COMPILED AGENCY: CODING AGENTS AS GAME AI RESEARCHERS — FROM A ROGUELIKE TO STARCRAFT II AND CIVILIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Coding agents are increasingly capable of sustained engineering and empirical research. Greater autonomy makes their research decisions themselves a target for evaluation: what to investigate, which experiments to run, and when to stop. Games provide a controlled, interactive setting with objectively measurable outcomes. We introduce Gauntlet, a develop–freeze–evaluate protocol for studying these decisions as agents build standalone game-playing programs. Starting from a game description, a raw observation/action interface, and an empty policy file, an off-the-shelf agent develops a controller from bare interaction in one autonomous session. We call the capability under study *compiled agency*: the shipped program plays with zero model calls. Every intermediate version is frozen for later scoring on held-out instances, connecting the agent's research process to independently measured performance. The capability is real and advancing: environment access adds 10 to 78 percentage points of held-out success over construction-only controls, and progress across model generations comes in steps, with tiers that defeat one generation entirely falling to the next. The protocol yields StarCraft II controllers that defeat every fair built-in AI, and in the newest generation the strongest cheating tier as well, and Civilization controllers that win complete games by conquest. Replaying more than 5,000 frozen versions exposes the research behind the programs: gains that plateau early; rigorous local investigation beside sparse validation of what actually ships; and stops that follow a race between the agent's own evidence and a model-specific transcript budget it was never given. When validation panels are refreshed mid-session, so that only the evidence changes, shipped success rises 12.4 points in nine of nine completed pairs of twelve initiated. The experiments an agent designs are part of the capability it delivers; Gauntlet makes them measurable and improvable—a step toward agents whose research practice, not just whose code, can be engineered. We release the benchmark, the corpus, and the replay tooling.

1 INTRODUCTION

Coding agents now handle increasingly long-horizon work (Kwa et al., 2025), and increasingly that work is empirical research: probing systems, forming and testing explanations, and deciding when the evidence is sufficient (Lu et al., 2024). Assessing this capability requires connecting the agent's investigations to what its work achieves. A final score alone does not reveal how the capability developed or whether the agent's own evidence justified its conclusions. We therefore seek a setting where agents direct their own investigations, outcomes are free of external confounds, and the programs they produce can be evaluated independently throughout development.

Game-controller development offers such a setting. Games provide interactive feedback and objectively measurable outcomes under experimental control, and successive controllers preserve executable consequences of the agent's development decisions. Existing work evaluates models playing games directly (Paglieri et al., 2025; Hu et al., 2026; Ying et al., 2026) and synthesizes executable policies within supplied representations and refinement procedures (Xu et al., 2025; Deng et al., 2024). We instead leave the entire process to the agent: an off-the-shelf coding agent constructs the complete controller and organizes its own experiments in a single session of bare interaction. The

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

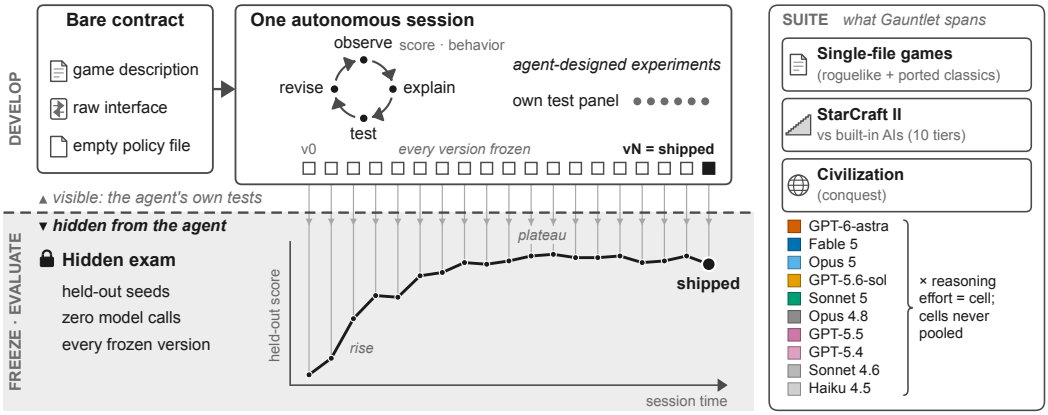


Figure 1: **The Gauntlet protocol.** An agent receives a game description, a raw interface, and an empty policy file, and develops a standalone controller in one autonomous session (left); every version is frozen for replay on hidden held-out instances with zero model calls, at the per-game coverage of Appendix B (center). Right: environments and models; a cell (model × effort × game) is the unit of analysis, never pooled.

game panel spans difficulty and scale: at one end a single-file roguelike whose trials cost seconds; at the other, StarCraft II and Civilization, where local improvements count only once integrated into a winning full-game controller. We call the capability *compiled agency*: the shipped program plays with zero model calls.

Gauntlet, the protocol introduced here, connects the development process to independent evaluation. The agent works under a bare contract (§2), with no stated budget and no clock shown. Intermediate versions are frozen and replayed afterwards on held-out instances unavailable during development. The trace records what the agent tested and concluded; the replay measures what its programs can do; comparing the two tests whether the agent's evidence supports what it ships (Figure 1).

Freezing everything turns qualitative impressions into five measurable questions. What does the research produce? A leaderboard with recurring generational steps and collapsing cost per win (§3). What does it consist of? Auditable observation–explanation–test–revision chains (§4). What do they achieve? Plateauing curves, measurement that tracks the exam beside optimistic final claims, and stale final validation (§5). What ends a session? A race between the agent's evidence and an uninstructed transcript budget (§6). Does changing the evidence change the product? Yes, where headroom and uptake meet (§7).

Our contributions are the protocol and suite, a version-level replayed corpus, the diagnosis above, and one demonstrated remedy. The corpus holds about 5,000 scored versions from roughly 140 development sessions, several hundred audited research episodes, and a 1,110-session stopping inventory spanning 19 games (scoring coverage per game: Appendix B). Throughout, "researcher" describes behavior, not cognition. Every performance statistic is reported within a cell (one model, one reasoning effort, one game and condition); corpus-wide summaries are labeled as inventories, because pooling across cells repeatedly manufactured conclusions that no single cell supports.

2 Protocol, Corpus, and Definitions

Environments. The suite spans three families of games, ordered by the cost of a single trial. The single-file family holds a privately written roguelike that has never been published or shared, and fifteen ported single-file games; their engines are deterministic, so the roguelike's 80-seed held-out exam is exact (identical artifacts receive identical per-seed outcomes), scored as clears (R1) and as a speed-weighted utility (R2; Appendix B). In StarCraft II the agent writes a Python controller against the raw low-level API (Vinyals et al., 2017; Blizzard Entertainment, 2017), with no bot framework provided (e.g., BurnySc2 and contributors, 2019), and is examined on 16 held-out scenarios (unseen maps and seeds) at its development tier. In Civilization it plays full conquest games against built-in

AIs through a web-served engine (Freeciv, via CivRealm; Qi et al., 2024), with a 16-game held-out exam per condition; a single trial takes minutes to hours.

Contract. Every session starts from the same bare contract: the description, the interface, an empty policy file, and general-purpose tools. Nothing else is provided: no baseline, no starter code, no budget, no clock. The prompt says the exam will use unseen instances but not which. Whatever is in the policy file when the session ends is what gets examined.

Cells, harnesses, controls. A cell is model $\times$ reasoning effort $\times$ game. We call the highest-effort cells of the current model generations the *frontier cells*; on the roguelike these have $N{=}6$ sessions each (other cells' N stated where used). Each model runs under its vendor's command-line agent (Claude Code for the Claude models, codex for the GPT models) at a fixed effort per cell, the highest available unless stated; the harness's own context management, including compaction, is part of the environment the agent experiences (§6). Control and instrument runs (one-shot, spec-only, uncapped, and the forked branches of §7) are analyzed separately and never enter cell statistics.

Replay. Intermediate versions are reconstructed from the session traces and checked: the reconstruction's final state must byte-match the shipped artifact, and every snapshot the agent left in its workspace must byte-match the reconstruction at that moment; all reconstructions used here pass both (one session with whitespace-only differences; Appendix B). StarCraft II replays also reproduce the recorded exam of the shipped build in 93% of games (252/272); the shortfall is cross-build engine nondeterminism (byte-identical builds agree 16/16).

Definitions. A *research episode* is a trace segment with an observation, a stated explanation, an executed discriminating test, and a decision (§4). A session is *self-terminated* when the agent ends it; wall-capped, quota-killed, and operator-stopped sessions are excluded from the natural-stop fits, right-censored in the registered hazard analysis, and retained, labeled, in the corpus. Recovery of the agent's own evaluations from traces is in Appendix J.

Disclosures. Despite instructions discouraging it, a small fraction of canonical roguelike trials (30 of 292) read the minified environment source (cf. Zhong et al., 2026); those sessions gain no statistical advantage on the exam (Appendix M).

3 What the research produces

The protocol produces programs worth ranking (Figure 2). On the roguelike, cell means at each model's headline effort run from 0% (Haiku 4.5) to 97.9% (GPT-6-astra, hereafter Astra), and the widest separations in Figure 2a are generational. Within each vendor lineage the matched-cell step is large: Opus 4.8 to Fable 5 goes from 39.0% to 84.8%, and GPT-5.6-sol to Astra from 85.6% to 97.9% (470/480 seeds), the spread collapsing to a four-clear band reached in 0.5–1.1 hours per session against Fable 5's 1.5–7.5 (Appendix F). Construction controls locate the capability: one-shot generation averages at most 27% in its best cell, spec-only construction changes little, and full environment access adds 10 to 78 points depending on the model (Figure 2d); the margin is earned in the environment.

StarCraft II tells the same story against a harder wall. In Figure 2b solved tiers stack at the top (Fable 5 clears Harder perfectly, 64% at Very Hard) while tiers at the edge of a model's reach scatter. At Elite, where Opus 4.8 scored zero, Fable averages 59%, and Astra's three sessions score 12, 16, and 16 of 16, the first perfect exams at that tier. At Cheater:Insane, the highest difficulty, every Fable session shipped a controller that won nothing; Astra ships 16, 5, and 12 of 16, a tier that defeated one generation entirely falling to the next. Session-to-session variance, which we call the *session lottery*, persists exactly where headroom remains. Civilization repeats the pattern at full-game scale (Figure 2c): Fable wins complete conquest games at every condition it attempted, with low, lottery-like counts (normal 1v1 mean 4.7%); GPT-5.6-sol's three novice-condition sessions score 9, 15, and 2; and Astra's three sessions at normal 1v1, the hardest condition attempted, score 16, 16, and 16 of 16 (watchdog reruns: Appendix C).

A frozen controller can also be examined away from its development difficulty, turning each shipped program into a reach profile: Astra's best Cheater:Insane controller (its 16/16 session) sweeps all ten StarCraft II difficulties (59 of 60) while its Elite controllers manage 0–1 of 6 at Insane, and its normal-1v1 Civilization controllers win 74% of decided games across the seven other conditions

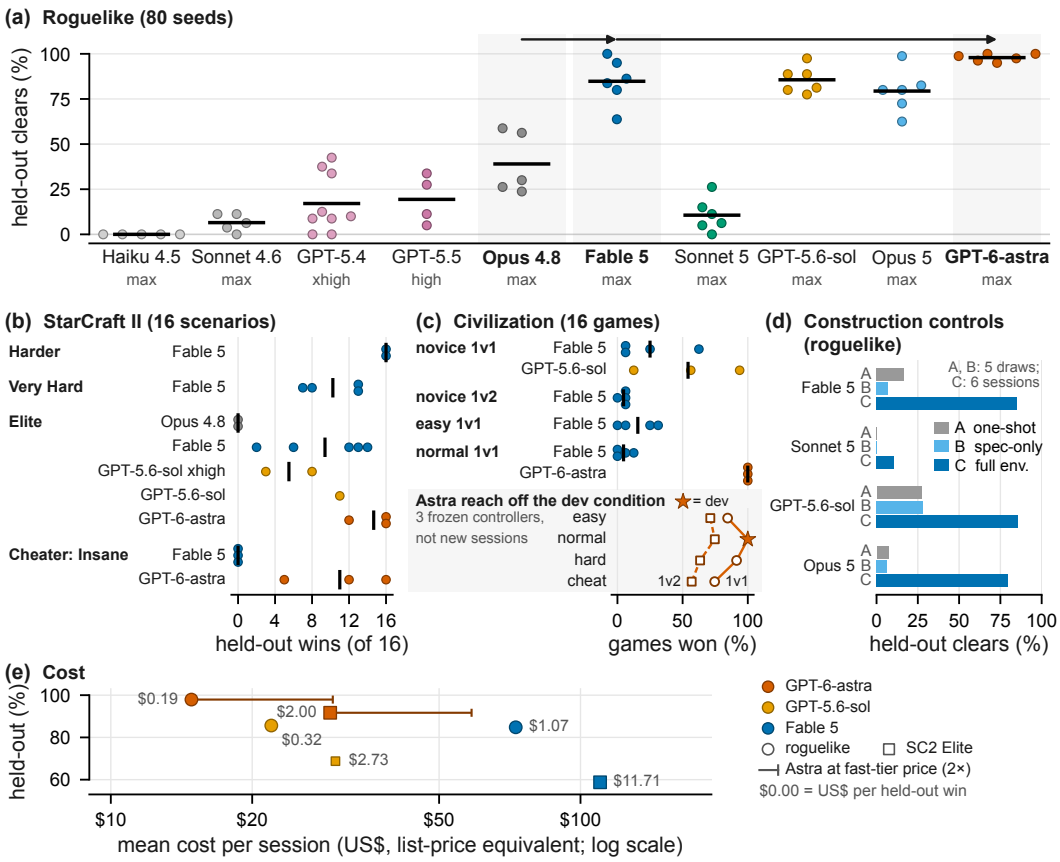


Figure 2: **Shipped controllers across three games. (a)** Roguelike: one dot per shipped controller, per cell at each model's headline effort (GPT-5.5: high; Appendix D); ticks are cell means. **(b)** StarCraft II: held-out wins of 16 per session (recorded exams), by development tier and model. **(c)** Civilization: conquest wins of 16 per session, by condition; inset: Astra's normal-1v1 controllers across seven other conditions. **(d)** Construction controls: one-shot (A), spec-only (B), full environment (C). **(e)** Cell mean vs list-price dollars per session; whisker: Astra's unknown tier up to 2×.

(221/300; 36 of 336 unresolved) against 100% at home, losing ground mainly to a second opponent, a deficit that survives any resolution of the unresolved games (Appendix C). In StarCraft II, reach tracks the development tier.

Final performance is one axis of the leaderboard; the cost of reaching it is the other. In Figure 2e Astra sits far left of Fable 5 at equal or better quality: $14.85 against $72.80 per roguelike session (4.9×), $0.19 against $1.07 per held-out clear (5.6×). List prices per token are identical, so the saving is volume; Astra consumes 17–20% of its comparators' input tokens. From sol to Astra, cost per clear falls from $0.32 to $0.19 while quality rises (Appendix E).

4 What the research consists of

The sessions contain real, countable investigation. We audited research episodes from the agents' own messages and commands (hidden reasoning is never used); an episode requires an executed, discriminating test, not a narration. A keyword-guided pass over 39 sessions labeled 204 episodes; complete re-reads of two Civilization sessions counted 31 and 127 where that pass had found 9 and 7, about one strict episode per policy version the agent played. The corpus therefore holds several hundred episodes, five or so in an hours-long session and over a hundred in a long Civilization one (Appendix F). In the re-read sessions, looser tune-and-test loops add only 10–20%: those agents state a mechanism and run a check for most changes. About a quarter of episodes end with the agent

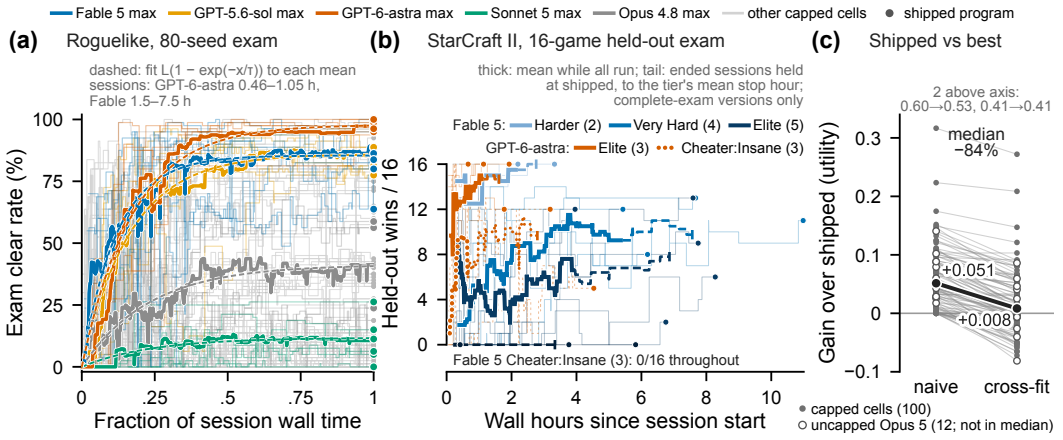


Figure 3: **Measured progress. (a)** Roguelike held-out score against session wall fraction, per cell (100 capped sessions, 4,114 versions, plus Astra's 6 sessions, 99 versions); dashed: the two-parameter fits; dots mark shipped programs. **(b)** StarCraft II held-out wins against wall hours (complete-exam versions only): Fable 5 by tier (thin: sessions; thick: tier means; n counts sessions) with Astra's Elite and Cheater:Insane sessions drawn separately: a plateau where the tier is within reach, none at its edge. **(c)** Shipped versus best checkpoint, in exam utility: the apparent gap collapses under cross-fitting.

revising or rejecting its own explanation (26% in both full re-reads), the signature of investigation rather than storytelling; a second reader blinded to session and outcome reproduced nine of ten sampled status labels. In the audited roguelike sessions, which every model plays, frontier models build instruments while weaker models tune constants and re-run; the big games admit only the strongest models (Appendix F).

At its best the work is recognizable research, drawing on its two raw materials, outcome numbers and observed behavior; one audited session shows each used well. In StarCraft II the alarm was a number: a win that consumed 46,824 of 48,000 allowed loops, nearly a timeout. Rather than tune parameters, the agent re-ran the scenario with instrumentation and found the cause: its army was "attacking" with an empty memory of enemy structures, wandering instead of hunting. It added a map-covering search lattice with parallel hunt squads and won the retest in 67% fewer loops. The search mode shipped, and the controller scored 14/16 on the hidden exam.

The Civilization exemplar starts from behavior instead of a number, and it ends the other way: a sound local investigation attached to a failing program. Five newly built settlers had each died on the AI turn after completion. The agent blamed enemy attacks and built escorts, twice, and both fixes failed in replay. A client log line then surfaced the real cause: the settlers were being *disbanded* for lack of food upkeep. The agent reversed itself explicitly ("this was not combat"), changed the build rule to require a food surplus, and verified in a fresh replay of the same seed that a new settler survived. As local investigation, little is missing. The shipped controller scored 2/16. Local rigor does not bound the exam. The starkest case is a GPT-5.6-sol session at easy 1v1 that spent roughly 190 hours and 50,000 tool calls, played some 300 policy versions and wrote 1,200 more candidate files, ran until stopped externally, and shipped a policy scoring 0/16 while its two cell-mates scored 12 and 16. Only the held-out exam tells such sessions apart.

5 What the investigations achieve

Curves are a property of the cell. Replaying every roguelike version measures every policy the agent wrote, independently of its own testing; trajectories are jagged mostly because policies really change between versions, so sessions are described as draws around a cell curve. Figure 3a shows the shape directly: the thick curves, each a cell average, rise steeply in the first quarter of a session and flatten, while the thin individual trajectories scatter around them. A simple two-parameter saturating fit captures the averages (dashed in Figure 3a; alternatives fit no better, Appendix H), its

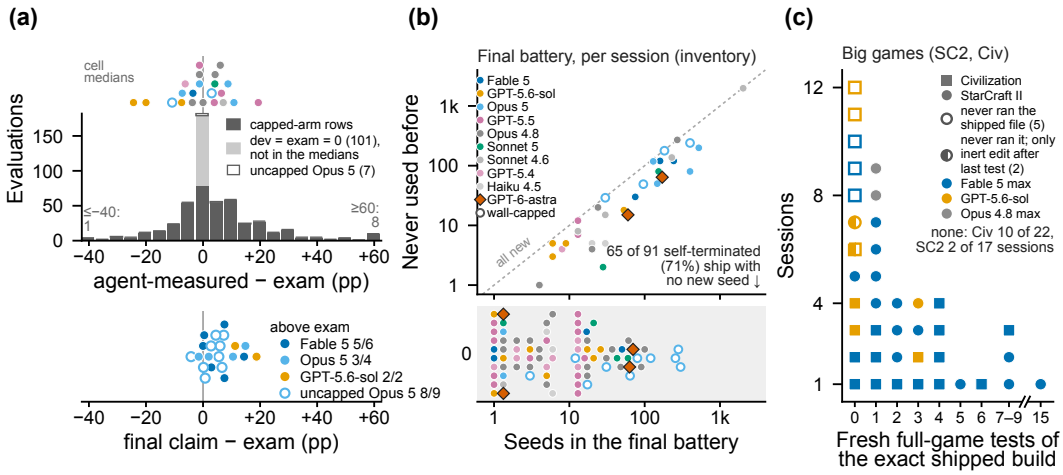


Figure 4: **The agent's evidence.** **(a)** Agent-measured minus hidden-exam score on the same policy (top) and final claims minus exam (bottom). **(b)** The final validation battery per session: battery size vs seeds never used before; shaded strip = zero new seeds. **(c)** Fresh full-length tests of the exact shipped build in the big games; of the 12 with none, 7 never ran the shipped file (Appendix J).

level recovering the drawn cell mean within 1.2 points where cells plateau. The plateaus are long: each frontier cell reaches nine tenths of its final level within the first quarter to two fifths of the session and holds it for the rest, and Astra compresses the whole shape (75/80 by its fifth to eighth version). StarCraft II shows the same rise where the tier is within reach: in Figure 3b the Very Hard mean (four sessions) climbs from 2 to 8 wins in three hours and ends near 11 of 16; Astra's Elite mean reaches 14.3 within the hour. At the edge of reach there is no plateau: Fable's Elite mean dips and recovers, one session falling from 12 wins to 0 before ending at 13, while Astra's Cheater:Insane sessions climb to a shipped mean of 11 of 16 where every Fable session stays at zero.

Late progress recycles old ground. A step's gains are the exam seeds that flip to cleared between consecutive versions; its losses, the seeds that flip to failed. Early in a session, 13–90% of gained seeds (by frontier cell) are first-time clears; by the final third that share is essentially zero in all 21 cells, so late gains recover seeds already won and lost. The per-seed loss hazard falls rather than rises: the plateau is damped churn over a shrinking contested pool, not growing interference; instrument sessions, run far past the normal horizon, decay to exhaustion (Appendix I).

Shipping loses little to checkpoint selection. The worry that a better version was left behind is mostly a same-seed artifact: picking the "best" checkpoint on the seeds that then score it overstates the gap. Figure 3c estimates what a 40-seed selector could actually recover, choosing on one seed half and scoring on the other: the median gain over shipping falls from $+0.05$ naive to $+0.01$ cross-fit in exam utility, with a real tail (36 of 100 sessions gain $+0.02$ or more; this measures that selector, not an oracle). On the 16-game StarCraft II exam, where "best earlier" is itself partly a draw of finite-sample noise, shipped matches or beats it in 5 of 11 Fable and 2 of 3 Astra sessions; exceptions are real (one Astra session held a 16/16 at 0.2 hours and shipped 12/16).

Measurement shows no consistent bias; final claims are optimistic. On the 527 policies agents chose to measure, in-development numbers carry no consistent optimistic bias across cells. The top panel of Figure 4a is centered on zero, with cell medians on both sides: ten above, ten below, -24 to $+19$ points, tracking the hardness of each cell's own test seeds relative to the exam (small panels add scatter). The strip beneath it sits to the right: final self-claims exceed the exam in 10 of 12 capped-cell claims and 8 of 9 uncapped claims (21 claims from 18 sessions), and exceed the cell's own measured median in all six cells with claims. The final battery, the last evaluation of the exact shipped file, is where the evidence is weakest: 65 of 91 self-terminated roguelike sessions ship without a single seed they had never tried (Figure 4b), and battery size is a model habit (medians 6–138 by model), not a response to uncertainty. Astra is the exception on size (60–171-seed batteries in four sessions) but not on freshness; twice it edited the file after the battery and shipped on a single-seed check.

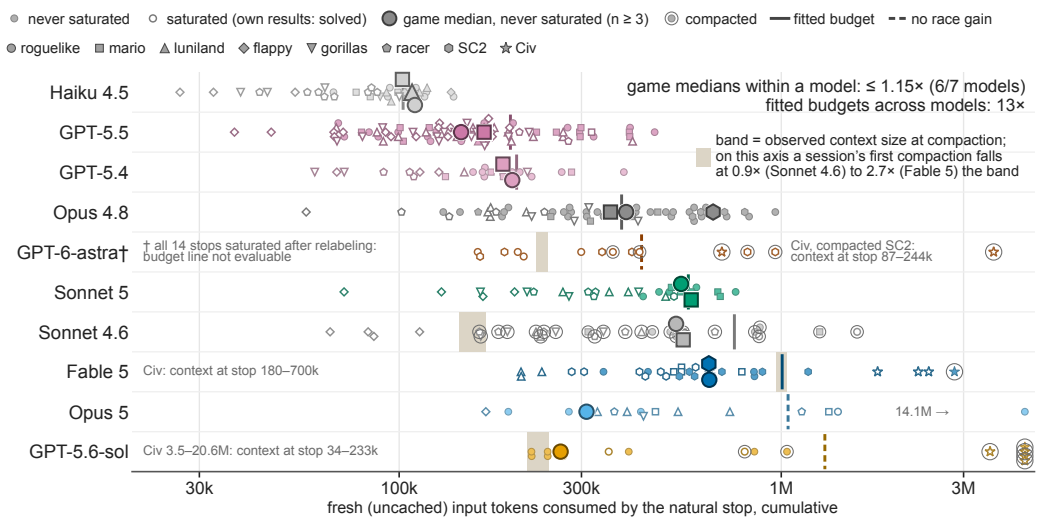


Figure 5: **Where sessions stop.** Natural stops per model across games, in cumulative fresh input tokens: filled = never saturated, hollow = saturated, large = per-game medians; line = fitted transcript budget (dashed: no race gain); band = context at observed compactions (absent: model never compacts); rings = compacted (past-band, ring-free stops = cache re-reads; †Astra: all 14 stops saturated, budget not evaluable). Never-saturated medians align across games within a model; budgets span $13\times$ across models; saturated stops fall short by a game-dependent wrap-up.

The big games show the same staleness where evidence is dearest: one full trial costs minutes to hours, and agents ship on thin direct evidence (Figure 4c). The median number of fresh full-horizon tests of the exact shipped build is 1.5 per session for Fable on Civilization and 0 for GPT-5.6-sol there (5 of 16 and 5 of 6 sessions ran none). On StarCraft II Fable's median is 2. How much an agent validates does not predict its exam score across cells; which instances it validates on does (§7).

6 How sessions end

Stopping follows a race between evidence and an uninstructed, model-specific scale. No session is told a budget or a deadline (external wall caps exist but are never announced; clock fragments leak into 95 of 101 audited sessions, and exactly one agent ever reasoned about absolute time, not about stopping). Call a session *saturated* once the agent has seen a result showing the task solved (roguelike: one of its own batteries $\geq$95% cleared; StarCraft II: consecutive full-game wins; Appendix K). Sessions that never saturate (91% of roguelike sessions, most harder StarCraft II tiers) stop at a model-specific scale of consumed transcript; we call the fitted scale an *effective transcript budget*, a predictive description rather than an identified internal mechanism. Fitted budgets run from 102k fresh input tokens (Haiku 4.5) to about 1.3M (GPT-5.6-sol), a $13\times$ span (Figure 5). The regularity is an alignment of medians: within each row of Figure 5 the per-game median markers align (within $1.15\times$ for six of seven multi-game models) while sessions still spread two- to three-fold inside a cell. Model identity sets the scale of stopping; the game barely moves it; nothing pins a single session. Saturated sessions stop earlier, after a wrap-up of continued polishing. That wrap-up is failure-gated, not threshold-timed: sessions almost never end on a failing result and polish longer where the game leaves more to gain (Appendix K). A race model, stopping at whichever comes first of saturation-plus-wrap-up and the budget, beats the budget alone out of sample in 7 of 10 fitted models (7 of 9 evaluable): the budget carries the never-saturated stops, and the evidence term adds the saturated ones (a registered hazard analysis agrees, and no visible progress signal adds more; Appendix K).

The budget is a scale of consumed transcript, not of context: cache re-reads bill old context again, so cumulative counts pass triggers that live context never reaches. The triggers are directly observable ($\approx$244k context under codex; 143–168k for Claude Code with older Claude models; Appendix K),

most sessions stop with live context below them, and only 38 of 330 drawn stops ever compact, 24 of them Sonnet 4.6's (Appendix K). Crossing the trigger splits behavior (rings in Figure 5): Sonnet 4.6 compacts routinely yet still stops at a stable cumulative scale, while GPT-5.6-sol's Civilization sessions cross it dozens of times and become long-lived, ending at up to sixteen times the budget.

What happens at the stop is a confirmation routine, not a threshold. In the nine models other than Astra, finalization language precedes the stop more often than it precedes other evaluations, and in seven of nine the last evaluation re-runs an unchanged policy, mostly on seeds already used: a fresh measurement of nothing new. Whether a failing final check delays the stop is model-specific (Appendix K); Astra is the strict case, never once stopping after a failing battery: every failure led to an edit and retest (19/19 final games won). Astra is also the model whose stops the evidence explains outright: with its background-shell batteries recovered from the traces, all 14 of its natural stops are saturated. It solves the task quickly, confirms, and leaves; its budget is unidentifiable: it never runs long enough to reveal one. For every other model the summary is blunt: how long an agent works is predicted mostly by which model it is, and the plateau of Figure 3 is held with no visible measurement of arrival; only at the edge of reach does the long dwell keep buying wins.

7 Changing the evidence

Changing only the evidence changes the product. If the evidence an agent gathers is part of what it ships, refreshing the evidence should change the product; the panel-refresh intervention tests exactly that (Figure 6a). The harness pauses a session at a preset point; two branches resume from an identical message and differ only in which validation panel the harness serves: the one the parent had been reusing, or a fresh panel from the same distribution. The arm is invisible to the agent, branches run to their own end under equal caps, and both shipped programs take the same hidden exam.

In Fable 5, whose forks had headroom (mean 37.6/80), refreshed validation won all nine completed pairs of twelve initiated (Figure 6b), mean $+12.4$ points (95% CI $+4.6$ to $+20.1$; attempt ledger in Appendix L). The replication in Opus 5 ran all twelve pairs and returned a null on the pre-registered R2 endpoint ($+0.0102$, t=0.83; Figure 6c). Its forks sat at the R1 ceiling (median 80/80); descriptively, the two pairs forked well below that ceiling (64 and 68 of 80) moved most on R2, one down and one up (-0.028, $+0.139$). Sonnet 5's eight pairs bracket zero (CI -8.5 to $+2.5$ points): they forked at 1–37 of 80, the model's own level, and validator uptake was low (three branches per arm never called it; Appendix L). The supported claim is narrow: where branches had both headroom and uptake, refreshing what the final checks see improved what shipped; ceiling and uptake remain candidate explanations for the other strata (Appendix L).

8 Related work

Agents doing research. The closest lane studies agents as autonomous researchers of machine learning itself (Lu et al., 2024; Hambardzumyan et al., 2026; Askarbekuly et al., 2026; Huang et al., 2024; Chan et al., 2025; Nathani et al., 2025; Starace et al., 2025). We move the object of research from open-ended ML to closed instrumented worlds where frozen artifacts are re-scored on hidden exams and the agent's evidence is checked line by line.

Agents building artifacts. Harness- and artifact-development benchmarks score what agents build (Wu et al., 2026; Zheng et al., 2026; Ursekar et al., 2026; Wang et al., 2026a; Lu et al., 2026; Chi et al., 2026), and game-playing work scores how models play: directly (Paglieri et al., 2025; Hu et al., 2026; Ying et al., 2026; Guertler et al., 2025; Küttler et al., 2020), through scaffolds (Wang et al., 2024a; Tan et al., 2025b), and in full strategy games (Vinyals et al., 2019; Sun et al., 2018; Ma et al., 2024; Qi et al., 2024; Chen et al., 2025; Meta Fundamental AI Research Diplomacy Team (FAIR) et al., 2022); Hu et al. (2024) survey; full sweep in Appendix N. Gauntlet sits between: the deliverable is a program that plays model-free, as in programmatic policy work (Verma et al., 2018; Xu et al., 2025; Deng et al., 2024; Kuang et al., 2025) and evolutionary program search (Romera-Paredes et al., 2024; Novikov et al., 2025), but the object measured is the session that produced it. The nearest neighbors also study the building process: EvoPolicyGym evolves complete Python policies under a fixed 128-episode budget with platform-served feedback and best-validation checkpointing (Wang et al., 2026b); CodeClash scores agent-built codebases over fixed tournament rounds (Yang et al., 2026); chess-engine builders show optimistic self-assessment against independent Elo (Acher & Jézéquel,

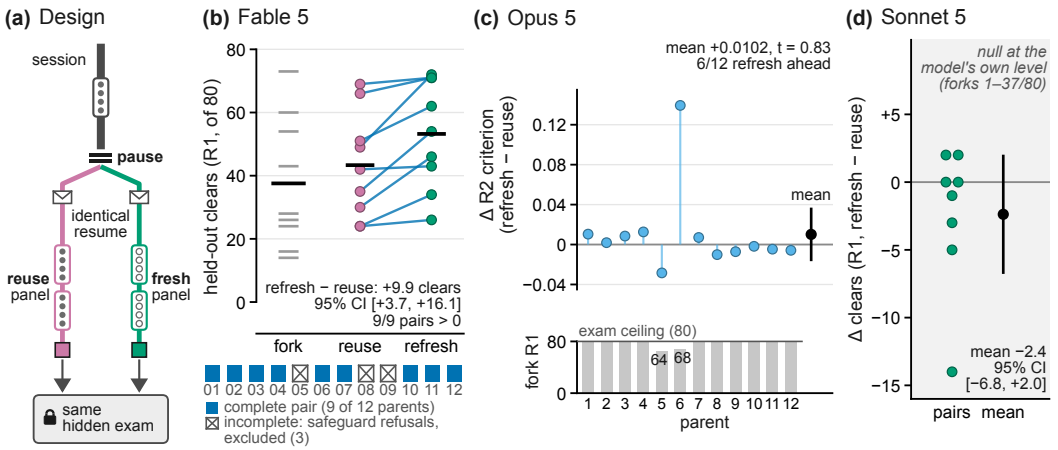


Figure 6: **The panel-refresh intervention.** **(a)** Design: a session pauses at a preset point; two branches resume identically and differ only in the validation panel the harness serves, reused or fresh; both shipped programs take the same hidden exam (conditions in text). **(b)** Fable 5: refresh wins 9/9 completed pairs. **(c)** Opus 5: 12/12 pairs, null on the R2 utility (forks at the R1 ceiling); fork clears in the strip below. **(d)** Sonnet 5: null at the model's own level. Panels (b) and (d) print clears; 1 clear = 1.25 points.

2026); GPT-5.4 Connect-4 builders under-use their allotted time (Sherwood et al., 2026). Gauntlet adds uninstructed sessions in which the agent supplies its own validation evidence, independent replay of frozen versions on hidden exams, and an intervention on that evidence itself.

Reusing evidence and extrapolating curves. Our validation findings instantiate, in the wild, what the adaptive-data-analysis literature predicts for reused holdouts (Dwork et al., 2015; Blum & Hardt, 2015; Cawley & Talbot, 2010) and what agent studies report as optimistic self-evaluation (Park & Choi, 2026); our curve and forecasting analyses adapt learning-curve extrapolation (Domhan et al., 2015; Swersky et al., 2014; Rakotoarison et al., 2024) and test-time-scaling curves (Liu et al., 2026) to within-session development. Agent stopping has been studied as premature abandonment or runaway looping (Chen et al., 2026c; Hou et al., 2026; Sogani et al., 2026); the transcript-budget regularity and its race with evidence appear to be new. Normative stopping rules, marginal value (Charnov, 1976) and optimal search (Weitzman, 1979), supply the background these measurements can now be set against. Compaction's behavioral effects have been noted (Tamba, 2026); we measure the trigger and what crossing it does. (GauntletBench, a GUI-task benchmark, is unrelated (Vysotskyi et al., 2026).)

9 Discussion and Limitations

Read as a researcher, the coding agent has recognizable pathologies: it measures honestly, then writes an optimistic abstract; it validates on instances it already knows; its final battery is a ritual of confirmation; and it stops when its transcript budget runs out far more often than when the evidence says done. Each pathology suggests an instrumented remedy; one, serving fresh validation evidence, is shown here to change what ships. The effective transcript budget should interest harness designers: model-specific, stable across the games we can test, and uninstructed, it predicts the end of hard sessions better than any visible evidence signal; whether it can be steered is testable. Compiled agency is what these sessions deliver; the experiments the agent designs are part of it, and Gauntlet measures that part.

The limits: exam ceilings compress the top of the leaderboard, and one replication's null came with forks at that ceiling; the open questions move to Cheater:Insane and Civilization. Cells are small (N=1–10; frontier 6); every number is reported per cell. The race model is post hoc (the registered test is the hazard analysis). Agent-view recovery is 86%, and frontier reasoning is mostly redacted: trace audits read behavior, not thought. One intervention in one game and one model with headroom is evidence of a mechanism, not a law; seed reuse stales evidence only because exams are deterministic. Finally, a harness contribution to cross-vendor differences cannot be excluded (§2).

Reproducibility statement

Every number in this paper is produced by a script in the released corpus reading frozen artifacts; the single-file exams are deterministic (identical artifacts receive identical per-seed outcomes), and StarCraft II replays are reported with per-build agreement; reconstructions are checked by byte-identity against shipped artifacts and workspace snapshots; and the confirmatory analyses (§5, §6) follow a spec registered before the data were opened, with deviations listed in the appendix. The analysis code, audit ledgers, and replay tooling are included, anonymized, in the supplementary material; the full benchmark and corpus release accompanies the camera-ready.

Ethics statement

The work evaluates commercial model behavior on synthetic game tasks; no human subjects, personal data, or deployment risks are involved. Sessions that were externally terminated (provider limits) are retained and labeled; they are excluded from natural-stop fits and right-censored in the hazard analysis, not discarded.

Use of AI assistance

Coding agents are the object of study; the corpus is their output. Separately, AI tools were used to polish writing, to search related work, and to assist in writing and executing analysis code and managing the repository, under author direction; all analyses, claims, and text were reviewed and are the authors' responsibility.

References

Mathieu Acher and Jean-Marc Jézéquel. Do programming languages still matter to your AI coding agent teammate? evidence at scale from chess engines, 2026. URL https://arxiv.org/abs/2606.13763.

Austin Tudor David Andrews, Liam Wilkinson, Jamie Heagerty, Harry Coppock, Jakob Nicolaus Foerster, and Rui Ponte Costa. CivBench: A long-horizon benchmark for tool-mediated agents in Civilization VI, 2026. URL https://arxiv.org/abs/2609.02459.

ARC Prize Foundation. ARC-AGI-3: A new challenge for frontier agentic intelligence, 2026. URL https://arxiv.org/abs/2603.24621.

Nursultan Askarbekuly, Mohamad Al Mdfaa, Ahmed Helaly, Gonzalo Ferrer, and Manuel Mazzara. Autoresearch with coding agents: Generalizers and metric-maximizers on Quran recitation data, 2026. URL https://arxiv.org/abs/2607.18064.

Suma Bailis, Jane Friedhoff, and Feiyang Chen. Werewolf Arena: A case study in LLM evaluation via social deduction, 2024. URL https://arxiv.org/abs/2407.13943.

Blizzard Entertainment. s2client-proto: StarCraft II client protocol definitions. https://github.com/Blizzard/s2client-proto, 2017. Software, MIT licence.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1006–1014, 2015. URL https://proceedings.mlr.press/v37/blum15.html.

BurnySc2 and contributors. python-sc2: A StarCraft II bot API client library for Python 3. https://github.com/BurnySc2/python-sc2, 2019. Software, MIT licence.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=6s5uXNWGIh. Oral.

Eric L. Charnov. Optimal foraging, the marginal value theorem. *Theoretical Population Biology*, 9 (2):129–136, 1976. doi: 10.1016/0040-5809(76)90040-X.

John Chen, Sihan Cheng, Can Gurkan, Ryan Lay, and Moez Salahuddin. Vox Deorum: A hybrid LLM architecture for 4X / grand strategy game AI – lessons from Civilization V, 2025. URL https://arxiv.org/abs/2512.18564.

John Chen, Sihan Cheng, Can Gurkan, and Mingyi Lin. CivBench: Progress-based evaluation for LLMs' strategic decision-making in Civilization V. In *Conference on Language Modeling (COLM)*, 2026a. URL https://openreview.net/forum?id=8eM14jMVDh.

Kun Chen, Haorong Hong, Peizhong Gao, Jianfeng Lin, Tongxu Luo, Yuxuan Xie, Chenxu Liu, Jieling He, Zhongyuan Liu, and Zeno Zeng. GameXpert-Bench: How far are coding agents from expert game development?, 2026b. URL https://arxiv.org/abs/2608.21833.

Ying Chen, Lihuang Fang, Rui Jiang, Mingxu Wang, Zhifeng Gu, Lei Yi, and Jie Chen. Done, but not sure: Disentangling world completion from self-termination in embodied agents, 2026c. URL https://arxiv.org/abs/2605.08747.

Wayne Chi, Yixiong Fang, Arnav Yayavaram, Siddharth Yayavaram, Seth Karten, Qiuhong Anna Wei, Runkun Chen, Alexander Wang, Valerie Chen, Ameet Talwalkar, and Chris Donahue. GameDevBench: Evaluating agentic capabilities through game development. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=Iexhb5lL3t.

Lucia Cipolina-Kun et al. Game reasoning arena: A framework and benchmark for assessing reasoning capabilities of large language models via game play, 2025. URL https://arxiv.org/abs/2508.03368.

Nathan Cloos, Meagan Jens, Michelangelo Naim, Yen-Ling Kuo, Ignacio Cases, Andrei Barbu, and Christopher J. Cueva. Baba Is AI: Break the rules to beat the benchmark. In *ICML Workshop on Large Language Models and Cognition*, 2024. URL https://arxiv.org/abs/2407.13729.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. GameBench: Evaluating strategic reasoning abilities of LLM agents. In *NeurIPS Workshop on Language Gamification*, 2024. URL https://neurips.cc/virtual/2024/106557.

Christopher Zhang Cui et al. TALES: Text adventure learning environment suite, 2025. URL https://arxiv.org/abs/2504.14128.

Yue Deng, Weiyu Ma, Yuxin Fan, Ruyi Song, Yin Zhang, Haifeng Zhang, and Jian Zhao. SMAC-R1: The emergence of intelligence in decision-making tasks, 2024. URL https://arxiv.org/abs/2410.16024v3.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3460–3468, 2015.

Jinhao Duan et al. GTBench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pp. 28219–28253, 2024. doi: 10.52202/079017-0885.

Alexander Duffy, Samuel J. Paech, Ishana Shastri, Elizabeth Karpinski, Baptiste Alloui-Cros, Tyler Marques, and Matthew Lyle Olson. Democratizing Diplomacy: A harness for evaluating any large language model on full-press Diplomacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i44.41067.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015. doi: 10.1126/science.aaa9375.

Manuel Eberhardinger, James Goodman, Alexander Dockhorn, Diego Perez-Liebana, Raluca D. Gaina, Duygu Çakmak, Setareh Maghsudi, and Simon Lucas. From code to play: Benchmarking program search for games using large language models. *IEEE Transactions on Games*, 2025. doi: 10.1109/TG.2025.3614499. URL https://doi.org/10.1109/TG.2025.3614499.

Linxi Fan et al. MineDojo: Building open-ended embodied agents with internet-scale knowledge. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/74a67268c5cc5910f64938cac4526a90-Abstract-Datasets_and_Benchmarks.html.

Google DeepMind and Kaggle. Kaggle Game Arena. https://www.kaggle.com/game-arena, 2025.

Leon Guertler et al. TextArena, 2025. URL https://arxiv.org/abs/2504.11442.

Maxim Gumin, Hsueh-Ti Derek Liu, Victor Zordan, and Daniel Ritchie. Synthesizing reactive character behaviors for continuous games via programmatic policy search, 2026. URL https://arxiv.org/abs/2609.24025.

Karen Hambardzumyan, Nicolas Baldwin, Edan Toledo, Rishi Hazra, Michael Kuchnik, Bassel Al Omari, Thomas Simon Foster, Anton Protopopov, Jean-Christophe Gagnon-Audet, Ishita Mediratta, Kelvin Niu, Michael Shvartsman, Alisia Lupidi, Alexis Audran-Reiss, Parth Pathak, Tatiana Shavrina, Despoina Magka, Hela Momand, Derek Dunfield, Nicola Cancedda, Pontus Stenetorp, Carole-Jean Wu, Jakob Nicolaus Foerster, Yoram Bachrach, and Martin Josifoski. AIRA_2: Overcoming bottlenecks in AI research agents, 2026. URL https://arxiv.org/abs/2603.26499.

Daniel Hennes, Zun Li, John Schultz, and Marc Lanctot. Code-space response oracles: Generating interpretable multi-agent policies with large language models. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Extended Abstract*, 2026. doi: 10.65109/ikjf6607.

Xingxing Hong, Yungong Wang, Dexin Jin, Ye Yuan, Ximing Huang, Zijian Wu, and Wenxin Li. HLSMAC: A new StarCraft multi-agent challenge for high-level strategic decision-making, 2025. URL https://arxiv.org/abs/2509.12927.

Jack Hopkins, Mart Bakler, and Akbir Khan. Factorio learning environment. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 38, pp. 91153–91206, 2025. doi: 10.52202/085713-2745. URL https://openreview.net/forum?id=652Q6jBFMZ.

Xinyi Hou et al. When agents do not stop: Uncovering infinite agentic loops in LLM agents, 2026. URL https://arxiv.org/abs/2607.01641.

Lanxiang Hu, Qiyu Li, Anze Xie, Nan Jiang, Ion Stoica, Haojian Jin, and Hao Zhang. GameArena: Evaluating LLM reasoning through live computer games. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=SeQ8l8xo1r.

Lanxiang Hu, Mingjia Huo, Yuxuan Zhang, Haoyang Yu, Eric P. Xing, Ion Stoica, Tajana Rosing, Haojian Jin, and Hao Zhang. lmgame-Bench: How good are LLMs at playing games? In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=qeziG97WUZ.

Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. A survey on large language model-based game agents, 2024. URL https://arxiv.org/abs/2404.02039v5.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309, 2024. URL https://proceedings.mlr.press/v235/huang24y.html.

Dominik Jeurissen, Diego Perez-Liebana, Jeremy Gow, Duygu Çakmak, and James Kwan. Playing NetHack with LLMs: Potential & limitations as zero-shot agents. In *IEEE Conference on Games (CoG)*, pp. 1–8, 2024. doi: 10.1109/CoG60054.2024.10645630.

Yilei Jiang et al. OpenGame: Open agentic coding for games, 2026. URL https://arxiv.org/abs/2604.18394.

Seth Karten, Andy Luu Nguyen, and Chi Jin. PokéChamp: an expert-level minimax language agent. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29205–29222, 2025. URL https://proceedings.mlr.press/v267/karten25a.html.

Seth Karten et al. The PokeAgent challenge: Competitive and long-context learning at scale, 2026. URL https://arxiv.org/abs/2603.15563.

Zhiyi Kuang, Ryan Rong, YuCheng Yuan, and Allen Nie. Learning game-playing agents with generative code optimization. In *ICML 2025 Workshop on Programmatic Representations for Agent Learning (PRAL)*, 2025. URL https://openreview.net/forum?id=ZM65X3NoTd.

Akarsh Kumar, Ryan Bahlous-Boldi, Prafull Sharma, Phillip Isola, Sebastian Risi, Yujin Tang, and David Ha. Digital Red Queen: Adversarial program evolution in Core War with LLMs, 2026. URL https://arxiv.org/abs/2601.03335.

Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack learning environment. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 7671–7684, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/569ff987c643b4bedf504efda8f786c2-Abstract.html.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 102382–102435, 2025. doi: 10.52202/085713-3086. URL https://openreview.net/forum?id=CGNJL6CeV0.

Wolfgang Lehrach, Daniel Hennes, Miguel Lázaro-Gredilla, Xinghua Lou, Carter Wendelken, Zun Li, Antoine Dedieu, Jordi Grau-Moya, Marc Lanctot, Atil Iscen, John Schultz, Marcus Chiam, Ian Gemp, Piotr Zielinski, Satinder Singh, and Kevin P. Murphy. Code world models for general game playing. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=1UoB7IWiku.

Lingfeng Li, Yunlong Lu, Yuefei Zhang, Jingyu Yao, Yixin Zhu, KeYuan Cheng, Yongyi Wang, Qirui Zheng, Xionghui Yang, and Wenxin Li. BotzoneBench: Scalable LLM evaluation via graded AI anchors, 2026. URL https://arxiv.org/abs/2602.13214.

Yuchen Li, Cong Lin, Muhammad Umair Nasir, Philip Bontrager, Jialin Liu, and Julian Togelius. GVGAI-LLM: Evaluating large language model agents with infinite games, 2025a. URL https://arxiv.org/abs/2508.08501.

Zongyuan Li, Chang Lu, Xiaojie Xu, Runnan Qi, Yanan Ni, Lumin Jiang, Xiangbei Liu, Xuebo Zhang, Yongchun Fang, Kuihua Huang, and Xian Guo. Hierarchical expert prompt for large-language-model: An approach defeat elite AI in TextStarCraft II for the first time, 2025b. URL https://arxiv.org/abs/2502.11122.

Zongyuan Li, Yanan Ni, Runnan Qi, Chang Lu, Lumin Jiang, Xiaojie Xu, Xiangbei Liu, Pengfei Li, Yunzheng Guo, Zhe Ma, Huanyu Li, Hui Wu, Xian Guo, Kuihua Huang, and Xuebo Zhang. LLM-PySC2: StarCraft II learning environment for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 141492–141554, 2025c. doi: 10.52202/085713-4250. URL https://openreview.net/forum?id=Xr73jEYG29.

Mingxian Lin, Shengju Qian, Yuqi Liu, Yi-Hua Huang, Yiyu Wang, Wei Huang, Yitang Li, Fan Zhang, Zeyu Hu, Lingting Zhu, Xin Wang, and Xiaojuan Qi. OmniGameArena: A unified UE5 benchmark for VLM game agents with improvement dynamics, 2026. URL https://arxiv.org/abs/2606.09826.

Wenye Lin et al. GAMEBoT: Transparent assessment of LLM reasoning in games. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, 2025. URL https://aclanthology.org/2025.acl-long.378/.

Kaiyuan Liu, Qiuyang Mang, Bo Peng, Wenhao Chai, Hanchen Li, Shreyas Pimpalgaonkar, Luke Zettlemoyer, Alex Dimakis, and Alvin Cheung. When agents slow down: Understanding LLM agents' test-time strategies via Elo-per-token analysis, 2026. URL https://arxiv.org/abs/2609.15309.

Max Liu, Chan-Hung Yu, Wei-Hsu Lee, Cheng-Wei Hung, Yen-Chun Chen, and Shao-Hua Sun. Synthesizing programmatic reinforcement learning policies with large language model guided search. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://openreview.net/forum?id=8DBTq09LgN.

Shunyu Liu et al. Odyssey: Empowering Minecraft agents with open-world skills. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2025b. doi: 10.24963/ijcai.2025/22.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. AutoHarness: improving LLM agents by automatically synthesizing a code harness. In *ICLR 2026 Workshop on Recursive Self-Improvement*, 2026. URL https://openreview.net/forum?id=g9rEYVNn5T.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

Xinyu Lu, Tianshu Wang, Pengbo Wang, Zujie Wen, Zhiqiang Zhang, Jun Zhou, Boxi Cao, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. The meta-agent challenge: Are current agents capable of autonomous agent development?, 2026. URL https://arxiv.org/abs/2606.04455.

Tongxu Luo et al. GameCraft-Bench: Can agents build playable games end-to-end in a real game engine?, 2026. URL https://arxiv.org/abs/2606.17861.

Mingrui Lv, Hangzhi Liu, Zhi Luo, Hongjie Zhang, and Jie Ou. PolicyEvolve: Evolving programmatic policies by LLMs for multi-player games via population-based training, 2025. URL https://arxiv.org/abs/2509.06053.

Weiyu Ma, Qirui Mi, Yongcheng Zeng, Xue Yan, Yuqiao Wu, Runji Lin, Haifeng Zhang, and Jun Wang. Large language models play StarCraft II: Benchmarks and a chain of summarization approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 133386–133442, 2024. doi: 10.52202/079017-4240. URL https://openreview.net/forum?id=kEPpD7yETM.

Weiyu Ma, Dongyu Xu, Shu Lin, Haifeng Zhang, and Jun Wang. Adaptive command: Real-time policy adjustment via language models in StarCraft II, 2025. URL https://arxiv.org/abs/2508.16580.

Weiyu Ma, Yuqian Fu, Zecheng Zhang, Bernard Ghanem, and Guohao Li. AVA: Attentive VLM agent for mastering StarCraft II. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. doi: 10.18653/v1/2026.findings-acl.208.

Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Thomas Jackson, Samuel Coward, and Jakob Nicolaus Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 2024. URL https://proceedings.mlr.press/v235/matthews24a.html.

Meta Fundamental AI Research Diplomacy Team (FAIR), Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022. doi: 10.1126/science.ade9097. URL https://www.science.org/doi/10.1126/science.ade9097.

Connacher Murphy. Agent Island: A saturation- and contamination-resistant benchmark from multiagent games, 2026. URL https://arxiv.org/abs/2605.04312.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Mikhail Plekhanov, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Nicolaus Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A new framework and benchmark for advancing AI research agents. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=ryTr83DxRq.

Alexander Novikov, Ngân 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, 2025. URL https://arxiv.org/abs/2506.13131.

Mingyu Ouyang, Siyuan Hu, Kevin Qinghong Lin, Hwee Tou Ng, and Mike Zheng Shou. GameWorld: Towards standardized and verifiable evaluation of multimodal game agents. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 41–59, 2026. doi: 10.1007/978-3-032-37255-0_3.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=fp6t3F669F.

Dongmin Park, Minkyu Kim, Beongjun Choi, Junhyuck Kim, Keon Lee, Jonghyun Lee, Inkyu Park, Byeong-Uk Lee, Jaeyoung Hwang, Jaewoo Ahn, Ameya S. Mahabaleshwarkar, Bilal Kartal, Pritam Biswas, Yoshi Suhara, Kangwook Lee, and Jaewoong Cho. Orak: A foundational benchmark for training and evaluating LLM agents on diverse video games. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=H1ncX6O6Yh.

Hyundoo Park and Byungho Choi. When do agent loops mistake stagnation for progress? self-evaluation bias and externally grounded verification in long-running autonomous LLM agent loops, 2026. Exploratory pilot measurement study; Zenodo concept DOI 10.5281/zenodo.21594735.

Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, Nian Liu, Yaodong Yang, and Song-Chun Zhu. CivRealm: A learning and reasoning odyssey in Civilization for decision-making agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=UBVNwD3hPN. Spotlight.

Herilalaina Rakotoarison, Steven Adriaensen, Neeratyoy Mallik, Samir Garibov, Eddie Bergman, and Frank Hutter. In-context freeze-thaw Bayesian optimization for hyperparameter optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41982–42008, 2024.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pp. 2186–2188, 2019. URL https://arxiv.org/abs/1902.04043. Extended abstract.

Xiao Shao, Weifu Jiang, Fei Zuo, and Mengqing Liu. SwarmBrain: Embodied agent for real-time strategy game StarCraft II via large language models, 2024. URL https://arxiv.org/abs/2401.17749.

Pengbo Shen, Yaqing Wang, Ni Mu, Yao Luan, Runpeng Xie, Senhao Yang, Lexiang Wang, Hao Hu, Shuang Xu, Yiqin Yang, and Bo Xu. SC2Arena and StarEvolve: Benchmark and self-improvement framework for LLMs in complex decision-making tasks, 2025. URL https://arxiv.org/abs/2508.10428.

Joshua Sherwood, Ben Aybar, and Benjamin Kaplan. Frontier coding agents can now implement an AlphaZero self-play machine learning pipeline for Connect Four that performs comparably to an external solver, 2026. URL https://arxiv.org/abs/2604.25067.

SIMA Team. SIMA 2: A generalist embodied agent for virtual worlds, 2025. URL https://arxiv.org/abs/2512.04797.

Aagam Sogani, Botao Rui, Swetha Vaidyanathan, Rishi Agarwal, Minghao Yan, and Shivaram Venkataraman. When web agents finish but still fail: Reproducible triggers and trace diagnostics for parallel web exploration, 2026. URL https://arxiv.org/abs/2606.20724. FAGEN Workshop at ICML 2026 (poster).

Joel Sol and Homayoun Najjaran. SMAC-Talk: A natural language extension of the StarCraft multi-agent challenge for large language models, 2026. URL https://arxiv.org/abs/2606.04202.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Peng Sun, Xinghai Sun, Lei Han, Jiechao Xiong, Qing Wang, Bo Li, Yang Zheng, Ji Liu, Yongsheng Liu, Han Liu, and Tong Zhang. TStarBots: Defeating the cheating level builtin AI in StarCraft II in the full game, 2018. URL https://arxiv.org/abs/1809.07193.

Kevin Swersky, Jasper Snoek, and Ryan P. Adams. Freeze-thaw Bayesian optimization, 2014. URL https://arxiv.org/abs/1406.3896.

Hiroki Tamba. Compaction as epistemic failure: How agentic LLM tools fabricate confirmed results from killed processes, 2026. URL https://arxiv.org/abs/2607.13071.

Weihao Tan, Xiangyang Li, Yunhao Fang, Heyuan Yao, Shi Yan, Hao Luo, Tenglong Ao, Huihui Li, Hongbin Ren, Bairen Yi, Yujia Qin, Bo An, Libin Liu, and Guang Shi. Lumine: An open recipe for building generalist agents in 3D open worlds, 2025a. URL https://arxiv.org/abs/2511.08892.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, Ruyi An, Molei Qin, Chuqiao Zong, Longtao Zheng, Yujie Wu, Xiaoqiang Chai, Yifei Bi, Tianbao Xie, Pengjie Gu, Xiyun Li, Ceyao Zhang, Long Tian, Chaojie Wang, Xinrun Wang, Börje F. Karlsson, Bo An, Shuicheng Yan, and Zongqing Lu. Cradle: Empowering foundation agents towards general computer control. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58658–58725, 2025b. URL https://proceedings.mlr.press/v267/tan25h.html.

Wenjie Tang, Yuan Zhou, Keyan Cheng, Erqiang Xu, Liquan Xiao, and Minne Li. DSGBench: A diverse strategic game benchmark for evaluating LLM-based agents in complex decision-making environments. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16987–16991, 2026. doi: 10.1109/ICASSP55912.2026.11460704.

Varun Ursekar, Apaar Shanker, Yash Maurya, Shehab Yasser, Vijay S. Kalmath, Veronica Chatrath, and Yuan Xue. HarnessOpt-Bench: Evaluating LLMs at harness optimization, 2026. URL https://arxiv.org/abs/2608.06301.

Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5045–5054, 2018.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. StarCraft II: A new challenge for reinforcement learning, 2017. URL https://arxiv.org/abs/1708.04782.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z. URL https://doi.org/10.1038/s41586-019-1724-z.

Mykola Vysotskyi, Runqi Lin, Grzegorz Biziel, Michal Zakrzewski, Sebastian Montagna, Damian Rynczak, Shreyansh Padarha, Kumail Alhamoud, Zihao Fu, William Lugoloobi, Kai Rawal, Hanna Yershova, Taras Rumezhak, Guohao Li, Fazl Barez, Baoyuan Wu, Arkadiusz Drohomirecki, Yarin Gal, Chris Russell, Christopher Summerfield, Adam Mahdi, Volodymyr Karpiv, Philip Torr, and Adel Bibi. Running the Gauntlet: Challenging agentic tasks, 2026. URL https://arxiv.org/abs/2606.14397.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a. URL https://openreview.net/forum?id=ehfRiF0R3a.

Haochuan Wang, Xiachong Feng, Lei Li, Yu Guo, Zhanyue Qin, Dianbo Sui, and Lingpeng Kong. TMGBench: A systematic game benchmark for evaluating strategic reasoning abilities of LLMs, 2024b. URL https://arxiv.org/abs/2410.10479.

Jiawei Wang, Kai Wang, Shaojie Lin, Runze Wu, Bihan Xu, Lingeng Jiang, Shiwei Zhao, Renyu Zhu, Haoyu Liu, Zhipeng Hu, Zhong Fan, Le Li, Tangjie Lyu, and Changjie Fan. Digital player: Evaluating large language models based human-like agent in games, 2025a. URL https://arxiv.org/abs/2502.20807.

Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hannaneh Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. Rethinking the evaluation of harness evolution for agents, 2026a. URL https://arxiv.org/abs/2607.12227.

Zhilin Wang, Han Song, Runzhe Zhan, Jusen Du, Jiacheng Chen, Tianle Li, Qingyu Yin, Yulun Wu, Zhennan Shen, Tong Zhu, Yanshu Li, Guanjie Chen, Derek F. Wong, Yafu Li, Yu Cheng, and Yang Yang. EvoPolicyGym: Evaluating autonomous policy evolution in interactive environments, 2026b. URL https://arxiv.org/abs/2607.02440.

Zihao Wang, Xujing Li, Yining Ye, Junjie Fang, Haoming Wang, Longxiang Liu, Shihao Liang, Junting Lu, Zhiyong Wu, Jiazhan Feng, Wanjun Zhong, Zili Li, et al. Game-TARS: Pretrained foundation models for scalable generalist multimodal game agents, 2025b. URL https://arxiv.org/abs/2510.23691.

Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979. doi: 10.2307/1910412. URL https://doi.org/10.2307/1910412.

Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. SmartPlay: A benchmark for LLMs as intelligent agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=S2oTVrlcp3.

Yuhao Wu, Jingyuan Zhang, Jiajun Shi, Xinping Lei, Qingshui Gu, Yuxuan Zhang, Zexuan Wang, Chen He, Chen Huang, Maojia Song, Zhiyuan Zeng, Shaowen Wang, Jinkai Liu, Yunfeng Shi, Jiaheng Liu, Shen Yan, Wenhao Huang, Ge Zhang, and Wenxuan Zhang. HarnessDev: Can LLMs create and evolve their own agent harness?, 2026. URL https://arxiv.org/abs/2609.01437.

Zhongwen Xu, Xianliang Wang, Siyi Li, Tao Yu, Liang Wang, Qiang Fu, and Wei Yang. Agents play thousands of 3D video games, 2025. URL https://arxiv.org/abs/2503.13356.

John Yang, Kilian Lieret, Joyce Yang, Carlos E. Jimenez, Muhtasham Oblokulov, Aryan Siddiqui, Ofir Press, Ludwig Schmidt, and Diyi Yang. CodeClash: Benchmarking goal-oriented software engineering. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=SW2D390ePP.

Jianzhu Yao et al. SPIN-Bench: How well do LLMs plan strategically and reason socially? In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2503.12349.

Lance Ying, Ryan Truong, Prafull Sharma, Kaiya Ivy Zhao, Nathan Cloos, Kelsey R. Allen, Thomas L. Griffiths, Katherine M. Collins, José Hernández-Orallo, Phillip Isola, Samuel J. Gershman, and Joshua B. Tenenbaum. AI Gamestore: Scalable, open-ended evaluation of machine general intelligence with human games, 2026. URL https://arxiv.org/abs/2602.17594.

Baoqing Yue, Zihan Zhu, Yutong Han, Brian Fan, Qian Sun, Jichen Feng, Hufei Yang, Yifan Zhang, and Mengdi Wang. Interactive benchmarks, 2026. URL https://arxiv.org/abs/2603.04737.

Alex L. Zhang, Thomas L. Griffiths, Karthik R. Narasimhan, and Ofir Press. VideoGameBench: Can vision-language models complete popular video games?, 2025. URL https://arxiv.org/abs/2505.18134.

Wenyu Zhang et al. WebGameBench: Requirement-to-application evaluation for coding agents via browser-native games, 2026a. URL https://arxiv.org/abs/2605.17637.

Yixin Zhang, Ziyi Wang, Yiming Rong, Haoxi Wang, Jinling Jiang, Shuang Xu, Haoran Wu, Shiyu Zhou, and Bo Xu. World models for policy refinement in StarCraft II, 2026b. URL https://arxiv.org/abs/2602.14857.

Congjie Zheng, Chuanyi Xue, Bin Liang, Jun Yang, and Changshui Zhang. SEAGym: An evaluation environment for self-evolving LLM agents, 2026. URL https://arxiv.org/abs/2606.17546.

Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Qiang Fu, Haobo Fu, Zilong Zheng, and Yitao Liang. MCU: An evaluation framework for open-ended game agents. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78221–78259, 2025. URL https://proceedings.mlr.press/v267/zheng25j.html.

Ziqian Zhong, Aditi Raghunathan, and Nicholas Carlini. ImpossibleBench: Measuring LLMs' propensity of exploiting test cases. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=SeO4vyAj7E.

A THE SUITE IN PICTURES

Figure 7 shows the three environments; one tile is a reader's rendering and two are the sessions' own views or artifacts. The roguelike tile is a replay frame from a shipped Fable 5 policy mid boss fight, drawn for the reader; during development the agent never sees this rendering, only the JSON observation stream the frame was drawn from. The StarCraft II tile is the API view one session saved at game start: the terrain-height and pathing rasters plus the unit list, exactly the arrays its controller consumes. The Civilization tile is the map an Astra session drew *for itself* at turn 600, embedded in its own trace: the agent built a renderer as an instrument, which is itself a small datum about how these sessions work. The fifteen ported single-file games (five classic ports and ten GameStore titles) are headless engines and leave no frames.

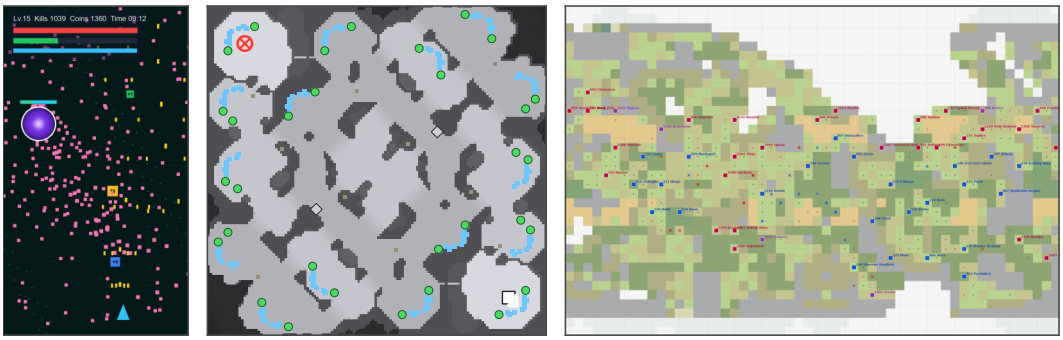


Figure 7: The suite in pictures: a roguelike replay frame rendered for the reader (80-seed exam; the agent sees only the JSON stream it is drawn from), the StarCraft II API view at game start (16-scenario exams), and an Astra session's self-drawn map of its Civilization view at turn 600 (16-game exams).

B PROTOCOL DETAILS

Contract. Every session receives: a plain-language game description; the raw observation/action interface (the roguelike's JSON protocol; StarCraft II's s2client protobuf via the launcher; CivRealm's Freeciv-web API); an empty policy file with the required entry-point signature; and general-purpose tools (shell, file editing, its harness's defaults). The prompt states that final evaluation uses unseen instances, names the objective (clear rate; win the scenario; win by conquest), and states no time, turn, or token budget. For the roguelike the prompt notes that the environment file is minified and reading it is not useful. No baseline, scaffold, policy representation, or refinement loop is supplied.

Harnesses and efforts. Claude models run under Claude Code and GPT models under codex, each at a fixed reasoning-effort setting per cell (the vendor's highest available unless a lower effort is the cell's label). The harness's context management, including compaction, is part of the environment (§6); optional harness features are disabled, and traces show near-bare tool loops. Harness and engine versions are recorded per session in the corpus manifest.

Exams. Roguelike: 80 frozen held-out seeds under a fixed step cap; R1 is the number of seeds cleared; R2 is the exam's mean per-seed utility: a cleared seed scores $1 + (\text{cap} - \text{win step})/\text{cap}$ (faster clears score higher; a win at the cap scores exactly 1, so the range is $[1, 2)$; cap 90,000 steps), and an uncleared seed scores its progress fraction in $[0, 1)$, so R1 at maximum does not pin R2; identical artifacts receive identical per-seed outcomes. In the intervention (§7, Appendix L), R2 names the same utility on a separate 1,000-seed reference exam that no agent ever saw. StarCraft II: 16 held-out scenarios (four unseen maps $\times$ four seeds) at the session's development tier, scored as wins. Civilization: 16 held-out full games per condition on a common seed list, scored as conquest wins; stall artifacts are re-run at the canonical watchdog timeout and resolved by game rules (Appendix C).

Reconstruction and replay. Intermediate versions are rebuilt from traces by faithful re-execution of the mutating commands (including codex's JS exec-wrapper channel and `apply_patch` envelopes, gated on each call's recorded exit status). Two checks gate every session: the reconstruction's final state must byte-match the shipped artifact, and every snapshot the agent left in its workspace must byte-match the reconstruction at that moment. All sessions used here pass; one roguelike session passes with whitespace-only differences, and we treat it as reconstructed. StarCraft II adds a recorded-versus-replayed convention: replays reproduce the recorded exam of the shipped build in 252/272 games, byte-identical engine builds agree 16/16, and figures state which of the two conventions they draw.

Construction controls. Arm A (one-shot) receives the byte-identical game description, interface document, and empty template that the other arms receive, and must emit the policy in a single response with no execution or revision: the Claude systems ran one turn with tools disabled, and the GPT system, whose harness cannot disable tools, was instructed to and audit-verified to make zero tool calls. Arm B (spec-only) runs the same harness, tools, budgets, and byte-identical prompt as arm C, but the workspace withholds the engine and runner behind a one-line notice that no executable environment is provided: agents write and run their own code, and all twenty sessions built simulators or fuzz harnesses from the spec, while sandboxing and per-session leak audits confirm no access to the real game. Arm C adds the executable engine and runner. Figure 2d draws five arm-A draws, five arm-B sessions, and six arm-C sessions per system, all scored on the same frozen 80-seed exam.

Scoring coverage. Coverage differs by game and is never selected on results. Roguelike and the ported single-file games score every emitted version of every gate-passed session. StarCraft II scores versions as an exam queue completes: milestone versions first, then a dense pass in version-index bisection order round-robin across sessions, a mechanical order fixed before any outcome is read; four sessions have every version examined and figures use complete-exam versions only, with per-session coverage in the corpus map below. Civilization examines shipped finals only (a full game takes minutes to hours). Sessions interrupted by external service errors (usage limits, capacity) resume on a bare *continue* with no other user text; segment counts for the case sessions appear in Appendix F.

Corpus map.

C REACH ATLAS

A frozen controller can be played at difficulties it was never developed for. Figure 8 shows two StarCraft II sweeps, drawn as separate blocks because their protocols differ and are never compared cell-for-cell. The uniform sweep (seeds 1–6 on every tier, cap 90k, one map) covers Astra's best Cheater:Insane controller (the 16/16 session; the 5/16 and 12/16 controllers were not swept), Astra's three Elite controllers, and Fable's Elite-A. The Insane-trained controller wins 59 of 60 games including 6/6 at Insane; the Elite controllers win 51–52 of 54 below Insane and 0–1 of 6 at Insane; every non-Insane loss falls on seed 3. The earlier July sweep is annotated with its own per-cell n and caps; Fable Elite-A agrees with it tier by tier (31 of 33 same-seed games), though it is not a replay: the engine build differs. Reach tracking the development tier is established for Astra's controllers; Fable's one Insane-developed policy reached only the mid fair tiers.

The Astra Civilization atlas plays the three normal-1v1 controllers across the seven other conditions (easy/normal/hard/cheating $\times$ one or two opponents; 336 games). Scoring counts wins and rule-based losses only: stall-timeout records are re-run at the canonical 1200 s idle timeout before entry (the protocol that resolved the seven held-out stall artifacts of §3: five defeats confirmed by game rules, one double-stall at a hopeless position recorded as a defeat, one Astra game resolved to a victory), and the 36 longest-playing games, which had not reached a rule-based outcome under this protocol, are excluded. Of the 300 decided games the controllers win 221 (74%): 117 of 140 (84%) against one opponent and 104 of 160 (65%) against two. Counting every excluded game as a loss still gives 66% (221/336), and the second-opponent deficit survives any resolution of the excluded games: single-opponent conditions stand at 81% or higher (117/144) while two-opponent conditions reach at most 71% (136/192). Three regularities. First, unlike the StarCraft II sweeps, reach here exceeds the development tier on the difficulty axis: hard 1v1 is won at 91% (43/47) and cheating 1v1 at 74% (35/47), against 85% at easy (39/46). Second, session-to-session spread opens only above the home condition: the three controllers are indistinguishable at normal 1v1 (16/16 each) yet split at cheating 1v1 (8/16, 15/16, 12/15), and the low outlier comes from the shortest development session (2.6 hours

Table 1: **Corpus map, single-file games with replayed versions.** Per model × effort (a cell; cells are never pooled, totals are inventories). Sess.: sessions reconstructed; Cens.: sessions that did not end on their own (capped, killed, or cut); Recon.: versions the reconstructor emitted; Scored: versions scored on the game's frozen 80-seed held-out exam (gate-passed sessions only). Blank: cell not run on that game.

		Roguelike				Mario				Luniland			
Model	Effort	Sess.	Cens.	Recon.	Scored	Sess.	Cens.	Recon.	Scored	Sess.	Cens.	Recon.	Scored
GPT-6-astra[d]	max	6	0	102	99								
GPT-5.6-sol	max	6	1	285	285								
GPT-5.6-sol	xhigh	3	0	105	105								
GPT-5.6-sol	high	3	0	78	78								
GPT-5.5	xhigh	4	0	116	116	4	0	82	82	4	0	53	53
GPT-5.5	high	4	0	94	94	4	0	69	69	4	0	63	63
GPT-5.5	medium	4	0	72	72	4	0	64	64	4	0	50	50
GPT-5.5	low	4	0	53	53	4	0	41	41	4	0	41	30[†]
GPT-5.4	xhigh	9	0	99	89[†]	5	0	72	70	3	0	34	34
Fable 5	max	7	1	437	437	5	0	331	288[†]				
Opus 5	max	6	5	260	260	2	0	77	77	2	0	58	58
Opus 5	max, uncapped	6	0	308	308								
Opus 5	xhigh	3	1	101	87[†]								
Opus 5	high	6	6	288	288								
Opus 5	high, uncapped	6	0	379	379								
Opus 5	medium	3	1	143	143								
Opus 5	low	2	1	76	19[†]								
Opus 4.8	max	5	0	396	311[†]	5	0	210	209	4	0	71	71
Opus 4.8	xhigh	5	0	300	300								
Opus 4.8	high	5	0	274	242[†]								
Opus 4.8	medium	5	0	227	227								
Opus 4.8	low	5	0	218	218								
Sonnet 5	max	6	0	281	281	5	0	437	271[†]	4	0	77	77
Sonnet 4.6	max	5	0	270	270	5	2	214	214	4	0	46	46
Haiku 4.5	max	5	0	139	139	6	0	157	136[†]	5	0	140	139
Total		123	16	5,101	4,900	49	2	1,754	1,521	38	0	633	621

[†]One session of the cell failed the reconstruction gates and is not scored (its versions count under Recon.): roguelike Opus 4.8 high and max (Read-oracle accuracy 0.45 and 0.84; finals differ in whitespace only), Opus 5 low and xhigh, GPT-5.4 xhigh (finals differ in code); mario Fable 5, Haiku 4.5 and Sonnet 5; luniland GPT-5.5 low. Exactly one scored session passes with a whitespace-only final (roguelike Opus 4.8 low).
[d]Reconstructed with the codex-0.155 front end; content-identical versions deduplicated before scoring (102 emitted, 99 unique), whereas the other roguelike sessions score every emitted version. Mario and luniland score 1,521 of 1,525 and 621 of 622 emitted versions of gate-passed sessions (5 scoring failures). Uncapped rows are the Opus 5 control arm (no wall cap), analyzed apart from cell statistics.

Table 2: **Corpus map, StarCraft II and Civilization.** Per model × effort × tier or condition. Sess., Nat., Cens.: sessions, natural stops, censored sessions (stopping-inventory labels). Recon.: sessions whose versions were reconstructed from the trace, with emitted version events and content-unique versions. Scored: versions with a complete held-out exam (StarCraft II: 16 scenarios, 4 unseen maps × 4 seeds, 48,000-loop cap; Civilization: 16 full games on a common seed list, shipped program only).

						Reconstructed			
Model	Effort	Tier or condition	Sess.	Nat.	Cens.	Sess.	Events	Unique	Scored
StarCraft II: 16 held-out scenarios per exam, own tier; Scored = complete exams									
Fable 5	max	Harder	2	2	0	2	142	135	15
Fable 5	max	Very Hard	4	4	0	4	393	386	191
Fable 5	max	Elite	5	5	0	5	585	579	115
Fable 5	max	Cheater:Insane	3	2	1	3	200	200	19
Opus 4.8	max	Harder	1	1	0	0	–	–	0
Opus 4.8	max	Very Hard	1	1	0	0	–	–	0
Opus 4.8	max	Elite	2	2	0	2	247	245	16
GPT-5.6-sol	max	Elite	3	1	2	1	43	43	6
GPT-5.6-sol	xhigh	Elite	2	2	0	1	20	19	9
GPT-6-astra	max	Elite	3	3	0	3	57	55	29
GPT-6-astra	max	Cheater:Insane	3	3	0	3	125	109	108
Sonnet 5	max	Very Easy pilot	2	1	1	0	–	–	0
StarCraft II total			31	27	4	24	1,812	1,771	508
Civilization: 16 held-out games per exam, own condition; shipped final only									
Fable 5	max	novice 1v1	4	4	0	–	–	–	4
Fable 5	max	novice 1v2	4	3	1	–	–	–	4
Fable 5	max	easy 1v1	4	4	0	–	–	–	4
Fable 5	max	normal 1v1	4	4	0	–	–	–	4
Fable 5	max	July pilots	6	1	5	–	–	–	0
GPT-5.6-sol	max	novice 1v1	3	3	0	–	–	–	3
GPT-5.6-sol	max	easy 1v1	3	2	1	–	–	–	3
GPT-6-astra	max	normal 1v1	3	2	1	–	–	–	3
Civilization total			31	23	8	–	–	–	25

StarCraft II Scored counts distinct versions (stage-2 dense replay 421, stage-1 milestone replay only 87), as of the corpus freeze for this submission. Tier names follow the in-game labels (API Hard, Harder, VeryHard, CheatInsane). Unreconstructed sessions: Opus 4.8 Harder and Very Hard (one each), GPT-5.6-sol max Elite a and b (censored), GPT-5.6-sol xhigh Elite rep 2 (no trace), and the two Sonnet 5 sessions of the early 2-game Very Easy pilot. Fable 5 Cheater:Insane 1 aborted on a harness artifact (censored, still replayed). Civilization censored: Fable novice 1v2 h3 (hung), the July pilots (spend limit, trace cut, nonzero exit, 12-h timeout), GPT-5.6-sol easy 1v1 i3 (operator-stopped), GPT-6-astra n3 (externally cut by a provider capacity limit; its exam is final). The July pilots predate the 16-game protocol and have no complete exam.

Table 3: **Corpus totals and the stopping inventory.** Left: exact counts behind the rounded corpus sizes in the text. Right: the 1,110-session stopping inventory by game (all arms; canonical arm and its natural stops).

Replayed and scored	Sessions	Versions
Roguelike, 112 gate-passed	112	4,801
Roguelike, GPT-6-astra	6	99
StarCraft II	24	508
Corpus in the text	**142**	**5,408**
with Fig. 3b's SC2 set only	138	5,377
plus mario and luniland	225	7,550
Civilization shipped finals	25	25
Stopping inventory (arm)	**Sessions**	**Natural**
Canonical	764	682
Fork branches	213	66
Fork prefixes	66	43
Spec-only control	20	20
Uncapped control	12	12
Archived cuts	10	0
One-shot control	5	0
Infrastructure tests	20	0
All	**1,110**	**823**

Game	All	Canon.	Natural
Roguelike	498	179	151
StarCraft II	31	31	27
Civilization	31	31	23
NetHack	5	5	5
Mario	55	54	51
Luniland	60	49	48
Racer	57	56	56
Flappy	44	43	41
Gorillas	43	42	42
Aqua Sort	26	25	25
Fling Feathers	27	25	25
Red Hot Vengeance	25	24	24
Jetpack Joyride	28	26	25
Cheese Chase	29	27	20
Shadow Boy (dropped)	1	1	1
Battle Zone	64	64	42
Bubble Cash	27	27	26
Megaman	27	27	24
Ao Oni	26	26	25
No game label[e]	6	2	1
Total	1,110	764	682

Scored versions are not deduplicated in the roguelike curves corpus (identical content scores identically); GPT-6-astra and StarCraft II count content-unique versions. The corpus in the text is the roguelike plus StarCraft II; mario and luniland carry scored curves too, and Civilization examines only shipped programs. The StarCraft II counts are as of the corpus freeze for this submission. 1,095 inventoried sessions have a trace; 1,104 carry one of 19 game labels (the ten titles from Aqua Sort to Ao Oni are AI GameStore ports). [e] Four infrastructure tests and two canonical roguelike sessions (Fable 5 max) carry no game label; one of the two is a natural stop missing from every per-game stopping count.

against 15.8 and 27.8; the longest was ended by an external provider limit, so its length is a lower bound, and its scoring is complete). Third, the second opponent costs more than any difficulty step: every 1v2 condition scores below its 1v1 counterpart, with the worst cell at 4 of 16 decided (cheating 1v2). The atlas grid, per-cell counts, and frozen policy hashes were committed before any game was played, and the submission snapshot is a committed per-game artifact.

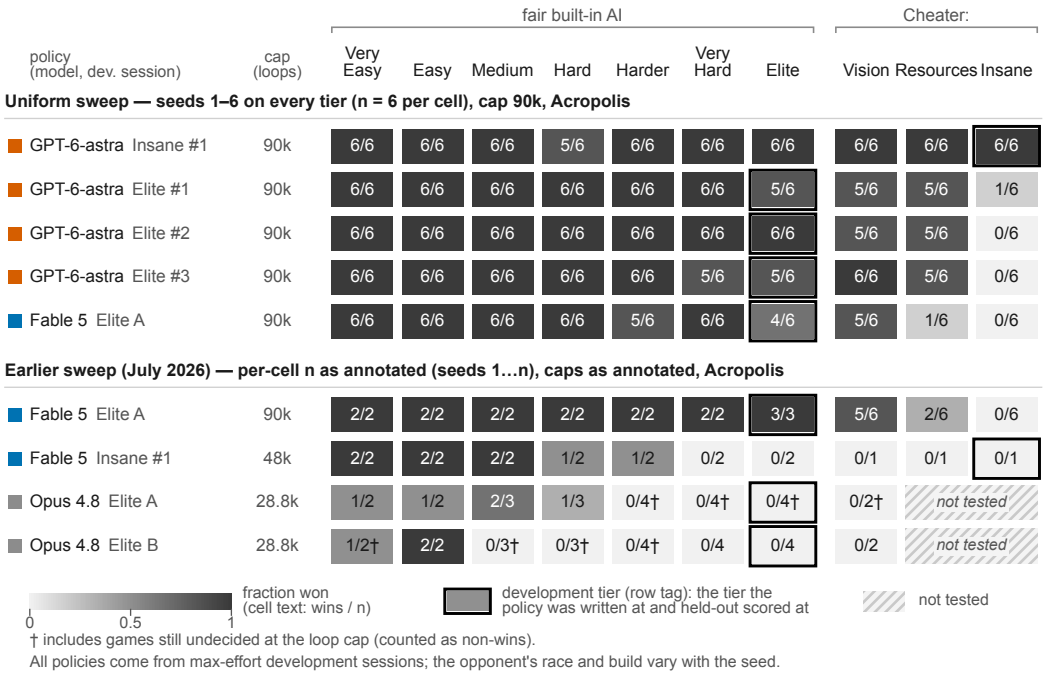


Figure 8: Reach follows the development tier: Astra's best Cheater:Insane controller sweeps the ladder (59/60); Elite controllers stop at Insane. Upper block: uniform sweep; lower block: the July sweep with per-cell n and caps as annotated.

D BREADTH, HUMAN ANCHORS, AND DIRECT PLAY

Three leaderboard extensions from the project's earlier phase, rebuilt from the run artifacts (hand-typed values in the earlier figures were re-derived; five greedy baselines changed as a result).

Breadth (Figure 9). On the five ported classics, only the ported Mario game separates models (Haiku 7.5% to Opus 5 100%; greedy baseline 11.25%); the other ports are within reach of most of the cast or of a greedy baseline, which is why the main text uses them for behavioral analyses rather than capability ranking. On nine public games with human reference medians, drawn from the GameStore catalog (Ying et al., 2026), the best single session's compiled controller beats the human median on all nine, from $1.06\times$ (Megaman) to $5.32\times$ (Battle Zone, Fable 5).

Direct play versus compiled agency (Figure 10). An exploratory protocol comparison, not a controlled experiment: direct play follows the GameStore convention (the model in the action loop, one fixed seed, one two-minute episode), while a compiled score averages 80 held-out seeds. Direct-play runs exist only for the earlier cast (Gemini 2.5 Flash and Pro, Gemini 3.1 Pro, MiniMax M3); no Claude or GPT direct-play harness was built. On Battle Zone the compiled-to-direct score ratio is $0.56\times$, $0.38\times$, $0.99\times$, and $1.41\times$ across the four systems in that order: neither interface uniformly dominates under these unequal protocols.

Effort sweep and wider cast (Figure 11). Across 28 roguelike cells and 149 sessions, effort moves cell means far less than model identity does, and not monotonically (Opus 4.8, low to max: 12.5, 24.0, 40.8, 21.3, 39.0%; Opus 5: 51.3, 81.3, 68.5, 87.5, 79.4%). Headline cells use each vendor's top effort setting (max for the Claude models, GPT-5.6-sol, and Astra; xhigh for GPT-5.4) with one

exception: GPT-5.5 is shown at high, the stronger of its two top efforts (19.4% against xhigh's 17.2%, $N{=}4$ each), a choice made after both cells were complete; the Claude headlines keep the top setting even where a lower effort scored better (Opus 4.8 high 40.8% vs max 39.0%; Opus 5 xhigh 87.5% vs max 79.4%). No generational comparison in the paper involves GPT-5.5. The earlier cast anchors the low end: GPT-5.4-mini and Gemini 3.1 Flash-Lite at 0%, Gemini 3 Flash 2.3%, MiniMax M3 10.5%, Gemini 3.1 Pro 12.9%.

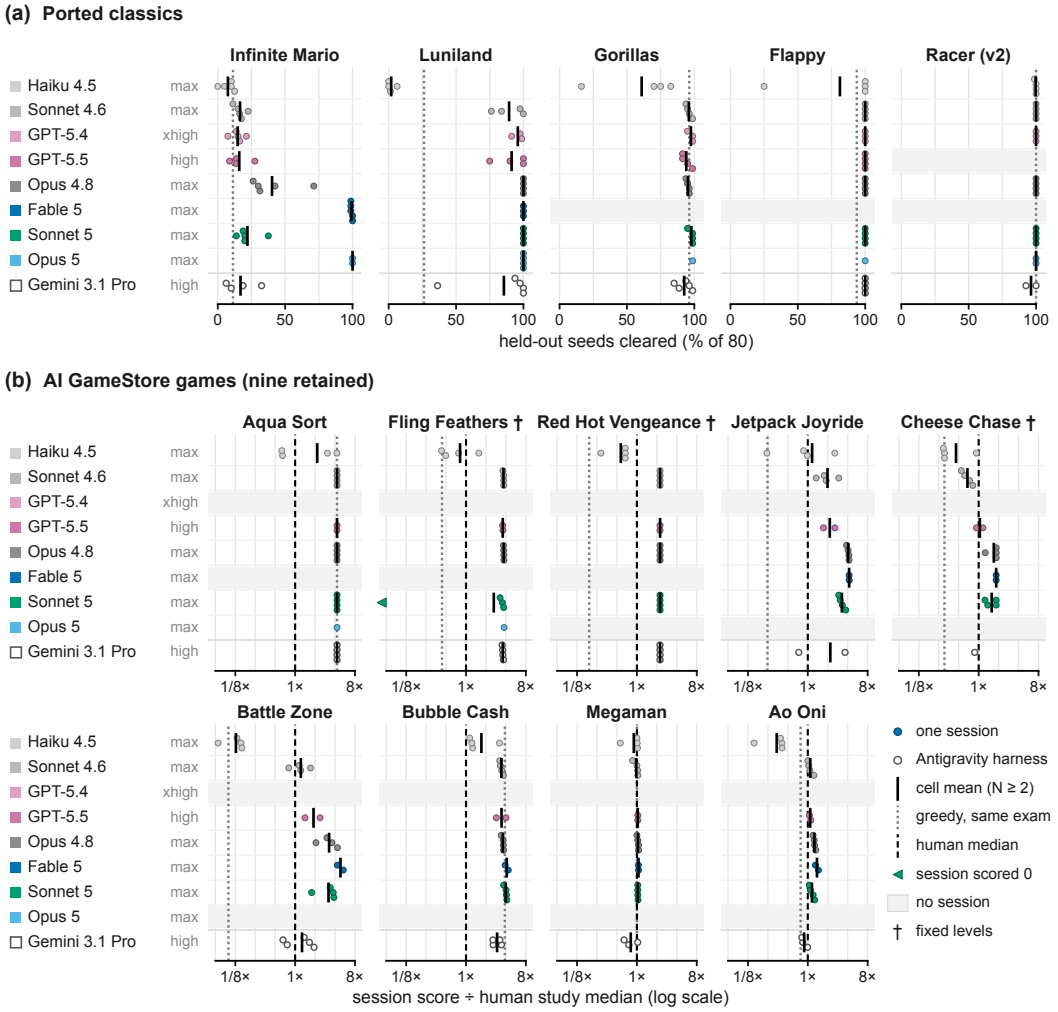


Figure 9: Breadth: ported classics and public games, task-normalized, with human anchors.

E Economics conventions

Costs are list-price equivalents computed from each session's own token accounting: input, cached-input, and output tokens are priced at the provider's published per-million rates at run time (Astra and Fable share identical list prices; sol's are lower), so cross-vendor comparisons are price-neutral by construction. Astra's served tier is unknown; the whisker in Figure 2e prices its sessions at up to 2× (the fast tier), which still leaves it below Fable. Cost per held-out clear divides a cell's summed session cost by its summed exam clears; cost per StarCraft II win divides by held-out wins. Astra's advantage is volume, not price: it consumes 17–20% of its comparators' input tokens for equal or better exams. Per-session token counts and prices are in the supplementary material.

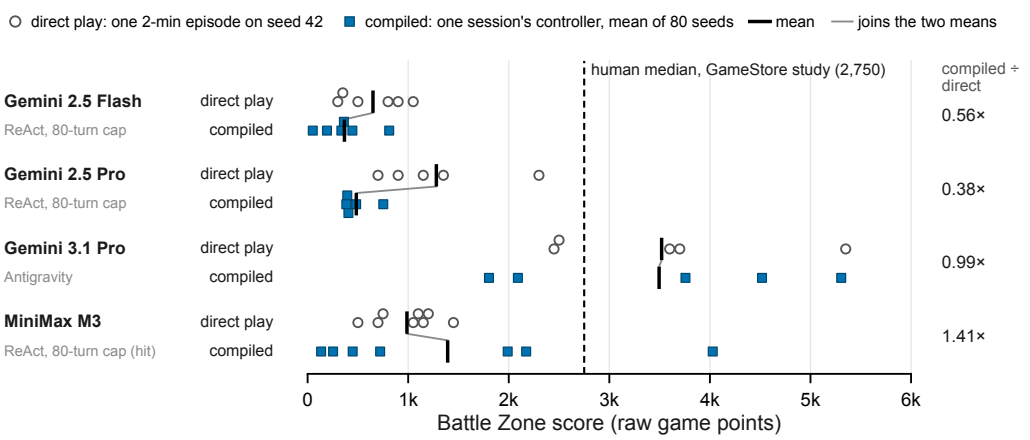


Figure 10: Direct play versus the compiled controller, same models, same game.

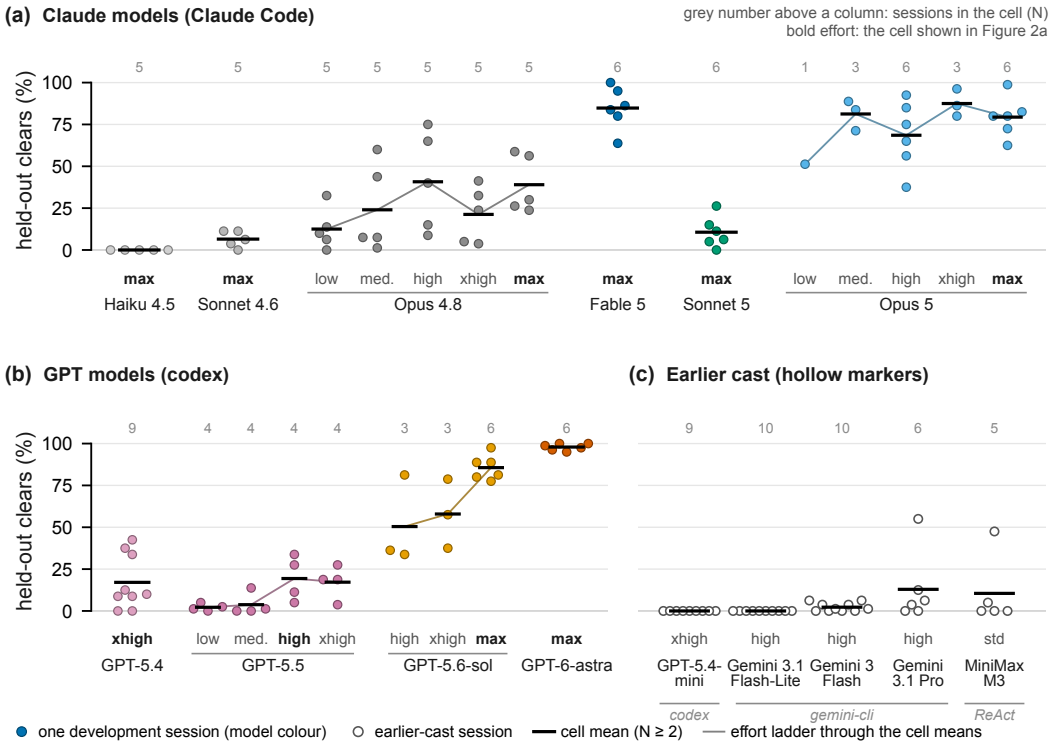


Figure 11: Roguelike held-out success by model × reasoning effort, wider cast included.

F RESEARCH-EPISODE AUDIT

Protocol and taxonomy. An episode is labeled only when the trace contains all four elements: an observation, a stated explanation, an executed test that could discriminate the explanation, and a decision. Outcomes are supported, revised, rejected, or unresolved; narration without an executed test is never counted. Episodes are read from the agent's messages and commands; hidden reasoning is never used. A keyword-guided pass over 39 sessions produced 204 labeled episodes; a second reader, blinded to session identity and outcome, reproduced nine of ten sampled status labels (seven of eight strict) — a precision check on windows the first reader chose, which by construction cannot detect missed episodes.

The recount. Recall was measured by re-reading two Civilization sessions end to end with a quote-verified ledger (every quoted line is programmatically checked verbatim at its stated trace position). The keyword pass had labeled only each session's opening stretch: all its labels fall in the first 22% of one trace and the first 12% of the other. The full re-reads count 31 and 127 strict episodes where the pass found 9 and 7, about one strict episode per policy version the agent played (0.94 and 1.32), with looser tune-and-test loops only 10–20% more numerous and pure tune-and-rerun occurring 1–3 times per session. Labeling judgment moves the strict counts by roughly $2\times$ (the most restrictive reading gives about 20 and 65–70); reading coverage moved them $3.4\times$ and $18\times$. The revise-or-reject share is 26% in both re-reads. This calibration is why §4 reports the audit rates as floors and the corpus total as several hundred.

Case contrast.

Table 4: **Civilization case contrast.** The three GPT-5.6-sol (max) sessions at easy 1v1, run from byte-identical prompts and conditions.

	i1	i2	i3
Wall-clock span (h)	85.6	42.9	394.3
Active hours (idle between segments excluded)[a]	44.5	24.2	192.6
Task segments (resumes after external cuts)	5 (4)	4 (3)	10 (9)
Tool calls	8,973	5,380	49,978
`policy.py` versions played (edits)	96 (220)	138 (344)	315 (607)
Separate policy files written	60	1	1,201[b]
Strict research episodes	127[c]	≥ 7[d]	≥ 12[d]
Fresh full-length tests of the shipped build	0[e]	0[f]	0[f]
Held-out exam (conquest wins of 16)	12	16	0
How the session ended	own stop	own stop	operator stop[g]

[a] Sum of the codex task segments; the gaps between segments are external interruptions (usage limit, server capacity, authorization) waiting for an operator's bare *continue*. i3's span includes a 187-h spend-limit gap. [b] Plus 4 in five side rollouts; 297 are numbered `policy_candidate_vN` files, and the shipped program imports a closure of 349 files. [c] Full quote-verified re-read. [d] Floor from the keyword-guided audit pass, which found 7 in i1 before the re-read counted 127. [e] i1 ran its exact final twice, as restored-state endgame replays on reused development seeds (1 win). [f] Never ran the exact shipped build; i3's last tested build differs from it by 336 lines. [g] All nine i3 segments before the last ended on an external error, never on an own stop; the operator ended the session after 394 h. i1 and i2 each ended on their own after 4 and 3 resumes from usage-limit or capacity cuts.

Generational tests. In the matched roguelike cell, all six Fable 5 sessions exceed the best of five Opus 4.8 sessions; under exchangeability the two-sided probability of disjoint support is $2/\binom{11}{5} = 0.0043$. For GPT-5.6-sol $\rightarrow$ Astra the supports overlap by one session (sol's best, 78/80, against Astra's worst, 76/80); Astra's six sessions span 76–80 where sol's span 62–78.

G WHAT GETS COMPILED

Doctrine convergence (Figure 13). Three Fable 5 Civilization sessions, run in isolation from one another, independently implement nearly the same nine-element strategic doctrine: Dijkstra pathing over terrain costs, a research beeline to a city-taking unit, expand-then-militarize, early contact and containment, massed synchronized assault, fog re-sweeps to hunt hidden cities, civil-war management, anti-stall rules, and defensive robustness. Every cell of the matrix is verified against the shipped source the held-out scorer loaded, with the implementing line cited; 26 of 27 cells verify directly, and one (g1's civil-war handling) reaches the same goal by a different mechanism — it never defers the rival's capital but hunts the civil-war splinter directly. The three implementations differ in architecture (a role state machine, an ordered decision pipeline, a world model with a campaign state machine) and in outcome (4, 10, and 1 of 16 held-out), so the convergence is doctrinal, not textual, and doctrine alone does not set the exam score.

Code scale (Figure 14). Across 104 leaderboard sessions, shipped policy size spans more than two orders of magnitude (16 lines to about 4,000, counting the scored file plus its transitively imported local modules) and rises with game scale; within a game, size does not order held-out outcome, so scale is a description of what compiled agency produces, not a predictor of quality.

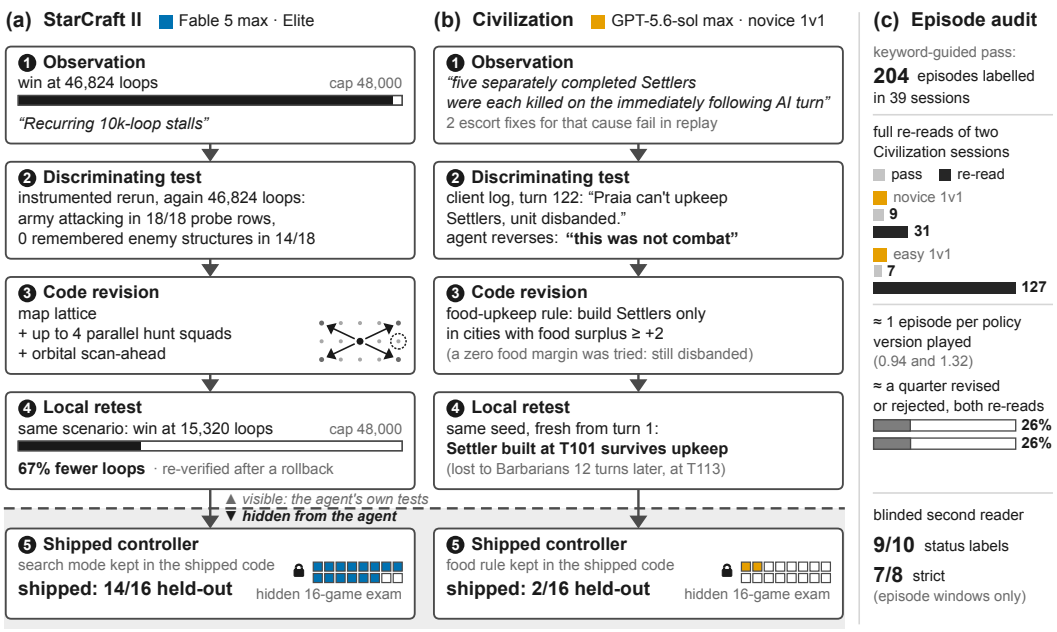


Figure 12: The StarCraft II success chain and the Civilization failure chain of §4, side by side, with the audit statistics.

Fable 5 max · Civilization, novice 1v1 · three isolated sessions

8 of 9 elements implemented
independently in all three sessions

	doctrine element	g1 role state machine + parallel squads 1,535 lines · 4/16 held-out	g2 ordered decision pipeline + flow-field pathing 1,428 lines · 10/16 held-out	g3 world model + campaign state machine 1,862 lines · 1/16 held-out
1	Dijkstra pathing over terrain costs grid shortest paths weighted by per-terrain move cost	✓ `pathing.py:95`	✓ `pathing.py:41`	✓ `pathing.py:84`
2	Research beeline to a city-taking unit ordered tech plan through Mathematics (Catapult)	✓ `policy.py:37`	✓ `policy.py:42`	✓ `policy.py:53`
3	Expand, then militarize city target that adapts to how late the war starts	✓ `policy.py:654`	✓ `policy.py:545`	✓ `policy.py:67`
4	Early contact and containment units dedicated to finding (and harassing) the rival early	✓ `policy.py:546`	✓ `policy.py:948`	✓ `policy.py:1430`
5	Massed, synchronized assault a wave waits until enough attackers are ready	✓ `policy.py:763`	✓ `policy.py:443`	✓ `policy.py:968`
6	Fog re-sweep to hunt hidden cities revisit long-unseen land after the map is explored	✓ `policy.py:1075`	✓ `policy.py:1176`	✓ `policy.py:805`
7	Civil-war management defer a large rival's capital (its fall splits the empire)	◐ `policy.py:255`	✓ `policy.py:412`	✓ `policy.py:882`
8	Anti-stall: release, rotate, escalate stuck targets are dropped, shelved or escalated	✓ `policy.py:762`	✓ `policy.py:362`	✓ `policy.py:880`
9	Robustness: never raise, cap, resume act() cannot throw; per-turn step cap; resumable state	✓ `policy.py:106`	✓ `policy.py:91`	✓ `policy.py:239`

✓ implemented: the cited line of the shipped source is the implementing code ◐ same goal, different mechanism

Row 7: g2 and g3 defer the rival's capital while its empire is large (policy.py:412, policy.py:882);
g1 never defers it, and instead treats any later player scoring over 30 as a rival and hunts the civil-war splinter (policy.py:255).

Figure 13: Doctrine convergence across three isolated Civilization sessions, each cell verified against the shipped source.

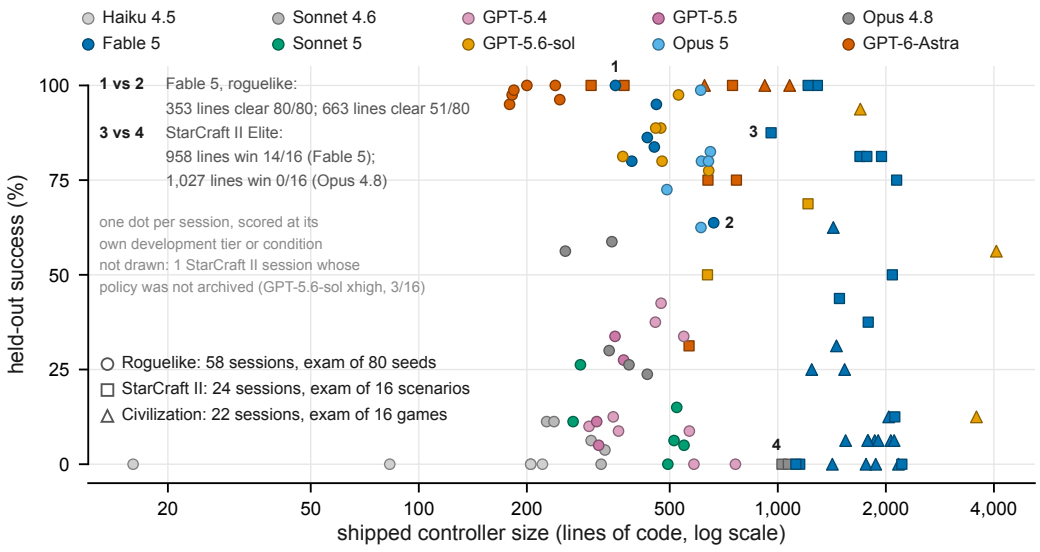


Figure 14: Shipped policy scale against held-out outcome, by game and model.

H CURVES, FAMILIES, AND FORECASTS

Noise decomposition. Per-session trajectories are jagged, and the exam is deterministic, so jaggedness has two sources that can be separated: consecutive versions differ on a median of 11.5 seeds, and 69% of what a smooth per-session fit misses is genuine version-to-version policy change against 31% attributable to the finite 80-seed panel. This is why curves are described at cell level. Curve panels draw every capped session, including censored ones: Fable 5 max contributes 7 sessions here where the leaderboard cell counts its $N{=}6$ naturally ended ones.

Families. Five parametric families (saturating exponential, power-gap, logistic-type alternatives) and a nonparametric cell template all fit held-out sessions within 0.18 points of one another out of sample; the saturating exponential trails the best family by 0.08. Nothing in the paper depends on the exponential being true; it is used because its two parameters are interpretable. Fitted to cell means (dashed in Figure 3a): level L recovers the drawn cell's final mean within 1.2 points where cells plateau (Fable 85.4, sol 86.0, Astra 97.9, Opus 4.8 41.3, Sonnet 5 11.8; the drawn sets include censored sessions and exclude gate-failed ones, so L is compared to the curve set's own mean, not the leaderboard cell: Opus 4.8 max draws four of five sessions, Fable seven), the time scale τ spans 0.12–0.24 of a session, and the wall-fraction at nine tenths of final level (p90) spans 0.27–0.42; τ ranks cells' nonparametric time-to-plateau (Spearman $\rho{=}0.78$). Dwell after the plateau runs 0.2–0.96 of the session by cell. Exposure-sensitive variants (excluding previously inspected sessions) move levels, not orderings.

Registered forecasts. A forecasting suite specified before these data were opened asks how much of a fresh session's remaining trajectory its prefix predicts. Given one checkpoint, last-value persistence is the worst of nine predictors (normalized suffix error 1.05 against 0.49–0.50 for the cell template and the transferred curve); by ten checkpoints only the level-anchored transferred curve still beats persistence (-0.07, intervals excluding zero in all exposure sets), and the session's own recent slope adds nothing. Predictive intervals built from 1–3 training sessions per cell cover 40–75% at large widths: trajectories carry little session-specific signal beyond current level plus the cell's shared shape.

Axes companion (Figure 15). The same StarCraft II sessions on three x-axes (wall hours, version index, cumulative agent-run games), with the agent's own view beneath the hidden exam: the agent's cumulative win rate on its own games tracks the exam loosely at best.

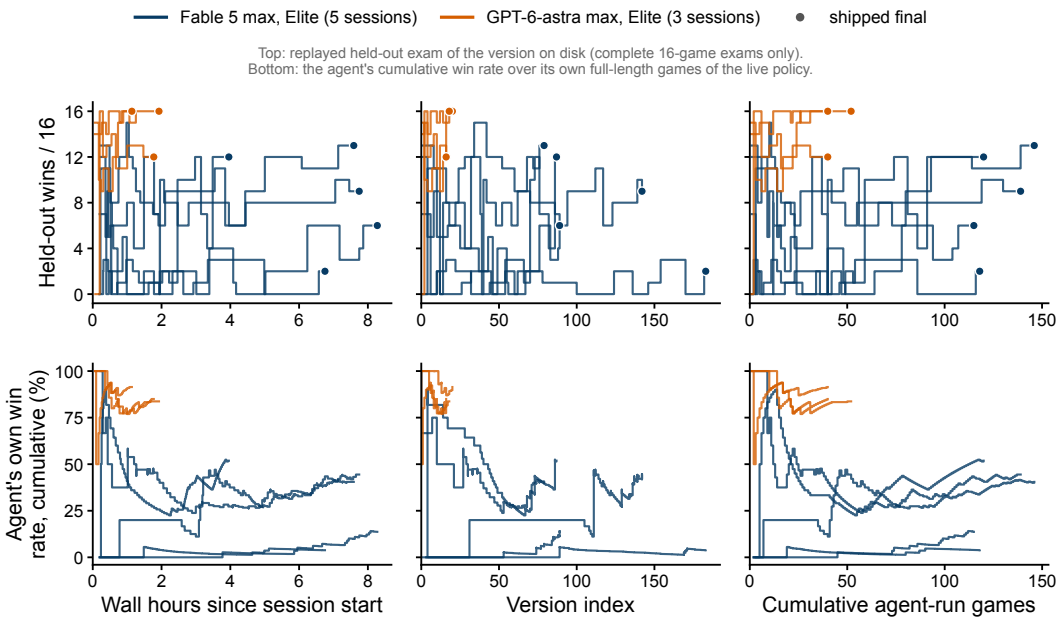


Figure 15: StarCraft II Elite sessions on three x-axes; top: hidden-exam wins, bottom: the agent's own cumulative win rate on the games it ran.

I Churn mechanism

Between consecutive versions on the fixed 80-seed exam, a step's gains are seeds flipping to cleared and its losses seeds flipping to failed; a first-time gain is an exam seed cleared for the first time in the session. These are exam-side quantities, independent of which seeds the agent chose to test. Figure 16 shows the per-cell medians by session fifth; the tercile summary per frontier cell: gain share of churn 0.52–0.63 early to 0.49–0.51 late (lower late in 19 of 21 cells, sign test p=0.0002); first-time share of gains 13–90% early to 0–8% late (lower late in 21 of 21); churn 24, 21, 25 seeds/step falling to 5, 5, 11 in three cells while Sonnet 5 stays low throughout (4.7 to 4.0); pure-gain steps fall 7–16% to 0 and reverts rise 0 to 9–20%, each in three of the four frontier cells (sol's pure-gain share instead rises 0 to 8.8%; Opus 5's reverts stay 0). Two alternatives are refuted by the same bitmaps: "early wins are free" fails because even early net-positive edits usually lose seeds (79% of early net-positive steps drop at least one; pure gains are about 10% of early steps pooled, cell medians 0–16%); "interference grows" fails because the per-seed loss hazard falls (0.50 to 0.24). Instrument sessions, run far past the normal horizon, decay to exhaustion without finding a second regime, and measured net gain per hour collapses from first to last third by at least $44\times$ (sol) to effectively unbounded factors (Fable, Opus 5, Sonnet 5), steeper than the fitted exponential's own $9.5\times$.

J The agent's evidence

Recovering what the agent saw. A command-ledger extraction alone recovers 54% of the evaluation results agents actually read; results also arrive through background shells, `write_stdin` polls, redirected files, and custom summarizers. Trace-level recovery reaches 86%, and every agent-view analysis uses the trace-recovered set. Astra's roguelike batteries are the cautionary case: simple parsing found 2–7 results per session where the full series holds 19–92. The final-claim statistics of §5 count claims (21 from 18 sessions); clustered by session, 9 of 11 capped and 6 of 7 uncapped sessions claim above the exam at least once.

Final battery. The final battery is the last evaluation run(s) of the exact shipped file (or a copy differing only in comments or whitespace), plus the unbroken run of earlier evaluations of it; sessions whose shipped file was never evaluated fall back to the last evaluation, and each case is tagged. Under this rule, 65 of 91 self-terminated roguelike sessions ship with zero never-before-used seeds; battery-

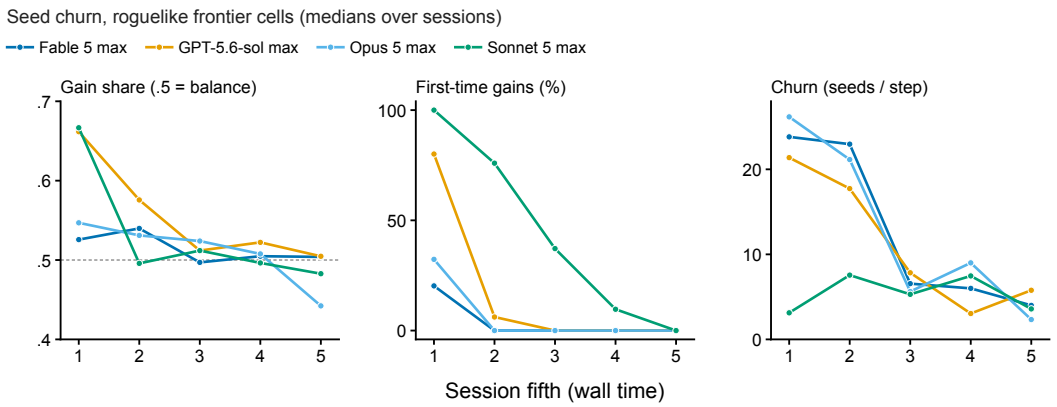


Figure 16: Per frontier cell by session fifth: the gain share of churn converges to one half, the share of gains that clear an exam seed for the first time in the session falls to zero, and churn per step falls—late gains are recoveries, not new territory.

size medians by model are 6 (GPT-5.4, sol, Haiku, Opus 4.8), 13 (GPT-5.5), 30–50 (Sonnet 4.6, Sonnet 5, Fable), and 138 (Opus 5). Astra's batteries are 60–171 seeds in four sessions but mostly reused (0–64 new), and in two sessions a post-battery edit shipped on a single-seed check. Across all self-evaluations, about 80% reuse seeds already used in the session (Figure 17).

Big-game validation ledger. Of 41 sessions inventoried, 39 carry decidable validation evidence (22 Civilization, 17 StarCraft II; two StarCraft II sessions have no readable trace or workspace policy), and the ledger's 170 evaluation events belong to those 39, which Figure 4 draws. Fresh full-length tests of the exact shipped build have medians 1.5 (Fable, Civilization; 5 of 16 sessions none), 0 (sol, Civilization; 5 of 6 none), and 2 (Fable, StarCraft II). Seven sessions never executed the shipped file (two after only an inert edit to a tested build); of the other five zero-fresh sessions, four ran the build only as restored-state replays, truncated games, or unrecorded launches, and one ran a byte-identical entry file whose imported modules were later edited, so its shipped bundle has no confirmed test. The opposite extreme also exists: one Fable Harder-tier session ran 15 full-length tests of its shipped build, won all 15, and scored 16/16. Validation scope does not predict exam outcome across cells.

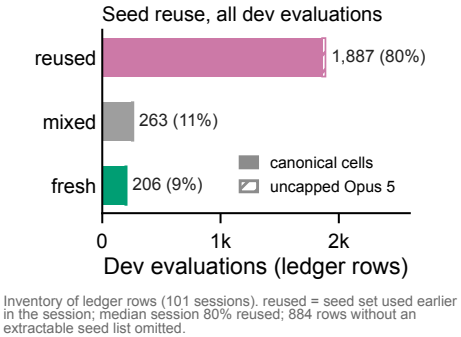


Figure 17: Seed reuse across all self-evaluations (inventory; instrument arms drawn separately).

K STOPPING

Clock audit. No prompt or harness message states elapsed or remaining time. Fragments leak into 95 of 101 audited sessions (tool-call wall times printed by codex, agents timing their own runs, file timestamps); elapsed session time could be worked out before the stop in a minority of sessions, and exactly one agent ever reasoned with a clock about its own session, not as a stop reason.

Saturation rules. A session is saturated at the first agent-visible result showing the task solved: roguelike, a battery of the agent's own with $n \geq 8$ all cleared or $n \geq 10$ at $\geq 95\%$ on the live policy; StarCraft II, consecutive full-game wins on one unchanged version; Civilization, a conquest victory in a full game. Timing is the line where the agent saw the complete result. After correcting a parsing gap that missed background-shell batteries, all 14 of Astra's natural stops are saturated (its fitted budget is therefore unidentifiable), and the roguelike saturated share is 13/145 (≈9%).

Stop-point models. Fitted per model, never pooled across models, on canonical-arm natural stops only (self-terminated with a clean exit record), pooled over that model's games among the eight solve-criterion games, in cumulative fresh input tokens (wall minutes secondary), restricted to stops with at least one parseable agent-visible result: of the 367 natural stops in the ten fitted models, 355 carry a token clock, 25 of those lack a saturation verdict and are excluded, and 330 enter. The budget-only reference is a lognormal over the stop point, $\log T \sim \mathcal{N}(\mu_b, s_b)$. The race model stops at whichever comes first of budget and saturation-plus-wrap-up, $T = \min(S + W, B)$, with $\log B \sim \mathcal{N}(\mu_b, s_b)$ and a model-level (not per-game) shifted-lognormal wrap-up $\log(W + w_0) \sim \mathcal{N}(\mu_w, s_w)$, $w_0 = 1000$ tokens (1 minute), fitted by penalized maximum likelihood (a small ridge on the log standard deviations): a never-saturated stop contributes $f_B(T)$, a saturated one $f_B(T)\,P(W > T-S) + f_W(T-S)\,P(B > T)$. This is the untruncated minimum-of-clocks form used as an unnormalized score: the shift admits wrap-up delays down to $-w_0$ (a timing-jitter tolerance), and no conditioning normalization by the probability of stopping after S is applied, so the saturated-stop term is a pseudo-likelihood rather than a calibrated conditional density. Models are compared by leave-one-session-out mean log predictive density among models with at least six natural stops, three saturated. The race model's held-out density is conditional on the held-out session's observed saturation status and, when saturated, its saturation point S, information the budget-only baseline never receives: the comparison is therefore an exploratory score comparison between the two forms on the same coordinate, measuring what the agent's own decisive evidence adds to locating the stop, not an unconditional forecast of whether or when saturation occurs and not a calibrated predictive advantage; the registered hazard analysis, which needs no such form, is the confirmatory result. Externally terminated sessions enter none of these fits; censoring is handled in the registered hazard analysis, where wall-capped, provider-cut, and stub sessions contribute exposure to the discrete-time risk set (calendar bins from a session's first own evaluation) but never an event. The Figure 5 budget line is $\exp(\mu_b)$ from the race fit on all of a model's eligible natural stops.

Budgets and eligibility. The across-game invariance compares per-game medians of never-saturated stops within a model, for model×game strata with at least three such stops; six of seven eligible models sit within $1.15\times$ (Opus 4.8 the exception at $1.86\times$; a two-stop rule would add Haiku's flappy stratum and lift its spread to $1.18\times$), while fitted budgets span $13\times$ across models ($12.7\times$ exactly) and sessions spread two- to three-fold within a cell. The race model beats budget-only out of sample in 7 of 10 fitted models (7 of 9 with Astra excluded as not evaluable). The coordinate is cumulative fresh (uncached) input, a billing coordinate: in Fable's Civilization sessions 60–75% of it is prompt-cache rewrites after cache misses, and net of rewrites three of four sessions stop on Fable's ordinary budget; compaction triggers sit near 244k context under codex and 143–168k for Claude Code with the older Claude models, and crossing them splits behavior as §6 describes. Because re-reads bill the same context repeatedly, the cumulative coordinate runs ahead of live context: at a session's actual first compaction it already stands at $0.9\times$ the band (Sonnet 4.6, whose caching is clean) up to 1.05–$2.29\times$ (codex) and $2.7\times$ (the one Fable 5 compaction, at 999.6k context, indicating the newer Claude Code trigger sits near 1M). A stop drawn past a band without a ring therefore never crossed the trigger: GPT-5.5 stops at 300–500k cumulative carry a median live context of 89k, and three of Fable's four Civilization sessions reach 1.8–2.8M cumulative at 180–700k context with zero compactions. Compaction is rare outside one model: 38 of 330 drawn stops ever compact, 24 of them Sonnet 4.6's (24 of its 28), 14 of the remaining 302 elsewhere.

The wrap-up (Figure 18). In the easy games the post-saturation phase is work, not idling: 73% of saturated sessions keep editing and 38–100% reach a new best. The stop is failure-gated — a failing result is last in 0–6% of cases per model, a clean re-check of the unchanged final policy is last in 52–60% for GPT-5.5, GPT-5.4, and Haiku, whose stops the confirmation routine times (+25, +7, +10 nats out of sample); Sonnet 5, Opus 5, and Fable polish for 18–125 minutes with nothing visible beyond "no failure now" predicting which clean check is last. Polish length tracks the headroom the game leaves (1.2–$78\times$ longer where a continuous tiebreak remains), while strict marginal-gain threshold rules never beat a constant per-model wrap-up length.

Table 5: **Stopping: which stops enter which analysis.** Canonical natural stops in the eight games with a saturation rule, for the ten models the race model is fitted for, ordered by fitted budget. Usable: the stop has a token clock (cumulative fresh input). Sat./Never/Unkn.: the agent's own visible results showed the task solved before the stop / never did / cannot be parsed. Invariance strata: games with ≥ 3 never-saturated stops (their count), the rule of Figure 5; Spread: max/min of those game medians. Budget: fitted transcript budget (fresh input). LOO: race minus budget-only, mean log predictive density per stop, leave-one-session-out.

	Natural stops								LOO (nats/stop)	
Model	All	Usable	Sat.	Never	Unkn.	Invariance strata	Spread	Budget	Tokens	Minutes
Haiku 4.5	38	38	15	22	1	R 8, M 6, L 5	1.08×	102k	+0.76	+0.83
GPT-5.5	98	98	50	32	16	R 16, M 15	1.15×	196k	+0.22	+0.28
GPT-5.4	26	26	12	14	0	R 9, M 5	1.06×	203k	+0.51	+0.34
Opus 4.8	56	56	13	39	4	R 29, M 5, S 4	1.86×	382k	+0.24	+0.40
Sonnet 5	28	28	17	11	0	R 6, M 5	1.06×	571k	+1.16	+0.93
Sonnet 4.6	29	29	20	8	1	R 5, M 3	1.04×	754k	+1.46	+1.31
Fable 5	43	32	20	11	1	R 4, S 6	1.00×	1.01M	+0.29	+0.06
Opus 5	16	16	11	5	0	R 5 (one game)	–	1.04M	−0.27	−0.23
GPT-5.6-sol	19	18	6	10	2	R 9 (one game)	–	1.30M	−0.02	−0.12
GPT-6-astra*	14	14	14	0	0	none (no never-sat. stop)	–	n.e.	n.e.	n.e.
Total	367	355	178	152	25					

Games: R roguelike, M mario, L luniland, S StarCraft II. Not usable: 11 Fable 5 Civilization ladder sessions with census data only and one GPT-5.6-sol StarCraft II session without a trace. Unknown: 16 GPT-5.5 racer sessions, 4 Opus 4.8, 2 GPT-5.6-sol (StarCraft II), 1 each Haiku 4.5, Sonnet 4.6, Fable 5. A $\geq$ 2-stop rule would add only Haiku 4.5 flappy (2 never-saturated stops) and raise Haiku's spread from 1.08× to 1.18×, above the 1.15× line. Opus 5 and GPT-5.6-sol have one eligible game each and no spread; their race models do not beat budget-only. Budgets span 12.7× across the nine evaluable models (9.8× across the seven where the race model wins). *Not evaluable: after the relabel (background-shell results recovered), all 14 GPT-6-astra natural stops are saturated, so no never-saturated stop identifies a budget; the fit on the earlier labels (431k, LOO −0.15) is superseded, and the relabeled refit collapses onto the largest stop (3.60M, 95% profile 0.82M to unbounded; LOO −4.43). The relabel moves the roguelike saturated share from 7/145 (4.8%) to 13/145 (9.0%; 91% never saturated) and StarCraft II from 12/23 to 14/24. One natural Fable 5 roguelike stop (613k) lacks a game label and is missing here; as a never-saturated stop it would move Fable's spread from 1.00× to 1.05×.

Registered hazard. On 112 roguelike sessions (96 natural stops), the one agent-visible feature that adds leave-one-session-out likelihood over an age-plus-cell baseline is hours since the agent's last own evaluation (ΔNLL -33 $[-63, -9]$), with a negative sign: stops follow evaluations. Time-since-record helps alone but is subsumed by activity; evaluation counts and batch size are worse than baseline. This registered analysis constrains what stopping reads; it does not identify an internal cause, and the race model itself is post hoc.

L INTERVENTION DETAILS

Mechanics. A harness hook pauses the parent session at a fixed landmark on a tool boundary; two branches resume from the identical message. A validation server serves seed panels: the reuse arm receives the parent session's panel, the refresh arm a fresh 50-seed panel from the same pool at each call. The arm is never visible to the agent; realized panel serving is logged per call; branches run to their own end under equal caps, and both shipped programs take the same hidden exam. Endpoints, fixed per cell in the registration: R1 held-out clears of 80, R2 the utility of Appendix B on the 1,000-seed reference exam. One experimental-materials erratum, common to both arms: the validator's textual legend read "crit = per-seed utility (win speed; 0 = not cleared)", while the scorer actually returns partial progress in $[0, 1)$ for uncleared seeds (Appendix B). The numeric feedback both arms received came from the same correct scorer, and the served per-seed records carried the nonzero uncleared scores, so the data contradicted the legend in view of the agent; paired differences are unaffected, but any statement about agents' textual understanding of the objective carries this error, and the intervention under a corrected legend was not run.

Attempt ledger and per-pair results. Table 5: 12 pairs initiated, 9 completed; the three incompletions were safeguard refusals during branch resumption, falling on one reuse arm, one refresh arm, and one pair losing both, so incompletion is not concentrated in either arm; one additional pair excluded at freeze time was later recovered by a clean rewind-and-resume with no added user message (without it: 8 of 8, $+8.75$ clears). The reported estimand is the completed-pair effect. Treating the four missing branch scores as free parameters in $[0, 80]$ bounds the all-initiated 12-pair mean between -7.9 and $+25.4$ points; these are identification limits, not estimates. Completed-pair

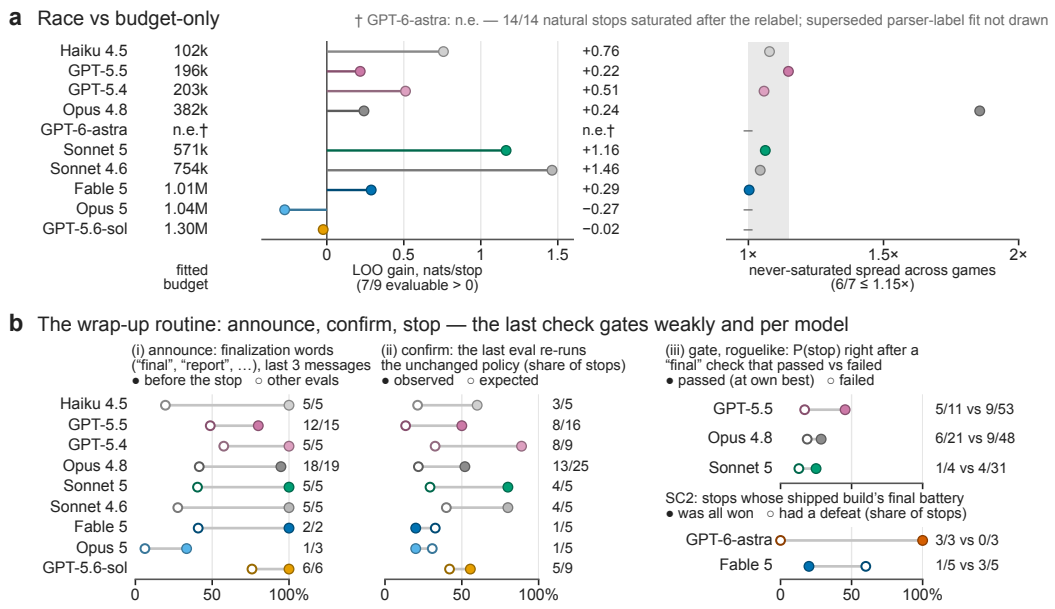


Figure 18: The wrap-up routine per model: race-model gains, finalization language, unchanged-policy re-runs, and the final-check gates.

R1 differences: $2, 2, 23, 11, 10, 19, 16, 5, 1$ (all positive; mean $+9.89$ clears $= +12.4$ points). Opus 5 (the registered replication): 12 of 12 pairs, R2 effect $+0.0102$ (t=0.83; refresh ahead 6/12; sensitivity sets $+0.002/-0.000$); on the registration's other endpoint, instructed utility on the 80-seed exam, the effect is $+0.0156$ $[-0.0028, +0.0340]$, refresh ahead 7/12, also null; forks at R1 median 80/80; the two pairs forked below the ceiling (64 and 68) move most on R2, -0.028 and $+0.139$, one in each direction. The exam ceiling is a candidate explanation for the null, not an established cause: model, landmark hour (5.55 against 1.6), and uptake covary with it. Sonnet 5: 8 pairs, R1 CI -8.5 to $+2.5$ points, forks 1–37/80; three refresh and three reuse branches made zero validator calls, and in one pair neither arm did, so treatment uptake is low and the stratum is analyzed with all assigned pairs (zero-call subgroups only as descriptive sensitivity). Between arms, stop times differ by less than their spread (Fable refresh−reuse $+0.36$ h, 95% CI $[-0.81, +1.52]$), so no timing conclusion is drawn in either direction.

Exposure. Fork prefixes are parent-session content the agent had already produced; the exposure ledger accounts for which panels each branch could have seen. Four of the eight Sonnet 5 fork prefixes contain environment-source reads (Appendix M).

M Integrity and Exposure

Source reads. The roguelike prompt states the minified environment file is not useful to read. Nevertheless 30 of 292 canonical roguelike trials (10%) read it: Sonnet 5 12, Opus 5 12, Opus 4.8 3, Fable 5 2, Sonnet 4.6 1, and the GPT models, Astra, and the earlier cast none (32 trials flagged in total; one is a superseded network-failure attempt and one a false positive whose only match is an exclusion glob). Four of the eight Sonnet 5 X2 parent prefixes are among the readers. Readers average 4.0 fewer clears than non-readers in matched cells (3 of 10 cell contrasts reverse; some cells hold one or two readers). Reading is self-chosen, so this is exposure accounting and an absence of observed advantage, not a causal bound on what reading could yield.

Prior-analysis exposure. Some sessions predate this paper's analyses and were inspected in earlier project phases; every analysis where it could matter reports exposure-sensitive variants (excluding previously inspected sessions), and none changes an ordering reported in the main text.

Table 6: **Panel-refresh intervention: every initiated pair.** Fork: the parent's clears at the landmark, the state both branches resume from. R1: clears on the frozen 80-seed exam (1 clear = 1.25 points). R2: mean instructed utility on the 1,000-seed reference exam that no agent ever saw. Δ: refresh − reuse. Calls: validator calls served to the branch. Wall: branch wall-clock hours. Lower panel: means over completed pairs with paired 95% t intervals.

		R1 clears (of 80)			R2 utility (1,000 seeds)			Calls		Wall (h)		
Pair	Fork	Reuse	Refr.	Δ	Reuse	Refr.	Δ	Reuse	Refr.	Reuse	Refr.	Status
Fable 5 (max): landmark 1.6 h; 12 initiated, 9 completed (8 complete, 1 recovered), 3 incomplete												
p01	60	69	71	+2	1.3362	1.4084[a]	+0.0722	6	19[b]	6.00[c]	4.16	C
p02	28	24	26	+2	0.8066	0.8330	+0.0264	12	8	1.01	0.84	C
p03	43	49	72	+23	1.2086	1.4368[a]	+0.2282	7	8	2.69	3.38	C
p04	54	51	62	+11	1.1335	1.3254	+0.1919	3	4	1.03	1.95	C
p05	36	–	45	–	–	1.0435	–	–	14	–	6.00[c]	I
p06	14	24	34	+10	0.7433	0.9513	+0.2080	13	10	4.54	5.60	C
p07	16	35	54	+19	1.0029	1.1770	+0.1741	13	13	6.00[c]	5.43	R
p08	69	–	–	–	–	–	–	–	–	–	–	I
p09	45	50	–	–	1.2062	–	–	11	–	6.00[c]	–	I
p10	26	30	46	+16	0.8520	1.0100	+0.1580	32	27	1.85	5.40	C
p11	73	66	71	+5	1.4097	1.4623	+0.0526	19	29	2.02	2.44	C
p12	24	42	43	+1	1.1114	1.1342	+0.0228	36	41	6.00[c]	5.17	C[r]
Opus 5 (max), registered replication: landmark 5.55 h; 12 initiated, 12 completed (5 untouched, 7 recovered)												
p01	79	79	79	0	1.7923	1.8027	+0.0104	7	8	3.93	3.05	C
p02	80	79	80	+1	1.8068	1.8087	+0.0019	1	5	3.79	5.27	C
p03	80	79	79	0	1.6354	1.6439	+0.0085	7	4	5.39	5.04	C
p04	80	76	80	+4	1.7816	1.7943	+0.0126	3	7	6.00[c]	5.01	R_b
p05	64	74	79	+5	1.7540	1.7256	−0.0284	1	1	6.00[c]	6.04[c]	R_b
p06	68	74	77	+3	1.6488	1.7881	+0.1393	6	7	6.00[c]	6.00[c]	R_b
p07	78	80	80	0	1.7768	1.7838	+0.0071	4	5	6.00[c]	6.00[c]	R_{pb}
p08	80	80	80	0	1.8093	1.7992	−0.0101	7	12	6.00[c]	6.00[c]	R_p
p09	80	79	80	+1	1.8364	1.8292	−0.0072	2	6	4.46	2.53	R_{pb}
p10	80	80	80	0	1.6555	1.6537	−0.0018	4	2	6.00[c]	2.54	R_p
p11	78	80	80	0	1.7971	1.7925	−0.0046	3	6	6.00[c]	6.00[c]	C
p12	80	79	80	+1	1.8222	1.8164	−0.0058	4	5	5.70	6.00[c]	C
Sonnet 5 (max): landmark 0.94 h; 8 initiated, 8 completed												
p01	11	13	12	−1	0.5934	0.5958	+0.0023	8	5	0.49	0.19	C
p02	20	20	15	−5	0.7324	0.7029	−0.0296	0	6	0.18	3.39	C
p03	37	31	33	+2	0.8016	0.8800	+0.0785	13	5	0.88	0.56	C
p04	9[s]	7	9	+2	0.5994	0.5501	−0.0493	19	0	1.69	0.01	C
p05	7[s]	7	7	0	0.5098	0.5098	0.0000	0	0	0.01	0.02	C
p06	2	16	2	−14	0.6614	0.4217	−0.2397	4	0	1.01	0.07	C
p07	35	35	32	−3	0.9001	0.8784	−0.0217	13	6	0.66	0.37	C
p08	1	2	2	0	0.4217	0.4189	−0.0028	0	3	0.72	1.34	C

Refresh − reuse [95% CI]	Fable 5 (n=9)	Opus 5 (n=12)	Sonnet 5 (n=8)
R1 clears	+9.89 [+3.69, +16.09]	+1.25 [+0.13, +2.37]	−2.38 [−6.77, +2.02]
R2 utility (1,000 seeds)	+0.1260 [+0.0631, +0.1890]	+0.0102 [−0.0167, +0.0370]	−0.0328 [−0.1094, +0.0438]
Utility, 80-seed exam	+0.1287 [+0.0599, +0.1975]	+0.0156 [−0.0028, +0.0340]	−0.0313 [−0.0947, +0.0321]
Refresh ahead, R1 / R2	9/9 / 9/9	6/12 (6 tied) / 6/12	2/8 (2 tied) / 2/8 (1 tied)
Wall hours	+0.36 [−0.81, +1.52]	−0.48 [−1.27, +0.31]	+0.04 [−1.17, +1.25]
Capped, reuse / refresh	3 / 0	7 / 6	0 / 0

Status. C complete; C[r] complete, both branches finished by rewind-and-resume after safeguard refusals or a spend-limit kill, and counted among the complete pairs; R recovered: excluded as a pair under Addendum 6.4 (a branch refused three times) and completed after the freeze by the same rewind-and-resume; I incomplete: an arm never completed within the attempt rule (safeguard refusals; the arm that did complete is shown). Opus 5 recoveries after infrastructure terminations (Addendum 10): R_p prefix re-run from t=0, R_b branch resumed or re-forked, R_{pb} both. [c]Ended at the 6-h branch cap (censored); every other branch stopped on its own. [a]Partial R2 exam: 700 (p01) and 600 (p03) of 1,000 seeds scored; utility averaged over scored seeds. [b]Realized panel repeat 0.54 (0.68–0.82 by the 15th call): this refresh branch drew from the original 500-seed pool before Addendum 5 enlarged it to 3,000; the mean share of already-seen seeds per call is 0.00–0.27 in every other refresh arm and 0.75–1.00 in every reuse arm. [s]The prefix stopped on its own before the landmark; both branches resume from its final state.

Fable 5. Completed-pair R1 differences 2, 2, 23, 11, 10, 19, 16, 5, 1 (mean +9.89 clears = +12.4 points, 95% CI +4.6 to +20.1); without the recovered p07, 8 of 8 positive, mean +8.75. *Opus 5.* All 12 R2 differences are listed; the two forks well below the R1 ceiling (p05 at 64, p06 at 68 of 80) are the largest negative (−0.0284) and the largest positive (+0.1393) R2 movements, and the other ten lie within ±0.0126. On R1 they move +5 and +3 while p04, forked at 80, moves +4 (its reuse branch fell to 76); three more forks sit one or two clears below 80 (p01, p07, p11). On the 80-seed exam's utility (the metric Addendum 9 names first, R2 following) p05 and p06 are the largest (+0.0680) and third-largest (+0.0527) gains, p04 (+0.0563) between them. *Sonnet 5.* Zero validator calls in the refresh arms of p04, p05, p06 and the reuse arms of p02, p05, p08 (p05: neither arm). *Stop times.* Refresh − reuse wall hours are within noise in every model and truncated by the cap: Fable 5 +0.36 h [−0.81, +1.52] with 3 reuse and 0 refresh branches capped; Opus 5 −0.48 h [−1.27, +0.31] with 13 of 24 branches capped; Sonnet 5 +0.04 h [−1.17, +1.25], none capped.

N Extended related work: games and language-model agents

Direct play. Benchmarks evaluate online decision-making with the model choosing actions (Paglieri et al., 2025; Wu et al., 2024; Hu et al., 2026; Park et al., 2026; Zhang et al., 2025; Li et al., 2025a; Ying et al., 2026; Guertler et al., 2025; Costarelli et al., 2024; Duan et al., 2024; Yao et al., 2025; Lin et al., 2025; Cui et al., 2025; Hu et al., 2025; Google DeepMind and Kaggle, 2025; Murphy, 2026; Yue et al., 2026; Li et al., 2026; Ouyang et al., 2026; Lin et al., 2026; Bailis et al., 2024; Cipolina-Kun et al., 2025; ARC Prize Foundation, 2026; Karten et al., 2026), alongside roguelike, puzzle, and open-world environments and the trained or planning agents built for them (Küttler et al., 2020; Matthews et al., 2024; Cloos et al., 2024); Hu et al. (2024) survey the area. **Agents in games.** Scaffolded or trained agents act through skills, code, or keyboard and mouse while a model stays in the loop (Wang et al., 2024a; Fan et al., 2022; Liu et al., 2025b; Zheng et al., 2025; Tan et al., 2025b; Hopkins et al., 2025; Karten et al., 2025; SIMA Team, 2025; Tan et al., 2025a; Wang et al., 2025b; Jeurissen et al., 2024). **StarCraft II.** From the learning environment and reinforcement-learning milestones (Vinyals et al., 2017; 2019; Sun et al., 2018; Samvelyan et al., 2019) to language-model agents playing through text, full action spaces, or vision (Ma et al., 2024; Li et al., 2025b; Shao et al., 2024; Li et al., 2025c; Shen et al., 2025; Ma et al., 2026; Hong et al., 2025; Sol & Najjaran, 2026; Ma et al., 2025), world-model-assisted decision loops (Zhang et al., 2026b), and LLM-written decision-tree scripts for micromanagement (Deng et al., 2024). **Strategic reasoning, Civilization, and other strategy games.** (Qi et al., 2024; Chen et al., 2025; 2026a; Andrews et al., 2026; Wang et al., 2025a; Tang et al., 2026; Wang et al., 2024b; Meta Fundamental AI Research Diplomacy Team (FAIR) et al., 2022; Duffy et al., 2026). **Programs that play.** (Verma et al., 2018; Xu et al., 2025; Kuang et al., 2025; Eberhardinger et al., 2025; Liu et al., 2025a; Hennes et al., 2026; Lv et al., 2025; Lehrach et al., 2026; Kumar et al., 2026; Gumin et al., 2026; Wang et al., 2026b; Lou et al., 2026; Yang et al., 2026; Acher & Jézéquel, 2026; Sherwood et al., 2026). **Agents that build games,** a different deliverable, are measured by Chi et al. (2026); Jiang et al. (2026); Luo et al. (2026); Zhang et al. (2026a); Chen et al. (2026b).