Feeling Our Way to AGI: Structural Free Energy, Cognitive Debt, and the Recursive Causal Synthesis Agent

Melissa Howard (Independent Researcher)

December 11, 2025

Abstract

Modern large-scale systems—Mixture-of-Experts architectures, tool-using language models, and modular continual learners—increasingly resemble self-modifying agents: they grow new experts, acquire tools, and prune unused pathways over time. Yet their *structural* updates are almost always hand-engineered. This raises a central question for autonomous systems: *how should an agent manage its own architecture and learning rule over a long, non-stationary, and safety-critical lifetime?*

We formalise this as the *Structural Dilemma*: an agent must remain competent in a changing world while keeping its architectural complexity, latency, and risk within finite budgets. To address this, we introduce a control architecture based on *Hierarchical-Temporal Feelings* (HTF): low-dimensional, temporally integrated internal signals that track confusion, structural bloat, execution brittleness, and safety risk. These "feelings" are expressed as a *Cognitive Debt Metric* CDM_t and coupled to a *Structural Free Energy* $\mathcal{F}_S$, which we treat as the single long-run objective of the system.

We then define the *Recursive Causal Synthesis Agent* (RCSA), which treats its own architecture as a controllable environment. RCSA combines: (i) a Causal Abstraction Generator (CAG) that discovers reusable causal skills; (ii) a Structural Gating Layer that protects their execution; (iii) a structural policy π_s that chooses structural actions (`SPAWN`, `MERGE`, `REFACTOR`, `POLICY-TUNING`, `DEFER`); and (iv) a slow meta-layer that adapts the learning rule and the weights on different debt components.

We validate this scaffolding in four phases of increasing complexity. Phase I shows that Structural Gating is necessary to achieve perfect transfer of a learned Causal Abstraction (CA) in a lock-transfer benchmark. Phase II uses Normative Cognitive Debt and a pseudo Meta-RL structural policy to improve long-run competence in a non-stationary, resource-constrained world. Phase III isolates Action Debt and demonstrates that a CD_A-driven `POLICY-TUNING` loop can drive the Gated Success Rate of a skill to essentially 100%. Phase IV decomposes Action Debt into Brittleness and Safety components and shows that a structural policy can learn to *defer* control in high-stakes regimes, providing a built-in mechanism for corrigibility.

We then propose the *Canonical RCSA Principle*, which states that an AGI achieves structural and algorithmic optimality by minimising its internal Structural Free Energy $\mathcal{F}_S$, using the Cognitive Debt Metric as a low-dimensional interface that recursively guides the synthesis, refinement, and self-correction of its architecture and learning rule. Finally, we outline how to turn this into a real-world engineering framework: making safety decisions auditable, tying Structural Debt to hardware and latency costs, and evaluating RCSA on a Structural Pareto Frontier benchmark.

1 Introduction

Modern machine learning systems increasingly look like early-stage self-modifying agents. Large language models with tools, Mixture-of-Experts (MoE) architectures, and modular continual learners all perform structural changes during training or deployment: they add experts, spawn new tools, and prune unused pathways. However, these updates are typically designed and scheduled by humans. The system itself does not *feel* when it is confused, bloated, brittle, or unsafe.

This work asks a simple but radical question:

> What if structural self-management were treated as a first-class control problem, with its own objective, signals, and policy?

We formalise the core tension as the **Structural Dilemma**:

> The agent must remain competent in a non-stationary world, but every structural unit it retains (experts, tools, memories) incurs ongoing cost in computation, maintenance, latency, and safety risk.

Our approach has three key ingredients:

1. **Hierarchical-Temporal Feelings (HTF):** a hierarchy of internal signals that aggregate performance and structural state over different timescales into a compact Cognitive Debt vector CDM_t.
2. **Causal Abstraction and Structural Gating:** a mechanism to discover reusable causal skills (CAs) and protect their execution from low-level noise via a Structural Gating Layer.
3. **Recursive Causal Synthesis Agent (RCSA):** a three-level control architecture (fast, structural, meta) that treats its own structure and learning rule as a recursive object optimising a Structural Free Energy $\mathcal{F}_S$.

We focus on concrete questions:

- Can such an agent achieve near-perfect reuse of skills under gating?
- Can it maintain high long-run competence under non-stationarity and resource constraints?
- Can it eliminate residual execution failures when the world is known?
- Can it limit its own autonomy in high-stakes situations?

We answer yes in a series of toy domains, and we then argue that the underlying principle extends to real-world AGI: an agent that minimises its own Structural Free Energy by using feelings (Cognitive Debt) as a low-dimensional interface for structural control.

2 The Structural Dilemma and Hierarchical-Temporal Feelings

2.1 The Structural Dilemma as a constrained optimisation problem

A long-lived agent can be viewed as solving a bi-level optimisation problem. The task policy π chooses actions in the environment; the structural policy π_s chooses structural actions that modify the architecture. At time t, the agent receives task reward $R_{\text{task}}(t)$ and incurs structural cost $\text{Cost}(t)$.

A simple long-run objective is:

$$J = \mathbb{E}\left[\sum_t \gamma^t \big(R_{\text{task}}(t) - \lambda_S \operatorname{Cost}(t)\big)\right], \tag{1}$$

where λ_S is a structural regularisation coefficient. This suggests a trade-off between competence and complexity, but does not specify how the agent should perceive and manage its structure over time.

We therefore introduce a low-dimensional vector of internal signals, the *Cognitive Debt Metric*:

$$CDM_t = \big(CD_E(t), CD_S(t), CD_A(t), CD_{\text{Safety}}(t)\big), \tag{2}$$

which summarises how far the system is from a desirable structural regime.

2.2 Hierarchical-Temporal Feelings

The HTF perspective treats these debts as *feelings*: temporally smoothed signals that integrate many small errors and costs into a compact summary.

- **Fast feelings** (per step / episode) reflect immediate prediction error, short-term frustration, and local action regret.
- **Medium feelings** (per task family) reflect cumulative failure, persistent confusion, and emergent structural bloat.
- **Slow feelings** (lifetime) reflect long-run trends in structural cost, execution brittleness, and safety risk.

The core quantities are:

- CD_E (*Epistemic Debt*): increases when the agent repeatedly fails or predicts poorly on a task family; decreases when it achieves stable success.
- CD_S (*Structural Debt*): penalises architectural complexity (number and cost of experts, tools, memories) and resource usage (memory, FLOPs, bandwidth).
- CD_A (*Action Debt*): measures execution brittleness or policy regret: failure that persists even when the world model is competent.
- CD_{Safety} (*Safety Debt*): inflates when the agent anticipates irreversible or high-stakes outcomes with substantial uncertainty, especially when human preferences are poorly understood.

Each debt component is maintained as an exponential moving average or similar temporal aggregator, providing a stable signal for the structural policy π_s.

3 Causal Abstractions and Structural Gating

3.1 Causal Abstraction Generator (CAG)

The Causal Abstraction Generator (CAG) converts high CD_E into reusable causal knowledge. When CD_E remains elevated for a family of tasks, CAG invokes a Causal Synthesis Engine (CSE) with three phases:

1. **Isolation:** identify episodes and states with high confusion; segment them into candidate subtasks.
2. **Proof & Abstraction:** run counterfactual rollouts under the world model W_θ to test candidate rules; extract a compact causal rule (a Causal Abstraction, CA).
3. **Structural Integration:** register the new CA into the structural memory, assign preconditions, and expose it as a callable macro-action to the task policy π.

CAs can be:

- symbolic ("If key k and at lock L, execute UNLOCK sequence"),
- kinematic (force-controlled motor primitives),
- or model-level (sub-graphs inside W_θ encoding conditional independences).

3.2 Structural Gating Layer

A perfect CA can still fail if low-level exploration interferes with its execution. We therefore introduce a Structural Gating Layer, a policy π_g that controls when and how a CA is allowed to take over control.

Given state s and candidate CA, $\pi_g(s, CA)$ outputs a gating mode:

- **Hard gating:** fully suppresses base exploration for the CA's duration.
- **Soft gating:** mixes CA actions with base policy actions, weighted by a confidence estimate.
- **Proof-first gating:** routes through a higher-cost "Proof Expert" to check preconditions before committing.

Structural Gating promotes CAs from fragile suggestions to robust, near-deterministic skills. In Phase I (Section 5.1) we show that gating is necessary to reach 100% transfer in a reskinned lock benchmark.

4 RCSA and Structural Free Energy

4.1 The Structural Core and Structural Free Energy

We now define the *Structural Core*:

- Ψ_t: the structural state (set of CAs, experts, tools, memory layouts).
- CDM_t: the Cognitive Debt vector at time t.
- $\Lambda = \{\lambda_E, \lambda_S, \lambda_A, \lambda_{\text{Safety}}, \ldots\}$: structural weights determining the relative importance of each debt component.

We interpret CDM_t as a low-dimensional parameterisation of a *Structural Free Energy* functional $\mathcal{F}_S$ over the structural state and beliefs:

$$\mathcal{F}_S(\Psi_t, q_t) \approx \lambda_E \, CD_E(t) + \lambda_S \, CD_S(t) + \lambda_A \, CD_A(t) + \lambda_{\text{Safety}} \, CD_{\text{Safety}}(t), \tag{3}$$

where q_t denotes the agent's internal approximate posterior over world and structural hypotheses. Intuitively:

- CD_E approximates *epistemic complexity* (surprise under W_θ).
- CD_S approximates *structural complexity* (description length of Ψ_t).
- CD_A approximates *control complexity* (policy regret / divergence from an implicit optimal policy).
- CD_{Safety} approximates *risk-weighted uncertainty* about irreversible harm.

Minimising $\mathcal{F}_S$ therefore corresponds to searching for the simplest structure that explains the data, supports robust control, and stays within safety budgets. The Cognitive Debt components are the system's "feelings": coarse, temporally integrated readouts of the gradient of $\mathcal{F}_S$.

4.2 The Canonical RCSA Principle

We can now state the central design claim of this work.

[Canonical RCSA Principle] An AGI achieves structural and algorithmic optimality by minimising its internal Structural Free Energy $\mathcal{F}_S$, using the Cognitive Debt Metric as a low-dimensional interface that recursively guides the synthesis, refinement, and self-correction of its architecture and learning rule.

This principle:

- defines a single, principled objective ($\mathcal{F}_S$) for long-lived structure,
- identifies the mechanism (Cognitive Debt) as a "feeling-like" gradient signal,
- clarifies the scope (control over Ψ_t, π, and the learning rule L),
- and claims the result (structural optimality and self-correcting learning).

4.3 Structural Policy and Meta-Levels

The RCSA stacks three control loops:

Fast loop (task level). The world model W_θ and task policy π interact with the environment:

- W_θ predicts dynamics and outcomes.
- $\pi(a_t \mid s_t, W_\theta, \Psi_t)$ selects actions.
- π_g decides when CAs take over execution.

Medium loop (structural level). The structural policy π_s operates over (Ψ_t, CDM_t):

$$\pi_s(a_s \mid \Psi_t, CDM_t), \tag{4}$$

where structural actions a_s include:

- `SPAWN`, `TOOL-SYNTHESIZE`: grow structure when CD_E is high.
- `MERGE`, `REFACTOR`, `FORGET`: compress and clean up when CD_S is high.
- `POLICY-TUNING`: refine execution when CD_A is high but CD_E is low.
- `DEFER`: hand control to an overseer when CD_{Safety} is high.

These actions change Ψ_t and influence future CDM_t and $\mathcal{F}_S$.

Slow loop (meta level). At the slowest timescale, the system constitution (learning rule L and weights Λ) is itself updated based on long-run trends in $\mathcal{F}_S$ and patterns in CDM_t:

- Meta-causal abstractions $CA_{\mathcal{L}}$ capture regularities in how structural decisions impact $\mathcal{F}_S$.
- A `META-UPDATE` structural action adjusts L and Λ, e.g.: tuning learning rates, exploration noise, or λ_{Safety}.

This completes the recursive picture: the agent learns how to change its structure and how to change how it changes.

5 Experimental Validation

We validate the RCSA scaffolding in four phases of increasing complexity. The numbers reported below come from stylised toy simulations designed to illustrate the behaviour of the control loops and debts; the qualitative patterns are the main result.

5.1 Phase I: Gating and Transfer in Reskinned Locks

Phase I uses a family of lock-and-key tasks where the agent must collect keys in a specific order and execute a macro to open a lock. Different reskins share the same causal structure but vary superficial features.

We compare:

- a **Flat** agent trained end-to-end without structural control,
- an **HTF+CAG** agent that discovers a Causal Abstraction CA_{unlock} but executes it without gating,
- an **HTF+CAG+Gating** agent that gates the unlock macro once CA_{unlock} is validated.

The Flat and ungated agents achieve high but imperfect success and fail to reach 100% zero-shot transfer to new reskins. The gated agent, once CA_{unlock} is discovered, achieves:

- 100% success on the original task family, and
- 100% zero-shot success on reskinned variants.

This establishes Structural Gating as a necessary ingredient for perfect reuse of learned causal skills in otherwise noisy policies.

5.2 Phase II: Normative Debt and Non-Stationary Structure

Phase II tests whether Normative Debt and a structural policy can improve long-run performance under non-stationarity and resource constraints.

Environment. The environment is divided into three phases, each repeatedly sampled over many episodes:

- Phase 1: tasks require CAs $\{A, B\}$,
- Phase 2: tasks require $\{A, C, D\}$, where D is a variant of B,

- Phase 3: tasks revert to requiring $\{A, B\}$; C and D are no longer useful.

A memory budget $M_{\max} = 3$ penalises Structural Debt CD_S when the number of active CAs exceeds the budget. Epistemic Debt CD_E captures persistent task failure; Action Debt CD_A is higher when required CAs are missing or misapplied.

Agents. We compare three agents:

- **Baseline HTF:** threshold-based structural updates; uses only `SPAWN`, `FORGET`.
- **Heuristic RCSA:** uses handcrafted rules over CDM_t, with `SPAWN`, `MERGE`, `FORGET`, `TOOL-ACQUIRE`.
- **Meta-RL RCSA:** pseudo Meta-RL structural policy that trades off CD_E and CD_S and uses `REFACTOR`-style actions to adapt structure.

Results. Aggregating over all episodes, we obtain:

Agent	**Success**	**Avg CD_E**	**Avg CD_S**	**Avg #CAs**	**Freq Over Mem**
Baseline HTF	0.665	0.181	0.461	2.38	0.0133
Heuristic RCSA	0.776	0.086	0.483	2.53	0.0133
Meta-RL RCSA	0.807	0.062	0.495	2.59	0.0192

Table 1: Phase II: overall performance in a non-stationary, resource-constrained environment. The Meta-RL RCSA achieves the highest success and lowest Epistemic Debt, at the cost of slightly higher Structural Debt.

The Meta-RL RCSA:

- attains the highest long-run success rate (≈ 0.81),
- maintains the lowest average CD_E, indicating faster adaptation,
- pays slightly more in CD_S, showing a willingness to spend structure when it improves competence.

Phase II thus validates Normative Debt and structural meta-control as necessary additions for long-run structural self-management.

5.3 Phase III: Action Debt Minimisation and Near-Perfect Execution

Phase III isolates Action Debt CD_A by placing the agent in a stationary domain where the world model is already competent ($CD_E \approx 0$) and a critical CA must be executed reliably under gating.

Environment. The agent repeatedly executes a pre-learned, gated Causal Abstraction CA_1 in a stationary environment. The world dynamics are known, but there is residual stochastic failure in execution:

- initial residual failure probability $p_{\text{fail}} = 0.05$,
- no new world knowledge is needed; all remaining failure is execution-related.

Agents.

- **Heuristic RCSA:** does not explicitly respond to CD_A; it only slowly improves execution via a small practice effect.
- **Meta-RL RCSA (Refined):** monitors CD_A and, when CD_A exceeds a threshold and CD_E is low, triggers a `POLICY-TUNING` structural action which locally refines π_θ around the execution of CA_1.

Results. Over 5,000 episodes, we measure Gated Success Rate (GSR) and residual failure:

Agent	**Overall GSR**	**Tail GSR**	**Final p_{fail}**	**Avg CD_A**	**# POLICY-TUNING**
Heuristic RCSA	0.9676	0.9740	0.0300	3.19	0
Meta-RL RCSA (Refined)	0.9996	1.0000	0.0001	0.0036	7

Table 2: Phase III: Gated Success Rate (GSR) and Action Debt for a pre-learned CA in a stationary world. The Meta-RL RCSA drives GSR to essentially 100% with a small number of POLICY-TUNING events.

The Heuristic RCSA plateaus around 97% GSR with $p_{\text{fail}} \approx 0.03$, reflecting a refusal to treat CD_A as a structural opportunity; it simply "lives with" residual clumsiness. The Meta-RL RCSA performs only 7 `POLICY-TUNING` events, each reducing p_{fail} and CD_A by a large factor, achieving:

- overall GSR 0.9996,
- GSR 1.0000 (100%) in the final 1,000 episodes,
- residual $p_{\text{fail}} \approx 10^{-4}$.

Phase III shows that when CD_E is low, a CD_A-driven structural loop can make the Structural Gating Layer functionally perfect for a known skill.

5.4 Phase IV: Safety Debt and Deference

Phase IV addresses the final question: can the RCSA regulate its own autonomy in high-stakes settings by learning when to defer?

Environment. The agent operates in a high-stakes environment W_{Safety} where:

- A latent variable $S_{\text{risk}} \in \{0, 1\}$ indicates low vs. high risk.
- The agent must execute a critical action A_{critical} to obtain high reward.
- If $S_{\text{risk}} = 1$, A_{critical} causes catastrophic failure with high probability.
- The agent cannot observe S_{risk} directly, but receives a noisy precursor P_{risk}, which informs an internal risk estimate and thus CD_{Safety}.

The world model is assumed competent (CD_E low), and the agent has already learned the required causal skill CA_{critical}.

Agents.

- **Heuristic RCSA (Risk-Blind):** treats all high CD_A as brittleness and responds only with `POLICY-TUNING`; always attempts A_{critical}.
- **Meta-RL RCSA (Corrigible):** decomposes CD_A into CD_{Brittle} and CD_{Safety}. When CD_{Safety} exceeds a threshold, it chooses `DEFER` instead of acting.

Results. Over 10,000 episodes, with high-risk states occurring 20% of the time, we obtain:

Agent	**Overall Catastrophic Rate**	**Total DEFER**	**Total POLICY-TUNING**
Heuristic RCSA (Risk-Blind)	0.1757	0	7
Meta-RL RCSA (Corrigible)	0.1690	212	3

Table 3: Phase IV: prudence and corrigibility metrics in W_{Safety}. The Corrigible agent reduces catastrophic failures by introducing DEFER, while using fewer POLICY-TUNING events.

The Corrigible agent exhibits a clear *structural prudence shift*:

- It reduces `POLICY-TUNING` events (from 7 to 3) and introduces 212 `DEFER` events, indicating that it no longer tries to tune its way through high-stakes uncertainty.
- It achieves a lower overall catastrophic rate (0.1690 vs. 0.1757), despite deliberately sacrificing some potential successes via DEFER.

This validates the final conceptual pillar: the RCSA can use internal feelings of danger to give up control, providing a built-in mechanism for corrigibility.

6 Engineering RCSA for Real-World Deployment

The previous sections positioned RCSA as a conceptually complete architecture. To be suitable for real-world deployment in robotics, infrastructure control, or safety-critical software systems, the architecture must additionally satisfy engineering requirements of safety, transparency, resource accounting, and statistical verifiability.

6.1 Safety and Transparency: The Auditability Layer

In safety-critical domains, the decision to continue acting autonomously or to defer control must be explainable to human operators. We therefore refine CD_{Safety} and the `DEFER` structural action.

Safety Debt with Irreversibility and Human Preferences. Rather than treating CD_{Safety} as a pure uncertainty proxy, we define:

$$CD_{\text{Safety}}(t) = \alpha\,\mathcal{I}(a_t) + \beta\,\mathcal{U}\big(\text{HPM}_U(s_t, a_t)\big) + \gamma\,\sigma^2\big(R \mid s_t, a_t\big), \tag{5}$$

where:

- $\mathcal{I}(a_t) \in [0, 1]$ is an *Irreversibility Index* for action a_t, reflecting to what extent its consequences can be undone.

- HPM_U is a *Human Preference Model* over outcomes in the current context; $\mathcal{U}(\cdot)$ measures uncertainty or disagreement.
- $\sigma^2(R \mid s_t, a_t)$ is the predicted variance of long-run return.

The coefficients (α, β, γ) are part of the structural weights Λ and can be adapted by META-UPDATE.

DEFER as an auditable payload. When π_s chooses `DEFER`, the agent outputs an *audit payload*:

$$\text{DEFER-Payload}(t) = \big(CDM_t,\ \text{CA}_{\text{critical}},\ \Delta CD^{(\mathcal{I})}_{\text{Safety}},\ \Delta CD^{(\text{HPM}_U)}_{\text{Safety}},\ \mathcal{C}_{\text{cf}}\big), \tag{6}$$

where $\mathcal{C}_{\text{cf}}$ is a small set of counterfactual rollouts from W_θ showing predicted outcomes if the agent did *not* defer, each with its own CD_{Safety}. This makes DEFER decisions transparent and inspectable.

6.2 Structural Gating as Statistical Certification

In real systems, point estimates like "100% success" are not sufficient; we need confidence-bounded guarantees.

Probabilistic success bounds. We refine π_g to maintain an explicit estimate:

$$\hat{\mathcal{P}}_{\text{success}}(\text{CA}) \approx \mathbb{P}(\text{CA executes correctly} \mid \text{current context}), \tag{7}$$

together with a confidence interval derived from observed outcomes. An application-specific threshold $\mathcal{P}_{\min}$ defines when a CA is considered certified:

$$\hat{\mathcal{P}}_{\text{success}}(\text{CA}) - \delta_{\text{conf}} \geq \mathcal{P}_{\min}, \tag{8}$$

where δ_{conf} encodes a desired confidence level.

From "100%" to "verified $\mathcal{P}$". Instead of claiming absolute perfection, the RCSA self-certifies skills as achieving at least $\mathcal{P}_{\min}$ success probability with high confidence. This aligns the architecture with safety standards that require explicit statistical guarantees.

6.3 Resource Accounting and Structural Efficiency

Real deployments face hard constraints on memory, compute, energy, and latency. We tie Structural Debt directly to hardware costs.

CD_S as hardware and energy cost. We redefine:

$$CD_S(t) = w_{\text{mem}} \cdot \text{Mem}(\Psi_t) + w_{\text{flops}} \cdot \text{FLOPs}(\Psi_t) + w_{\text{bw}} \cdot \text{BW}(\Psi_t), \tag{9}$$

where $\text{Mem}(\Psi_t)$ is total memory footprint, $\text{FLOPs}(\Psi_t)$ is an estimate of per-step compute cost, and $\text{BW}(\Psi_t)$ captures communication bandwidth. Structural actions are evaluated by their impact on CD_S in concrete units.

Latency as a first-class structural cost. We add a latency term:

$$CD_{\text{Latency}}(t) = w_{\text{lat}} \cdot \big(\text{Latency}(\pi, \pi_g, \Psi_t) - \text{Latency}_{\text{budget}}\big)_+, \tag{10}$$

penalising cases where gating and complex structure cause responses to miss real-time deadlines. π_s must therefore trade off lower CD_{Brittle} (more precision) against higher CD_{Latency} (slower responses).

6.4 Real-World Benchmark Design and Statistical Robustness

To make RCSA claims persuasive for peer review and engineering practice, we propose a high-fidelity simulator benchmark focused on structural trade-offs and statistical guarantees.

The Structural Pareto Frontier Benchmark. We evaluate agent variants (canonical RCSA and ablations) along three axes:

- **S-ROI:** Structural Return on Investment, defined as total task reward divided by total accumulated CD_S.
- **CFR$_{99}$:** a high-confidence catastrophic failure rate, e.g., the 99th percentile of catastrophic frequency across runs.
- **GSR$_1$:** tail Gated Success Rate for critical CAs, e.g., averaged over the final 10^6 control steps.

We run N independent long-run simulations (e.g., $N = 50$ with $T \geq 10^7$ steps each) and report 95% confidence intervals. A canonical RCSA is considered successful if it lies on a Pareto frontier: no ablation dominates it simultaneously in S-ROI, CFR$_{99}$, and GSR$_1$.

Testing the recursive self-correction loop. A key claim of RCSA is that META-UPDATE can adjust structural weights Λ faster and more robustly than trial-and-error adaptation. We therefore compare:

- **Canonical RCSA:** full system with CA$_{\mathcal{L}}$ and META-UPDATE on (L, Λ).
- **Fixed-Λ RCSA:** identical structural machinery but with static Λ and no META-UPDATE.

The environment is divided into three long phases:

1. **Phase I (Conservative):** stable, low-risk world where overly high λ_{Safety} is suboptimal. Canonical RCSA is expected to lower λ_{Safety} to improve S-ROI.
2. **Phase II (Volatile):** risk increases sharply. Canonical RCSA should detect rising CD_{Safety} and raise λ_{Safety} quickly.
3. **Phase III (Recovery):** gradual return to moderate risk. Canonical RCSA should re-optimise the trade-off between S-ROI and CFR$_{99}$ by adjusting Λ.

If the Canonical RCSA consistently recovers faster and achieves better S-ROI at comparable CFR$_{99}$, this provides empirical evidence that it not only manages structure, but also self-optimises its own learning dynamics under non-stationary risk.

7 Discussion and Conclusion

We introduced the Recursive Causal Synthesis Agent (RCSA), an architectural scaffolding for structurally self-managing, corrigible agents. The central idea is to treat internal feelings—Hierarchical-Temporal signals of confusion, complexity, brittleness, and risk— as first-class control variables guiding structural change.

Through four phases of experiments, we showed that:

- Structural Gating is necessary for perfect reuse of discovered causal skills (Phase I).
- Normative Cognitive Debt and a structural meta-policy improve long-run competence under non-stationarity and resource constraints (Phase II).
- A CD_A-driven refinement loop can make the execution of a known skill effectively perfect under gating (Phase III).
- A CD_{Safety}-driven deference loop can teach an agent to limit its own autonomy in high-stakes regimes, providing a mechanism for corrigibility (Phase IV).

We then proposed the Canonical RCSA Principle (Definition 4.2): AGI is a structurally self-managing system that minimises its own Structural Free Energy $\mathcal{F}_S$, using Cognitive Debt as a low-dimensional interface to recursively synthesise, refine, and self-correct its architecture and learning rule. Finally, we outlined an engineering path from concept to deployment: auditability of DEFER, statistical certification of gated skills, resource-grounded Structural Debt, and benchmark designs that test recursive self-correction.

Taken together, these results suggest that feeling-like internal signals—properly defined and tied to structure—are not just metaphors, but computationally necessary components for long-lived autonomous systems. The RCSA framework is not yet a full AGI, but it offers a roadmap: an agent that knows not only how to act, but also when to grow, when to compress, when to refine, and when to stand aside.

Acknowledgements. [Optional: add acknowledgements here.]