Emergent Complexity in Reality and Engineered AI Systems

What Makes Complex Structure Appear, Persist, and Become Actionable

A non-equilibrium, feedback, and multiscale account with operational metrics for high-complexity AI systems

Dr. Tamás Nagy

`tnagyphd@gmail.com`

Draft • March 2026

Executive Summary (Non-Technical)

Many systems in nature and technology show structure that is not obvious from their local parts. Cells, ecosystems, markets, and large software systems can all exhibit behavior that looks "more than the sum of components." This paper asks a direct question: what part of reality makes that possible?

The central claim is that emergent complexity is not a mystery property. It is a consequence of a specific regime: **open, non-equilibrium systems with feedback, memory, and multiscale coupling**. When these ingredients coexist, stable higher-level patterns can appear and persist even if low-level rules remain local.

This has a practical implication for AI engineering. A modern AI research or verification stack is itself a high-complexity object. It should be managed as a dynamical system, not as a static codebase. The key control problem is not to predict every micro-event, but to keep global behavior inside a stable and productive regime.

The paper contributes a compact framework that separates: - complexity generation, - complexity stabilization, - complexity collapse, - and complexity observability.

It also proposes operational metrics and intervention policies for real AI systems, including throughput-quality trade-offs, truthful blocker taxonomy, and branch-level control actions.

The paper does **not** claim a universal closed-form law for all complex systems. It claims a structured and testable mechanism class, plus engineering guidance for systems where full microscopic prediction is impossible but reliable macro-control is achievable.

Abstract

We study emergent complexity as a regime property of open dynamical systems rather than as an intrinsic label of objects. The main thesis is that robust emergent structure appears when five conditions are jointly active: (i) non-equilibrium drive, (ii) nonlinear interaction, (iii) feedback closure, (iv) memory-bearing state, and (v) multiscale coupling. Under this regime, macroscopic organization can become stable and information-bearing even when microscopic trajectories remain sensitive or computationally irreducible.

We provide a decomposition of complexity dynamics into four operators: generation, stabilization, collapse, and observability. This decomposition supports both conceptual analysis and system design. We then specialize to engineered AI research systems, where human cognition, agent workflows, orchestration, and formal verification form a coupled adaptive stack. In this setting, we propose measurable control surfaces for maintaining productive complexity: accepted-gain throughput, duplicate suppression, closure rate, blocker truthfulness, and intervention latency.

The result is a bridge between complexity theory and practical AI systems engineering: full micro-level prediction is often impossible, but policy-level macro-control remains tractable via explicit state abstractions and disciplined feedback policies.

---

1. Introduction

Complexity is often discussed as an adjective ("this system is complex") rather than as a regime property ("this system is operating in a complexity-generating regime"). That framing mistake hides the mechanism.

The useful scientific question is:

> Which structural conditions make higher-level order appear and persist, even when low-level dynamics are local, nonlinear, and partially unpredictable?

This paper argues that emergent complexity is not metaphysical residue. It is a regime effect of open dynamical systems with feedback, memory, and cross-scale coupling.

This matters directly for modern AI engineering. A high-capability AI research stack is not a static code artifact; it is a coupled adaptive system of: - human hypothesis generation, - agent execution, - workflow orchestration, - verification and review, - and memory surfaces.

Such systems can amplify insight, but they can also amplify noise and false closure. The same mechanisms that create emergence can create collapse.

1.1 Core thesis

Emergent complexity is best represented as a regime map:

$$\text{Emergence intensity} = \mathcal{R}(D, N, F, M, S),$$

where: - D: non-equilibrium drive, - N: nonlinearity intensity, - F: feedback topology and gain, - M: memory depth and fidelity, - S: multiscale coupling strength.

When $\mathcal{R}$ crosses a threshold, macroscopic regularities become persistent enough to support prediction, intervention, and design.

1.2 Main contributions

This paper contributes:

1. A formal emergence definition based on compression, predictive lift, and intervention coherence.
2. A four-operator decomposition: generation, stabilization, collapse, observability.
3. A regime-level phase model with testable threshold hypotheses.
4. A control-oriented mapping to engineered AI systems, including operational metrics and policy actions.
5. A concrete specialization to resident verification engines where dependency-graph structure drives complexity costs.

1.3 Non-claims

This paper does **not** claim: - a universal scalar complexity measure across all domains, - exact long-horizon prediction of microscopic trajectories in chaotic regimes, - or that emergence implies hidden teleology.

It claims a tractable mechanism class and an engineering control language.

1.4 Descriptive complexity vs good complexity

The conversation record highlights a crucial distinction that this paper adopts explicitly:

- **Descriptive complexity**: how rich, layered, and information-demanding a system slice is, independent of usefulness.
- **Good complexity**: complexity that is functionally productive, stable enough to be reused, and intervention-relevant.

This distinction prevents a common category error: high complexity is not automatically valuable. Pure noise can be descriptively complex but low in good complexity; adaptive living or cognitive systems can be high in both.

The operational target in this paper is therefore not complexity maximization, but good-complexity maximization under stability and truthfulness constraints.

---

2. Formal Setup: What Counts as Emergence

2.1 State layers and coarse-graining

Let the system have microstate $x_t \in \mathcal{X}$, memory state $m_t \in \mathcal{M}$, control $u_t \in \mathcal{U}$, and disturbance η_t:

$$x_{t+1} = f(x_t, m_t, u_t, \eta_t), \qquad m_{t+1} = g(m_t, x_t).$$

A macro-observable is induced by projection:

$$y_t = \pi(x_t), \quad y_t \in \mathcal{Y}.$$

The projection π is useful only if it is both compressed and operationally meaningful.

2.2 Emergence criteria over horizon H

We call a macro-structure emergent on horizon H if three conditions hold:

1. **Nontrivial compression**

$$H(y_t) \ll H(x_t),$$

   or an equivalent model-class compression criterion.

2. **Predictive lift over baseline projection** $\tilde{\pi}$ \$\$ I(y_t; y_{t+1:t+H})

- I($\tilde{y}t;\ y\{t+1:t+H\}) \geq \delta$_{pred} > 0.\$\$

3. **Intervention coherence**

$$P(y_{t+1} \mid do(u), y_t) \neq P(y_{t+1} \mid do(u'), y_t)$$

   for at least one control pair (u, u'), with stable direction of effect.

The third criterion rules out decorative abstractions that compress but do not support control or explanation.

2.3 Four complexity operators

Define operational complexity potential $\mathcal{K}(x_t)$ and macro-projection π.

1. **Generation**

$$G_t := \Delta\mathcal{K}^{+}(x_t).$$

2. **Stabilization**

$$S_t := \Pr\big[d(\pi(x_{t+\tau}), \varphi_\tau(\pi(x_t))) \leq \varepsilon\big].$$

3. **Collapse**

$$C_t := \Delta\mathcal{K}^{-}(x_t).$$

4. **Observability**

$$O_t := I(z_{1:t}; \pi(x_t)),$$

   where $z_{1:t}$ are measured traces.

High G_t with low S_t gives turbulence. High G_t and S_t with low O_t gives unmanageable hidden complexity.

2.4 Entropy is not the opposite of complexity

It is tempting to treat entropy and complexity as strict opposites, but that is usually false.

Let S_{ent} denote thermodynamic or information entropy, and let C_{good} denote useful emergent complexity (structured, reusable, and intervention-relevant organization). In many systems, the relation is non-monotone:

$$C_{\text{good}} = f(S_{\text{ent}}) \quad \text{with an interior maximum.}$$

This “inverted-U” intuition has three regimes:

1. **Very low entropy**: highly ordered but often rigid systems (for example perfect crystals) with low adaptive complexity.
2. **Intermediate entropy**: enough variability to explore configurations, but enough constraints to retain stable macro-structure; this is where useful complexity typically peaks.
3. **Very high entropy**: high disorder/noise with weak persistent structure, so useful complexity declines.

Hence entropy supports complexity up to a point by supplying variation, but excessive entropy destroys persistence. Emergent complexity therefore needs a balance:

$$\text{variation source} + \text{stabilizing constraints} + \text{memory}.$$

This also explains why local complexity growth is compatible with global entropy growth in open systems: local structure can increase while total entropy production across system plus environment remains non-negative.

Conceptual figure (single-panel, publication-intent). Plot normalized entropy on the horizontal axis and useful emergent complexity C_{good} on the vertical axis. The curve is inverted-U with: - a left low-entropy rigid zone, - a middle productive corridor around the interior maximum, - and a right high-entropy noise-dominated zone.

The intended takeaway is not that entropy should be minimized, but that useful complexity is maximized in a constrained intermediate regime.

2.5 From the conceptual curve to a derivation program

For empirical work, a first approximation can be written as:

$$C_{\text{good}}(s) \approx \beta_1 s - \beta_2 s^2, \qquad \beta_1, \beta_2 > 0,$$

where s is normalized entropy. This gives an interior maximizer:

$$s^\star = \frac{\beta_1}{2\beta_2}.$$

This is only a local approximation, but it immediately yields a testable object: the productive entropy corridor around $s^\star$. Higher-order fits and domain-specific nonlinear forms can replace the quadratic without changing the core hypothesis (interior maximum rather than monotone behavior).

---

3. Regime Theory

3.1 Regime coordinates and measurable proxies

The five coordinates in $\mathcal{R}$ are abstract but measurable:

- D_t (drive): normalized energy/information throughput or entropy-production proxy.
- N_t (nonlinearity): curvature or interaction intensity (for example Jacobian/Hessian nonlinearity indices).

- F_t (feedback): closed-loop gain proxy, including cycle structure and spectral radius.
- M_t (memory): retained information from past to present, e.g. $I(x_{t-k:t}; x_{t+1})$ or state-retention fidelity.
- S_t (scale coupling): cross-scale transfer intensity (graph-wavelet or multiresolution coupling proxies).

3.2 Regime index and corridor picture

A multiplicative index is often practical:

$$R_t = D_t^{\alpha_D} N_t^{\alpha_N} F_t^{\alpha_F} M_t^{\alpha_M} S_t^{\alpha_S},$$

with $\alpha_i > 0$ chosen by calibration.

Operationally, systems often exhibit three bands:

- **Laminar/underdriven**: low generation, weak novelty.
- **Productive corridor**: sustained generation with manageable stabilization.
- **Overdriven/chaotic**: high churn, rising collapse, poor closure integrity.

3.3 Proposition 1 (equilibrium-memory barrier)

If: - $D_t \to 0$ (asymptotic equilibrium drive), - $M_t \to 0$ (no retained memory), - and microdynamics are globally contractive,

then sustained positive complexity growth is impossible:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} G_t \leq 0.$$

Proof sketch. Without drive, no persistent free-energy/information injection exists. Without memory, novel structure cannot be retained. Contraction suppresses separation of trajectories. Thus no mechanism remains for sustained production of stable new macro-structure.

3.4 Proposition 2 (feedback window)

For many adaptive systems there exists a feedback-gain interval $[g_{\min}, g_{\max}]$ such that:

- $g < g_{\min}$: generation is weak ($G_t \approx 0$),
- $g \in [g_{\min}, g_{\max}]$: productive complexity corridor,
- $g > g_{\max}$: collapse frequency rises (oscillation/churn dominates).

A useful diagnostic form is:

$$\lambda_{\text{micro}} > 0, \qquad \lambda_{\text{macro}} < 0,$$

meaning local exploration is expansive while macro-policy remains stabilizing.

3.5 Proposition 3 (observability-control duality)

There exists an observability threshold $O^\star$ such that:

- if $O_t < O^\star$, false-closure rate has a positive lower bound under any finite policy;
- if $O_t \geq O^\star$ and intervention latency is bounded, macro-control error is bounded.

Interpretation: complexity that cannot be observed cannot be controlled truthfully.

---

4. Complexity Budget Dynamics

4.1 Budget equation

At a control timescale, define:

$$K_{t+1} = K_t + G_t - C_t + \epsilon_t,$$

where: - K_t is effective complexity stock, - G_t is generated usable structure, - C_t is lost structure, - ϵ_t captures unmodeled shocks.

The target is not maximal K_t at any cost, but stable positive drift under bounded collapse and bounded control error.

4.2 Productive vs degenerative complexity

A practical condition for productive operation on window W:

$$\sum_{t\in W} G_t > \sum_{t\in W} C_t, \quad \text{and} \quad \text{closure integrity}_W \geq \tau.$$

If generation exceeds collapse but closure integrity fails, the system is producing high-volume but low-truth output.

4.3 Collapse modes

Common collapse pathways:

1. **Underdrive**: insufficient novelty injection.
2. **Overdrive**: exploration noise overwhelms stabilization.
3. **Memory poisoning**: stale or incorrect retained state dominates control.
4. **Topology lock-in**: feedback graph traps policy in local loops.
5. **Measurement blindness**: observability falls below control threshold.

4.4 Early-warning signals

Operational early warnings:

- duplicate retry ratio rises while accepted gain stalls,
- blocker classification latency rises,
- reopened-closed branch ratio rises,

- escalation rate rises without gain,
- local improvements fail to propagate globally.

These are leading indicators of corridor exit.

5. Directions: What a Good Complexity Concept Can Be Used For

This section is intentionally directional rather than implementation-specific. If we obtain a robust complexity concept, it becomes a reusable scientific and engineering primitive.

5.1 Scientific direction: a cross-domain regime language

A strong complexity concept gives a common language for: - biological adaptation, - economic instability, - social coordination, - and adaptive AI ecosystems.

Instead of domain-specific metaphors, we can compare systems on shared regime coordinates (D, N, F, M, S) and ask whether they are in underdriven, productive, or collapse-prone zones.

5.2 Predictive direction: early-warning diagnostics

A good complexity concept should support early warning before visible failure. The practical use is not perfect forecasting, but horizon extension: - detect corridor exit early, - distinguish productive novelty from destructive churn, - and trigger interventions before irreversible collapse.

5.3 Design direction: complexity-aware architecture

If complexity is measurable, architecture can be optimized against it: - where to add memory and where to constrain it, - where to increase exploration and where to damp it, - which feedback loops should be amplified or bounded.

This converts "complexity management" from ad hoc tuning into model-guided design.

5.4 Control direction: policy synthesis under partial observability

A strong complexity concept can define admissible policy classes under uncertainty. The control objective is: - keep generation positive, - keep collapse bounded, - keep observability above control threshold.

This is useful for any high-complexity adaptive stack, including AI systems.

5.5 Epistemic direction: from intuition to durable knowledge

A robust complexity concept also clarifies knowledge production itself. In human-AI work, session-level transitions can be formalized:

$$\text{private thought} \rightarrow \text{shared representation} \rightarrow \text{durable artifact}.$$

This supports a principled view of idea survival, reuse, and long-horizon knowledge accumulation.

5.6 Anthropogenic complexity generation

Another high-value direction from the session export is to treat the human not only as a complex object, but as a **complexity generator**.

A useful working hypothesis is a layered-memory stack:

1. biological memory (genetic and embodied priors),
2. neural memory (learned world models and adaptive control),
3. symbolic-cultural memory (language, mathematics, writing, institutions, code).

The third layer enables cumulative non-reset dynamics across generations. This suggests an immediate research direction: model human-driven research systems as memory-amplified complexity generators rather than as static optimization agents.

5.7 Human-AI complementarity as crystallization

Another transcript-level insight worth preserving is that the human-AI pair is not best modeled as two interchangeable workers. It is an asymmetric complement: - humans inject high-semantic-entropy seeds (purpose, framing, direction), - AI amplifies formalization, expansion, and durable persistence.

A compact operator view is:

$$\mathcal{C}_{\text{HA}} : \text{seed}_{\text{human}} \mapsto \text{durable structured artifact}.$$

This gives a mechanistic reason for why session quality dominates long-horizon knowledge yield in hybrid research systems.

---

6. Derivable Consequences from a Robust Complexity Concept

Given the regime model, several concrete derivation programs become available.

6.1 Complexity-efficiency frontier

For target capability level $\mathcal{F}^\star$, define:

$$K^\star(\mathcal{F}^\star) = \inf_{m \in \mathcal{M}_{\mathcal{F}^\star}} \mathcal{K}(m).$$

This yields an irreducible complexity frontier: how much complexity is minimally required for a given capability class.

6.2 Productive-corridor condition

From the budget equation:

$$K_{t+1} = K_t + G_t - C_t + \epsilon_t,$$

a first-order productive condition follows:

$$\mathbb{E}[G_t - C_t] > 0 \quad \text{with bounded collapse variance.}$$

This gives a derivable criterion for sustained constructive growth.

6.3 Observability lower bound for controllability

If policy error is constrained by $\varepsilon_{\text{ctrl}}$, we expect a bound of form:

$$I(z_{1:t}; \pi(x_t)) \geq I^{\star}(\varepsilon_{\text{ctrl}}).$$

So one can derive minimum instrumentation requirements from target control quality.

6.4 Intervention granularity principle

Cross-scale transfer implies that intervention effectiveness depends on intervention scale. A derivation target is a law mapping intervention scale to macro impact, with failure regions where interventions are either too coarse (no leverage) or too fine (local gains that do not propagate).

6.5 Collapse taxonomy to policy map

From collapse modes (underdrive, overdrive, memory poisoning, lock-in, blindness), one can derive a policy lookup structure: - identify collapse class, - apply class-specific stabilizing intervention, - and estimate expected recovery horizon.

This is a derivable control object, not only a qualitative checklist.

6.6 Entropy-corridor width and resilience

Using the local quadratic approximation around normalized entropy s:

$$C_{\text{good}}(s) \approx \beta_1 s - \beta_2 s^2, \qquad s^{\star} = \frac{\beta_1}{2\beta_2},$$

let the acceptable degradation band be $C_{\text{good}}(s) \geq C_{\max} - \Delta$. Then the productive entropy corridor has half-width:

$$w_{\Delta} = \sqrt{\frac{\Delta}{\beta_2}}.$$

So curvature β_2 has direct operational meaning: - high β_2 -> narrow corridor, fragile regime, - low β_2 -> wider corridor, more tolerant regime.

This motivates a local resilience proxy:

$$\mathcal{R}_{\text{corr}} := \frac{1}{\beta_2}.$$

Even if the global relation is non-quadratic, local second-order fits around the operating point provide comparable corridor-width estimates.

6.7 Estimation pipeline from real logs

To make the corridor operational, we need a reproducible estimator for s_t, β_2, and w_Δ from observational data.

Step 1: Build a normalized entropy coordinate s_t.

Define a feature vector of run-window state summaries:

$$u_t = [u_t^{(1)}, \dots, u_t^{(p)}],$$

then compute normalized Shannon entropy on the induced discrete distribution or a continuous entropy proxy:

$$s_t \in [0, 1].$$

Practical choices include: - action-type distribution entropy, - branch-transition entropy, - outcome-class entropy.

Step 2: Construct a useful-complexity target C_t.

Use a weighted target that rewards durable novelty and penalizes collapse:

$$C_t = \omega_1 \,\text{novel\_durable\_yield}_t - \omega_2 \,\text{reopen\_rate}_t - \omega_3 \,\text{instability\_loss}_t.$$

The exact components are domain-specific; the key is that C_t should reflect persisting, reusable structure rather than raw activity.

Step 3: Local quadratic fit around operating point.

On a rolling window $\mathcal{W}$ centered near current s_0, fit:

$$C_t = a + \beta_1 s_t - \beta_2 s_t^2 + \varepsilon_t$$

with robust regression (Huber or quantile fit) to reduce outlier sensitivity.

Step 4: Extract corridor descriptors.

From fitted coefficients:

$$\hat{s}^\star = \frac{\hat{\beta}_1}{2\hat{\beta}_2}, \qquad \hat{w}_\Delta = \sqrt{\frac{\Delta}{\hat{\beta}_2}}, \qquad \widehat{\mathcal{R}}_{\text{corr}} = \frac{1}{\hat{\beta}_2}.$$

Step 5: Uncertainty quantification.

Estimate confidence intervals with block bootstrap over time windows to respect temporal dependence.

Step 6: Drift-aware monitoring.

Track $\hat{s}_t^\star$ and $\hat{\beta}_{2,t}$ as time series. A sharp rise in $\hat{\beta}_{2,t}$ indicates corridor narrowing and reduced resilience.

Step 7: Policy trigger layer.

Use a simple trigger set: - if $|s_t - \hat{s}_t^\star| > \kappa \hat{w}_{\Delta,t}$, enter stabilization mode; - if $\hat{\beta}_{2,t}$ rises above threshold, reduce exploration amplitude; - if uncertainty intervals widen materially, hold major policy changes.

This yields a directly executable measurement-to-control pipeline from the entropy-complexity hypothesis.

---

7. Research Program and Falsifiable Hypotheses

H1. Regime-threshold hypothesis

There exists a threshold surface in (D, N, F, M, S) above which macro-regularity persistence increases nonlinearly.

Test. Controlled sweeps in synthetic and real systems across feedback and memory settings.

H2. Productive-corridor hypothesis

Useful output is maximized in an intermediate complexity corridor, not at minimum or maximum complexity.

Test. Compare underdriven, corridor, and overdriven regimes under matched resources.

H3. Observability-control hypothesis

Below an observability threshold, truthful control quality cannot be maintained.

Test. Reduce measurement channels systematically and quantify control degradation.

H4. Complexity-frontier hypothesis

Capability classes have an irreducible complexity floor $K^\star(\mathcal{F}^\star)$.

Test. For fixed task classes, estimate minimal model/process complexity that retains target performance.

H5. Knowledge-crystallization hypothesis

Structured capture and linkage increase long-horizon idea reuse.

Test. Measure downstream reuse and artifact linkage for captured vs uncaptured ideas.

H6. Corridor-narrowing hypothesis

Systems with larger local entropy-curvature (β_2) exhibit shorter stability windows under perturbations, even when baseline output quality is similar.

Test. Estimate local curvature from intervention windows and compare corridor width against observed time-to-instability.

H7. Autonomous discovery as net complexity production

A further preserved idea is that autonomous discovery should be scored by net complexity production, not by raw activity volume.

Define:

$$\Gamma_{\text{disc}} := \mathbb{E}[G_t - C_t].$$

Systems with higher Γ_{disc} should produce more durable and reusable outputs per unit resource than systems with similar action volume but lower net complexity production.

Test. Compare matched systems by Γ_{disc} and downstream artifact survival/reuse rates.

---

8. Experimental Protocol (Cross-Domain)

8.1 Data surfaces

- state trajectories,
- interaction graphs,
- intervention logs,
- observability channel logs,
- and downstream outcome traces.

8.2 Study design

Use staged intervention studies:

1. Baseline regime identification.
2. Single-coordinate perturbations (for example feedback gain only).
3. Coupled perturbations (feedback + memory, drive + observability).
4. Post-intervention recovery tracking.
5. Rolling local-fit estimation of $(\hat{s}_t^\star, \hat{\beta}_{2,t}, \hat{w}_{\Delta,t})$.

8.3 Primary endpoints

- regime classification stability,
- early-warning lead time,
- control error under bounded interventions,
- collapse frequency and recovery time,
- efficiency relative to complexity stock,
- estimated entropy-corridor width w_Δ,
- and local curvature β_2 around operating points.

8.4 Acceptance constraints

A proposed intervention policy is accepted only if: - it improves at least one primary endpoint, - it does not materially worsen collapse risk, - and its gains remain robust across neighboring regimes.

---

9. Limitations and Open Problems

- The proposed regime coordinates are measurable but not unique; multiple proxy families may fit the same conceptual variable.
- Complexity potential $\mathcal{K}$ is model-class dependent.
- Causal identification of regime effects may require active interventions, not passive logs.
- Human strategy remains a major exogenous variable in current AI research systems.
- A full theorem-level formalization of these propositions in Lean is future work.

---

10. Conclusion

Emergent complexity is best treated as a controllable regime phenomenon, not as a vague descriptor. Reality supports emergence where openness, nonlinearity, feedback, memory, and scale coupling coexist.

For high-complexity AI systems, the practical objective is not omniscient micro-prediction. It is macro-level policy control with explicit metrics, truthful closure semantics, and bounded interventions.

This yields a concrete engineering program: - generate selectively, - stabilize honestly, - collapse safely, - observe adequately, - and control by corridor-aware policy rather than by reactive patching.

Appendix A: Notation (working)

- x_t: microstate
- m_t: memory state
- $y_t = \pi(x_t)$: macro-observable
- D, N, F, M, S: regime coordinates
- R_t: regime index
- G_t, S_t, C_t, O_t: generation, stabilization, collapse, observability operators
- K_t: effective complexity stock
- S_{ent}: entropy (thermodynamic/information-theoretic, context dependent)
- C_{good}: useful emergent complexity
- s: normalized entropy coordinate
- $s^\star$: entropy level where useful complexity is locally maximized
- β_1, β_2: local entropy-complexity fit coefficients
- w_Δ: productive corridor half-width at degradation level Δ
- $\mathcal{R}_{\text{corr}}$: corridor resilience proxy
- $\hat{s}^\star_t, \hat{\beta}_{2,t}, \hat{w}_{\Delta,t}$: rolling estimators from logs

- $\mathscr{C}_{\mathrm{HA}}$: human-AI crystallization operator
- Γ_{disc}: net autonomous discovery complexity-production rate
- $W, \mathrm{stride}, B, \kappa$: pilot rolling-estimator control parameters
- $K^{\star}(\mathscr{F}^{\star})$: irreducible complexity frontier for target capability
- $I^{\star}(\varepsilon_{\mathrm{ctrl}})$: observability lower bound for target control error

---

Appendix B: Rolling Estimator Pseudocode (paper-ready)

This appendix provides an executable pseudocode sketch for the measurement-to-control loop introduced in Section 6.7.

```
INPUTS
  logs_t            : time-ordered event/run logs
  W                 : rolling window length
  stride            : window step
  Delta             : acceptable complexity degradation level
  kappa             : corridor breach multiplier
  B                 : bootstrap replicates
  beta2_alert       : curvature alert threshold

OUTPUTS
  s_hat_star_t      : rolling estimate of entropy optimum
  beta2_hat_t       : rolling estimate of local curvature
  w_hat_delta_t     : rolling estimate of corridor half-width
  ci_t              : confidence intervals for the above
  trigger_t         : policy trigger state

PROCEDURE
  for each window w in rolling_windows(logs_t, W, stride):
    # 1) Build entropy coordinate
    u = build_state_features(w)
    s = normalized_entropy_proxy(u)        # s in [0, 1]

    # 2) Build useful complexity target
    C = omega1 * novel_durable_yield(w)
        - omega2 * reopen_rate(w)
        - omega3 * instability_loss(w)

    # 3) Local robust fit
    fit C = a + beta1*s - beta2*s^2 + eps
      using robust_regression(Huber_or_quantile)

    beta1_hat, beta2_hat = fit.coefficients
    s_hat_star = beta1_hat / (2 * beta2_hat)
    w_hat_delta = sqrt(Delta / beta2_hat)
```

```
    # 4) Block bootstrap uncertainty
    ci = block_bootstrap_ci(w, model_spec, B)

    # 5) Trigger logic
    if abs(current_s(w) - s_hat_star) > kappa * w_hat_delta:
      trigger = "stabilize"
    elif beta2_hat > beta2_alert:
      trigger = "reduce_exploration"
    elif ci_width(ci, beta2_hat) is "wide":
      trigger = "hold_policy_change"
    else:
      trigger = "corridor_ok"

    emit(timestamp(w), s_hat_star, beta2_hat, w_hat_delta, ci, trigger)

NOTES
  - If beta2_hat <= 0 in a window, mark the local fit as non-corridor-consistent
    and escalate to higher-order or constrained fitting.
  - Use overlap-aware validation to avoid overconfident CI under strong temporal de
  - Keep feature definitions versioned; changing feature schemas invalidates direct
    time-series comparability.
```

This pseudocode is intentionally implementation-agnostic while preserving the minimal reproducible logic needed for empirical validation and policy integration.

Appendix C: Minimal Synthetic Benchmark Protocol

This appendix defines a minimal benchmark for validating the corridor-estimation logic before full deployment on heterogeneous real-world traces.

C.1 Goal

Validate whether the pipeline can correctly recover: - an interior complexity optimum $\hat{s}^{\star}$, - local curvature $\hat{\beta}_2$, - and corridor width $\hat{w}_{\Delta}$

under controlled perturbations and known generative structure.

C.2 Synthetic generator

Simulate a latent entropy coordinate and a useful-complexity response:

$$s_t \sim \text{AR}(1) \text{ clipped to } [0,1],$$

$$C_t = a + \beta_1 s_t - \beta_2 s_t^2 + \xi_t, \qquad \xi_t \sim \text{heavy-tailed noise.}$$

Inject regime shocks: - **underdrive episodes**: reduce β_1, - **corridor narrowing episodes**: increase β_2, - **measurement degradation episodes**: increase observation noise and missingness.

C.3 Minimal benchmark scenarios

1. **Clean baseline**
    - moderate noise, stationary coefficients.
2. **Drifted optimum**
    - slow time variation in β_1, β_2.
3. **Narrowing corridor stress**
    - abrupt increase in β_2.
4. **Observability stress**
    - missing blocks and noisy proxy features.
5. **Outlier burst**
    - clustered extreme disturbances in ξ_t.

C.4 Evaluation metrics

Primary recovery metrics:

$$\mathrm{MAE}(s^\star) = \frac{1}{T}\sum_t |\hat{s}_t^\star - s_t^\star|, \qquad \mathrm{MAE}(\beta_2) = \frac{1}{T}\sum_t |\hat{\beta}_{2,t} - \beta_{2,t}|.$$

Corridor metrics: - relative error of $\hat{w}_{\Delta,t}$, - sign accuracy of corridor narrowing events, - detection lead time before synthetic instability windows.

Trigger metrics: - false-alarm rate, - missed-instability rate, - average trigger delay.

C.5 Pass/fail criteria (minimal)

The benchmark is considered minimally successful if: - $\mathrm{MAE}(s^\star)$ and $\mathrm{MAE}(\beta_2)$ remain bounded below predefined tolerance bands in baseline and drift scenarios, - narrowing events are detected with positive lead time in stress scenarios, - false alarms remain below a fixed operational threshold.

C.6 Practical escalation rule

Only after passing Appendix C should the estimator be promoted to mixed real logs. If Appendix C fails, adjust feature construction, robust-fit settings, or bootstrap design before claiming corridor-level interpretability.

Appendix D: End-to-End Pilot Rollout Checklist

This appendix turns the full concept into a multi-step executable rollout with explicit gates and practical defaults.

D.1 Phase map

- **Phase 0**: Instrumentation and schema freeze.
- **Phase 1**: Synthetic benchmark qualification (Appendix C).
- **Phase 2**: Shadow mode on historical real logs.
- **Phase 3**: Live passive mode (observe, no control action).
- **Phase 4**: Live guarded trigger mode (limited interventions).
- **Phase 5**: Full policy integration with periodic recalibration.

D.2 Master TODO list (sequential gates)

1. **Schema definition**
    - Freeze u_t feature schema version v1.
    - Define C_t components and weights (omega1..omega3).
    - Gate: schema review approved.
2. **Data quality audit**
    - Verify timestamp integrity, missingness profile, outlier policy.
    - Gate: missingness and timestamp anomalies under agreed limits.
3. **Synthetic generator setup**
    - Implement Appendix C scenarios exactly.
    - Gate: deterministic reproducibility with fixed seeds.
4. **Estimator implementation**
    - Implement rolling robust quadratic fit + block bootstrap CI.
    - Gate: unit tests pass for edge cases (beta2 <= 0, sparse windows).
5. **Synthetic qualification run**
    - Run all Appendix C scenarios.
    - Gate: pass/fail thresholds satisfied.
6. **Shadow historical replay**
    - Replay estimator on historical logs without triggering actions.
    - Gate: stable \hat{s}^\star_t, plausible \hat{\beta}_{2,t} dynamics.
7. **Trigger simulation backtest**
    - Apply trigger logic offline, score false alarms and missed events.
    - Gate: trigger quality above target floor.
8. **Live passive deployment**
    - Start online estimation and dashboarding, no policy actions.
    - Gate: two full monitoring cycles without data/fit regressions.
9. **Guarded intervention pilot**
    - Enable only low-risk triggers (hold_policy_change, reduce_exploration).
    - Gate: no material degradation in core system KPIs.
10. **Full integration**
    - Enable full trigger set and periodic recalibration cadence.
    - Gate: governance sign-off and rollback plan tested.

D.3 Practical default parameters (starting values)

These are operational defaults, not claims of optimality:

- rolling window length: W = 200 events,
- window stride: stride = 20 events,

- bootstrap replicates: B = 200,
- corridor breach multiplier: kappa = 1.25,
- alert quantile for curvature: top 15% of historical \hat{\beta}_2,
- minimum consecutive breaches before action: 2.

These defaults should be tuned per domain after Phase 2.

D.4 Governance and rollback

Minimum governance requirements: - explicit owner of schema and feature versioning, - explicit owner of trigger thresholds, - weekly calibration review in pilot phase, - rollback switch to passive mode in one operation.

Rollback triggers: - sudden false-alarm surge above threshold, - confidence interval explosion across consecutive windows, - sustained KPI degradation after trigger activation.

D.5 First practical use-cases

1. **AI research workflow reliability**
    - detect corridor narrowing before quality collapse.
2. **Autonomous system supervision**
    - convert noisy behavior into interpretable regime states.
3. **Ops risk early warning**
    - monitor instability drift from entropy-curvature changes.

D.6 What counts as pilot success

Pilot success is not only model fit; it requires: - robust recovery of corridor descriptors, - actionable early warnings with acceptable false alarms, - and measurable operational benefit under guarded interventions.

Appendix E: 4-Week Pilot Execution Sheet (Autonomous Runbook)

This appendix provides the concrete weekly execution plan with explicit deliverables, owners, go/no-go gates, and fallback actions.

E.1 Roles and ownership

- **Pilot Owner (PO)**: decides go/no-go at each weekly gate.
- **Data Owner (DO)**: owns schema, feature versioning, and data quality.
- **Model Owner (MO)**: owns estimator fitting and uncertainty logic.
- **Ops Owner (OO)**: owns trigger integration, safety constraints, and rollback.

If one person plays multiple roles, responsibilities still remain logically separated.

E.2 Week-by-week execution

Week 1 — Foundation and synthetic qualification

Objectives - finalize schema v1, - finalize C_t component definition, - run full Appendix C benchmark.

Deliverables - schema document with feature dictionary, - synthetic benchmark report (all scenarios), - initial estimator configuration file.

Go/no-go gate - proceed only if Appendix C pass criteria are met.

Fallback if failed - freeze deployment, - adjust feature set / robust fit / bootstrap settings, - rerun synthetic before any real-log move.

Week 2 — Historical shadow replay

Objectives - run estimator on historical logs in shadow mode, - estimate trigger quality offline, - calibrate first threshold set.

Deliverables - replay metrics table (\hat{s}^\star_t, \hat{\beta}_{2,t}, \hat{w}_{\Delta,t}), - false-alarm and miss-rate report, - threshold proposal v1.

Go/no-go gate - proceed only if trigger quality exceeds predefined floor and no major instability appears in replay diagnostics.

Fallback if failed - keep in shadow mode, - simplify trigger rules, - increase minimum consecutive breaches, - rerun Week 2.

Week 3 — Live passive monitoring

Objectives - run online estimation in production-like environment, - no automated control actions, - verify data/fit stability under live conditions.

Deliverables - live monitoring dashboard snapshots, - CI stability report, - drift report for \hat{\beta}_{2,t} and \hat{s}^\star_t.

Go/no-go gate - proceed only after two full monitoring cycles with no critical regressions.

Fallback if failed - return to Week 2 configuration, - isolate live-only failure source, - patch instrumentation or estimator robustness.

Week 4 — Guarded intervention pilot

Objectives - enable limited low-risk triggers, - measure operational impact and safety.

Allowed trigger set - hold_policy_change, - reduce_exploration.

Deliverables - intervention impact report (before/after), - safety report (false alarms, missed instabilities, KPI drift), - full integration recommendation memo.

Go/no-go gate - move to full integration only if guarded interventions improve at least one primary endpoint and do not cause material KPI degradation.

Fallback if failed - immediate rollback to live passive mode, - threshold retuning, - rerun guarded pilot window.

E.3 Practical weekly checklist (compact TODO board)

1. Week 1 done -> synthetic pass evidence attached.
2. Week 2 done -> replay and trigger diagnostics signed.
3. Week 3 done -> live passive stability signed.
4. Week 4 done -> guarded pilot impact signed.
5. Final decision -> promote or hold with explicit reasons.

E.4 Fast default run cadence

- Daily: estimator health and CI-width check.
- Twice weekly: threshold review.
- Weekly: gate review and rollback-readiness check.

E.5 Absolute stop conditions

Stop autonomous promotion immediately if any holds: - sustained KPI degradation after trigger activation, - confidence interval instability across consecutive windows, - unexplained schema drift or timestamp integrity failure.

Appendix F: Pre-Registration Pack (Filled Template)

This appendix is the pre-registered protocol for empirical evaluation. Its purpose is to prevent post-hoc target drift and to make replication straightforward.

F.1 Study identity

- **Study title**: Entropy-Corridor Estimation and Control in Adaptive Complex Systems.
- **Protocol version**: v1.0.
- **Primary scope**: validate corridor estimation and trigger value under synthetic, replay, and guarded live settings.
- **Registration timestamp field**: use first-run deployment timestamp of Phase 1.

F.2 Locked hypotheses (H1-H7)

1. **H1 Regime-threshold**: macro-regularity persistence rises nonlinearly beyond a threshold surface in (D, N, F, M, S).
2. **H2 Productive-corridor**: useful output is maximized in an intermediate complexity corridor.
3. **H3 Observability-control**: below an observability threshold, truthful control quality degrades materially.
4. **H4 Complexity frontier**: target capability classes have an irreducible complexity floor $K^\star(\mathcal{F}^\star)$.
5. **H5 Knowledge crystallization**: structured capture and linkage improve long-horizon reuse.
6. **H6 Corridor narrowing**: larger local curvature (β_2) predicts shorter stability windows.
7. **H7 Net discovery production**: higher Γ_{disc} predicts higher durable yield at matched resource levels.

No additional primary hypotheses may be introduced after Phase 2 starts.

F.3 Primary and secondary endpoints (locked)

Primary endpoints - corridor-estimator recovery error: - MAE($s^\star$), - MAE(β_2), - trigger performance: - false-alarm rate, - missed-instability rate, - detection lead time, - operational outcomes: - collapse frequency, - recovery time, - efficiency relative to complexity stock.

Secondary endpoints - CI width stability over windows, - drift rate of $\hat{s}_t^\star$ and $\hat{\beta}_{2,t}$, - policy-action volume vs benefit ratio.

F.4 Data partitions and evaluation order (locked)

1. **Synthetic partition** (Appendix C) -> must pass first.
2. **Historical replay partition** -> no live actions.
3. **Live passive partition** -> estimation only.
4. **Guarded intervention partition** -> limited trigger set.

Order may not be changed. Advancement requires prior gate pass.

F.5 Baselines and ablations (mandatory)

Baselines - static-threshold baseline (no rolling fit), - monotone-entropy baseline (no interior-optimum model), - random-trigger baseline (rate-matched).

Ablations - remove curvature term (beta2 = 0 model class), - remove bootstrap uncertainty gating, - remove consecutive-breach condition.

F.6 Statistical analysis plan (locked)

- Rolling robust quadratic fit for core estimator.
- Block bootstrap for uncertainty intervals.
- Window-level comparisons with matched-resource normalization.
- Predefined significance and effect criteria:
    - confidence level: 95%,
    - minimum practical improvement threshold set before each phase starts.

If assumptions break (for example persistent non-positive $\hat{\beta}_2$), switch to predeclared fallback model class and report separately.

F.7 Reproducibility controls (hard constraints)

- fixed schema version (v1) during a phase,
- fixed feature-construction code hash per run batch,
- explicit random seeds for synthetic generation and bootstrap,
- immutable run manifest including:
    - code commit,
    - dependency lock,
    - parameter file,
    - dataset snapshot identifier,
    - output artifact hashes.

Any schema or feature change triggers a new protocol subversion (v1.x) and requires phase restart from the nearest valid gate.

F.8 Exclusion and missing-data policy (locked)

- windows with timestamp corruption are excluded and logged,
- missingness beyond threshold enters observability-stress labeling,
- outliers are handled by robust fit, not silent deletion,
- all exclusions are reported with counts and reasons.

F.9 Decision rules

Promote phase only if: - primary endpoint gates are satisfied, - no absolute stop condition is active.

Hold phase if: - confidence intervals widen materially, - trigger quality is inconclusive.

Rollback if: - sustained KPI degradation follows trigger activation, - schema integrity is compromised.

F.10 Reporting template (required)

Each phase report must include: - hypothesis status table (H1-H7: supported/unsupported/inconclusive), - endpoint table vs baselines and ablations, - uncertainty and sensitivity summary, - failures and negative results, - next-phase recommendation with explicit rationale.

Appendix G: Lane-Close Discussion Capture

This appendix records the key conceptual conclusions from the current lane so they remain in the durable research object rather than only in chat history.

G.1 Complexity vs entropy (final clarified stance)

- Entropy and complexity are not strict opposites.
- Useful emergent complexity is typically non-monotone in entropy.
- The practical model in this paper is an interior-maximum relation: low entropy can be rigid, high entropy can be noisy, and useful complexity peaks in an intermediate corridor.

G.2 Why regime change is a complexity question

Regime change is treated as a structural transition in the coupled dynamics: - feedback geometry changes, - memory effects change, - correlation structure changes, - and macro-level stability properties shift.

Therefore regime change is a direct empirical test surface for a dynamic complexity concept, not a side topic.

G.3 Positioning against existing complexity definitions

This work does not claim to replace classical definitions (algorithmic, information, computational mechanics, network-structural). Its claim is that many existing definitions are descriptive but not directly control-operational for adaptive systems under drift.

The added value here is the full bridge: - measurable state proxies, - corridor geometry, - uncertainty-aware estimation, - and policy trigger logic.

G.4 Network-theory relation

Network complexity remains important as structural signal. In this framework, it is a component-level view that can feed features into the corridor estimator. The distinction is: - network metrics describe topology, - corridor dynamics describe operational regime and intervention timing.

Both can be combined; they are not mutually exclusive.

G.5 External, non-internal validation direction

To avoid internal-only validity, the lane identified open-data validation paths: - public market regime-shift datasets, - public energy-demand/price instability datasets, - public system-telemetry incident datasets.

The paper's pre-registration and benchmark appendices are intended to make those external validations auditable and comparable.

G.6 Practical impact statement

Near-term practical value: - earlier instability detection, - better exploration vs stabilization decisions, - lower collapse risk under bounded intervention.

Potential high-impact outcomes: - academically: reproducible cross-domain evidence for corridor geometry, - operationally: deployable stability layer for adaptive systems, - commercially: a corridor-intelligence monitoring and control product class.

References (working list)

1. Prigogine, I. *From Being to Becoming.* W. H. Freeman.
2. Haken, H. *Synergetics: An Introduction.* Springer.
3. Ashby, W. R. *An Introduction to Cybernetics.* Chapman and Hall.
4. Holland, J. H. *Hidden Order: How Adaptation Builds Complexity.* Addison-Wesley.
5. Crutchfield, J. P., and Young, K. "Computation at the onset of chaos." In complexity literature.
6. Mitchell, M. *Complexity: A Guided Tour.* Oxford University Press.
7. Wolfram, S. *A New Kind of Science.* Wolfram Media.
8. Relevant internal verification and workflow artifacts (to be finalized in a companion appendix).