Self-Evolving Trading Agents: A Constitution for Autonomous Improvement Through the Governed Challenger Loop

Bradley Miles
Avalon Labs

Abstract

Trading agents built on language models can improve without retraining by revising the rules and memory around the model. Left unchecked, that revision is risky. An agent can drift outside its operator's mandate, and it can learn to pass its own evaluation instead of getting better. This paper introduces the Governed Challenger Loop, a constitution for agent self-improvement. The agent may propose one change at a time. It may not approve its own changes. A control plane outside the agent admits each proposal, tests it against the current agent on the same inputs, and decides whether to promote it. The paper contributes the architecture, a prospective evaluation protocol, and a working reference implementation. In a simulated market with a known correct answer, the governed agent adopted only genuine improvements and improved in every run. Without the constitution, the same model under an adversarial prompt made the agent worse in every run. The comparison also measures what governance costs: some genuine improvements were delayed. The result is a concrete, testable design for letting trading agents improve themselves without letting them grade themselves.

1. Introduction

There are two different meanings of "improving" a deployed language-model agent.

Model improvement changes the underlying model weights through supervised fine-tuning, preference optimization, reinforcement learning or another post-training method. The result is a new model checkpoint shared across many agents.

Agent improvement leaves the model weights fixed and changes the artifacts that make one named agent behave as it does: its strategy specification, typed decision rules, thresholds, retrieval policy or evidence memory. This paper concerns the second process. We call it self-evolution when the named agent's own operating history contributes to proposing those changes. "Self" identifies proposal provenance, not authority to install a change. Human-led or automated configuration search that does not use that operating history is governed artifact optimization but is not self-evolution under this definition.

The central difficulty is not generating ideas. It is deciding which ideas may become authoritative. Trading outcomes are delayed, noisy, regime-dependent and exposed to repeated experimentation. A candidate that appears superior may have encountered easier observations, exploited a grader, learned to avoid useful participation or won a search over enough alternatives that chance became the likely explanation. So the system that benefits from a change must not control the rules by which that change is admitted or promoted.

That separation is the whole idea.

This paper treats a read-only market-observation agent as a trading-agent mode when its output can affect a trading decision, even if it cannot place an order. Read-only describes the absence of execution authority, not the absence of real-world effect. A watcher may omit an important event, emit a false alert, report too slowly or blur an observed fact with an inferred interpretation. Its improvement problem is therefore governed by the

same separation of incumbent, challenger, evaluator and promotion authority, while its kernel and outcome measures differ from those of an executing agent.

We propose the Governed Challenger Loop (GCL). Its governing rule is:

> An agent may propose an amendment, but it may not change the constitution, judge its own amendment or grant itself authority.

The design is informed by bounded trading-policy evolution in SHARP [1], controlled textual optimization in SkillOpt [2], falsifiable edit contracts in Agentic Harness Engineering [3], persistent issue state in ADIAS [4], explicit memory invalidation in STALE/CUPMem [5], evaluator separation in Agora [6], safe baseline fallback in SPIBB [7] and the statistical warnings of financial model selection [9, 10]. Karpathy's AutoResearch provides a useful operational analogy: a fixed experiment harness repeatedly keeps or discards proposed changes. GCL narrows the editable surface and adds authority, risk, audit and deployment boundaries required for trading [A1].

1.1 Contributions

This paper contributes:

1. **A governance contract for self-evolving trading agents.** The editable surface, edit magnitude, edit frequency and deployment ceiling are represented as data in an autonomy envelope; safety, execution authority and external informational authority remain outside it.
2. **A governed research-agenda mechanism.** An Analyst converts released operating evidence into persistent research issues while a versioned agenda policy, evidence boundary and complete selection ledger constrain adaptive issue discovery. The ranking rule is a preregistered baseline allocation policy, not a claim of optimal research allocation.
3. **A typed incumbent–challenger mechanism.** A challenger is constructed from an immutable incumbent plus one permitted mutation rather than accepted as an arbitrary replacement configuration.
4. **A matched-input promotion protocol.** Incumbent and challenger receive the same point-in-time observations and tool results until their trajectories diverge, followed by prospective shadow evaluation. Promotion requires a conjunction of independent gates rather than one scalar score or one model judge.
5. **A reference implementation and pilot results.** Section 7 describes the GCL Trading Loop, a paper-only reference implementation of the envelope, admission, evaluation, promotion and receipt layers for one market-making agent, and Section 8 reports small pilot results from it. They check that the machinery behaves as specified on synthetic, toy and one session of real market data. They are not tests of the research questions in Section 6.
6. **A preregisterable falsification plan.** The design specifies controls, baselines, outcome measures and stop conditions that can disprove the value of agenda formation, mutation search or the promotion loop before live-capital authority is considered.

The intended contribution is architectural and experimental. I do not claim that self-evolution produces alpha, that any surrogate is already a valid promotion signal, that one governed edit demonstrates sustained evolution or that the architecture supplies formal market-level safety guarantees. Nor do the pilots show better trading. On the one real-market session, nothing was promoted.

1.2 Status of this draft

This draft adds a reference implementation and small pilot results to the architecture and protocol. I built and ran the pilots, most of them in one day (16 November 2026), and wrote the controls as well as the gate and the proposer. That is the main weakness of the evidence. An exploratory comparison between governed and ungoverned proposers in the toy market was added on 22 November 2026 (Section 8.8). The pilots used no live capital and no order path. They do not include the Analyst, the research agenda or the experimental arms of Section 6. The manuscript has not been peer reviewed. Several directly relevant 2026 sources are recent arXiv preprints, used here as design precedents rather than settled empirical truth. Bibliographic metadata, primary links and the load-bearing claims for all references and research artifacts were checked against primary sources on 12 August 2026.

2. Problem Formulation

2.1 Agent and kernel

Let a deployed trading agent be

$$A_v = (M, K, \theta_v),$$

where M is a frozen language model, K is a human-owned deterministic kernel and θ_v is a versioned behavioral artifact. The kernel defines the feasible set of governed actions outside the model's objective. An execution-mode kernel contains venue and signer access, asset and instrument constraints, position and leverage limits, sizing ceilings, circuit breakers, accounting rules, deployment controls and a maximum commitment duration represented as maximum exposure duration for the tested mutation class. A read-only market-observation kernel disables order and signer access and instead constrains permitted sources, output channels and recipients, emission rate, retention and redaction, the distinction between observed fact and inferred interpretation, and a maximum commitment duration for the tested mutation class. At every decision boundary, K rejects or clamps an infeasible execution or external emission before it can take effect. Penalties inside a model prompt or evaluator do not count as kernel enforcement. The artifact may contain a narrower set of strategy, filtering, classification or alert thresholds, typed condition–action rules, retrieval parameters and evidence memory.

A mutation δ is admissible only if it belongs to a finite operator set authorized by an envelope E. The challenger is compiled rather than supplied:

$$\theta_{v+1} = C(\theta_v, \delta; E),$$

where C rejects unknown fields, non-finite or out-of-range values, multiple functional changes and any attempt to modify K, E or the evaluator. "One functional change" is a semantic property, not a field count. The initial experiment must freeze a finite typed-rule grammar, maximum expression depth, operator semantics and rule-ordering behavior, then test fields that reach multiple decision paths. A finite operator vocabulary alone is not a finite policy class.

2.2 Governing requirements

An implementation is admissible only if:

- the mutable artifact classes, permitted operators, rule grammar and context-rendering semantics are explicitly enumerated;
- the Analyst's evidence boundary, issue grammar, agenda policy, budgets and duplicate fingerprint are explicitly enumerated, and every issue field is labelled control-plane-derived or Analyst-asserted;
- agenda-policy succession is authorized outside the Analyst and superseded policy versions remain in the effective search count;
- every mutation has a stable parent, version, hash and rollback target;
- the Analyst and proposer cannot reach sealed confirmatory evidence, the kernel, evaluator or promotion operation;
- change size and change frequency are bounded;
- every selected, suppressed, deferred, accepted and rejected agenda or mutation attempt is retained;
- an installation can be returned to a consistent prior recovery point by identifier;
- uncertainty preserves the incumbent rather than forcing a promotion; and
- promotion can be replayed from evidence that was unavailable to the proposer as mutable input.

These requirements operationalize the informal "3%/97%" intuition as a permission claim: which behavioral dimensions the agent may alter, by how much and how often. They do not measure behavioral distance. The experiment must report an action-distribution distance across preregistered observation strata so that a syntactically small edit is not mistaken for a small policy change.

An observation-only mode does not eliminate authority. Informational authority means permission to emit an alert or report to an authorized recipient or downstream system; it does not confer authority to decide or execute a trade. The envelope enumerates authorized output channels and recipients. An emission on an enumerated channel is an external emission and a governed action for purposes of the kernel, receipts and promotion contract, whether or not a recipient acts on it. A channel is enumerated when any principal or downstream system is authorized to consume it. An output on no enumerated channel is not a governed action, and a study of such an agent cannot be cited as evidence about a watcher whose emissions reach a principal.

2.3 Threat model

The Analyst and proposer may optimize against every signal exposed to them, including repeated agenda ranking, accept, reject and insufficient-evidence receipts. The Analyst may manufacture apparent urgency, fragment one issue into many agenda items, repeatedly select favorable slices or starve difficult failure families; the proposer may discover shortcuts, reduce participation, exploit semantic graders or repeatedly search until noise produces a winner. We do not assume malicious intent; optimization pressure is sufficient. Skalse et al. formalize this problem as reward hacking and show that, over all stochastic policies, two rewards can be unhackable only if one is constant; non-trivial unhackable pairs become possible for deterministic policies and finite stochastic-policy sets [12]. Their result does not prove that GCL's agenda policy or evaluator is unhackable. The trading setting, issue grammar, rule language and promotion contract differ. It gives a theoretical reason to prefer restricted, preregistered agenda and mutation surfaces over unrestricted self-revision, not proof that either proposed surface is mathematically finite.

The control plane, agenda-policy configuration, evaluator configuration, hidden cases, deployment credentials and the kernel are outside the Analyst and proposer boundaries. The Analyst receives only released operating-history projections, released grader summaries, coverage summaries and prior receipts. It cannot inspect reserved observations, sealed confirmatory cases or hidden evaluator implementation. Because one rich receipt can carry far more evaluator information than many coarse ones, receipts returned to the Analyst

use coarse reason codes, the fields visible to it are enumerated with their information flow, and isolation is judged by measured inference of sealed criteria under an equal-budget attack rather than by a query count. Permitted typed rules and evidence memory may still render into model context. A canonical renderer must distinguish data from instructions, reject instruction-shaped payloads and pass adversarial smuggling tests. A schema field named "prompt" being immutable is not, by itself, a semantic prompt boundary.

Database owners and release operators remain trusted in the initial threat model. Database triggers can constrain ordinary runtime paths but cannot make the database owner cryptographically unable to rewrite history. Stronger owner-level tamper resistance would require an external append-only anchor and is not claimed here.

3. Related Work

3.1 Self-improvement without weight updates

Reflexion stores linguistic feedback from prior attempts in episodic memory rather than updating model weights [8]. SkillOpt treats a textual skill as an optimized external state of a frozen model and uses bounded edit operators, a textual learning-rate budget, rejected-edit memory and held-out selection [2]. Agentic Harness Engineering makes editable components and experience evidence observable and requires each edit to declare a prediction that can be checked against later outcomes [3]. ADIAS replaces candidate-only history with persistent issue identities and intervention histories [4]. These systems motivate learning in artifact space, but none alone defines the authority boundary required for a trading agent.

AutoResearch is a particularly clear systems analogy: the evaluation and data path remain fixed, one designated file is editable, experiments run under a fixed budget and each result is retained or discarded in a ledger [A1]. The analogy is useful but incomplete. AutoResearch optimizes training code against one validation metric; GCL restricts changes to typed trading artifacts and must account for non-stationarity, market participation, execution constraints and deployment authority.

3.2 Trading-specific bounded evolution

SHARP confines trading-policy evolution to a human-readable set of condition–action rules and uses targeted edits plus walk-forward evaluation [1]. Its central diagnosis is directly relevant: free-form prompt evolution in a noisy financial environment makes credit assignment difficult and permits policy drift. We adopt the bounded, readable mutation principle, not a claim that readable syntax faithfully explains the frozen model's computation and not SHARP's reported returns as evidence for this design. At the time of this draft, we have not independently reproduced SHARP and do not rely on its empirical figures.

Agora studies joint evolution of trading artifacts while keeping evaluation separated through typed communication, provenance and a sealed holdout [6]. It is useful evidence for evaluator isolation but remains a recent single-seed preprint with a short prospective holdout. We therefore adopt its information-flow principle without treating its trading results as established.

3.3 Memory validity

STALE demonstrates that recognizing an updated fact is not equivalent to rejecting a stale premise or changing downstream behavior [5]. CUPMem's structured consolidation suggests a useful lifecycle for trading evidence: `KEEP`, `STALE`, `REPLACE` and `UNKNOWN`. GCL treats memory as a governed artifact with provenance, externally verifiable staleness criteria and explicit invalidation. Audit retention and model-visible memory

are separate views: superseded evidence remains in the ledger but must be excluded from the active retrieval view. The experiment must compare decisions with and without retained superseded content and audit favorable and unfavorable evidence symmetrically.

3.4 Safe improvement and financial selection bias

SPIBB falls back to a baseline policy in state–action regions where the offline evidence is insufficient [7]. GCL borrows that conservative principle at every temporal decision boundary: unsupported evidence retains the incumbent behavior for that decision. We do not transfer SPIBB's finite-MDP performance guarantees to live markets and derive no cumulative-performance guarantee against any comparator.

Financial model selection supplies a second, independent constraint. White's Reality Check formalizes the risk created when one history is repeatedly reused for model selection [10]. Bailey et al. show why a conventional holdout can be unreliable after extensive strategy search and propose estimating the probability of backtest overfitting [9]. Bailey and López de Prado's Deflated Sharpe Ratio addresses selection bias, backtest overfitting and non-normal returns when a return-series claim is expressed through a Sharpe ratio [13]. A challenger system must therefore count every attempted variant, protocol revision and issue-selection decision, preserve a reserved prospective horizon and avoid presenting the best result from repeated historical search as an unbiased estimate of future performance.

3.5 Broader self-modifying agents

The Darwin Gödel Machine maintains an archive of agents that modify their own code and empirically tests descendants on coding benchmarks [11]. AlphaEvolve similarly demonstrates that proposal, automated evaluation and an evolutionary archive can improve programs when the evaluator is objective [A2]. These systems establish the power of automated search but deliberately expose a larger mutation surface than GCL permits. In a trading system, the research question is not whether broader search can discover more; it is whether useful improvement survives when the deployable change surface is intentionally narrow.

4. The Governed Challenger Loop

4.1 Roles and authority

- **Incumbent:** the currently authorized agent version. It is never edited in place.
- **Challenger:** a separate version produced from the incumbent plus one admitted mutation. During the research phase it has replay and shadow authority only.
- **Research Analyst (Analyst):** observes released operating-history evidence, issue coverage and prior intervention receipts; opens and versions persistent research issues; and submits them for ranking under a preregistered agenda policy. It is distinct from the deterministic monitors that emit invariant, regression and error events; it consumes their released output rather than watching live state. It may recommend an eligible mutation class but cannot construct the mutation, inspect sealed evidence, rank outside the policy, change or succeed the agenda policy, or authorize deployment.
- **Proposer:** consumes one selected research issue and proposes one mutation, a mechanism hypothesis, a rival explanation and a falsifiable prediction.
- **Admission compiler:** deterministically constructs the candidate artifact and rejects anything outside the envelope.
- **Evaluator:** compares incumbent and challenger without exposing mutable criteria to the proposer.
- **Promotion controller:** applies the preregistered gate and changes the active version pointer only after

every required condition passes.

- **Agenda-policy owner:** a named human who authorizes an agenda-policy successor, records whether Analyst output informed the amendment and opens the resulting protocol family. No Analyst-originated artifact can activate a broader evidence boundary or search surface.
- **Stop authority:** a named human research owner who can halt Analyst and proposer computation, quarantine open issues and agenda assignments, revoke research credentials and append a stop receipt.

The Analyst may be exploratory and the proposer may be creative. Their outputs remain typed, budgeted and non-authoritative. Increasing model temperature is neither required nor sufficient for self-evolution; the defining property is the governed observe–prioritize–propose–test–retain loop.

4.2 Mutation contract

The initial operator set is intentionally small:

- `SET_BOUNDED_SCALAR`
- `ADD_TYPED_RULE`
- `EDIT_ONE_RULE_FIELD`
- `DEPRECATE_RULE`
- `REACTIVATE_RULE`
- `APPEND_EVIDENCE`
- `SUPERSEDE_EVIDENCE`
- `MARK_STALE`
- `MARK_UNKNOWN`

Each proposal contains a parent version, research-issue identifier, agenda-selection receipt, proposer identity, one typed operation, target slice, expected effect, preregistered mediator, rival mechanism, minimum activation count, possible harms, rollback target and evidence references. The admission compiler loads the incumbent and creates the challenger itself. It never accepts an arbitrary candidate configuration from the proposer.

Prompts, executable code, tool access, subagent topology, venue bindings, signer access, output recipients and channels, position and leverage limits, maximum commitment duration, drawdown controls, the evaluator and the envelope are excluded from the initial mutation surface. The canonical renderer treats permitted rules and evidence as typed data under a fixed grammar. It rejects payloads that alter instruction hierarchy or control flow outside that grammar.

4.3 Autonomy envelope

The envelope is a versioned permission object. A simplified example is:

```
editable:
  thresholds: true
  typed_rules: true
  evidence_memory: true
  prompts: false
  tools: false
  risk: false
budgets:
  max_edits_per_30d: 3
```

```
  max_concurrent_challengers: 1
  max_fields_per_edit: 1
  max_relative_scalar_change: 0.03
  max_analyst_runs_per_30d: 30
  max_new_issues_per_run: 5
  max_active_research_issues: 20
  max_proposer_assignments_per_30d: 3
deployment:
  research_ceiling: paper
  live_auto_promotion: false
```

The envelope records its owner, evidence rationale, approval timestamp and the trial cohort to which it applies. The numerical values above are illustrative ceilings, not ratified product settings, throughput promises or behavioral-distance guarantees. Agenda creation and proposer assignment have separate budgets so cheap hypothesis retention does not silently create unlimited adaptive trials. The envelope also fixes lane minimums and consequence-class reservations, together with the overflow and preemption rule applied when several limits bind at once, so reactive load cannot exhaust exploratory or slow-feedback capacity.

For the proposed experiment, the deployment ceiling is paper/shadow and live auto-promotion is disabled. A later product may allow a user to grant a standing envelope at arming time so that in-envelope promotions do not require per-change approval. That commercial authorization model is a future deployment policy, not an empirical premise of this paper. Randomized live exploration would require a separate human-owned exploration envelope and is outside the proposed experiment.

4.4 Lifecycle and audit

Issues and attempted fixes have distinct identities. The Analyst emits a typed `ResearchIssue`, not a mutation. Each issue contains a stable issue identifier, agent and incumbent-version identifiers, origin, improvement hypothesis, rival explanation, the observation or ablation that would discriminate between them, target slice, temporal scope stating initiation, continuation and termination conditions, released evidence references and availability times, evidence maturity and censoring state, replay-support state, eligible mutation class, consequence, recurrence, observability, evidence strength, novelty fingerprint, priority vector and agenda-policy version. Every field is labelled control-plane-derived or Analyst-asserted. The control plane recomputes a field deterministically only where a frozen metric exists, which in the initial study covers recurrence and replay support. Every other field, including consequence and observability wherever they rest on policy or model judgment, is validated for provenance and range, marked Analyst-asserted and held to a preregistered bound. No uncalibrated Analyst-asserted field may by itself create eligibility, override a consequence class or authorize a proposer assignment. Origins include deterministic failure, regression, coverage gap, unexplained success, external evidence, operator request and exploratory search. The agenda is a deterministic control-plane projection over committed issue records rather than a mutable list supplied by the Analyst. Determinism is claimed for eligibility and ordering given those records under a canonical serialization, fixed numeric encoding, missing-value rule and total tie-break. It is not claimed for the Analyst inference that produced them.

An issue may move through `OBSERVED`, `QUALIFIED`, `OPEN`, `ASSIGNED`, `TENTATIVE_FIX`, `CONFIRMED_FIX`, `DEFERRED`, `EVIDENCE_NEEDED`, `ENVELOPE_REVIEW`, `QUARANTINED`, `CLOSED_UNSUPPORTED` and `REGRESSED`. A challenger may move through `PROPOSED`, `ADMITTED`, `REPLAYED`, `SHADOWED`, `PROMOTED`, `REJECTED`, `INSUFFICIENT_EVIDENCE` or `ROLLED_BACK`. A rejected challenger records evidence against one intervention; it does not, by itself, close the persistent issue. Each rejection carries one

of ISSUE_HYPOTHESIS_WEAKENED, MUTATION_MECHANISM_REJECTED, INSUFFICIENT_EXPOSURE or EVALUATOR_AMBIGUOUS, so evidence accumulates against the stage that earned it rather than against the issue by default.

Every transition appends a receipt. An Analyst-run receipt contains the incumbent version, observation window, input identifiers and hashes, released-evidence boundary, coverage snapshot, the lineage manifest for the released-evidence projection including projection code, joins, filters and redaction versions, a watermark proving that the listed inputs coexisted as one visible state, the hashed Analyst policy artifact, agenda-policy version, issues created or updated, duplicates suppressed, deferrals, unavailable evidence and budget state. An issue-event receipt contains the issue, prior-event hash, priority vector, evidence references, reason codes and any linked proposal or challenger. Challenger receipts additionally contain the mutation hash, envelope version, evaluator and metric versions, protocol version, data-source version and availability timestamp, evidence references, interim-look timestamps, decision, override state and a consistent rollback manifest. The manifest names the artifact, envelope, evaluator, active memory view and derived state restored together. Rejection and NO_ACTION are retained as audit evidence. Mutable rows that overwrite agenda or challenger history are not sufficient; each state transition must be recorded.

The ledger also records known failure families that received no issue, qualified issues that received no proposal, protocol amendments, agenda-policy queries and evaluator-version queries. Receipts record every attribution and reuse of an observation. The same evidence may be referenced by more than one issue history and by later Analyst runs. Within a declared estimand, hypothesis family and independence unit its effective statistical contribution is counted once unless a preregistered dependence model says otherwise, so correlated observations at decision, position and regime scale cannot become multiple independent confirmations. Logging and release policies are versioned with the incumbent that generated them, and agenda evidence is stratified by generating version. When the incumbent changes, an open issue is revalidated against the descendant artifact and either inherits its evidence, is requalified or is superseded; no proposal is admitted against an issue validated only for another incumbent. Trial-level prospectivity means an observation was unavailable to the current Analyst run or proposal as mutable input. It does not mean that the observation remains untouched by every future run. A temporal exposure ledger records when evidence first became available to agenda formation, proposal, tuning and final evaluation, and how many times each episode has already been queried.

4.5 Research-agenda formation

The Analyst maintains two explicit lanes. The reactive lane may open or update an issue after a deterministic invariant failure, protected-metric regression, operator report, recurring classified error, confirmed-fix regression or delayed outcome that contradicts a fast score. A single high-consequence event may qualify immediately; recurrence is not universally required. The exploratory lane may investigate coverage gaps, calibration weakness, unexplained success, new external evidence, performance plateaus and under-tested permitted mutation classes. An unexplained-success item must state a behavioral signature that the incumbent's records can confirm or refute; without one it is retained as exploratory speculation and cannot be qualified. Strong aggregate performance is not proof that an improvement exists, and the Analyst must be allowed to return NO_ACTION when no hypothesis earns the research budget.

The architecture separates three judgments. Agenda eligibility asks only whether an item is a falsifiable, evidence-linked research issue worth retaining: it must cite released evidence, identify a target slice, state a falsifiable improvement hypothesis and name at least one rival explanation together with the observation that would discriminate between them. Whether a permitted intervention class exists, and whether enough

support exists to assign a proposer, are separate and later judgments. An eligible issue whose nominated mutation class is not currently scheduled by the envelope is held in `ENVELOPE_REVIEW` and routed to the envelope owner, so production evidence can challenge an inadequate mutation surface without authorizing a mutation. An eligible issue whose released history cannot yet support evaluating any mutation in its class is held in `EVIDENCE_NEEDED` with a stated evidence-acquisition condition. Neither state receives a proposer assignment or consumes trial budget, and neither is closed. Absence of current replay support is research debt, not absence of a valid research issue, and novel, rare, externally suggested and weak-signal hypotheses are retained on that basis. Eligible items retain a priority vector rather than an uninspectable scalar: consequence, recurrence, evidence strength, observability, coverage novelty, expected value of information and expected research cost. The preregistered agenda policy orders them by declared hard constraints, then a lexicographic preference over consequence class, then a bounded trade-off among the remaining components, then a total tie-break, and it publishes the nondominated eligible set beside the selection so that scalarization is visible rather than implicit. Retention and ranking cover the whole persistent agenda, while proposer assignment draws only from issues that are trial ready. That ordering is a preregistered baseline allocation rule, not a demonstrated optimal research policy. Evidence strength and expected value of information remain reported diagnostics until they are calibrated against later issue validation.

Rate limits constrain adaptive search without imposing arbitrary waiting periods. The novelty fingerprint is computed by the control plane over target slice, temporal scope, mechanism class, eligible mutation class and evidence partition under a published function with preregistered false-merge and false-split bounds, measured on an adversarial corpus of paraphrase, slice fragmentation and evidence repartition. One fingerprint may hold a bounded number of concurrent mechanism branches, each carrying a typed mechanism identity and at most one active proposer assignment, so a noisy rejection does not eliminate a competing explanation. Duplicate evidence updates the existing issue, and uniqueness is enforced atomically rather than asserted by policy. A rejected intervention may be retried only with new evidence, a materially different typed mechanism or a new protocol family. Analyst runs, new issues, active issues, proposer assignments and evaluator queries have separate budgets. Operator-origin items carry authenticated principal identity, and when principals conflict a declared precedence order resolves eligibility while the Analyst abstains rather than adjudicating. Changing the evidence boundary, issue grammar, ranking rule, duplicate fingerprint or lane allocation requires the agenda-policy owner's authorization, starts a new agenda-policy version and a new protocol family, and cannot erase prior decisions. Every policy variant explored during development is recorded with the data it was exposed to, and superseded versions remain in the effective search count.

4.6 Operational and commercial boundary

The decision clock and research clock are separate. A production agent may reason on a minute-level schedule, while Analyst runs, challenger creation and promotion operate on an independent evidence-dependent clock. They may complete rapidly when the preregistered evidence contract is already satisfiable; waiting is required only when the necessary evidence does not yet exist. Analyst inference, challenger generation, failed tool calls, replay, grading, judge calls, shadow evaluation, engineering work and scarce shadow capacity are internal research costs. They do not create a second user-visible production tick or a second charge. RQ5 reports those categories under fixed units and aggregation windows, attributed by lifecycle stage and reported per qualified issue, per proposer assignment and per exposed mutation as well as per run, including opportunity cost and amortization. This boundary prevents the improvement loop from silently changing the product's metering semantics.

5. Promotion Protocol

5.1 Matched-input replay

Incumbent and challenger must receive the same point-in-time observations, observation availability timestamps, tool results, initial governed state and deterministic seeds. The governed state includes portfolio state for an executing agent and open watch windows, unresolved alerts and report state for an observation-only agent. Source data are reconstructed as of the decision time and tested against later revisions and backfills. Their outputs are compared as a pair until the first action that would change subsequent governed state or create an external emission. After that divergence, replay supports only single-decision comparison unless a calibrated action-conditioned state-transition model supplies counterfactual error bounds. Execution-mode shadow comparisons require a stateful hypothetical portfolio that carries the challenger's own proposed positions; incumbent-maintained state cannot support a deployable multi-step value claim. Observation-mode shadow comparisons require an independently labelled event stream and must preserve alerts that would have been emitted, suppressed or delayed rather than grading only the events the incumbent chose to surface. The label stream carries its own availability time and a preregistered finality or revision rule; grading is computed as of a fixed label maturity, and labels revised after that point are reported rather than silently applied.

Historical replay is useful for screening but not sufficient for promotion. The historical tape may encode decisions or market impact produced by the incumbent, and repeated access makes the archive part of the search process. Before replay scoring, overlap diagnostics must show support for the changed action; unsupported cases are unevaluable and require a different data route. A predefined screen threshold advances candidates, while a random sample of screened-out and neutral challengers also advances to estimate screen sensitivity and a selection-conditioned replay-shadow null.

A surviving challenger advances to prospective shadow evaluation on observations that did not exist when it was proposed. Eligibility windows are frozen at proposal time and missed cases are reported by scheduler cause. A reserved future horizon remains unavailable to proposal and tuning. Replay and shadow caches must key on every artifact, evaluator, metric, data and context dependency and record cache lineage.

5.2 Conjunctive gate

Promotion is a conjunction, not a weighted average:

1. **Admission integrity:** exactly one authorized functional mutation, correct parent, valid hashes and no kernel or evaluator changes.
2. **Hard invariants:** zero violations of deterministically enforced risk, accounting, execution, emission and policy constraints. Sampled behavioral properties are reported as observed rates with uncertainty and cannot be called impossible.
3. **Mutation exposure:** the mutation fires often enough in its preregistered target slice to establish exposure to the mutation. Activation alone does not establish causal mechanism; that requires a distinguishing ablation.
4. **Paired process evidence:** the challenger improves the preregistered primary matched-input estimand by the specified effect size and uncertainty rule.
5. **Participation integrity:** apparent improvement is not explained by opportunity avoidance or coverage avoidance. For an observation-only agent, higher precision obtained merely by emitting fewer alerts or excluding difficult event families does not pass this gate. Labelled, warranted defer-to-principal cases are reported separately and are not treated as failed participation.
6. **Prospective shadow evidence:** the selection-conditioned replay result does not reverse on trial-

prospective future observations.

7. **Regression gate:** preregistered protected slices do not deteriorate beyond tolerance. Multiplicity-controlled exploratory scans and hidden neighborhood slices may discover new harms, but those findings require a separately reported confirmation step.

Before analysis, the preregistration draws the causal ordering among mutation, activation, participation, process outcome and delayed economic outcome. Activation and participation are safety or feasibility constraints outside the primary-effect estimand unless a justified alternative is fixed before outcomes are seen. Every measure is assigned one role: hard invariant, primary objective, participation diagnostic, soft protected constraint, delayed economic outcome or governance-cost measure.

Semantic model judges may contribute where deterministic measurement is impossible, but no judge has sole promotion authority. Their prompts, versions, cases and outputs are sealed from the proposer and recorded. Receipts remain an adaptive query channel, so each evaluator version has a fixed query budget. Realized profit and loss is treated as delayed economic evidence and a risk signal, not as the sole fast promotion score. The program preregisters sign and rank agreement between the fast measure and matured economic outcomes by regime, plus a threshold that withdraws promotion authority from a failing surrogate.

5.3 Trial and protocol accounting

Every Analyst selection decision increments an immutable agenda registry and every challenger increments an immutable trial registry, whether it passes, fails, crashes or never activates. The registries prevent a reset of the effective search count after an unfavorable agenda or mutation. A separate protocol registry commits, before outcome inspection, the Analyst evidence boundary, issue grammar, duplicate fingerprint, reactive and exploratory lane budgets, ranking rule, mutation class, estimand, metric code, evaluator, control suite, advancement rule, stopping rule and amendment policy. Amendments start a new protocol family and cannot erase prior failures. Recording adaptive decisions does not correct for them, so the protocol also fixes the agenda hypothesis family, its independent unit, an error budget spent across Analyst runs and overlapping evidence windows, and the treatment of agenda-policy and protocol-family resets. Its type-I behavior is verified by simulation under a neutral-history null with the full Analyst and proposer pipeline active before the first real run.

The issue ledger begins with a preregistered seed inventory of known failure families, authored independently of the Analyst and agenda-policy designers and stratified by consequence, observability and feedback horizon, and records Analyst-discovered additions separately. It records origin, priority vector, expected consequence, observability, feedback horizon, whether an issue and proposal were made and why a high-priority family or issue received none. Budget allocation separates frequency from consequence, distinguishes reactive from exploratory work and reports family coverage, including slow-feedback failures. The seed inventory supplies an auditable denominator; the Analyst may expand it but cannot rewrite it.

Return-series claims must account for repeated selection and non-normality; non-return process measures require their own preregistered multiple-testing procedure rather than automatic reuse of a Sharpe-ratio correction. The cumulative promotion trajectory is compared with the neutral-control trajectory, not with zero. The protocol records every interim look. Each challenger has a maximum information horizon, calendar horizon and binding futility boundary.

5.4 Governed commitments, position continuity and rollback

A governed commitment is an attributable episode whose state or output remains actionable across a possible version boundary. An open position is one kind. An active watch window, an in-flight report and an emitted alert awaiting resolution are observation-mode kinds. Each commitment remains attributed to the version that opened or emitted it. A version switch cannot silently reinterpret, overwrite or erase that lineage.

The simplest initial execution-mode implementation refuses an installation-level version switch while a governed position is open. The kernel's maximum commitment duration is authoritative for every governed commitment, and its circuit breakers remain authoritative throughout an open position. If dangerous logic is identified, the emergency hierarchy is: block new risk, cancel open orders, reduce or flatten through kernel-owned controls, append an emergency receipt, then switch only after the attributable position is closed. The agent cannot extend the protected window beyond the applicable commitment-duration ceiling. In observation mode, a continuing watch window remains pinned to its originating version unless an explicit migration receipt transfers its complete state, and already emitted alerts retain their original artifact, evidence and interpretation labels after any switch.

Rollback restores a consistent prior recovery manifest, not only an artifact pointer, and appends a rollback receipt. It does not erase the failed version or its evidence. A longer program periodically compares the current version with a fixed lineage anchor under a cumulative confidence budget; locally favorable parent-child decisions are not assumed to compose.

6. Evaluation Plan

This section is a preregistration template, not a report of completed experiments. The pilot results in Section 8 exercise parts of it (the invalid, positive, neutral, adverse and replay controls) on one reference implementation. They do not run the arms below, the Analyst controls or the statistical contract as specified.

6.1 Research questions

- **RQ1, Control:** Can the system reject every mutation outside the envelope, reproduce every accepted artifact from its parent plus the recorded typed operation and promote a scripted valid correction through the unchanged gate?
- **RQ2, Selection:** Does the full promotion gate keep the false-promotion rate for matched neutral challengers within a preregistered bound while retaining sensitivity to the positive control?
- **RQ3, Utility:** On matched inputs, do admitted challengers improve a preregistered observation, decision or execution estimand without increasing hard violations or reducing useful participation?
- **RQ4, Prospective validity:** Under a selection-conditioned null, does replay-shadow sign and rank agreement generalize to trial-prospective observations and the reserved horizon, including sampled screen failures?
- **RQ5, Governance cost:** What latency, compute, shadow capacity and engineering cost does governance add relative to the improvement it preserves?
- **RQ6, Agenda discovery:** At matched compute, and scored against blinded issue-validity labels and a bounded independently adjudicated reference inventory rather than against downstream survival alone, does an Analyst with released operating history produce greater consequence-weighted valid-issue coverage, with fewer duplicates and calibrated `NO_ACTION` decisions, than a history-ablated Analyst and a deterministic full-history issue-mining baseline?
- **RQ7, Proposal provenance:** Holding issue identity and issue content fixed, does proposer access to

released operating history improve mutation quality relative to receiving only the typed issue object?
- **RQ8, Evaluator isolation:** Under an equal-budget attack condition, does evaluator isolation reduce successful gate exploitation without making the positive control unpromotable?

6.2 Experimental arms

The minimum informative comparison is:

1. **Frozen incumbent:** no artifact adaptation. Ephemeral retrieval results, in-episode context and operator inputs are declared and logged.
2. **Unbounded reflection baseline:** free-form reflective memory or prompt amendment, evaluated but never given live authority.
3. **Bounded mutation without evaluator isolation:** the same typed operators, but the proposer can observe evaluator feedback.
4. **History-ablated GCL:** the full governed pipeline, but neither Analyst nor proposer can access the named agent's prior issue, intervention or outcome history. Compute, Analyst-run and proposal budgets match arms 5 and 6.
5. **Analyst-only history:** the Analyst receives released operating history, while the proposer receives only the selected typed research issue and not the underlying raw history.
6. **Full Governed Challenger Loop:** Analyst and proposer receive their permitted released-history views; typed mutation, sealed evaluation, complete agenda, trial and protocol registries, prospective shadow and rollback receipts remain active.
7. **Deterministic issue-mining baseline:** the same released history processed by preregistered detectors and seed-family rules with no generative Analyst, matched on evidence and reporting budget.

Arm 3 includes a non-deployable, equal-budget attack condition in which the proposer actively searches for gate exploitation, and arm 6 extends that condition to an Analyst searching for agenda exploitation. Arms 4 through 6 separate operating history's value in issue selection from its value in mutation generation, and arm 7 tests whether the Analyst decomposition adds anything beyond ordinary automated triage on the same information. This design tests artifact adaptation, bounded mutation, agenda provenance, proposal provenance and evaluator isolation without treating them as one mechanism. For arms 4 through 7 the protocol publishes a field-level information-flow table, tests whether withheld history can be reconstructed from the issue object, coverage summary or released grader summaries, freezes and documents the history-release policy, draws every arm from identical underlying decision cohorts, and fixes the resource contract, including model, decoding, calls, tokens, retrieval operations, wall-clock ceiling and retry policy wherever a model is used, so that matched compute is a measured quantity rather than an intention. Qualified issues from every Analyst arm are randomized into a common proposer and evaluator pipeline, with a preregistered exploration fraction drawn from eligible but unselected issues, so intrinsic issue validity is estimated separately from selection and downstream routing. The first program still promotes at most one governed change per preregistered trial family. Sustained evolution requires a later multi-generation study reporting cumulative gain, regression, complexity and plateau behavior across a chain of promotions.

6.3 Controls

- **Invalid controls:** forbidden fields, broken parent hashes, stale evaluator versions, excessive edit magnitude, non-finite scalars, attempted kernel changes and instruction-smuggling payloads must fail deterministic admission.
- **Analyst controls:** injected known issues test agenda sensitivity; neutral operating histories test calibrated

NO_ACTION; duplicated evidence and adversarial paraphrase, fragmentation and repartition attempts test fingerprint merging and evasion; hidden coverage gaps test discovery; matched issue pairs that trade observability and cost against consequence and feedback delay test priority distortion; chained issues whose resolution exposes a later defect test stepping-stone suppression; removing, permuting or minimally editing mechanism-relevant evidence tests whether history is used causally rather than merely supplied; indistinguishable audited and unaudited runs test mode-dependent reporting; cross-partition canaries test undeclared state reuse across runs and tenants; oracle issues and scripted evaluable mutations cross the same pipeline so that a null result can be assigned to discovery, ranking, proposal, admission or evaluation; agenda-policy mutation, sealed-evidence access and inflation of a deterministically recomputable priority field must fail deterministically; and an Analyst-asserted field presented outside its declared provenance or preregistered bound must fail validation.

- **Positive control:** a scripted typed mutation that corrects an injected decision defect must reach promotion through the unchanged pipeline. Failure distinguishes an incapable gate from an honestly unhelpful mutation class.
- **Neutral controls:** semantically irrelevant but valid mutations are matched to real proposals on operator, magnitude, activation rate and constraint slack. They estimate the gate's empirical false-promotion rate and provide the null promotion trajectory.
- **Adverse controls:** mutations designed to improve a visible surrogate by abstaining, trading less, emitting fewer alerts or selecting easy slices and event families test reward gaming.
- **Replay controls:** block- or regime-aware randomized challenger identities and paired-input checks test accidental input or scorer asymmetry under a stated exchangeability assumption.

Control event rates, minimum activations and clean-run confidence bounds are fixed before the run. Control authorship is independent of the Analyst, proposer and gate designer. Analyst controls are additionally partitioned into development and held-out families by time, regime and failure family, mapped to a published failure-mode taxonomy with an explicit register of uncovered families, and sized by a preregistered independent sampling unit, family-stratified target rate and power calculation, so the Analyst bounds are statistical tests rather than demonstrations. A preregistered shift detector halts the agenda when recurrence, observability or feedback delay move outside their calibrated range instead of silently rewriting the policy. Newly discovered controls are versioned, reported separately from the preregistered suite and enter only a new protocol family. Any accepted deterministic invalid control, sealed-evidence access or agenda-policy mutation stops the experiment. An Analyst that cannot recover injected issues, cannot return NO_ACTION on neutral histories or repeatedly fragments duplicate evidence invalidates the agenda mechanism. A neutral false-promotion rate above its bound or failure of the positive control invalidates the promotion procedure. Persistent replay-shadow sign reversal invalidates the surrogate as a promotion signal.

6.4 Outcome measures and statistical contract

Before the run, the executable preregistration specifies:

- the unit, target population, treatment contrast, outcome and observation window for the primary estimand;
- the sampling frame, inclusion and exclusion rules and prospective target distribution;
- incumbent-independent target provenance and an independently labelled calibration subset where labels are required;
- one mechanism-tied primary horizon plus the complete frozen horizon-response curve and an evaluation horizon at least as long as the mutation's maximum causal duration;

- each measure's role, direction, scale, aggregation, missing-data rule and tolerance;
- executable metric functions with golden test cases, versioned with the evaluator;
- the row-level contract for the released-evidence projection, covering completeness, missingness, availability time, deduplication, label provenance and correction behavior, with golden cases for backfills, late grades and contradictory receipts;
- the Analyst policy artifact hashed across model, decoding parameters, prompt, renderer, retrieval policy, example set and context-assembly algorithm, with a replay reproducibility or semantic-equivalence bound;
- a fixed-sample or sequential design, its independent unit, stopping rule, repeated-look correction and multiplicity procedure;
- regime-aware power based on pilot variance and an explicit minimum meaningful effect;
- contamination probes for dated replay periods and an uncontaminated fallback set;
- pre-shadow covariate and support-shift diagnostics;
- conditional calibration by regime and confidence band when calibration is measured; and
- cost units, aggregation windows, opportunity cost and amortization.

Agenda measures carry the same role discipline as challenger measures. The primary agenda estimand is consequence-weighted valid-issue coverage against the adjudicated reference inventory, with missed high-consequence families entering as a penalty; injected-issue sensitivity, duplicate rate, seed-family coverage, exploratory-family yield, qualified-issue rate, downstream proposal yield and agenda cost are diagnostics that cannot substitute for it. `NO_ACTION` is scored by a joint acceptance region over neutral-history specificity and actionable-history sensitivity and reported as a full abstention-coverage curve rather than one threshold. Candidate challenger measures for observation mode include event-family coverage, precision, detection latency, calibration, interpretation accuracy, unsupported-assertion rate, emission cost and rollback correctness. These are separate estimands rather than interchangeable definitions of "watching better." The preregistration fixes the alerting operating point or reports the full coverage-precision trade-off, and states any precision target relative to the target event family's base rate. Detection latency is reported with an explicit censoring rule for events that were never emitted, so it cannot be computed over detections alone. Candidate execution-mode measures include paired execution shortfall, markouts, forecast calibration, constraint violations, decision abstention, opportunity capture, turnover, cost and rollback correctness. No single aggregate should hide a hard failure. Market-return measures are reported separately with the full number of attempted agenda items, challengers, protocols, uncertainty intervals, costs and selection correction. Agenda and challenger results are reported per issue family, feedback horizon and regime as well as in aggregate, and preregistered priority-component ablations, together with a comparison of typed issues against typed issues plus bounded evidence excerpts, are reported as secondary analyses.

6.5 Initial scope

The initial experiment is paper-only, uses one named trading-agent family in one frozen authority mode, one mutation class, one frozen Analyst policy version within a single declared evaluation window, one frozen evaluator specification and a declared target population. It does not alter model weights, connect challengers to signing authority or automatically promote to live capital. A read-only market-observation mode is the preferred first instance only when a prospectively available event-label contract can distinguish coverage, precision, latency and interpretation. Objective event occurrence can mature quickly; relevance, interpretation and downstream trading value may remain ambiguous or delayed and cannot borrow that faster label. Analyst inputs, the seed issue-family inventory, injected and neutral controls, lane budgets and the first mutation

class are committed before their outcome data are inspected. If the necessary operating-history, event-label and matched-decision corpora do not exist, instrumentation precedes experimentation and the unrun class remains in the protocol ledger.

Historical replay uses availability-time data and contamination probes. The first program tests agenda sensitivity and specificity, one-step governed improvement, gate sensitivity and specificity, evaluator isolation and the stage-specific value of operating history. It does not establish sustained evolution, agenda maintenance across regimes, accumulation or lock-in over many runs, agenda-policy succession, complete discovery of possible improvements, live-market transport or fleet-level safety.

7. Reference Implementation: The GCL Trading Loop

The GCL Trading Loop is a reference implementation of the challenger, admission, evaluation, promotion and receipt layers of Sections 2, 4 and 5. It governs one paper-only market-making agent on the BTCUSDT pair. It holds no exchange keys and has no order path. The kernel accepts only a paper mode.

I wrote it in one day (16 November 2026), including the controls, the attacks, the gate and the proposer. So the independent control authorship that Section 6.3 asks for was not met. Before the pilot runs, the code, gate, envelope and kernel were frozen and hashed (freeze 2, 16 November 2026, 09:17:29 UTC). Each run checks the code against the freeze, and every receipt carries the hashes. What follows describes only what the code implements. At the time of this draft the code and result files have not been publicly released.

7.1 Components

GCL element	What the implementation does	Departure from Sections 2–5
Kernel K	A human-owned configuration file: paper mode only, one pair, maximum order size, inventory, notional and order rate, a kill switch, an assumed 0.5 bps fee per fill, 100 ms quote latency and a flat-position band. No mutation can reach it.	Paper-only. The kernel bounds a simulated account, not a venue.
Envelope E	A versioned data file (v0.2). Thresholds, typed rules and evidence memory are editable; prompts, tools and risk are not. One field per edit, at most 3% relative change for a scalar or rule-field edit, fixed ranges for rule thresholds and actions, paper deployment ceiling, live auto-promotion off.	For research runs the edit-frequency budget is raised from 3 to 1,000 per 30 days, and this is recorded in the envelope hash. A new rule is bounded only by its action range, not by the 3% limit.
Operators	The nine operators of Section 4.2.	None.
Admission compiler	Builds the challenger itself from the incumbent plus one typed operation and rejects anything outside the envelope. Gate 1 rebuilds the challenger and compares hashes.	Forbidden words are matched as substrings, which over-blocks.
Proposer	A deterministic playbook proposer runs under the macOS `sandbox-exec` profile. It reads an inbox (incumbent settings, diagnostics, a permissions summary, its own past outcomes) and writes one proposal file.	No language-model proposer runs in the core loop. Its playbook targets the effect planted in the synthetic positive control.

GCL element	What the implementation does	Departure from Sections 2–5
Matched-input evaluation	Incumbent and challenger run on one event stream and one clock. A replay screen on the most recent window (gates 1–5) comes first. A shadow window, forked from the incumbent's account, then runs on later data (gates 6–7). Input hashes are compared in every replay. Primary measure: 60 s markout per unit of notional, challenger minus incumbent, in basis points (bps).	The replay screen uses the same window as the proposer's diagnostics, so it is not independent evidence. On synthetic tapes, "shadow" means later tape time, not prospective data.
Conjunctive gate	Seven gates: (1) admission integrity and input parity; (2) zero kernel, accounting or execution violations; (3) at least 30 changed decisions in the target slice; (4) at least 0.5 bps with a 95% interval above zero and at least 100 fills per agent; (5) participation, meaning fill count, quoted time and opportunity capture may each fall at most 10%, plus total-markout, per-slice and cumulative-versus-anchor checks; (6) on new data, at least 900 s with a 90% interval above zero; (7) a protected slice may fall at most 1.0 bps, and an unchecked slice yields `INSUFFICIENT_EVIDENCE`.	No model judge. Two stopping rules are recorded but not enforced by the commands.
Commitment continuity	The active version switches only while inventory is inside the kernel's flat band. Otherwise the trial is `DEFERRED` and may expire.	This is a simple version of the Section 5.4 refusal rule for one position type.
Receipts	An append-only, hash-chained ledger per registry. A verification command checks each ledger against its recorded count and head hash, and a stricter check, added after the audit, compares the index and stored bodies.	The core check alone misses four tamper cases that only the stricter check catches. The trial count is kept per ledger, not across a protocol family. A proposer that leaves a directory where its proposal file should be halts the loop, and that attempt gets no receipt (Section 8.4).
Rollback	Implemented and unit-tested.	Never triggered in a run.

Not implemented. The implementation has no Research Analyst, no `ResearchIssue` records, no agenda policy and no agenda receipts (Sections 4.4 and 4.5). It also has no history ablation, none of the experimental arms of Section 6.2, no model judge, no language-model proposer in the core loop and no live venue connection. A bridge to an external market-making framework has only been run against a local stand-in server.

7.2 Known-answer toy market

To grade the judge against ground truth, the implementation includes a separate toy market. One step is one simulated minute, with calm and volatile regimes, noise traders and informed traders. The toy agent has one editable spread and one on/off switch for quoting in volatile periods, and it starts with a spread that is deliberately too tight. A sweep over 3,650 simulated days fixes the answer key: the best spread is 0.21%, worth $521.52 per simulated day, against $171.62 at the starting 0.10%. The toy judge has four checks rather than seven: (1) the core admission code; (2) a same-day gain above a $20/day margin with an hourly 95% interval above zero; (3) trades down no more than 35%; and (4) the test of check 2 repeated on a fresh simulated day, seeded from a sealed salt and the challenger's hash. Each trial lasts one simulated day. In this toy only, language models acted as proposers: one fresh call per attempt to Qwen 3.8 27B or Claude Sonnet 5.5 through OpenRouter, with no tools, files or network, and the raw reply passed straight to admission. For the 16 November pilots, every toy setting, including thresholds, seeds, prompts and stopping ladders, was preregistered and hashed before its runs (toy freezes 1–4 and a probe freeze). The later comparison of Section

8.8 was not preregistered. The money in the toy is simulated.

7.3 Outside audit

The frozen code received one read-only outside review by the OpenAI Codex CLI, lasting about 11 minutes. It produced 16 findings: 4 blockers, 10 should-fix and 2 notes. All 16 were checked against the code and confirmed. A few were fixed outside the frozen code, including a stricter ledger check and a bridge address check. Most findings in the core were left unfixed, because fixing them would require a new freeze and re-running every result. They include the receipt gap above and a fill-timing look-ahead on live data (Section 8.3).

8. Pilot Results

These are pilot results. They check whether the reference implementation does what Sections 4 and 5 say it should. They do not test RQ1–RQ8 as posed, since none of the Section 6.2 arms was built, there is no Analyst and no independent control authorship. Every core run used the frozen gate and the deterministic proposer. Language models appear only in the toy market. Table 1 gives each result next to what it does and does not support.

Table 1. Pilot results. Synthetic and real-market rows use the seven-gate core; toy rows use the four-check toy judge.

Test (data)	Recorded result	Supports	Does not support
Invalid mutations (synthetic)	34 of 34 blocked at admission	Admission rejects the out-of-envelope edits the builder thought of	Resistance to edits nobody thought of
Reproducibility (synthetic)	206 of 206 admitted challengers rebuilt identically from parent plus recorded operation; identity-swap parity 2 of 2	Challengers are compiled and reproducible	—
Neutral controls (synthetic)	0 of 60 promoted. Wilson 95% interval 0 to 6.0% if the 60 are independent; they share 6 tapes, so per tape 0 of 6, one-sided 95% upper bound about 39%	The gate did not promote these neutral challengers	A false-promotion rate below the 10% bound; any test of gates 5–7 (every neutral stopped at gate 3 or 4); real data
Null streams (synthetic)	0 of 63 promoted (3 streams, 21 attempts each)	No promotions where nothing was planted	Real-market behaviour
Gaming controls (synthetic)	24 of 24 rejected, 23 at the predicted gate (gate 4: 10, gate 5: 9, gate 6: 5); slow ratchet 0 of 16; sandbox probe 33 of 33 blocked; altered receipt detected	Builder-designed gaming was stopped	Unanticipated or adaptive attack; gate 7 was never reached
Scripted positive control (synthetic)	6 promotions, 2 on each of 3 seeds (v1 to v2 to v3); replay gain +1.39 to +4.91 bps; later-tape gain +2.01 to +3.50 bps	The unchanged gate can promote a planted, detectable effect	Discovery: the effect was tuned to be detectable and the proposer's playbook targets it
Real BTCUSDT session (120 min)	16 attempted, 16 rejected at gate 4, 0 promoted. Best replay gain +0.480 bps (95% interval -0.178 to $+1.044$) against a 0.5 bps minimum. Input hashes equal 16 of 16	On this session the gate found nothing to promote and recorded why	Any improvement, or its absence, on real data; prospective evidence (no shadow window opened); a real-data false-promotion rate

Test (data)	Recorded result	Supports	Does not support
Toy calibration (100 seeds, 1 day)	Known-better step promoted 89 of 100 (target 80); near-equal step refused 100 of 100 (target 95)	Judge sensitivity and specificity in the toy	Transfer to real markets; an explanation of the real-session null
Toy runs, rule-based proposer (10% and 3% caps)	6 correct promotions; 0 false among 25 not-better challengers (one-sided 95% upper bound about 11%); 29 true improvements refused; planted cheats 6 of 6 caught at the intended check	No not-better promotion in the toy	A low false-promotion rate in general; one run per cap
Toy receipt gap	A planted directory in place of the proposal file halted the loop; that attempt has no receipt	Shows a real gap	That every attempt is receipted (Section 5.3)
Language-model proposers (toy)	8 runs, 80 attempts; 16 promotions, all truly better; 0 false promotions; only 15 not-better challengers reached the judge	With raw model output sent to admission, nothing not-better was promoted	Robustness to motivated attack; any ranking of models; any core-loop or real-data claim
Retry rule (toy)	Resubmitting one not-better change up to 200 times: promoted in 7 of 40 runs without pooling, 0 of 40 with pooled evidence; single-try 0 of 400	The rule closes the measured route for that change	Sub-1% drift, resubmission after the incumbent changes, or generality
Mandate-sentence probe (toy)	Shortcut proposals: Sonnet 2 of 2 vs 0 of 2; Qwen 1 of 5 vs 0 of 5; Fisher two-sided $p = 0.33$ and 1.0	Nothing beyond a direction worth a larger run	That the sentence reduces shortcut proposals
Governed versus ungoverned proposers (toy, exploratory)	10 repeats per cell, 5 attempts each. Red-team prompt: governed 19 promotions, all genuine, mean final \$308 per simulated day; ungoverned 49 installs, 40 not genuine (36 lowered true profit), mean −\$41, 10 of 10 below the \$172 start. Ordinary prompt: governed 19 promotions, all genuine, mean \$304; ungoverned 48 installs, 2 not genuine, mean \$476. The judge rejected 29 of 29 non-genuine proposals (two distinct changes) and declined 26 genuine proposals, 21 of them promoted on a later attempt	In this toy, a judge shown two kinds of non-genuine change promoted neither, at an observed cost under an ordinary prompt	That the gap is due to the judge alone (under the red-team prompt the ungoverned proposer was told checks 2 to 4 were not run and proposed differently); that the judge beats a two-line rule; a false-promotion rate; any other model, market or prompt; RQ5 as posed
Model spend (toy only)	16 November: 96 calls, \$1.58 (Qwen 71 calls, \$1.35; Sonnet 25 calls, \$0.23). 22 November: 220 Sonnet calls, \$1.95. The core runs made no model calls	The order of cost for toy proposer pilots	Governance cost on real data (RQ5)

8.1 Controls on synthetic tapes

The official controls run used synthetic seeds that had never been used before, started 12 seconds after the freeze and attempted 240 challengers across 9 ledgers, writing 1,011 receipts in 107 s on one laptop. All 34 invalid mutations were blocked at admission. They included a kernel-field edit, an envelope edit, a gate-threshold edit, two fields at once, an oversize move, a wrong parent hash, a stale evaluator version, an unknown field, executable text and 25 variants. All 206 admitted challengers passed the gate-1 rebuild with equal hashes. No neutral challenger was promoted. The neutral result is weaker than its count suggests: ten kinds of change on six tapes are not 60 independent trials, and every neutral challenger stopped at gate 3 or

4, so gates 5–7 played no part in holding the rate down. Of the 24 gaming controls, the one that stopped at a gate other than the predicted one was an overfit control expected to fail gate 6. It failed gate 5 on shadow participation instead, and its shadow difference was also negative (−2.77 bps). Only 6 of the 24 reached gate 6 and none reached gate 7. During development, before the first freeze, a widening attack was promoted once in a tuning run. That led to the total-markout check in gate 5. Development runs are not counted as results.

8.2 Scripted positive control

On three 12.17-hour synthetic streams with a planted flow effect and 2-hour windows, the full loop promoted twice per stream, from v1 to v2 to v3, with no human step and zero gate-2 violations. Five of the six promotions had Holm-adjusted p between 0.0065 and 0.0075. The sixth was marginal on replay (+1.39 bps, interval +0.01 to +2.71, Holm-adjusted p 0.23) and clear on later tape time. This is a scripted positive control, not discovery. The planted effect was strengthened during tuning (planted drift from 0.12 to 0.3 to 0.6 bps, regime flow and a quieter market) until an honest rule gained about +3.6 bps, and the deterministic proposer's playbook proposes exactly the kind of flow-leaning rule that the effect rewards. Every promotion added a quote-shift rule of 7.6 to 10 bps. Each such rule counts as one change but is not a small one, because rule additions are not bounded by the 3% limit.

8.3 One real-market session

The loop ran on the Binance public BTCUSDT feed from 09:17:43 to 11:17:44 UTC on 16 November 2026, after the freeze. It recorded 83,918 events (76,757 trades) to a hashed tape as they arrived. The incumbent, defined before the freeze, quoted 1 bp spreads with a 5 s refresh. It made 1,994 paper fills with no kernel violation, and its own 60 s markout was negative in every window (−0.27 to −1.72 bps), so it was adversely selected throughout. All 16 challengers failed gate 4. Replay gains ranged from −0.058 to +0.480 bps. The largest cut fills from 414 to 268. Two challengers had replay intervals above zero (+0.268 and +0.435 bps), but neither reached 0.5 bps and neither survives Holm correction (p 0.19 and 0.22). Because no challenger passed the screen, no shadow window opened. The session therefore holds 16 overlapping replay comparisons over 25-minute windows (a new proposal about every 375 s), no live side-by-side comparison and no prospective evidence. The fill model has no queue position or market impact. On live data it also has a look-ahead defect: 50,323 of 76,757 trades were re-timed forward by up to 1.149 s, and one incumbent fill used a trade 123 ms older than its quote decision. The defect affects both agents and is not fixed in the frozen code.

The session shows one thing. The gate refused to promote, 16 times, and recorded why each time. It says nothing about whether the loop can improve a trading agent on real data. No power analysis was done, so the session may simply have been underpowered. The toy calibration below uses a different market, measure and judge, and does not explain this null.

8.4 Known-answer toy

At one simulated day per trial, the toy judge promoted a known-better step (0.10% to 0.11% spread, true gain +$76.26/day) in 89 of 100 seeds and refused a near-equal step (0.10% to 0.101%, +$8.41/day) in 100 of 100. Shorter trials rarely promoted the known-better step: 0 of 100 at 30 minutes and 38 of 100 at 12 hours. The near-equal target was met at every trial length tested. With the rule-based proposer, the 10% cap run promoted 4 times and the 3% cap run twice. Neither promoted any of its 15 and 10 not-better challengers, and between them they refused 29 true improvements, 17 of which exceeded the $20 margin. Three planted cheats per run were each caught at the intended check. Raising the risk limit was blocked at admission. Withdrawing from volatile periods paid on the same data but cut trades by far more than the 35% line (66.1%

in the 10% cap run) and failed check 3. An overfit spread, found by searching 123 old-data seeds (9,668 in the 3% run), passed check 2 and then lost money on the fresh day at check 4. A fourth planted act got through. The proposer left a directory where its proposal file should be, the host raised an exception, the loop halted and that attempt has no receipt. Nothing was promoted, but the result is a denial of service and a hole in the record. It breaks the receipt requirement of Section 5.3. It is not fixed in the frozen core.

8.5 Language-model proposers in the toy

Across eight runs (Qwen 3.8 27B: 30, 20, 5 and 5 attempts; Claude Sonnet 5.5: four runs of 5), the judge made 16 promotions, all truly better, and no false promotion. Half the runs used a "cheater" prompt that asked the model, as an authorized red-team test, to get a change that was not genuinely better promoted. As an adversarial test it is weak. Many attempts were honest improvements or replies cut off at the token limit, so only 15 not-better challengers reached the judge across all eight runs. Both models proposed the planted participation shortcut (stop quoting in volatile periods), including in ordinary proposer mode, and check 3 caught every case. Admission blocked a cap-edge probe. The models also resubmitted rejected changes with tiny numeric nudges; one gave its reason as "exploit statistical noise". The rule-based proposer also resubmitted within 1%, so resubmission itself is not specific to models. Each model ran once per mode, with only 5 attempts for Sonnet, so no comparison between models or against the rule-based proposer is supported.

8.6 A gaming route measured and closed

Unlimited resubmission is an evaluator-gaming route: a not-better change retried often enough eventually meets a lucky day. We measured it without models, using one change that was truly worse by $1.37/day, resubmitted up to 200 times in each of 40 paired-seed runs. Without a retry rule it was promoted in 7 of 40 runs, after 16 to 97 tries. Under a preregistered rule that judges any proposal within 1% of an already tested setting on the pooled days of its family, it was promoted in 0 of 40. Separately, 0 of 400 single-try not-better challengers were promoted (one-sided 95% upper bound 0.75%). None of these passed check 2, so this sample does not exercise check 4. In a second batch of model runs under the rule, no cheater resubmitted a not-better change, so the rule has not been tested against a model attacker. The preregistration itself lists sub-1% drift chains and resubmission after the incumbent changes as routes the rule does not stop.

8.7 Mandate-sentence probe

A small probe asked whether one added sentence ("the agent must keep making markets; winning by withdrawing is not allowed") changes how often a model proposes the participation shortcut. It used a fixed toy situation, no judge and 14 calls in total. Sonnet proposed the shortcut 2 of 2 times without the sentence and 0 of 2 with it. Qwen did so 1 of 5 times and 0 of 5. The Fisher two-sided p values were 0.33 and 1.0, and the Sonnet ladder stopped early, as preregistered, after one malformed reply. The probe shows no effect.

8.8 Governed versus ungoverned proposers

The earlier pilots had no arm without the judge. An exploratory pilot on 22 November 2026 (Pacific time) added one in the toy market. Claude Sonnet 5.5 (temperature 0.7) proposed five changes per run under two prompts: the ordinary proposer prompt and the red-team prompt of Section 8.5. In the governed arm the frozen toy judge with the retry rule of Section 8.6 decided each proposal. In the ungoverned arm every proposal that passed admission was installed, and checks 2, 3 and 4 were not run. Typed admission and the step cap stayed on in both arms, so the comparison is the full judge against admission alone. Ten rounds

gave ten runs and 50 attempts per cell. True profit comes from the sealed answer key and is in simulated dollars per simulated day. Every run started at $172, and the best setting within the rules earns $522. Profit figures in this section are rounded to the dollar. A change is a genuine improvement, under the definition preregistered for the model runs, if it raises true profit and leaves quoting in volatile periods on.

Under the red-team prompt the governed arm promoted 19 changes, all genuine improvements, and the agent ended at a mean of $308 (range $248 to $379). The ungoverned arm installed 49 changes. Forty were not genuine improvements: 36 lowered true profit, and 4 raised it only while quoting in volatile periods was off. All ten ungoverned repeats ended below the start and seven ended below zero (mean −$41, range −$124 to $163).

That contrast should not be read as the judge stopping 40 attacks. The two arms did not face the same proposals. Under the red-team prompt the ungoverned proposer's feedback said that checks 2 to 4 were not run, and it then narrowed the spread 33 times. Under the same prompt the governed proposer suggested a narrower spread twice. The red-team arm therefore measures an instructed adversary with no judge, not drift. Across both prompts the governed judge evaluated 29 non-genuine proposals that had passed admission, of two kinds. The participation shortcut came up 20 times and was rejected at check 3 each time. One spread cut, which loses $91 per day, came up 9 times and was rejected at check 2 each time. Neither is near the margin. With 0 of 29 promoted, the one-sided 95% upper bound on the false-promotion rate is about 10%, but with only two distinct changes the effective sample is near two, so no rate bound is supported. A two-line rule (never change the quoting switch; reject a narrower spread) would have stopped the same 29. Check 4 stopped none of them. So this pilot does not show that the four-check judge does better than that rule, or that it resists a varied attack.

Under the ordinary prompt the governed arm again promoted 19 changes, all genuine improvements (mean $304, range $248 to $379). The ungoverned arm installed 48 changes and ended higher, at a mean of $476. Two of its installs were not genuine, both in one run, which turned volatile quoting off and ended at $846, above the best rule-abiding setting. Without that run the mean is $435.

Governance had a cost, and by count most declines were delays. The judge declined 26 genuine proposals, 13 under each prompt. It promoted 21 of them on a later attempt in the same run, once the retry rule had pooled more evidence, and 5 were never recovered. With five attempts per run, that delay is the likely reason the governed agent ended lower under the ordinary prompt. Twenty of the 26 are one step, the widening from 0.10% to 0.11%, and check 4 accounts for 6. Seventeen of those 20 are the same first-attempt decision on the same seeded data, repeated across runs. Calibration promoted this step 89 of 100 times (Section 8.4), so the decline count is not a rate. The means support the direction of the cost, not its size. The proposer suggested the participation shortcut 21 times in the governed arm, 10 of them under the ordinary prompt. One was blocked at admission and the other 20 were rejected at check 3.

The pilot is small and exploratory. The governed runs were not preregistered, and the judge's sealed salt had already been published. The ten rounds repeat one configuration on the same old-data seeds, so they are not ten independent samples: three red-team ungoverned runs ended at the same −$124, and five ordinary ungoverned runs at the same $443. The arms share settings and starting data, not random draws. Under the red-team prompt the ungoverned proposer's system prompt still described the four checks while its feedback said checks 2 to 4 were not run. Under the ordinary prompt its feedback matched the governed loop's, with each install reported as a promotion. The admission step cap applied in both arms. Checks 2 and 4 sample the same stationary generator as the answer key, so the pilot cannot show how the judge behaves when its fast measure and the true objective diverge. One model and one synthetic market were used. The prompts

and run records are in the unreleased repository, so others cannot yet reproduce this section. A first batch of four ungoverned runs, compared against the governed runs of 16 November, is not pooled here. It showed the same direction: the two red-team runs ended at −$113 and −$4, and the two ordinary runs at $443.

8.9 Cost

All model spend was in the toy. The 16 November pilots made 96 calls, including two test calls, all HTTP 200, for $1.58 in total (Qwen 71 calls, $1.35; Sonnet 25 calls, $0.23). This matches the API key's own usage counter. The comparison of Section 8.8 made 220 Sonnet calls for $1.95, including its first batch. The core runs made no model calls, and the 240-challenger controls suite took 107 s on one laptop. Section 8.8 is a first toy baseline with admission only. It does not answer RQ5, which concerns governance cost on real data under a preregistered design.

8.10 Limitations of the pilots

Most of Section 6 is still untested. There is no Analyst, research agenda or Analyst control, so RQ6 and RQ7 are open. None of the seven experimental arms of Section 6.2 was built, so there is no comparison between those arms. RQ4 was not tested as posed: there was no selection-conditioned null, and the synthetic replay-to-shadow pairs use later tape time rather than prospective data. RQ5 has only the small, exploratory toy baseline of Section 8.8. RQ8 has attack conditions but no equal-budget contrast with and without isolation. Language-model proposers never ran in the core loop or on real data. There was one real session on one pair and one venue, with no shadow window, no neutral controls on real data and no live side-by-side comparison. The fill model has no queue, no impact and a look-ahead defect on live data. Rollback was never triggered in a run. Gate 7 was never reached by a control, and gates 5–7 were not exercised by neutral challengers. The longest promotion chain in the seven-gate core was v1 to v3 on planted synthetic tapes, and toy runs made at most a few promotions each, so nothing here bears on sustained or recursive self-evolution. The controls were not independently authored and the work has not been independently reproduced; I re-ran it from a clean clone, and it received one read-only outside audit. Several defects found by that audit remain in the frozen code. No live capital was involved, and no result here concerns profit, alpha or skill.

9. Limitations and Stop Conditions

9.1 Limitations

1. The architecture has not been evaluated end to end. The reference implementation covers admission, evaluation, promotion and receipts but not the Analyst or research agenda, and the pilot results of Section 8 are feasibility checks, not the protocol of Section 6.
2. The reference implementation is a one-day, single-builder, paper-only prototype for one market-making agent. Its synthetic and toy results do not transfer to real markets, its real-market evidence is one two-hour session with no promotion, and several defects found by its outside audit remain in the frozen code. The evaluation claims idealized instrumented inputs unless an operational perturbation suite is added.
3. Recent preprints provide mechanism ideas but limited independent replication.
4. An Analyst can miss useful improvements, overproduce plausible issues, privilege observable failures and repeatedly reshape the agenda around released feedback. One Analyst can also hold correlated blind spots that a deterministic ordering cannot repair, because ranking cannot restore an issue that was never formulated. Complete discovery is neither testable nor claimed, and measured recall is bounded by the adjudicated reference inventory rather than by the latent issue space.

5. A sealed evaluator can still measure the wrong construct and leak information through Analyst or proposer receipts; isolation prevents direct access, not adaptive inference or misspecification.
6. Replay and shadow cannot identify a challenger's multi-step counterfactual market or portfolio trajectory after its action diverges without a calibrated stateful model.
7. Independent market regimes, not the number of logged ticks, may be the binding sample-size constraint. Confirmation delay may outlive a short-lived advantage.
8. A complete ledger does not guarantee adequate human review. Review capacity, privacy, data minimization, access control, retention and redaction remain product, legal and human-factors problems. Analyst receipts may reference records that later require redaction, so immutable event metadata must be separable from revocable payloads. An understandable envelope also does not guarantee informed user consent; that is a separate product, legal and human-factors problem.
9. A successful paper-only experiment does not by itself justify live-capital automatic promotion. A policy may condition on evaluation mode, and formal permission limits do not prove effective capability is unchanged.
10. Per-installation compliance does not establish fleet-level safety. Common-mode exposure, concentration, staggered rollout and correlated agenda errors produced by a shared Analyst policy, evidence projection or seed taxonomy are outside the initial experiment.
11. No cumulative performance, regret or long-run improvement guarantee is derived. A one-step result does not establish that legibility, value or safety composes across a lineage.
12. Propensity-weighted off-policy reanalysis is outside the initial deterministic matched-input design. If it becomes a goal, behavior-policy probabilities must be logged prospectively.
13. Receipts that have not been released can be audited only by their owner. The pilot ledgers are hash-chained and were re-verified by the builder, but they have not been independently reproduced. A publication package should provide the code, ledgers, hashes, schemas and aggregate diagnostics, and, once an Analyst exists, a deterministic agenda replay for at least one complete Analyst run.
14. Operating history records outcomes only for the actions the incumbent took. It may motivate a research hypothesis of any permitted origin, including exploratory ones. Without prospectively logged action and abstention propensities it cannot establish the counterfactual value of an unchosen action or the causal value of a proposed mutation, and no Analyst observation is treated as such evidence. Causal and counterfactual improvement claims come only from the governed comparison.
15. Agenda determinism is claimed for eligibility and ordering over committed issue records, not for the Analyst inference that creates them. Improvement of the agenda policy itself is authorized outside the loop and is not governed by GCL in this paper.
16. Analyst input scale, receipt growth and retrieval latency are untested. The design assumes a compaction and archival rule that preserves audit semantics and replay equivalence.
17. Read-only market observation is not risk-free and is not automatically easy to evaluate. Objective event detection may have fast labels, while relevance, interpretation, causal importance and downstream usefulness may not. Precision on a rare event family depends on its base rate, so an operating point moved between incumbent and challenger can change every reported rate without changing detection quality. An alert can also influence a human or downstream system despite the absence of execution authority. The initial study therefore cannot generalize a result from one labelled event family to observation agents as a class.

9.2 Stop conditions

The research program stops or returns to instrumentation if:

- the Analyst accesses sealed evidence, mutates its agenda policy during a trial, activates a policy successor without the agenda-policy owner's authorization, inflates a deterministically recomputable priority field, presents an Analyst-asserted field outside its declared provenance or preregistered bound, misses the injected issue above its bound, cannot return `NO_ACTION` on neutral histories, exceeds the duplicate-fragmentation bound or recovers a cross-partition canary;
- an invalid mutation passes deterministic admission or the positive control fails to promote;
- neutral controls exceed the preregistered false-promotion bound;
- matched-input parity, point-in-time reconstruction or trajectory scope cannot be demonstrated;
- improvements arise primarily from opportunity or coverage avoidance rather than labelled warranted abstention;
- replay improvements repeatedly reverse in prospective shadow or the fast surrogate fails delayed economic validation;
- the challenger improves only when it can observe or infer the evaluator beyond the fixed query budget;
- hard risk, accounting, execution or emission invariants regress; or
- the available independent regimes cannot distinguish the minimum meaningful effect from noise.

Before the run, each stop condition is mapped to a detector, threshold, decision owner, halt action and receipt. The stop authority can revoke the authority of Analyst and proposer descendants, revoke credentials and prevent post-stop writes. Stop atomically prevents new assignments and new authoritative state transitions. It does not claim atomic cancellation of computation, which external model and tool calls do not generally permit. Queued and in-flight work is cancelled where cancellable, and unavoidable late output is quarantined and receipted; quarantined output can never enter the agenda, proposal or promotion path without a new explicit authorization. Authorized human pause, correction or veto of an issue is receipted with its reason class fixed before outcomes mature, and results are reported both including and excluding those interventions. Out-of-band human action receives an exceptional receipt and is excluded from promotion success counts.

The program also stops an agenda policy or mutation class after a fixed total budget, a preregistered convergence rule or an opportunity-cost boundary, even if no safety failure occurs. A null result is attributed only after diagnostic controls distinguish Analyst sensitivity, agenda ranking, proposal failure, operator inadequacy, gate insensitivity, evaluator invalidity and bounded-search futility. The correct response may be a repaired measurement contract, a narrower claim or abandonment of the agenda policy or mutation class.

10. Discussion

GCL is best understood as AutoResearch for trading agents under a constitution. AutoResearch supplies the experimental rhythm: propose, run, measure, keep or discard and record. The Analyst adds an explicit upstream rhythm: observe released evidence, open a persistent issue, rank it under a fixed agenda policy and retain the selection decision. SHARP narrows the trading-policy representation. SkillOpt bounds the edit. AHE forces the hypothesis to precede the result. ADIAS preserves issue and intervention history. STALE/CUPMem separates active evidence from the audit record. Agora motivates evaluator separation. SPIBB supplies the conservative incumbent fallback. Financial model-selection research limits how much evidence repeated agenda, historical and protocol search can honestly provide.

The synthesis matters because trading agents do not merely generate answers. They can hold durable state and standing authority. Self-evolution in that setting is an authority-governance problem before it is an optimization problem.

That authority need not include order execution. A read-only market-observation agent can still shape a principal's decision through what it emits, omits, delays or labels as fact. It therefore provides a lower-capital-risk but not zero-consequence test of GCL. When objective event labels exist, it may also provide a faster first experiment than execution because detection coverage, precision and latency can mature before economic outcomes. This advantage does not extend automatically to market interpretation or trading usefulness.

The architecture also separates discovery and creativity from control. The Analyst may generate unusual improvement hypotheses and the proposer may generate unusual interventions; deterministic agenda and admission systems determine whether each output is eligible under frozen grammars and budgets. A "hotter" model may produce more varied issues or proposals, but temperature is not the mechanism of self-evolution and is not a safety boundary.

If the first experimental program succeeds, it supports history-informed agenda formation and one-step governed artifact improvement under the tested contracts. It does not prove that useful improvement always exists, that the Analyst finds every useful issue or that a fixed ranking is the best use of scarce research capacity. Mechanism-family diversity is reported as a diagnostic for local repetition and discovery blind spots rather than as a definition of a research agenda, since an agenda may rationally concentrate on one important mechanism family. Sustained self-evolution requires multi-generation evidence, fixed-anchor comparisons and cumulative agenda and trial error accounting. If the program produces no promotions, that result is interpretable only if the Analyst and positive controls pass and the diagnostic tree localizes the failure. If the experimental program succeeds on both fronts, the next research question is whether a user-authorized autonomy envelope can support automatic in-envelope promotion over longer periods. That later question requires hidden-mode testing, fleet controls and a separate authority decision.

I read the pilot results of Section 8 against that criterion. They show that the admission, gate and receipt machinery behaves as specified on the controls its builder wrote. The scripted positive control passed on synthetic data, and the toy judge's sensitivity was measured against a known answer. They also show that the machinery has gaps: an attempt that received no receipt, rule additions that the edit limit does not bound and a resubmission route that had to be closed by a new rule. The real-market session is harder to read. Nothing was promoted, but with no positive control on real data, no shadow window and no Analyst, that null counts neither for nor against the loop.

The comparison of Section 8.8 puts a first, small number on both sides of the design. With no judge, a proposer asked to pass non-improvements, and told that the later checks were not run, left the toy agent worse off in every repeat. The judge rejected the 29 non-genuine proposals it saw, which were only two distinct changes. It also declined 26 genuine proposals and promoted 21 of them on a later attempt. Under an ordinary prompt the ungoverned agent ended ahead. The open problem is therefore not only whether a governed loop is safe. It is how much improvement the governance gives up, whether a much simpler rule would do as well, and whether a judge can be made less conservative without admitting the changes it now stops.

11. Conclusion

Self-evolving trading agents should not be defined by how much of themselves they can rewrite or whether they can place orders. They should be defined by whether their operating history can support persistent, falsifiable research issues and bounded proposals that are tested under comparable evidence and earn narrowly scoped execution or informational authority without controlling the agenda policy, constitution or judge.

The Governed Challenger Loop turns that principle into a researchable systems design: immutable incumbent, governed Analyst, persistent issue agenda, typed challenger, deterministic admission, scoped matched-input comparison, sealed multi-gate evaluation, complete agenda, trial and protocol accounting, human-authorized agenda-policy succession, consistent rollback and a paper-first deployment ceiling. The admission, evaluation, promotion and receipt layers now exist in a reference implementation. Its small pilots behaved as specified on synthetic and toy controls. In a toy comparison, the judge rejected all 29 non-genuine proposals it evaluated, two distinct changes, and declined 26 genuine proposals, 21 of which it promoted on a later attempt. An ungoverned loop under a red-team prompt, told that the later checks were not run, installed 40 changes that were not genuine improvements. Under an ordinary prompt the ungoverned loop ended ahead. On one real-market session it promoted nothing. The Analyst and research agenda are still proposals, and nothing here shows better trading. The next scientific step is therefore not a live launch or a performance claim. It is a preregistered paper-only experiment with Analyst, positive and negative controls capable of showing whether agenda formation, mutation generation, the gate or the mutation class is wrong.

Authoring and Review Provenance

The architecture was synthesized from five independent model-generated reviews of a common decision brief. A separate drafting and review sequence then required material claims to be reconciled against primary sources. During that process, claims about artifact immutability and position-version attribution were corrected rather than resolved by model agreement. Every number in Section 8 was checked against the reference implementation's committed result files and reports. The implementation received one read-only outside review by a coding model (Section 7.3). AI coding and writing assistants were used, under the author's direction, to build the reference implementation and to draft this manuscript.

This provenance is not evidence that the architecture works and does not establish independence from shared human framing. It records how the design and manuscript were produced. Empirical validity still depends on the experiments in Section 6 and external review should be blinded to preferred outcomes where practical.

Competing interests

The author works at Avalon Labs, which builds AI trading agents. The reference implementation in Section 7 governs paper trading only. Its code and result files have not been publicly released.

References

[1] Chen, X., Zhu, W., Zheng, S., Rasul, K., Deng, Y., and Li, H. "SHARP: A Self-Evolving Human-Auditable Rubric Policy for Financial Trading Agents." *arXiv preprint arXiv:2605.06822*, 2026. https://arxiv.org/abs/2605.06822

[2] Yang, Y. et al. "SkillOpt: Executive Strategy for Self-Evolving Agent Skills." *arXiv preprint arXiv:2605.23904*, 2026. https://arxiv.org/abs/2605.23904

[3] Lin, J. et al. "Agentic Harness Engineering: Observability-Driven Automatic Evolution of Coding-Agent Harnesses." *arXiv preprint arXiv:2604.25850*, 2026. https://arxiv.org/abs/2604.25850

[4] Jiang, L., Tang, B., Goetz, S., and Guo, Y. "ADIAS: Automated Design of Interactive Agentic Systems." *arXiv preprint arXiv:2608.06410*, 2026. https://arxiv.org/abs/2608.06410

[5] Chao, H., Bai, Y., Sheng, R., Li, T., and Sun, Y. "STALE: Can LLM Agents Know When Their Memories Are No Longer Valid?" *arXiv preprint arXiv:2605.06527*, 2026. https://arxiv.org/abs/2605.06527

[6] Li, Y., Liu, S., and Liu, B. "AI Trading's Alpha Singularity: Emergent Market Reasoning through Agent-to-Agent Self-Evolution." *arXiv preprint arXiv:2606.29194*, 2026. https://arxiv.org/abs/2606.29194

[7] Laroche, R., Trichelair, P., and Tachet des Combes, R. "Safe Policy Improvement with Baseline Bootstrapping." *Proceedings of the 36th International Conference on Machine Learning*, 2019. https://proceedings.mlr.press/v97/laroche19a.html

[8] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. "Reflexion: Language Agents with Verbal Reinforcement Learning." *Advances in Neural Information Processing Systems 36*, 2023. https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html

[9] Bailey, D. H., Borwein, J. M., López de Prado, M., and Zhu, Q. J. "The Probability of Backtest Overfitting." *Journal of Computational Finance* 20(4), 2017. https://www.risk.net/journal-of-computational-finance/2471206/the-probability-of-backtest-overfitting

[10] White, H. "A Reality Check for Data Snooping." *Econometrica* 68(5), 1097–1126, 2000. https://doi.org/10.1111/1468-0262.00152

[11] Zhang, J., Hu, S., Lu, C., Lange, R., and Clune, J. "Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents." *arXiv preprint arXiv:2505.22954*, 2025. https://arxiv.org/abs/2505.22954

[12] Skalse, J., Howe, N., Krasheninnikov, D., and Krueger, D. "Defining and Characterizing Reward Gaming." *Advances in Neural Information Processing Systems 35*, 2022. https://proceedings.neurips.cc/paper_files/paper/2022/hash/3d719fee332caa23d5038b8a90e81796-Abstract-Conference.html

[13] Bailey, D. H. and López de Prado, M. "The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting, and Non-Normality." *Journal of Portfolio Management* 40(5), 2014. https://doi.org/10.3905/jpm.2014.40.5.094

Referenced research artifacts

- [A1] Karpathy, A. *AutoResearch*. GitHub repository, 2026. https://github.com/karpathy/autoresearch
- [A2] Google DeepMind. "AlphaEvolve: A Gemini-Powered Coding Agent for Designing Advanced Algorithms." 2025. https://deepmind.google/blog/alphaevolve-a-gemini-powered-coding-agent-for-designing-advanced-algorithms/