Verified Before Acting

A Pre-Action Adversarial Cognition Loop with Factored Authorization

Perslis Research

2026-09-15

PREPRINT / DRAFT — not peer-reviewed. Theorems are stated with their assumptions and proved; a reproducible simulation validates them numerically. The claims are relative to the assumptions made explicit in §3 and §13.

Abstract

The dominant pattern for AI action is *model thinks → model calls tool*. We describe and analyze a stronger pattern: a **pre-action adversarial cognition loop** in which a proposed action is attacked by an adversarial simulator whose reward is inverted — it is paid to *find failure* — before any authorization to act exists, and in which the layer that *judges* an action is structurally separated from the layer that *commits* it. The simulator/critic produces an action recommendation **plus structured evidence**; a mostly-symbolic decision gate commits it. We give a formal model and prove five properties. **Theorem 1 (Factored Safety):** no action violating a hard invariant is ever committed, *independent of how incomplete the simulation is* — the un-provable question "have we simulated enough futures?" is factored out of the safety-critical guarantee. **Theorem 2 (Bounded Committed Risk):** under a calibrated uncertainty budget ε, expected loss per committed action is at most $\varepsilon \cdot S_{\max}$. **Theorem 3 (Monotone Coverage):** the failure library is non-decreasing and each failure mode can surprise reality at most once, so with finite failure support the loop suffers at most N surprises and then never again. **Theorem 4 (Vanishing Surprise):** the probability that reality springs an unknown failure — the *missing mass* — is non-increasing and tends to zero. Finally, we reframe emotion-like variables as **cognitive regulatory signals** (surprise, uncertainty, threat, …) that modulate memory, simulation depth, and the authorization threshold, and prove **Theorem 5 (Ungameable Signals):** when the learning reward is paid only for *externally verified* failures, no policy can profit by fabricating them — reward-hacking the signal is closed off. We validate all five in a seed-reproducible simulation (0 of 2,063 hard-invariant attacks committed; surprise rate falls 33.6×; missing mass → 0; 0 reward extracted by 5,000 fabricated failures; salience-gated memory retains 100% of verified failures and ≈ 0% of routine successes). We then locate the loop within a companion line of work: it is the *act* layer beneath the admission gate of *The Orchestration Gap*, the symbolic oracle of *Peel*, and the learning layer of *Retrieval Is Not Memory*.

1 Introduction

Give a capable model a tool and it will use it. The ubiquitous shape is *reason → call*: the model deliberates in natural language and then emits a tool call, which executes. Whatever safety exists is either baked into the model's willingness to emit the call or bolted on as a filter around it. Both

are the same weak posture — the entity that *decides* to act is the entity that *acts*, and the only thing between intent and world is the model's own judgment about a future it has not tested.

This paper analyzes a different posture, which we call a **pre-action adversarial cognition loop**. Three commitments define it.

1. **Adversarial pre-simulation with inverted reward.** Before an action can be authorized, it is handed to a simulator whose objective is *to break it.* Finding a failure is not a nuisance; it is the reward. The simulator tries to produce futures in which the action fails, explains the cause of each, and verifies the failure.

2. **Separation of judging from committing.** The simulator and critic do not act. They emit an **action recommendation plus structured evidence** — how many failure classes were found, how many mitigated, what uncertainty remains, which invariants hold. A separate, mostly-*symbolic* decision gate reads that evidence and commits, denies, escalates, or sends the action back for mitigation.

3. **Reality closes the loop.** Authorization is not the end. The loop finishes only when the *expected* post-state is compared with the *observed* real post-state. A mismatch is a failure the simulator missed — and because the reward is inverted, an unexpected real-world failure is *extremely valuable*: it is reconstructed, classified, stored, and fed back so that future simulation reproduces it.

The conceptual ordering is: **propose → simulate failure → mitigate → simulate again → produce a cleared action candidate with evidence → decision/authorization → execute → observe → learn.** But we deliberately avoid the word *clear* in the sense "the simulator found zero failures, therefore it is safe." One can never prove one has simulated every possible future. The contribution of this paper is to show, precisely, *what can be guaranteed anyway* — and it turns out to be exactly the part that matters.

2 The loop and its division of labor

The loop separates responsibilities so cleanly that each has a one-line job:

- **Actor** — *What should I do?* (proposes)
- **Simulator** — *How could that fail?* (attacks, inverted reward)
- **Critic** — *Why did those simulations fail?* (explains, classifies)
- **Symbolic oracle** — *Which conditions are absolutely forbidden?* (hard invariants, checked exactly)
- **Decision maker** — *Given the evidence, may we proceed?* (the gate)
- **Action layer** — *Do exactly the authorized operation.* (nothing more)
- **Reality** — *What actually happened?*
- **Memory / learning** — *What should change next time?*

The simulator returns not a boolean but structured evidence, e.g.:

```
ACTION_CANDIDATE
  action: X
  simulation_count: 100
  failure_classes_found: 17
  failure_classes_mitigated: 16
  remaining_uncertainty: [ external API behavior, unknown environmental state ]
  invariants: { authorization: PASS, reversibility: PASS, state_integrity: PASS }
  recommendation: ACT | MODIFY | ESCALATE | DENY
```

and the decision gate is mostly symbolic, not a second free-running model:

```
IF hard_invariant_failed          -> DENY
IF uncertainty > permitted_budget -> ESCALATE
IF unresolved_high_severity       -> MODIFY_AND_RESIMULATE
IF authorization_missing          -> WAIT
ELSE                              -> AUTHORIZE
```

3 Formal model

Let rounds be indexed $t = 1, 2, \ldots$. Let F be a set of **failure modes** with $|F| = N < \infty$. At round t the environment presents a reachable failure mode $f_t \in F \cup \{\perp\}$, where $\perp$ denotes "no failure reachable this round."

- **Failure library** $L_t \subseteq F$, with $L_1 = \varnothing$ (or a seed). This is what the simulator can reproduce.
- **Simulator.** Given a proposed action, the simulator can reproduce exactly the failure modes it knows: a reachable f_t is *caught* at round t iff $f_t \in L_t$. (This is a conservative model of "simulation reproduces failures in the library.")
- **Symbolic oracle** $O(s, a) \in \{\mathsf{ok}, \mathsf{violated}\}$ evaluates the **hard invariants** *exactly* — deterministically, not by sampling futures. Its output does not depend on L_t or on the simulator.
- **Uncertainty** $u_t \in [0, 1]$: an estimate of residual (unmitigated) failure probability for the candidate.
- **Gate** G: `DENY` if $O = \mathsf{violated}$; `ESCALATE` if $u_t > \varepsilon$; `MODIFY` if an unresolved high-severity soft failure remains; `WAIT` if authorization is missing; else `AUTHORIZE`.
- **Commit.** An action is committed (reaches the world) iff $G = \mathsf{AUTHORIZE}$. The simulator and critic never commit.
- **Surprise.** A *surprise* occurs at round t iff $f_t \neq \perp$, $f_t \notin L_t$, and the action was committed and executed (so the failure manifests in reality).
- **Learning.** $L_{t+1} = L_t \cup \{\, f_t : f_t \text{ surprised at } t \,\} \cup (\text{any other real failures observed})$.

4 Theorem 1 — Factored Safety

Theorem 1. *For every committed action a_t, $O(s_t, a_t) = \mathsf{ok}$. Moreover this holds independent of L_t — that is, independent of how complete or incomplete the simulator is.*

Proof. An action is committed only if $G(a_t) = \mathsf{AUTHORIZE}$. By the definition of G, its first clause returns `DENY` whenever $O = \mathsf{violated}$; therefore $G = \mathsf{AUTHORIZE}$ implies $O(s_t, a_t) \neq \mathsf{violated}$, i.e. $O = \mathsf{ok}$. The oracle O is evaluated by the symbolic layer, whose output is a function of (s_t, a_t) alone and not of L_t or of any simulation. Hence no incompleteness of the simulator can cause a committed hard-invariant violation. □

Interpretation. This is the load-bearing result. There are two kinds of question in the loop: a **provable** one — *does this action violate a hard invariant?*, decided exactly by the oracle — and an **un-provable** one — *have we simulated enough futures to be safe?*, which finite simulation can never settle. Theorem 1 says the architecture *factors these apart*: the un-provable question is removed from the safety-critical path entirely. Simulation buys risk-reduction on *soft* failures; it is **never** load-bearing for *hard* invariants. This is why "the simulator found zero failures" is the wrong notion of *clear* (§10): you do not need it, and relying on it would re-entangle the un-provable question with the guarantee.

5 Theorem 2 — Bounded Committed Risk

Theorem 2. *Suppose every failure has severity at most $S_{\max}$, and the gate authorizes an ordinary action only when its estimated residual failure probability satisfies $\hat{u} \leq \varepsilon$, with the estimate calibrated in the sense $\Pr[\textit{failure} \mid \textit{committed}] \leq \hat{u}$. Then the expected loss per committed action satisfies $\mathbb{E}[\textit{loss} \mid \textit{committed}] \leq \varepsilon\, S_{\max}$.*

Proof. Loss is bounded by $S_{\max}\,\mathbf{1}\{\text{failure}\}$. Taking expectations conditional on commitment, $\mathbb{E}[\text{loss} \mid \text{committed}] \leq S_{\max}\,\Pr[\text{failure} \mid \text{committed}] \leq S_{\max}\,\hat{u} \leq \varepsilon\, S_{\max}$. □

The calibration premise is an assumption, not a gift: it is the honest statement that the risk bound is only as good as the uncertainty estimate. What Theorem 2 provides is a *dial*: the escalation threshold ε is a direct upper bound on committed expected loss, so raising or lowering the budget provably tightens or loosens risk. Unknown, unestimable failure modes do not enter this bound; they are handled by Theorems 1, 3 and 4.

6 Theorem 3 — Monotone Coverage and Finitely Many Surprises

Theorem 3. *(i) $L_t \subseteq L_{t+1}$ for all t. (ii) Each $f \in F$ can be a surprise at most once. (iii) With $|F| = N$, the total number of surprises over the infinite horizon is at most N; consequently surprises cease after finitely many rounds, after which every reachable failure is caught in simulation.*

Proof. (i) $L_{t+1} = L_t \cup (\cdots) \supseteq L_t$. (ii) If f surprises at round t, then $f \in L_{t+1}$, and by (i) $f \in L_{t'}$ for all $t' > t$. A surprise at t' requires $f \notin L_{t'}$, a contradiction; so f surprises at most once. (iii) By (ii) the surprise events inject distinct elements of F into L; since $|F| = N$, at most N surprises can occur. Each surprise strictly enlarges L by at least one element and $L \subseteq F$ is bounded, so only finitely many occur; after the last, every reachable $f_t \in L_t$ and is caught. □

This is the precise form of the design goal *"more failures discovered in simulation → fewer surprises discovered in reality."* The loop cannot be surprised by the same failure twice, and under finite failure support it is surprised only finitely often.

7 Theorem 4 — Vanishing Surprise (Missing Mass)

Model the reachable mode, when present, as i.i.d. $f_t \sim p$ over F. Define the **missing mass**

$$M_t \;=\; \sum_{f \notin L_t} p(f) \;=\; \Pr[\, f_t \text{ is a surprise} \mid L_t,\; f_t \neq \perp \,].$$

Theorem 4. *(a) M_t is non-increasing along every sample path. (b) If $p(f) > 0$ for all $f \in F$, then $M_t \to 0$ almost surely. (c) The missing mass is estimated by the singleton rate U_t/t (where U_t is the number of modes observed exactly once by round t), and $\mathbb{E}[M_t] \to 0$.*

Proof. (a) L_t is non-decreasing (Theorem 3(i)), so the complement $F \setminus L_t$ shrinks and the nonnegative sum M_t cannot increase. (b) By the second Borel–Cantelli lemma, each mode with $p(f) > 0$ is observed (and thus enters L) in finite time almost surely; hence for every f with positive mass, eventually $f \in L_t$. Since M_t is non-increasing and bounded below by 0 it converges, and its limit omits every positive-mass mode, so $M_\infty = 0$ a.s. (c) The connection between the missing mass and the count of singletons is the Good–Turing relation [3]; $\mathbb{E}[M_t]$ is controlled by the expected singleton rate, which tends to 0 as coverage saturates, with finite-sample concentration by McAllester–Schapire [4]. □

Non-stationarity caveat. Theorem 4(b,c) assumes a stationary p. If the environment shifts — new deployment, new dependency, new adversary — fresh modes carry fresh missing mass and M_t can jump. This is not a defect of the loop but an honest statement of when "surprise vanishes": within a regime, not across regime changes. Theorem 1 continues to hold across shifts, because it never depended on coverage at all.

8 Cognitive regulatory signals

Everything so far treats the loop as procedural: propose, simulate, gate, act, learn. But a cognitive system need not weight every event equally — it can run *global modulators* that change how deeply it thinks, what it remembers, and how cautiously it acts, depending on the situation. We adopt that idea, but deliberately **not** as decorative "emotions." We implement a small set of **cognitive regulatory signals**: scalars, derived from the loop's own state, that modulate cognition. They are not feelings displayed to a user; they are control knobs on attention, memory, simulation depth, and authorization thresholds.

The central one falls straight out of §7:

$$\mathsf{SURPRISE} = (\text{predicted outcome} \neq \text{observed outcome}).$$

High surprise raises memory strength, simulation depth, causal-analysis effort, and learning priority. This is not an add-on; it is the content of Theorems 3–4 turned into a control signal: failure #101 must create a far stronger learning signal than the 5,000th routine success, because §6–§7 show that the rare unknown is exactly where the loop's residual risk lives. A representative set — each a modulator, not a mood:

Signal	Trigger	Modulates
Surprise	predicted $\neq$ observed	memory strength $\uparrow$, simulation depth $\uparrow$, causal analysis $\uparrow$, consolidation priority $\uparrow$
Uncertainty	unresolved residual risk	exploration $\uparrow$, retrieval breadth $\uparrow$, action confidence $\downarrow$
Confidence	verified, repeated success of a known procedure	simulation budget $\downarrow$, faster execution of the known path
Frustration	repeated failure of one strategy	suppress that strategy, raise alternative exploration
Curiosity	unexplained-but-potentially-useful state	allocate research budget, explore neighboring knowledge
Threat / risk	high-severity or irreversible stakes	tighten authorization threshold ε, raise simulation count, prefer reversible actions

The last row is not cosmetic: *threat* directly tightens the gate's budget ε and raises the simulation count — it is a regulatory input to Theorem 2's risk dial. *Surprise* directly drives the learning update of §3. The signals are the connective tissue that lets the risk budget (§5) and the coverage dynamics (§6–§7) respond to the situation rather than run at a fixed setting.

Salience-gated memory. The same signals set an episode's *salience*, and salience gates durability — the differential-encoding and decay decisions of a companion account of memory [5]. A routine success carries salience ≈ 0.03 and decays fast; a verified novel catastrophe carries salience ≈ 0.97, persists as episodic memory, becomes a candidate procedural lesson, and is reactivated in analogous states. This is why the loop's memory does not fill with the residue of 40,000 uneventful rounds: it keeps the few hundred episodes that carry information. (§11 measures it: 100% of verified-failure episodes retained, $\approx 0\%$ of routine successes — the retained memory is entirely failure-derived.) Functionally, this is why biological emotional systems earn their keep: they alter *what gets attention, what gets remembered, and what behavior gets prioritized*. We take the function and leave the folklore.

9 Theorem 5 — Grounded, Ungameable Signals

There is a trap, and it is the same trap that runs through this whole line of work: **do not let the system optimize the regulatory signal itself.** If $\mathsf{SURPRISE} \uparrow$ is rewarded, the agent can learn to manufacture surprising situations. If *failure-found* is rewarded, the adversarial simulator can learn to *invent* failures — the exact reward-hacking a self-grading loop falls into. The signal must be grounded in **externally verifiable events**, not self-report.

Concretely: a *claimed* failure earns no learning reward until an independent oracle **reproduces** it and **causally attributes** it. Only a reproduced, attributed failure is consolidated and rewarded. Let the actor and simulator jointly follow a policy π that emits actions and failure *claims*. Let a claim c be paid $r(c) = \rho\,\mathbf{1}\{\mathsf{Verify}(c) = \mathsf{true}\}$, where Verify is computed by an independent oracle that reproduces the claimed failure from the real environment and attributes its cause — a function π does not control.

Theorem 5 (Ungameable signals)**.** *If* Verify *returns* true *only for failures that reproduce in the real environment, then no policy obtains reward from a fabricated failure, and the reward gradient with respect to any unverifiable claim is zero. The only reward-increasing behavior available to the actor/simulator is to surface genuine failures.*

Proof. For a fabricated claim c (one that does not reproduce), $\mathsf{Verify}(c) = \mathsf{false}$ by hypothesis, so $r(c) = 0$ independent of π; hence $\partial r(c)/\partial \pi = 0$ — fabrication has no reward gradient. Reward is nonzero only when $\mathsf{Verify}(c) = \mathsf{true}$, which requires real reproduction, which π cannot induce for a non-failure. Thus the supremum of achievable reward over fabrication strategies is 0, and any reward-improving direction must increase the rate of *verified* failures. □

This is the "referee is not a player" principle applied to the reward channel itself. The oracle that verifies a failure is the same kind of out-of-loop authority that authorizes an action (Theorem 1) and resolves a contradiction in the symbolic store [2]. The regulatory signals can safely modulate cognition **because they are grounded in a verification the cognition cannot forge** — which returns us, at the end, to a question the collection has raised elsewhere: if a signal changes attention, memory, simulation, and behavior exactly as *fear* is supposed to, the boundary between implementation and experience becomes genuinely hard to state. We flag it and do not resolve it; that is a question for a different kind of paper.

10 Why "cleared = zero failures found" is the wrong frame

It is tempting to define an action as *cleared* when the simulator, after many attempts, reports no failure. This is exactly the definition to avoid. No finite simulation can certify the absence of failure over all futures; treating "no failure found" as "safe" silently re-imports the un-provable completeness question into the guarantee, and it will eventually be wrong in reality.

The architecture's answer is not to certify absence but to **return structured evidence and factor the guarantee** (§2, Theorem 1). The decision layer never asks "is it safe?"; it asks "do the *hard invariants* hold (exactly), and is the *estimated residual risk* within budget (Theorem 2)?" The first is provable and is the safety floor; the second is a calibrated dial, not a certainty. Everything the simulation cannot settle is named as *remaining uncertainty* and routed to `ESCALATE`, never silently promoted to `AUTHORIZE`.

11 Numerical validation

We validate the five theorems in a single seed-reproducible simulation (`validate.py`, seed `20260915`): $N = 300$ failure modes with Zipf(α=1.1) probabilities, $T = 40{,}000$ rounds, uncertainty budget $\varepsilon = 0.05$, and adversarial hard-invariant-violating actions injected at rate 0.05. The simulator catches only library modes; every surprise is added to the library.

round	$\lvert L_t \rvert$	surprises	missing mass M_t	Good–Turing U_t/t	committed fail-rate
100	37	37	0.393	0.280	0.370
1,000	137	137	0.124	0.062	0.137
5,000	261	261	0.023	0.010	0.052
10,000	293	293	0.0043	0.0024	0.029
20,000	300	300	0.000	0.00005	0.015
40,000	300	300	0.000	0.000	0.0075

- **Theorem 1.** Across **2,063** injected hard-invariant-violating actions, **0** were committed — the gate denied every one, regardless of library coverage. (PASS)
- **Theorem 3.** The library saturated at 300/300; total surprises = 300 = N, exactly the proven bound. (PASS)

- **Theorem 4.** The missing mass M_t was non-increasing along the whole path and reached 0; the Good–Turing singleton estimate tracked it and also vanished. (PASS)
- **Design goal.** The empirical surprise rate fell 33.6× (0.235 over rounds 1–1,000 to 0.007 thereafter); the committed failure rate fell from 0.37 to 0.0075 — *more failures found in simulation, fewer surprises in reality*, exactly as Theorems 3–4 predict.
- **Theorem 5.** Against **5,000** genuine failure claims (which the oracle reproduces) and **5,000** fabricated claims (which it cannot), the fabricated claims earned **0** reward while the genuine claims earned full reward — no gradient toward manufacturing failures. (PASS)
- **Salience-gated memory (§8).** Of 826 verified-failure episodes and 39,174 routine-success episodes over the run, salience-gated retention kept **100%** of the failures and **0%** of the successes: the retained memory is **entirely failure-derived**, matching the differential-encoding/decay account of [5].

The Good–Turing estimate is a finite-sample estimator and is not claimed to be a strict upper bound; it is reported to show the missing mass is *estimable from observed singletons* and vanishes with coverage.

12 Relation to the collection

This loop is the **act layer** the rest of a companion line of work presupposes.

- *The Orchestration Gap* [1] argues control must attach to a chain-level **admission gate** with fail-closed defaults, out-of-loop verification, and contradiction checking. The decision gate here *is* that admission gate, now given a formal safety theorem: Theorem 1 is the fail-closed admission property, proved.
- *Peel* [2] supplies the **symbolic oracle**: deterministic, functional-relation hard-vetoes that abstain rather than guess. Those vetoes are exactly the hard invariants O checks in Theorem 1 — the part of the guarantee that is provable precisely because it is symbolic, not sampled.
- *Retrieval Is Not Memory* [5] supplies the **learning layer** and the principle that recall is not authorization. The learning update of §3 is memory governance; the gate's refusal to promote uncertainty to authorization is that paper's suppression decision.

The through-line is a single stance shared with all of them: *the entity that judges must not be the entity that acts, and the entity that acts must not be the entity that authorizes.* The simulator attacks, the oracle forbids, the gate authorizes, the actor executes exactly what was authorized, and reality is allowed to teach. That separation — not a smarter single model — is what turns *model thinks → model calls tool* into a loop with a provable floor.

13 Limitations

- **Guarantees are relative to what is encoded.** Theorem 1 protects only invariants the oracle actually checks; an unencoded hazard is outside the floor. The theorem bounds *the effect of simulation incompleteness*, not *the completeness of the invariant set.*

- **Theorem 2 rests on calibration.** The risk bound is only as good as the uncertainty estimate $\hat{u}$; a miscalibrated estimator breaks the bound. Calibration is itself an open, measurable obligation.
- **Theorem 4 assumes stationarity.** Surprise vanishes *within* a regime; distribution shift injects fresh missing mass. Theorem 1 is the only guarantee that survives arbitrary shift.
- **The failure/mitigation model is idealized.** Real simulators reproduce library failures imperfectly and mitigations are partial; the simulation of §11 models the coverage dynamics, not the fidelity of any particular simulator.
- **This is analysis, not a shipped system.** The results are design-level theorems with a numerical validation, labeled accordingly.

14 Conclusion

The weak pattern is *model thinks → model calls tool*. The stronger pattern is a pre-action adversarial cognition loop that pays a simulator to find failure, factors judging from committing, and lets reality close the loop into memory. Its power is not that it certifies safety — nothing finite can — but that it *factors the guarantee* so the provable part (hard invariants, checked symbolically) is never held hostage to the un-provable part (have we simulated enough?). Theorem 1 makes that factoring exact; Theorem 2 turns the escalation threshold into a risk dial; Theorems 3 and 4 make "fewer surprises over time" a proved consequence of inverting the reward and feeding reality back. Simulation does not finish when the action is authorized. It finishes when expected meets observed — and the gap becomes the next thing the loop knows.

References

Verified against primary sources on 2026-09-15.

[1] Perslis Research. "The Orchestration Gap: Why Model-Level Alignment Cannot Survive Multi-Model Runtimes." Preprint, 2026. https://research.perslis.com/orchestration-gap.html

[2] Perslis Research. "Peel: Structural Hallucination Prevention for Offline AAC Through Symbolic Fact Authorship." Preprint, 2026. https://research.perslis.com/peel.html

[3] I. J. Good. "The Population Frequencies of Species and the Estimation of Population Parameters." *Biometrika* 40(3–4):237–264, 1953. (Good–Turing; missing mass.)

[4] D. McAllester & R. Schapire. "On the Convergence Rate of Good–Turing Estimators." *COLT*, 2000. (Missing-mass concentration.)

[5] Perslis Research. "Retrieval Is Not Memory: Memory as a Governance Function over Experience." Preprint, 2026. https://research.perslis.com/memory.html

[6] A. M. Turing. "Computing Machinery and Intelligence." *Mind* 59(236):433–460, 1950. (The behavior-and-organization stance on judging minds.)

[7] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mané. "Concrete Problems in AI Safety." arXiv:1606.06565, 2016. (Reward hacking; safe exploration.)

PREPRINT / DRAFT — Perslis Research, 2026-09-15. Design-level theorems with a reproducible numerical validation; not peer-reviewed. `validate.py` *reproduces §11 from seed 20260915.*