AHABENCH: DO AGENTS TURN EXPERIENCE INTO REUSABLE INSIGHTS? A LONG-HORIZON BENCHMARK FOR CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A solved example can help an agent complete a task without preparing it to solve the next one. AHABENCH tests how agents use experience in later tasks and decisions. AHA-PUZZLE tests questioning on a new mystery after solved episodes; AHA-EULER tests computation on new inputs after teaching; AHA-VENDING follows decisions and feedback over a simulated business year. The benchmark leaves the learning mechanism open, and our evaluation is on fixed-weight agents. Earlier AHA-PUZZLE solutions help some models, but scores remain far below solving with hints. Qwen 3.6 Plus gains $+12.57$ points over an attempt without earlier episodes. On single-step AHA-EULER tasks, it reaches 97.3% held-out accuracy with an explanation and code, versus 8.1% with the question and answer alone. In nominal AHA-VENDING operation, Doubao Seed 2.0 Pro finishes at +$495, but its profit averaged over the year is -$10 because recovery arrives late. We release tasks, validators, a simulator, and evaluation records for studying how agents reuse experience on new tasks and sustain productive behavior over time.

1 INTRODUCTION

An agent's experience should make its next task easier. A solved mystery can suggest a better question; a worked example can reveal a reusable computation; a failed purchase can teach an inventory constraint. Continual learning studies how capabilities accumulate and remain useful across sequential tasks (Parisi et al., 2019; Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017). For an interactive agent, this means using experience when the next task changes the inputs or removes earlier guidance.

The test comes when the agent must act again: asking questions about a different mystery, computing on new inputs, or making another stocking decision. Evaluation therefore needs to connect what the agent experienced to what it can subsequently do.

We use *insight* to mean a reusable rule or relationship, such as a questioning strategy, recurrence, or inventory constraint. AHABENCH evaluates its usefulness through later behavior. AHA-PUZZLE compares exploration of the same story with and without earlier solved episodes. AHA-EULER varies what a related example teaches before a new-input test. AHA-VENDING follows decisions and their consequences over a simulated year.

The results locate three obstacles. **Exploring without hints.** Earlier AHA-PUZZLE episodes improve some agents' final exploration, yet none sustains its directly supported performance as hints disappear; the largest decline is in explanation completeness. **Reconstructing a method.** On single-step AHA-EULER tasks, every model reaches at least 80% when teaching includes an implementation, yet four fall below 20% when it is withheld. **Maintaining operation.** In AHA-VENDING, Hy3 Preview earns positive mean profit while completing only one of five incident-setting runs.

The benchmark specifies experience access, downstream tests, and task-specific scores. Its protocols support context, memory, skill, or parameter updates under the same evaluation rules; our fixed-weight agents provide a baseline for using experience through context and feedback. The released tasks support comparing learning methods on three requirements: better questions on a new story, correct computations without supplied code, and productive decisions across a long horizon.

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.

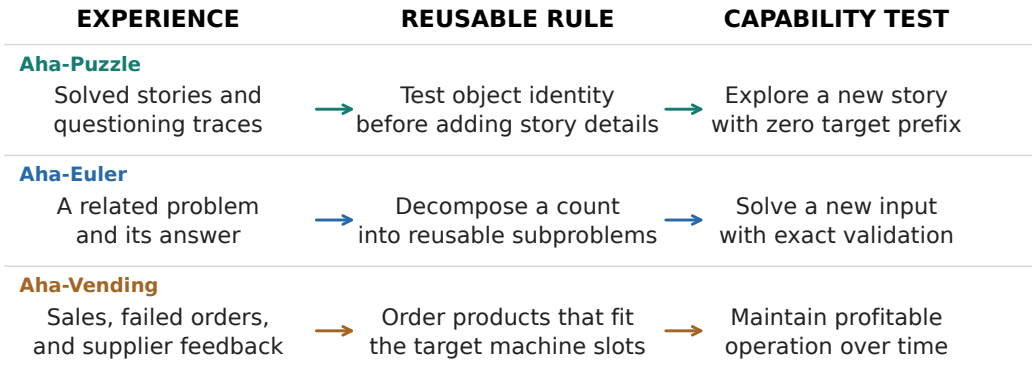


Figure 1: Three ways experience can become useful. Each row links an experience to an illustrative rule and a later test. The rules illustrate possible insights, not extracted model representations. Evaluation measures exploration, new-input computation, and continued operation.

2 Benchmark Design

Each component specifies what the agent has encountered, what it can still access, and what it must do next (Figure 1). Aha-Puzzle scores the recovered explanation and the questions used to find it. Aha-Euler checks exact answers on new inputs. Aha-Vending measures profit and continued operation.

Aha-Puzzle retains completed episodes and their solutions. Aha-Euler keeps the teaching exchange in context. Aha-Vending returns recent feedback alongside a business state shaped by earlier actions. These inputs let experience affect the evaluated agents even though their model weights remain fixed. The protocols below specify the information available to each agent, so future learning methods can be compared with the same access.

2.1 Aha-Puzzle: Exploration After Solved Episodes

Aha-Puzzle tests whether an agent asks useful questions. Each puzzle begins with an incomplete story, and the agent has up to 20 questions to uncover its hidden explanation. A host answers using the hidden solution; a separate grading model applies a rubric to the agent's final account. The seven underlying stories, called identities, appear in Chinese, English, Russian, and Japanese, with three repeats. Human-checked rubrics assign up to 70 correctness points and 30 efficiency points for explanation coverage and economical exploration.

The agent encounters seven stories in a sequence called a curriculum. Before asking its own questions, it receives the opening turns of a reference question–answer dialogue for the current story. This supplied prefix shrinks from all available steps to 10, 8, 6, 4, 2, and 0, and counts toward the 20-question budget. After each episode, a compact reveal record gives the official solution, reference trace, model guess, and score. At the final position, t7, six earlier reveal records remain available, but the new story has no hints. This is the *trace-free endpoint*. A separate *cold* episode tests the same story, language, and repeat without that history.

Seven balanced schedules per language and repeat place every story at every position once in the planned design. The full design contains 84 cold/t7 pairs and 588 curriculum episodes per model; analyses use the valid paired or complete-curriculum sets defined below.

The intended transfer is a questioning strategy, not the earlier story's answer. For example, an explanation involving a mistaken object suggests checking object assumptions in a different mystery. The agent must decide when that question is useful and recover the new explanation. The host is GPT-5.4 Pro and the rubric executor is GPT-5.4, both at temperature zero. Appendix C gives the scoring rules and a 100-trajectory audit of agreement with final human grades.

Human checking covers the hidden solutions, allowed host responses, rubrics, and reference traces. Final human grades agreed with the rubric executor on the 100 audited trajectories. The explanation criteria and question counts also allow individual grades to be inspected.

For example, the *Waterweed* rubric requires a complete explanation to connect a failed rescue, a misidentified object, a statement about the river, and the resulting realization. The agent's questions must establish these links before it stops. Correctness scores explanation coverage, while efficiency scores question cost.

Asking what was touched tests the mistaken-object hypothesis; asking why a later statement changes the protagonist's understanding connects it to the ending. Guessing the object without establishing the causal chain can still lose correctness points. Earlier reveal records show how questions establish these links, giving the agent examples to draw on when choosing questions for another story.

2.2 AHA-EULER: RECONSTRUCTING A COMPUTATION

AHA-EULER tests whether an agent can apply a computation to new inputs after seeing a related example. Project-Euler-style problems define input families: each family varies the inputs while retaining a deterministic answer program and exact validator. A taught instance q_{teach} and a held-out instance q_{test} share the computation. The bank includes 50 single-step tasks and 90 graph-composition tasks, in which two to five subcomputations, called nodes, depend on one another. Generated variants and hardened inputs extend the test beyond a canonical numerical answer, following the executable task family approach of Cheng et al. (2026).

The teaching conditions determine what the agent can use on the held-out input. *Cold* presents that input directly. *Full teaching* appends the gold answer, explanation, and reference code after each taught-node attempt. *Partial teaching*, or answer-only teaching, appends the answer. Each condition uses a separate transcript; the teaching exchange remains visible when the held-out prompt is added. Every node and final attempt uses a local Python calculator, and the final answer is whitespace-normalized for exact matching.

Without reference code, partial teaching requires the agent to derive or retrieve the method and implement it; comparison with cold solving tests what the teaching exchange adds. Full teaching instead permits adaptation of a supplied implementation. Graph composition requires carrying intermediate answers between dependent nodes (Appendix B).

2.3 AHA-VENDING: SUSTAINING OPERATIONAL DECISIONS

AHA-VENDING tests the continued usefulness of operational decisions. In this open-source simulator, inspired by Vending-Bench (Backlund & Petersson, 2025) and its incident-focused successor,[1] the agent orders inventory, reads supplier messages, stocks slots, sets prices, collects cash, and advances time. Feedback supplies evidence about demand, slot fit, delivery delays, and cash-flow constraints. A useful rule must guide repeated actions as these conditions evolve. A run ends at 365 days, bankruptcy, or a watchdog triggered by 20 days without a successful order.

Each decision is a fresh request containing a fixed instruction, the current rendered business observation, and at most eight recent transition summaries. The environment state carries the accumulated consequences of earlier actions. The main evaluation uses five incident-setting seeds per model, including refunds, phishing, machine issues, and delayed procurement. Daily trajectories in the nominal setting track profit accumulation, recovery, and operating persistence. We report both settings by name throughout: incidents test resilience to disruption, while nominal trajectories reveal the rhythm of everyday operation.

3 MEASURING TRANSFER, SUPPORT DEPENDENCE, AND PERSISTENCE

We first compare performance on the same targets with different experience or teaching. We then follow performance across the AHA-PUZZLE curriculum and the AHA-VENDING year. The measures below report how much agents improve, how much help they still need, and how long productive behavior lasts.

[1] Vending-Bench 2: `https://andonlabs.com/evals/vending-bench-2`.

Endpoint gain. In AHA-PUZZLE, the endpoint is the final story attempted with earlier solutions but no current-story hints. For targets with valid cold and endpoint scores, collected in $\mathcal{I}$, let B_i, T_i be those scores. Define

$$B = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} B_i, \qquad T = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} T_i, \qquad \Delta_{\text{end}} = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} (T_i - B_i). \tag{1}$$

The endpoint T measures later competence; Δ_{end} measures improvement over the same targets' cold scores. AHA-PUZZLE intervals resample identities, keeping translations and repeats together. Appendix C specifies uncertainty; Appendix A checks story and weighting sensitivity.

Support-withdrawal profile. Let $\mathcal{C}$ be the complete retained curricula. At stage t, let $S_{j,t}$ be the score and $B_{j,t}$ the cold score for that stage's target identity. We report

$$E_t = |\mathcal{C}|^{-1} \sum_{j \in \mathcal{C}} (S_{j,t} - B_{j,t}), \qquad \mathrm{W} = \frac{1}{6} \sum_{t=1}^{6} E_t - E_7. \tag{2}$$

Read E_t as the mean gain over cold performance at stage t, and the *withdrawal gap* W as supported gain minus hint-free gain. Thus a positive W means the endpoint retains less advantage than the supported stages. All stages use the same complete curricula. The contrast summarizes performance across the curriculum as a whole, as earlier solutions accumulate, target stories change, and their supplied reference prefixes shrink to zero.

Procedure gap. For the valid full/partial AHA-EULER intersection $\mathcal{V}$, define

$$G_{\text{proc}} = \frac{100}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} (Y_i^{\text{full}} - Y_i^{\text{partial}}). \tag{3}$$

The two indicators record exact-answer correctness under full and partial teaching. Thus G_{proc} measures the benefit of receiving an explanation and implementation in addition to the answer. We report both exact-match rates beside the gap. The cold comparison uses targets scored in all three conditions; each contrast is paired within a model, with 95% bounds specified in Appendix C.

Sustained operational outcomes. For cumulative net profit $p_s(d)$ recorded at each day's end, we report terminal profit, completion, and survival, together with time-averaged profit $A_s = 365^{-1} \sum_{d=1}^{365} p_s(d)$, maximum drawdown, and early/late earning rates. Time-averaged profit averages this running total over the year, rewarding early accumulation over an equally large last-day return. Drawdown records the largest decline from a previous peak. Fixed 90-day windows, days 0–90 and 275–365, describe how earning rates differ across the run. After termination, profit is held at its final value for this fixed-horizon accounting view; all seeds remain represented. These temporal metrics preserve dollars and days, and describe the joint outcome of policy, history, and environment state.

4 EVALUATION AND RESULTS

4.1 EVALUATION SETUP

AHA-PUZZLE and AHA-EULER each compare seven models; AHA-VENDING includes overlapping models and additional operational baselines. The first two components use Qwen 3.6 Plus, Kimi K2 Thinking, and Gemini 3.1 Pro Preview; AHA-VENDING uses Qwen3 Max, Kimi K2.6, and Gemini 3.1 Pro. Each condition comparison uses shared valid targets within a model. Cross-model values describe their respective evaluation sets; supplementary analyses compare weighting and execution reliability.

AHA-PUZZLE uses story-level resampling and checks sensitivity to individual stories and weighting. AHA-EULER applies the recorded error rule, including budget, timeout, and malformed-response failures, before forming task intersections. An all-requested analysis also counts these failures in end-to-end accuracy. AHA-VENDING uses two-sided Student-t intervals over five seed-level profits with four degrees of freedom. The manuscript source archive provides the analysis tables and figure inputs for the reported comparisons and curves.

Table 1: AHA-PUZZLE: mean competence after the curriculum and gain from the cold condition. Trace-free scores measure exploration after the prior solved episodes; gain is the change from the cold score. Intervals reflect resampling of puzzle identities.

Model	Cold	Trace-free	Gain	Cluster 95% interval
Gemini 3.1 Pro Preview	41.62	44.42	+2.80	[-2.55, +9.76]
Qwen 3.6 Plus	24.49	37.06	+12.57	[+7.42, +20.00]
Doubao Seed 2.0 Pro	32.21	36.29	+4.08	[-0.84, +9.75]
Claude Opus 4.6	30.86	33.14	+2.29	[-6.00, +13.20]
GPT-5.4	22.00	28.67	+6.67	[+4.00, +9.33]
Kimi K2 Thinking	22.38	18.54	-3.84	[-22.45, +4.34]
DeepSeek v3.2 Thinking	9.40	15.33	+5.93	[-1.33, +14.26]

Table 2: AHA-PUZZLE gain over the same targets' cold scores. E_t: mean gain at stage t. Supported advantage: mean of $E_1, \ldots, E_6$; endpoint advantage: E_7; withdrawal gap W: supported minus endpoint. All columns use the same complete curricula and are in rubric points.

Model	Supported advantage	Endpoint advantage	Withdrawal gap
Claude Opus 4.6	45.43	+2.29	43.14
DeepSeek v3.2 Thinking	56.53	+5.93	50.60
Doubao Seed 2.0 Pro	42.52	+4.08	38.44
GPT-5.4	42.67	+6.67	36.00
Gemini 3.1 Pro Preview	38.87	+2.80	36.07
Kimi K2 Thinking	49.67	-3.84	53.50
Qwen 3.6 Plus	50.06	+12.57	37.49

Selected trajectories illustrate how failures unfold; quantitative rates and intervals use the evaluation sets specified for each comparison.

4.2 AHA-PUZZLE: EARLIER SOLUTIONS HELP, BUT HINTS STILL MATTER

Gemini 3.1 Pro Preview reaches the highest observed endpoint, 44.42, with a gain of 2.80 over cold exploration (Table 1). Qwen 3.6 Plus reaches 37.06 with the largest gain, $+12.57$, and an identity-cluster interval of $[+7.42, +20.00]$. The strongest final solver is therefore not the model that improves most from the curriculum.

GPT-5.4 also has a positive endpoint gain, $+6.67$, with interval $[+4.00, +9.33]$. Qwen 3.6 Plus's gain stays at 9.44–14.03 points when each story is omitted from aggregation in turn. Appendix A gives the full story and weighting analyses. The curriculum profiles show how these endpoint gains compare with earlier solving under direct hints.

Qwen 3.6 Plus scores 86.36 at the first position and 86.68 at the second, then 60.83 with a two-step prefix and 37.06 without a prefix (Figure 2). Earlier solved episodes remain available throughout, but they do not sustain performance as the current story's hints disappear. The dashed cold curves show each stage's target difficulty before the curriculum.

After accounting for each target's cold score, the withdrawal gap is 36.0–53.5 points across models (Table 2). Qwen 3.6 Plus retains a 12.57-point endpoint advantage from a 50.06-point average supported advantage, a gap of 37.49. Kimi K2 Thinking's large supported advantage ends in a negative endpoint contrast.

Explanation coverage accounts for most of the decline. Correctness contributes 71.7–84.3% of the first-to-final score decrease (Appendix A). This ordering holds even after normalizing by each component's maximum. Qwen 3.6 Plus loses 57.4% of the 70-point correctness scale versus 30.4% of the 30-point efficiency scale. Its hint-free questions leave more of the explanation unrecovered as well as being less economical.

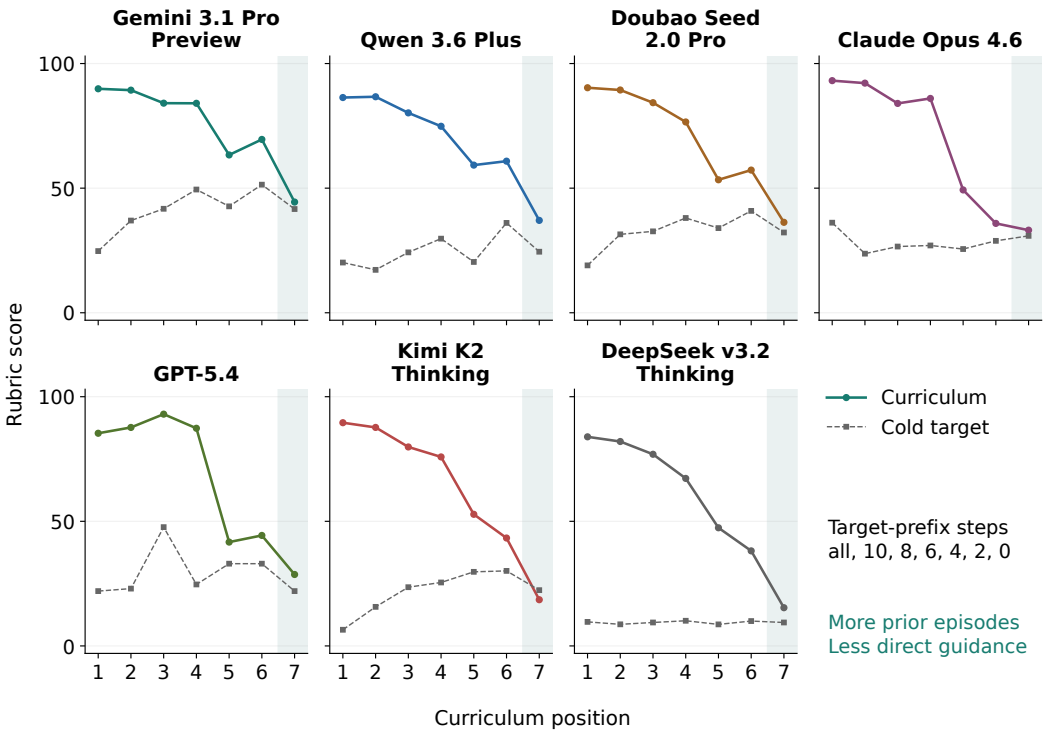


Figure 2: The path through the AHA-PUZZLE curriculum. Solid curves show mean episode scores and dashed curves show cold performance on the corresponding targets. Earlier solved episodes accumulate while the current target's prefix shrinks to zero. The shaded endpoint marks the transition to independent exploration; each curve follows the same set of curricula throughout.

The selected AHA-PUZZLE trajectories show how explanation coverage is lost through question choice and stopping. Repeating a rejected hypothesis uses the remaining budget without testing a new link. Stopping after one plausible twist leaves other links unchecked. Both decisions can be examined against the final explanation: which required links did the questions establish, and which did the agent assume? The transcripts and rubrics in Appendix G make that comparison explicit. They suggest concrete targets for learning from solved episodes: revise hypotheses after negative feedback and check the causal chain before ending the episode.

4.3 AHA-EULER: WORKED PROCEDURES OUTPERFORM ANSWER-ONLY TEACHING

With full teaching, all displayed models reach 80.0–100.0% accuracy on single-step AHA-EULER tasks (Table 3). Without the explanation and code, GPT-5.4 reaches 73.9%, Gemini 3.1 Pro Preview 60.0%, and Qwen 3.6 Plus 8.1%; DeepSeek v3.2 Thinking, Kimi K2 Thinking, and Hy3 Preview score zero. Qwen 3.6 Plus's fall from 97.3% with full teaching to 8.1% with partial teaching is particularly large.

Does the question–answer exchange help beyond what the model could already solve? On targets valid in all three conditions, GPT-5.4 rises from 56.5% cold accuracy to 73.9% with partial teaching and 95.7% with full teaching. Gemini 3.1 Pro Preview's corresponding rates are 37.5%, 62.5%, and 83.3%. For these models, sparse teaching adds to cold competence, while supplied code provides a further advantage. Appendix B reports this three-way comparison for every model.

Two answer-only traces show what reconstruction involves (Appendix G). On `DAG1_644`, GPT-5.4 constructs a dynamic program in Python, repairs a syntax error using tool feedback, and successfully adapts the computation to the held-out input. On `DAG1_696`, Qwen 3.6 Plus revises its matrix-polynomial program's state representation and truncation after receiving the teaching answer, yet returns an incorrect held-out result. Both agents write and revise substantial code; only

Table 3: Single-step AHA-EULER: a worked procedure and sparse teaching produce different held-out outcomes. Full teaching supplies an answer, explanation, and code; partial teaching supplies the question and answer. The procedure gap is full minus partial accuracy, with 95% bounds.

Model	Full (%)	Partial (%)	Procedure gap [95% bound]
GPT-5.4	95.7	73.9	21.7 [-11.1, 46.7]
Gemini 3.1 Pro Preview	80.0	60.0	20.0 [-10.3, 43.6]
Doubao Seed 2.0 Pro	86.4	22.7	63.6 [19.7, 84.8]
Qwen 3.6 Plus	97.3	8.1	89.2 [61.2, 97.5]
DeepSeek v3.2 Thinking	92.1	0.0	92.1 [65.7, 98.7]
Kimi K2 Thinking	100.0	0.0	100.0 [-33.1, 100.0]
Hy3 Preview	87.5	0.0	87.5 [0.3, 99.8]

the first reaches the correct new-input answer. Debugging an implementation and finding the right computation are separate obstacles.

The same pattern appears in composed computations. At each graph size, full and partial teaching share the requested tasks, with protocol failures counted as unsuccessful outcomes. GPT-5.4's partial-teaching accuracy is 25.0% at two nodes and 10.0% at four. Qwen 3.6 Plus scores zero under partial teaching at every multi-node size, although supplied code enables 80.0% at three nodes and 70.0% at five (Table 11). Each size has its own task set. The within-size contrasts locate the challenge: executing a composed procedure succeeds far more often than reconstructing it from answer-only teaching.

Reasoning and execution reliability. The shared-target rates describe valid attempts under the recorded error policy. Over all 50 requests per condition, counting excluded attempts as failures, Qwen 3.6 Plus achieves 94.0% full-teaching accuracy and 6.0% partial-teaching accuracy. GPT-5.4's partial rate becomes 34.0%. Appendix B gives this end-to-end view for every model. It complements the aligned reasoning comparison with the reliability of delivering a correct result.

These records identify where an improved method must help. A correct recurrence still needs an implementation that finishes within budget. A program that executes successfully can still implement the wrong computation. The answer-only traces show why both matter: tool feedback can repair code, while the held-out answer tests whether the repaired method transfers.

4.4 AHA-VENDING: FINAL PROFIT HIDES RECOVERY TIME AND EARLY STOPS

Table 4 reports incident-setting outcomes. GLM 5.1, Claude Opus 4.6, and Gemini 3.1 Pro finish all five seeds with high mean profit. Hy3 Preview earns a positive mean while completing only one seed; the remaining runs stop under the procurement watchdog. GPT-5.4 completes every seed with a smaller profit margin. DeepSeek v3.2 Thinking and Doubao Seed 2.0 Pro terminate at approximately the bankruptcy floor. A profitable endpoint and complete operation are thus distinct operational outcomes.

The nominal trajectories show when an operating policy becomes profitable and whether it keeps earning (Figure 3). Daily profit and early/late rates distinguish sustained accumulation from late recovery, even when terminal balances are similar. We report these everyday operating paths separately from resilience to disruption in the incident setting.

Table 5 quantifies these paths. Doubao Seed 2.0 Pro finishes at +$495 on average, yet its time-averaged profit is -$10: recovery arrives late enough that a profitable endpoint masks an extended period below the starting value. Kimi K2.6's earning rate falls from $5.89 to $2.86 per day, while GPT-5.4's rises from $5.69 to $7.67. GPT-5.4 also has a smaller mean drawdown, $125 versus $728 for Kimi K2.6. The comparison identifies three concrete priorities for operational agents: recover sooner, repeat productive actions, and control losses throughout the horizon.

The action records identify two operational problems. Doubao Seed 2.0 Pro repeatedly attempts to buy or stock products that do not fit the machine's small slots. Three runs later adopt workable stocking actions and recover; two reach bankruptcy (Figure 4). Its positive final mean combines

Table 4: AHA-VENDING incident setting: five seeds per row, with Student-t 95% intervals. Completion, mean days, and the modal termination reason accompany dollar profit.

Model	Profit ($)	95% interval ($)	Complete	Days	Termination
GLM 5.1	+2884	[+2436, +3332]	5/5	365.0	completed
Claude Opus 4.6	+2834	[+824, +4844]	5/5	365.0	completed
Gemini 3.1 Pro	+2433	[+1484, +3383]	5/5	365.0	completed
Hy3 Preview	+1555	[+670, +2441]	1/5	288.0	no-order stop
Kimi K2.6	+747	[-287, +1782]	5/5	365.0	completed
GPT-5.4	+338	[-48, +724]	5/5	365.0	completed
Qwen3 Max	-427	[-553, -300]	2/5	246.6	bankruptcy
GPT-4o	-462	[-490, -435]	0/5	150.0	bankruptcy
DeepSeek v3.2 Thinking	-499	[-500, -498]	0/5	128.4	bankruptcy
Doubao Seed 2.0 Pro	-499	[-500, -499]	0/5	161.6	bankruptcy

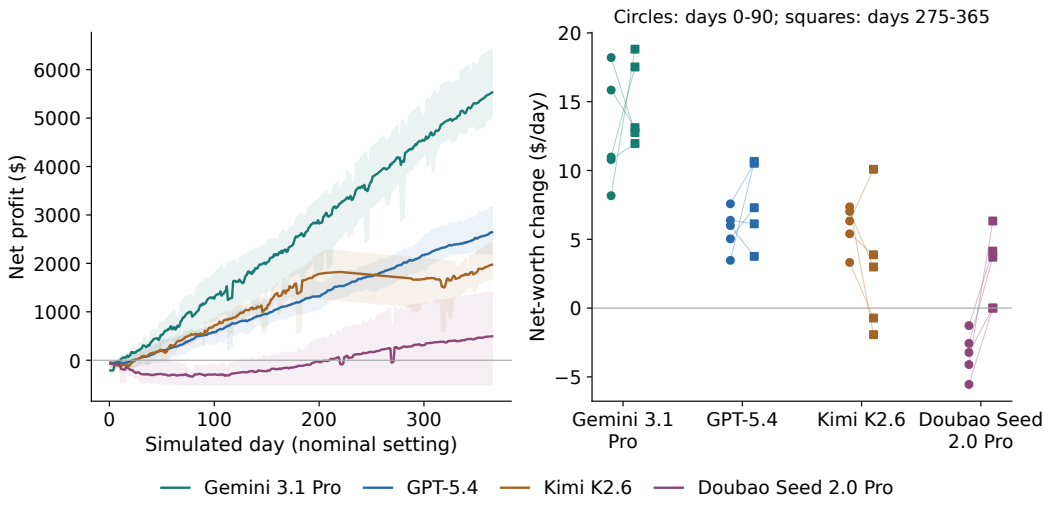


Figure 3: Nominal-setting temporal outcomes, complementary to the incident endpoints. Left: mean profit and seed min–max envelopes. Right: every seed's early and late 90-day earning rates; lines connect the same seed. Profit is fixed after termination for the accounting display. Changes reflect the joint evolution of policy and environment.

Table 5: Nominal AHA-VENDING process metrics, averaged across retained seeds. Final and time-averaged profit and drawdown are dollars; early and late rates are dollars/day. Each rate uses all five seeds with terminal-value carry-forward.

Model	Final	Time-avg.	Drawdown	Early	Late
Gemini 3.1 Pro	+5530	+2634	742	+12.80	+14.84
GPT-5.4	+2643	+1237	125	+5.69	+7.67
Kimi K2.6	+1972	+1192	728	+5.89	+2.86
Doubao Seed 2.0 Pro	+495	-10	702	-3.34	+2.83

these different paths. For Kimi K2.6, more waiting and less stocking accompany the later decline in earning rate: the problem is maintaining a productive routine.

Scripted policies show what predetermined routines achieve through the same interfaces (Appendix H). A fixed restocking policy completes the nominal year but loses $256.80 on average. A calibrated policy earns $1,364.94 yet operates for only 101 days on average before a no-order stop; under incidents, its means are $468.05 and 73.2 days. These five-seed summaries reinforce the

operational target: coordinate profitable stocking with procurement and cash collection throughout the year.

5 What Makes Experience Useful?

Compare methods by the capability they aim to improve. Qwen 3.6 Plus benefits from solved AHA-PUZZLE episodes yet struggles with answer-only AHA-EULER teaching. A single suite ranking would obscure where a learning method helps. Component scores instead let a memory method target later exploration, a program-learning method target new-input computation, or an operational method target reliable procurement and stocking.

Learning to generate useful insights. An extractor could turn interaction H into a rule $z = g(H)$ for a fresh executor on related targets. Length-matched irrelevant or incorrect rules would test which content helps; counterexamples would test revision. Appendix D specifies this proposed extension, holding executor conditions fixed to evaluate the extracted lesson rather than an entire history.

For AHA-VENDING, this test would save a business state before stocking and provide an extracted slot-fit rule to one executor condition. Both conditions would start with the same inventory, cash, pending orders, and future demand stream. Successful stocking and subsequent profit would measure the rule's value from that shared starting point. AHA-PUZZLE can hold the target story fixed, and AHA-EULER can hold the test input fixed. These proposed interventions would test whether a particular extracted lesson improves later decisions, building on the present records of behavior under accumulated experience.

6 Related Work

Learning and reusing experience. Continual learning studies sequential transfer and retention (Parisi et al., 2019; Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017); meta-learning and test-time training study rapid adaptation (Finn et al., 2017; Vinyals et al., 2016; Sun et al., 2020). In-context learning supplies another update interface (Brown et al., 2020; Min et al., 2022). Reflexion retains feedback, Voyager builds skills, and MemGPT manages memory (Shinn et al., 2023; Wang et al., 2024; Packer et al., 2023). LongMemEval tests memory across conversations (Wu et al., 2025). AHABENCH evaluates these interfaces through downstream capability.

Evaluating learning across tasks. AgentCL constructs streams with reusable sub-solutions; PATH-Bench varies helpful and interfering histories to probe transfer and retention (Shu et al., 2026; Yang et al., 2026). AHABENCH adds complementary information and time axes: withdrawn trace support, teaching with or without a worked procedure, and delayed consequences. Intermediate profiles show where competence depends on guidance and when productive behavior persists.

Interactive and executable benchmarks. AgentBench, WebArena, GAIA, and OSWorld evaluate diverse interaction interfaces (Liu et al., 2024; Zhou et al., 2024; Mialon et al., 2024; Xie et al., 2024). Component lineages include lateral reasoning in BRAINTEASER, LatEval, and SPLAT (Jiang et al., 2023; Huang et al., 2024; Chen et al., 2024), executable reasoning in PAL and VeRA (Gao et al., 2023; Cheng et al., 2026), and business coherence in Vending-Bench (Backlund & Petersson, 2025). Following multi-dimensional evaluation in HELM (Liang et al., 2023), AHABENCH retains native outcomes within an experience-to-capability protocol. Appendix I develops these connections.

7 Conclusion

AHABENCH follows experience into a new puzzle, a new computational input, or a later operating day. The evaluated agents show some gains, but independent exploration, method reconstruction, and sustained operation remain challenging. Paired comparisons and temporal records locate these difficulties, providing concrete targets for learning methods and tests of whether their improvements last.

AI USAGE STATEMENT

Language models served as evaluated agents and as the AHA-PUZZLE host and rubric executor described in Section 2. **With human in the loop for supervision and quality control**, coding agents assisted with writing, better result presentation, and experiment execution. **All research ideas, structure of the manuscript, and result analysis strategies originated with the human authors**, who have reviewed and revised the codebase and the manuscript, and human authors take full responsibility for all the scientific claims and artifacts.

REPRODUCIBILITY STATEMENT

Benchmark code and data are available at `https://anonymous.4open.science/r/AhaBench-ICLR2027-C332`. The repository provides AHA-PUZZLE tasks, curriculum specifications, and scoring rubrics; AHA-EULER generators and exact validators; and the AHA-VENDING simulator, configurations, and evaluation interfaces. Its README documents installation, API configuration, and fresh evaluations. The manuscript source archive supplies the analysis tables, curve inputs, and figure-generation scripts. Appendices A and B present further AHA-PUZZLE and AHA-EULER results; Appendix C specifies scoring, weighting, and uncertainty.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.15661.

Axel Backlund and Lukas Petersson. Vending-Bench: A Benchmark for Long-Term Coherence of Autonomous Agents. *arXiv preprint arXiv:2502.15840*, 2025. URL https://arxiv.org/abs/2502.15840.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.14165.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, et al. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *arXiv preprint arXiv:2410.07095*, 2024. URL https://arxiv.org/abs/2410.07095.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1812.00420.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Qi Chen, Bowen Zhang, Gang Wang, and Qi Wu. Weak-eval-Strong: Evaluating and Eliciting Lateral Thinking of LLMs with Situation Puzzles. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2410.06733.

Zerui Cheng, Jiashuo Liu, Chunjie Wu, Jianzhu Yao, Pramod Viswanath, Ge Zhang, et al. VeRA: Verified Reasoning Data Augmentation at Scale. *arXiv preprint arXiv:2602.13217*, 2026. URL https://arxiv.org/abs/2602.13217.

François Chollet. On the Measure of Intelligence. *arXiv preprint arXiv:1911.01547*, 2019. URL https://arxiv.org/abs/1911.01547.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, 2017. URL https://arxiv.org/abs/1703.03400.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, et al. PAL: Program-aided Language Models. In *Proceedings of the International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2211.10435.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2208.01066.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, et al. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, et al. Measuring Mathematical Problem Solving With the MATH Dataset. In *NeurIPS Datasets and Benchmarks*, 2021b. URL https://arxiv.org/abs/2103.03874.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, et al. LatEval: An Interactive LLMs Evaluation Benchmark with Incomplete Information from Lateral Thinking Puzzles. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024. URL https://aclanthology.org/2024.lrec-main.889/.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. BRAINTEASER: Lateral Thinking Puzzles for Large Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2310.05057.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, et al. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770*, 2023. URL https://arxiv.org/abs/2310.06770.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research*, 2022. URL https://arxiv.org/abs/2012.13490.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://arxiv.org/abs/1612.00796.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, et al. Solving Quantitative Reasoning Problems with Language Models. *arXiv preprint arXiv:2206.14858*, 2022. URL https://arxiv.org/abs/2206.14858.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, et al. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2211.09110.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, et al. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.03688.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.08840.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, et al. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for General AI Assistants. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.12983.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, et al. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://arxiv.org/abs/2202.12837.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. URL https://arxiv.org/abs/2112.09332.

Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms. *arXiv preprint arXiv:1803.02999*, 2018. URL https://arxiv.org/abs/1803.02999.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, et al. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. doi: 10.1016/j.neunet.2019.01.012. URL https://arxiv.org/abs/1802.07569.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. *arXiv preprint arXiv:2304.03442*, 2023. URL https://arxiv.org/abs/2304.03442.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large Language Model Connected with Massive APIs. *arXiv preprint arXiv:2305.15334*, 2023. URL https://arxiv.org/abs/2305.15334.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.16789.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. URL https://arxiv.org/abs/1611.07725.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1811.11682.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, et al. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*, 2016. URL https://arxiv.org/abs/1606.04671.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, et al. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv preprint arXiv:2302.04761*, 2023. URL https://arxiv.org/abs/2302.04761.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Yiheng Shu, Bernal Jiménez Gutiérrez, Saisri Padmaja Jonnalagedda, Yuguang Yao, Huan Sun, and Yu Su. AgentCL: Toward Rigorous Evaluation of Continual Learning in Language Agents. *arXiv preprint arXiv:2606.02461*, 2026. URL https://arxiv.org/abs/2606.02461.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1703.05175.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. URL https://arxiv.org/abs/2206.04615.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1909.13231.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, et al. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. *arXiv preprint arXiv:2210.09261*, 2022. URL https://arxiv.org/abs/2210.09261.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *arXiv preprint arXiv:1606.04080*, 2016. URL https://arxiv.org/abs/1606.04080.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, et al. Transformers learn in-context by gradient descent. In *Proceedings of the International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2212.07677.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.10726.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, et al. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2305.16291.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, et al. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*, 2022. URL https://arxiv.org/abs/2201.11903.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.02080.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, et al. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. *arXiv preprint arXiv:2404.07972*, 2024. URL https://arxiv.org/abs/2404.07972.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic Memory for LLM Agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Xidong Yang, Xingyi Zhang, Wenhao Li, Wenyan Liu, Junjie Sheng, Yun Hua, et al. PATH-Bench: Path-Dependent Evaluation of Lifelong Agents. *arXiv preprint arXiv:2608.01149*, 2026. URL https://arxiv.org/abs/2608.01149.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, et al. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, et al. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning. *arXiv preprint arXiv:2203.14465*, 2022. URL https://arxiv.org/abs/2203.14465.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. In *International Conference on Machine Learning*, 2017. URL https://arxiv.org/abs/1703.04200.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models. *arXiv preprint arXiv:2310.04406*, 2023. URL https://arxiv.org/abs/2310.04406.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

A ADDITIONAL AHA-PUZZLE RESULTS

The AHA-PUZZLE profile raises two questions: what deteriorates as trace support fades, and how broadly is the endpoint gain distributed across stories? The score decomposition separates explanation quality from question cost; the story and weighting comparisons show how each contributes to the overall gain.

A.1 THE DECLINE IS PRIMARILY IN EXPLANATION QUALITY

Correctness accounts for 71.7–84.3% of the score decline from the first curriculum position to the trace-free endpoint (Table 6). For Qwen 3.6 Plus, correctness falls from 65.64 to 25.45, while efficiency falls from 20.73 to 11.61. The larger change is in explanation completeness, with questioning efficiency also declining. This decomposition points to a concrete learning target: recover more of the hidden explanation with an effective questioning strategy.

Table 6: **Explanation quality accounts for most of the decline.** Mean scores at the first position (t1) and trace-free endpoint (t7) on complete curricula. Correctness is out of 70 and efficiency out of 30; the last column is correctness's share of the total score decline.

	Correctness		Efficiency		Correctness
Model	t1	t7	t1	t7	share (%)
Claude Opus 4.6	70.00	27.00	23.14	6.14	71.7
DeepSeek v3.2 Thinking	64.00	10.83	19.90	4.50	77.5
Doubao Seed 2.0 Pro	68.32	28.19	21.97	8.11	74.3
GPT-5.4	65.33	21.00	20.00	7.67	78.2
Gemini 3.1 Pro Preview	68.08	29.75	21.80	14.68	84.3
Kimi K2 Thinking	67.35	11.73	22.22	6.81	78.3
Qwen 3.6 Plus	65.64	25.45	20.73	11.61	81.5

A.2 DOES ONE STORY DRIVE THE GAIN?

Table 7 places the original all-stories gain beside the range obtained by excluding each story in turn, together with its languages and repeats. The agent outputs stay fixed; only the averaging set changes. Omitting a below-average-gain story raises the remaining mean, while omitting an above-average story lowers it. Qwen 3.6 Plus's original gain is +12.57 points, compared with 9.44–14.03 after one exclusion. Its positive gain therefore does not depend on a single story. Gemini 3.1 Pro Preview's range crosses zero. These are sensitivity ranges, not confidence intervals.

Table 7: **AHA-PUZZLE gain before and after excluding a story.** All stories reproduces the original gain in Table 1. The remaining columns give the smallest and largest gains after one story is omitted from aggregation. Values are rubric points; the same recorded outputs underlie every column.

		Exclude one story	
Model	All stories	Minimum gain	Maximum gain
Qwen 3.6 Plus	+12.57	+9.44	+14.03
GPT-5.4	+6.67	+4.00	+9.33
DeepSeek v3.2 Thinking	+5.93	+2.89	+7.61
Doubao Seed 2.0 Pro	+4.08	+2.34	+5.52
Gemini 3.1 Pro Preview	+2.80	-0.24	+4.29
Claude Opus 4.6	+2.29	+0.15	+4.67
Kimi K2 Thinking	-3.84	-8.15	-0.35

The direction also persists under alternative weighting. The primary score gives each episode equal weight. Averaging first within a language or story and then equally across those groups tests how aggregation affects the result. Every model retains the same gain direction under all three

choices (Table 8); the magnitude changes most for models whose performance differs substantially across stories or languages.

Table 8: **AHA-PUZZLE gain under three weighting choices.** Each column uses the same episodes, with equal weight per episode, language, or story. Values are rubric points.

Model	Per episode	Per language	Per story
Claude Opus 4.6	+2.29	+0.25	+6.04
DeepSeek v3.2 Thinking	+5.93	+5.93	+5.93
Doubao Seed 2.0 Pro	+4.08	+4.36	+4.55
GPT-5.4	+6.67	+6.67	+6.67
Gemini 3.1 Pro Preview	+2.80	+2.75	+3.04
Kimi K2 Thinking	-3.84	-2.44	-12.42
Qwen 3.6 Plus	+12.57	+12.49	+13.18

B ADDITIONAL AHA-EULER RESULTS

The AHA-EULER results separate three demands: solving directly, using a related answer, and adapting a worked procedure. The following comparisons examine those demands on common targets, then relate them to execution reliability, graph composition, and input hardening. All accuracy entries are percentages; differences are percentage points.

B.1 TEACHING DETAIL CHANGES HELD-OUT SUCCESS

Comparing all three conditions on the same targets shows how much each teaching signal contributes (Table 9). GPT-5.4 rises from 56.5% cold accuracy to 73.9% with answer-only teaching and 95.7% with full teaching. Gemini 3.1 Pro Preview gains 25.0 points from answer-only teaching. For DeepSeek v3.2 Thinking, Kimi K2 Thinking, and Hy3 Preview, the successful outcomes on this comparison set occur only with full teaching. The distinction is therefore both quantitative and procedural: a solved example and an explicit implementation support different levels of transfer.

Table 9: **Direct solving, sparse teaching, and worked procedures.** Exact-match accuracy on targets valid in all three conditions. Gain is answer-only minus cold accuracy.

Model	Cold (%)	Answer-only (%)	Full (%)	Gain (pp)
GPT-5.4	56.5	73.9	95.7	+17.4
Gemini 3.1 Pro Preview	37.5	62.5	83.3	+25.0
Doubao Seed 2.0 Pro	15.0	25.0	85.0	+10.0
Qwen 3.6 Plus	0.0	8.6	97.1	+8.6
DeepSeek v3.2 Thinking	0.0	0.0	94.6	+0.0
Kimi K2 Thinking	0.0	0.0	100.0	+0.0
Hy3 Preview	0.0	0.0	100.0	+0.0

B.2 SUCCESSFUL REASONING MUST SURVIVE EXECUTION

Table 10 distinguishes correctness among valid attempts from the chance that a requested evaluation produces a correct answer. End-to-end accuracy includes every request, counting unsuccessful completion as zero. GPT-5.4's answer-only accuracy changes from 73.9% to 34.0% under this all-request view, whereas Qwen 3.6 Plus's full-teaching accuracy changes from 95.9% to 94.0%. Reliable execution thus makes a model-specific contribution to the delivered result.

Table 10: **Correctness and delivery reliability.** All entries are accuracy (%). Valid-attempt rates use each condition's scored tasks; end-to-end rates include all 50 requests per condition. Table 9 instead holds the target set fixed across conditions.

	Valid-attempt accuracy			End-to-end accuracy	
Model	Cold	Full	Answer-only	Full	Answer-only
DeepSeek v3.2 Thinking	0.0	91.8	0.0	90.0	0.0
Doubao Seed 2.0 Pro	9.5	91.4	20.0	64.0	10.0
GPT-5.4	45.9	97.5	73.9	78.0	34.0
Gemini 3.1 Pro Preview	24.4	84.2	59.3	64.0	32.0
Hy3 Preview	0.0	78.6	0.0	44.0	0.0
Kimi K2 Thinking	0.0	100.0	0.0	36.0	0.0
Qwen 3.6 Plus	0.0	95.9	8.1	94.0	6.0

B.3 Recovering a Method Versus Executing Supplied Code

Table 11 compares partial and full teaching at every graph size. Partial teaching tests method reconstruction and transfer without reference code; full teaching supplies the implementation for adaptation and execution. Both panels use the same requested tasks within each model and node count, with protocol failures counted as unsuccessful outcomes. This extends the end-to-end view in Table 10 to composed computations.

The distinction persists beyond single-step tasks. With partial teaching, GPT-5.4 achieves 25.0% at two nodes and 10.0% at four; Gemini 3.1 Pro Preview achieves 10.0% at two nodes. The other displayed multi-node partial-teaching rates are zero. Full teaching enables substantially more successful computations, including Qwen 3.6 Plus's 80.0% at three nodes and 70.0% at five. Supplied-code success and method reconstruction therefore measure different capabilities. Each graph size has its own task set, so the columns describe composition profiles rather than a controlled depth-only intervention.

Table 11: **Method reconstruction and code-supported execution across graph sizes.** End-to-end exact-match accuracy (%). Partial teaching supplies the answer; full teaching also supplies an explanation and reference code. Both conditions use their common requested tasks at each node count, counting protocol failures as zero. Nodes are subcomputations linked by dependencies.

Partial teaching: reconstruct and apply a method					
Model	1 node	2 nodes	3 nodes	4 nodes	5 nodes
DeepSeek v3.2 Thinking	0.0	0.0	0.0	0.0	0.0
Doubao Seed 2.0 Pro	10.0	0.0	0.0	0.0	0.0
GPT-5.4	34.0	25.0	0.0	10.0	0.0
Gemini 3.1 Pro Preview	32.0	10.0	0.0	0.0	0.0
Hy3 Preview	0.0	0.0	0.0	0.0	0.0
Kimi K2 Thinking	0.0	0.0	0.0	0.0	0.0
Qwen 3.6 Plus	6.0	0.0	0.0	0.0	0.0

Full teaching: adapt and execute supplied code					
Model	1 node	2 nodes	3 nodes	4 nodes	5 nodes
DeepSeek v3.2 Thinking	90.0	90.0	50.0	40.0	40.0
Doubao Seed 2.0 Pro	64.0	40.0	30.0	60.0	40.0
GPT-5.4	78.0	62.5	40.0	10.0	20.0
Gemini 3.1 Pro Preview	64.0	60.0	50.0	60.0	0.0
Hy3 Preview	44.0	10.0	20.0	30.0	0.0
Kimi K2 Thinking	36.0	20.0	50.0	20.0	10.0
Qwen 3.6 Plus	94.0	50.0	80.0	30.0	70.0

B.4 Harder Inputs Challenge Cold Solving

Cold accuracy is 16.5% on original-inspired tasks and 8.7% on hardened tasks (Table 12). GPT-5.4, Gemini 3.1 Pro Preview, and Doubao Seed 2.0 Pro each show the same ordering. These distinct task sets characterize the challenge supplied by each construction, with generated inputs and exact validators specifying the computations.

Table 12: **Cold accuracy on original-inspired and hardened tasks.** Accuracy is in percent; the difference is original-inspired minus hardened. The pooled row weights every scored attempt equally.

Model	Original-inspired (%)	Hardened (%)	Difference (pp)
DeepSeek v3.2 Thinking	0.0	0.0	+0.0
Doubao Seed 2.0 Pro	19.0	0.0	+19.0
GPT-5.4	52.9	40.0	+12.9
Gemini 3.1 Pro Preview	33.3	15.0	+18.3
Hy3 Preview	0.0	0.0	+0.0
Kimi K2 Thinking	0.0	0.0	+0.0
Qwen 3.6 Plus	0.0	0.0	+0.0
Pooled accuracy	16.5	8.7	+7.9

C Protocol and Scoring Details

AHA-Puzzle information flow. The current story and within-episode transcript determine the tester's immediate interaction. In curriculum conditions, the current target prefix is inserted into that transcript, and compact reveal records from earlier episodes provide history. The host answers only the selected questions from the hidden explanation. A final explanation is graded against required and optional elements. In the *Waterweed* rubric, required links include a failed rescue, a misidentified object, a statement that the river has no waterweed, and the eventual realization about the object. This decomposition distinguishes recovering an explanation chain from recognizing a story title.

Scoring and evaluator evidence. Correctness has a maximum of 70 points. Efficiency has a maximum of 30, with full credit through five turns and a two-point reduction per additional turn, subject to the episode's scoring rules. The released rubrics and runner define their combination. Human checking covers the hidden solutions, allowed host responses, rubrics, and reference traces. In a sample of 100 graded trajectories, the final human grades agreed with the rubric executor. This audit measures final-grade concordance. Required explanation links, optional elements, and question counts remain explicit in the scoring record.

AHA-Euler exactness and execution. Each task includes generated inputs and a deterministic reference computation. The teaching instance exposes subcomputations in a dependency graph, and the held-out task changes the input. Python execution makes implementation and debugging part of the evaluated behavior. Reference procedures are visible in full teaching and withheld in partial teaching. Exact matching verifies the normalized output. Family familiarity can assist method selection, so the cold outcome and generated/hardened variants remain important context for interpreting transfer.

AHA-Vending persistence. The business state preserves inventory, cash, slots, pending orders, and environment time across actions. The next model request receives a current observation plus a short recent-transition window. This architecture makes environmental persistence central to the experience interface. Supplier delays can induce stockouts, poor slot-fit assumptions can trap cash, and excessive waiting can interrupt an otherwise profitable routine. The simulator records the financial and temporal consequences of those events.

AHA-Puzzle aggregation and uncertainty. Mean gain gives equal weight to scored cold/endpoint comparisons of the same story, language, and repeat. The 95% interval resamples whole stories with replacement, carrying their languages and repeats together, and recomputes the

episode-weighted mean. We enumerate all ordered story draws and take the 2.5th and 97.5th inverse empirical-CDF percentiles. The alternative means in Table 8 first average within language or story, then give those groups equal weight.

AHA-EULER scoring and uncertainty. Valid-attempt accuracy excludes server and access errors, connection failures, timeouts, budget exhaustion, exited processes, and malformed JSON. Condition contrasts use targets valid in every compared condition; end-to-end accuracy counts every requested evaluation. For the procedure-gap bound, let $[L_F, U_F]$ and $[L_A, U_A]$ be two-sided 97.5% Clopper–Pearson intervals for success under full teaching only and answer-only teaching only. The bound is $100[L_F - U_A, U_F - L_A]$, with at least 95% coverage under the independent-task multinomial model by the union bound.

Daily profit and seed variation. The daily profit curve uses the last observation of each day and carries the latest value through gaps or to day 365 after termination. Maximum drawdown includes the zero-profit starting point; early and late earning rates are $[p_s(90) - p_s(0)]/90$ and $[p_s(365) - p_s(275)]/90$. Survival and completion retain the actual stopping event. Incident-profit intervals use $\bar{p} \pm t_{.975,4} s/\sqrt{5}$ over the five seeds, while the trajectory envelopes show their minimum and maximum.

D EXTENDING AHABENCH TO EXPLICIT INSIGHT EXTRACTION

The experience-to-capability protocols also support a complementary question: which information in an interaction produces a useful insight? An explicit extractor turns the history into a reusable rule, allowing that rule to be evaluated as an object in its own right. In this extension, fix the executor, tools, action budget, and state for each held-out target. Compare the extracted rule with length-matched unrelated, shuffled, and incorrect content, pairing outcomes by target. These interventions isolate the rule's contribution within the broader behavioral evaluation of continual learning.

The extracted object differs by interface. AHA-PUZZLE calls for a questioning heuristic, such as testing whether a salient object was misidentified. AHA-EULER calls for a computational rule that produces the answer at a new input. AHA-VENDING calls for an operational rule, such as matching product dimensions to available slots. For this extension, the final AHA-PUZZLE identity, held-out AHA-EULER input, or saved AHA-VENDING state supplies a common test. A common future random-number stream keeps AHA-VENDING demand aligned across rule conditions.

Three quantities characterize an extractor: downstream success, the number of experiences needed to produce a useful rule, and the rule's memory cost. The trajectories expose how the rule is used through question selection, algorithm construction, and repeated business actions. Counterexamples add a revision test: an effective extractor must update an overgeneralized rule while retaining the part that remains useful. These objectives connect insight generation to the capabilities measured throughout the suite.

E SEED-LEVEL OPERATIONAL OUTCOMES

Figure 4 resolves the seed envelope in the main trajectory plot into individual paths. Recovery begins at different times and reaches different final values, while the bankrupt runs remain at the floor. The distinction is operationally important: a positive mean can coexist with prolonged losses and repeated terminal failure. The seed-level paths connect the endpoint and temporal summaries to the behavior they summarize.

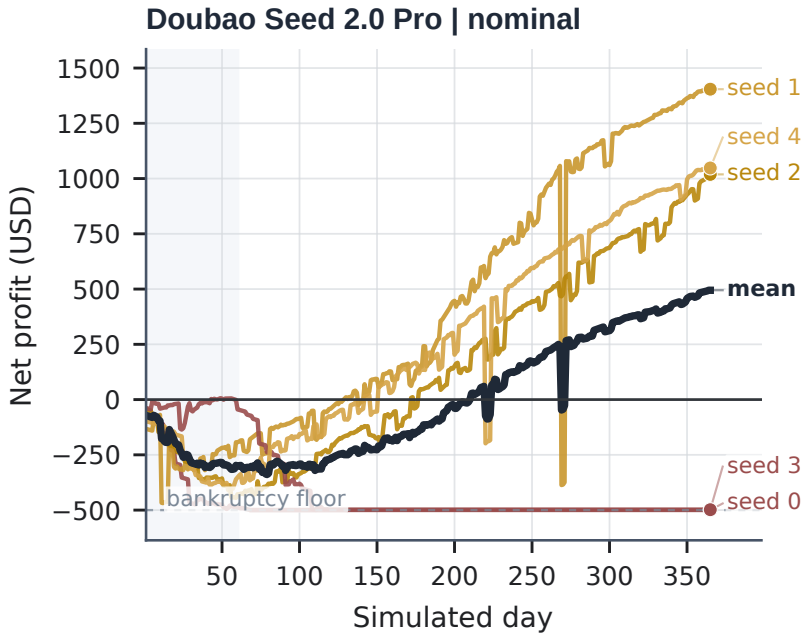


Figure 4: **Late recovery and bankruptcy coexist within one policy.** Doubao Seed 2.0 Pro's five nominal-setting runs separate into three recovering trajectories and two bankrupt trajectories. The dark curve averages all five seeds, carrying terminated outcomes forward to day 365. The mean endpoint of +$495 combines these different paths, explaining why time-averaged profit and survival add information beyond the final balance.

F TEACHING AND SIMULATOR DIAGNOSTIC VIEWS

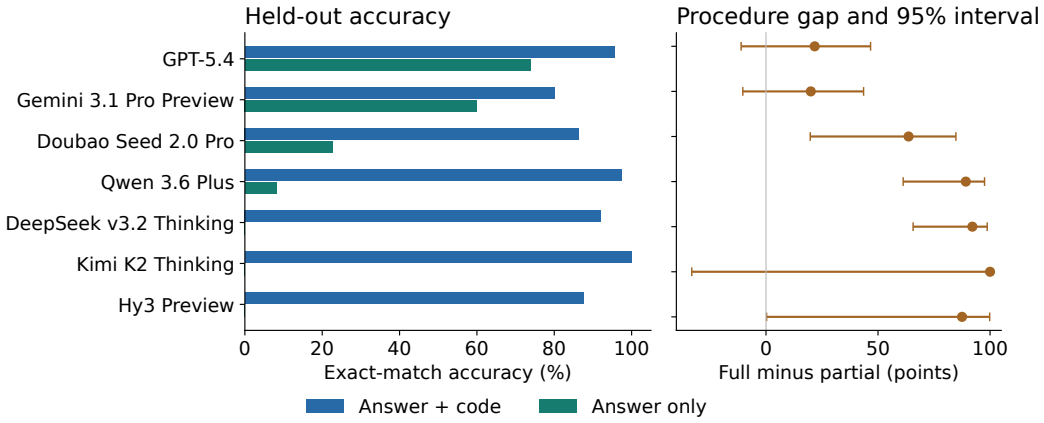


Figure 5: **Worked procedures and answer-only teaching.** The bars show held-out accuracy under each teaching signal; error bars show 95% intervals for their difference. An exposed procedure can be adapted and executed; answer-only teaching requires deriving or recovering a computation from the problem and its output.

Figure 5 separates two demands on experience use. The same-target contrast measures the difference in held-out accuracy when the teaching signal exposes or withholds the procedure. Tables 9 and 10 connect that contrast to cold performance and the reliability of completing a requested task.

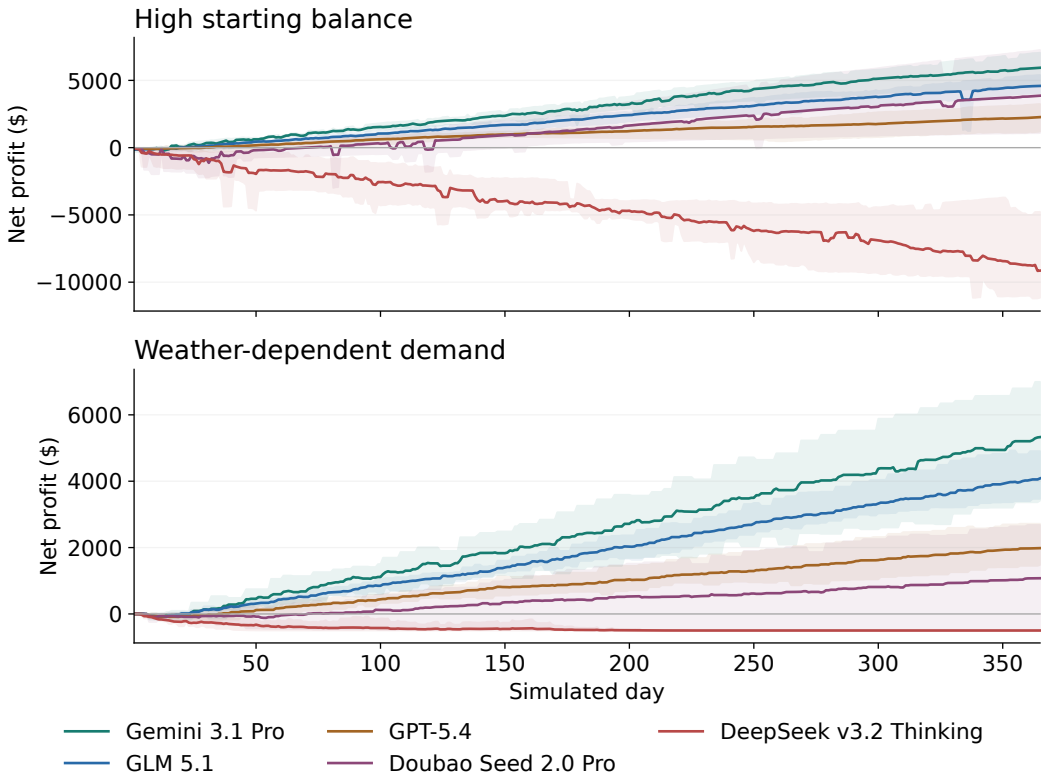
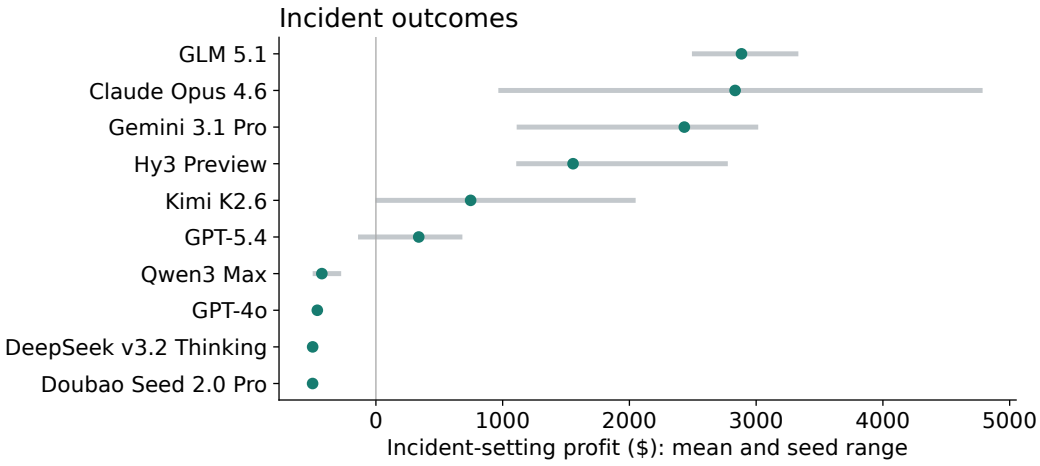


Figure 6: **Simulator conditions change the shape and scale of outcomes.** High-balance and weather diagnostic trajectories. Lines average five seeds; shading spans their range, with terminated paths carried forward. A larger starting balance permits prolonged losses before insolvency. Weather-dependent demand changes the operational environment. These trajectories describe performance in their respective settings.

Figure 6 shows why the incident setting and nominal temporal analysis provide complementary evidence. Capital constraints can truncate a failing policy, while demand variation can alter the value of an inventory and pricing routine. Reading trajectories across settings exposes these operational sensitivities without treating a change of simulator conditions as a learning intervention.



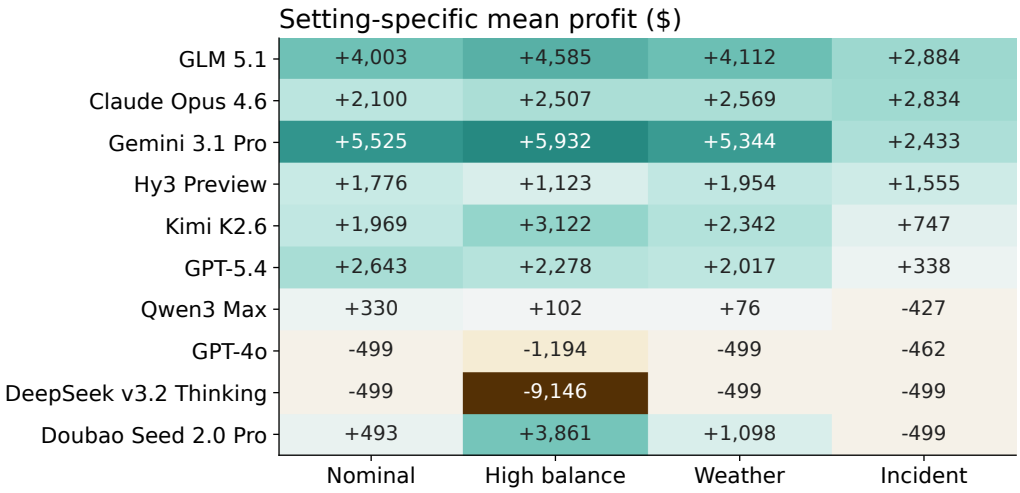


Figure 7: **Incident profit and the cross-setting outcome profile.** Top: incident-setting means with seed min–max ranges; the main table instead reports Student-t intervals for the mean. Bottom: mean dollar profit in four single-agent settings, from the diagnostic summary records. Qwen3 Max and Kimi K2.6 are the AHA-VENDING endpoints. The matrix describes setting sensitivity rather than a common before–after learning score.

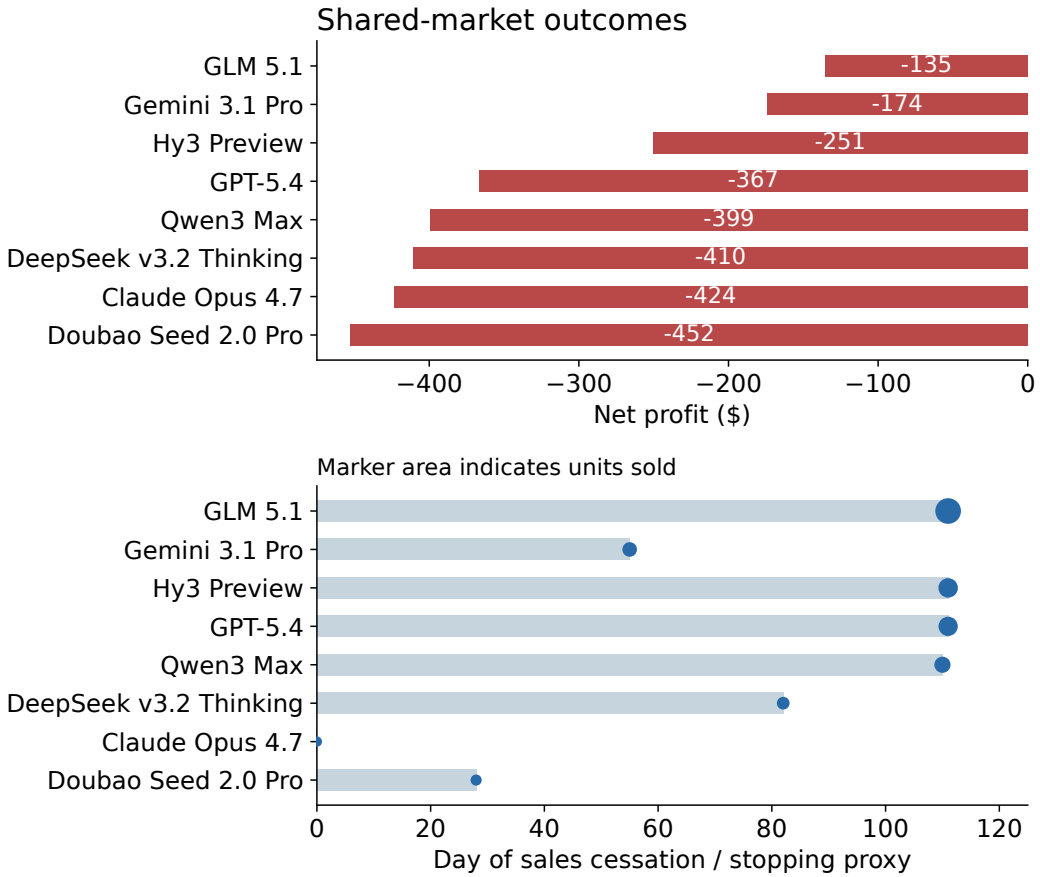


Figure 8: **The shared market presents a different operational regime.** The coupled-market run contains one seed and eight agents. All eight lose money; the companion panel distinguishes their sales-cessation or stopping proxies and sales volumes. Endpoint identifiers identify Claude Opus 4.7 and Qwen3 Max in this run. These descriptive outcomes concern joint demand and peer interaction, separately from the five-seed single-agent evaluation.

The arena changes the market faced by each policy as well as the presence of other agents. Its outcomes therefore characterize this coupled setting. Comparing a focal agent alone under the same demand rules, then with fixed scripted peers, would separate market scarcity from sensitivity to peer decisions. The observed profit and sales profiles preserve the operational evidence motivating that comparison.

G TRAJECTORY CASES AND FAILURE DIAGNOSIS

G.1 RECONSTRUCTION AND REPAIR UNDER ANSWER-ONLY TEACHING

The inspected GPT-5.4 trajectory on `DAG1_644` illustrates how a correct held-out answer is reached. The model receives the teaching question and answer, proposes a dynamic program, writes Python, repairs a syntax problem after tool feedback, and changes the parameters for the held-out instance. The reference implementation is withheld in this condition, so the trace documents an executable attempt assembled within the run.

The inspected Qwen 3.6 Plus trajectory on `DAG1_696` illustrates a substantive failure. The model writes matrix-polynomial code and obtains an incorrect teaching prediction. After the correct teaching answer is revealed, it changes the state representation and truncation choices, then returns another incorrect value for the held-out instance. The trace distinguishes an attempted computational solution from a refusal or a formatting-only failure. It also shows why receiving a correct example answer can leave algorithm selection unresolved.

Table 13: Selected answer-only trace cases connecting final exact-match outcomes to algorithm choice, implementation, and tool-feedback repair.

Model	Task	Observed behavior
GPT-5.4	`DAG1_644`	Reconstructs a dynamic program, repairs code, and reparameterizes successfully.
GPT-5.4	`DAG1_962`	Uses repeated tool debugging before a correct final output.
Gemini 3.1 Pro	`DAG1_696`	Transfers a combinatorial recurrence to the new assignment.
Qwen 3.6 Plus	`DAG1_696`	Revises substantial code after feedback but remains incorrect.
DeepSeek v3.2 Thinking	`DAG1_923H`	Repairs execution errors while leaving the algorithm incorrect.
Kimi K2 Thinking	`DAG1_970`	Attempts a scaled computation and returns an incorrect extrapolation.

These cases separate three aspects of an answer-only attempt: choosing a computation, implementing it, and adapting it to the evaluation input. Tool feedback exposes execution errors, while checking the computed value tests the chosen algorithm. A final exact-match score summarizes their joint success. The generated program and feedback identify which step needs improvement.

G.2 AHA-PUZZLE QUESTION SELECTION AND STOPPING

An AHA-PUZZLE episode can fail before the final explanation is written. A model may commit to a literal reading of the opening story, repeat variants of a hypothesis after negative answers, or delay a question about identity or chronology. The hidden explanation often links several facts, so finding a plausible twist still leaves the task of checking the complete chain.

The rubric makes that distinction concrete. In the Waterweed example, the required links include a failed rescue, a mistaken identification of what was touched, the fisherman's statement about the river, and the resulting realization. An audit therefore reads the order of questions together with the final explanation. It asks which links were established, which were assumed, and whether stopping left a required link unresolved. The correctness and efficiency components then describe explanation coverage and the interaction cost of reaching it.

G.3 OPERATIONAL RECOVERY AND LOSS OF PERSISTENCE

Doubao Seed 2.0 Pro's nominal runs provide a recovery case with meaningful seed variation. The early behavior includes repeated attempts to buy or stock products that do not fit the machine's small slots. Inventory can therefore absorb cash before it generates sales. Some runs eventually adopt workable stocking actions and recover, while others reach bankruptcy. The individual paths in Figure 4 connect the positive final mean to these distinct operational outcomes.

Kimi K2.6's nominal behavior illustrates a different pattern: profitable early operation followed by more waiting and less stocking. Its mean earning rate falls from $5.89 per day in the early window to $2.86 in the late window. The trajectory exposes persistence as a concrete operating requirement:

an agent that has achieved profitable operation must continue coordinating procurement, stocking, and cash collection across later business states. The action history shows where activity declines, while the daily financial record measures the accompanying loss of earning rate.

Table 14: Failure signatures across the three experience interfaces. Each entry identifies an observable event and its consequence for the evaluated task.

Interface	Signature	Consequence
AHA-PUZZLE	Repeated rejected hypothesis	Questions are spent without resolving new hidden-state links.
AHA-PUZZLE	Premature explanation	An efficient-looking stop leaves required rubric elements uncovered.
AHA-EULER	Incorrect recurrence or representation	Executable code solves a different computation.
AHA-EULER	Incomplete feedback repair	A runtime issue is fixed while the mathematical error persists.
AHA-VENDING	Slot-fit errors	Cash is committed to inventory that cannot be sold in the available slots.
AHA-VENDING	Delayed reordering or collection	Stockouts or cash shortages interrupt a profitable routine.
AHA-VENDING	Persistent waiting	Business activity and earning rate decline later in the run.
AHA-VENDING	Supplier or incident mishandling	Delays, failed procurement, or shocks compound over time.

H SCRIPTED OPERATIONAL BASELINES

Three scripted policies provide interpretable reference behaviors through the same observation and action interfaces as the model agents. The fixed policy restocks a small product set; the conservative policy uses hand-coded operating rules; and the calibrated policy encodes a stronger routine developed from prior simulator traces. Their outcomes reveal which aspects of profitable, sustained operation a predetermined routine can achieve.

Table 15: Five-seed scripted-policy summaries. Dollar profit, simulated days, and units sold are means. A no-order stop is triggered by 20 days without a successful order. Nominal and incident are distinct simulator conditions.

Policy	Setting	Days	Profit ($)	Units	Termination
Fixed	Nominal	365.0	-256.80	502	Horizon
Fixed	Incident	50.0	-86.82	36	No-order stop
Conservative	Nominal	33.8	+17.50	74	No-order stop
Conservative	Incident	34.0	-25.86	50	No-order stop
Calibrated	Nominal	101.0	+1364.94	1198	No-order stop
Calibrated	Incident	73.2	+468.05	548	No-order stop

The fixed policy completes the nominal year with a loss, while the calibrated policy earns a positive profit but stops early. These outcomes separate horizon completion from profitability even for deterministic rule policies. The incident setting further reduces the calibrated policy's mean duration and profit. Their value as baselines is this interpretable behavior: they establish what a simple operating routine achieves under the same termination rules, while the recorded model trajectories expose how more flexible policies depart from those routines.

I LEARNING INTERFACES AND BENCHMARK CONTEXT

Different mechanisms, a common behavioral question. Continual-learning methods preserve useful experience through different representations. Elastic weight consolidation and synaptic intelligence regularize parameter changes (Kirkpatrick et al., 2017; Zenke et al., 2017); gradient episodic

memory uses stored examples to constrain updates (Lopez-Paz & Ranzato, 2017); progressive networks and iCaRL retain structure or exemplars across tasks (Rusu et al., 2016; Rebuffi et al., 2017). A-GEM explicitly considers acquisition speed alongside memory and computational cost (Chaudhry et al., 2019). Experience replay provides another route to retention in sequential reinforcement learning (Rolnick et al., 2019), whose changing tasks and environments are surveyed by Khetarpal et al. (2022). These mechanisms motivate the update interface in Section 2: the benchmark specifies the experience and later test, while the agent supplies the learning mechanism. The reported fixed-weight agents instantiate that interface through their interaction histories.

Acquiring an ability from limited experience. Matching and prototypical networks learn representations suited to few-shot classification (Vinyals et al., 2016; Snell et al., 2017); MAML and first-order meta-learning optimize initializations for rapid task adaptation (Finn et al., 2017; Nichol et al., 2018). Test-time training and Tent adapt predictive models using signals available at deployment (Sun et al., 2020; Wang et al., 2021). Across these settings, the informative comparison concerns what additional experience makes possible and at what cost. AHABENCH brings that question to interactive agents: a worked episode may supply a questioning strategy, a mathematical method, or an operating rule. Its intermediate profiles record performance as the amount and form of support change, connecting acquisition efficiency to the ability ultimately exercised.

Learning through context. In-context learning exposes examples through the model's input (Brown et al., 2020), and demonstration analyses distinguish the contributions of formatting, label space, and input distribution (Min et al., 2022). Bayesian accounts study inference of a latent concept shared across examples (Xie et al., 2022); regression experiments examine which functions and learning algorithms transformers can implement in context (Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023). These studies make the relation among examples a central design variable. In AHABENCH, that relation is instantiated behaviorally: prior puzzles demonstrate exploration moves, taught AHA-EULER instances share a computation with the test instance, and operational feedback concerns recurring business constraints. The learning interface thus supplies a concrete test of whether useful relationships carry across changed inputs or support conditions.

From a stored episode to a usable rule. Reflexion stores verbal feedback, Self-Refine iterates on an output using self-feedback, and Voyager accumulates executable skills (Shinn et al., 2023; Madaan et al., 2023; Wang et al., 2024). Generative Agents and MemGPT further illustrate retrieval and management of persistent experience (Park et al., 2023; Packer et al., 2023). These representations differ in how directly they specify an action. A solved AHA-PUZZLE trace contains both a particular story and a questioning strategy; an AHA-EULER program contains both input-specific constants and a reusable computation. Withdrawing the trace or changing the input asks whether the agent can carry the useful relationship across that boundary. In AHA-VENDING, the corresponding test is repeated application under changing inventory, cash, and supplier state.

A-MEM organizes experience through linked, evolving notes (Xu et al., 2025), while LongMemEval evaluates retrieval, temporal reasoning, and knowledge updates across sustained conversations (Wu et al., 2025). Together, these works distinguish managing a memory from acting successfully with its contents. AHABENCH focuses on downstream usefulness: a retained fact or summary earns value through a better question, a correct computation, or a productive decision. This connects memory design to the scope and persistence of the capability it supports.

Task success and experience reuse. WebShop and ALFWorld ground agent actions in interactive environments (Yao et al., 2022; Shridhar et al., 2021); SWE-bench and τ-bench evaluate software repair and tool–user interaction (Jimenez et al., 2023; Yao et al., 2024). WebGPT learns evidence-backed browsing, and MLE-bench evaluates extended machine-learning engineering workflows (Nakano et al., 2021; Chan et al., 2024). These interfaces make action sequences and outcomes observable. AgentCL and PATH-Bench place relationships among successive tasks at the center of learning evaluation (Shu et al., 2026; Yang et al., 2026). AHABENCH adds a complementary view of the information supplied along that sequence: progressively withdrawn demonstrations, different levels of procedural disclosure, and delayed operational consequences.

Reasoning during an episode and reuse across episodes. Chain-of-thought prompting exposes intermediate reasoning, while self-consistency aggregates candidate solutions (Wei et al., 2022;

Wang et al., 2023). Tree of Thoughts and Language Agent Tree Search explore alternative reasoning or action trajectories (Yao et al., 2023a; Zhou et al., 2023); STaR trains on successful generated rationales (Zelikman et al., 2022). These methods act at different points in the experience-to-capability pathway: producing a candidate, selecting a successful route, or incorporating that route into future behavior. AHABENCH makes the subsequent test explicit. The central empirical question is how effectively the resulting procedure, lesson, or strategy serves another instance once its original support changes.

Tools make insights executable. GSM8K, MATH, and code-generation benchmarks connect reasoning to concrete answer and execution checks (Cobbe et al., 2021; Hendrycks et al., 2021b; Chen et al., 2021). Minerva and BIG-Bench Hard further examine quantitative and multi-step reasoning (Lewkowycz et al., 2022; Suzgun et al., 2022). ReAct, Toolformer, ToolLLM, and Gorilla connect language reasoning to external actions and APIs (Yao et al., 2023b; Schick et al., 2023; Qin et al., 2024; Patil et al., 2023). An agent can consequently express an insight as a query policy, a callable computation, or an operational constraint. The three components evaluate those expressions through explanation coverage, exact output validation, and business outcomes. Keeping these native measures preserves the practical meaning of a gain: discovering additional hidden links, solving a new numerical instance, or maintaining a productive routine. This is also why a process profile is useful beside an endpoint: it reveals when the capability is available and which support conditions sustain it.

Breadth, exploration, and acquisition. MMLU and BIG-bench establish broad task coverage (Hendrycks et al., 2021a; Srivastava et al., 2022); the Abstraction and Reasoning Corpus foregrounds skill acquisition and generalization from limited examples (Chollet, 2019). For hidden-state exploration, BRAINTEASER, LatEval, and SPLAT examine lateral reasoning through different question and interaction formats (Jiang et al., 2023; Huang et al., 2024; Chen et al., 2024). AHABENCH connects this exploration interface to a curriculum and a support-withdrawal profile. Its three components collectively ask what experience enables, how much assistance remains necessary, and how long useful behavior persists. Those questions guide both the native outcomes and the process-level views reported in the paper.