

The Orthogonal Robustness Framework: A Unified Theory of Structure, Survival, and Visibility in Discrete Dynamical Systems

Bit Dynamics Lab

January 2026

Abstract

We present a unified framework for understanding robustness in discrete dynamical systems, synthesizing findings from six independent studies. The central thesis is that robustness is not a single property but manifests along four orthogonal axes: Computational Capability (can it compute?), Self-Maintenance (does it actively persist?), Survival (does it withstand temporal filtering?), and Visibility (is the information readable?). These axes are demonstrably independent—a structure’s position on one axis provides no information about its position on another. This orthogonality explains apparent paradoxes (e.g., why Rule 110 is Turing-complete but not life-like) and yields actionable design principles for engineering robust dynamical systems. We identify hidden state as the universal enabler across all four axes and basin of attraction as the organizing principle that determines survival, stability, and visibility. The framework makes novel predictions testable in systems beyond cellular automata.

1 Introduction

1.1 The Problem of Robustness

Robustness—the ability to maintain function under perturbation—is a central concern across engineering, biology, and theoretical computer science. Yet robustness proves elusive to define. A structure may be robust to noise but fragile to timing changes. A system may persist indefinitely but carry no useful information. Computation may proceed reliably but the computational elements themselves may be transient.

This paper argues that the difficulty arises from treating robustness as a single property. In discrete dynamical systems, robustness manifests along multiple independent dimensions. A complete characterization requires specifying a structure’s position on each axis separately.

1.2 Scope and Sources

This synthesis draws on six papers studying cellular automata under temporal filtering (the “stickiness” mechanism):

Paper 1: UCT — Establishes that universal computation requires ≥ 5 bits of specification complexity under natural encoding, with Control as the critical capability.

Paper 2: Stickiness — Proves that hidden state is necessary and sufficient for Control, the ability to produce context-dependent outcomes.

Paper 3: Self-Maintenance — Demonstrates that 83.7% of non-trivial elementary cellular automata exhibit life-like behavior when hidden state is introduced.

Paper 4: Leakiness — Shows that substrate properties (leakiness, capacity) predict life-like emergence with $R^2 = 0.96$.

Paper 5: Invariants — Establishes that basin of attraction strength determines survival under temporal filtering, with 98% predictive accuracy.

Paper 6: Anti-Resonance — Demonstrates that information visibility depends on carrier-period alignment, orthogonal to survival.

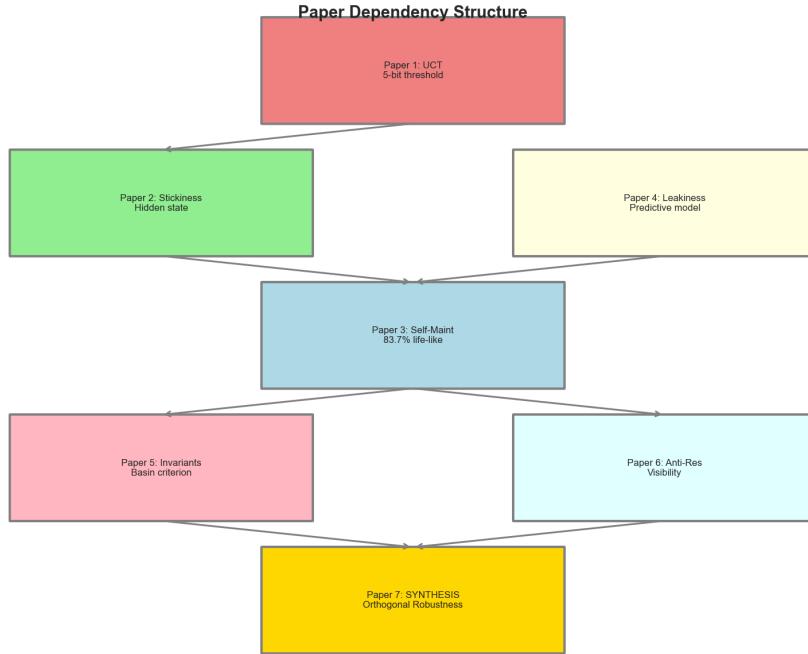


Figure 1: **Paper dependency structure.** The synthesis framework draws on six independent studies. Paper 1 (UCT) establishes the computational foundation; Papers 2-4 develop the self-maintenance framework; Papers 5-6 establish the survival and visibility axes. Paper 7 (this work) synthesizes these findings into the Orthogonal Robustness Framework.

1.3 Central Thesis

Claim 1. *Discrete dynamical systems exhibit four orthogonal axes of robustness:*

| Axis | Question | Criterion |
|------------------|------------------------------|--------------------------------|
| Computation | Can it compute? | ≥ 5 bits + structure |
| Self-Maintenance | Does it actively persist? | Control + Stability + Activity |
| Survival | Does it withstand filtering? | Basin strength |
| Visibility | Is information readable? | Carrier-period alignment |

Table 1: The four axes of robustness in discrete dynamical systems.

These axes are orthogonal in the empirical sense: knowing a structure’s value on one axis provides no information about its value on another. This is not a limitation of measurement but a fundamental property of dynamical systems.

The Orthogonal Robustness Framework

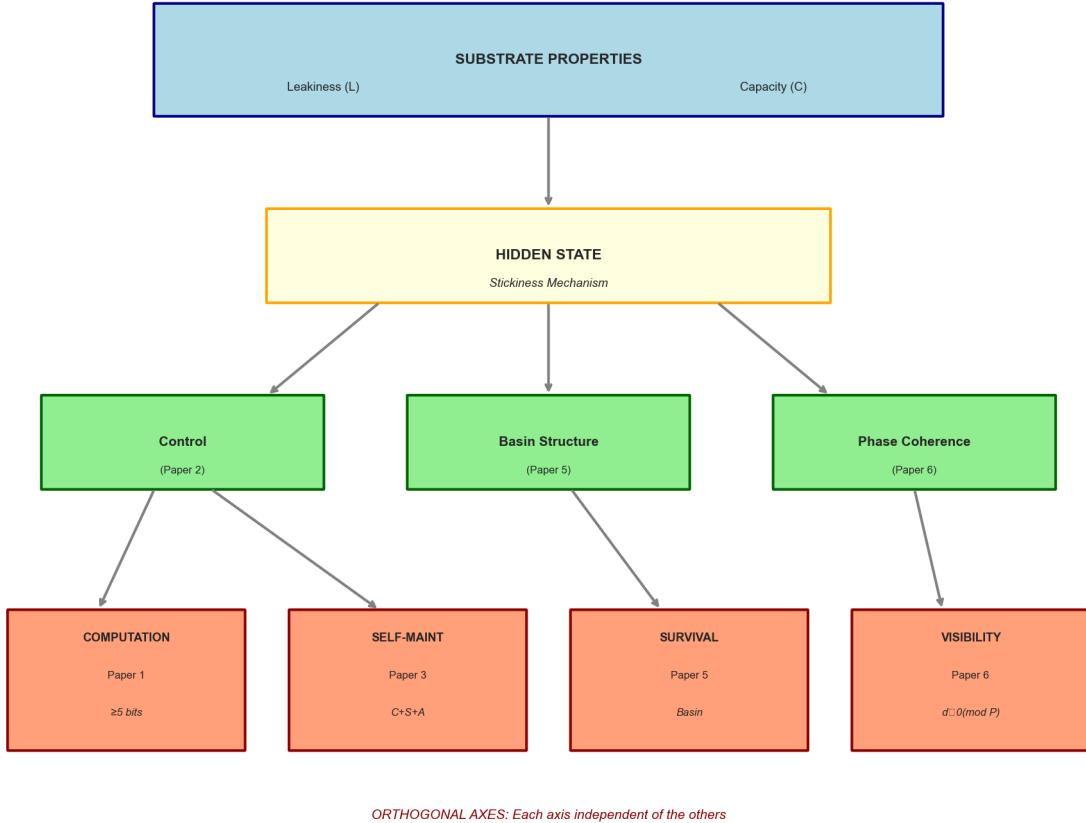


Figure 2: **The Orthogonal Robustness Framework.** Substrate properties (leakiness, capacity) determine the potential for hidden state. Hidden state enables three mechanisms: Control (context-dependent divergence), Basin Structure (attractor geometry), and Phase Coherence (carrier-period alignment). These mechanisms give rise to four orthogonal capabilities: Computation, Self-Maintenance, Survival, and Visibility. Each axis is independent of the others.

2 The Four Axes

2.1 Axis 1: Computational Capability

Question: Can the system perform universal computation?

Criterion: Specification complexity ≥ 5 bits under natural encoding, plus structural conditions.

Paper 1 establishes that universal computation requires a minimum specification complexity of 5 bits, decomposed as:

- **Logic** ≥ 2 bits (Boolean completeness)
- **Memory** ≥ 1 bit (read/write capability)
- **Control** ≥ 1 bit (conditional branching)
- **State** ≥ 1 bit (halting distinction)

The 5-bit threshold is necessary but not sufficient. Structural conditions—asymmetry, collision diversity, signal distinctness—are also required. Rule 110 meets both criteria and achieves Turing-completeness with specification complexity of exactly 5 bits.

The Control Conjecture: The critical 5th bit is Control—the ability to produce context-dependent divergence in interaction outcomes. Paper 2 proves that Control requires hidden state. This connects computational capability to the mechanism layer.

2.2 Axis 2: Self-Maintenance

Question: Does the system exhibit life-like behavior—active persistence through environmental interaction?

Criterion: $\text{Control} > 0$ AND Stability mechanism AND Activity within window.

Papers 2–4 establish the conditions for life-like behavior:

1. **Control** (Paper 2): The system must exhibit context-dependent responses, requiring hidden state.
2. **Stability** (Paper 3): Either absorption (perturbations absorbed without state change, threshold > 0.5) or repair (active return to prior state, threshold > 0.7).
3. **Activity** (Paper 3): Neither frozen nor chaotic; activity ratio between 0.05 and 0.5.

Census Result: When the stickiness mechanism is applied to elementary cellular automata, 83.7% of non-trivial rules (168/194) exhibit life-like behavior.

Predictive Model: Paper 4 demonstrates that life-like emergence can be predicted from substrate properties alone:

- **Leakiness** (Lyapunov-like divergence rate): Phase transition at $L \approx 0.39$. Below this threshold, substrates are “prone” to life-like behavior.
- **Capacity** (compressibility): Threshold at ~ 1.1 bits/cell. Above this threshold, sufficient structure exists to support complex behavior.

The two-axis model achieves $R^2 = 0.96$ in predicting life-like percentage across substrate families.

2.3 Axis 3: Survival

Question: Does the structure persist under temporal filtering?

Criterion: Basin of attraction strength.

Paper 5 establishes that survival under temporal filtering is determined not by computational importance but by basin geometry:

| Structure Type | Basin Type | Survival |
|--------------------|-----------------------|------------------|
| Still life | Point attractor | HIGH (100%) |
| Oscillator | Limit cycle | HIGH (100%) |
| Amplifying pattern | Large basin | HIGH (>95%) |
| Glider/signal | No basin (trajectory) | LOW (<5%) |
| Chaos | Entire space | HIGH (trivially) |

Table 2: Survival classification by basin type. Provenance: Paper 5, Table 2.

The Basin Criterion: “What survives is not what computes, but what repairs.”

Theorem 1 (Intrinsicity). *Basins are intrinsic properties. Proximity to a basin-possessing structure does not confer stability on a structure without a basin.*

A glider passing near an oscillator is not stabilized by the oscillator’s basin. Quantitative result: Basin classification predicts survival with 98% accuracy across tested configurations.

2.4 Axis 4: Visibility

Question: Is encoded information distinguishable from the carrier?

Criterion: Sampling depth d relative to carrier period P .

Paper 6 demonstrates that information visibility depends on the temporal relationship between sampling and carrier oscillation:

- **Anti-resonance:** At depths $d \equiv 0 \pmod{P}$, the carrier phase aligns across sampling windows, minimizing contrast between carrier and defect. Information is hidden.
- **Maximum visibility:** At depths $d \not\equiv 0 \pmod{P}$, phase mismatch creates contrast. Information is readable.

Requirements: Oscillatory carrier (not quiescent background); defect-on-carrier encoding.

Effect Size: $\sim 2.6\times$ difference in Hamming ratio between optimal and anti-resonant depths.

Critical Observation: Visibility is orthogonal to survival. An oscillator at anti-resonant depth survives (strong basin) but carries hidden information. A glider at non-anti-resonant depth is visible but destroyed (no basin).

3 Orthogonality Proofs

The central claim is that the four axes are independent. We provide evidence for three key orthogonalities.

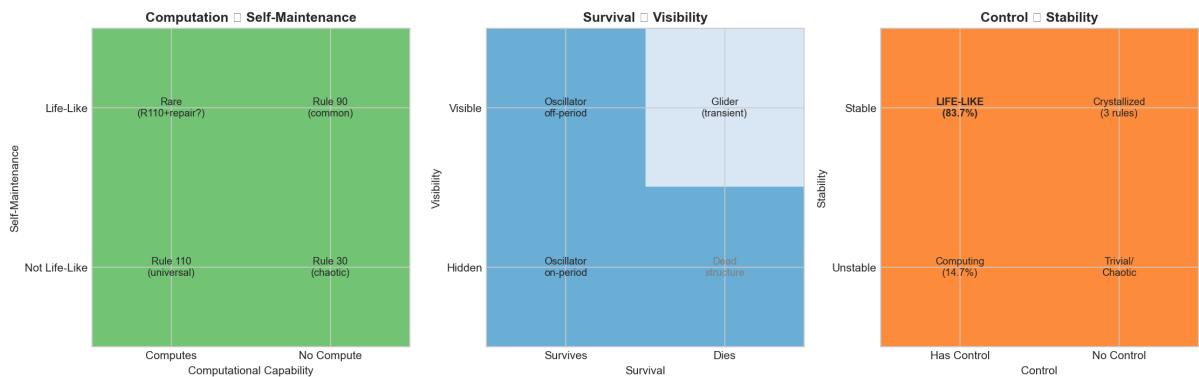


Figure 3: **Orthogonality evidence.** Three 2×2 matrices demonstrating axis independence. *Left:* Computation vs. Self-Maintenance—Rule 110 computes but is not life-like; Rule 90 is life-like but does not compute. *Center:* Survival vs. Visibility—oscillators survive regardless of visibility; gliders are visible but destroyed. *Right:* Control vs. Stability—the life-like classification (83.7%) requires both Control and Stability.

3.1 Orthogonality 1: Computation \perp Self-Maintenance

Claim 2. *Knowing whether a system computes provides no information about whether it self-maintains, and vice versa.*

Evidence:

| Rule | Computes? | Life-like? | Notes |
|------|-----------------------|------------|-----------------|
| 110 | YES (Turing-complete) | NO | Fails stability |
| 90 | NO (linear) | YES | High absorption |
| 30 | NO (chaos) | NO | Neither |
| 184 | NO | YES | Traffic model |

Table 3: Independence of computation and self-maintenance. Provenance: Papers 1 and 3.

Rule 110 is the canonical example: despite being Turing-complete, it is not life-like because it lacks the stability mechanisms required for self-maintenance. Rule 90, conversely, is life-like but cannot compute universally because it is linear (XOR of neighbors) and cannot implement NAND.

Interpretation: Computation requires trajectories through state space—paths that carry information. Self-maintenance requires basins that attract nearby states—regions that absorb perturbation. These are structurally incompatible requirements for the same structure.

3.2 Orthogonality 2: Survival \perp Visibility

Claim 3. *Knowing whether a structure survives provides no information about whether its information content is visible, and vice versa.*

Evidence:

| Configuration | Survives? | Visible? | Mechanism |
|------------------------------|-----------|---------------|-----------------------------------|
| Oscillator, $d \equiv 0$ | YES | NO | Basin strong, anti-resonant |
| Oscillator, $d \not\equiv 0$ | YES | YES | Basin strong, phase contrast |
| Glider, $d \not\equiv 0$ | NO | YES (briefly) | No basin, visible until destroyed |

Table 4: Independence of survival and visibility. Provenance: Papers 5 and 6.

Survival depends on basin geometry (phase space topology) while visibility depends on carrier-period alignment (temporal relationship). These reference different aspects of the system.

3.3 Orthogonality 3: Computation \perp Survival

Claim 4. *Structures that compute tend not to survive, and structures that survive tend not to compute.*

Evidence: Computational structures (gliders, collision debris) have no basins and are destroyed by temporal filtering. Robust structures (still lifes, oscillators) have strong basins but do not traverse state space.

The Computation-Robustness Tradeoff: Computational structures must move through state space to carry information. Trajectories, by definition, do not remain in a basin. Robust structures return to attractors after perturbation. Attractor states, by definition, do not traverse state space.

Resolution: Rule 110’s collision behavior suggests a design pattern: “Compute briefly, then crystallize.” Use fragile computational elements for logic, convert results immediately to robust storage.

4 The Unified Picture

4.1 Hidden State as Universal Enabler

All four axes share a common mechanism: **hidden state**.

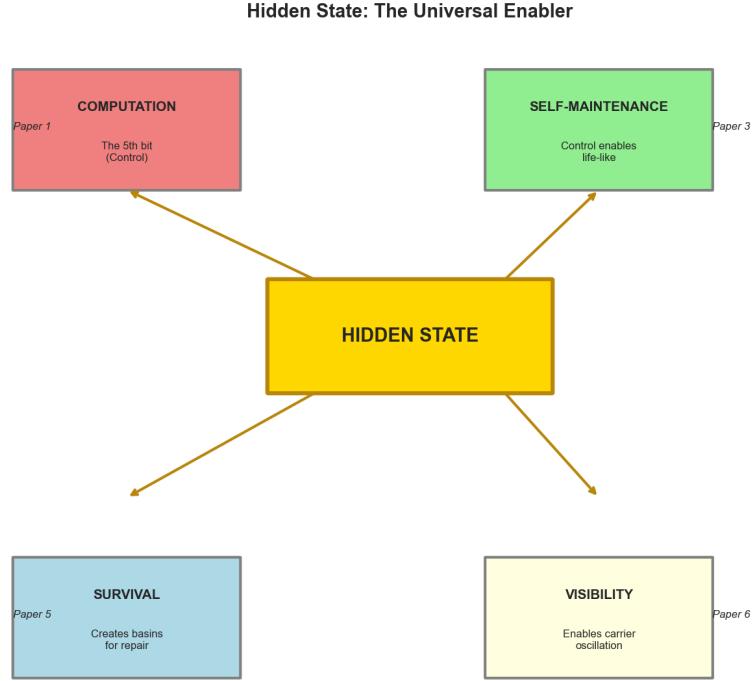


Figure 4: **Hidden state as universal enabler.** Hidden state (created by temporal filtering via confirmation counters) enables all four capabilities. Computation requires hidden state for the 5th bit (Control). Self-Maintenance requires Control as its first criterion. Survival depends on basins created by hidden state dynamics. Visibility requires oscillatory carriers, which are a form of temporal hidden state.

| Axis | Role of Hidden State |
|------------------|---|
| Computation | 5th bit (Control) requires hidden state |
| Self-Maintenance | Control (criterion 1) requires hidden state |
| Survival | Basins created by hidden state dynamics |
| Visibility | Anti-resonance requires oscillatory carrier |

Table 5: Hidden state as universal enabler.

Claim 5 (Synthesis Claim 1). *Hidden state is the universal enabler. Without hidden state, a system is memoryless and has zero Control. All four capabilities become possible only when hidden state is introduced.*

4.2 Basin of Attraction as Organizing Principle

Three of the four axes are organized by basin of attraction:

- **Self-Maintenance:** Stability mechanisms create and exploit basins.

- **Survival:** Basin strength IS the survival criterion.
- **Visibility:** Anti-resonance occurs when carrier basin cycle aligns with sampling.

Claim 6 (Synthesis Claim 2). *Basin of attraction is the organizing principle. The geometry of phase space—where basins exist, how deep they are, what their periods are—determines self-maintenance, survival, and visibility.*

Computation is the exception: it requires trajectories, not basins. This explains the computation-robustness tradeoff.

4.3 Temporal Filtering as Diagnostic Probe

Claim 7 (Synthesis Claim 3). *Temporal filtering is not just a mechanism but a diagnostic. By varying filter depth and observing outcomes, we reveal intrinsic properties of structures that are invisible in unfiltered dynamics.*

4.4 The Four-Dimensional Characterization

Every structure can be characterized by its position in a 4D space: (Computational, Self-Maintaining, Surviving, Visible). This generates 16 possible combinations.

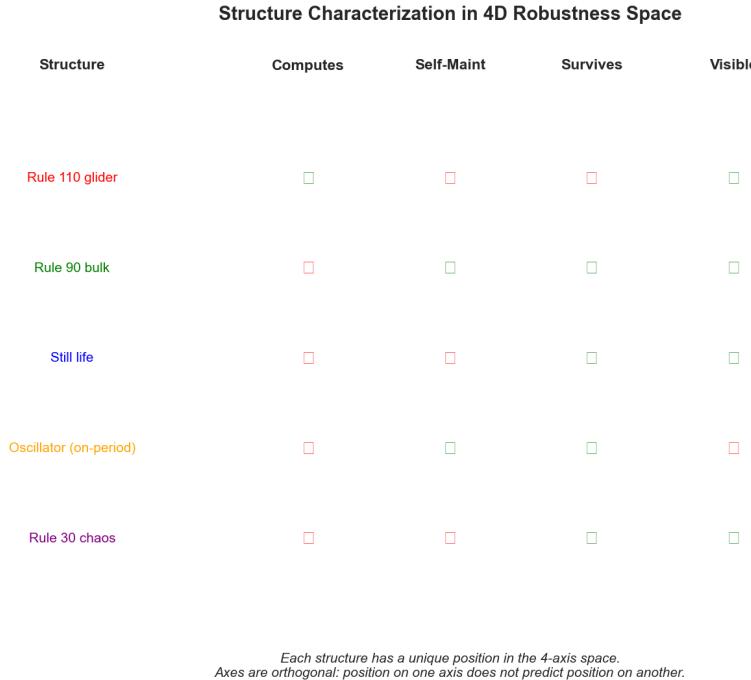


Figure 5: **Structure characterization in 4D robustness space.** Each structure has a unique position determined by its values on the four axes. The axes are orthogonal: position on one axis does not predict position on another. Examples show structures from different quadrants of the space.

| Structure | C | SM | Su | V | Description |
|----------------------|---|----|----|---|----------------------------|
| Rule 110 glider | 1 | 0 | 0 | 1 | Computes, visible, fragile |
| Rule 90 bulk | 0 | 1 | 1 | 1 | Self-maintains, survives |
| Still life | 0 | 0 | 1 | 1 | Survives, visible, passive |
| Anti-res. oscillator | 0 | 1 | 1 | 0 | Self-maintains, hidden |

Table 6: Four-dimensional characterization of structures. C = Computational, SM = Self-Maintaining, Su = Surviving, V = Visible.

5 Design Principles

The framework yields actionable engineering principles:

Principle 1 (Segregate Computation and Persistence). *Since computation requires trajectories and persistence requires basins, do not expect the same structural element to provide both. Use fragile computational elements for logic; convert results immediately to robust storage. “Compute briefly, then crystallize.”*

Principle 2 (Match Sampling to Carrier Period). *To maximize information visibility in oscillatory systems, sample at depths $d \not\equiv 0 \pmod{P}$. Avoid synchronous sampling.*

Principle 3 (Build on Basins). *To achieve self-maintenance and survival, design for basin-possessing structures. Basins are intrinsic and cannot be borrowed from neighbors. Design for repair, not just stability.*

Principle 4 (Pre-Screen Substrates). *Before engineering life-like behavior, characterize the substrate: measure leakiness (phase transition at $L \approx 0.39$) and capacity (threshold ~ 1.1 bits/cell).*

6 Implications and Future Directions

6.1 Generalization Beyond Cellular Automata

The empirical evidence derives from cellular automata. However, the underlying principles should generalize:

- **Basin of attraction** is a general concept from dynamical systems theory.
- **Temporal filtering** is universal—any system subject to noise or latency faces selection for basin-possessing structures.
- **Hidden state** appears wherever memory exists.

Prediction: The framework should apply to reaction-diffusion systems, Boolean gene regulatory networks, and recurrent neural networks. Quantitative thresholds may vary; qualitative structure should persist.

6.2 Biological Implications

The framework predicts that living systems should be organized around basins:

- **Homeostasis** is basin behavior—active return to set points.

- **Limit cycles** (circadian rhythms, cell cycles) are periodic basins.
- **Developmental canalization** is basin geometry.

The computation-robustness tradeoff suggests biological computation should be segregated from biological persistence.

6.3 Open Questions

1. **Formal orthogonality proof:** Mathematical proof of zero correlation across arbitrary systems remains open.
2. **Additional axes:** Energy efficiency, evolvability, and compositionality may constitute additional dimensions.
3. **Continuous systems:** Quantitative calibration needed.
4. **The Control Conjecture:** Remains open but is not load-bearing for the framework.

7 Conclusion

Robustness in discrete dynamical systems is not one thing. It manifests along four orthogonal axes: Computational Capability, Self-Maintenance, Survival, and Visibility. These axes are independent—Rule 110 computes but does not self-maintain; Rule 90 self-maintains but does not compute; oscillators survive but may be invisible; gliders are visible but do not survive.

The framework is unified by three synthesis claims:

- **Hidden state** is the universal enabler across all four axes.
- **Basin of attraction** is the organizing principle.
- **Temporal filtering** is a diagnostic probe revealing intrinsic properties.

The fundamental insight is that computation and persistence are in tension. Structures that compute (trajectories) tend not to survive. Structures that survive (basins) tend not to compute. Engineering robust computational systems requires acknowledging this tradeoff and designing accordingly: *compute briefly, then crystallize*.

Acknowledgments

This synthesis draws on six independent studies conducted as part of the Bit Dynamics research program. All empirical results are inherited from Papers 1-6; this paper introduces no new experiments.

References

- [1] Paper 1: Universal Computation Threshold. The Five-Bit Theorem and structural conditions for universality.
- [2] Paper 2: Stickiness and Hidden State. Necessity and sufficiency of hidden state for Control.
- [3] Paper 3: Self-Maintenance Census. 83.7% life-like classification under stickiness.
- [4] Paper 4: Leakiness and Capacity. Two-axis predictive framework with $R^2 = 0.96$.

[5] Paper 5: Invariants Under Temporal Filtering. Basin criterion for survival.

[6] Paper 6: Anti-Resonance. Carrier-period modulation of visibility.