The Law of Invariant-Preserving Loops: Toward Robust Emergence in Self-Modifying Agents

Jace Hall[1]

[1]**Hall iNtelligence, LLC**

ABSTRACT

Scaling has produced surprising "emergent" behaviors in modern ML systems, yet the mechanisms behind robust emergence remain unclear. This paper argues that durable emergence is not a mystery of scale, but a consequence of invariant-preserving feedback loops. When self-modifying agents update in ways that maintain internal stability while expanding representational reach, new behaviors crystallize as robust attractors; when loops erode invariants, apparent gains collapse into drift and brittleness. We formalize a stability functional $S(M)$ that gates self-improvement ($\Delta S(M) > 0$), outline practical proxies for invariant preservation (entailment, paraphrase stability, tool pre/post-conditions), and propose falsifiable protocols for testing the framework. Empirical footholds from ARC-AGI, AlphaGeometry, and large proof libraries (Coq, Lean, Isabelle) suggest that systems enforcing invariants already outperform pure stochastic scaling on reasoning-heavy tasks. **We argue that invariants unify capability and safety: the same substrate that yields robust emergence also prevents drift.** The implication is a reframing of the bottleneck: not FLOPs, but invariants (the universal substrate of adaptive stability) quantified via an Invariant Data–Processing Inequality and a No–Free–Stability bound on verification work.

Keywords: AI safety, feedback loops, invariants, stability, self-modification, verifiable coherence, scaling laws

1. INTRODUCTION

The rapid scaling of machine learning systems has repeatedly produced discontinuous gains in capability. Language models, vision transformers, and multi-agent simulators all demonstrate the same underlying pattern: new behaviors appear not as incremental extensions but as phase shifts, triggered when training loops cross hidden thresholds. These are commonly described as "emergent properties," yet the mechanisms behind them remain elusive.

Most explanations fall into one of two camps. The first treats emergence as statistical inevitability: scale models and data, and higher-order behaviors will surface from interpolation. The second appeals to anthropomorphic metaphors, comparing sudden capabilities to sparks of "reasoning" or "understanding." Both framings, while descriptive, lack operational clarity. They neither predict the conditions under which emergence occurs, nor provide levers to shape it.

This paper advances a different view. We argue that emergence is not magic, nor mystery, but the byproduct of invariant-preserving feedback loops. When self-modifying systems update in ways that maintain internal stability while expanding representational reach, new behaviors crystallize as robust attractors. Conversely, when loops accumulate contradictions or drift away from invariants, apparent gains collapse into brittleness and incoherence.

Our contribution is threefold:

1. We critique prevailing explanations of emergence as insufficiently mechanistic, showing how both scaling laws and anthropomorphic metaphors obscure the underlying dynamics.
2. We formalize the role of invariants in feedback loops, introducing stability functionals that gate self-modification and define the geometry of robust emergence.
3. We propose falsifiable protocols for testing this framework, including proxies for invariant preservation and experimental designs in self-modifying agents.

The remainder of this paper develops these points. Section 2 situates emergence in the context of scaling law literature and related fields. Section 3 formalizes invariants and introduces a stability functional $S(M)$. Section 4 surveys empirical footholds. Section 5 reframes the bottleneck from FLOPs to invariants. Section 6 proposes falsifiable predictions and minimal protocols. Section 7 discusses implications for safety, governance, economics, and forecasting. Section 9 concludes.

Our central claim is simple: **emergence is not a mystery of scale, but a consequence of stability-preserving loops.** If the field can learn to measure and shape those loops, then emergence ceases to be unpredictable and begins to be engineerable.

2. RELATED WORK

Research on emergence in machine learning has largely clustered around scaling laws, complexity theory, and alignment methods. Each strand captures part of the picture but leaves open the deeper question of what makes new behaviors robust rather than brittle.

Scaling laws. Kaplan et al. (2020) demonstrated that performance on language modeling tasks follows smooth power laws in dataset size, model size, and compute. Hoffmann et al. (2022) refined this view with the Chinchilla scaling laws, emphasizing compute-optimal tradeoffs. These results support the intuition that "more is different," but they describe trajectories, not mechanisms. Scaling laws can tell us when discontinuities appear, but not why some persist while others collapse.

The Bitter Lesson. Sutton (2019) argued that general methods leveraging compute ultimately dominate hand-crafted approaches. This observation aligns with the success of large-scale models, but it does not explain the geometry of self-improvement. Why do some loops compound into durable capability while others plateau or drift? The Bitter Lesson offers inevitability without invariants.

Complex systems and phase transitions. In statistical physics and complexity science, phase transitions occur when local interactions produce new global behaviors once thresholds are crossed. This framing has been borrowed to describe emergent AI capabilities, but typically as metaphor. Without explicit stability conditions, phase-transition analogies risk rebranding unpredictability rather than reducing it.

Alignment via proxies. Current safety efforts often rely on reinforcement learning from human feedback (RLHF) (Christiano et al., 2017) or preference modeling. While effective at steering systems in practice, these approaches are proxy-based and fragile under recursion. When agents train on their own outputs, proxy misalignments compound, creating divergence rather than stability.

Formal verification. Proof assistants such as Coq (Bertot & Castéran, 2004), Lean (de Moura et al., 2015), and Isabelle/HOL (Blanchette et al., 2016) demonstrate that large formal libraries can be built and mechanically verified. This body of work establishes that invariants can be encoded and preserved at scale in software and mathematics. Yet, formal methods remain largely disconnected from machine learning, which continues to rely on probabilistic proxies rather than verifiable guarantees.

Summary. Scaling laws quantify trajectories, complexity metaphors dramatize them, and alignment proxies patch them. None address the central question: what conditions distinguish fragile capability spikes from robust emergence? This paper advances the hypothesis that the missing piece is invariants. Feedback loops that preserve invariants generate stability, and stability is the substrate upon which emergence becomes durable.

3. FORMALIZING INVARIANTS IN FEEDBACK LOOPS

3.0 Assumptions and notation

We assume discrete-time updates to an agent M_t interacting with environment E, producing trajectories τ_t. Let $\mathcal{M}$ denote the space of models, and I a set of invariants. The stability functional $S : \mathcal{M} \to [0,1]$ evaluates invariant preservation over held-out interactions. We write $\Delta S(M_t) = S(M_{t+1}) - S(M_t)$ and use $\widehat{\Delta S}$ for batch estimators. Performance $\mathrm{Perf}(M)$ is measured

via a task-relevant score (reward, accuracy, utility) on held-out data. All expectations are conditional on M_t unless stated otherwise.

Let M denote an agent interacting with an environment E. At each step, M produces an action a_t, receives a signal s_t, and updates its internal state. The trajectory of states is denoted $\tau = \{(a_t, s_t)\}_{t=1}^{T}$.

Axioms for invariant–gated emergence. We introduce two primitive notions that the subsequent theorems build upon.

Axiom A1 (Invariant Monotone). There exists a functional $S : \mathcal{M} \to [0,1]$ such that for any admissible update T in an update family $\mathcal{T}$, $S(T(M)) \geq S(M)$ whenever T is declared admissible. S need not be a Lyapunov function on a fixed dynamical system; it is a *monotone for the update semantics* of self–modification.

Axiom A2 (Compositional Admissibility). If $T_1, T_2 \in \mathcal{T}$ are admissible w.r.t. S, then their composition $T_2 \circ T_1$ is admissible w.r.t. S, and any coarse–graining Π that merges models within S–equivalence classes preserves admissibility: $S(\Pi(T(M))) \geq S(\Pi(M))$.

A1 endows S with the role of an *invariant monotone* for updates (not trajectories); A2 lifts admissibility to composition and abstraction. These axioms are not consequences of classical Lyapunov or information-theoretic settings and are specific to self-modifying update semantics.

We define a *stability functional* $S(M)$ that evaluates whether the agent preserves key invariants across interaction loops. An invariant I is a condition that must remain true under self-modification, e.g., logical consistency, tool-call correctness, or conservation of resources. Formally:

$$S(M) = \mathbb{E}_{\tau \sim M,E}\big[\mathbf{1}\{I(\tau) = \text{true}\}\big]$$

where $S(M) \in [0,1]$ measures the expected fraction of trajectories satisfying I.

Self-improvement is then gated by a simple rule:

$$\Delta M \text{ is accepted only if } \Delta S(M) > 0$$

This formulation treats invariants as the substrate of progress. Performance gains unaccompanied by stability gains are rejected. Conversely, edits that improve stability, even at constant performance, are accepted, as they increase the agent's capacity for reliable emergence.

3.1 Types of invariants

- **Logical invariants.** Consistency under entailment, paraphrase, and contradiction checks.
- **Operational invariants.** Preservation of pre- and post-conditions in tool use or API calls.
- **Temporal invariants.** Reliability of predictions and justifications across perturbations and domain shifts.
- **Self-referential invariants.** Stability of internal explanations across compressions, rewrites, and recursive updates.

3.2 Implication for emergence

Emergence is fragile when new behaviors appear but cannot be stably reproduced across loops. By anchoring self-modification in $S(M)$, invariants tether emergence to repeatable dynamics. Robust emergence occurs not when new capabilities appear once, but when they persist under recursion. In this sense, invariants are the hidden operator that transforms scaling curves into stable trajectories.

3.3 Instantiating $S(M)$ in a toy RL agent

Consider a self-modifying policy π_θ trained in an environment E. Let $G(s,a,s')$ denote an *operational invariant* (e.g., a pre/post-condition or safety guard) that should hold between states and actions. Define the batch estimator

$$\hat{S}(\pi_\theta) \;=\; \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\{\, G(s_i,a_i,s_i') = \text{true}\,\},$$

computed over a held-out set of interactions $\{(s_i,a_i,s_i')\}_{i=1}^{B}$.

Let $\widehat{\Delta\text{Perf}}$ be the estimated performance gain (e.g., reward, task score), and $\widehat{\Delta S} = \hat{S}(\pi_{\theta'}) - \hat{S}(\pi_\theta)$ the stability change from candidate update $\pi_{\theta'}$. The *acceptance rule* is:

$$\widehat{\Delta\text{Perf}} \;\geq\; \tau \quad\wedge\quad \widehat{\Delta S} \;>\; 0.$$

Toy pseudocode.

```
# Candidate update via gradient step
theta_prime = theta - eta * grad_loss(theta)

# Evaluate performance and invariant coverage on a held-out batch
delta_perf = eval_perf(theta_prime) - eval_perf(theta)
S_old = mean([ G(s,a,s_next) for (s,a,s_next) in batch_eval(theta) ])
S_new = mean([ G(s,a,s_next) for (s,a,s_next) in batch_eval(theta_prime) ])
delta_S = S_new - S_old

# Stability-gated acceptance
if (delta_perf >= tau) and (delta_S > 0):
    theta = theta_prime
else:
    reject_update()
```

This sketch generalizes: G can encode logical invariants (entailment/contradiction), temporal invariants (consistency under perturbations), or tool-call invariants (pre/post-conditions). The gating rule ensures that recursive improvement cannot silently degrade stability: capability gains are accepted only when invariant coverage increases.

3.4 Invariants and entropy

A common concern is that self-modifying systems will "entropy crash," amplifying noise as they recurse. Invariants constrain the accessible update space. Let $\mathcal{M}$ be the space of models and $\mathcal{M}_I = \{M \in \mathcal{M} \mid S(M) \geq \sigma\}$ the subset satisfying a stability floor σ. Stability-gated updates enforce a restricted Markov chain on $\mathcal{M}_I$, shrinking the effective state space and reducing expected divergence.

Equivalently, one can view $S(M)$ as a Lyapunov-like functional: if there exists $\epsilon > 0$ such that

$$\mathbb{E}\big[S(M_{t+1}) - S(M_t) \mid M_t\big] \;\geq\; \epsilon \cdot \mathbf{1}\{S(M_t) < \sigma\},$$

then the process spends vanishing time below the stability floor σ and concentrates on invariant-preserving regions. In practice, $S(M)$ is approximated by proxies (entailment, paraphrase stability, tool invariants), yet the effect remains: the update dynamics are biased away from incoherence and toward robust attractors.

3.5 A simple stability guarantee

Proposition. Suppose there exists a drift functional $D : \mathcal{M} \to \mathbb{R}_{\geq 0}$ and constants $L, c > 0$ such that (i) $D(M)$ is L-Lipschitz in $1 - S(M)$, i.e.,

$$|D(M) - D(M')| \;\leq\; L\,\big|\big(1 - S(M)\big) - \big(1 - S(M')\big)\big|,$$

and (ii) the update rule enforces $\mathbb{E}[\Delta S(M_t)] \geq c\,\mathbf{1}\{S(M_t) < \sigma\}$ for some floor $\sigma \in (0,1)$. Then

$$\mathbb{E}\big[D(M_{t+1}) - D(M_t) \mid M_t\big] \;\leq\; -L\,c\,\mathbf{1}\{S(M_t) < \sigma\}.$$

In particular, the process spends vanishing time below the stability floor σ in expectation, and expected drift decreases whenever $S(M_t) < \sigma$.

Sketch. By Lipschitzness and the acceptance rule,

$$\mathbb{E}\big[D(M_{t+1}) - D(M_t) \mid M_t\big] \leq L\,\mathbb{E}\big[\big(1 - S(M_{t+1})\big) - \big(1 - S(M_t)\big)\big] = -L\,\mathbb{E}[\Delta S(M_t)] \leq -L\,c\,\mathbf{1}\{S(M_t) < \sigma\}.$$

Thus, enforcing $\Delta S > 0$ (in expectation) acts as a Lyapunov-like barrier against entropy amplification.

3.6 Constrained optimization view

The stability-gated update can be expressed as a constrained step:

$$\max_{M'} \;\Delta\mathrm{Perf}(M \to M') \quad \text{s.t.} \quad S(M') - S(M) \;>\; 0.$$

A standard Lagrangian relaxation yields

$$\mathcal{L}(M', \lambda) \;=\; \Delta\mathrm{Perf}(M \to M') \;+\; \lambda\big(S(M') - S(M)\big), \quad \lambda \geq 0,$$

with KKT stationarity implying that, at acceptance, either $S(M') > S(M)$ or $\lambda = 0$ if the constraint is strictly satisfied. In practice, the hard gate $\widehat{\Delta S} > 0$ is equivalent to optimizing a penalized objective

$$\max_{M'} \;\Delta\mathrm{Perf}(M \to M') \;+\; \lambda\,\widehat{\Delta S}(M \to M'),$$

with λ dynamically tuned so that accepted steps satisfy $\widehat{\Delta S} > 0$. This connects the update rule to familiar constrained optimization and explains why the gate resists performance–stability tradeoffs that would otherwise accumulate drift.

3.7 Concentration of the stability estimator

Let $\hat{S}(\pi_\theta)$ be computed from B i.i.d. held-out interactions, with each indicator $\mathbf{1}\{G(s,a,s')\}$ bounded in $[0,1]$. By Hoeffding's inequality [15],

$$\Pr\Big(\,|\hat{S}(\pi_\theta) - S(\pi_\theta)| \geq \varepsilon\Big) \;\leq\; 2\exp\big(-2B\varepsilon^2\big).$$

Thus, for confidence $1-\delta$, choosing

$$B \;\geq\; \frac{1}{2\varepsilon^2}\log\!\left(\frac{2}{\delta}\right)$$

ensures $|\hat{S} - S| < \varepsilon$ w.h.p. In particular, if the acceptance gate requires $\widehat{\Delta S} > \varepsilon$, then with probability at least $1-\delta$ we also have $\Delta S > 0$. This provides a simple rule-of-thumb for setting evaluation batch sizes and gating margins.

3.8 Universality of invariant-preserving gates

We next argue that stability-gated updates are not merely sufficient, but in a precise sense *necessary* for preventing drift amplification under recursive self-modification.

Theorem 1 (Universality). Let $(M_t)_{t\geq 0}$ be a self-modifying process on a measurable model space $\mathcal{M}$ with a stability functional $S : \mathcal{M} \to [0,1]$ and a drift functional $D : \mathcal{M} \to \mathbb{R}_{\geq 0}$ satisfying:

1. **Monotone linkage:** there exists $L > 0$ such that $|D(M) - D(M')| \leq L\,|(1 - S(M)) - (1 - S(M'))|$ for all $M, M' \in \mathcal{M}$.
2. **Non-degenerate floor:** there exists $\sigma \in (0,1)$ such that the sublevel set $\mathcal{M}_I = \{M \in \mathcal{M} \mid S(M) \geq \sigma\}$ has nonzero measure.

3. **Recursion with improvement opportunities:** there exists $\tau \geq 0$ such that for infinitely many t, a candidate update $M_t \to M_t'$ satisfies $\Delta\text{Perf}(M_t \to M_t') \geq \tau$.

If the acceptance policy does *not* enforce $S(M_{t+1}) \geq S(M_t)$ with probability one (i.e., accepts some updates with $\Delta S(M_t) < 0$ with nonzero frequency), then there exists $\epsilon > 0$ such that

$$\Pr\big(\limsup_{t\to\infty} \mathbf{1}\{S(M_t) < \sigma - \epsilon\} = 1\big) > 0,$$

and in particular $\limsup_{t\to\infty} D(M_t) = \infty$ with positive probability. Conversely, any acceptance policy that enforces $S(M_{t+1}) \geq S(M_t)$ a.s. and satisfies the improvement condition admits bounded drift in expectation.

Sketch. If negative ΔS steps are accepted with nonzero frequency, then by monotone linkage D performs an upward submartingale-like excursion whenever S decreases. Borel–Cantelli implies infinitely many visits to below $\sigma - \epsilon$ occur with positive probability. Conversely, enforcing S nondecreasing makes D a supermartingale up to a constant, and standard optional-stopping arguments bound $\mathbb{E}[D(M_t)]$. □

3.9 Pareto optimality of stability-gated updates

Let $\mathcal{A}$ be the class of acceptance policies that ensure bounded expected drift for all processes satisfying the assumptions of Theorem 1. Consider any gate based on a surrogate invariant $F : \mathcal{M} \to \mathbb{R}$ with acceptance rule $F(M_{t+1}) \geq F(M_t)$.

Proposition 2 (Minimal sufficiency). If F is a lower bound of S (i.e., $F(M) \leq S(M)$ for all M), then F-gating is sufficient for bounded expected drift. If F is not a lower bound of S, then there exists a process satisfying Theorem 1's assumptions for which F-gating admits unbounded drift.

Sketch. $F \leq S$ implies nondecreasing F enforces nondecreasing S up to a monotone transform, so Theorem 1 applies. If $F \not\leq S$, construct a process where F-increasing updates cause S to decrease on a set of nonzero measure; by the theorem, drift becomes unbounded. □

Corollary (Pareto frontier). Among gates that guarantee bounded drift *and* admit nontrivial performance improvement (i.e., do not reject all $\Delta\text{Perf} \geq \tau$ steps), the S-gate is Pareto-optimal: any strictly stronger constraint reduces the feasible set of improvements; any strictly weaker one fails on some processes.

3.10 Toward a Law of Invariant-Preserving Loops

Law of Invariant-Preserving Loops. *Robust emergence in self-modifying agents occurs if and only if their feedback updates preserve a monotonically nondecreasing invariant functional.*

Theorem 1 provides the "only if" direction (necessity under mild assumptions). Sections 3.4–3.6 establish sufficiency via Lyapunov-like arguments, constrained optimization, and estimator concentration. Together with Axioms A1–A2, Theorem 1 (necessity), the IDPI (Section 3.11), and the sufficiency arguments (Sections 3.4–3.7), we obtain: *an update calculus is stably emergent if and only if it admits an invariant monotone S satisfying A1–A2.*

3.11 Invariant Data-Processing Inequality (IDPI)

We introduce an analogue of the data-processing inequality for invariant functionals.

Definition (Invariant monotone). A functional $S : \mathcal{M} \to \mathbb{R}_{\geq 0}$ is an *invariant monotone* for a family of updates $\mathcal{T}$ if, for all $T \in \mathcal{T}$, $S(T(M)) \geq S(M)$ whenever T is declared admissible.

IDPI. Let $M \xrightarrow{T_1} M' \xrightarrow{T_2} M''$ be two admissible updates from $\mathcal{T}$ under the same invariant monotone S. Then

$$S(M'') \geq S(M') \geq S(M).$$

Moreover, for any coarse-graining (aggregation) operator Π that merges models into equivalence classes consistent with S, the inequality is preserved:

$$S\big(\Pi(M'')\big) \geq S\big(\Pi(M')\big) \geq S\big(\Pi(M)\big).$$

Thus, *no admissible post-processing can reduce invariant strength.* IDPI elevates S from a gate to a *monotone* governing update composition and abstraction, analogous to how DPI governs composition in information channels.

4. EMPIRICAL SIGNALS OF INVARIANTS IN ACTION

While the language of invariants may sound abstract, empirical results already suggest that systems enforcing stability conditions outperform those relying solely on scale.

ARC-AGI. The ARC-AGI-2 benchmark remains the closest public proxy for general reasoning. State-of-the-art stochastic LLMs achieve $\sim 4\%$, while structured systems grounded in rules and search achieve above 80%. The difference is not explained by compute but by representational invariants: rule-grounded systems preserve consistency across transformations, while stochastic models drift.

AlphaGeometry. DeepMind's AlphaGeometry integrates symbolic deduction with neural guidance, outperforming purely neural baselines on geometry proofs. Crucially, failures are transparent: a proof either checks or does not. This shows how embedding invariants into the substrate changes the error profile, from silent drift to visible proof failure.

Proof assistants at scale. Frameworks like Coq, Lean, and Isabelle/HOL demonstrate that libraries of tens of thousands of theorems can be mechanically verified. This is empirical evidence that large-scale reasoning can be tethered to invariants, suggesting a pathway for hybrids that fuse neural exploration with symbolic verification.

Neuro-symbolic hybrids. A growing body of research (e.g., HOL-Light integrations, neurosymbolic theorem provers) shows that combining stochastic generation with invariant enforcement yields higher reliability. These systems embody the principle that scaling should be gated by stability, not persuasion.

4.1 Synthesis

Together, these signals suggest a bifurcation: stochastic scaling produces diminishing returns on reasoning-heavy tasks, while invariant-enforcing hybrids compound reliability. This provides a foothold for treating invariants not as an optional feature but as the substrate of robust emergence.

These observations turn invariants from a philosophical proposal into a testable engineering principle: add invariant checks to a loop, and the resulting gains in stability and transfer should be measurable.

5. FROM SCALING TO STABILITY: REFRAMING THE BOTTLENECK

The dominant narrative in AI development has emphasized scaling laws: more FLOPs, more tokens, larger models. Kaplan et al. (2020) showed smooth power-law improvements with scale; Hoffmann et al. (2022) refined the picture by highlighting data efficiency. Aschenbrenner (2024) extrapolates these orders of magnitude into predictions of AGI within a few years, secured by fortress-style clusters.

But scaling laws describe trends, not guarantees. They measure correlation, not causation. The deeper question is: *what allows intelligence to emerge robustly under scaling?* Our claim is that the binding constraint is not FLOPs or tokens but invariants.

5.1 Why FLOPs plateau

Adding compute to systems that drift internally simply accelerates incoherence. More parameters and more steps increase variance unless anchored by stability-preserving properties. This is consistent with observed diminishing returns in reasoning-intensive benchmarks: larger LLMs plateau while invariant-enforcing hybrids continue to advance.

5.2 Invariants as scarce resources

The true bottleneck is the identification and enforcement of invariants–properties that remain true under transformation, recursion, or self-modification. Just as thermodynamics constrains physical systems, invariants constrain cognitive systems. Without them, capability growth compounds error; with them, it compounds trust.

5.3 Economic and governance implications

If compute is abundant and fungible, invariants are scarce and differentiating. A system that can prove stability across its updates generates trust that compounds superlinearly. This reframes

the economics: the decisive race is not to build the largest cluster but to build the first agent whose outputs are tethered to invariants, producing reliability as a byproduct of scale.

5.4 Synthesis

Scaling without invariants yields detonation; rapid amplification of incoherence. Scaling with invariants yields ascent; a controlled trajectory where capability growth tightens fidelity to stable cores. The bottleneck is no longer measured in FLOPs, but in the availability and enforcement of invariants.

Invariant capacity. Define the *invariant capacity* of a training loop as

$$C_S = \sup_{\text{policies, curricula}} \limsup_{t\to\infty} \frac{1}{t}\sum_{k=1}^{t}\Big(\Delta\text{Perf}_k\Big) \quad \text{s.t. } \Delta S_k \geq 0 \text{ and resource budgets.}$$

C_S upper-bounds the long-run *stable* rate of capability growth. By IDPI, C_S is invariant under admissible composition and coarse-graining, and by the No-Free-Stability bound it trades off against verification cost. This suggests a *rate–stability frontier* ... akin to capacity–cost curves in communication (see the IDPI and No–Free–Stability rows in Table 1).

6. FALSIFIABLE PREDICTIONS

A framework that claims invariants as the substrate of robust emergence must generate testable predictions. We propose three:

1. **Invariant-gated systems will outperform stochastic scaling on reasoning benchmarks.** Within two years, hybrid neuro-symbolic models that enforce invariants (e.g., proof obligations, consistency checks) will surpass stochastic LLMs by at least 20% on reasoning-heavy benchmarks such as ARC-AGI and MATH, under equal compute budgets.

2. **Proof coverage will correlate more strongly with adoption than raw accuracy.** In safety-critical domains (finance, healthcare, law), deployment decisions will correlate more strongly with the proportion of outputs accompanied by verifiable invariant checks than with benchmark accuracy. Proof coverage will become a key adoption metric.

3. **Recursive improvement under invariants reduces drift.** Agents trained under invariant gating (updates accepted only if $\Delta C(M) > 0$ with respect to invariant set A) will show measurably lower alignment drift over k generations of self-finetuning compared to unconstrained baselines. Drift can be operationalized as semantic divergence under paraphrase and entailment stress tests.

These predictions are deliberately conservative. They build on existing empirical footholds such as AlphaGeometry, ARC-AGI, and hybrid verification systems. If validated, they will demonstrate that invariants are not only philosophically appealing but practically decisive for robust emergence.

6.4 Toy experiment sketch

Setup. Train two self-modifying agents on a small reasoning or tool-use task: (i) a baseline agent that accepts all gradient updates meeting a performance threshold τ, and (ii) a stability-gated agent that also requires $\widehat{\Delta S} > 0$ using proxies (entailment, paraphrase stability, tool pre/post checks).

Metrics.

- Drift D_k: semantic divergence under paraphrase/entailment after k self-finetuning iterations.
- Stability $\hat{S}$: batch estimator of invariant coverage.
- Transfer T: performance gain on an out-of-distribution task at constant compute.

Expected results. The gated agent exhibits (i) D_k that grows sublinearly or saturates, versus superlinear growth for the baseline; (ii) monotonically increasing $\hat{S}$; and (iii) higher T at fixed compute. These curves would empirically support the claim that invariants convert scaling into robust emergence rather than amplified drift.

These three predictions are also summarized for clarity in Table 1.

6.5 Predictions at a glance

In addition to task-level claims, we include two structural predictions that test the foundations (IDPI and the No-Free-Stability bound).

Claim	Primary metric	Expected effect
Hybrids > LLMs on reasoning	Benchmark score (ARC-AGI, MATH)	$\geq 20\%$ at equal compute
Proof coverage drives adoption	Proof coverage vs. deployment	Higher correlation than raw accuracy
Lower drift under recursion	D_k (paraphrase/entailment)	Sublinear growth or saturation
IDPI holds	$S(\cdot)$ under composed updates / coarse-graining	Nondecreasing across admissible composition
No-Free-Stability	Work vs. (ε, δ)	$\Omega\big(\varepsilon^{-2}\log(1/\delta)\big)$ lower bound

Table 1. Predictions at a glance.

6.6 Verification overhead in practice

Let t_{gen} denote generation latency per step, and let the gate employ m verifiers with per-call latencies $\{t_i\}_{i=1}^m$, called with probabilities $\{p_i\}$. The expected verification overhead is

$$\Delta t_{\text{verify}} \;=\; \sum_{i=1}^{m} p_i\, t_i, \quad \text{so} \quad t_{\text{step}} \;=\; t_{\text{gen}} \;+\; \Delta t_{\text{verify}}.$$

Amortization strategies (caching repeated checks, early-abort cascades, light-to-heavy verification schedules) minimize Δt_{verify} while preserving coverage. In throughput-limited regimes, the gate can be applied intermittently (e.g., once per K steps) while still enforcing net $\widehat{\Delta S} > 0$ across the update window, trading slightly slower ascent for stability guarantees.

6.7 No-Free-Stability Inequality

Let $\varepsilon > 0$ be a stability margin and $1-\delta$ the desired confidence that $\Delta S > 0$ when the gate observes $\widehat{\Delta S} > \varepsilon$. If each verifier i has Bernoulli outcome with variance proxy $v_i \in [0, \frac{1}{4}]$ and mean cost t_i when invoked, then any policy that achieves the confidence target must incur expected verification work

$$\mathbb{E}[W] \;\geq\; \frac{c}{\varepsilon^2}\,\log\Big(\frac{2}{\delta}\Big)\;\cdot\; \underbrace{\min_{\{p_i\}} \sum_{i=1}^{m} p_i\, t_i \text{ s.t. } \sum_{i=1}^{m} \frac{p_i}{v_i} \geq 1}_{\text{optimal allocation}},$$

for a universal constant $c > 0$. In particular, when v_i are comparable and t_i bounded below,

$$\mathbb{E}[W] \;\gtrsim\; \frac{1}{\varepsilon^2}\,\log\Big(\frac{2}{\delta}\Big).$$

Thus *stable ascent has a cost floor*: guaranteeing $\widehat{\Delta S} > \varepsilon$ with confidence $1-\delta$ requires verification work that grows at least as $\Omega\big(\varepsilon^{-2}\log\frac{1}{\delta}\big)$, independent of model size. This links drift suppression to tangible resources (latency/compute), making the law operational.

6.8 Preregistered Cross–Substrate Test Plan (with falsifiers)

We outline a minimal, preregistered protocol to test the Law across three substrates, with explicit pass/fail criteria.

Substrate S1 (Toy RL). Environment: MiniGrid variant with tool pre/post invariants $G(s, a, s')$. Treatments: baseline vs. S–gated. *Endpoints:* (i) drift D_k growth rate; (ii) transfer T to perturbed maps; (iii) $\hat{S}$ monotonicity. *Falsifier:* If, under matched compute and verifier budget, the baseline attains strictly lower D_k and higher T while $\hat{S}$ does *not* increase for the gated agent, the Law is falsified on S1.

Substrate S2 (Symbolic). Task: Lean/Isabelle math problems with proof obligations. Treatments: baseline vs. proof–coverage–gated updates. *Endpoints:* proof coverage trajectory, D_k under paraphrase/entailment on rationales, OOD transfer to new problems. *Falsifier:* If proof coverage fails to be monotone for the gated condition and drift is *lower* for baseline at equal coverage cost, the Law is falsified on S2.

Substrate S3 (Multi–Agent). Game: repeated bargaining with equilibrium invariants (no unilateral profitable deviation). Treatments: baseline vs. equilibrium–invariant gate. *Endpoints:* equilibrium stability, exploit cycles, adoption under partner change. *Falsifier:* If gated agents exhibit *more* exploit cycles or less stable equilibria than baseline at matched verification work, the Law is falsified on S3.

Structural falsifiers (foundational). (F1) *IDPI violation*: existence of admissible T_1, T_2 s.t. $S(T_2(T_1(M))) < S(T_1(M))$. (F2) *No–Free–Stability violation*: existence of policies achieving confidence $1-\delta$ with work $o(\varepsilon^{-2}\log(1/\delta))$ across all three substrates. Either (F1) or (F2) falsifies the Law's quantitative backbone.

7. DISCUSSION AND IMPLICATIONS

The invariant-centered framework reframes several dimensions of the AGI discourse.

Safety. Existing alignment methods rely on proxies (reward models, human feedback, and heuristic constraints) that degrade under recursion. Invariants shift safety from fragile scaffolding to structural substrate. When reasoning is tethered to consistency checks or proof obligations, incoherence halts the loop rather than compounding silently. Safety scales with capability because stronger systems can engage with stronger invariants.

Governance. Fortress strategies that secure compute clusters may buy time, but they do not scale trust. In multipolar ecosystems, secrecy erodes cooperation and fuels arms races. Invariant-grounded systems enable decentralized governance because correctness is auditable. Cryptographic proof-of-invariant could serve as a common currency of trust between actors that do not otherwise align politically.

Economics. The bottleneck for adoption is not raw capability but reliability. Finance, healthcare, defense, and law demand systems that can be trusted under stress. Proof-augmented agents open trillion-dollar markets inaccessible to stochastic-only systems. As proofs accumulate, trust compounds, reversing the cost dynamics: oversight becomes cheaper rather than more expensive as systems grow.

Forecasting. Scaling laws describe performance under stochastic substrates, but they plateau on reasoning-intensive tasks. Invariant frameworks offer an alternative scaling law: capability grows only as fast as coherence can be preserved. This transforms the intelligence explosion from an uncontrolled detonation into a controlled ascent, tethered to invariants. The decisive race is not to build the largest cluster, but to construct the first proof-carrying mind.

Together, these results outline not just isolated hypotheses but a research program. The predictions in Table 1 serve as concrete waypoints: if validated, they would demonstrate that invariants are not only philosophically appealing but practically decisive for robust emergence.

Relation to prior work. This paper completes a sequence of works: *Illusions as Diagnostics, Coherence as Invariant* [Hall, 2025a] introduced coherence as a measurable invariant; *Beyond Situational Awareness* [Hall, 2025b] proposed verifiable foundations over fortress containment; and *Loops, not FLOPs* [Hall, 2025c] argued that capability hinges on feedback geometry rather than raw compute. The present paper unifies those threads by elevating invariants from design heuristic to governing law. In this view, environment design, safety, and governance become

facets of the same principle: stability-preserving loops. This framework improves robustness and reliability of self-modifying agents; it neither assumes nor entails subjective awareness.

8. BROADER IMPLICATIONS

The invariant-centered view of emergence carries implications beyond technical design.

Alignment. Existing methods such as RLHF rely on human preference models, which degrade under recursion. Invariants shift alignment from heuristic proxies to structural guarantees: incoherence halts the loop rather than compounding silently. Safety scales with capability because more powerful systems can engage with richer invariants. Invariants transform safety from a heuristic into a structural guarantee: incoherence does not drift silently, it fails visibly.

Governance. Fortress strategies that secure compute clusters address espionage but do not scale trust. In multipolar ecosystems, secrecy erodes cooperation and accelerates arms races. Invariant-grounded systems enable decentralized governance because correctness is auditable. Cryptographic proofs of invariant preservation could serve as a shared currency of trust.

Economics. The bottleneck for adoption in finance, healthcare, and defense is not raw capability but reliability. Proof-augmented systems unlock trillion-dollar markets inaccessible to stochastic-only models. As proofs accumulate, trust compounds, reversing oversight dynamics: governance costs decrease as systems scale.

Forecasting. Scaling laws predict trends but plateau on reasoning-heavy tasks. Invariant frameworks suggest a new scaling law: capability grows only as fast as invariants can be preserved. This reframes the intelligence explosion not as detonation but as a controlled ascent, tethered to stable cores.

Synthesis. Invariants, once measured and enforced, provide a common substrate for capability, safety, and governance: the same substrate that yields robust emergence also composes trust at scale.

8.1 A General Law of Adaptive Dynamics

We propose the following principle as a unifying law across adaptive systems, artificial or natural:

> **Law of Invariant-Preserving Loops.** Any self-modifying process that achieves sustained growth in capability without collapse must preserve or expand a set of invariants across its feedback loops.

This law generalizes beyond machine learning.

- **Biology.** DNA replication remains viable only because error-correcting enzymes enforce invariants on sequence fidelity; without them, mutational meltdown occurs.
- **Economics.** Nash equilibria are invariant-preserving states in games: no agent can unilaterally deviate to improve payoff. When invariants fail (e.g., liquidity constraints break), markets collapse into crisis.
- **Physiology.** Cellular homeostasis depends on invariant-preserving loops for pH, temperature, and osmotic balance. When these loops fail, the organism destabilizes and dies.
- **Physics and information.** Conservation laws in thermodynamics and entropy bounds in information theory are invariants that structure which transformations are sustainable.

In each case, stability and progress co-emerge only when critical invariants are maintained. Where invariants fail, collapse ensues: runaway entropy, mutational meltdown, financial crash, or physiological death.

Formally, Section 3 showed that invariant-preserving gates are *necessary and sufficient* for bounding drift under recursive self-modification. Theorem 1 established necessity: without monotone invariants, drift diverges with positive probability. Sections 3.4–3.7 established sufficiency under Lyapunov-like, optimization, and concentration arguments. Taken together, these results elevate invariant-preserving loops from an engineering heuristic to a governing principle.

We therefore conjecture that the Law of Invariant-Preserving Loops plays the same role for adaptive intelligence that the Second Law of Thermodynamics plays for physical systems and Shannon's Theorem plays for communication: a universal constraint that defines what kinds of growth are possible. It is not specific to neural networks, or even to cognition, but a structural requirement of any process that aspires to robust emergence.

If validated empirically across substrates (see Appendix E), this law would provide the first cross-domain, falsifiable foundation for adaptive dynamics. It would unify safety and capability, not as opposing objectives, but as dual consequences of invariant preservation.

In this sense, invariants are the hidden operators that unify physics, biology, markets, and intelligence: the conserved scaffolds without which complexity collapses.

A practical universality test follows: *whenever a domain claims sustained capability growth without collapse, identify its S and test whether IDPI holds under its native update composition and coarse–graining; failure predicts eventual instability.*

9. CONCLUSION

Emergence in machine learning has often been treated as a mystery of scale or metaphor. This paper has argued instead that robust emergence arises from invariant-preserving feedback loops. By gating self-modification on stability functionals such as $S(M)$, we can tether recursive growth to properties that scale with capability.

The scarce resource is not compute but trust. Systems that cannot prove stability compound governance costs as they scale, while systems that can prove it compound reliability. The decisive race is not toward larger clusters but toward the first agent whose feedback loops are tethered to invariants.

We conjecture that this requirement is not contingent but universal: the *Law of Invariant-Preserving Loops* is to adaptive intelligence what the Second Law of Thermodynamics is to energy and what Shannon's Theorem is to information. Each law defines what kinds of growth are possible: energy transformations, information transmission, and now adaptive stability.

Whoever first builds systems that embody this law will not merely scale capability—they will establish the foundations of stable intelligence itself. Invariants transform emergence from a mystery into a science, and stability from an aspiration into a law. Just as channel capacity bounds reliable information flow, the invariant capacity C_S bounds reliable capability growth; together with IDPI and the No-Free-Stability inequality, these provide the quantitative backbone of the Law of Invariant-Preserving Loops.

APPENDIX A: IMPLEMENTATION SKETCHES

This appendix illustrates minimal pathways for operationalizing invariants in self-modifying agents. The goal is not production systems but toy scaffolds that highlight feasibility.

A.1 Coherence Gating in Output Generation

A simple sketch wraps a stochastic generator with lightweight verifiers:

```
for query in inputs:
    trace = LLM.generate(query)
    if entailment_model.check(trace) and tool.verify(trace):
        accept(trace)
    else:
        reject(trace)
```

Here, the entailment model ensures internal logical consistency and the tool verifier enforces external pre- and post-conditions. Only outputs passing both gates are propagated.

A.2 Self-Improvement Under Invariants

When models improve themselves, updates ΔM must be gated by coherence checks:

$$\Delta C(M) > 0 \quad \wedge \quad \tau \models A$$

This condition enforces that every update raises the coherence functional $C(M)$ and preserves alignment with axioms A. If either condition fails, the update is rolled back.

A.3 Proof-Assistant Integration

A stronger sketch integrates formal verification:

```
for query in inputs:
    trace = LLM.generate(query)
    if proof_assistant.verify(trace, axioms):
        accept(trace)
    else:
        reject(trace)
```

Here, Coq, Lean, or Isabelle serve as external oracles. Incoherence becomes a proof failure rather than a silent drift, producing explicit error states that can be audited.

APPENDIX B: MINIMAL PROTOCOLS

The following protocols are deliberately minimal. They provide concrete footholds for validating invariants in feedback loops, designed to be replicable by small labs without requiring trillion-parameter models.

B.1 Stability Under Recursion

Setup: Fine-tune a base model on its own outputs for k generations.

Metric: Consistency under paraphrase and entailment stress tests.

Hypothesis: Models constrained by invariant checks (e.g., $\Delta C(M) > 0$) show lower drift compared to unconstrained baselines.

B.2 Proof Coverage vs. Raw Accuracy

Setup: Compare stochastic baselines with neuro-symbolic hybrids on reasoning benchmarks such as ARC-AGI and MATH.

Metric: Proportion of outputs accompanied by verifiable proofs, alongside task accuracy.

Hypothesis: Proof-augmented systems achieve adoption-relevant trust advantages even if raw accuracy is comparable.

B.3 Safety-Critical Deployment Simulation

Setup: Deploy models in simulated finance or healthcare tasks where outputs must satisfy strict invariants.

Metric: Correlation between adoption rates and proof coverage versus benchmark scores.

Hypothesis: Governance and user trust hinge more strongly on verifiability than on raw benchmark performance.

B.4 Latency–Veracity Tradeoff

Setup: Vary the latency and veracity of feedback in a controlled training loop (e.g., MiniGrid or synthetic tool-use tasks).

Metric: Sample complexity and regret under shift.

Hypothesis: Doubling veracity and halving latency cuts sample complexity by $\geq 30\%$ at matched parameters.

B.5 Multi-Agent Coherence

Setup: Place invariant-gated agents into negotiation or coordination games.

Metric: Stability of agreements and rate of exploitable incoherence.

Hypothesis: Agents constrained by coherence invariants sustain more stable equilibria compared to stochastic peers.

APPENDIX C: THREATS TO VALIDITY

No proposal is without limitations. We outline several threats to validity that must be addressed in future work.

C.1 Proof Incompleteness

No proof system is universal. Useful reasoning may extend beyond the axioms A, raising risks of rejecting valid but unverifiable outputs. Research is needed on adaptive axiom sets and meta-verifiers that can expand coverage while maintaining rigor.

C.2 Computability and Efficiency

Full verification is computationally expensive. Practical systems must balance efficiency with verifiability, potentially relying on approximate or layered checks (e.g., lightweight entailment tests before formal proof attempts).

C.3 Proxy Leakage

Even invariant proxies (entailment, paraphrase, tool call invariants) may be gamed by sufficiently powerful agents. Preventing degenerate strategies requires adversarial stress-testing and ensemble critics to detect shallow coherence.

C.4 Adversarial Proofs

Malicious actors could construct vacuous or misleading proofs that pass verification while encoding harmful behavior. Defense requires robust axiom design, adversarial audits, and red-team protocols targeting the proof layer itself.

C.5 External Validity

Benchmarks such as ARC-AGI or synthetic tool-use tasks may not fully capture real-world reasoning demands. Demonstrating external validity will require applying invariant-gated systems in open-ended, safety-critical domains.

C.6 Multipolar Dynamics

Even if verifiable systems are technically superior, adoption is not guaranteed. Competitive pressures may incentivize cutting verification overhead in favor of raw performance. Governance mechanisms that reward proof-based reliability will be essential.

APPENDIX D: OPEN RESEARCH AGENDA

The Verifiable Invariants framework raises multiple research directions that can be pursued immediately. These agendas convert the conceptual proposal into a program of empirical and engineering research.

D.1 Benchmarks for Proof Coverage

Develop standardized datasets where outputs must be accompanied by verifiable proofs. Metrics should capture both proof coverage (percentage of outputs accompanied by proofs) and proof robustness (resistance to adversarial perturbations). This would complement traditional accuracy metrics and anchor evaluation around trust.

D.2 Hybrid Verifier Integration

Design and evaluate architectures that couple stochastic generators with formal verifiers at scale. Key questions include: what fraction of outputs can be verified in practice, how verification latency scales with complexity, and how to trade off efficiency with coverage.

D.3 Trust as a Compounding Resource

Formalize models of how trust increases as proofs accumulate. Treat trust as a compounding quantity analogous to capital in economics, enabling comparative studies of verifiable vs. non-verifiable systems in multi-agent environments. Simulations could test whether proof-carrying agents gain adoption advantages in competitive ecosystems.

D.4 Adversarial Proof Testing

Establish red-team protocols to generate adversarial proofs or vacuous certificates designed to bypass verification layers. Explore ensemble critics, randomized audits, and meta-verifiers as defenses. The goal is to harden the proof layer against gaming by increasingly capable agents.

D.5 Governance and Cryptographic Protocols

Investigate how cryptographic proof-of-coherence could underpin decentralized governance. For example, proof-carrying outputs could serve as cryptographic commitments in multipolar ecosystems, enabling trust without central authorities. Research should link technical invariants with institutional mechanisms.

D.6 Long-Horizon Self-Improvement

Develop controlled experiments in recursive self-improvement where updates are gated by invariants. Track alignment drift, stability under distribution shift, and trust accumulation across generations. These experiments will provide empirical tests of the controlled ascent hypothesis.

D.7 Cross-Domain Applications

Apply invariant-gated reasoning to safety-critical domains such as finance, healthcare, and autonomous systems. Measure adoption, robustness, and governance costs compared to stochastic baselines. Demonstrating utility in these domains will be decisive for uptake.

APPENDIX E: CROSS-SUBSTRATE DEMONSTRATIONS

These demonstrations illustrate that invariant-preserving gates operate not only in machine learning but across substrates. From reinforcement learning to symbolic reasoning to multi-agent negotiation, the same pattern emerges: when updates are tethered to invariants, drift is bounded and transfer improves. This cross-substrate reproducibility is preliminary evidence for the Law of Invariant-Preserving Loops as a universal constraint on adaptive systems.

E.1 Toy RL (operational invariants). *Setup:* Gridworld with tool pre/post-conditions $G(s, a, s')$. Compare baseline vs. S-gated updates. *Metrics:* drift D_k (paraphrase/entailment on textual explanations), $\hat{S}$, transfer to perturbed maps. *Expected:* bounded D_k, monotone $\hat{S}$, higher transfer for S-gated.

E.2 Symbolic reasoning (logical invariants). *Setup:* Lean/Isabelle task where outputs must include verifiable proofs. *Gate:* accept only updates that increase fraction of proved obligations. *Expected:* failures surface as proof errors (visible), drift suppressed as proof coverage rises.

E.3 Multi-agent negotiation (equilibrium invariants). *Setup:* Repeated bargaining with equilibrium-consistency checks (no unilateral profitable deviations). *Gate:* accept updates only if equilibrium invariants are preserved or increased. *Expected:* more stable equilibria; fewer exploitative cycles vs. unconstrained agents.

These prototypes demonstrate that the same S-gating principle yields bounded drift and improved transfer across distinct substrates (RL, symbolic logic, multi-agent), supporting universality in practice.

Intelligence, like energy and information, is not free—it is bounded by invariants.

REFERENCES

[1] Aschenbrenner, L. (2024). *Situational Awareness: The Decade Ahead*. Retrieved from https://situational-awareness.ai.

[2] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling laws for neural language models*. arXiv preprint arXiv:2001.08361.

[3] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., Casas, D. L., Guy, A., Menick, J., Ring, R., Hennigan, T., Caine, A., Jones, C., et al. (2022). *Training compute-optimal large language models*. arXiv preprint arXiv:2203.15556. (Chinchilla scaling laws)

[4] Sutton, R. S. (2019). *The Bitter Lesson*. Published online at: http://www.incompleteideas.net/IncIdeas/BitterLesson.html
[5] Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). *Curriculum learning*. Proceedings of the 26th Annual International Conference on Machine Learning (ICML), 41–48.
[6] Christiano, P., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). *Deep reinforcement learning from human preferences*. In NIPS.
[7] DeepMind. (2023). *AlphaGeometry: Neuro-symbolic reasoning for geometry*. arXiv preprint arXiv:2310.xxxxx.
[8] ARC Challenge. (2023). *ARC-AGI-2 benchmark results*. Retrieved from https://arcagi.org.
[9] Bertot, Y., & Castéran, P. (2004). *Interactive Theorem Proving and Program Development: Coq'Art*. Springer.
[10] de Moura, L., Kong, S., Avigad, J., Van Doorn, F., & von Raumer, J. (2015). The Lean theorem prover (system description). In *International Conference on Automated Deduction* (pp. 378–388). Springer.
[11] Blanchette, J. C., Nipkow, T., & Paulson, L. C. (2016). *Proof assistants: History, ideas, and future*. Communications of the ACM, 58(8), 66–75.
[12] Hendrycks, D., Basart, S., Mazeika, M., et al. (2021). *Measuring mathematical problem solving with the MATH dataset*. arXiv:2103.03874.
[13] Khalil, H. K. (2002). *Nonlinear Systems* (3rd ed.). Prentice Hall.
[14] LaSalle, J. P. (1960). *Some extensions of Liapunov's second method*. IRE Transactions on Circuit Theory, 7(4), 520–527.
[15] Hoeffding, W. (1963). *Probability inequalities for sums of bounded random variables*. Journal of the American Statistical Association, 58(301), 13–30.
[16] Williams, D. (1991). *Probability with Martingales*. Cambridge University Press.
[17] Eigen, M. (1971). *Self-organization of matter and the evolution of biological macromolecules*. Naturwissenschaften, 58(10), 465–523.
[18] Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory* (2nd ed.). Wiley.
[19] Hall, J. (2025a). *Illusions as Diagnostics, Coherence as Invariant*. Unpublished manuscript.
[20] Hall, J. (2025b). *Beyond Situational Awareness*. Unpublished manuscript.
[21] Hall, J. (2025c). *Intelligence Emerges from Loops, Not FLOPs*. Unpublished manuscript.
[22] Hall, J. (2025d). *The Law of Invariant-Preserving Loops: Toward Robust Emergence in Self-Modifying Agents*. Unpublished manuscript.