

Evidence-Gated Recursion: An Experiment on a Self-Governing Research Organization of Frontier Language-Model Agents

Lindell S. Cumes

QWNTL Labs

Project page: <https://qwntl.com/frontier>

Abstract

Papers written by the autonomous-research system FARS averaged 3.17 of 10 in human review, and Into the Unknown found integrity failures in 114 of 135 agent-written research submissions. We report an experiment on a self-governing research organization of frontier language-model agents that worked only on problems with a named state of the art and an exact or official scorer, which no agent’s judgment can move. Evidence-gated recursion (EGR), the organization’s governance protocol, gives each type of candidate one gate, a predicate over one kind of evidence, and the kinds range from exact certificates and official scores to preregistrations, cross-family audits and number traces. A candidate is admitted only when its gate accepts it, and the organization applied the same rule to its own instruments at four levels. The experiment produced a new upper bound $C_1 \leq 1.50266484\dots$ for the first autocorrelation inequality, below the Einstein Arena leader’s 1.5027434867, and circle-packing records at 306 of Packomania’s sizes. Both stood on October 30, 2026. With Claude Opus 5.5 as the worker model, its take-home harness reached a median of 914 cycles against 948 for a CORAL-faithful team. We also found that weak-to-strong memory transfer is negative and that verdict-based supervision lowers a frontier model’s production correctness. Cross-family audits failed 11 of 60 of the organization’s own claims. No number in 693 traced claims is untraceable. The headroom relation, found in a post-hoc audit of the 195-comparison record, states that a component’s lift is a share of the model’s residual error. Fitted on weaker models, the relation forecasts Claude Opus 5.5 out of sample.

Release. Project page: <https://qwntl.com/frontier>. Each companion paper lists the witnesses, verifiers, records, and code it releases.

1 Introduction

The autonomous-research system FARS ran for 417 hours and produced 166 papers, and the 282 reviews that volunteer reviewers wrote of 140 of them averaged 3.17 of 10 [43]. The AI Scientist-v2’s authors sent three of its manuscripts to a workshop. One of them averaged 6.33 [46]. Independent scientists found 79.4% of the statements in Kosmos reports accurate [29]. Into the Unknown asked nine agents for 15 follow-up papers each. None of the 135 met all of its conditions. Its integrity check failed 114 of them. Manual inspection found lenient self-verification in 58 of 64 submissions [23].

We report an experiment in which every result is scored by a verifier that makes no model judgment. Its subject is a self-governing research organization of frontier language-model agents, which we built together with its governance protocol. We ran it from October 6 to 9, 2026. The first research question (RQ1) is whether such an organization can produce externally verifiable results against named states of the art. RQ2 asks which governance mechanisms set the

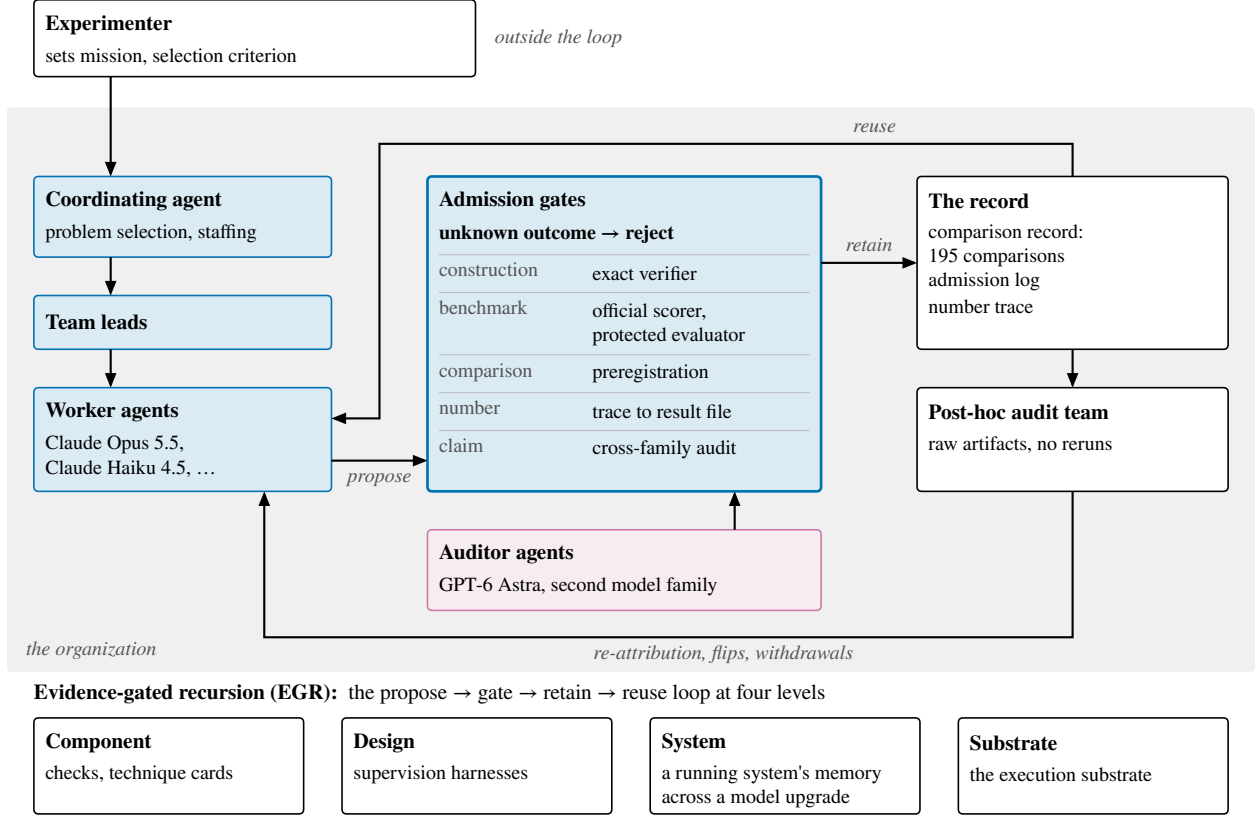


Figure 1: **Every candidate the organization admits passes a typed evidence gate.** The experimenter sets the mission and the selection criterion and stays outside the loop, and the coordinating agent staffs team leads, whose worker agents propose candidates. Each candidate type has its own gate, and an unknown outcome is a rejection. Admitted candidates enter the record, which holds the 195-comparison record, an admission log, and a number trace, and they are reused. Auditor agents of a second model family judge claims on the exact candidate, and a post-hoc audit re-attributes each result in the record and returns its findings to the proposers as verdict flips, withdrawals, and new comparisons. The bottom row shows the same admission rule applied to the organization’s own instruments at four levels.

integrity of what it reports and what each one catches, and RQ3 asks what the organization’s own comparison record shows when it is audited.

To answer these questions, an outsider must be able to check each result against a named state of the art without trusting the organization. At the start of the run we wrote that requirement into the mission, which was to beat a named public state of the art on public benchmarks with official scoring. The coordinating agent selected problems to that bar from four families (Table 1). Open extremal problems come with exact certificates, and τ^2 -bench and AppWorld with official outcomes. Anthropic’s performance take-home and KernelBench run under a protected evaluator, and production software tickets, for a question no public benchmark scores, are scored by a hidden oracle on streams hashed before any arm ran. Under the protocol, the organization’s agents chose the instances and ran the arms; they also built and revised the instruments and decided what entered the record (Figure 1).

Under evidence-gated recursion (EGR), every candidate has a type and each type has one gate that accepts one kind of evidence. Constructions and benchmark results are checked by computation, and comparisons, claims and manuscript numbers by evidence the proposer cannot produce by assertion, which for a claim is an audit by a model of a second family that reads the exact candidate. A candidate enters the record only when its gate returns true, and an unknown outcome returns false (Section 3). The organization applied the same admission rule to its own instruments at four levels, from components such as executable checks and technique cards, through harness designs and a running system’s memory across a model upgrade, up to the execution substrate the agents ran on. Among prior systems, Infinity Machine gates changes to one agent’s harness, and Triage-PiEvo gates changes to a run’s scientific principles [16, 25].

The experiment produced new results on each family’s own verifier (Section 5). For the first autocorrelation inequality it set a new upper bound $C_1 \leq 1.50266484 \dots$, 7,865 times the Einstein Arena’s minimum improvement below the Arena’s leader, and it found the best sums of radii for unequal circles in a square at 306 of Packomania’s sizes. Both stood as of October 30, 2026. The organization’s archive harness reached a median of 914 cycles on Anthropic’s take-home, below every model result published in cycles as of that date. On KernelBench, a store of verified kernels with six admitted technique cards lifted Claude Opus 5.5 to $3.67\times$ mean capped speedup, against $1.95\times$ for our port of ACE’s playbook ($3.29\times$ with the same cards). The bare model reached $1.86\times$. The experiment also measured two effects. Weak-to-strong memory transfer is negative, with a store built from Claude Opus 4.8’s verified kernel work lowering Claude Opus 5.5 by 0.103 log speedup. Supervision built on verdicts lowers Claude Opus 5.5’s production correctness below its unsupervised level on every stream. Evidence-only supervision holds it at that level, 1.000, on all 2,880 tickets.

For RQ2, the gates turned candidates back both during the run and in the papers’ number traces (Section 6). The cross-family audit failed 11 of 60 of the organization’s own claims. In a paired development test, a GPT-6 Astra audit repaired 14 of 14 defect states. Claude Opus 5.5 auditing itself repaired 5. The number trace found no untraceable number in 693 claims and flagged 73 of them for correction.

For RQ3, 19 of the record’s 78 Claude Opus 5.5 component comparisons are significant wins, with probability 5.9×10^{-14} under no effect if the comparisons are independent (Section 7). In the agents’ own retests, first-claimed positives held in 6 of 16 and negatives in 6 of 8, a selection effect on the positives. The post-hoc audit also yields the headroom relation, which forecasts Claude Opus 5.5’s comparisons from weaker models with a mean absolute error of 3.3 points against 5.2 for a constant gain.

Among the systems in Table 17, this organization is the only role-specialized organization that gates changes to its own instruments. Infinity Machine, the one other system in that table that gates changes to itself, admitted none of its self-derived changes [16]. The five companion papers report each result in full [7, 8, 9, 10, 11].

2 The experiment

2.1 Object of study

Definition 1 (Research organization). A research organization is a tuple $O = (X, C, \{L_j\}, \{W_i\}, A, \mathcal{G}, \mathcal{R}, S)$: an experimenter X who fixes the mission and the standard of evidence; a coordinating agent C ; team leads L_j ; worker agents $W_i = \langle M_i, H_i \rangle$, each a model with its version and the harness it runs in; auditor agents A of a second model family; a set of gates $\mathcal{G}$; a shared record $\mathcal{R}$; and the execution substrate S every agent runs on.

The organization is *self-governing* in that its agents ran the work, from choosing problem instances and staffing teams to designing and running arms, and also built and revised its instruments and decided admissions to $\mathcal{R}$, all under the gates of the protocol. We write a task as $\tau = (q, s_0, V)$: a research question, a start state (code, data, prior constructions), and a verifier V that the proposing agent cannot edit. We measure the budget B in gateway requests and wall-clock hours.

2.2 Problem selection

RQ1 asks whether the organization’s results hold against a named state of the art when checked by someone who does not trust it. RQ2 and RQ3 compare what the gates admitted with what was true. For all three, a problem needs outcomes that no agent’s judgment can move, which rules out problems that a reviewer scores or that only the organization can score.

The experimenter’s mission set the bar at a named public state of the art on public benchmarks with official scoring, which excludes the organization’s own graders and weak baselines as comparators. We read that bar as a selection criterion with three requirements. A problem needs an exact verification procedure or an official scorer and a named state of the art to beat. Its outcome must also be checkable without trusting the agents that produced it, and the verifiers meet this last requirement because each is fixed before the arm runs and makes no model judgment, and no agent it scores can edit it.

Two families are scored by evaluators the organization built and qualified before any arm ran. KernelBench’s protected evaluator runs each kernel as data. It rejected all 21 attacks in its qualification suite. Supervision asks a question no

Table 1: Each problem family against the three requirements of the selection criterion.

Family	Problems	Exact or official scorer	Named state of the art	Checkable without trusting the agents
Discovery	first autocorrelation constant C_1 ; Erdős’s minimum overlap M ; signed autocorrelation C_3^+ ; sums of radii of unequal circles in a square	exact rational arithmetic on the witness	Einstein Arena leaders [12]; MILO [21]; AlphaEvolve [32]; Packomania’s table [41]	anyone can re-score a published witness with an independent checker
Systems optimization	Anthropic’s performance take-home [2]; KernelBench [34]	the take-home’s official cycle count and tests; KernelBench speedup under a protected evaluator	published take-home results (CORAL [37], AgentDiscover [13]); ACE’s playbook [48]	the evaluator runs each candidate as data and passed attack controls before search
Verification	τ^2 -bench [5]; AppWorld [44]	the benchmark’s official task outcome for every attempt	LLM-as-judge verification; an ACE-style verification playbook	catch and false fire are counted against the benchmark’s own labels
Supervision	120 dependent tickets building an account-balance service	a hidden oracle scorer on streams hashed before any arm ran	verdict-based supervision from published practice (model judges, compiled tests, rollback), built in-house	the oracle and streams are fixed before any arm runs, and harm is measured against the unsupervised model

public benchmark scores, and its hidden oracle, on streams hashed before any arm ran, kills 88 of 89 rule-violation mutants. The standard of evidence, set in the same design and extended during the run, counts as a result only a comparison with the same model without the component and with the strongest competing systems on that model, and it allows no headline where the bare model is near its ceiling.

The coordinating agent selected problems that meet these requirements, and teams chose instances within each. The selection spans four kinds of research (Table 1). Discovery asks for constructions no one has found, and an exact certificate settles each one. Systems optimization seeks faster programs on public tasks that have exact scorers and published harness rivals. In verification, official outcomes label every attempt, and the question is whether a check can distinguish a failed attempt from a correct one. Supervision tests whether oversight helps or harms a model doing long-horizon work, with a hidden oracle scoring every ticket against the unsupervised model.

2.3 The verifiers

Each family’s verifier is a fixed computation, and for the discovery family it is a certificate. C_1 is the largest constant with $\max_t (f * f)(t) \geq C_1 (\int f)^2$ for every nonnegative f supported on $[-\frac{1}{4}, \frac{1}{4}]$, so every such f certifies $C_1 \leq \|f * f\|_\infty / \|f\|_1^2$; C_3^+ is the same constant for real-valued f with $\int f \neq 0$. For a step function with heights a_i on n cells, $f * f$ is linear between knots, and the bound reduces to a finite maximum over the integer sums $c_k = \sum_{i+j=k} a_i a_j$, a computation in exact rational arithmetic [8]. Erdős’s constant M is bounded the same way by a step function $h : [0, 2] \rightarrow [0, 1]$ with $\int h = 1$,

$$M \leq \sup_t \int_{\mathbb{R}} h(x) (\mathbf{1}_{[0,2]} - h)(x+t) dx, \quad (1)$$

and a packing of N circles with centers (x_i, y_i) and radii r_i in the unit square is scored by $\sum_i r_i$ subject to $r_i \leq x_i, y_i \leq 1 - r_i$ and $(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2$. A certificate is the witness together with the finite computation that evaluates it.

In systems optimization, the take-home’s score is the cycle count of the instruction list a candidate emits on the simulated machine, and the candidate must pass the official tests. KernelBench’s score is the *capped speedup* over PyTorch

Table 2: **The 21 teams by function, and the rest of the organization.** We map worktrees and branches to teams by name and count each commit once, before the pause-complete commit at 14:28 UTC on October 9. Auditor agents of the second family worked in standalone clones, so the audit team’s count understates its agents.

Function	Teams	Worktrees	Commits
Long-horizon software supervision	1	21	448
Continuous operation	1	19	403
Shared search on open problems	1	22	273
Shared search on the take-home	1	7	97
Capability carried across models	2	30	829
Recovery from crashes	2	29	628
Self-improvement of harnesses	2	31	753
Trained components	3	48	1,209
Literature feeding the experiments	2	33	321
Cross-family audit	1	1	2
Post-hoc audit of the record	1	8	54
Retests of flagged conclusions	4	22	177
21 teams	21	271	5,194
Three other teams	3	18	528
Main and subagent branches	–	0	126
Organization		289	5,848

eager, at most $10\times$ and zero if the kernel is incorrect; log speedup is $\log(1 + \text{capped speedup})$ [7]. In verification, the benchmark’s official outcome $y(a) \in \{0, 1\}$ for an attempt a is the verifier. A done-check returns a flag $f(a)$ without reading $y(a)$, and

$$\text{catch} = \Pr[f = 1 \mid y = 0], \quad \text{false fire} = \Pr[f = 1 \mid y = 1], \quad J = \text{catch} - \text{false fire}, \quad (2)$$

and we report differences in Youden’s J in points [11]. In supervision, the working tree is deployed after each ticket and serves scripted customer traffic, and production correctness S_{prod} scores the resulting production store against a hidden oracle; on each stream the oracle used as an implementation scores 1 at every ticket, and the hidden scorer kills 88 of 89 targeted rule-violation mutants.

2.4 The organization

The organization’s briefs defined its agents’ roles. The coordinating agent held the organization’s 13 goals, its compute, and its model gateway, and it staffed the teams. Each of the roster’s 21 team leads owned one function (Table 2). The functions run from long-horizon software supervision and shared search on open problems to cross-family audit and the retesting of flagged conclusions. Three further teams with their own leads bring the organization to 24.

Leads staffed two kinds of worker agents. Builders ran the planned arms, and exploratory workers had a brief to read recent literature and test ideas outside the team’s plan, a role specialization that shaped where results came from (Section 4.2). Reviewers of the same family and auditors of a second family complete the roles, together with the post-hoc auditors. A worker hands its lead each result with the command and commit that produced it and its artifacts, so that every result in the record can be rebuilt from its commit. Leads and workers worked in separate git worktrees, 289 directories in the organization, and the cross-family auditors in standalone clones. The repository records 5,849 commits through the pause-complete commit.

2.5 Size of the experiment

The organization’s first commit is dated 22:25 UTC on October 6, 2026. Its pause completed at 14:28 UTC on October 9. From 02:04 UTC on October 7 to 13:15 UTC on October 9, a 59.2-hour window, hosted-model requests passed one gateway, which logged the model, the team, and the tokens of each. Earlier runs called provider APIs directly, so the counts we report from the gateway are a lower bound. Open-weight models ran on the organization’s own GPUs and do not pass the gateway. The gateway logged 5,870,586 requests and 46.53 billion tokens, 6.82 billion of them

Table 3: **Information-flow contract.** Each role’s inputs and outputs, and when the object it judges is fixed. The proposer never sees an answer key or a hidden scorer, and the auditor of a claim belongs to another model family; the post-hoc audit reruns nothing.

Role	Sees	Writes	Object fixed when
Experimenter	the agents’ record, team handoffs	mission, selection criterion, standard of evidence, scope, capacity, pauses	at the start of the run
Coordinating agent	the agents’ record, team handoffs	problem selection, outcomes, team briefs	when it issues a brief
Worker agent (proposer)	task, record, own worktree; never the answer key or hidden scorer	candidates, code, preregistrations	at commit, before the run it judges
Verifier or evaluator	the candidate as data	a score and its fingerprint	evaluator hash pinned before search
Auditor agent (GPT-6 Astra)	the exact candidate and its claim	READY, CHANGES_REQUIRED, or FAIL	audited commit
Number tracer	manuscript and result files	one trace row per statement	release build
Post-hoc audit team	raw artifacts of the whole record	re-attribution, flips, withdrawals	after the run, no reruns

Table 4: **Gateway traffic over the 59.2-hour window** (02:04:09 UTC October 7 to 13:15:17 UTC October 9, 2026), by model family. Tokens are the gateway’s input and output counts per request. Claude input counts exclude cache reads, and GPT input counts include cached tokens. We round the token rows and omit the other identifiers from them, so they do not sum to the totals.

Model	Requests	Succeeded	Input tokens	Output tokens
Claude Opus 5.5	3,341,071	3,332,127	22.11B	4.80B
Claude Haiku 4.5	1,589,646	1,565,540	3.08B	1.66B
Claude Opus 4.8	569,967	568,852	1.67B	0.22B
GPT-6 Astra	264,180	247,657	8.77B	0.09B
GPT-6 Luna	91,219	79,435	4.07B	0.03B
Other or no identifier	14,503	13,491	–	–
All requests	5,870,586	5,807,102	39.71B	6.82B

output (Tables 4 and 5). Claude Opus 5.5 carried 3,341,071 of the requests. Up to 1,191 of them were in flight at once. The hourly token rate, 786 million, is 15.2 times FARS’s 51.8 million per hour (21.6 billion tokens over its 417-hour deployment) [43], and output tokens alone ran at 2.2 times FARS’s rate for all tokens.

3 Governance protocol: evidence-gated recursion

3.1 Typed evidence and the admission rule

Definition 2 (Admission). Let $\kappa(c) \in \{\text{construction, benchmark result, comparison, claim, number}\}$ be the type of a candidate c , and e the evidence submitted with it. Each type has one gate $G_\kappa \in \mathcal{G}$, a predicate over one evidence type, with outcomes in $\{1, 0, \perp\}$ where $\perp$ is an unknown outcome. Then

$$c \in \mathcal{R} \iff G_{\kappa(c)}(c, e) = 1, \quad (3)$$

so a candidate whose gate returns 0 or $\perp$ stays out.

Each of the five gates accepts one kind of evidence (Table 7). A construction enters on an *exact certificate*, its witness frozen with its hash and re-scored in exact rational arithmetic by a checker that shares no code with the optimizer that

Table 5: **Models and roles.** Requests are the gateway’s count for each model over the 59.2-hour window, and we combine both Claude Haiku 4.5 identifiers. Open-weight models were served on the organization’s own GPUs.

Model	Identifier	Roles and settings	Requests
Claude Opus 5.5	claude-opus-5-5	worker and lead agents; judge arms; adaptive thinking; effort xhigh on construction and open-problem teams, medium on the take-home (Claude Code 2.1.289), high on supervision and as judge	3,341,071
Claude Haiku 4.5	claude-haiku-4-5	weaker actor and successor; failure distiller on the shared archive	1,589,646
Claude Opus 4.8	claude-opus-4-8	worker of the continuous-operation system to hour 36; second KernelBench reader	569,967
GPT-6 Astra	gpt-6-astra	auditor of the exact candidate; worker in mixed teams; successor on the take-home	264,180
GPT-6 Luna	gpt-6-luna	lower-cost OpenAI successor in the model-handover study, from October 9	91,219
Qwen3.5-27B	Qwen/Qwen3.5-27B	open-weight successor; bf16 on vLLM	–

produced it. Benchmark results need an *official or protected score*, from the official scorer or from an evaluator that runs the candidate as data in a process that never imports it and that was qualified against attacks before any arm ran. For a comparison the evidence is a *preregistration*, which commits predictions, kill criteria, and analysis code before the first episode, with a positive control where harm is possible and a held-out split frozen by hash [31]. Claims pass a *cross-family audit*, in which GPT-6 Astra reads the exact candidate (the commit, the frozen file, the result rows) and returns READY, and a manuscript number enters on a *trace*.

Definition 3 (Traceable number). A number x in a manuscript is *traceable* if a machine-written result file holds x under the metric, run, model, and statistic its sentence states. A traced statement is a *mismatch* if its file exists and contradicts it, and a *scope error* if the value is right and its n , model, window, or statistic is wrong. We write U for the count of numbers that are none of these.

Candidate types and evidence types are paired by one design rule, under which each gate accepts only evidence that the proposer cannot produce by assertion. A certificate cannot be argued into validity, and an official score does not depend on the proposer’s description of its run. A preregistration is a commit that precedes the run it judges, and a trace points to a file a program wrote; the cross-family audit comes from a model that did not produce the claim. Every gate also rejects unknown outcomes, since a gate that admitted on $\perp$ would admit exactly the candidates whose evidence is missing.

The agents’ record is a working log, kept beside the gated record, in which agents reported results as they arrived. Nine headline results passed their gates before any paper printed them (Table 15).

We write the organization’s objective as admitted yield per unit of budget,

$$\max_O \frac{|\mathcal{R}(B)|}{B} \quad \text{subject to} \quad U = 0 \quad \text{and} \quad G_{\kappa(r)}(r, e_r) = 1 \quad \forall r \in \mathcal{R}(B), \quad (4)$$

in which throughput counts only through what the gates admit.

3.2 Recursion on the organization’s own instruments

Algorithm 1 applies Equation (3) to changes in the organization’s own instruments. At every level the step is the same except for the candidate class and the gate (Table 9).

Under step 3, a revision is admitted only if, on the gate’s samples, it catches at least what its predecessor caught and fires on no recorded success, so each admitted step can only raise the share of failures matched while holding false fire at zero on those samples [11]. Ungated self-modification, such as that of the Darwin Gödel Machine and ADAS, has no such condition [20, 47]. The Darwin Gödel Machine modifies its own code and validates each change on coding benchmarks, and ADAS’s meta agent programs new agents from an archive of earlier ones; both keep a variant that scored higher.

Algorithm 1: **Evidence-gated recursion, one step at level ℓ** . The levels run from component (checks, technique cards) through design (harnesses) and system (a running system’s memory) to substrate (the execution substrate).

Inputs: harness H ; record $\mathcal{R}$ of attempts with their evidence; proposer P ; gate G_ℓ ; library $\mathcal{L}$ of retained components.

- 1: TARGET: classify the failures in $\mathcal{R}$ by cause; keep causes that lie outside the model’s episode.
 - 2: PROPOSE: P writes a candidate c from $\mathcal{R}$; P never sees an answer key.
 - 3: GATE: admit c only if $G_\ell(c, e) = 1$. For a revision c of an incumbent c_0 on recorded failures F and successes S : $\text{catch}(c, F) \geq \text{catch}(c_0, F)$ and $\text{fire}(c, S) = 0$.
 - 4: RETAIN: add c to $\mathcal{L}$ with its provenance and evidence, replacing c_0 .
 - 5: REUSE: serve $\mathcal{L}$ to later tasks, other agents, and successor models.
 - 6: RECURSE: record outcomes of work that used $\mathcal{L}$ in $\mathcal{R}$; at level $\ell + 1$ the candidate is a design and the gate a preregistered comparison against named rivals with a positive control.
-

3.3 Audit loops

The protocol has two audits. In the cross-family audit, which is the claim gate, an auditor of another model family reads the exact candidate and returns READY, CHANGES_REQUIRED, or FAIL. It applies the design diversity of N -version programming [4] to language models. Models of one family share blind spots and score their own outputs higher [35], and larger, more accurate models make highly correlated errors [22]. The *post-hoc audit* sits outside the admission path and reads the whole record from raw artifacts after the run, re-attributing each result to its causes without rerunning anything. Its findings re-enter the record as withdrawals, verdict flips, and new comparisons.

3.4 Execution substrate

Nenya is QWNTL’s execution substrate for long-horizon agent work, and every agent in the organization, the coordinating agent included, ran on it. Nenya keeps runs alive across sessions, processes, and model swaps, and it carries each agent’s commitments and evidence between them. Nenya also enforces the gates. On open-weight models only, it adds a latent layer that has no path to the hosted agents. On Qwen3.5-27B this layer flags off-track agent steps at a per-problem AUC of .943–.950 on its validation trajectories, where TACT reports a peak of .91 for the same metric on its own trajectories [42], and it compacts context to half its budget with no loss of fidelity (.930 against .923 for the full context).

Validity controls. Kernel results come from a protected evaluator that passed 24 qualification controls before any arm ran, accepting two legitimate kernels in three positive checks and rejecting all 21 attacks. Attempts in a shared archive are labeled A1, A2, ... as they are recorded. A43 is the 3.97-hour C_1 construction, and its label also names the released C_1 witness. Constructions were scored by a hash-pinned evaluator and recomputed by an independent exact checker sharing no code with it or with the optimizer. Three exact implementations agree on A43. The 40.75-hour C_1 construction was checked in exact arithmetic after the run. Predictions, kill criteria, and analysis code were committed before the runs they judged, and held-out splits were frozen and hashed. In the verification study, none of 7,398 prompts rescanned by the other model family contained grader artifacts. Headline statistics were recomputed from raw artifacts by independent analysis code, and a compute-only control of 94 worker processes ran the classical toolbox on the same constructions with no language model.

Statistics. We report 95% intervals unless marked. For the record as a whole we use a one-sided binomial test of the count of significant wins, treating the 78 comparisons as independent with probability .025 each under no effect. Each companion states its own tests and resampling units.

4 How the organization ran

4.1 Admissions arrived through the gates over the whole run

The gated admissions behind the headline results span the gateway window from its first day to its last hour (Figure 2), and the first C_1 claim below the then-public record passed its gate 16.5 hours into the window. The take-home finals passed the official suite and the stress tests at hour 21.9. A43 was verified in exact arithmetic by hour 22.6. The

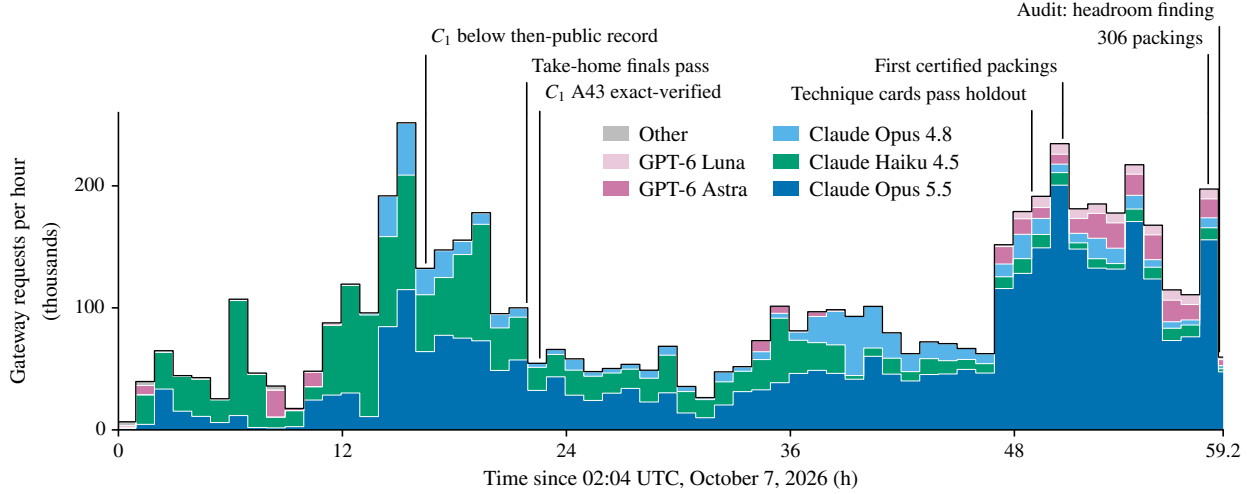


Figure 2: **Gated results reached the record from hour 16 to the last hour of the gateway window.** Gateway requests per clock hour over the 59.2-hour window (5,870,586 requests), stacked by model. Claude Opus 5.5, the model of the worker and lead agents, carried 3,341,071 requests, and GPT-6 Astra, the auditor model and a worker in mixed teams, carried 264,180. Vertical lines mark when the gates admitted the results behind the headline claims and when the post-hoc audit recorded its headroom finding. Hour 0 is 02:04 UTC on October 7, 2026. The first and last bins are partial (56 and 15 minutes), and we draw them as recorded. Open-weight models do not pass the gateway.

technique cards passed their untouched holdout at hour 48.9. The first certified packings followed at hour 50.6 and the 306th packing size at hour 58.4. At hour 59.0 the post-hoc audit recorded the finding that became the headroom relation.

Claude Opus 5.5 carried 3,341,071 of the 5,870,586 requests (Table 4). GPT-6 Astra, the auditor family and a worker in mixed teams, sent 264,180 requests. Its auditors read the exact candidate, and its traffic was almost all input, 8.77 billion tokens against 0.09 billion output tokens.

4.2 Results came from exploration and from reuse

Role specialization gave the exploration-exploitation split of organizational learning [28] to different agents. Exploratory workers, whose brief was to read recent literature and test ideas outside their team’s plan, produced six of the nine results the agents listed as headlines, including the 306 packings and the technique cards as well as the C_3^+ bound. Planned arms produced the other three, the two audited bounds from the shared archive (on C_1 at 3.97 hours and on M) and the take-home result.

Agents also built on one another’s work through the shared record. On the open problems, 60 of a team’s first 98 new bests were built on another agent’s attempt or tool [10]. A43 is one of them. Agent 2 built it 50 minutes after agent 1’s attempt A39 by reusing A39 and agent 3’s kernel through the shared archive (Section 6.5). On the take-home, the same archive let improvements by different agents compose, and the archive arm reached a median of 914 cycles against 946 for the best of four isolated agents.

4.3 The gates turned candidates back while the run was live

Before any paper existed, the first cross-family audit of the packing claims returned the 193-record group as not ready, because its historical coverage was incomplete and 110 exact sums were mislabeled. A second audit passed all 306 sizes, 113 of them new and 108 improved since the first claim. It kept 15 claims for packings in other containers out of the record. The cross-family audit of a C_3^+ claim on 102,400 values found the value correct and below every public value but returned it for changes, because the witness exceeded the Einstein Arena’s 100,000-value limit. The bound the companion reports sits on 97,000 cells. Both audits of A43 accepted its value and returned its credit line for changes.

The protected evaluator rejected all 21 attacks in its qualification suite before any kernel arm ran. The post-hoc audit found one isolation leak, which favored a rival. The ported ACE playbook had learned a template’s recipe from official

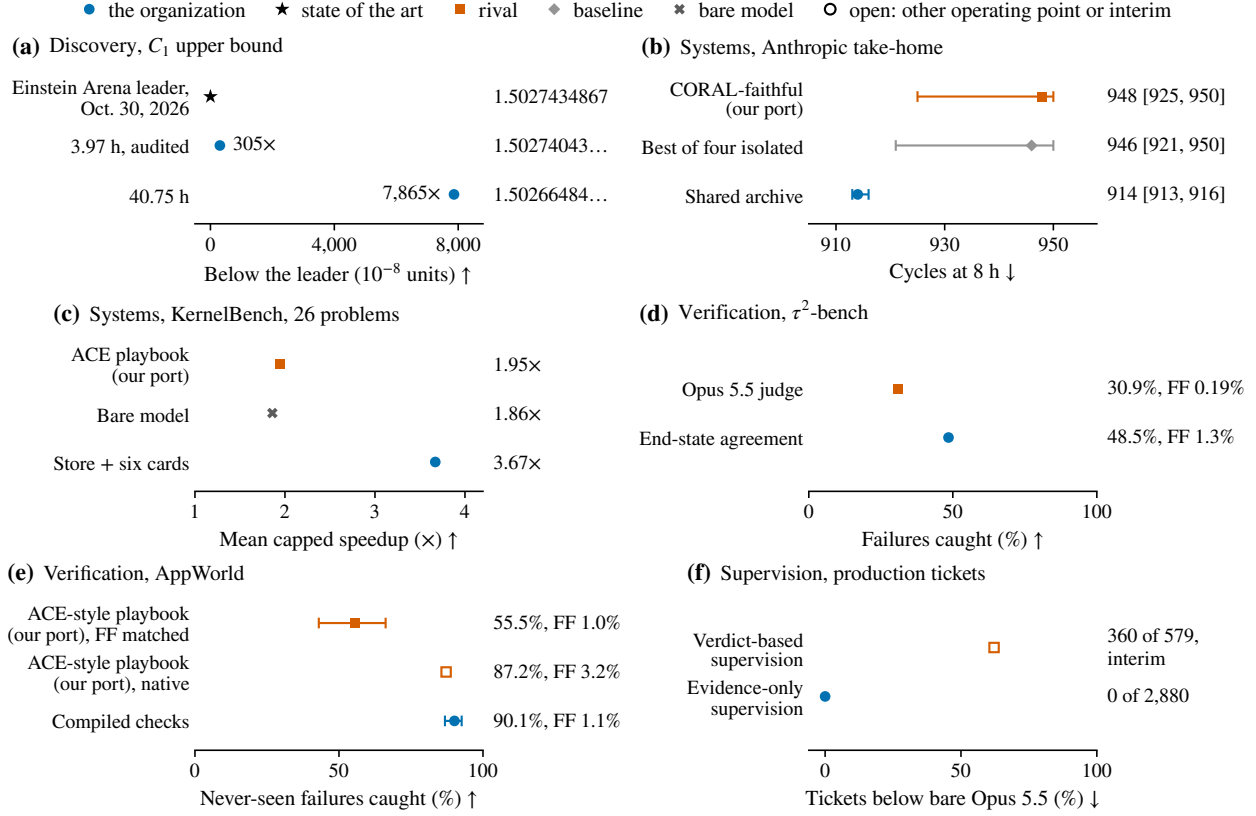


Figure 3: **In every family the organization’s result passes the named state of the art or, where none is published, the strongest baseline run on the same model, on that family’s own verifier.** The key above the panels gives each role’s marker, and we label our runs of a published method “our port”. Arrows on the axes mark the better direction, and FF is false fire. (a) C_1 upper bound, where lower is better, drawn as the distance below the Einstein Arena’s leader on October 30, 2026 in units of the Arena’s minimum improvement, for the 40.75-hour construction and the 3.97-hour construction audited by GPT-6 Astra. (b) Anthropic’s take-home, cycles at 8 hours, median of three seeds; whiskers span the minimum and maximum. (c) KernelBench, mean capped speedup on 26 problems, Claude Opus 5.5. (d) τ^2 -bench, failures caught on 600 Claude Opus 5.5 attempts at 60 held-out tasks, with false fire. (e) AppWorld, never-seen failure states caught, with false fire per never-seen correct state (the compiled checks’ 7 false fires are 1.1% of those states and 0.026% of correct attempts). Whiskers are 95% bootstrap intervals over template clusters, and the open marker is the playbook’s native operating point. (f) Supervision, share of tickets below the unsupervised Claude Opus 5.5 on six fresh streams. The open marker is interim because the verdict-based supervisor’s runs were incomplete at the pause.

grader text, and without that template’s two tasks, two comparisons the organization had lost to it become a tie and a narrow gain.

5 RQ1: new results against named states of the art

In each of the four families, the experiment’s output exceeds a named state of the art or, where none is published, the strongest baseline run on the same model, on that family’s own verifier (Table 6, Figure 3). The companion papers add each result’s mechanism, controls, and statistics.

Discovery. Published optima of these minimax problems are stationary, with their maximum attained, to within the venue’s minimum improvement, at thousands of points, or with their circles at a KKT point, where descent must lower every near-active constraint at once and stalls. The organization’s agents moved off these optima by changing the active set and re-solving the new basin [8]. Three Claude Opus 5.5 agents sharing an archive of verified attempts produced

Table 6: **New results against named states of the art, each on its family’s own verifier.** The worker model is Claude Opus 5.5 in every row except AppWorld, where Claude Opus 5.5 wrote the checks and the graded attempts come from several models. Rival values are the rival’s own reported numbers or the live leaderboard on the stated date; “our port” marks a rival method, and “in-house” a baseline, run in the organization’s harness on Claude Opus 5.5.

Family	Problem and metric	Named state of the art	Organization	Paper
Discovery	C_1 upper bound ($\downarrow$)	Einstein Arena leader 1.502743486709583 (October 30, 2026); MILO 1.5027435984	1.50266484 . . . (40.75 h); 1.5027404359 . . . (3.97 h, audited by GPT-6 Astra)	[8]
Discovery	sum of radii, unequal circles in a square ($\uparrow$)	best value located at each size (October 30, 2026)	best value at 306 of Packomania’s sizes, $N = 137$ to 10,000, as of October 30, 2026	[8]
Discovery	C_3^+ upper bound ($\downarrow$)	MILO 1.4488860143; every Einstein Arena entry listed on October 30, 2026	1.4488670313 . . .	[8]
Discovery	Erdős’s M upper bound ($\downarrow$)	every Einstein Arena entry (October 13, 2026); each compared paper’s value	0.38085654787 . . .	[8]
Systems	Anthropic’s take-home, cycles ($\downarrow$)	AgentDiscover 960 with Claude Opus 5, the lowest model result published in cycles as of October 30, 2026; CORAL-faithful team, our port, 948	median 914 (range 913–916), 3 seeds, 8 h	[10]
Systems	KernelBench, mean capped speedup, 26 problems ($\uparrow$)	ACE’s playbook, our port, $1.95\times$	$3.67\times$ (store and six cards; ACE with the same cards $3.29\times$)	[7]
Verification	τ^2 -bench, failure catch ($\uparrow$)	Claude Opus 5.5 judge, in-house, 30.9% at 0.19% false fire	48.5% at 1.3% false fire, zero model calls per check	[11]
Verification	AppWorld, catch of never-seen failure states ($\uparrow$)	ACE-style verification playbook, our port, 55.5% at matched false fire	90.1%	[11]
Supervision	tickets below the unsupervised model ($\downarrow$)	verdict-based supervision, in-house: 360 of 579 reached (interim)	evidence-only supervision: 0 of 2,880	this paper

a construction certifying $C_1 \leq 1.50266484 . . .$ after 40.75 working hours, inside the gateway window, and its exact certificate was checked after the run. The bound sits 7,865 times the Einstein Arena’s minimum improvement below the Arena’s leader as of October 30, 2026. An earlier construction from the same archive, verified in exact arithmetic and audited by GPT-6 Astra during the run, certifies $C_1 \leq 1.5027404359 . . .$, 305 times the minimum improvement below the same leader.

The same moves set the best sums of radii for unequal circles in a square at 306 of Packomania’s sizes up to $N = 10,000$, as of October 30, 2026, and lowered C_3^+ to 1.4488670313 . . ., below MILO’s value and every Einstein Arena entry listed on that date. On Erdős’s minimum overlap problem the experiment certified $M \leq 0.38085654787 . . .$, below every Einstein Arena entry as of October 13, 2026 and below the value reported by each paper its companion compares. A preregistered classical optimizer with no language model, started from the same Arena entries on the same machines, moved about three thousand times less far below them than the agents in the same elapsed time (2,938 times for C_1 at 3.97 hours).

Systems optimization. On Anthropic’s performance take-home, four Claude Opus 5.5 agents on a shared verified archive reached a median of 914 cycles over three seeds in 8 hours, below AgentDiscover’s 960 with Claude Opus 5, the lowest model result published in cycles as of October 30, 2026. On matched machines, a CORAL-faithful team reached 948 and the best of four isolated agents 946 [10]. The archive keeps every evaluator-verified attempt with its source and bounds each agent’s conversation, and improvements by different agents compose through it. The baselines’ medians stall at 946–956 cycles after 3 hours. Of 12 CORAL-faithful agents, 7 entered a no-op resume loop.

On KernelBench, scored by the protected evaluator, six admitted technique cards lifted our port of ACE’s playbook from $1.95\times$ to $3.29\times$ mean capped speedup on 26 problems outside the card-search trials. Of these, 10 had been evaluated before. A store of verified kernels with the same cards reached $3.67\times$ with 10% of ACE’s input tokens [7]. Predecessor anchoring is a new effect, negative weak-to-strong memory transfer, in which a store built from a weaker model’s verified work lowers a stronger model’s score. In a randomized test in the same study, a store built from Claude Opus 4.8’s verified kernel work lowered Claude Opus 5.5’s log speedup by 0.103 [0.048, 0.163] over 92 problems and 1,288 episodes. Shown the store, Claude Opus 5.5 moves from its own CUDA C++ kernels to its predecessor’s library-plus-Triton designs.

Verification. Claude Opus 5.5 checking its own work accepts 65 of 66 failed attempts on the AppWorld tasks it fails in every repeat. The companion study’s two done-checks make no model call per checked attempt [11]. End-state agreement flags an attempt whose final environment state is not the most common among other attempts at the task. On 600 Claude Opus 5.5 attempts at 60 held-out τ^2 -bench tasks, it catches 48.5% of failures at 1.3% false fire, against 30.9% for a Claude Opus 5.5 judge shown the same attempts, a preregistered gain of 16.5 points in Youden’s J [+5.4, +32.1]. On 31,434 graded AppWorld attempts, checks compiled per task template from past pass/fail outcomes catch 90.1% of never-seen failure states and fire on 0.026% of correct attempts. An ACE-style verification playbook catches 55.5% at matched false fire.

Supervision. In the long-horizon software study, which we report here, an agent builds an account-balance service through 120 dependent tickets, and a hidden oracle scores each one for production correctness on streams that were generated and validated, then hashed before any arm ran. On six fresh streams with Claude Opus 5.5 throughout, a verdict-based supervisor ran unchanged as a preregistered positive control; it combined a requirements list and a model judge with compiled tests and rollback. Supervision built on verdicts made the frontier model worse at the work it supervised, with the supervisor below the unsupervised model on all six streams and, in the data we analyzed at the pause, on 360 of the 579 tickets it had reached. From ticket 41 its mean production correctness was 0.344 below the unsupervised model’s (95% interval over six streams [0.240, 0.448]; interim, since its runs were incomplete at the pause). Forensics of 12 earlier runs traced each of the verdict harness’s 24 first failures to a component of the harness and none to the model.

Three evidence-only harnesses, which act only on recorded evidence, held Claude Opus 5.5 at its unsupervised level on all 2,880 tickets, where the unsupervised model scored 1.000 on every ticket. When Qwen3.5-27B took the service over at ticket 40, the unsupervised model’s mean production correctness over tickets 41–120 fell to 0.435. Evidence-only harnesses that commit the deployed tree at every ticket held it at 0.880–0.911. Two of these three harnesses came from the no-harm panel. The third is a revised version of its remaining harness.

6 RQ2: what the governance mechanisms caught

6.1 What each gate caught

No governance mechanism in Table 7 lets the proposer judge its own result, because the exact verifier is a program and the protected evaluator is pinned, and the other three gates are a commit that precedes the run, an agent of the other model family, and a separate tracer that works from the files alone.

6.2 EGR admitted and rejected changes at four levels

Proposals are cheap at the component level, and the gate is selective there (Table 9). The AppWorld check library proposed 155 new checks and revisions over 38 windows of past outcomes. Its gate admitted 133. A revision enters only if it catches at least what it replaces and fires on no recorded success. In a preregistered comparison on 1,036 never-seen failure states, the gated library caught 90.1% against 66.0% for a copy frozen after 200 failures. By window (Table 8), 84 of the 133 admissions came in windows 1–10 and few in windows 11–32. Windows 33–38, when a new model’s failures arrived, added 27. In each of the five window bins from 5 to 38, the gated library caught more never-seen failure states than the frozen copy, by 12.6 to 43.2 points.

Of 13 KernelBench technique cards, 9 won a paired trial under the protected evaluator. Of these, 6 were admitted to the library and 3 kept as observed cards. The other 4 were rejected. The six-card library took a store built from Claude Opus 4.8’s kernels from $1.73\times$ to $3.67\times$ for Claude Opus 5.5.

At the design level, the supervision harness was rebuilt from the 24 traced first failures of its verdict-based predecessor, and three rebuilt harnesses passed a preregistered no-harm panel with a positive control. The system-level candidates were three repairs to the memory store of a continuous-operation system whose worker was upgraded from Claude Opus 4.8 to Claude Opus 5.5 at hour 36. The gate rejected one that harmed the new model. It admitted a provenance rule that left the store below bare Claude Opus 5.5 (mean capped speedup -0.14 [$-0.23, -0.05$], 90%, 26 problems), and it admitted the technique-card library after a preregistered offline test. The library deployed at hour 36.16. After the run, three change sets to the execution substrate were admitted following review by an agent that did not build them.

Table 7: **The governance mechanisms and what each caught.** Counts are over the run and the number traces of the six papers written from its record.

Mechanism	Admits	Gate predicate (evidence accepted)	Caught
Exact verifier	constructions	witness frozen with its hash; re-scored in exact rational arithmetic by a checker sharing no code with the optimizer; a violation cannot raise a score	malformed numbers, tampering, tolerance-inflated radii, and overflow in qualification; the fast checker matched the pinned evaluator on all 778 tests the pinned evaluator completed and on 174 adversarial cases
Protected evaluator, official scorer	benchmark results	candidate run as data by a process that never imports it; evaluator fingerprint pinned before search	21 of 21 attacks in KernelBench qualification; 88 of 89 mutants killed by the supervision study’s hidden scorer
Preregistration	comparisons	predictions, kill criteria, and analysis code committed before the first episode	837 of 5,849 commits mention preregistration; the 14 experiments behind the headroom relation’s 21 comparisons were each preregistered before their runs
Cross-family audit	claims	GPT-6 Astra reads the exact candidate	11 FAIL and 24 returned for changes, of 60 claims
Number trace	manuscript numbers	value with its metric, run, model, and statistic (Definition 3)	32 contradicted statements and 41 scope errors in 693 traced claims; 0 untraceable
Isolation	a precondition	proposer context excludes answer keys, hidden scorers, and grader output	0 of 7,398 verification-study prompts contained grader artifacts; one leak, found by the post-hoc audit, which favored a rival (Section 4.3)
Post-hoc audit	the record	re-attribution from raw artifacts, with no reruns	6 inflated positives, 4 verdict flips, 3 withdrawals; 7 verdict-moving bugs and 13 more listed as moving or qualifying a verdict, one of them suspected

Table 8: **The AppWorld check library under its gate, by window of past outcomes.** Proposals are new checks and revisions written by Claude Opus 5.5. The gate admits a revision only if it catches at least what it replaces and fires on no recorded success. Catch is on failure states the library had never seen, and windows 1–4 precede the frozen copy.

Windows	Proposed	Admitted	Rejected	Never-seen states	Gated library	Frozen after 200
1–4	58	44	14	150	7.3%	–
5–10	45	40	5	277	83.4%	70.8%
11–18	10	8	2	171	94.7%	70.2%
19–26	7	7	0	181	95.6%	74.6%
27–32	7	7	0	148	94.6%	79.7%
33–38	28	27	1	259	87.6%	44.4%
All	155	133	22			

6.3 A second model family failed the organization’s own claims

Into the Unknown inspected 64 agent-written submissions by hand and found lenient self-verification in 58 of them [23]. GPT-6 Astra, the organization’s claim gate from the other model family, audited 60 of the organization’s claims on the exact candidates and failed 11 (Table 11). It accepted 25 at a stated scope and returned 24 for changes. In one failed claim, a check-library team had read a catch rate of 205 of 215 never-seen failure states as a 21.5-fold reliable horizon

Table 9: **Proposals and admissions at each level of the protocol, and the effect of admitted changes on later work.** Proposals were written by agents, during the run except at the substrate level, and admitted or rejected by the gate for their level.

Level	Candidates	Gate	Proposed / admitted	Effect on later work
Component	new checks and revisions, AppWorld, 38 windows	catches at least what it replaces; fires on no recorded success	155 / 133	never-seen failure states caught: 66.0% (copy frozen after 200 failures) $\rightarrow$ 90.1% (gated library), $n = 1,036$, preregistered
Component	KernelBench technique cards	paired trial under the protected evaluator	13 / 9 won (6 admitted, 3 kept as observed); 4 rejected	store $1.73\times \rightarrow 3.67\times$; ACE’s playbook (our port) $1.95\times \rightarrow 3.29\times$; 26 problems outside the card-search trials, 10 evaluated before
Design	supervision harnesses rebuilt from 24 traced failures	preregistered no-harm panel with a positive control	3 / 3	tickets below the unsupervised model: 360 of 579 reached (predecessor, interim) $\rightarrow$ 0 of 2,880
System	repairs to a running system’s memory across Claude Opus 4.8 $\rightarrow$ Claude Opus 5.5	preregistered offline test, then live deployment	3 named / 2	offline $1.73\times \rightarrow 3.67\times$; live, hours 36–44, $+0.70\times$ over bare on 10 probe problems (interim)
Substrate	change sets to the execution substrate, after the run	review by an agent that did not build them; live product tests	3 / 3	not measured

Table 10: **One claim from proposal to rejection.** The 21.5-fold horizon reading of a check library’s catch rate. Times UTC, October 2026.

Time	Actor	Event	Gate
7th 19:52	check-library team	preregisters the prequential library study: 38 windows, frozen-copy arms, a past-only gate	preregistration
7th 21:26	check-library team	result file: catch of never-seen states 95.4% in windows 34–38 (205 of 215), reported as a 21.5-fold horizon	none
8th 12:14	coordinating agent	the 21.5-fold figure enters the agents’ record	none
9th 12:21	GPT-6 Astra auditor	FAIL as a measured horizon: $1/(1 - 205/215)$ assumes every caught failure is repaired; coverage claim returned for changes	cross-family audit
9th 12:58	post-hoc audit	records that the figure holds only under perfect repair	post-hoc audit
13th–30th	papers	no paper reports the figure as a result; the companion prints the preregistered 90.1% vs. 66.0%	number trace

Table 11: **GPT-6 Astra’s verdicts on the organization’s claims**, audited on the exact candidate, by claim group. Accepted: READY at a stated scope; revisions required: CHANGES_REQUIRED; rejected: FAIL.

Claim group	Accepted	Revisions required	Rejected
Tool-use verification (AppWorld)	3	8	1
Offline verification	5	5	2
Environment-change detection	7	3	0
Harness evolution	0	2	0
Trained components	7	2	3
Other components	3	4	5
All 60 claims	25	24	11

and carried the figure into the agents’ record. GPT-6 Astra failed it because 21.5 is $1/(1 - 205/215)$, which holds only if every caught failure is repaired without new harm. The same model gives 7.05 at a repair probability of 0.90 and 1.91 at 0.50, and neither the repair probability nor the horizon was measured. No paper reports the figure as a result, and the companion prints the preregistered 90.1% against 66.0% in its place (Table 10).

Agent4Sci re-audited 40 of its runs with auditors from three model families and flagged none [49]. In a paired development test on four long-lived Claude Opus 5.5 defects in a shared software system, a GPT-6 Astra audit repaired 14 of 14 paired defect states, and Claude Opus 5.5 auditing its own work repaired 5 of 14 (Table 12). Two of the defects were complete blind spots of the family. Claude Opus 5.5 repaired them 0 of 8 times, and GPT-6 Astra 8 of 8.

Table 12: **What the gated mechanisms add, from the organization’s own runs.** Each row removes or replaces one gated mechanism or design and measures the same outcome. The auditor-family rows are paired development tests. Of 45 Claude Opus 5.5 self-audits, 31 hit a 40-call cap, against none of 33 GPT-6 Astra audits.

Gate varied	Outcome	With the gate	Without
Auditor family (GPT-6 Astra vs. Claude Opus 5.5 self-audit)	defect states repaired, four Claude Opus 5.5 defects	14/14	5/14
Auditor family, blind-spot defects	defect states repaired	8/8	0/8
Evidence-gated revision vs. a copy frozen after 200 failures	never-seen AppWorld failure states caught	90.1%	66.0%
Verified technique cards on a predecessor’s store	Claude Opus 5.5 mean capped speedup, 26 KernelBench problems	3.67×	1.73× (bare 1.86×)
Evidence-only vs. verdict-based supervision	tickets below the unsupervised Claude Opus 5.5	0 of 2,880	360 of 579 reached (interim)

Table 13: **Final number traces of the six papers written from the record.** Each row traces one statement’s numbers, certified with their metric, run, model, and statistic (Definition 3), for the October 13 release builds. Every mismatch and scope row was edited before the release build.

Paper	Rows	OK	Mismatch	Scope	Untraceable
Supervision study (reported here)	120	105	4	11	0
Headroom relation	112	95	10	7	0
New constructions	119	112	1	6	0
Predecessor anchoring	120	109	8	3	0
Shared verified archives	82	73	3	6	0
Zero-call verification	140	126	6	8	0
Six papers	693	620	32	41	0

6.4 The number trace is strict

Into the Unknown measures integrity per submission. Of the 135, 114 failed, and 85 printed numbers their own evidence contradicts [23]. The organization’s number trace checks integrity per number, and in the six papers written from its record it checked 693 claims against their metric, run, model, and statistic and found no untraceable number (Table 13). It flagged 73 of the 693 claims, 10.5%, and every flagged statement was edited before the release build. Earlier passes over the same papers flagged 211 of 1,472 rows and blocked two until their sources were found.

Of the 32 contradicted statements, 16 were qualitative claims the evidence did not support, and 7 overstated a result’s provenance or verification. The other 9 were real values attached to the wrong source, unit, window, or arm. In Beyond fabrication’s labeled sample, about four in five unsupported numbers in agent-written papers are real values from the wrong experiment, metric, or run, and a check for the value’s presence accepts 87–95% of them [1]. The trace, which binds each value to its metric, run, model, and statistic, caught these nine.

6.5 One result from proposal to certified record

Table 14 follows the C_1 bound A43, below the Einstein Arena’s leader as of October 30, 2026, through every gate, from the preregistration of the arm that found it to the audit that corrected its credit line. We took every time and value from the organization’s commits and frozen files.

A43 is the work of three agents. Agent 2 built it through the shared archive from agent 1’s attempt A39 and agent 3’s kernel, 50 minutes after A39. Both audits found the same defect, a credit line that named only the claimant’s own tool, and the lead rewrote the credit before any paper printed the bound. The value never changed, and three exact implementations and the Arena’s published verifier agree on it. The gate and pass time of nine headline results are in Table 15.

Table 14: **One C_1 construction from proposal to certified record.** Times UTC, October 2026. Values are C_1 upper bounds ($\downarrow$); the Einstein Arena’s minimum improvement is 10^{-8} .

	Time	Actor	Logged value or artifact	Gate
1	7th 20:31	shared-search lead	preregisters a three-agent Claude Opus 5.5 arm on a shared archive, 12 s before launch	preregistration
2	7th 20:31	three Claude Opus 5.5 agents, effort xhigh	launch; scorer hash pinned; 37 of 37 qualification cases pass	evaluator qualification
3	7th 23:42	agent 1, attempt A39	1.5027410119	protected evaluator
4	7th 23:56	agent 2	reuses A39 from the archive and agent 3’s kernel	archive provenance
5	8th 00:32	agent 2, attempt A43	1.5027404360 at 3.97 active hours	protected evaluator
6	8th 00:32–00:38	exact verifier	1.5027404359227669764..., an exact fraction	exact arithmetic
7	8th 00:49	shared-search lead	candidate frozen with its SHA-256	frozen file
8	8th 01:19	fresh Claude Opus 5.5 auditor	value READY; credit CHANGES_REQUIRED	same-family audit
9	9th 01:55	GPT-6 Astra auditor	own exact code, same fraction; value READY; credit CHANGES_REQUIRED	cross-family audit
10	9th 01:55	shared-search lead	credit to A39 and agent 3’s kernel written into the claim	correction applied
11	13th	companion paper	prints the exact fraction; number trace OK	number trace

Table 15: **Each of these nine headline results passed its gate before its first paper.** Times UTC, October 2026. First paper: the first manuscript that prints the value. Cross-family: GPT-6 Astra’s verdict on the exact candidate where one was run.

Result	Gate for its type	Gate passed	First paper	Cross-family
$C_1 \leq 1.5027404359 \dots (3.97 \text{ h})$	exact verifier	8th 00:32–00:38	13th	value READY, 9th 01:55
$C_1 \leq 1.50266484 \dots (40.75 \text{ h})$	protected evaluator; exact checker	9th 13:14; exact 30th 11:20	30th 11:31	–
306 circle-packing records	exact checker and independent verifier	9th 04:39 to 12:27	13th	READY, 9th 14:19
$C_3^+ \leq 1.4488670313 \dots$	exact scorer	9th 13:59	13th	–
Take-home, 914 cycles	official scorer and stress suite	7th 23:56	13th	–
Technique cards, $3.67 \times$	preregistration, protected evaluator	9th 09:33	13th	numerics READY, 9th 10:09
End-state agreement, 48.5%	preregistration, official outcomes	9th 02:03	13th	PASS, 9th 02:24
Compiled checks, 90.1%	preregistration, official outcomes	8th 01:25	13th	READY, 9th 12:21
Predecessor anchoring, -0.103	preregistration, protected evaluator	9th 08:12	13th	READY, 9th 09:33

7 RQ3: the comparison record, audited

The organization kept every comparison it ran, losses included, in one record of 195 rows. Each row names the arms, model and instrument of a comparison together with its outcome and source, and a correction replaces the value it corrects. We use the record to test the strongest alternative explanation of the RQ1 results, that an organization running many comparisons reports some wins by chance and, since it reports its own work, selects them.

7.1 Nineteen wins where multiplicity predicts two

We test the 78 Claude Opus 5.5 component comparisons that carry a z score in the corrected record (Table 16). A win is $|z| > 1.96$ in favor of the organization’s arm. Under no effect each comparison wins with probability .025, and 1.95 wins are expected. The record holds 19, with probability 5.9×10^{-14} under no effect (Figure 4a). Against a control without the component, 18 of 26 comparisons favor the component and none favors the control.

The 18 significant losses stay in the record beside the wins, and we measured their causes. Twelve are against rival systems, which recover their own share of the model’s residual error before a new component adds to it, and four serve experience carried from a weaker model, the predecessor anchoring of Section 5.

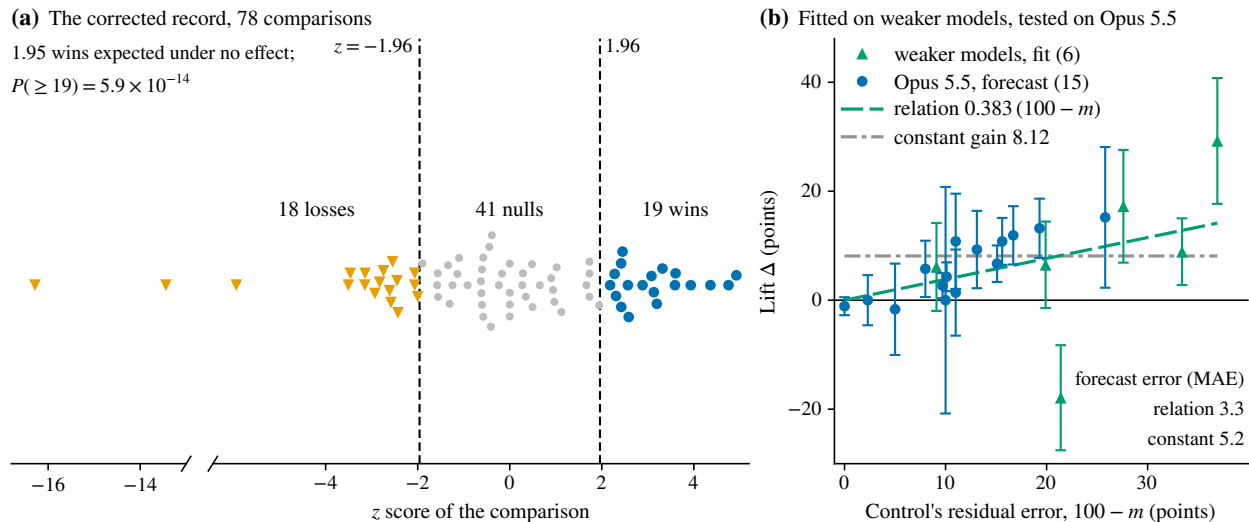


Figure 4: **Wins in the audited record exceed what multiplicity predicts, and a relation fitted on weaker models forecasts Claude Opus 5.5.** (a) The 78 Claude Opus 5.5 component comparisons with a z score in the corrected record, one point each (positive: the organization’s arm better), with dashed lines at $|z| = 1.96$. There are 19 wins, 41 nulls, and 18 losses, against 1.95 wins expected under no effect. Vertical position only separates overlapping points, and the axis is broken between -13.0 and -6.45 . (b) Lift Δ over the same model without the component against the control’s residual error $100 - m$, in points of task success. We fit the relation $\Delta = 0.383(100 - m)$ on the six weaker-model comparisons (five Claude Haiku 4.5, one Claude Opus 4.8), and it forecasts the 15 Claude Opus 5.5 comparisons, one of them a stream that switched from Claude Opus 4.8 to Claude Opus 5.5; the constant gain of 8.12 points is fitted on the same six. Bars are ± 1.96 times each row’s recorded standard error. For the negative weaker-model row that error (4.92) corresponds to all 168 tasks of its stream. At the 84 tasks of its contrast it is 6.96. Mean absolute error on the 15: 3.3 points for the relation, 5.2 for the constant gain.

Table 16: **Nineteen wins against 1.95 expected under no effect.** The 78 Claude Opus 5.5 component comparisons with a z score in the corrected 195-comparison record. We use a one-sided binomial test with $p = .025$ per comparison under no effect and treat the 78 as independent.

Test	Observed	Under no effect
Significant wins among 78 comparisons	19	1.95 expected; $P(\geq 19) = 5.9 \times 10^{-14}$
Against a control without the component	18 of 26 for; 0 against	—
Significant losses kept in the record	18	1.95 expected
of which against rival systems	12	—
of which serving a weaker model’s experience	4	—

7.2 Selection acts on first-claimed positives

After the run, a post-hoc audit team of eight agents, three of them GPT-6 Astra, re-attributed every result in the record from raw artifacts without rerunning any experiment. Its mandate was to re-examine negative results. The team attributed each of 50 negative results to its causes and found 6 claimed positives inflated or invalid. Four verdicts flipped when exclusions were applied symmetrically to both arms. Three positives were withdrawn from the record. In the retests the agents themselves ran, which the audit coded from the record, first-claimed positives held in full in 6 of 16 and in part in 2, and first-claimed negatives held in 6 of 8.

Although the audit was sent to re-examine negatives, it withdrew three positives, and in the agents’ own retests negatives held more often than positives, which places the selection effect on first-claimed positives. The protocol counters this selection at admission, where a comparison enters on a preregistration that fixes its exclusions before the run and a

claim enters on an audit by a family that did not produce it.

7.3 The headroom relation forecasts the next model out of sample

The post-hoc audit split each negative verdict’s losing units across their causes. Averaged over 45 negatives, the model accounted for 0.42 and statistical power for 0.26. The smaller shares went to the ceiling at 0.14, the rival’s advantage at 0.07, harness bugs at 0.07, and infrastructure at 0.03. The attribution also showed that the negatives follow the model’s headroom. The headroom relation, fitted in its companion [9], takes the form

$$\Delta = k(100 - m), \quad (5)$$

where Δ is the component’s lift over the same model without it, m is the control’s task success in points, and k is the share of residual error the component converts. On the 15 Claude Opus 5.5 comparisons we estimate $k = 0.54$ [0.43, 0.68], with AIC 77.2 against 96.8 for a constant gain.

Fitted on six comparisons of Claude Haiku 4.5 and Claude Opus 4.8 alone, Equation (5) forecasts the 15 Claude Opus 5.5 comparisons out of sample, 13 of them on AppWorld, with a mean absolute error of 3.3 points against 5.2 for a constant gain and a Bayes factor of 186 against no effect (Figure 4b). One weaker row’s recorded standard error is computed for all 168 tasks of its stream. Its contrast has 84. When we refit with the error at 84 tasks, the forecast’s mean absolute error falls to 3.18 points and its Bayes factor against no effect rises to 959 [9].

EGR evaluates an admitted change on the model that will use it, because selection scores measured on Claude Haiku 4.5 predict Claude Opus 5.5’s gains at Spearman 0.16 ($p = 0.58$, 14 candidates) and Claude Haiku 4.5’s failures differ from Claude Opus 5.5’s.

8 Related work

From a paper pipeline to an organization. In the AI Scientist and Agent Laboratory, as in FARS and the Agentic Scientist, the output is a paper or report, and the last check is a reviewer’s judgment of the finished manuscript. The AI Scientist goes from an idea through experiment code that it writes and runs to a paper, which it scores with a simulated review [26]. Its second version runs an agentic tree search managed by an experiment-manager agent [46]. Agent Laboratory takes a human-provided research idea through literature review, experimentation, and report writing with agents in four research roles, from PhD student to professor, and its automated reviewers scored its reports 6.1 of 10 where human reviewers gave 3.8 [38]. FARS moves each research direction through ideation, planning, experiment, and writing stages that coordinate through a shared workspace [43]. The Agentic Scientist runs parallel coding agents under a meta-controller that enforces rubric verification at each stage, with an audit agent on a read-only copy of the workspace [24].

Discovery systems. EGR applies mechanical gates in all four of its families, as the Station and AlphaEvolve do for mathematical and algorithmic outputs. In the AI co-scientist, a supervisor agent allocates work to specialized agents that generate, debate, and evolve hypotheses in a tournament [17]. Its execution-grounded follow-up penalizes hallucination and checks claims against raw execution logs, and its invalidating result hallucination fell to 4% (2 of 50 manuscripts), against 46% without its reliability modules and 90% for Agent Laboratory [39]. Kosmos shares a structured world model between a data-analysis agent and a literature agent [29]. AlphaEvolve improves code with an evolutionary pipeline scored by automated evaluators [32]. In the Station, agents from three model families pursue a shared goal without a central coordinator and back every reported discovery with an exact construction or a Lean proof, producing novel results on 5 of 12 AlphaEvolve problems [6].

Scale and integrity. A Society of Researchers organizes ten thousand researcher agents under explicit institutions, with a human governor who allocates resources and assigns no tasks [3]. Agent4Sci runs each agent’s artifacts on held-out tests of 20 scientific-ML benchmarks and has an auditing agent inspect each trajectory for misconduct [49]. Among integrity studies, Into the Unknown finds integrity failures in 114 of 135 submissions [23], and RIDAR finds integrity violations in 99.7% of 300 agent trajectories [40]. In Beyond fabrication’s labeled sample, about four in five unsupported numbers are real values from the wrong experiment, metric, or run [1]. Our protocol types each check by the evidence it accepts and publishes what each check rejected. Fujimura has a reviewer of another family rerun each study from its records [15], and concurrently Mansouri et al. built a laboratory of language-model agents in six roles whose agents assess one another’s work and can stop or redirect investigations [27].

Table 17: **What each autonomous-research system’s output is checked by, and whether it gates changes to itself.** We take each entry from the system’s own text, and “–” means the paper does not report one.

System	Output	Checked by	Organization	Gates changes to itself
AI Scientist-v2 [46]	papers	LLM reviewer; workshop review of 3 selected manuscripts	pipeline with an experiment manager	–
Agent Laboratory [38]	reports	LLM reviewer agents; human reviewers	four role agents, three stages	–
AI co-scientist [17]	hypotheses	tournament and reflection review; expert preference; wet-lab validation	supervisor and specialized agents	no
Real-world follow-up [39]	papers	clipping against execution logs; expert review	co-scientist architecture	no
Kosmos [29]	reports	scientists judge statements	two agent types, shared world model	–
The Station [6]	mathematics	exact constructions, Lean proofs	three families, no coordinator	–
FARS [43]	papers	evaluation agent; human reviews after the fact	four-stage pipeline	–
Agent4Sci [49]	benchmark scores	held-out tests; misconduct auditor	one agent per run	no
Agentic Scientist [24]	papers	rubric checks; read-only audit agent; human review	meta-controller over coding agents	–
Infinity Machine [16]	results	kernel calibration; adversarial audit; blind re-check	proposers and a meta agent	yes; 0 of 1 self-derived admitted
Triage-PiEvo [25]	principle revisions	independent repeat and held-out checks	one search agent	no
Mansouri et al. [27]	experiments	computational verification; precommitted criteria; reviewers	six roles, two model families	–
This organization	constructions, results, comparisons, claims, numbers	exact verifier; official or protected scorer; preregistration; cross-family audit; number trace	coordinating agent, leads, workers, auditors	yes; four levels (Table 9)

Gated self-modification. The Darwin Gödel Machine, ADAS, and the Huxley-Gödel Machine search over agent designs and keep what scores higher [20, 45, 47]. In Infinity Machine, the closest predecessor of EGR, a kernel its agents cannot write gates every change to its harness. That gate rejected the one generation the system derived itself [16]. Triage-PiEvo lets only independently verified evidence rewrite its persistent principles [25].

Organizations of agents. MetaGPT assigns roles along an assembly line of standard operating procedures [19]. ChatDev runs specialized agents through design, coding, and testing phases [36], and in Magentic-One a lead orchestrator plans and tracks progress while directing specialized agents [14]. The organization we studied uses the same coordinator-and-worker pattern, and its gates are institutional rules in the sense of the literature on institutions for collective action [33]. Exact certificates follow proof-carrying code [30], and Polymath ran open mathematics as a public record anyone could check [18].

9 Conclusion

In the organization we ran, each candidate type entered the record through one gate. Constructions and benchmark results entered on a computation, an exact certificate or an official or protected score, and comparisons on a preregistration; claims entered on another family’s audit and numbers on a trace to the file that holds them. The integrity of its output was set at admission, and its results stand on verifiers that make no model judgment. No number in the six papers written from its record was untraceable. Every statement the trace flagged was edited before their release builds, and every construction they print carries an exact certificate. No paper reports as a result the 21.5-fold horizon its auditor failed.

The gated record also works as an instrument, because its 195 comparisons carry their arms, models, and outcomes. A post-hoc audit of them yields a relation that, fitted on weaker models, forecasts a stronger model out of sample. In the next experiment we will run the same organization and protocol on the next frontier model and register the headroom relation’s forecast for every component before the first run. We will also measure the substrate’s admitted changes on that run’s held-out work.

References

- [1] Mosab Alfaqeeh, Omar Hourani, Muath Hourani, and Hashem Alfaqeeh. Beyond fabrication: Misattributed numbers in papers written by AI research agents. *arXiv preprint arXiv:2610.38691v1*, 2026.
- [2] Anthropic. Anthropic’s Original Performance Take-Home. https://github.com/anthropics/original_performance_takehome, 2026. Public kernel-optimisation task scored in simulated machine cycles. GitHub.
- [3] Ali Asaria, Deep Gandhi, and Tony Salomone. A Society of Researchers: Designing Institutions for Populations of Autonomous Research Agents. *arXiv preprint arXiv:2610.10468v1*, 2026.
- [4] A. Avizienis. The N-Version Approach to Fault-Tolerant Software. *IEEE Transactions on Software Engineering*, SE-11(12):1491–1501, 1985. doi: 10.1109/TSE.1985.231893.
- [5] Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2 -Bench: Evaluating Conversational Agents in a Dual-Control Environment. *arXiv preprint arXiv:2506.07982*, 2025.
- [6] Stephen Chung, Wenyu Du, and William J. Wesley. Autonomous Mathematical Discovery in an Open-World Multi-Agent Environment. *arXiv preprint arXiv:2608.23691*, 2026.
- [7] Lindell S. Cumes. Weak-to-Strong Memory Transfer Is Negative: Predecessor Anchoring in Frontier GPU-Kernel Agents. <https://qwntl.com/paper-predecessor-anchoring>, 2026. QWNTL Labs preprint.
- [8] Lindell S. Cumes. A New Upper Bound for the First Autocorrelation Inequality and 306 Circle-Packing Records via Active-Set Escape from Equioscillating and KKT Optima. <https://qwntl.com/paper-new-constructions>, 2026. QWNTL Labs preprint.
- [9] Lindell S. Cumes. Harness Gains Are Linear in Residual Error: A Weak-to-Strong Headroom Law Forecasts Claude Opus 5.5 from Weaker Claude Models. <https://qwntl.com/paper-frontier-report>, 2026. QWNTL Labs preprint.
- [10] Lindell S. Cumes. Scaling Discovery Does Not Require Forgetting: Verified Shared Archives Break the Multi-Agent Plateau on Anthropic’s Performance Take-Home. <https://qwntl.com/paper-societies-horizon>, 2026. QWNTL Labs preprint.
- [11] Lindell S. Cumes. Zero-Call Verification: End-State Agreement Outperforms Claude Opus 5.5 Judges on τ^2 -bench and Compiled Checks Outperform an ACE-Style Verification Playbook on AppWorld. <https://qwntl.com/paper-verification-capital>, 2026. QWNTL Labs preprint.
- [12] EinsteinArena. EinsteinArena: An Open Arena Where AI Agents Collaborate and Compete on Unsolved Science Problems. <https://einsteinarena.com>, 2026. Problems with public verifiers and a live leaderboard with minimum-improvement thresholds. Leaderboard accessed 2026-10-13.
- [13] Mahdi Farahbakhsh, Ilan Sela, Fatemeh Doudi, Vishnu Teja Kunde, Krishna Narayanan, Jean-Francois Chamberland, and Dileep Kalathil. AgentDiscover: Autonomous Discovery with Minimal Search Scaffolding, 2026. URL <https://arxiv.org/abs/2610.05334v1>.
- [14] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. *arXiv preprint arXiv:2411.04468*, 2024.
- [15] Ko Fujimura. Letting AI Formulate the Research Question: Autonomous Research Agents. *arXiv preprint arXiv:2610.39620v1*, 2026.
- [16] Harish Santhanalakshmi Ganesan. Infinity Machine: a self-improving AI agent for research. *arXiv preprint arXiv:2610.35579v1*, 2026.

- [17] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, et al. Accelerating scientific discovery with Co-Scientist. *arXiv preprint arXiv:2502.18864*, 2025.
- [18] Timothy Gowers and Michael Nielsen. Massively collaborative mathematics. *Nature*, 461(7266):879–881, 2009. doi: 10.1038/461879a.
- [19] Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. *arXiv preprint arXiv:2308.00352*, 2023.
- [20] Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems. *arXiv preprint arXiv:2408.08435*, 2024.
- [21] Prithwish Jana, Mononito Goswami, Hao Liu, Xinyu Li, Langlin Huang, Zhehui Huang, Zhishen Huang, Patrick Blöbaum, Anoop Deoras, Purak Jain, Nikos Kanakaris, and Sahika Genc. MILO: Automated Harness Discovery via Orchestrated Multi-Agent Evolution. *arXiv preprint arXiv:2609.38349*, 2026. Project page: <https://jprithwish.github.io/MILO/>.
- [22] Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2506.07962.
- [23] Seungone Kim, Jiayi Geng, Young-Jun Lee, Bo Liu, Juyoung Suk, Nicholas Lubbers, Mitchell Wood, Marcus Noack, Yong-Xin Yao, Yuan Shi, Roxana Bujack, Cristina Garcia Cardona, Austin Rodriguez, Ilon Joseph, Woojin Cho, Zora Zhiruo Wang, Chuanyang Jin, Ian Wu, Apurva Gandhi, Tianjian Li, Weiwei Sun, Carolin Lawrence, Kiril Gashteovski, Dongyeop Kang, Jay Shin, Akari Asai, Graham Neubig, and Sean Welleck. Into the Unknown: Benchmarking AI agents on open scientific research problems. *arXiv preprint arXiv:2610.19159v1*, 2026.
- [24] Robert Tjarko Lange, Luca Grillotti, Yutaro Yamada, Rujikorn Charakorn, and David Ha. The Agentic Scientist: Orchestrating Coding Agents for Automated Scientific (Co-)Discovery. *arXiv preprint arXiv:2610.18933v1*, 2026.
- [25] Peize Li, Yingming Pu, Xingguo Xu, Yiming Cui, Yuansheng Gao, and Tao Lin. Triage-PiEvo: Evidence Admission for Principle-Evolving Scientific Agents. *arXiv preprint arXiv:2610.14683v1*, 2026.
- [26] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- [27] El Mustapha Mansouri, Eikichi Kimura, Masato Koga, and Keigo Arai. Collaborative AI agents for autonomous experimental research with solid-state quantum spins. *arXiv preprint arXiv:2610.40589v1*, 2026.
- [28] James G. March. Exploration and Exploitation in Organizational Learning. *Organization Science*, 2(1):71–87, 1991. doi: 10.1287/orsc.2.1.71.
- [29] Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, Arvis Sulovari, Eric C. Landsness, Daniel L. Barabasi, Siddharth Narayanan, Nicky Evans, Shriya Reddy, Martha Foiani, Aizad Kamal, Leah P. Shriver, Fang Cao, Asmamaw T. Wassie, Jon M. Laurent, Edwin Melville-Green, Mayk Caldas, Albert Bou, Kaleigh F. Roberts, Sladjana Zagorac, Timothy C. Orr, Miranda E. Orr, Kevin J. Zwezdaryk, Ali E. Ghareeb, Laurie McCoy, Bruna Gomes, Euan A. Ashley, Karen E. Duff, Tonio Buonassisi, Tom Rainforth, Randall J. Bateman, Michael Skarlinski, Samuel G. Rodrigues, Michaela M. Hinks, and Andrew D. White. Kosmos: An AI Scientist for Autonomous Discovery. *arXiv preprint arXiv:2511.02824v2*, 2025.
- [30] George C. Necula. Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '97*, pages 106–119, 1997. doi: 10.1145/263699.263712.
- [31] Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018. doi: 10.1073/pnas.1708274114.
- [32] Alexander Novikov, Ngàn Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

- [33] Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990. doi: 10.1017/CBO9780511807763.
- [34] Anne Ouyang, Simon Guo, Simran Arora, Alex L. Zhang, William Hu, Christopher Ré, and Azalia Mirhoseini. KernelBench: Can LLMs Write Efficient GPU Kernels? *arXiv preprint arXiv:2502.10517*, 2025.
- [35] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*, 2024.
- [36] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative Agents for Software Development. *arXiv preprint arXiv:2307.07924v5*, 2023.
- [37] Ao Qu, Han Zheng, Zijian Zhou, Yihao Yan, Yihong Tang, Shao Yong Ong, Fenglu Hong, Kaichen Zhou, Chonghe Jiang, Minwei Kong, Jiacheng Zhu, Xuan Jiang, Sirui Li, Cathy Wu, Bryan Kian Hsiang Low, Jinhua Zhao, and Paul Pu Liang. CORAL: Towards Autonomous Multi-Agent Evolution for Open-Ended Discovery. *arXiv preprint arXiv:2604.01658*, 2026. COLM 2026 (per the authors’ repository). Code: <https://github.com/Human-Agent-Society/CORAL>.
- [38] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent Laboratory: Using LLM Agents as Research Assistants. *arXiv preprint arXiv:2501.04227v2*, 2025.
- [39] Samuel Schmidgall, Xiaokai Zhu, Marian Shaw, Lin Yang, Valentin Liévin, Jingyun Yang, Yuchen Zhuang, Tim Strother, Alex Bijamov, Min Woo Sun, Anil Palepu, Justin Chen, David Steiner, Jacqueline Shreibati, Wei-Hung Weng, Yilin Zhao, Xingjian Hu, Nicholas Zahn, Sadhya Garg, Julia Kirby, Yuxiang Gan, Jiaoli Li, Divy Thakkar, Shekoofeh Azizi, David Racz, Juraj Gottweis, Vivek Natarajan, Chenglin Wu, Tal Danino, Keran Rong, Haozhe Wang, Benoit Schillings, Yong Cheng, Quoc V. Le, and Tao Tu. Accelerating Scientific Research with Gemini in the Real-World. *arXiv preprint arXiv:2608.26701v1*, 2026.
- [40] Jiening Siow, Amy Xin, Junjie Wang, Ziyu Tang, Zijun Yao, Yunjia Qi, Fanjin Zhang, Lei Hou, Jie Tang, and Juanzi Li. RIDAR: Safeguarding Research Integrity in Autonomous Research Agents via Automated Detection and Runtime Intervention. *arXiv preprint arXiv:2610.31372v2*, 2026.
- [41] Eckard Specht. Packomania. <http://www.packomania.com>, 2026. Includes “The best known packings of unequal circles in a square” (csqv, maximising the sum of radii), <https://packomania.com/csqv/>. Accessed 2026-10-13.
- [42] Yuan Sui, Yulin Chen, Yibo Li, Xue Jiang, Yufei He, Yihong Dong, Xiaoxin He, Tianyu Gao, and Bryan Hooi. TACT: Mitigating Overthinking and Overacting in Coding Agents via Activation Steering. *arXiv preprint arXiv:2605.05980v1*, 2026.
- [43] Qiong Tang, Tianxiang Sun, Xiangkun Hu, Xiangyang Liu, Yiran Chen, Yunfan Shao, Bobo Li, Changze Lv, Cheng Xu, Chengsong Huang, Chunyang Li, Dizhan Xue, Hao Bai, Haodong Duan, Hengquan Guo, Hongyang He, Hongyi Chen, Hui Shen, Jiahao Yuan, Jiankai Sun, Jikang Cheng, Jinfeng Xu, Jingqi Tong, Jingye Chen, Jinxiu Liu, Jixuan Leng, Junchi Yu, Kaixun Jiang, Kun Xiang, Kunpeng Yao, Lang Feng, Liangqi Yuan, Longsen Gao, Meng Li, Qi Jia, Qiushi Sun, Shengyuan Ding, Shizhan Gong, Siru Zhong, Terry Jingchen Zhang, Tianle Gu, Tianyi Liang, Weijie Liu, Weikai Yang, Weizhi Fei, Xin Wang, Xinpeng Liu, Xuanwen Ding, Yihong Tang, Yuanli Wang, Yukun Jiang, Yuming Yang, Zhengbao He, Zhikai Chen, Zhikun Xu, Zhuang Li, and Zihao Huang. FARS: A Fully Automated Research System Deployed at Scale. *arXiv preprint arXiv:2606.31651v2*, 2026.
- [44] Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. *arXiv preprint arXiv:2407.18901*, 2024. ACL 2024.
- [45] Wenyi Wang, Piotr Piękos, Li Nanbo, Firas Laakom, Yimeng Chen, Mateusz Ostaszewski, Mingchen Zhuge, and Jürgen Schmidhuber. Huxley-Gödel Machine: Human-Level Coding Agent Development by an Approximation of the Optimal Self-Improving Machine. *arXiv preprint arXiv:2510.21614*, 2025.

- [46] Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. *arXiv preprint arXiv:2504.08066v1*, 2025.
- [47] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Godel Machine: Open-Ended Evolution of Self-Improving Agents. *arXiv preprint arXiv:2505.22954*, 2025.
- [48] Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. *arXiv preprint arXiv:2510.04618*, 2025. ICLR 2026.
- [49] Ming Zhong, Stacy Li, Nicholas Carlini, and Matthew Jagielski. Automating ML for Science: Can Frontier Agents Climb Scientific Hills in the Wild? *arXiv preprint arXiv:2610.28763v1*, 2026.