A macro-operator  $g$  is induced on a package  $\Pi$  when the packaged description closes, at least approximately, under the underlying dynamics:

$$\Pi(F(z)) \approx g(\Pi(z)).$$

At the level of distributions, if  $Q_f$  denotes pushforward through the lens and  $U_f$  denotes a canonical lift back to packaged microdescriptions, the associated empirical endomap is

$$E_{\tau, f}(\mu) = U_f(Q_f(\mu P^\tau)).$$

This is the reduce–evolve–repackage route. In finite-state language, the same descent problem is the one studied by lumpability theory for Markov chains [9, 13–15]. A genuine macro law requires it

to agree, up to a controlled defect, with the route that reasons directly at the packaged level. P1 enters here because effective operators sometimes need to be rewritten or promoted to close cleanly. P3 enters here because route mismatch is precisely the sign that ordering, hidden protocol state, or evaluation schedule is still leaking through the coarse description.

The two views are inseparable in practice. Feasibility without operator closure gives semantics that cannot be executed. Operator closure without feasibility semantics gives a convenient signal-processing description but not yet a logic. A logic layer appears only when the same packaged distinctions support both views at once.

**Definition 1** (Logic layer). Fix a microstate space  $Z$ , a lens  $f : Z \rightarrow X$ , and an evaluation window  $\tau$ . A *logic layer* at  $(f, \tau)$  consists of a nontrivial family  $\mathcal{B} \subseteq \Sigma_f$  of binary packaged predicates together with induced macro-updates such that:

1. **Stability.** Each  $b \in \mathcal{B}$  is metastable on the timescale  $\tau$ .
2. **Closure.** The relevant micro-updates descend to macro-operators on the packaged variables with small closure defect.
3. **Feasibility semantics.** Each  $b \in \mathcal{B}$  is the indicator of a stable feasible set, and the connective operations required at the layer remain definable there.
4. **Auditability.** Persistence, route agreement, and information loss are measurable by explicit diagnostics.

When these conditions fail, a symbolic description may still be convenient, but it does not constitute a closed logic layer in the sense used in this paper.

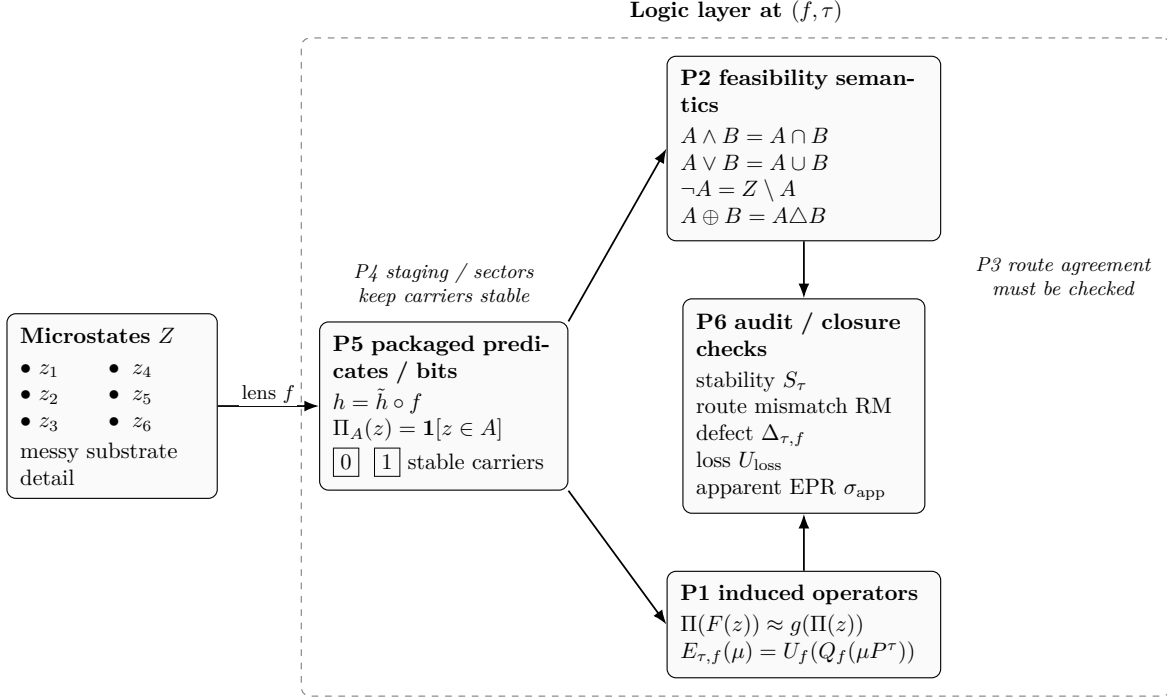
Figure 1 summarizes the architecture of the notion introduced in Definition 1: microstates enter the candidate logic layer only through packaging, and the resulting predicates count as logical only when both their feasibility semantics and their induced operators survive explicit audit.

### 2.3 Why the criterion is layer-relative

Definition 1 is intentionally stricter than merely being able to label states by zero and one. A candidate bit must remain stable long enough to matter, must support a descended operator calculus, and must survive audit. This is why the six primitives are complementary rather than redundant. P5 packages the carrier; P4 keeps it from dissolving immediately; P2 makes it semantically meaningful; P1 determines whether an effective operator vocabulary can be stabilized; P3 detects route dependence and protocol residue; P6 determines what must be paid to maintain the whole arrangement.

The criterion is also intentionally broader than classical Booleanity. Complement and negation appear naturally when the relevant complements remain definable and stable at the lens. But nothing in Definition 1 forces every layer to have those complements available at low cost. For that reason we speak of a *logic layer* rather than a *Boolean layer*. The finite laboratories studied in this paper are classical-looking enough to realize NOT, AND, and XOR cleanly, but the framework itself is meant to diagnose which closures a layer actually supports rather than to decree them in advance [7, 17, 18].

This perspective clarifies what the rest of the paper will measure. Section 3 turns the four clauses of Definition 1 into concrete observables: predicate stability measures the first clause, route mismatch and closure defect probe the second, feasible-set and gate semantics ground the third, and unretained input information and entropy-production proxies support the fourth. The results sections then use those diagnostics to show that parity is unusually robust as a packaged distinction,



**Figure 1.** A logic layer is an audited closure of packaged predicates. A lens  $f$  and a packaging map compress microstates into stable binary carriers. Those carriers support two complementary views at once: feasibility semantics on definable sets and induced macro-operators on packaged symbols. The layer counts as logical only when the resulting closure survives explicit audit through stability, route mismatch, closure defect, and unretained input information diagnostics.

that familiar gates occupy different closure regimes, and that reversible embedding and erased output are distinct logical objects rather than notational variants.

### 3 Diagnostics and Methods

Definition 1 gives four requirements for a logic layer: stability, closure, feasibility semantics, and auditability. This section turns those four requirements into measurable quantities and summarizes the controlled laboratories in which they are evaluated. The important methodological point is that the paper is not reporting live computations from the manuscript source. All numerical values reported in Sections 4–7 are taken from the frozen result bundle under `results/final_claims/` and the source artifacts it summarizes. The manuscript does not recompute those values during drafting; it only defines the diagnostics and interprets the frozen outputs. Methodologically, this follows the broader Six Birds insistence that closure claims be paired with explicit audit rather than inferred from symbolic convenience alone [11, 12].

#### 3.1 Stability and closure diagnostics

The first diagnostic is *predicate stability*. For a binary packaged predicate  $b : Z \rightarrow \{0, 1\}$ , a Markov kernel  $P$  on  $Z$ , an evaluation horizon  $\tau$ , and an initial micro-distribution  $\mu$ , we measure

$$S_\tau(b; \mu) = \sum_{i,j} \mu_i(P^\tau)_{ij} \mathbf{1}[b(i) = b(j)].$$

Large  $S_\tau$  means that the packaged distinction survives the timescale on which the layer is supposed to operate. In the experiments below, this quantity is used both for stable input bits in the gate laboratories and for candidate partitions in the discovery pipeline.

The second diagnostic is *route mismatch* (RM), which asks whether states placed in the same coarse cell really have the same coarse transition behavior. For a lens  $f : Z \rightarrow X$ , define the coarse transition profile of a microstate  $i$  by

$$R_\tau^f(i, x) = \sum_{j:f(j)=x} (P^\tau)_{ij}.$$

If  $B_x = f^{-1}(x)$  is the fiber of  $x$  and  $\bar{\omega}_i^{(x)}$  are normalized weights within that fiber, then the induced macro row is

$$K_{\tau,f}(x, \cdot) = \sum_{i \in B_x} \bar{\omega}_i^{(x)} R_\tau^f(i, \cdot),$$

and the route mismatch is the weighted within-fiber discrepancy

$$\text{RM}_\tau(f) = \sum_x \sum_{i \in B_x} \bar{\omega}_i^{(x)} \|R_\tau^f(i, \cdot) - K_{\tau,f}(x, \cdot)\|_1.$$

RM vanishes when coarse-grained transition behavior is fiberwise consistent. It grows when the lens collapses states whose macro futures differ. In the parity experiment we compare RM across parity, AND, and random binary lenses; in the phase-diagram and reversible-versus-erased experiments we also use stationary weighting to prevent rare states from dominating the metric [9, 13, 14].

A third closure diagnostic compares two different reduction routes. Given pushforward  $Q_f$  through the lens and a canonical lift  $U_f$  back to a packaged microdescription, the *closure defect*, implemented here as a distribution commutation defect, is

$$\Delta_{\tau,f}(\mu) = \|Q_f(\mu P^\tau) - Q_f(U_f(Q_f(\mu))P^\tau)\|_1.$$

The first term evolves microscopically and then reduces; the second reduces, repackages, evolves, and reduces again. Small defect means that the packaged description is genuinely closed under the dynamics. Large defect indicates that hidden microstructure or protocol residue is still leaking through the purported logic layer [8, 10, 15].

Taken together,  $S_\tau$ , RM, and  $\Delta_{\tau,f}$  operationalize the first two clauses of Definition 1. Stability measures persistence. RM and closure defect measure whether closure has actually been achieved rather than merely asserted.

### 3.2 Gate-channel and information diagnostics

For gate-like behavior we work with induced input–output channels. Let  $f_{\text{in}}$  be a current-state input labeling and  $f_{\text{out}}$  a future-state output labeling. For each input label  $u$ , we initialize uniformly over the microstates satisfying  $f_{\text{in}} = u$ , evolve for  $\tau$  steps, and read the output label  $y$ . This defines a channel

$$C_\tau(y | u).$$

All gate metrics in the paper are derived from this induced channel, using a uniform prior over input patterns unless explicitly noted otherwise.

If  $t(u)$  denotes the intended truth-table output for input  $u$ , then the *truth error* is

$$\text{err}_{\text{truth}} = \frac{1}{|U|} \sum_{u \in U} [1 - C_\tau(t(u) | u)].$$

This compares the induced channel to a chosen logical target. To measure how deterministic the channel is even without an externally supplied truth table, we also use the *induced error*

$$\text{err}_{\text{induced}} = \frac{1}{|U|} \sum_{u \in U} \left( 1 - \max_y C_\tau(y | u) \right),$$

which vanishes only when the best-fit deterministic output is certain for every input class.

The corresponding uncertainty measure is the conditional entropy

$$H(\text{out} | \text{in}) = -\frac{1}{|U|} \sum_{u \in U} \sum_y C_\tau(y | u) \log_2 C_\tau(y | u),$$

with the usual convention that  $0 \log 0 = 0$ . This is the main entropy quantity reported in the phase-diagram results. It tracks the same phenomenon as induced error, but without collapsing uncertainty to an argmax summary.

For the accounting comparisons we also use standard channel-information quantities. With the same uniform input prior,

$$I(\text{in}; \text{out}) = H(\text{out}) - H(\text{out} | \text{in}),$$

and we define the *unretained input information* by

$$U_{\text{loss}} = H(\text{in}) - I(\text{in}; \text{out}).$$

This is the primary information-loss proxy in the reversible-versus-erased analysis. It measures how much input information is not recoverable from the reported output channel. We also track the simpler entropy-drop proxy

$$\Delta H = H(\text{in}) - H(\text{out}),$$

but the main comparison later in the paper is expressed in terms of  $U_{\text{loss}}$ , because reversible embeddings can preserve output entropy while still hiding or discarding part of the input description. The reversible-versus-erased comparison in Section 6 is framed by the classical literature on irreversibility, reversible computation, and the thermodynamics of information [1, 2, 4, 19].

Finally, to connect information processing with embodiment, we compute entropy-production rates. For a stationary distribution  $\pi$  of the micro-kernel  $P$ , the micro-level entropy production rate is

$$\sigma(P) = \sum_{i,j} \pi_i P_{ij} \log \frac{\pi_i P_{ij}}{\pi_j P_{ji}},$$

measured in nats per step. The corresponding *apparent* coarse-grained entropy production is obtained by inducing a macro-kernel  $K_{\tau,f}$  on a lens  $f$  and then evaluating the same expression on that macro-kernel:

$$\sigma_{\text{app}}(f, \tau) = \sigma(K_{\tau,f}).$$

Micro and apparent EPR coincide on the identity lens and can diverge under coarse-graining. In this paper they serve as audit variables rather than as stand-alone thermodynamic claims. All Shannon quantities are reported in bits; all entropy-production quantities are reported in nats per step [20, 21].

### 3.3 Finite logic laboratories

The experiments are carried out in finite Markov laboratories built from a macro-register space  $X$  together with a micro-degeneracy factor  $d$ , so that

$$Z = X \times \{1, \dots, d\}.$$

A single tick of the dynamics first applies a deterministic macro-update, then injects controlled output noise, then applies memory flips to designated stable bits, and finally redistributes probability uniformly across the  $d$  microstates of the destination macrostate. This last step enforces rapid within-fiber mixing and makes packaging explicit rather than implicit.

Three gate laboratories are used throughout. The NOT laboratory carries registers  $(a, c)$ , where  $a$  is a stable input bit and the gate sets  $c' = 1 - a$ . The AND laboratory carries  $(a, b, c)$ , where  $a$  and  $b$  are stable input bits and the gate sets  $c' = a \wedge b$ . The reversible XOR embedding is implemented as a CNOT laboratory with registers  $(a, b)$ , where  $a$  is the stable control and the target updates as  $b' = a \oplus b$ . Output noise is controlled by  $p_{\text{gate}}$ . Stable-input corruption is controlled either directly by  $p_{\text{mem}}$  or indirectly through a barrier parameter via

$$p_{\text{mem}} = \text{base\_mem\_noise} e^{-\text{barrier}}.$$

The phase-diagram sweeps vary  $p_{\text{gate}}$ , barrier, and  $\tau$ . For CNOT, the  $\tau > 1$  cases are interpreted as closure stress tests rather than as repeated evaluation of a single-step static truth table.

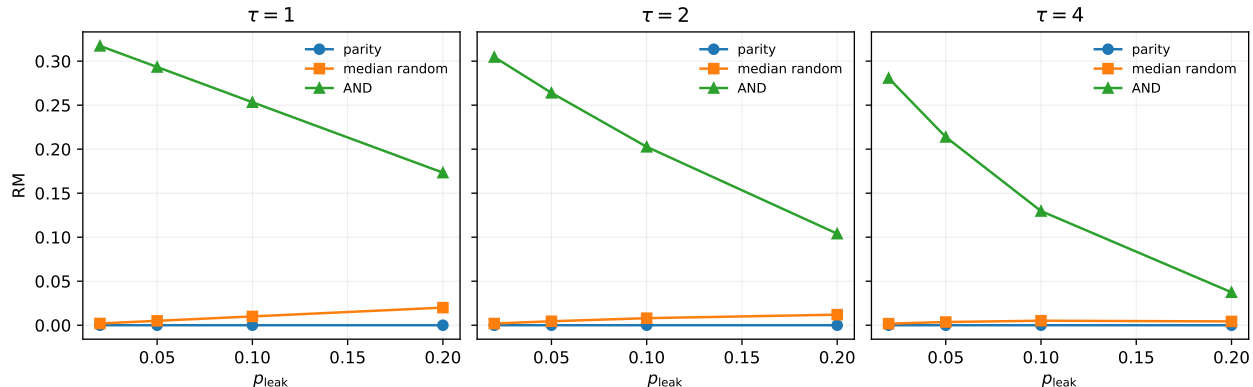
A fourth laboratory, the parity-sector model, is not a gate laboratory in the same sense. It is a two-bit system  $(a, b)$  with controlled leakage between the even-parity and odd-parity sectors. Its purpose is to ask which binary coarse variables remain robust as noise and timescale vary. This is the laboratory used for the parity-versus-random-lens comparison.

### 3.4 Discovery pipeline

The final part of the methods asks whether stable bits and gates can be recovered from transition structure without declaring them by hand. For input-bit discovery we compute the second eigenvector of a reversibilized, symmetrized  $\tau$ -step operator and sweep threshold cuts through that vector to generate binary partitions [16, 22]. Each candidate partition is scored by metastability minus RM, so the preferred candidate is both persistent and closure-friendly. In the NOT and parity laboratories, the best-scoring recovered partitions agree with the intended packaged bits at the level reported later in Section 7.

Output-bit discovery is treated separately in the NOT gate reconstruction. There we first build bidirectional behavioral classes from outgoing rows and incoming columns of the kernel, enumerate binary partitions of those classes, and rank the candidates by future information gain over present redundancy [10, 23, 24]. Intuitively, a good output partition should be informative about future behavior while not simply restating the current input partition. Once input and output partitions are selected, we form a balanced transition sample, infer the best-fit one-input gate, and record its error and conditional entropy. The recovered NOT result reported later in the paper is frozen in `results/final_claims/table_gate_discovery.csv` and the underlying source artifacts listed in the claim ledger.

The net effect of this methodology is that every clause of Definition 1 is tied to an explicit measurement. Stability is probed by  $S_\tau$ , closure by RM and closure defect, gate semantics by channel error and entropy, and embodiment by unretained input information and entropy-production audits. The remaining sections simply read those diagnostics off the frozen result bundle for four different empirical questions: which coarse variables are robust, which gates close under stress, what changes under erasure, and whether bits and gates can be discovered rather than imposed.



**Figure 2.** Route mismatch in the parity-sector laboratory for three candidate binary coarse variables: the parity lens  $a \oplus b$ , the AND lens  $a \wedge b$ , and the median balanced random binary partition. Each panel fixes  $\tau$  and scans leakage  $p_{\text{leak}}$ . Across the full tested grid, the parity lens remains essentially closure-aligned while the AND and median-random alternatives show systematically larger mismatch. The figure is rendered directly from `results/final_claims/figure_parity_robustness.csv`.

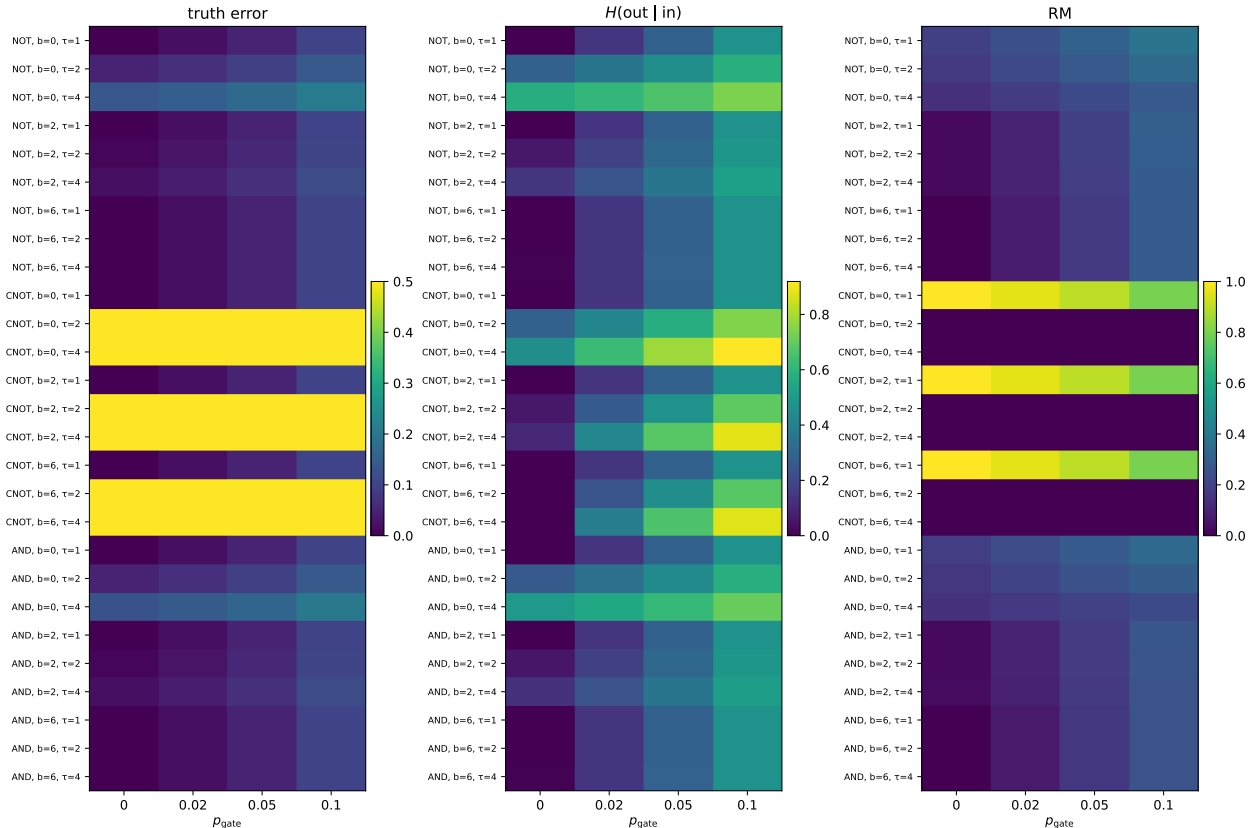
## 4 Parity Robustness

Figure 2 is the empirical anchor for the title claim of the paper. The parity-sector laboratory is designed around controlled leakage between the even- and odd-parity sectors of a two-bit system, so the natural candidate coarse variable is the parity bit  $a \oplus b$ . We compare that lens against two alternatives defined on the same underlying four macrostates: the familiar Boolean predicate  $a \wedge b$ , and the median of a family of balanced random binary partitions. The comparison is intentionally closure-based rather than purely predictive. The plotted quantity is route mismatch (RM), so smaller values mean that the proposed binary variable is better aligned with the induced coarse dynamics.

The main result is absolute across the tested grid. In the frozen bundle, the parity lens beats the median random binary partition at every one of the 12  $(p_{\text{leak}}, \tau)$  grid points, giving a win rate of 1. The representative values recorded in the claim ledger make the point sharply. At  $p_{\text{leak}} = 0.05$  and  $\tau = 1$ , the parity RM is  $1.11 \times 10^{-16}$ , whereas the median random RM is 0.005. At  $p_{\text{leak}} = 0.20$  and  $\tau = 1$ , the parity RM is 0, whereas the median random RM is 0.020. On the same two slices, the AND lens is far less closure-aligned, with RM 0.293 at  $p_{\text{leak}} = 0.05, \tau = 1$  and RM 0.173 at  $p_{\text{leak}} = 0.20, \tau = 1$ . Figure 2 shows that this is not an isolated comparison point but the qualitative pattern across all three timescales.

What makes the result conceptually useful is that parity remains closure-aligned even when it becomes noisy as a signal. The parity error grows from 0.05 at  $p_{\text{leak}} = 0.05, \tau = 1$  to 0.20 at  $p_{\text{leak}} = 0.20, \tau = 1$ , and at longer horizons it grows further, reaching 0.4352 at  $p_{\text{leak}} = 0.20, \tau = 4$ . Stability degrades in the same direction, from 0.95 at  $p_{\text{leak}} = 0.05, \tau = 1$  to 0.5648 at  $p_{\text{leak}} = 0.20, \tau = 4$ . Yet the parity RM remains numerically near zero throughout the grid. This is exactly the distinction the framework is designed to expose. A coarse variable can become less predictable under leakage and still remain the correct closure variable for the layer. Robustness here does not mean error-free transmission; it means that the packaged distinction continues to define the right macro partition for the induced dynamics.

The contrast with AND makes the structural point. AND is not a random or adversarial choice: it is a perfectly familiar Boolean predicate on the same underlying bits. But in the parity-sector laboratory it cuts across the dominant sector structure rather than aligning with it. Its RM remains orders of magnitude above the parity RM across the grid, and above the median random partition



**Figure 3.** Gate-closure phase portrait rendered directly from `results/final_claims/figure_gate_phase.csv`. Each row corresponds to a specific gate, barrier, and timescale  $\tau$ , while the columns scan output noise  $p_{\text{gate}}$ . The three panels report truth error, conditional entropy, and route mismatch on the output-bit lens. The figure is a frozen export of the full 108-row phase bundle.

at every reported point. The lesson is that semantically familiar predicates are not automatically closure-friendly. A substrate does not inherit its logic from our preferred vocabulary; it selects the distinctions that its own dynamics can stabilize.

This is why XOR appears in the title. In the present laboratory, parity is the earliest and cleanest packaged invariant: it compresses the microstate space in the same way the dynamics itself organizes it. The more modest point is that variables aligned with a laboratory’s dominant invariant or sector structure can be especially natural closure variables [8–10]. The next section asks how far that observation extends once we move from a single robust coarse variable to full gate families under noise, staging, and timescale stress (Section 5).

## 5 Gate Closure Phase Diagrams

Figure 3 shifts the discussion from a single robust coarse variable to full gate families under stress. The point of the figure is not that logic is either present or absent, but that closure occupies regimes. The frozen phase bundle spans a 108-row grid over gate family, staging barrier, evaluation horizon  $\tau$ , and output-noise level  $p_{\text{gate}}$ . For each condition it records three diagnostics on the same frozen data asset: truth error relative to the intended one-step gate, conditional entropy  $H(\text{out} | \text{in})$ , and route mismatch on the output-bit lens. Taken together, these panels show that familiar truth tables do not all fail in the same way, and that different diagnostics reveal different modes of nonclosure.

At the coarsest level, the frozen bundle shows a clean monotonic trend in truth error with increasing output noise and no anomaly rows. Averaging over barrier and  $\tau$ , the NOT truth error rises from 0.0236 at  $p_{\text{gate}} = 0$  to 0.1189 at  $p_{\text{gate}} = 0.1$ . AND rises from 0.0225 to 0.1180. CNOT rises from 0.3333 to 0.3667. These averages are useful as a synopsis, but Figure 3 also makes clear why averages alone are insufficient: barrier and horizon matter almost as much as gate noise, and the three diagnostics do not move together in the same way for all gates.

For NOT and AND, the intended gate is recovered cleanly when staging is strong and the evaluation window is short. At barrier 6,  $\tau = 1$ , and  $p_{\text{gate}} = 0$ , both gates are exact at the channel level in the frozen bundle. NOT records  $\text{err}_{\text{truth}} = 0$ ,  $H(\text{out} | \text{in}) = 0$ , and  $\text{RM} = 0.000496$ . AND records  $\text{err}_{\text{truth}} = 0$ ,  $H(\text{out} | \text{in}) = 0$ , and  $\text{RM} = 0.000496$ . Once the horizon is extended under weak staging, however, closure degrades even when output noise is absent because the stable input bits themselves begin to wander. At barrier 0,  $\tau = 4$ , and  $p_{\text{gate}} = 0$ , NOT rises to error 0.1355, entropy 0.5723, and  $\text{RM}$  0.1385; AND rises to error 0.1263, entropy 0.4974, and  $\text{RM}$  0.1419. The symbolic form of the gate has not changed, but the regime in which it is being evaluated has: carrier stability and closure have become the limiting factors.

CNOT behaves differently because the plotted closure lens is intentionally incomplete. At  $\tau = 1$  and  $p_{\text{gate}} = 0$ , the one-step reversible XOR embedding is exact as a gate channel: for barrier 6, the frozen bundle records  $\text{err}_{\text{truth}} = 0$  and  $H(\text{out} | \text{in}) = 0$ . But the  $\text{RM}$  of the output-bit lens is simultaneously maximal,  $\text{RM} = 1$ . This is not a contradiction. It is evidence that the output bit alone is not a closed state description for CNOT because its future depends on the retained control bit. The reversible two-bit gate is orderly at the full carrier level, while the one-bit output lens by itself is non-Markovian. CNOT therefore separates channel fidelity from closure in a way that NOT and AND do not.

That separation becomes even sharper at longer horizons. For CNOT, the  $\tau > 1$  rows are best read as closure stress tests rather than as repeated grading of the same static truth table. A second noiseless CNOT application toggles the target again, so the one-step XOR truth table is no longer the correct target at  $\tau = 2$ . This is exactly what appears in the frozen bundle. At barrier 6,  $\tau = 2$ , and  $p_{\text{gate}} = 0$ , the truth error jumps to 0.5, yet the induced error is only 0.000124, the conditional entropy is 0.0018, and the  $\text{RM}$  collapses to  $5.55 \times 10^{-17}$ . At  $\tau = 4$ , the same pattern persists, with truth error 0.5, induced error 0.000248, entropy 0.0033, and  $\text{RM}$   $2.22 \times 10^{-16}$ . The message is not that CNOT ceases to be structured. The message is that the relevant macro-operator has changed with the protocol horizon. Evaluating  $\tau > 1$  as a closure stress test rather than as a truth-table scorecard is therefore essential.

The figure as a whole should be read as a phase portrait of embodied logic. Output noise raises channel error in the expected way, but staging barriers and evaluation horizons determine whether that noise is the dominant failure mode or merely one contributor among several. For NOT and AND, the central transition is from a short-horizon, high-barrier regime in which the gate practically closes, to a longer-horizon, weak-staging regime in which memory corruption drives both entropy and mismatch upward. For CNOT, the key lesson is different: closure depends on retaining the right carrier set. A reversible two-bit gate can be channel-perfect at one horizon while its one-bit output lens remains maximally unclosed.

This regime-based reading sets up the accounting comparison in the next section. The phase portrait already suggests that reversible embedding and erased output are not equivalent descriptions of the same logical object. Section 6 makes that distinction explicit by comparing the retained CNOT view and the erased XOR view inside the same underlying laboratory.

**Table 1.** Three views of the same CNOT laboratory: the full microstate view, the retained four-state CNOT macro view, and the XOR-erased one-bit output view. The table reports closure defect, route mismatch, entropy-production rate, and channel-information quantities, showing how forgetting the retained control bit changes both closure and retained information. Rendered directly from `results/final_claims/table_reversible_vs_erased.csv`.

<i>Closure and audit metrics</i>							
View	States	Defect	RM	EPR			
CNOT micro	12	0	0	$2.72 \times 10^{-17}$			
CNOT macro	4	$3.47 \times 10^{-18}$	$1.70 \times 10^{-16}$	0			
XOR erased macro	2	0.96	0.96	0			
<i>Channel-information metrics</i>							
View	$H_{\text{in}}$	$H_{\text{out}}$	$H(\text{out}   \text{in})$	$I(\text{in}; \text{out})$	$\Delta H$	$U_{\text{loss}}$	
CNOT micro	2	2	0.141	1.859	0	0.141	
CNOT macro	2	2	0.141	1.859	0	0.141	
XOR erased macro	2	1	0.141	0.859	1	1.141	

## 6 Reversible Embedding versus Erased XOR

Table 1 isolates the accounting point of the paper in its cleanest form. All three rows come from the same underlying CNOT laboratory at the same noise setting. What changes is not the substrate but the package presented to the layer. The micro row keeps the full 12-state laboratory. The retained macro row packages those microstates into the full four logical CNOT macrostates  $(a, b)$ . The XOR-erased row keeps only the one-bit output lens and forgets the retained control bit. In Six Birds terms, the split happens at P5: retaining or discarding side information changes the carrier of the layer. P6 then records the consequences.

The first comparison to notice is that the micro and retained macro views are effectively the same logical object at different levels of description. The micro view has closure defect 0, RM 0, and a numerically tiny entropy-production rate  $2.72 \times 10^{-17}$ . The retained macro view has closure defect  $3.47 \times 10^{-18}$ , RM  $1.70 \times 10^{-16}$ , and apparent EPR 0. More importantly, their channel information measures agree at the logical level: both have  $H_{\text{in}} = 2$ ,  $H_{\text{out}} = 2$ ,  $H(\text{out} | \text{in}) = 0.141$ ,  $I(\text{in}; \text{out}) = 1.859$ , and  $U_{\text{loss}} = 0.141$ . The retained macro row is therefore not an information-poor shadow of the micro description. It is the same reversible embedding expressed at the carrier level that actually closes.

The erased XOR row is qualitatively different. Once the retained control bit is forgotten, the closure defect jumps to 0.96 and the RM jumps to 0.96. The frozen claims bundle records these as deltas of 0.96 and 0.96 relative to the retained macro view. At the same time, the channel information profile changes from a two-bit reversible embedding to a many-to-one reported output:  $H_{\text{out}}$  drops from 2 to 1,  $I(\text{in}; \text{out})$  drops from 1.859 to 0.859, and  $U_{\text{loss}}$  rises from 0.141 to 1.141. The resulting ratio of unretained input information is 8.07. This is the main accounting result of the section.

Two features of Table 1 are especially instructive. First, the conditional entropy  $H(\text{out} | \text{in})$  remains 0.141 in all three rows. The local noise on the reported output bit has not changed. What changes is how much of the input structure the reported channel can still carry once the control bit is forgotten. This is why the correct comparison is not simply entropy drop but unretained input

**Table 2.** Unsupervised binary partition recovery in the NOT and parity-sector laboratories. The table reports agreement with the intended packaged predicate, candidate-set size, and the top three closure scores. Rendered directly from `results/final_claims/table_partition_discovery.csv`.

Lab	Agree.	Cand.	Score	Meta.	RM	2nd	3rd
NOT	1	5	1.000	1.000	$5.55 \times 10^{-17}$	0.990	0.970
Parity sector	1	5	0.950	0.950	$1.70 \times 10^{-16}$	0.723	0.606

information. Second, the apparent EPR of the two macro views is 0 in both cases. The decisive audit signal here is therefore not a large coarse thermodynamic signature; it is the joint change in closure and retained information. Forgetting side information does not merely add a surcharge to the same logical object. It produces a different logical object.

That distinction is the core P5/P6 lesson. Packaging is not an innocent matter of notation. The retained macro row keeps the full reversible carrier and therefore supports a closed two-bit operator. The erased XOR row reports only a one-bit summary and thereby turns a reversible embedding into a many-to-one channel with large closure failure. What looks superficially like “the same XOR gate viewed more coarsely” is, at the level of the audited layer, a different object with different closure properties and a much larger loss ledger.

The point matters beyond this one example. Many discussions of physical logic treat reversible and irreversible descriptions as though the latter were simply the former with some bookkeeping omitted. Table 1 shows why that is too weak. Omitting the bookkeeping changes the closure status of the layer itself. The next section turns from this comparison of declared views to the inverse problem: whether stable packaged bits and a gate can be recovered directly from transition structure without declaring the relevant logical objects in advance (Section 7).

## 7 Discovering Bits and Gates

The final empirical step reverses the usual implementation story. Up to this point the paper has taken candidate packaged bits and gate lenses seriously enough to ask whether they close. Here we ask whether those bits and a gate can be recovered from transition structure without naming them in advance. The discovery pipeline does not begin with declared symbols such as “input bit,” “parity,” or “NOT.” It begins with the kernel itself: binary partitions are proposed from spectral structure, scored by metastability minus route mismatch, and only then interpreted as candidate packaged predicates. In the NOT reconstruction, candidate output partitions are ranked by future information gain over present redundancy so that a good output bit predicts the future without merely restating the current input partition. Tables 2 and 3 report the frozen outputs of that procedure.

Table 2 shows that unsupervised partition recovery succeeds exactly in both benchmark laboratories. In the NOT laboratory, the best recovered partition matches the packaged bit with agreement 1 over a candidate set of size 5. Its winning score is 1.000, with metastability 1.000 and RM  $5.55 \times 10^{-17}$ . The second and third candidate scores, 0.990 and 0.970, show that the laboratory contains nearby closure-friendly cuts, but the top-ranked partition is the exact packaged distinction. In the parity-sector laboratory, the best recovered partition also has agreement 1 with 5 candidates considered. Its winning score is 0.95, with RM  $1.70 \times 10^{-16}$ , while the second and third scores fall to 0.723 and 0.606. The inverse problem is therefore nontrivial but decisive: from transition structure alone, the intended packaged bits emerge as the top-ranked closure variables.

**Table 3.** Recovered NOT gate from discovered input and output partitions. The table reports the fitted truth table, fitted error, fitted entropy, sample size, and the discovery scores used to rank the recovered carriers. Rendered directly from `results/final_claims/table_gate_discovery.csv`.

Metric	Value	Metric	Value
Gate	NOT	Truth bits	[1,0]
Truth table	[0,1,1,0]	Error	0.0485
Entropy	0.280	Sample size	20000
Input score	1.000	$\Delta I_{\text{out}}$	0.714
$I_{\text{future}}$	0.714	$I_{\text{current}}$	0

Once the input partition has been found, the NOT reconstruction asks a harder question: which binary partition of behavioral classes is the right output bit? Table 3 records the winning answer. The recovered input partition enters the final gate fit with score 1.000. The selected output partition has  $\Delta I_{\text{out}} = 0.714$ ,  $I_{\text{future}} = 0.714$ , and  $I_{\text{current}} = 0$ . In other words, the chosen output bit carries future information while not merely duplicating the current input partition. On a balanced sample of 20000 transitions, the induced gate fit yields truth bits [1,0], equivalently the flattened one-step table [0,1,1,0]. That is precisely the NOT gate. The fitted error is 0.0485 and the conditional entropy is 0.280. The recovery is therefore not a symbolic relabeling of hand-specified registers; it is an induced operator found only after the carriers on which it closes have themselves been discovered.

Taken together, Tables 2 and 3 close the empirical arc of the paper. Earlier sections argued that logic should be diagnosed by stability, closure, and audit rather than assumed from symbolic convenience. The discovery results show that the same diagnostics can also work in reverse: they can first detect which packaged distinctions a substrate supports and then recover the operator induced on them. In that sense, the paper does not merely reinterpret known bits and gates inside the Six Birds vocabulary. It shows that a logic layer can be found from dynamics alone. Section 8 returns to the broader implication: logic is something a layer earns, not something an analyst is entitled to assume.

## 8 Discussion

This paper began with a prior question to the usual implementation story: when does a substrate actually earn a logic layer? The answer developed through Definition 1 and the results of Sections 4–7 is intentionally stricter than being able to label states by 0 and 1. A logic layer requires packaged predicates that persist, induced operators that close on those predicates, and an audit that measures mismatch and loss. In that sense, logic is not primitive. It is earned by closure.

The first empirical lesson is that parity specialness should be read as a measurement, not a mysticism. Figure 2 showed that, in the parity-sector laboratory, the parity lens beats the median random binary partition at every one of the 12 tested grid points, for a win rate of 1.0. The point of that result is not that XOR occupies a privileged metaphysical tier. It is that parity is the packaged distinction most naturally aligned with the sector structure of that laboratory. A substrate can therefore make some logical forms cheap and robust without making all familiar Boolean predicates equally natural. The title of the paper is meant in exactly that empirical sense: to “XOR a stone” is to discover that parity can be the first logical distinction a substrate stabilizes.

The second lesson is that gate closure is a regime phenomenon rather than a binary yes/no verdict. Figure 3 showed that NOT and AND close cleanly in short-horizon, high-barrier regimes and degrade under longer horizons and weaker staging, even when output noise is absent. CNOT

revealed a different kind of nonclosure: a one-step reversible XOR embedding can be channel-perfect while the one-bit output lens remains maximally unclosed. This is why the paper has treated  $\tau > 1$  CNOT rows as closure stress tests rather than as repeated scoring of the same static truth table. Different gates fail in different ways because closure depends not only on noise but also on the carrier set retained by the layer.

The third lesson is that forgetting side information can change the logical object itself. Table 1 did not compare two unrelated systems; it compared three views of the same CNOT laboratory. The retained macro view and the erased XOR view differ in closure defect by 0.96, in route mismatch by 0.96, and in unretained input information by a factor of 8.07. That is not adequately described as “the same gate with a larger thermodynamic bill.” It is a change in what counts as the state of the layer. In Six Birds terms, P5 and P6 are inseparable here: packaging determines the carrier, and the audit reveals that discarding the retained control bit converts a reversible two-bit operator into a many-to-one one-bit report with radically different closure properties [4, 19–21].

The fourth lesson is that the framework is operational in both directions. Tables 2 and 3 showed that the same diagnostics used to test closure can also be used to discover it. The best recovered partitions in the NOT and parity laboratories both achieved agreement 1.0, and the recovered NOT gate had truth table  $[0, 1, 1, 0]$  with fitted error 0.04845. Those results matter because the pipeline begins from transition structure, not from declared symbolic variables. The framework is therefore not merely a reinterpretation of gates we already knew were present. It provides a way to ask, from dynamics alone, which packaged distinctions deserve logical status and which induced operators actually live on them.

Several limitations define the scope of the present paper. First, the laboratories are finite controlled Markov systems chosen for diagnostic clarity rather than for direct substrate realism. Second, the discovery pipeline is restricted to binary partitions and simple gate recovery; it does not yet search higher-arity packages or richer operator vocabularies. Third, the paper is not a full survey of nonclassical logic. It does not attempt to classify all possible failures of complement, negation, or distributivity under coarse-graining. For partition-based routes from classical state spaces to complementary observables and non-Boolean propositional structures, see [7, 18]; for the canonical quantum-logical background, see [17]; for a Six Birds programmatic extension in that direction, see [25]. Those omissions are deliberate. They keep the central claim narrow, falsifiable, and testable: logic layers are measurable closures, not explanatory defaults.

The broader consequence is a pluralistic one. The question is not whether logic is “really there” behind every coarse description. The question is which closures a layer can stably support, at what horizon, and with what audit cost. A layer may support parity while failing conjunction, may support a reversible operator only on a larger carrier, or may admit packaged predicates without a low-cost global complement. The Six Birds viewpoint does not eliminate the diversity of logics; it explains that diversity by tying each logic to the closures a substrate can actually sustain.

If there is a final conclusion to draw from XOR, it is not that parity is magical. It is that substrates tell us which distinctions deserve logical status. Logic is not what a theorist can name at a layer, but what a substrate can stabilize, close, and audit there.

## A Reproducibility, frozen artifacts, and formal support

This appendix records the support structure behind the main text and nothing more. The goal is to make the paper reproducible without turning the appendix into a second results section. The main manuscript reads only frozen outputs; no scientific quantity is recomputed during drafting.

## A.1 Reproducibility and artifact manifest

The repository contains an end-to-end reproduction harness that regenerates the experimental outputs used in the manuscript. The command `make reproduce-all` rebuilds the six result families used in the paper: the parity-versus-AND sweep, the parity-robustness experiment, the gate-phase bundle, the reversible-versus-erased comparison, the discovery smoke run, and the NOT gate-discovery run. The resulting files are recorded in `results/final_manifest.json`, which stores relative paths, file hashes, and sizes for the generated artifacts. This manifest is the repository-level answer to the question “what exact files underwrite the paper?”

## A.2 Frozen claims bundle

The manuscript-level interface to those artifacts is the frozen claims bundle under `results/final_claims/`. The file `claims.json` stores the headline scalar values cited in the paper, while the CSV files store the figure-ready and table-ready data used to render the parity figure, the phase figure, the reversible-versus-erased comparison table, the partition-discovery summary, and the gate-discovery summary. The paper is written against this bundle rather than against live computations. This is why the numerical claims in Sections 4–7 can be checked line by line against a fixed artifact set.

## A.3 Code and data availability

All code used to generate the laboratories, diagnostics, discovery pipeline, frozen claims, and paper assets is contained in the public repository at <https://github.com/ioannist/six-birds-logic>. Core implementations live under `src/emergent_logic/`. Experiment entrypoints live under `experiments/`, configuration files under `configs/`, and reproducibility / validation utilities under `scripts/`. Rendered paper assets are generated from `results/final_claims/` by `scripts/render_paper_assets.py`. No external proprietary dataset is required for the results reported here. The Lean formalization supporting the definability and quotient-dynamics claims lives under `lean/LogicClosure/`.

## A.4 Overflow metric notes

For compactness, the main text reports only the diagnostics needed to support the paper’s central claims. The full frozen phase bundle, however, retains the complete 108-row grid of gate, barrier, horizon, and output-noise conditions in `results/final_claims/figure_gate_phase.csv`. Likewise, the parity-robustness asset preserves the full 12-point grid over leakage and horizon in `results/final_claims/figure_parity_robustness.csv`. Throughout the repository and manuscript, Shannon quantities such as conditional entropy, mutual information, and unretained input information are reported in bits, whereas entropy-production rates are reported in nats per step.

## B Brief note on the Lean formalization

The Lean development is intentionally small and local to the paper’s logical core. It formalizes three claims. First, definable predicates—those constant on the fibers of a lens—are closed under conjunction, disjunction, and negation. Second, if a micro-update respects an equivalence relation, then it induces a well-defined map on the corresponding quotient. Third, these quotient-dynamics ideas can be instantiated in a concrete finite example: a parity lens from Fin 4 to Fin 2 together with a parity-respecting update. The formalization is not intended as a mechanized version of the

full experimental pipeline. Its role is narrower: it certifies the quotient-and-definability skeleton on which the paper’s notion of a logic layer rests.

## References

- [1] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961. doi: 10.1147/rd.53.0183.
- [2] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973. doi: 10.1147/rd.176.0525.
- [3] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3-4):219–253, 1982. doi: 10.1007/BF01857727.
- [4] Charles H. Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21(12):905–940, 1982. doi: 10.1007/BF02084158.
- [5] Ioannis Tsiokos. Six Birds: Foundations of Emergence Calculus, January 2026. URL <https://arxiv.org/abs/2602.00134>. arXiv:2602.00134 [cs.LO].
- [6] Zdzisław Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982. doi: 10.1007/BF01001956.
- [7] David Ellerman. The logic of partitions: Introduction to the dual of the logic of subsets. *The Review of Symbolic Logic*, 3(2):287–350, 2010. doi: 10.1017/S1755020310000018.
- [8] Cosma Rohilla Shalizi and Cristopher Moore. What is a macrostate? subjective observations and objective dynamics. *Foundations of Physics*, 55:2, 2025. doi: 10.1007/s10701-024-00814-1. Article 2.
- [9] Herbert A. Simon and Albert Ando. Aggregation of variables in dynamic systems. *Econometrica*, 29(2):111–138, 1961. doi: 10.2307/1909285.
- [10] L. Barnett and A. K. Seth. Dynamical independence: Discovering emergent macroscopic processes in complex dynamical systems. *Physical Review E*, 108(1):014304, 2023. doi: 10.1103/PhysRevE.108.014304.
- [11] Ioannis Tsiokos. To Count a Stone with Six Birds: A Mathematics is A Theory, January 2026. URL <https://zenodo.org/records/18402004>.
- [12] Ioannis Tsiokos. To Lay a Stone with Six Birds: Finite-State Semantics for Packaging, Directionality, and Coarse-Graining, February 2026. URL <https://www.preprints.org/manuscript/202602.1699>. Preprint, posted February 27, 2026.
- [13] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. Springer-Verlag, New York, 1976. ISBN 9780387901923.
- [14] Peter Buchholz. Exact and ordinary lumpability in finite markov chains. *Journal of Applied Probability*, 31(1):59–75, 1994. doi: 10.2307/3215235.
- [15] Dror Givon, Raz Kupferman, and Andrew Stuart. Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity*, 17(6):R55–R127, 2004. doi: 10.1088/0951-7715/17/6/R01.
- [16] Peter Deuffhard and Marcus Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, 2005. doi: 10.1016/j.laa.2004.10.026.
- [17] Garrett Birkhoff and John von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37(4):823–843, 1936. doi: 10.2307/1968621.

- [18] Harald Atmanspacher and Peter beim Graben. Complementary observables and non-boolean logic outside quantum physics, October 2015. URL <https://arxiv.org/abs/1510.03325>. arXiv:1510.03325 [quant-ph].
- [19] Juan M. R. Parrondo, Jordan M. Horowitz, and Takahiro Sagawa. Thermodynamics of information. *Nature Physics*, 11(2):131–139, 2015. doi: 10.1038/nphys3230.
- [20] Massimiliano Esposito. Stochastic thermodynamics under coarse graining. *Physical Review E*, 85(4):041125, 2012. doi: 10.1103/PhysRevE.85.041125.
- [21] Kyogo Kawaguchi and Yohei Nakayama. Fluctuation theorem for hidden entropy production. *Physical Review E*, 88(2):022147, 2013. doi: 10.1103/PhysRevE.88.022147.
- [22] Cosma Rohilla Shalizi and James P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104(3-4):817–879, 2001. doi: 10.1023/A:1010388907793.
- [23] William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural Computation*, 13(11):2409–2463, 2001. doi: 10.1162/089976601753195969.
- [24] James P. Crutchfield and Karl Young. Inferring statistical complexity. *Physical Review Letters*, 63(2):105–108, 1989. doi: 10.1103/PhysRevLett.63.105.
- [25] Ioannis Tsiokos. A Six-Birds’ Eye View of Quantum Theory: Defining Objects by What We Can Distinguish, Not What We Imagine, February 2026. URL <https://zenodo.org/records/18608833>.