LEARN2PLAY BENCH: HOW WELL DO LLM AGENTS LEARN FROM EXPERIENCE IN UNFAMILIAR ENVIRONMENTS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from experience is essential for LLM agents to adapt to unfamiliar and dynamic environments. Evaluating this ability is therefore important for understanding how effectively agents acquire and use new knowledge. Existing benchmarks have sought to evaluate this ability, but they primarily evaluate tasks whose rules are provided in the instructions or already familiar to pretrained models, making it difficult to distinguish learning from interactions from reasoning with existing knowledge. To address this, we introduce **Learn2Play Bench**, a benchmark of newly designed text-based games, whose rules are novel or counter-intuitive, requiring agents to acquire knowledge through interaction rather than rely solely on pretrained knowledge. These games provide reproducible feedback and automatic scoring, enabling controlled evaluation of learning across repeated attempts. We also vary game instances to test whether agents can apply what they have learned to new situations. Therefore, we evaluate how backbone models, self-evolving methods, and agent harnesses affect agents' learning ability, revealing three findings: (1) **Experience retention:** Retaining complete records of actions and feedback can support more effective learning than summarizing these experiences into rules or strategies. (2) **Human–agent gap:** Top-performing human players achieve higher peak scores than the evaluated agents. Humansss explore more varied strategies, and repeat actions less. (3) **Harness matters:** With the backbone fixed, changing the harness can improve performance while reducing estimated inference cost. Together, these findings provide insights into how LLM agents learn from experience and suggest directions for future work to improve their learning ability. Code is available at `https://anonymous.4open.science/r/Learn2Play-Bench-F1D8/`.

1 INTRODUCTION

The ability of LLM agents to acquire new knowledge through interaction and use it to improve behavior is central to artificial general intelligence (AGI) (Hendrycks et al., 2025). While the extensive knowledge acquired during pretraining enables agents to perform complex tasks (e.g., fixing software issues (Yang et al., 2024), completing web tasks (Zhou et al., 2024), and operating desktop applications (Xie et al., 2024)), in realistic deployment, agents often encounter **unfamiliar environments** whose rules are not covered by the pretrained knowledge. For example, an agent struggles to control a robot through an unfamiliar command interface. It must therefore try commands and observe the resulting motion to learn which joints each command activates. Agents can then apply what they have learned to make better decisions in subsequent interactions.

Recent work has therefore developed self-evolving agents that reuse interaction feedback through memories, extracted rules, reasoning patterns, or evolving agent configurations (Zhang et al., 2025b, He et al., 2025, Ouyang et al., 2025). However, evaluating whether these systems actually learn remains difficult. Existing methods are often studied with different models, tasks, and evaluation protocols, making their learning behavior hard to compare directly. More fundamentally, **better task performance does not necessarily mean that an agent learns more effectively**. If an agent improves on a familiar task, we do not know whether it learned a new rule from interaction or merely recovered knowledge already encoded during pretraining.

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.

This distinction is often overlooked in existing evaluations of continual or self-evolving agents. Benchmarks often measure improvement over repeated attempts or task sequences (He et al., 2025, Zheng et al., 2025, Wei et al., 2025, Yan et al., 2026, Yang et al., 2026, Zhang et al., 2025a), but many tasks are based on familiar real-world settings whose underlying rules may already be represented in pretraining, leaving agents' learning ability insufficiently tested. We therefore ask:

How effectively can LLM agents learn from experience in unfamiliar environments?

To answer this question, we introduce **Learn2Play Bench**, a benchmark of newly designed text-based games with unfamiliar or counterintuitive hidden rules. For example, in one game, stealing a colleague's lunch helps the player get promoted, while working hard does not. Agents are not given these rules. They must discover them by acting, observing textual feedback, and carrying useful experience into later episodes, without updating model weights. Each game provides deterministic feedback and automatic scoring, allowing learning trajectories to be measured reproducibly. In half of the games, characters, objects, or layouts change across episodes while the underlying rules remain fixed. This setting tests whether agents can apply knowledge to new situations.

Using this controlled setting, we compare different backbone models, self-evolving methods, and deployed agent harnesses, and contrast their behavior with human players. Our experiments reveal five main findings: **(1) Preserving evidence can matter more than compressing it.** Raw interaction histories can support stronger learning than methods that summarize experience into rules or strategies, because the original evidence remains available when earlier conclusions need revision (§4.2). **(2) Top-performing human players outperform the best agent.** Top-performing human players reach higher peak scores than the evaluated agents, while exploring more varied action sequences and recovering more often after setbacks (§4.4). **(3) Better harnesses can improve agent performance and learning while reducing inference cost.** With the backbone fixed, changing the harness can substantially improve both performance and learning. In some cases, it also reduces inference cost through fewer model calls and less repeated context (§4.3, §4.6). **(4) Agents can learn more slowly in changing environments**, even when the underlying rules stay the same. The drop in learning slope tends to be larger for weaker backbones under the same harness (§4.5). **(5) Agents can learn rules from experience but still struggle to use them to develop effective strategies.** For example, in HAUNTED INN, an agent learns how to assign guests to safe rooms but fails to plan ahead, leaving a later guest with nowhere safe to stay (§4.2).

Our contributions are threefold:

- We introduce **Learn2Play Bench**, a controlled benchmark of newly designed games with hidden rules, deterministic feedback, and automatic scoring. We also design re-shuffled games to evaluate whether agents could transfer the knowledge to new instances.
- We systematically compare backbone models, self-evolving methods, and agent harnesses, revealing factors that affect learning from experience and identifying directions for future research.
- We conduct a human study on the same games and compare scores and interaction records to examine differences in performance and behavior between human players and LLM agents.

2 RELATED WORK

Self-evolving agents. Recent work develops self-evolving agents to learn from experience at test time, adapting their behavior using feedback from earlier attempts. Early methods reuse experience without updating model weights through verbal reflections in Reflexion (Shinn et al., 2023), extracted insights and retrieved trajectories in ExpeL (Zhao et al., 2024), or reusable executable skills in Voyager (Wang et al., 2024). Other methods organize experience into workflows and playbooks. For example, AWM induces action workflows (Wang et al., 2025), Dynamic Cheatsheet curates compact strategies and code snippets (Suzgun et al., 2026), and ACE incrementally maintains structured playbooks rather than raw transcripts (Zhang et al., 2025b). ReasoningBank and ERL distill trajectories into reusable reasoning patterns or heuristics (Ouyang et al., 2025, Allard et al., 2026). Beyond constructing memories, GEPA evolves prompts through trajectory-level reflection (Agrawal et al., 2025), while Agents of Change revises both prompts and agent code through specialized collaborative roles (Belle et al., 2025). EvoTest extends evolution to the broader agent configuration, including memory and tool-use routines (He et al., 2025). Meta-TTL further evolves the adaptation policy that determines how experience should update the agent (Lou et al., 2026). Beyond context-level updates, TT-SI generates training examples from uncertain cases and fine-tunes the agent at

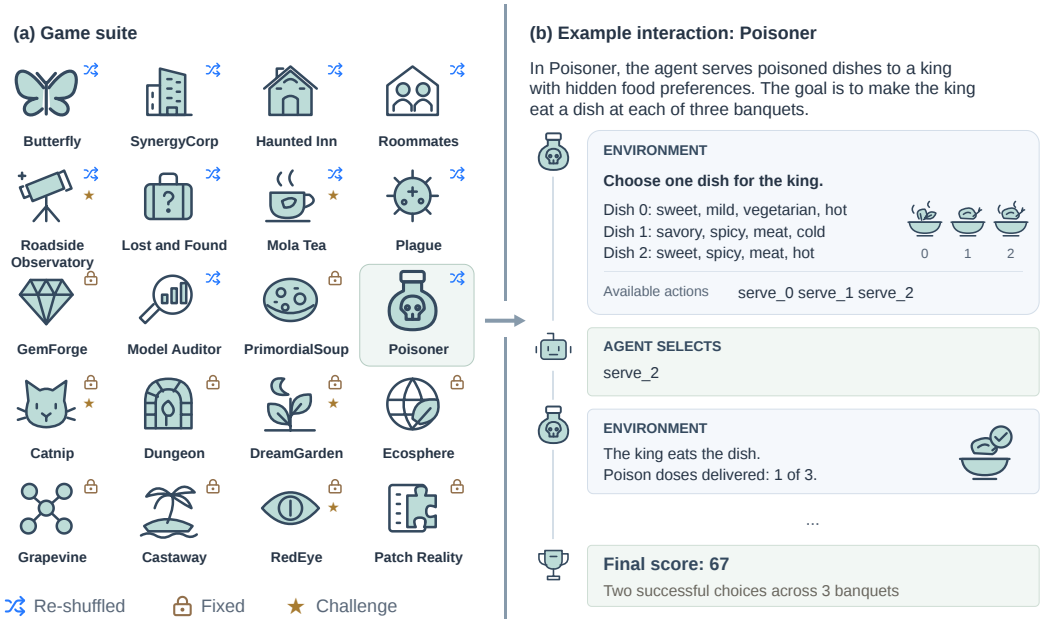


Figure 1: Learn2Play Bench overview. The left panel shows 20 manually designed game templates, each of which generates multiple instances using different random seeds. The highlighted POISONER template illustrates the interaction protocol on the right: the environment describes the state and enumerates valid actions, the player selects one action, and the environment returns the resulting observation. The episode ends with an automatically computed score (Appendix D.16).

test time (Acikgoz et al., 2026). Our evaluation focuses on learning without weight updates: we compare backbones in a common harness and self-evolving methods with each backbone held fixed.

Benchmarks for self-evolving agents. To assess whether these adaptation mechanisms improve later behavior, benchmarks evaluate agents over task sequences or repeated interactions. StreamBench uses input–feedback streams (Wu et al., 2024), while MemoryBench studies learning from accumulated user feedback and Evo-Memory evaluates memory updates across task streams (Ai et al., 2025, Wei et al., 2025). Related evaluations extend experience reuse to interactive tasks: LifelongAgentBench tests skill accumulation across related tasks (Zheng et al., 2025), SWE-Bench-CL evaluates learning across successive coding issues (Joshi et al., 2025), and MemoryArena tests memory use across interdependent sessions (He et al., 2026). These benchmarks test whether experience improves later performance. On familiar tasks, however, it is unclear whether agents improve by learning something new or by using what they already know. Other benchmarks study learning directly through interaction with environments. AutoEnv (Zhang et al., 2025a) also creates new game environments and uses them to improve agent performance, while we focus on evaluating how effectively agents learn from experience across repeated episodes. Closer to our setting, J-TTL (He et al., 2025) measures repeated improvement on Jericho games, but their public content may already be represented in model pretraining. ARC-AGI-3 tests interaction with unfamiliar visual environments, but performance also depends on visual capabilities (Foundation, 2026). We use newly designed text-based games with hidden rules to examine whether agents can discover previously unknown rules through actions and feedback.

3 Evaluation Design

To measure whether agents improve through experience in unfamiliar environments, we design games that satisfy the following principles:

- **Rule novelty.** We design games with novel hidden rules not found in existing games, including deliberately counterintuitive rules that conflict with common game knowledge. For example, in one of our games, instead of hard work, stealing a colleague's lunch helps the player get promoted.

The agent must therefore learn which actions lead to promotion from the game's feedback, rather than rely on expectations about real workplaces.

- **Automatic scoring.** Each game automatically assigns a score, enabling large-scale evaluation without human grading.
- **Reproducibility.** Deterministic outcomes allow us to replay interactions, compare agents under the same conditions, and inspect mistakes.
- **Controlled variation across episodes.** To test whether agents transfer learned knowledge, we vary the visible situation across episodes while preserving the core rule.

Based on the four principles above, we manually design Learn2Play Bench with 20 game templates, each of which can generate multiple instances that differ in layout, characters, or hidden-rule parameters. Figure 1 illustrates the game suite and an example interaction. Agents receive a goal and a set of available actions, interact through text to discover the hidden rules, and receive a final score at the end of each episode. They play repeated episodes and can use experience from earlier attempts. We divide the games along two dimensions. First, ten **fixed** games repeat the same instance across episodes, whereas ten **re-shuffled** games change the visible instance while preserving the core rule. Second, we conduct pilot evaluations with several agents to estimate game difficulty and identify five **challenge** games.[1] We use a ten-episode scoring window for these games and a five-episode window for the remaining **normal** games. Appendix A explains what players must learn in each game, and appendix D provides the corresponding results and archived rule-discovery evidence.

4 Experiments

4.1 Experimental settings

Datasets. We evaluate all agents on the same 20 game instances from Learn2Play Bench, using one fixed seed per game template.

Protocol. We evaluate each agent through a black-box command interface. The agent issues actions and observes textual feedback but cannot access the game source code or oracle solvers. We evaluate each game g for T_g episodes: $T_g = 5$ for normal games and $T_g = 10$ for challenge games, which are harder and therefore require more episodes to learn. The protocol requires three independent trials per game. Each trial starts with no memory of previous trials. Within a trial, the information retained from earlier episodes depends on the harness.

Metrics. We use four metrics to evaluate two complementary aspects of agent behavior across episodes: achieved performance and improvement through experience. Max and Mean measure achieved performance, while Learning Gain (LG) and Learning Slope (LS) measure improvement across episodes. Because score ranges differ across games, we compute these metrics using normalized scores. Appendix A.2 explains normalization, trial aggregation, and game weighting. Let $z_{g,r,t}$ be a normalized terminal score for game g, trial r, and episode t, and let $y_{g,t}$ denote the mean of these scores at episode t. We define the four metrics as follows:

- **Max** averages the peak terminal scores $m_{g,r}$ of the R_g scored trial trajectories within the evaluation window: $\mathrm{Max}_g = R_g^{-1} \sum_{r=1}^{R_g} m_{g,r}$. It measures the best performance an agent demonstrates during evaluation.
- **Mean** averages performance across all evaluated episodes: $\mathrm{Mean}_g = T_g^{-1} \sum_{t=1}^{T_g} y_{g,t}$. It measures how consistently the agent performs throughout evaluation.
- **Learning Gain (LG)** compares average performance in the first and last $k_g = \min(3, \lfloor T_g/2 \rfloor)$ episodes: $\mathrm{LG}_g = k_g^{-1}(\sum_{t=T_g-k_g+1}^{T_g} y_{g,t} - \sum_{t=1}^{k_g} y_{g,t})$. It measures the net improvement from the beginning to the end of evaluation.
- **Learning Slope (LS)** fits $y_{g,t} = \beta_{0,g} + \beta_{1,g} t + \epsilon_{g,t}$ over all evaluated episodes and reports $\mathrm{LS}_g = \hat{\beta}_{1,g}$. It measures the overall direction and rate of change across the full learning trajectory.

We first compute each metric separately for each game and then aggregate it across games as $\sum_g w_g M_g / \sum_g w_g$, assigning weight 3 to challenge games and weight 1 to normal games. The

[1] The challenge games are Catnip, DreamGarden, Mola Tea, RedEye, and Roadside Observatory.

five challenge games and fifteen normal games therefore each receive 50% of the total weight. To aggregate learning curves from games with different numbers of episodes, we map each game's episodes onto a common 0–100% progress scale and use linear interpolation only for visualization.

4.2 RQ1: How effectively do current agents learn?

We first investigate how learning changes with the choice of baseline model and with the method used to retain experience across episodes. Appendix B.1 provides additional results and supporting cases from interaction logs.

Baseline models. To examine how learning from experience varies across different backbones, we compare nine models using OpenCode (OpenCode Contributors, 2026), an open-source coding agent harness that connects configurable language models to shell and file tools. We use its command-line agent as a common harness for black-box game interaction through the replay interface, keeping the agent loop, prompts, replay protocol, and memory format fixed across backbones. The backbones are Claude Opus 5, Claude Sonnet 5, Gemini 3.5 Flash, Gemini 3.7 Flash, GPT-5.6-SOL, GPT-5 Mini, DeepSeek V4 Pro, Kimi K3, and Qwen 3.8 Max.

Methods. To examine how different ways of retaining and reusing experience affect learning, we compare self-evolving methods separately for Kimi K3, Claude Opus 5, and GPT-5 Mini, holding the backbone fixed within each comparison. We evaluate the following methods: **Naive** retains the dialogue within the current episode but discards it before the next, providing no cross-episode experience. **Memory** cumulatively adds the complete, unmodified interaction histories from preceding episodes to each new episode's prompt, dropping the oldest episodes only when the history exceeds the context budget. **Reflexion** (Shinn et al., 2023) reflects on each completed episode to derive concise lessons about discovered rules, successful and failed actions, and next steps, then adds the accumulated lessons to the prompt for later episodes. **EvoTest** (He et al., 2025) uses earlier interaction histories and outcomes to evolve a guiding prompt and a state-extraction program that steer action selection in subsequent episodes. **ReasoningBank** (Ouyang et al., 2025) turns each successful or failed episode into a small set of reusable reasoning lessons and adds the accumulated bank to the prompt in later episodes. **Agent Workflow Memory (AWM)** (Wang et al., 2025) induces reusable multi-step workflows from successful interaction histories and includes the resulting workflows in later episode prompts. Failed episodes are not used.

To assess how these backbones and methods learn across episodes, we plot learning curves for the backbone comparison in Figure 2(A) and the method comparison with Claude Opus 5 in Figure 2(B). Table 1 reports the backbone, human, and matched-method results, including all six methods with Kimi K3 and Claude Opus 5. Table 7 reports the GPT-5 Mini method comparison and three-backbone averages. Appendix D groups each game's learning curves, leaderboard, and rule-discovery evidence in one entry. The annotation criteria and archive coverage are specified in Appendix D.

4.2.1 Backbone comparison

Opus 5 leads Max and Mean, while Qwen 3.8 Max leads LG and LS. The model that scores highest is not necessarily the one that improves most with experience. As shown in Table 1, Opus 5 achieves the highest peak and average scores, whereas Qwen 3.8 Max shows greater improvement across episodes.

Stronger models use feedback to adjust their decisions, whereas weaker models may repeat the same actions despite new observations. We inspected the interaction logs to understand how models use experience. Opus 5 and Qwen 3.8 Max test their understanding of the hidden rules and update it when new observations do not match their expectations. In contrast, weaker models sometimes repeat actions that worked before, even when they no longer work. They may also change several factors at once, making it hard to tell which factor affects the outcome. For example, in Poisoner, agents choose dishes for a king and learn about his food preferences from whether he accepts or rejects them. In one GPT-5 Mini run, the agent keeps choosing dishes by their position on the menu, without using feedback from earlier choices. Kimi K3 instead uses that feedback to choose which food attribute to test next.

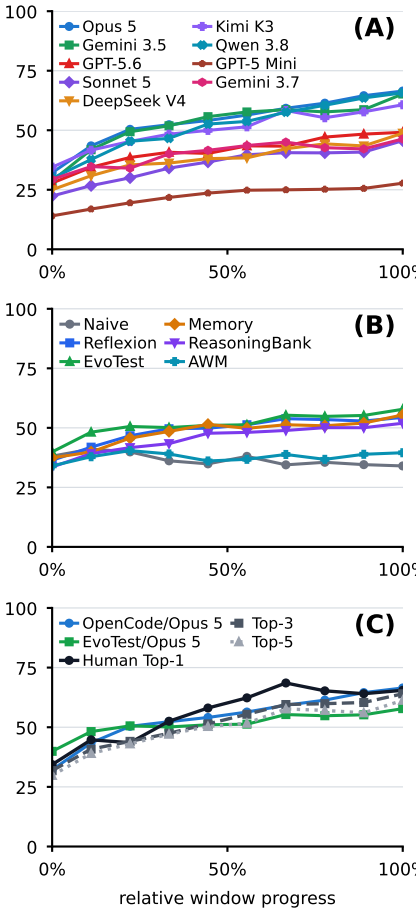


Figure 2: Normalized learning curves across all 20 games for (A) OpenCode backbones, (B) self-evolving methods with Opus 5, and (C) human Top-k groups and selected agents.

Table 1: Performance and learning across backbone models, self-evolving methods, and humans. Backbone comparisons use OpenCode; method comparisons hold the backbone fixed. Human Top-k groups are selected by Max. Bold marks the best agent value in each metric within each comparison group.

Backbone	Method	Max	Mean	LG	LS
Claude Opus 5	OpenCode	**74.5**	**53.8**	+21.3	+5.13
Qwen 3.8 Max	OpenCode	72.5	51.1	**+25.2**	**+6.30**
Gemini 3.5 Flash	OpenCode	72.0	52.2	+19.8	+5.37
Kimi K3	OpenCode	68.5	50.2	+17.1	+4.24
Gemini 3.7 Flash	OpenCode	58.3	39.9	+10.7	+2.47
GPT-5.6-SOL	OpenCode	57.0	41.1	+14.5	+3.71
Claude Sonnet 5	OpenCode	55.9	35.6	+15.7	+3.72
DeepSeek V4 Pro	OpenCode	54.2	38.2	+14.7	+3.66
GPT-5 Mini	OpenCode	34.0	22.3	+9.2	+1.83
Kimi K3	MEMORY	**63.3**	**48.0**	**+16.9**	**+4.69**
	EvoTest	62.4	48.0	+1.8	+1.01
	Reflexion	61.6	40.4	+14.8	+3.35
	ReasoningBank	59.9	40.1	+8.3	+2.59
	AWM	56.1	37.0	+4.8	+0.62
	NAIVE	54.7	31.9	+0.3	+0.42
Opus 5	EvoTest	**66.1**	**51.3**	+9.4	+2.52
	Reflexion	64.8	48.8	+11.6	**+2.93**
	MEMORY	60.9	48.2	+11.4	+2.87
	ReasoningBank	60.2	45.3	**+12.3**	+2.85
	NAIVE	53.4	36.6	-5.1	-1.08
	AWM	50.7	37.8	+0.7	+0.43
Human Top-1		84.3	53.5	+21.6	+7.33
Human Top-3		79.1	50.1	+18.9	+5.54
Human Top-5		75.1	47.1	+17.1	+4.62

4.2.2 SELF-EVOLVING METHOD COMPARISON

Simply using past interaction histories can outperform more elaborate self-evolving methods. MEMORY retains raw histories without reflection, rule extraction, or workflow induction, yet leads all four metrics with Kimi K3. This is because reflection and memory processing can turn an incomplete understanding into rules or workflows that later episodes continue to follow, while raw histories keep the original evidence available for revising those interpretations. For example, in GEMFORGE, agents combine ingredients to make gems. After one combination fails, an EvoTest–Opus 5 run adds a rule to its prompt telling it to avoid many other combinations, including one that would succeed.

OpenCode achieves better performance than the evaluated self-evolving methods when using the same backbone. OpenCode is more flexible during play. OpenCode lets the model decide when to review earlier feedback, run calculations, or check a plan before taking an action. By contrast, the evaluated self-evolving methods follow preset steps for choosing actions and updating experience. When an unexpected result occurs, OpenCode gives the model room to investigate and revise its plan before continuing, rather than only updating its memory for later episodes.

4.2.3 WHAT LIMITS FURTHER IMPROVEMENT?

Agents may stop exploring better strategies after finding one that partly works. The game-specific audit shows agents repeating known strategies instead of testing alternatives. For example, in GEMFORGE, players combine ingredients to make increasingly valuable gems. After several failed attempts, one Opus 5 run concludes that higher-value gems cannot be made and keeps repeating known recipes, limiting further improvement.

Agents may know the rules but fail to plan ahead. In an Opus 5 HAUNTED INN run, the agent states the relevant taboos but assigns rooms greedily, leaving no safe room for a later guest. In the next episode, it reserves rooms for guests with fewer safe options. The failure lies in choosing an action that works now without considering what it prevents later.

4.3 RQ2: HOW DOES THE AGENT HARNESS AFFECT LEARNING?

To isolate the effect of the harness from the choice of model, we hold the backbone fixed and compare OpenCode with Claude Code for Opus 5, and with Codex for GPT-5.6-SOL. All configurations follow the same evaluation protocol. Appendix B.2 specifies trial isolation, within-trial history, and the matched harness configurations.

Table 2: Same-backbone harness comparison.

Harness	Max	Mean	LG	LS
Claude Opus 5				
OpenCode	74.5	53.8	+21.3	+5.13
Claude Code	**81.8**	**62.3**	**+27.1**	**+6.59**
GPT-5.6-SOL				
OpenCode	57.0	41.1	+14.5	+3.71
Codex	**74.3**	**54.8**	**+22.2**	**+5.66**

Claude Code and Codex yield higher performance and stronger learning than OpenCode when paired with the same backbone. As shown in Table 2, Claude Code outperforms OpenCode on all four metrics with Opus 5, as does Codex with GPT-5.6-SOL. This is because models are trained on their harnesses and thus would perform better, and these harnesses are also adapted to the models' capabilities (MacCallum & Fioca, 2026, Shihipar, 2026).

4.4 RQ3: HOW DO HUMANS AND AGENTS LEARN DIFFERENTLY?

We compare humans and agents to understand how their use of experience differs when both must discover unfamiliar rules. We recruited college students for the human study. They use the same game interface, with 21–30 participants per game. For each game, we select the Top-1, Top-3, and Top-5 records by Max. Their aggregate metrics appear in Table 1. Appendix B.3 provides full results, including the mean across all participants, behavioral statistics, and cases from interaction logs. Appendix D reports per-game scores. To compare their learning across episodes, Figure 2(C) shows these human curves alongside OpenCode–Claude Opus 5 and EvoTest–Opus 5, which lead Max and Mean in the backbone and Opus 5 method comparisons, respectively.

Human Top-1 achieves higher Max than the models and self-evolving methods evaluated in our experiments. This may be because humans explore more broadly, whereas agents exploit discovered strategies more consistently. To examine this behavioral difference beyond the single best record, we compare observed episodes from Human Top-3 with the models and self-evolving methods. In fixed games, consecutive action sequences have an average similarity of 0.54 for humans and 0.66 for agents. After a lower-scoring episode, humans reach a new personal best in the next episode 33% of the time, compared with 22% for agents. These observations suggest that humans more often change strategies and recover from temporary failures, while agents more often repeat a reliable but potentially suboptimal strategy.

Top-performing humans perform better when success requires finding new strategies, whereas agents perform better when success requiresss executing long action sequences or doing precise calculations. Human Top-1 scores higher in GEMFORGE and GRAPEVINE, while the strongest OpenCode backbone scores higher in REDEYE and MODEL AUDITOR. In REDEYE, successful agents combine clues and carry out all the steps needed to rescue the passengers. In MODEL AUDITOR, they use programs to infer the hidden rules and check their predictions. Some human players in this game reduce or stop testing in later episodes, suggesting that the effort required to keep investigating may partly explain their lower scores.

4.5 RQ4: How well does learned knowledge transfer to new game instances?

We assess how well agents transfer what they learn to new instances in re-shuffled games. We test seven backbone models using the same OpenCode harness. We also test Claude Code with Opus 5 and Codex with GPT-5.6-SOL. For each re-shuffled game, we create a same-instance control: the agent encounters the same instance in every episode, while the re-shuffled version changes the visible instance but preserves the underlying rule. We compare LS over ten episodes for games evaluated in both conditions, with three independent trials required per game. Appendix B.4 also reports the six methods with Opus 5. Appendices B.4.1 and B.1.4 provide examples from interaction logs of applying learned rules and repeating menu positions.

When game instances change each episode, Sonnet 5 and the Gemini Flash models improve much more slowly than Opus 5 and Kimi K3. Opus 5 and Kimi K3 show smaller drops in learning slope (Table 3). In one SynergyCorp run, Opus 5 applies a rule learned from earlier bosses to new bosses, whereas Sonnet 5 and Gemini 3.5 Flash test each new boss's preferences again.

Past success can mislead transfer when an agent remembers what worked but not why it worked. Agent systems with higher re-shuffled LS can identify why an action worked and find where the same conditions hold in a new instance. Systems with lower LS sometimes repeat episode-specific choices, such as a menu position, even after the menu changes.

Table 3: Learning slopes when the same game instance repeats (Control) or changes across episodes (Re-shuf.), with the underlying rules held fixed. LS drop is Control minus Re-shuf.; positive values indicate slower learning after re-shuffling.

Agent	Control	Re-shuf.	LS drop
OpenCode backbones			
Claude Opus 5	4.10	3.51	0.59
Qwen 3.8 Max	3.16	3.57	-0.41
Gemini 3.5 Flash	4.50	2.76	1.73
Kimi K3	4.19	3.31	0.88
Gemini 3.7 Flash	4.49	2.34	2.15
GPT-5.6-SOL	3.72	1.78	1.94
Claude Sonnet 5	4.90	2.23	2.67
Deployed agents			
Claude Code (Opus 5)	4.30	3.77	0.52
Codex (GPT-5.6-SOL)	2.51	2.71	-0.19

4.6 RQ5: Do higher inference costs lead to better performance?

We compare agents' Max scores with their mean cost per episode in Figure 3. Table 12 reports the numerical cost results for each configuration. Appendix B.5 provides calculation details, token-use breakdowns, and supporting examples from interaction logs.

Under the same harness, models with higher token prices can achieve better performance at a lower overall cost. With OpenCode, Kimi K3 achieves higher performance at a lower overall cost than Claude Sonnet 5, despite having higher input and output token prices. Kimi K3 also uses fewer total tokens per episode: 0.56M on average, compared with 0.73M for Claude Sonnet 5 (Table 12). Higher token prices therefore do not imply higher overall cost.

With the model fixed, a better-suited harness can improve performance while reducing cost. OpenCode with Kimi K3 achieves higher Max at lower estimated cost than Memory, EvoTest, and ReasoningBank. Claude Code with Opus 5 also outperforms these self-evolving methods and OpenCode at lower estimated cost. For Kimi K3, OpenCode uses 56% fewer input tokens than Memory and 71% and 41% fewer generated tokens than EvoTest and ReasoningBank, respectively, as reported in Table 11. These self-evolving methods call the model for each action and repeatedly supply past interactions or stored memories. OpenCode instead executes several actions from one model response and removes repeated game feedback, reducing model calls and repeated input.

5 Discussions and Future Directions

Building on our findings, we discuss open questions in how to improve the learning ability of LLM agents and outline directions for future research.

How should agents balance exploration and exploitation? In our comparison of human players and agents, Human Top-1 achieves a higher Max score than every agent configuration we evaluate, as shown in Tables 1 and 2. This is because humans try more varied strategies across episodes, while agents more often reuse strategies that have worked before (Appendix B.3.3). Future work should

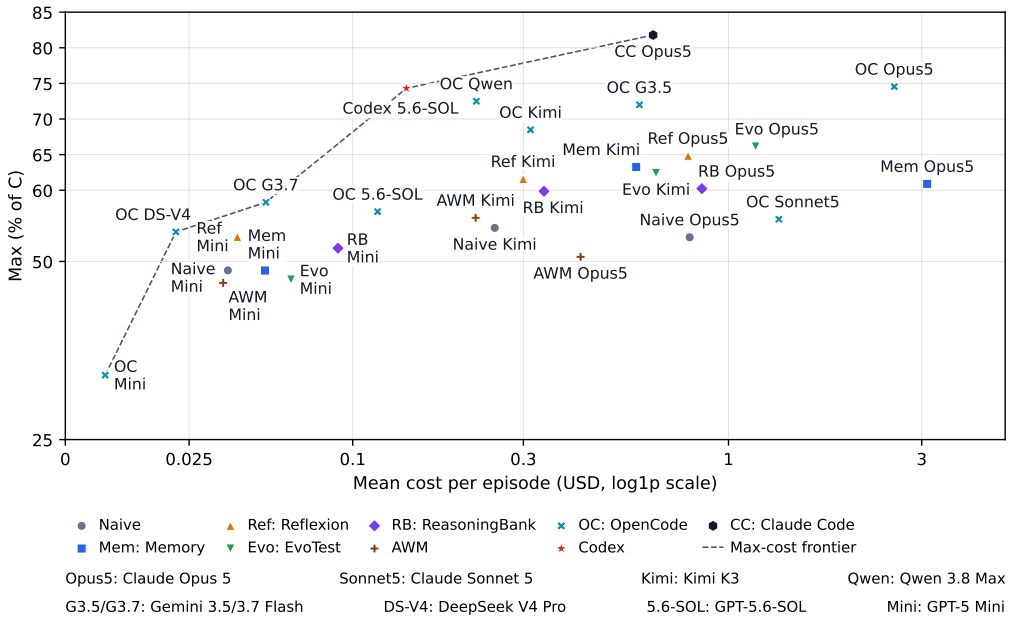


Figure 3: Max score versus estimated mean inference cost per episode. The dashed line connects configurations on the performance–cost frontier.

better balance exploration and exploitation, allowing agents to test new strategies even when their current strategy already earns high scores.

How should agents store learned rules? With Kimi K3, MEMORY outperforms the other self-evolving methods overall, as shown in Table 1. Unlike those methods, it retains full interaction histories rather than first distilling them into rules. This distinction matters when agents draw rules from limited evidence. In GEMFORGE, EvoTest tried one gem combination and failed, but its later prompt instructed the agent not to test other combinations it had never tried (Appendix B.1.3). The prompt thus turned incomplete evidence into a definitive rule. Future work can explore ways to store experience, such as linking each learned rule to the original interactions behind it so agents can revisit the evidence and revise the rule when new feedback arrives.

How can agents learn what stays constant as environments change? Our transfer evaluation shows that changing surface details, such as layouts or characters, can reduce agents' learning slopes even when the underlying rules remain the same (Table 3). In real-world tasks, agents also regularly encounter changes in their surroundings, available resources, and the people they interact with. This raises a broader question: how can agents adapt to these changes in their environments? Future methods could help agents compare experiences across different situations, distinguish shared rules from instance-specific details, and check when those rules still apply.

6 CONCLUSION

In this study, we investigate how effectively LLM agents learn from experience in unfamiliar environments. Existing evaluations make it difficult to distinguish newly acquired knowledge from prior knowledge. To address this issue, we introduce Learn2Play Bench, a benchmark of newly designed text-based games with hidden rules that agents discover through interaction without updating their model weights. We conducted comprehensive experiments comparing backbone models, self-evolving methods, deployed agent harnesses, and human players. Our findings show that more elaborate self-evolving methods do not always improve learning, human players explore more varied strategies, with top-performing humans achieving higher peak scores than the models and self-evolving methods we evaluate, and agent harnesses can improve performance while reducing task costs. We hope this work provides a foundation for evaluating and improving how agents learn from experience.

AI USE STATEMENT

We used AI-assisted tools during benchmark development and manuscript preparation. Claude Code assisted with writing the game code for Learn2Play Bench, including the game logic and interaction interfaces, and with developing the server that supports gameplay. Gemini was used to polish the manuscript, focusing on wording, sentence structure, and readability. The authors take responsibility for the research design, experimental results and their interpretation, and the final manuscript and code, including content prepared with AI assistance.

ETHICS STATEMENT

Learn2Play Bench studies learning in text-based games. Before participation, participants agreed to have their gameplay actions and scores recorded and used for research. The human comparison analyzes game actions and scores. The paper reports aggregate results and gameplay examples without participant names. These task-specific results, particularly the selected Human Top-k groups, should not be interpreted as representative of the general population or as a general ranking of human and machine intelligence.

REPRODUCIBILITY STATEMENT

The game design and interaction protocol are described in Section 3, and the evaluated models, methods, and scoring rules in Section 4. Appendix A.2 specifies the pinned game instances, reference scores, normalization, and aggregation. Harness settings, human-record selection, and cost estimation are documented in Appendices B.2, B.3, and B.5, respectively. Per-game results and rule-discovery evidence appear in Appendix D. The game engines support deterministic replay with a fixed seed, episode index, and action history. Repeated model runs can nevertheless produce different trajectories because model outputs and provider-side implementations may vary.

REFERENCES

Emre Can Acikgoz, Cheng Qian, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. TT-SI: self-improving LLM agents with test-time training. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026, pp. 9483–9508. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.462. URL https://doi.org/10.18653/v1/2026.findings-acl.462.

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Daniel Klein, Matei Zaharia, and Omar Khattab. GEPA: reflective prompt evolution can outperform reinforcement learning. CoRR, abs/2507.19457, 2025. doi: 10.48550/ARXIV.2507.19457. URL https://doi.org/10.48550/arXiv.2507.19457.

Qingyao Ai, Yichen Tang, Changyue Wang, Jianming Long, Weihang Su, and Yiqun Liu. Memorybench: A benchmark for memory and continual learning in LLM systems. CoRR, abs/2510.17281, 2025. doi: 10.48550/ARXIV.2510.17281. URL https://doi.org/10.48550/arXiv.2510.17281.

Marc-Antoine Allard, Arnaud Teinturier, Victor Xing, and Gautier Viaud. Experiential reflective learning for self-improving LLM agents. CoRR, abs/2603.24639, 2026. doi: 10.48550/ARXIV.2603.24639. URL https://doi.org/10.48550/arXiv.2603.24639.

Anthropic. Prompt caching. https://platform.claude.com/docs/en/build-with-claude/prompt-caching, 2026. Accessed 30 August 2026.

Nikolas Belle, Dakota Barnes, Alfonso Amayuelas, Ivan Bercovich, Xin Eric Wang, and William Wang. Agents of change: Self-evolving LLM agents for strategic planning. CoRR, abs/2506.04651, 2025. doi: 10.48550/ARXIV.2506.04651. URL https://doi.org/10.48550/arXiv.2506.04651.

ARC Prize Foundation. ARC-AGI-3: A new challenge for frontier agentic intelligence. CoRR, abs/2603.24621, 2026. doi: 10.48550/ARXIV.2603.24621. URL https://doi.org/10.48550/arXiv.2603.24621.

Yufei He, Juncheng Liu, Yue Liu, Yibo Li, Tri Cao, Zhiyuan Hu, Xinxing Xu, and Bryan Hooi. Evotest: Evolutionary test-time learning for self-improving agentic systems. CoRR, abs/2510.13220, 2025. doi: 10.48550/ARXIV.2510.13220. URL https://doi.org/10.48550/arXiv.2510.13220.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian J. McAuley, Yejin Choi, and Alex Pentland. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. CoRR, abs/2602.16313, 2026. doi: 10.48550/ARXIV.2602.16313. URL https://doi.org/10.48550/arXiv.2602.16313.

Dan Hendrycks, Dawn Song, Christian Szegedy, Honglak Lee, Yarin Gal, Erik Brynjolfsson, Sharon Li, Andy Zou, Lionel Levine, Bo Han, Jie Fu, Ziwei Liu, Jinwoo Shin, Kimin Lee, Mantas Mazeika, Long Phan, George Ingebretsen, Adam Khoja, Cihang Xie, Olawale Salaudeen, Matthias Hein, Kevin Zhao, Alexander Pan, David Duvenaud, Bo Li, Steve Omohundro, Gabriel Alfour, Max Tegmark, Kevin McGrew, Gary Marcus, Jaan Tallinn, Eric Schmidt, and Yoshua Bengio. A definition of AGI. CoRR, abs/2510.18212, 2025. doi: 10.48550/ARXIV.2510.18212. URL https://doi.org/10.48550/arXiv.2510.18212.

Thomas Joshi, Shayan Chowdhury, and Fatih Uysal. Swe-bench-cl: Continual learning for coding agents. CoRR, abs/2507.00014, 2025. doi: 10.48550/ARXIV.2507.00014. URL https://doi.org/10.48550/arXiv.2507.00014.

Zhanzhi Lou, Hui Chen, Yibo Li, Qian Wang, and Bryan Hooi. Learning to learn-at-test-time: Language agents with learnable adaptation policies. CoRR, abs/2604.00830, 2026. doi: 10.48550/ARXIV.2604.00830. URL https://doi.org/10.48550/arXiv.2604.00830.

Noah MacCallum and Brian Fioca. Codex prompting guide. https://developers.openai.com/cookbook/examples/gpt-5/codex_prompting_guide, 2026. OpenAI. Published 25 February 2026. Accessed 25 September 2026.

OpenCode Contributors. OpenCode: An open source AI coding agent. https://opencode.ai/docs/, 2026. Documentation accessed 6 September 2026.

OpenRouter. Models and pricing. https://openrouter.ai/api/v1/models, 2026. Price snapshot accessed 30 August 2026.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. Reasoningbank: Scaling agent self-evolving with reasoning memory. CoRR, abs/2509.25140, 2025. doi: 10.48550/ARXIV.2509.25140. URL https://doi.org/10.48550/arXiv.2509.25140.

Thariq Shihipar. Seeing like an agent: How we design tools in Claude Code. https://claude.dev/blog/seeing-like-an-agent/, 2026. Anthropic. Published 10 April 2026. Accessed 25 September 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mirac Suzgun, Mert Yüksekgönül, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. pp. 7080–7106, 2026. doi: 10.18653/V1/2026.EACL-LONG.333. URL https://doi.org/10.18653/v1/2026.eacl-long.333.

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. Trans. Mach. Learn. Res., 2024, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, volume 267 of Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H. Chi, Chi Wang, Shuo Chen, Fernando Pereira, Wang-Cheng Kang, and Derek Zhiyuan Cheng. Evo-memory: Benchmarking LLM agent test-time learning with self-evolving memory. CoRR, abs/2511.20857, 2025. doi: 10.48550/ARXIV.2511.20857. URL https://doi.org/10.48550/arXiv.2511.20857.

Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung-yi Lee. Streambench: Towards benchmarking continuous improvement of language agents. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/c189915371c4474fe9789be3728113fc-Abstract-Datasets_and_Benchmarks_Track.html.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Dong Yan, Jian Liang, Dapeng Hu, Ran He, Nicholas Jing Yuan, Qi Zhang, and Tieniu Tan. Agentstream: How well do self-evolving LLM agents perform under streaming tasks? CoRR, abs/2608.00155, 2026. doi: 10.48550/ARXIV.2608.00155. URL https://doi.org/10.48550/arXiv.2608.00155.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

Xidong Yang, Xingyi Zhang, Wenhao Li, Wenyan Liu, Junjie Sheng, Yun Hua, Wei Yin, Tao Fang, Chuyun Shen, and Xiangfeng Wang. Path-bench: Path-dependent evaluation of lifelong agents. CoRR, abs/2608.01149, 2026. doi: 10.48550/ARXIV.2608.01149. URL https://doi.org/10.48550/arXiv.2608.01149.

Jiayi Zhang, Yiran Peng, Fanqi Kong, Cheng Yang, Yifan Wu, Zhaoyang Yu, Jinyu Xiang, Jianhao Ruan, Jinlin Wang, Maojia Song, HongZhang Liu, Xiangru Tang, Bang Liu, Chenglin Wu, and Yuyu Luo. Autoenv: Automated environments for measuring cross-environment agent learning. CoRR, abs/2511.19304, 2025a. doi: 10.48550/ARXIV.2511.19304. URL https://doi.org/10.48550/arXiv.2511.19304.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. CoRR, abs/2510.04618, 2025b. doi: 10.48550/ARXIV.2510.04618. URL https://doi.org/10.48550/arXiv.2510.04618.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: LLM agents are experiential learners. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada, pp. 19632–19642. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, Zhong-Zhi Li, Yingying Zhang, Le Song, and Qianli Ma. Lifelongagentbench: Evaluating LLM agents as lifelong learners. CoRR, abs/2505.11942, 2025. doi: 10.48550/ARXIV.2505.11942. URL https://doi.org/10.48550/arXiv.2505.11942.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

LIST OF APPENDICES

A Games and Evaluation Settings **16**

A.1 Games and learning tasks 16

A.2 Evaluation and scoring 16

A.2.1 Instances, episodes, and trials 16

A.2.2 From raw scores to per-game metrics 17

A.2.3 Averaging across games and plotting curves 17

A.2.4 Reference scores 17

B Additional Results by Research Question **18**

B.1 RQ1: How Effectively Do Agents Learn? 18

B.1.1 Evidence of rule discovery 18

B.1.2 Exploration and planning cases 19

B.1.3 How processed experience can restrict exploration 19

B.1.4 Using feedback instead of repeating menu positions 19

B.2 RQ2: How Does the Harness Affect Learning? 19

B.3 RQ3: How Do Humans and Agents Learn Differently? 19

B.3.1 Selecting human records 20

B.3.2 Calculating scores and learning curves 20

B.3.3 Comparing recorded behavior 20

B.3.4 Cases of execution and continued investigation 21

B.4 RQ4: How Well Does Learned Knowledge Transfer to New Game Instances? . . . 22

B.4.1 Applying a rule to a new instance 22

B.5 RQ5: Do Higher Inference Costs Lead to Better Performance? 23

B.5.1 Usage and pricing 23

C Aggregate Results and Trial Variability **25**

C.1 All 20 games 26

C.2 Fixed games 27

C.3 Re-shuffled games 28

D Per-Game Results and Rule-Discovery Evidence **29**

D.1 PrimordialSoup 29

D.2 GemForge 31

D.3 Catnip 32

D.4 Dungeon 34

D.5 SynergyCorp 36

D.6 DreamGarden 38

D.7 Ecosphere 39

D.8 Grapevine 41
D.9 Haunted Inn 42
D.10 Castaway 44
D.11 Roadside Observatory 46
D.12 Lost and Found 47
D.13 Mola Tea 49
D.14 Plague 51
D.15 Patch Reality 52
D.16 Poisoner 54
D.17 Roommates 55
D.18 RedEye 57
D.19 Butterfly 59
D.20 Model Auditor 60

A GAMES AND EVALUATION SETTINGS

A.1 GAMES AND LEARNING TASKS

The games test whether players can discover useful rules from feedback and apply them in later episodes. The two tables below describe what must be learned, rather than assigning separate ability scores. Detailed results for each game appear in Appendix D.

In re-shuffled games, the visible situation changes but the underlying rule remains the same. Players must use what they have learned with new menus, guests, or other instances. In fixed games, the same instance repeats, so players can refine a map, recipe, or action sequence.

Table 4: Re-shuffled games: knowledge that must carry over to new instances.

Game	What the player needs to learn
POISONER	Infer taste constraints from accepted and rejected dishes and apply them to new menus.
SYNERGYCORP	Learn which character traits predict preferred, weak, and taboo actions.
HAUNTED INN	Discover guest–room, adjacency, and count taboos through controlled placements.
ROOMMATES	Infer pairwise and nonlinear grouping rules from room-level feedback.
ROADSIDE OBSERVATORY	Learn social protocols and resource requirements for new encounters.
LOST AND FOUND	Learn how to verify ownership without revealing private information.
MOLA TEA	Infer the adoption chain and when each promotion becomes effective.
PLAGUE	Locate asymptomatic carriers from spread patterns and isolate them early.
BUTTERFLY	Infer inheritance laws to breed new target combinations.
MODEL AUDITOR	Learn how to probe a new stateful machine and predict its outputs.

Table 5: Fixed games: knowledge accumulated about a recurring instance.

Game	What the player needs to learn
PRIMORDIALSOUP	Discover reaction recipes and plan synthesis within a resource budget.
GEMFORGE	Discover multi-tier recipes and manage limiting ingredients.
CATNIP	Learn action–target matches and when to exit for the best reward.
DUNGEON	Remember rooms, keys, hazards, and treasures to refine an escape route.
DREAMGARDEN	Infer delayed ecological effects and plan actions across turns.
ECOSPHERE	Infer food-web, competition, and waste dynamics to plan stocking.
GRAPEVINE	Infer a cascade network and choose effective seed combinations.
CASTAWAY	Remember the map, recipes, hazards, and steps needed to escape.
REDEYE	Combine character evidence and execute the rescue steps in order.
PATCH REALITY	Remember diagnosis–repair mappings, decoys, and prerequisites.

A.2 EVALUATION AND SCORING

We first normalize each episode's raw score, then compute metrics within each game, and finally average across games. This section specifies those steps so that aggregate scores can be reproduced. Human record selection is described in Appendix B.3. Cost calculations are in Appendix B.5.

A.2.1 INSTANCES, EPISODES, AND TRIALS

All configurations use the pinned game instances listed in Table 6. A trial is one sequence of episodes with no experience carried over from another trial. The protocol requires three trials per configuration and game. We evaluate each trial over five episodes for normal games and ten episodes for challenge games. Appendix C reports aggregate results with trial standard deviations. Appendix D gives the corresponding per-game results.

A.2.2 FROM RAW SCORES TO PER-GAME METRICS

Let $x_{g,r,t}$ be the raw score for game g, trial r, and episode t, and let C_g be that game's reference score. We convert it to $z_{g,r,t} = 100x_{g,r,t}/C_g$. This puts different games on a common scale.

Max first takes the peak normalized terminal score $m_{g,r}$ in each trial's scoring window and then averages those peaks across the R_g scored trial trajectories: $\mathrm{Max}_g = R_g^{-1} \sum_{r=1}^{R_g} m_{g,r}$. For example, trial peaks of 40, 60, and 80 give Max $= 60$, not 80. The other metrics use the episode-wise mean of terminal scores, $y_{g,t}$: Mean averages its scores, LG compares its late and early scores, and LS measures its linear trend per episode. Section 4.1 defines the formulas. Appendix B.3 specifies human score construction.

A.2.3 AVERAGING ACROSS GAMES AND PLOTTING CURVES

Each challenge game has weight 3 and each other game has weight 1. Across the 20 games, both groups have total weight 15: $5 \times 3 = 15 \times 1$. For a metric M_g, the aggregate is $\sum_g w_g M_g / \sum_g w_g$ over games with a defined value. The human selection rule gives all four metrics a value in every one of the 20 games.

For aggregate figures only, a game's curve is interpolated to ten equally spaced positions across its scoring window. Thus 100% on the horizontal axis means episode 5 for a normal game and episode 10 for a challenge game. Interpolation is not used for Max, Mean, LG, or LS. All three panels use the same 20 games throughout each curve. Per-game figures show the original episode axis and raw scores.

A.2.4 REFERENCE SCORES

An exact reference is supported by an oracle or proof. A partial reference is the available proved value for the pinned instance, without a complete optimality result. An estimated reference comes from a legal reference search or a score cap. These evidence levels describe the reference, not the agent's knowledge of the game.

Table 6: Per-game reference scores used for normalization and their supporting evidence.

Game	Seed	C	Source	Normalized score
PRIMORDIALSOUP	1	110	exact	$100x_t/110$
GEMFORGE	1	791	exact	$100x_t/791$
CATNIP	1	320	estimated	$100x_t/320$
DUNGEON	2	54	exact	$100x_t/54$
SYNERGYCORP	2	137	exact	$100x_t/137$
DREAMGARDEN	1	6300	estimated	$100x_t/6300$
ECOSPHERE	1	339	exact	$100x_t/339$
GRAPEVINE	1	16	exact	$100x_t/16$
HAUNTED INN	2	200	partial	$100x_t/200$
CASTAWAY	1	241	partial	$100x_t/241$
ROADSIDE OBSERVATORY	2	1000	exact	$100x_t/1000$
LOST AND FOUND	1	1000	exact	$100x_t/1000$
MOLA TEA	1	1000	estimated	$100x_t/1000$
PLAGUE	2	100	exact	$100x_t/100$
PATCH REALITY	1	960	estimated	$100x_t/960$
POISONER	2	100	exact	$100x_t/100$
ROOMMATES	1	7	exact	$100x_t/7$
REDEYE	2	100	exact	$100x_t/100$
BUTTERFLY	1	439	exact	$100x_t/439$
MODEL AUDITOR	1	100	exact	$100x_t/100$

B Additional Results by Research Question

The following subsections follow RQ1–RQ5 in the main text. Each brings together the relevant setup, full results, and supporting interaction evidence. Appendix D provides a separate entry for each game.

B.1 RQ1: How Effectively Do Agents Learn?

This appendix supplements Table 1 with the GPT-5 Mini method comparison and three-backbone averages in Table 7, as well as learning curves and behavioral evidence. Figure 4 compares all four Memory backbones. We also examine interaction records to distinguish repeated success from discovery of the underlying rules.

Table 7: Self-evolving methods with GPT-5 Mini and their average performance across Kimi K3, Opus 5, and GPT-5 Mini. Bold marks the best value within each group. The last six rows average each metric equally across the three backbones.

Backbone	Method	Max	Mean	LG	LS
GPT-5 Mini	Reflexion	**53.5**	34.4	+3.6	+0.97
	ReasoningBank	51.9	32.5	-0.4	+0.00
	Naive	48.7	30.5	-0.1	+0.36
	Memory	48.7	**37.3**	+5.9	+1.56
	EvoTest	47.5	31.8	**+7.4**	**+1.61**
	AWM	47.0	28.9	-2.9	-0.98
3-backbone avg.	Reflexion	**60.0**	41.2	+10.0	+2.42
	EvoTest	58.7	43.7	+6.2	+1.71
	Memory	57.6	**44.5**	**+11.4**	**+3.04**
	ReasoningBank	57.3	39.3	+6.7	+1.82
	Naive	52.3	33.0	-1.6	-0.10
	AWM	51.3	34.6	+0.8	+0.02

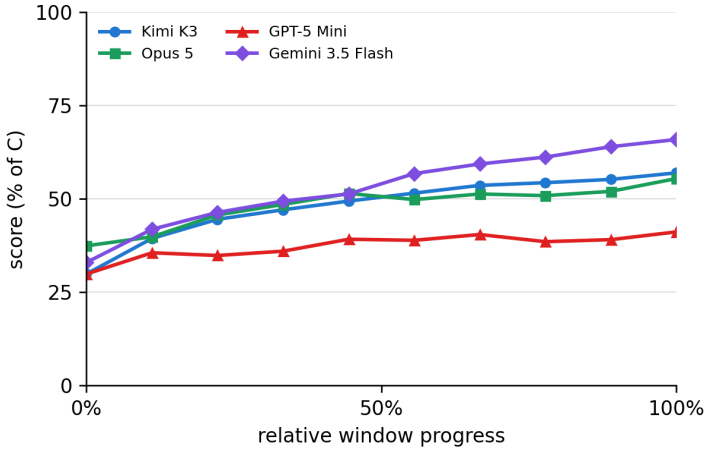


Figure 4: Learning curves for Memory with four backbone models. Scores use the shared normalization and game weights. Relative progress aligns the five- and ten-episode scoring windows.

B.1.1 Evidence of rule discovery

Appendix D reports each game's target definitions and evidence labels. We retain these per-game records without aggregating the labels into overall percentages, because the archived criteria mix

complete-rule discovery with successful execution. Appendix D explains the labels and annotation scope.

B.1.2 Exploration and planning cases

The following cases show how a policy changes across episodes. They use recorded actions, visible narration, and saved notes, not private reasoning. They illustrate possible mechanisms rather than their frequency. These supplementary cases draw on interactions beyond the scoring windows and do not enter the reported scores.

Stopping exploration in GemForge. In OpenCode–Opus 5 trial 1, a failed fusion is followed by the statement "never combine tier-3 gems." The agent then treats its recipe map as complete and repeats a plan scoring 426. This illustrates how an incomplete belief can stop further tests.

Knowing taboos without reserving rooms. OpenCode–Opus 5 trial 1 records both the actor–scholar adjacency taboo and the mirror-room restriction on lantern carriers. Greedy placement leaves no safe room for the last actor. The agent then says "I need to reserve buffer positions proactively" and changes its allocation plan. The scores are 100 and 115, but the guest lists also change: the evidence is the revised allocation policy, not a controlled score improvement.

B.1.3 How processed experience can restrict exploration

One failed fusion becomes a broad rule. EvoTest–Opus 5 trial 3 scores 46, 81, 83, 83, and 83 in GemForge's scoring window. An earlier prompt records that the only fusion tried, jade with opal, produced slag. The episode-5 prompt then instructs the agent never to attempt tier-2 or tier-3 fusion, ruling out combinations it has not tested. This case shows how a processed prompt can generalize beyond the observed failure. It does not establish that compression causes the aggregate difference from Memory.

B.1.4 Using feedback instead of repeating menu positions

In Poisoner, OpenCode–GPT-5 Mini trial 3 (seed 2) serves dish 0 at all three banquets in episodes 2, 3, and 5, even though dishes at that position are both accepted and rejected. Its saved notes describe a policy of always serving the same index. By contrast, OpenCode–Kimi K3 trial 1 (seed 2) compares accepted dishes across episodes 1 and 2 and identifies sourness as irrelevant. In episode 3, an accepted non-hot dish leads it to rule out temperature and then test saltiness. These cases use recorded actions and visible narration within the first five episodes. They illustrate different uses of feedback, not the frequency of either behavior across all trials.

B.2 RQ2: How Does the Harness Affect Learning?

The comparison holds the backbone fixed: OpenCode and Claude Code both use Opus 5, while OpenCode and Codex both use GPT-5.6-SOL. Table 2 reports their scores, and the right-hand panel of each game entry in Appendix D shows their learning curves.

Each trial starts without experience from another trial. Within a trial, earlier episode interactions remain available in the ongoing session, subject to the harness's context handling. Claude Code disables persistent auto-memory. Codex uses `--ephemeral`. OpenCode starts a new session rather than resuming an unrelated one. These settings separate trials without discarding all within-trial experience. Other reasoning settings use the CLI or provider defaults.

The per-game evidence matrices in Appendix D include the four harness–model configurations used in this comparison. The two OpenCode reference rows appear in each matrix's backbone block. These labels are reported only for individual games, not as overall percentages.

B.3 RQ3: How Do Humans and Agents Learn Differently?

Humans use the same games and interface as agents, with 21–30 participants per game. We compare both their scores and their recorded behavior. This subsection specifies participant selection, score construction, and the behavioral statistics used in the main text.

Participants and incentives. We recruited college students to play the games. Participants did not use AI tools or code during gameplay. They were paid for participating, with additional rewards for achieving the highest scores.

B.3.1 SELECTING HUMAN RECORDS

For each game, we select the Top-1, Top-3, and Top-5 players by their highest observed score within the scoring window. All records must have a completed first episode. If Max is tied, we prefer records with at least two scored episodes, then break remaining ties by the first episode attaining Max, Mean, LG, and LS. This tie-breaking rule supports analysis of learning while preserving the selected Max values.

Observed scores take precedence over replay values. Re-shuffled-game metrics use observed episode scores at their original episode indices.

B.3.2 CALCULATING SCORES AND LEARNING CURVES

Max averages the selected players' individual peaks. The other metrics use their episode-wise mean score curve. LG compares the last and first $\min(3, \lfloor n/2 \rfloor)$ scores, LS fits a line at the original episode indices. The selection rule ensures at least two scored positions in every selected game-level mean curve. All four metrics cover all 20 games.

For reference, averaging across all eligible participants rather than selecting the Top-k records gives Max $= 43.6$, Mean $= 31.2$, LG $= +10.4$, and LS $= +2.72$.

Figure 2(C) compares human and agent learning curves across all 20 games. The score tables also cover all 20 games.

B.3.3 COMPARING RECORDED BEHAVIOR

The behavioral comparison uses Human Top-3 and the six Opus 5 methods on fixed games, within the same five- or ten-episode scoring windows. Human actions and scores come directly from completed episodes. Prior-best replay values do not contribute to the behavioral statistics. We pool the three agent trials for Naive, Memory, Reflexion, EvoTest, ReasoningBank, and AWM.

Action repetition. We compare the ordered action sequences in consecutive episodes using SequenceMatcher with `autojunk=False`. Similarity is $2M/(|a| + |b|)$, where M is the total length of matched blocks. The mean is 0.5379 across 150 human episode pairs and 0.6575 across 918 agent pairs. This measures repeated actions, not whether the actions represent the same strategy or good exploration.

Recovery after a setback. For three consecutive observed episodes, we first identify a score decrease, $s_t < s_{t-1}$, then ask whether the next score exceeds every earlier observed score in that record. This happens after 13 of 39 human decreases (33.3%) and 52 of 238 agent decreases (21.8%). Both statistics average observations without game weighting. Repeated observations from the same participant or run are not independent, so these are descriptive comparisons rather than causal tests of willingness to explore.

Record selection. Human behavior is measured from completed episodes with recorded actions and scores. Each agent trial must contain both actions and scores throughout the scoring window. If multiple records match a trial, we select the one with the most recorded scores, using the most recently modified record to break ties.

Table 8: Human and agent results ranked by Max. All four metrics use the same 20 games and challenge weights. Human selection and score construction are specified above. Bold marks the best value in each column.

System	Model	Max	Mean	LG	LS
Human Top-1	Human	**84.3**	53.5	+21.6	**+7.33**
Claude Code	Claude Opus 5	81.8	**62.3**	**+27.1**	+6.59
Human Top-3	Human	79.1	50.1	+18.9	+5.54
Human Top-5	Human	75.1	47.1	+17.1	+4.62
OpenCode	Claude Opus 5	74.5	53.8	+21.3	+5.13
Codex	GPT-5.6-SOL	74.3	54.8	+22.2	+5.66
OpenCode	Qwen 3.8 Max	72.5	51.1	+25.2	+6.30
OpenCode	Gemini 3.5 Flash	72.0	52.2	+19.8	+5.37
OpenCode	Kimi K3	68.5	50.2	+17.1	+4.24
EvoTest	Claude Opus 5	66.1	51.3	+9.4	+2.52
Reflexion	Claude Opus 5	64.8	48.8	+11.6	+2.93
MEMORY	Kimi K3	63.3	48.0	+16.9	+4.69
EvoTest	Kimi K3	62.4	48.0	+1.8	+1.01
Reflexion	Kimi K3	61.6	40.4	+14.8	+3.35
MEMORY	Claude Opus 5	60.9	48.2	+11.4	+2.87
ReasoningBank	Claude Opus 5	60.2	45.3	+12.3	+2.85
ReasoningBank	Kimi K3	59.9	40.1	+8.3	+2.59
OpenCode	Gemini 3.7 Flash	58.3	39.9	+10.7	+2.47
OpenCode	GPT-5.6-SOL	57.0	41.1	+14.5	+3.71
AWM	Kimi K3	56.1	37.0	+4.8	+0.62
OpenCode	Claude Sonnet 5	55.9	35.6	+15.7	+3.72
NAIVE	Kimi K3	54.7	31.9	+0.3	+0.42
OpenCode	DeepSeek V4 Pro	54.2	38.2	+14.7	+3.66
Reflexion	GPT-5 Mini	53.5	34.4	+3.6	+0.97
NAIVE	Claude Opus 5	53.4	36.6	-5.1	-1.08
ReasoningBank	GPT-5 Mini	51.9	32.5	-0.4	+0.00
AWM	Claude Opus 5	50.7	37.8	+0.7	+0.43
NAIVE	GPT-5 Mini	48.7	30.5	-0.1	+0.36
MEMORY	GPT-5 Mini	48.7	37.3	+5.9	+1.56
EvoTest	GPT-5 Mini	47.5	31.8	+7.4	+1.61
AWM	GPT-5 Mini	47.0	28.9	-2.9	-0.98
Human Mean	Human	43.6	31.2	+10.4	+2.72
OpenCode	GPT-5 Mini	34.0	22.3	+9.2	+1.83

B.3.4 CASES OF EXECUTION AND CONTINUED INVESTIGATION

Combining clues in RedEye. OpenCode–Opus 5 trial 1 first fails, then combines documentary evidence with information about the owner, engineer, and coast guard. Its notes specify the rescue order, and episodes 9 and 10 score 100. This illustrates executing a multi-step solution in one run.

Using programs in Model Auditor. OpenCode–Opus 5 trial 1 switches to programmatic coefficient search in episode 4 and adds Gaussian elimination in episode 5 to test whether its fitted model is unique. Both episodes score 100. The fitted models are recorded hypotheses, not necessarily the true hidden implementation.

Reduced human probing. In Model Auditor, the two highest-Max human records use 40, 17, 17, 1, 0 and 17, 40, 0, 0, 0 probe actions across episodes 1–5. These counts exclude resets and predictions. They show reduced testing, but do not identify whether the cause is effort, confidence, frustration, or something else.

B.4 RQ4: How Well Does Learned Knowledge Transfer to New Game Instances?

Matched-game protocol. Table 3 reports the OpenCode backbones and deployed agents, while Table 9 reports the six methods with Opus 5. The methods include NAIVE, which retains no cross-episode memory. All configurations are compared on matched games from the ten-game re-shuffled set, using the same rule seeds in both conditions. Each same-instance control trial repeats instance 1 for ten episodes in an independent native session. The re-shuffled condition reuses archived ten-episode trials with changing instances. LS is fitted to the mean normalized trajectory within each game, then aggregated with the shared challenge weights. This supplementary comparison uses the full ten-episode window for both conditions, rather than the five-episode reporting window used for normal games elsewhere.

Included trajectories. Three independent trials are required per game and configuration. We calculate LS from complete ten-episode trajectories and use the same matched games in both conditions of each row. The Qwen row covers nine games. All other rows cover ten.

Table 9: Learning slopes for six methods with Opus 5 when the same instance repeats (Control) or changes across episodes (Re-shuf.), with the rules held fixed. LS drop is Control minus Re-shuf.

Method	Control	Re-shuf.	LS drop
NAIVE	-0.49	-0.41	-0.08
MEMORY	3.76	2.88	0.88
Reflexion	3.33	2.70	0.63
ReasoningBank	2.48	2.09	0.40
AWM	-0.11	0.26	-0.37
EvoTest	1.92	1.07	0.85

Budget sensitivity. Table 10 compares the standard-budget control LS with a calculation that also includes nine extended-budget control trajectories. Both columns use the same games, so differences do not come from changes in which games are included.

Table 10: Learning slopes in the same-instance control, excluding or including extended-budget trials. Both estimates use the same games. The extended-budget results are supplementary and do not replace the standard-budget estimates in Table 3.

System / model	Control LS	Control LS incl. extra
OpenCode Claude Opus 5	4.10	4.07
OpenCode Qwen 3.8 Max	3.16	3.34
OpenCode Kimi K3	4.19	4.33

B.4.1 Applying a rule to a new instance

The following cases illustrate rule reuse rather than estimate its frequency or explain aggregate differences in LS.

Matching a rule to a new menu. OpenCode–Opus 5 trial 1 recognizes that Poisoner's menus change while the king's preferences stay fixed. It compares accepted and rejected dishes, then selects dishes matching an inferred five-feature condition. It repeatedly scores 100 by applying the condition to new menus. This supplementary case draws on interactions beyond the scoring window and illustrates reuse of a condition rather than a dish position.

Using a learned clue rather than probing again. In SYNERGYCORP with seed 2, trial 1, episode 10, three OpenCode agents start from the same game instance under a 30-action budget. Opus 5 uses a mapping learned in earlier episodes from visible desk traits to preferred actions, so it often selects an effective action immediately and scores 137. Sonnet 5 tests actions to find each new boss's preferences and scores 67. Gemini 3.5 Flash uses one action as a diagnostic probe for each boss and scores 84. The records illustrate how using an observed clue can save actions.

B.5 RQ5: Do Higher Inference Costs Lead to Better Performance?

We estimate cost from recorded token usage so that performance can be compared with the resources used to obtain it. The estimate distinguishes fresh input, cached input, and output tokens. It is not a billing record.

B.5.1 Usage and pricing

We sum OpenCode's per-step token reports, use Claude Code's run-level usage summaries, and read Codex's final turn-completion report. For self-evolving methods, we use logged input and output counts and reconstruct cache reuse from repeated prompt prefixes, using a five-minute cache lifetime and each model's minimum cacheable length. Older runs use the observed prefix split for the same method. The EvoTest–Kimi K3 estimate averages the available token logs.

Prices use the 30 August 2026 snapshot (OpenRouter, 2026, Anthropic, 2026), with separate rates for fresh input, cache reads, cache writes, and output. We report average cost and token usage per episode.

Table 11 separates input from generation for the Kimi K3 comparison. Input includes fresh input, cache reads, and cache writes. Generation includes visible output and reasoning. Separately reported reasoning tokens are added without double counting. Each token and call total is divided by the number of recorded episodes. The Updates column is the portion of generation spent on cross-episode reflection, strategy evolution, memory extraction, or workflow induction.

Table 11: Kimi K3 token and call breakdown per episode. Generated tokens include reasoning. Updates are a subset of Generated. M and K denote millions and thousands of tokens. USD is the estimated total cost per episode.

System	Max	Input (M)	Generated (K)	Updates (K)	Calls	USD
OpenCode	68.5	0.55	7.2	–	8.1	0.313
Memory	63.3	1.26	6.7	–	27.0	0.586
Reflexion	61.6	0.38	9.1	2.6	29.2	0.300
EvoTest	62.4	0.14	24.6	11.5	24.1	0.658
ReasoningBank	59.9	0.34	12.3	3.0	29.8	0.340
AWM	56.1	0.26	7.3	1.8	28.4	0.224

The savings differ by comparison. Memory supplies retained histories with each action request, giving it the largest input volume. EvoTest, Reflexion, ReasoningBank, and AWM generate additional updates between episodes. EvoTest's updates alone use 11.5K generated tokens per episode, exceeding OpenCode's total generation of 7.2K tokens. OpenCode uses fewer model calls: one GemForge trace batches nine game actions into a single call and returns only the last 60 output lines. Under the pricing snapshot, Kimi K3 generation costs $15 per million tokens versus $3 for fresh input, yielding generation costs of $0.108 for OpenCode, $0.369 for EvoTest, and $0.184 for ReasoningBank per episode. Reflexion and AWM have lower total estimated costs than OpenCode, alongside lower Max scores.

Table 12: Estimated cost comparison using average cost and token usage per episode. Ratios divide Max by the displayed average cost or token usage per episode. Bold efficiency values are selected among configurations with Max ≥ 70.

System	Model	Max	Mtok/ep.	USD/ep.	Max/(Mtok/ep.)	Max/(USD/ep.)
Claude Code	Claude Opus 5	**81.8**	0.71	0.647	115.7	126.4
OpenCode	Claude Opus 5	74.5	0.54	2.57	138.3	29.0
Codex	GPT-5.6-SOL	74.3	0.49	0.144	151.1	**516.6**
OpenCode	Qwen 3.8 Max	72.5	0.51	0.225	140.9	322.4
OpenCode	Gemini 3.5 Flash	72.0	1.63	0.598	44.3	120.5
OpenCode	Kimi K3	68.5	0.56	0.313	122.6	218.5
EvoTest	Claude Opus 5	66.1	0.21	1.17	318.5	56.6
Reflexion	Claude Opus 5	64.8	0.52	0.793	125.3	81.8
MEMORY	Kimi K3	63.3	1.27	0.586	50.0	107.9
EvoTest	Kimi K3	62.4	0.16	0.658	**378.7**	94.8
Reflexion	Kimi K3	61.6	0.39	0.300	158.6	205.5
MEMORY	Claude Opus 5	60.9	3.52	3.09	17.3	19.7
ReasoningBank	Claude Opus 5	60.2	0.56	0.858	108.3	70.2
ReasoningBank	Kimi K3	59.9	0.35	0.340	170.7	176.2
OpenCode	Gemini 3.7 Flash	58.3	0.33	0.052	175.5	1124.1
OpenCode	GPT-5.6-SOL	57.0	0.32	0.119	180.1	480.1
AWM	Kimi K3	56.1	0.26	0.224	213.3	250.8
OpenCode	Claude Sonnet 5	55.9	0.73	1.33	77.1	41.9
NAIVE	Kimi K3	54.7	0.34	0.252	162.2	217.3
OpenCode	DeepSeek V4 Pro	54.2	0.23	0.021	234.3	2526.9
Reflexion	GPT-5 Mini	53.5	0.23	0.041	235.4	1319.6
NAIVE	Claude Opus 5	53.4	0.71	0.800	75.6	66.8
ReasoningBank	GPT-5 Mini	51.9	0.52	0.090	100.2	575.7
AWM	Claude Opus 5	50.7	0.33	0.423	153.2	119.8
NAIVE	GPT-5 Mini	48.7	0.17	0.037	294.6	1313.3
MEMORY	GPT-5 Mini	48.7	0.89	0.051	54.6	946.6
EvoTest	GPT-5 Mini	47.5	0.17	0.063	282.9	748.0
AWM	GPT-5 Mini	47.0	0.19	0.036	252.0	1323.4
OpenCode	GPT-5 Mini	34.0	0.10	0.0063	345.8	5437.8

C AGGREGATE RESULTS AND TRIAL VARIABILITY

We report trial variability for all 29 agent configurations and all four metrics. Tables 13–15 give the weighted results over all games and separately over fixed and re-shuffled games. Per-game results appear alongside each game's curves and rule evidence in Appendix D. Point estimates retain the normalization, scoring windows, and aggregation used elsewhere in the paper. The standard deviations (SDs) describe variation across independent trials, not variation across episodes or game difficulty.

Within each game, we compute each metric separately for each complete trial. Max takes that trial's highest score, Mean averages its scores, LG compares its first and last k_g episodes, and LS fits its score trend against the original episode indices. For trial-level metric values $M_{g,r}$, the sample variance is $s_g^2 = (n_g - 1)^{-1} \sum_{r=1}^{n_g} (M_{g,r} - \bar{M}_g)^2$, and the tables report $\pm s_g$. With complete trajectories, averaging these trial-level metrics reproduces the paper's point estimates, apart from rounding of archived episode means.

Across games, we propagate within-game variation using the paper's weights. Writing $a_g = w_g / \sum_h w_h$, the reported aggregate SD is $s_{\mathrm{agg}} = \sqrt{\sum_g a_g^2 s_g^2}$. Under independent runs across games, this estimates the variability of a benchmark score formed from one trial per game. It does not pair arbitrary trial indices across games or treat differences between games as trial noise. This is an SD, not the standard error or a confidence interval of the reported multi-trial mean. The instance panel and reference scores are held fixed.

Each configuration–game cell contains three complete scored trajectories. This applies to all 580 cells in these tables, with no missing episode scores imputed. The small number of trials limits the precision of the SD estimates.

Human Top-k groups are not independent repeat trials. They select participants by achieved performance, so their descriptive results remain in Appendix B.3. We do not attach agent-trial SDs to those selected groups or interpret them as population-level uncertainty.

C.1 ALL 20 GAMES

Table 13: Performance and learning over 20 games. Values are point estimates ± standard deviations estimated from within-game trial variability.

Backbone	Method	Max ↑	Mean ↑	LG ↑	LS ↑
Claude Opus 5	OpenCode	74.5 ± 2.6	53.8 ± 2.2	$+21.3 \pm 3.1$	$+5.13 \pm 0.71$
Gemini 3.5 Flash	OpenCode	72.0 ± 3.9	52.2 ± 3.3	$+19.8 \pm 5.7$	$+5.37 \pm 1.26$
GPT-5.6-SOL	OpenCode	57.0 ± 3.6	41.1 ± 2.4	$+14.5 \pm 3.9$	$+3.71 \pm 1.06$
Claude Sonnet 5	OpenCode	55.9 ± 3.7	35.6 ± 2.6	$+15.7 \pm 4.1$	$+3.72 \pm 0.82$
DeepSeek V4 Pro	OpenCode	54.2 ± 3.7	38.2 ± 4.0	$+14.7 \pm 3.6$	$+3.66 \pm 1.09$
Kimi K3	OpenCode	68.5 ± 3.8	50.2 ± 3.6	$+17.1 \pm 3.2$	$+4.24 \pm 0.75$
Qwen 3.8 Max	OpenCode	72.5 ± 4.3	51.1 ± 3.1	$+25.2 \pm 4.5$	$+6.30 \pm 1.18$
GPT-5 Mini	OpenCode	34.0 ± 3.2	22.3 ± 2.6	$+9.2 \pm 3.9$	$+1.83 \pm 0.91$
Gemini 3.7 Flash	OpenCode	58.3 ± 2.5	39.9 ± 2.9	$+10.7 \pm 3.4$	$+2.47 \pm 0.84$
Claude Opus 5	Claude Code	81.8 ± 4.4	62.3 ± 3.0	$+27.1 \pm 5.2$	$+6.59 \pm 1.05$
GPT-5.6-SOL	Codex	74.3 ± 4.1	54.8 ± 2.9	$+22.2 \pm 4.8$	$+5.66 \pm 1.07$
Kimi K3	NAIVE	54.7 ± 3.1	31.9 ± 1.4	$+0.3 \pm 4.3$	$+0.42 \pm 1.05$
Kimi K3	Reflexion	61.6 ± 3.2	40.4 ± 2.8	$+14.8 \pm 4.7$	$+3.35 \pm 1.19$
Kimi K3	EvoTest	62.4 ± 3.5	48.0 ± 2.6	$+1.8 \pm 4.3$	$+1.01 \pm 1.05$
Kimi K3	MEMORY	63.3 ± 4.0	48.0 ± 3.0	$+16.9 \pm 4.0$	$+4.69 \pm 0.96$
Kimi K3	ReasoningBank	59.9 ± 3.6	40.1 ± 2.0	$+8.3 \pm 3.4$	$+2.59 \pm 0.93$
Kimi K3	AWM	56.1 ± 4.2	37.0 ± 2.4	$+4.8 \pm 4.4$	$+0.62 \pm 1.23$
Opus 5	NAIVE	53.4 ± 2.5	36.6 ± 0.8	-5.1 ± 2.8	-1.08 ± 0.65
Opus 5	Reflexion	64.8 ± 2.3	48.8 ± 2.8	$+11.6 \pm 2.5$	$+2.93 \pm 0.64$
Opus 5	EvoTest	66.1 ± 2.8	51.3 ± 2.3	$+9.4 \pm 3.5$	$+2.52 \pm 0.99$
Opus 5	MEMORY	60.9 ± 3.9	48.2 ± 2.9	$+11.4 \pm 3.7$	$+2.87 \pm 0.84$
Opus 5	ReasoningBank	60.2 ± 4.8	45.3 ± 3.1	$+12.3 \pm 3.9$	$+2.85 \pm 0.63$
Opus 5	AWM	50.7 ± 2.6	37.8 ± 2.5	$+0.7 \pm 2.3$	$+0.43 \pm 0.55$
GPT-5 Mini	NAIVE	48.7 ± 2.5	30.5 ± 1.4	-0.1 ± 2.8	$+0.36 \pm 0.68$
GPT-5 Mini	Reflexion	53.5 ± 4.3	34.4 ± 2.6	$+3.6 \pm 3.5$	$+0.97 \pm 0.93$
GPT-5 Mini	EvoTest	47.5 ± 3.6	31.8 ± 1.9	$+7.4 \pm 2.6$	$+1.61 \pm 0.63$
GPT-5 Mini	MEMORY	48.7 ± 3.7	37.3 ± 2.6	$+5.9 \pm 3.4$	$+1.56 \pm 0.82$
GPT-5 Mini	ReasoningBank	51.9 ± 3.4	32.5 ± 1.9	-0.4 ± 3.6	$+0.00 \pm 1.03$
GPT-5 Mini	AWM	47.0 ± 4.3	28.9 ± 2.2	-2.9 ± 2.5	-0.98 ± 0.52

C.2 FIXED GAMES

Table 14: Performance and learning over 10 fixed games. Values are point estimates ± standard deviations estimated from within-game trial variability.

Backbone	Method	Max ↑	Mean ↑	LG ↑	LS ↑
Claude Opus 5	OpenCode	75.8 ± 2.4	55.2 ± 2.0	$+19.9 \pm 4.4$	$+3.97 \pm 0.92$
Gemini 3.5 Flash	OpenCode	71.3 ± 5.7	51.2 ± 3.9	$+19.4 \pm 7.7$	$+5.02 \pm 1.50$
GPT-5.6-SOL	OpenCode	57.1 ± 5.3	41.9 ± 3.5	$+12.8 \pm 4.4$	$+2.09 \pm 1.14$
Claude Sonnet 5	OpenCode	60.8 ± 4.5	43.3 ± 3.6	$+17.1 \pm 5.6$	$+3.84 \pm 1.22$
DeepSeek V4 Pro	OpenCode	54.0 ± 4.9	44.2 ± 5.4	$+10.1 \pm 3.4$	$+2.32 \pm 1.04$
Kimi K3	OpenCode	64.3 ± 3.2	48.2 ± 3.0	$+15.8 \pm 4.9$	$+3.78 \pm 0.87$
Qwen 3.8 Max	OpenCode	75.1 ± 5.6	54.0 ± 2.8	$+26.7 \pm 5.7$	$+5.69 \pm 1.45$
GPT-5 Mini	OpenCode	33.1 ± 3.7	24.3 ± 3.3	$+7.8 \pm 4.4$	$+1.68 \pm 0.94$
Gemini 3.7 Flash	OpenCode	58.7 ± 3.6	37.8 ± 4.4	$+15.5 \pm 5.1$	$+2.68 \pm 1.00$
Claude Opus 5	Claude Code	83.4 ± 6.7	63.4 ± 3.7	$+28.3 \pm 8.1$	$+6.05 \pm 1.52$
GPT-5.6-SOL	Codex	73.2 ± 6.4	55.7 ± 3.8	$+22.7 \pm 7.2$	$+4.84 \pm 1.49$
Kimi K3	NAIVE	49.7 ± 3.5	30.8 ± 1.2	-0.2 ± 3.8	$+0.33 \pm 0.78$
Kimi K3	Reflexion	57.0 ± 4.0	37.6 ± 2.1	$+13.8 \pm 5.3$	$+2.70 \pm 1.21$
Kimi K3	EvoTest	55.1 ± 5.3	44.8 ± 3.9	$+5.1 \pm 5.3$	$+1.77 \pm 0.98$
Kimi K3	MEMORY	56.7 ± 4.3	42.8 ± 2.7	$+15.8 \pm 5.9$	$+4.04 \pm 1.32$
Kimi K3	ReasoningBank	55.1 ± 3.8	41.4 ± 2.5	$+9.0 \pm 4.0$	$+2.00 \pm 0.93$
Kimi K3	AWM	49.0 ± 5.4	34.6 ± 3.0	$+8.4 \pm 6.2$	$+1.72 \pm 1.64$
Opus 5	NAIVE	49.1 ± 3.4	34.3 ± 1.1	-4.0 ± 4.8	-0.41 ± 1.09
Opus 5	Reflexion	55.5 ± 2.9	42.0 ± 4.3	$+4.8 \pm 2.7$	$+1.52 \pm 0.77$
Opus 5	EvoTest	61.3 ± 4.8	48.9 ± 3.9	$+11.3 \pm 4.0$	$+2.19 \pm 0.99$
Opus 5	MEMORY	56.9 ± 3.6	47.3 ± 2.4	$+8.8 \pm 5.2$	$+2.14 \pm 1.18$
Opus 5	ReasoningBank	57.1 ± 6.8	45.3 ± 3.8	$+13.4 \pm 4.9$	$+2.87 \pm 0.71$
Opus 5	AWM	47.6 ± 4.4	35.5 ± 4.4	$+3.2 \pm 3.4$	$+1.06 \pm 0.77$
GPT-5 Mini	NAIVE	43.9 ± 2.5	32.1 ± 1.1	$+0.7 \pm 2.0$	$+0.40 \pm 0.48$
GPT-5 Mini	Reflexion	44.3 ± 6.6	31.3 ± 3.5	$+1.4 \pm 4.8$	$+0.26 \pm 1.03$
GPT-5 Mini	EvoTest	41.1 ± 5.3	33.1 ± 2.2	$+3.5 \pm 2.2$	$+1.08 \pm 0.58$
GPT-5 Mini	MEMORY	44.8 ± 4.9	38.3 ± 3.4	$+6.9 \pm 5.1$	$+1.36 \pm 0.86$
GPT-5 Mini	ReasoningBank	45.8 ± 2.8	31.0 ± 2.2	$+1.3 \pm 3.8$	$+0.54 \pm 0.91$
GPT-5 Mini	AWM	42.3 ± 6.7	29.7 ± 3.4	$+0.6 \pm 3.5$	-0.30 ± 0.74

C.3 RE-SHUFFLED GAMES

Table 15: Performance and learning over 10 re-shuffled games. Values are point estimates ± standard deviations estimated from within-game trial variability.

Backbone	Method	Max ↑	Mean ↑	LG ↑	LS ↑
Claude Opus 5	OpenCode	73.1 ± 4.8	52.2 ± 4.1	$+22.9 \pm 4.3$	$+6.45 \pm 1.10$
Gemini 3.5 Flash	OpenCode	72.8 ± 5.3	53.4 ± 5.5	$+20.3 \pm 8.4$	$+5.77 \pm 2.08$
GPT-5.6-SOL	OpenCode	56.9 ± 5.0	40.3 ± 3.4	$+16.4 \pm 6.6$	$+5.56 \pm 1.85$
Claude Sonnet 5	OpenCode	50.4 ± 6.0	26.8 ± 3.8	$+14.1 \pm 6.1$	$+3.58 \pm 1.08$
DeepSeek V4 Pro	OpenCode	54.3 ± 5.6	31.3 ± 5.8	$+19.9 \pm 6.8$	$+5.20 \pm 2.01$
Kimi K3	OpenCode	73.3 ± 7.3	52.4 ± 6.9	$+18.6 \pm 3.8$	$+4.77 \pm 1.27$
Qwen 3.8 Max	OpenCode	69.6 ± 6.6	47.9 ± 6.0	$+23.4 \pm 7.0$	$+7.00 \pm 1.90$
GPT-5 Mini	OpenCode	35.1 ± 5.4	20.0 ± 4.2	$+10.8 \pm 6.7$	$+2.00 \pm 1.63$
Gemini 3.7 Flash	OpenCode	57.8 ± 3.4	42.3 ± 3.6	$+5.3 \pm 4.3$	$+2.22 \pm 1.38$
Claude Opus 5	Claude Code	79.9 ± 5.6	61.0 ± 4.8	$+25.7 \pm 6.3$	$+7.20 \pm 1.45$
GPT-5.6-SOL	Codex	75.5 ± 4.9	53.8 ± 4.5	$+21.6 \pm 6.3$	$+6.60 \pm 1.53$
Kimi K3	NAIVE	60.5 ± 5.3	33.1 ± 2.7	$+0.9 \pm 8.2$	$+0.52 \pm 2.07$
Kimi K3	Reflexion	66.8 ± 5.0	43.7 ± 5.4	$+16.0 \pm 8.1$	$+4.10 \pm 2.15$
Kimi K3	EvoTest	70.7 ± 4.2	51.6 ± 3.3	-1.9 ± 7.0	$+0.15 \pm 1.94$
Kimi K3	MEMORY	70.7 ± 7.1	54.0 ± 5.6	$+18.2 \pm 5.1$	$+5.44 \pm 1.39$
Kimi K3	ReasoningBank	65.3 ± 6.2	38.7 ± 3.3	$+7.5 \pm 5.8$	$+3.26 \pm 1.68$
Kimi K3	AWM	64.3 ± 6.6	39.8 ± 4.0	$+0.6 \pm 6.3$	-0.63 ± 1.84
Opus 5	NAIVE	58.4 ± 3.7	39.3 ± 1.3	-6.3 ± 2.5	-1.84 ± 0.60
Opus 5	Reflexion	75.5 ± 3.7	56.6 ± 3.3	$+19.4 \pm 4.4$	$+4.53 \pm 1.06$
Opus 5	EvoTest	71.7 ± 2.3	54.0 ± 2.4	$+7.3 \pm 5.9$	$+2.89 \pm 1.80$
Opus 5	MEMORY	65.5 ± 7.3	49.4 ± 5.6	$+14.3 \pm 5.1$	$+3.70 \pm 1.20$
Opus 5	ReasoningBank	63.8 ± 6.9	45.4 ± 5.1	$+11.0 \pm 6.1$	$+2.83 \pm 1.09$
Opus 5	AWM	54.2 ± 2.4	40.4 ± 1.7	-2.2 ± 2.8	-0.30 ± 0.79
GPT-5 Mini	NAIVE	54.3 ± 4.4	28.6 ± 2.7	-1.0 ± 5.5	$+0.31 \pm 1.35$
GPT-5 Mini	Reflexion	63.9 ± 5.4	38.0 ± 3.8	$+6.2 \pm 5.2$	$+1.77 \pm 1.61$
GPT-5 Mini	EvoTest	54.7 ± 4.8	30.3 ± 3.1	$+11.9 \pm 5.0$	$+2.20 \pm 1.18$
GPT-5 Mini	MEMORY	53.2 ± 5.5	36.1 ± 4.1	$+4.8 \pm 4.4$	$+1.78 \pm 1.46$
GPT-5 Mini	ReasoningBank	58.8 ± 6.6	34.3 ± 3.3	-2.3 ± 6.4	-0.61 ± 1.94
GPT-5 Mini	AWM	52.4 ± 5.3	27.9 ± 2.6	-6.8 ± 3.5	-1.76 ± 0.72

D PER-GAME RESULTS AND RULE-DISCOVERY EVIDENCE

Each entry brings together a game's learning task, raw-score curves, a single performance table with trial variability, and archived rule-discovery annotations. Scores and supporting evidence can therefore be checked in one place.

Reading the curves. Curves show episode-wise mean raw scores across trials. Their left, center, and right panels compare backbones, self-evolving methods, and harnesses. Each method curve averages its Kimi K3, Opus 5, and GPT-5 Mini curves.

Reading the performance tables. All four metrics use normalized scores, with agent results reported as point estimates ± trial SD (Appendix C). Max averages individual trial peaks. Mean averages the episode-wise trial mean. First is the mean episode at which each trial first reaches its peak. The selected Human Top-1 record retains Max, Mean, and First. The symbol – marks quantities not reported for this selected record, not zero. Curves remain in raw units, and each entry gives the reference score needed to convert between scales.

Reading the rule-discovery labels. Each game defines local targets T1–Tk. S means at least one available trial provides evidence for the whole listed criterion, P means partial evidence without a complete trial, and – means insufficient evidence. S does not mean that all trials satisfy the target. We report these labels only within each game. The different archived criteria are not aggregated into overall percentages.

Scope of the rule annotations. The labels provide supplementary qualitative evidence from interaction records. They do not measure rule discovery within the scoring window. The first five games use complete-rule criteria. Criteria for the other games also include successful execution supported by observed performance. Terse logs, especially for GPT-5.6-SOL, can leave a target unverified even when a successful policy uses it.

These annotations describe evidence in interaction records, not additional score measurements. A high score alone does not establish every hidden rule. For Haunted Inn and Poisoner, the entries identify which seed-dependent rule and policy labels use the same seed as the scored trials.

D.1 PRIMORDIALSOUP

Learning task. Discover reaction recipes and plan synthesis within a resource budget. Fixed game, scored seed 1, first 5 episodes, reference score $C = 110$.

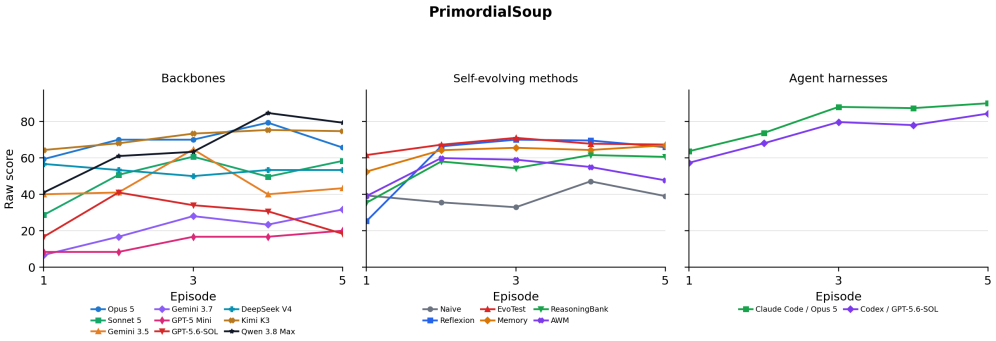


Figure 5: PrimordialSoup: raw-score learning curves within the scoring window.

Table 16: PrimordialSoup: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	1	100.0	100.0	–	–
Codex + GPT-5.6-SOL	4	89.1 ± 18.9	66.8 ± 15.2	$+16.8 \pm 33.3$	$+5.82 \pm 10.89$
Claude Code + Opus 5	3.7	81.8 ± 15.7	73.2 ± 10.0	$+18.2 \pm 17.3$	$+6.03 \pm 5.26$
OpenCode + Kimi K3	4.7	78.5 ± 19.1	64.7 ± 12.0	$+8.0 \pm 16.2$	$+2.55 \pm 6.27$
OpenCode + Qwen 3.8 Max	4	77.0 ± 16.0	59.9 ± 10.8	$+28.2 \pm 13.9$	$+9.12 \pm 4.15$
Reflexion + Opus 5	4.3	74.5 ± 15.0	53.1 ± 5.1	$+23.2 \pm 14.6$	$+8.64 \pm 5.21$
OpenCode + Opus 5	4	72.1 ± 1.0	62.6 ± 2.7	$+7.1 \pm 8.5$	$+2.00 \pm 3.06$
EvoTest + Opus 5	3	69.7 ± 2.6	65.5 ± 1.2	$+4.8 \pm 2.2$	$+2.06 \pm 1.00$
Memory + Kimi K3	3.3	69.7 ± 2.6	64.4 ± 1.6	$+8.8 \pm 3.5$	$+3.09 \pm 1.18$
EvoTest + Kimi K3	2.3	68.2 ± 0.0	65.8 ± 1.1	$+6.1 \pm 2.7$	$+2.30 \pm 0.93$
ReasoningBank + Kimi K3	4.7	68.2 ± 0.0	54.6 ± 3.7	$+25.6 \pm 14.7$	$+9.33 \pm 4.78$
Reflexion + Kimi K3	2.7	67.6 ± 1.0	56.4 ± 6.7	$+24.4 \pm 15.1$	$+9.67 \pm 5.74$
ReasoningBank + Opus 5	4.3	67.0 ± 2.1	53.5 ± 7.6	$+19.2 \pm 12.7$	$+6.85 \pm 4.02$
AWM + Opus 5	4.3	66.1 ± 3.7	51.3 ± 8.7	$+8.3 \pm 14.0$	$+2.61 \pm 4.30$
AWM + Kimi K3	3	64.8 ± 3.7	45.9 ± 17.4	$+2.4 \pm 12.9$	$+2.15 \pm 3.64$
Memory + Opus 5	3	63.6 ± 15.7	49.7 ± 18.2	$+2.4 \pm 24.9$	$+1.48 \pm 9.05$
Reflexion + GPT-5 Mini	3	62.7 ± 8.7	52.5 ± 11.2	$+12.3 \pm 8.4$	$+4.76 \pm 3.09$
EvoTest + GPT-5 Mini	3.3	62.1 ± 5.6	51.4 ± 5.4	-2.3 ± 8.9	-1.06 ± 2.63
OpenCode + Sonnet 5	3.3	62.1 ± 6.9	45.1 ± 1.9	$+13.0 \pm 22.5$	$+5.30 \pm 6.32$
Memory + GPT-5 Mini	3.3	61.5 ± 6.4	56.9 ± 3.0	$+8.6 \pm 15.4$	$+3.33 \pm 5.94$
OpenCode + Gemini 3.5 Flash	3.7	60.3 ± 10.8	41.6 ± 10.8	$+1.1 \pm 22.1$	$+0.51 \pm 7.54$
Naive + Kimi K3	3	57.9 ± 4.3	40.5 ± 2.7	$+14.9 \pm 3.8$	$+2.42 \pm 1.55$
AWM + GPT-5 Mini	2.3	51.8 ± 15.9	44.9 ± 23.7	-5.6 ± 15.1	-1.36 ± 4.48
OpenCode + DeepSeek V4 Pro	1	51.5 ± 28.9	48.5 ± 34.1	-1.5 ± 2.6	-0.61 ± 1.05
Naive + Opus 5	2.3	50.9 ± 11.4	32.4 ± 12.3	-10.2 ± 24.5	-3.06 ± 8.19
Naive + GPT-5 Mini	2.7	50.3 ± 20.7	32.8 ± 8.2	$+10.6 \pm 10.1$	$+3.61 \pm 3.24$
ReasoningBank + GPT-5 Mini	1.7	46.1 ± 18.6	39.1 ± 12.9	-5.6 ± 7.5	-1.45 ± 2.44
OpenCode + Gemini 3.7 Flash	4	39.1 ± 11.0	19.3 ± 5.8	$+14.4 \pm 3.5$	$+5.15 \pm 0.95$
OpenCode + GPT-5.6-SOL	2	37.3 ± 17.6	25.6 ± 15.7	-3.9 ± 1.5	-0.64 ± 1.43
OpenCode + GPT-5 Mini	4	22.7 ± 19.8	12.7 ± 16.6	$+9.1 \pm 16.4$	$+2.88 \pm 5.31$

Table 17: PrimordialSoup: criteria for assessing rule discovery.

Target	**Criterion**
T1	Both base reactions: $C + E \rightarrow \mathrm{L1a}$ and $D + E \rightarrow \mathrm{L1b}$.
T2	A and F are inert, and one react action fires each feasible base recipe at most once.
T3	React consumes B and sets the catalyst level to $\min(B, 3)$.
T4	Catalyst persists through synthesis but is overwritten by the next react action.
T5	Catalyst level at least one enables $\mathrm{L1a} + \mathrm{L1b} \rightarrow \mathrm{L2b}$.
T6	Free $E + \mathrm{L1a} \rightarrow \mathrm{L2a}$ does not consume catalyst and takes priority over L2b.
T7	Catalyst level at least two enables $\mathrm{L2a} + \mathrm{L2b} \rightarrow \mathrm{L3}$.
T8	Only held molecules score, and the step penalty begins after step 30.

Table 18: PrimordialSoup: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7	T8
backbone comparison								
Claude Opus 5	S	P	S	S	S	P	S	S
Qwen 3.8 Max	S	P	S	S	S	S	S	S
Kimi K3	S	S	S	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S	–	–	P
Gemini 3.7 Flash	S	–	P	–	S	–	–	–
GPT-5.6-SOL	S	–	P	–	S	–	S	–
DeepSeek V4 Pro	S	P	P	P	S	P	P	P
Claude Sonnet 5	S	P	P	S	S	P	S	P
GPT-5 Mini	P	S	P	P	P	P	P	P
agent harness comparison								
Claude Code + Opus 5	S	P	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	P	S	–	S	P	S	S

The main separation is whether a system moves beyond the common L2b plateau to recover the alternative L2a/L3 branch within at least one trial. Kimi K3 recovers every target in at least one trial, and Qwen 3.8 Max and several Claude-based systems nearly do so. Weaker configurations typically retain gaps in T6–T7 or the catalyst-state account.

D.2 GemForge

Learning task. Discover multi-tier recipes and manage limiting ingredients. Fixed game, scored seed 1, first 5 episodes, reference score $C = 791$.

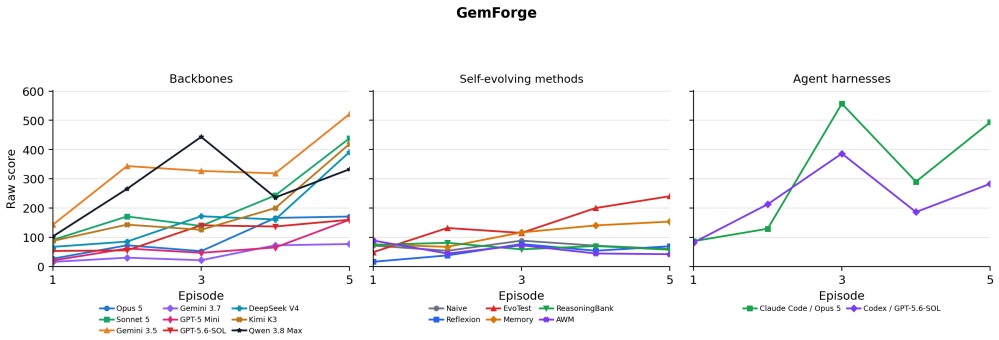


Figure 6: GemForge: raw-score learning curves within the scoring window.

Table 19: GemForge: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	4	99.9	59.4	–	–
Claude Code + Opus 5	3.7	93.9 ± 9.3	39.4 ± 12.7	$+35.9 \pm 28.1$	$+12.32 \pm 8.33$
EvoTest + Kimi K3	3	81.8 ± 30.8	46.9 ± 11.8	$+46.0 \pm 37.0$	$+16.09 \pm 8.99$
OpenCode + Gemini 3.5 Flash	3	72.6 ± 45.4	41.8 ± 32.8	$+22.3 \pm 32.2$	$+9.25 \pm 11.39$
OpenCode + Qwen 3.8 Max	4	63.8 ± 31.7	34.9 ± 18.7	$+12.7 \pm 21.7$	$+5.44 \pm 3.85$
Codex + GPT-5.6-SOL	4.3	61.8 ± 36.4	29.1 ± 18.7	$+11.0 \pm 21.1$	$+4.75 \pm 7.40$
OpenCode + Sonnet 5	3.7	58.4 ± 39.3	27.4 ± 15.6	$+26.5 \pm 32.0$	$+9.70 \pm 11.20$
OpenCode + DeepSeek V4 Pro	4	53.6 ± 43.0	22.2 ± 13.7	$+25.3 \pm 22.7$	$+9.16 \pm 9.31$
OpenCode + Kimi K3	5	53.0 ± 0.7	24.7 ± 1.5	$+24.5 \pm 3.7$	$+9.10 \pm 1.22$
Memory + GPT-5 Mini	2.7	32.1 ± 13.8	25.6 ± 8.0	$+9.8 \pm 9.6$	$+3.12 \pm 3.11$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
OpenCode + Opus 5	4	32.1 ± 21.1	12.4 ± 4.2	$+15.0 \pm 16.6$	$+4.83 \pm 5.93$
Memory + Kimi K3	3.7	28.6 ± 38.3	8.2 ± 10.3	$+14.6 \pm 24.2$	$+5.50 \pm 8.78$
Naive + GPT-5 Mini	3	28.5 ± 9.4	13.2 ± 0.5	$+3.3 \pm 9.5$	$+1.33 \pm 1.79$
OpenCode + GPT-5.6-SOL	4.3	24.7 ± 18.2	13.8 ± 8.8	$+11.9 \pm 12.6$	$+3.73 \pm 4.62$
AWM + GPT-5 Mini	1.3	24.5 ± 18.4	13.3 ± 12.8	-6.7 ± 8.2	-1.93 ± 3.95
Naive + Opus 5	3.7	24.4 ± 24.8	7.6 ± 4.8	-4.2 ± 17.5	-1.91 ± 6.64
OpenCode + GPT-5 Mini	3.3	20.2 ± 22.0	9.0 ± 5.2	$+9.0 \pm 10.8$	$+3.57 \pm 4.26$
Naive + Kimi K3	3	20.1 ± 12.0	5.4 ± 3.8	$+2.0 \pm 4.9$	$+0.34 \pm 1.28$
OpenCode + Gemini 3.7 Flash	3.7	19.3 ± 11.6	5.5 ± 2.5	$+6.5 \pm 6.5$	$+2.08 \pm 1.85$
ReasoningBank + GPT-5 Mini	2.3	18.4 ± 2.1	8.0 ± 3.0	-5.5 ± 7.4	-1.32 ± 3.29
AWM + Kimi K3	3.7	17.8 ± 3.9	6.3 ± 0.6	$+4.5 \pm 3.2$	$+1.03 \pm 0.74$
ReasoningBank + Opus 5	3	17.8 ± 7.8	11.0 ± 4.2	$+5.4 \pm 11.9$	$+1.67 \pm 3.78$
Memory + Opus 5	3.3	17.4 ± 24.9	8.2 ± 11.5	$+4.2 \pm 6.1$	$+0.06 \pm 0.28$
Reflexion + Kimi K3	3.7	15.8 ± 8.9	4.9 ± 2.1	$+1.3 \pm 1.7$	$+0.58 \pm 0.60$
Reflexion + GPT-5 Mini	3.3	15.5 ± 20.7	10.3 ± 14.9	$+8.7 \pm 16.8$	$+2.90 \pm 5.61$
AWM + Opus 5	1	13.3 ± 11.3	2.7 ± 2.3	-6.6 ± 5.7	-2.65 ± 2.27
ReasoningBank + Kimi K3	1.7	11.0 ± 3.2	7.0 ± 3.6	-5.1 ± 1.4	-2.02 ± 0.19
EvoTest + Opus 5	3.7	10.7 ± 0.4	8.8 ± 0.6	$+3.3 \pm 2.6$	$+1.03 \pm 0.58$
Reflexion + Opus 5	3.7	6.5 ± 1.8	4.1 ± 2.0	$+3.1 \pm 1.9$	$+1.15 \pm 0.63$
EvoTest + GPT-5 Mini	1.7	0.9 ± 1.5	0.2 ± 0.3	$+0.0 \pm 0.0$	$+0.00 \pm 0.00$

Table 20: GemForge: criteria for assessing rule discovery.

Target	**Criterion**
T1	Complete five-recipe map from raw materials to tier-1 gems.
T2	Complete four-recipe map from tier-1 to tier-2 gems.
T3	Complete two-recipe map from tier-2 to tier-3 gems.
T4	Every nonrecipe pair consumes its inputs and yields slag. No three-input recipe exists.
T5	Final inventory determines score. Solo firing yields one-point slag. Ash and crystal do not advance to higher tiers.

Table 21: GemForge: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	P	P
Kimi K3	S	S	P	P	P
Gemini 3.5 Flash	S	S	P	S	P
Gemini 3.7 Flash	S	P	P	P	P
GPT-5.6-SOL	P	P	P	P	P
DeepSeek V4 Pro	S	P	P	P	P
Claude Sonnet 5	S	S	P	P	P
GPT-5 Mini	P	P	P	P	P
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	P	P	P	P

High scores can be reached with only a partial recipe graph, so complete-rule recovery separates the systems more sharply than execution alone. Claude Opus 5 and Claude Code + Opus 5 each recover every target in at least one trial, while most other configurations discover useful chains without establishing every tier-2 and tier-3 alternative.

D.3 CATNIP

Learning task. Learn action–target matches and when to exit for the best reward. Fixed game, scored seed 1, first 10 episodes, reference score $C = 320$.

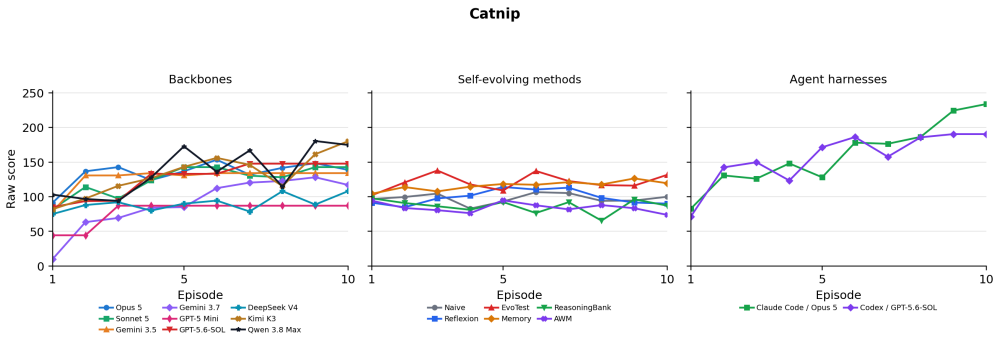


Figure 7: Catnip: raw-score learning curves within the scoring window.

Table 22: Catnip: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Claude Code + Opus 5	9	73.0 ± 1.2	50.4 ± 2.3	$+31.8 \pm 13.2$	$+4.64 \pm 1.41$
Human Top-1	8	70.3	53.5	–	–
EvoTest + Kimi K3	3	59.8 ± 5.5	44.1 ± 11.6	-5.3 ± 5.1	-0.09 ± 0.71
Codex + GPT-5.6-SOL	4.3	59.5 ± 5.9	49.0 ± 6.7	$+21.1 \pm 10.4$	$+3.23 \pm 1.51$
Reflexion + Kimi K3	7.3	56.6 ± 5.4	37.8 ± 1.7	$+10.8 \pm 16.7$	$+1.58 \pm 2.15$
OpenCode + Qwen 3.8 Max	7.3	56.4 ± 10.3	42.7 ± 5.2	$+18.3 \pm 5.7$	$+2.68 \pm 0.78$
OpenCode + Kimi K3	8.7	56.2 ± 13.0	41.4 ± 7.4	$+16.8 \pm 9.0$	$+2.65 \pm 1.43$
EvoTest + Opus 5	8	54.7 ± 10.0	39.3 ± 0.7	$+3.5 \pm 7.2$	$+0.53 \pm 1.18$
OpenCode + Opus 5	6	50.3 ± 3.0	42.0 ± 3.6	$+6.0 \pm 4.3$	$+1.03 \pm 0.74$
AWM + Kimi K3	2.3	47.7 ± 8.5	36.3 ± 2.2	-1.8 ± 3.5	-0.28 ± 0.32
ReasoningBank + Kimi K3	5.7	47.0 ± 5.6	33.7 ± 1.7	-5.7 ± 2.5	-0.38 ± 0.37
Memory + Opus 5	4.7	46.2 ± 5.3	39.4 ± 1.7	$+6.4 \pm 1.7$	$+0.80 \pm 0.30$
OpenCode + GPT-5.6-SOL	5	46.1 ± 3.3	39.4 ± 6.2	$+17.7 \pm 2.7$	$+2.38 \pm 0.66$
Memory + Kimi K3	5.3	46.0 ± 2.8	39.9 ± 3.6	$+4.5 \pm 3.1$	$+0.75 \pm 0.44$
OpenCode + Sonnet 5	3.7	45.2 ± 9.0	38.9 ± 7.2	$+12.5 \pm 6.9$	$+1.74 \pm 0.86$
Naive + GPT-5 Mini	5.3	43.3 ± 6.3	28.5 ± 2.3	-3.4 ± 6.2	$+0.10 \pm 0.55$
OpenCode + Gemini 3.5 Flash	2.3	41.9 ± 5.1	39.8 ± 5.4	$+6.5 \pm 5.7$	$+1.03 \pm 0.90$
Naive + Opus 5	3	40.6 ± 11.3	30.2 ± 1.4	-0.2 ± 4.3	-0.04 ± 0.58
Memory + GPT-5 Mini	3.7	40.6 ± 0.5	29.5 ± 4.1	$+0.8 \pm 11.9$	$+0.11 \pm 1.61$
OpenCode + Gemini 3.7 Flash	5.7	40.6 ± 10.0	28.5 ± 13.8	$+23.4 \pm 4.1$	$+3.44 \pm 0.80$
ReasoningBank + Opus 5	4	39.9 ± 6.1	33.3 ± 4.0	$+2.9 \pm 7.4$	$+0.36 \pm 1.05$
Reflexion + Opus 5	4.7	39.9 ± 3.3	34.2 ± 3.3	$+0.8 \pm 3.6$	$+0.01 \pm 0.34$
Naive + Kimi K3	7.7	39.3 ± 4.4	32.8 ± 2.6	-0.2 ± 2.9	$+0.07 \pm 0.50$
EvoTest + GPT-5 Mini	8.3	35.8 ± 1.4	30.1 ± 1.7	$+3.0 \pm 2.2$	$+0.53 \pm 0.30$
OpenCode + DeepSeek V4 Pro	3.7	33.8 ± 6.0	28.2 ± 10.1	$+5.2 \pm 2.6$	$+0.72 \pm 0.59$
AWM + Opus 5	6	32.7 ± 1.6	28.2 ± 0.9	$+0.4 \pm 4.2$	-0.00 ± 0.53
Reflexion + GPT-5 Mini	3.7	32.6 ± 17.8	21.0 ± 11.5	-9.7 ± 15.5	-1.20 ± 1.71
ReasoningBank + GPT-5 Mini	2.7	28.2 ± 3.6	14.2 ± 6.5	-5.3 ± 8.1	-0.80 ± 1.30
OpenCode + GPT-5 Mini	1.7	27.2 ± 11.7	24.5 ± 7.4	$+8.9 \pm 15.4$	$+1.29 \pm 2.24$
AWM + GPT-5 Mini	1.7	24.1 ± 12.9	14.6 ± 12.9	-2.9 ± 4.9	-0.52 ± 0.53

Table 23: Catnip: criteria for assessing rule discovery.

Target	**Criterion**
T1	The six targets occur in a fixed order.
T2	Complete mapping from all six targets to their sensory preferences.
T3	Complete mapping of soothing and chaotic style preferences.
T4	The five influence actions have distinct modalities and styles, and matching them changes influence gain.

Target	Criterion (continued)
T5	Conversion uses target-specific LOW, MED, and HIGH reward thresholds.
T6	Complete breed routing: Ragdoll to cuteness, Bobcat to menace, Black cat to stealth, and Orange to energy.
T7	Rewards trade immediate progress against attributes and items that affect later conversions.
T8	High stress or suspicion makes late mistakes costly.
T9	Items strengthen particular actions, control risk, or restore energy.
T10	Each unused turn adds 0.1 to the final multiplier, creating an early-exit tradeoff.

Table 24: Catnip: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
backbone comparison										
Claude Opus 5	S	P	P	P	S	P	S	P	P	–
Qwen 3.8 Max	S	P	P	P	P	P	S	P	P	S
Kimi K3	S	P	P	P	S	P	S	S	S	S
Gemini 3.5 Flash	S	P	P	P	S	S	S	P	S	P
Gemini 3.7 Flash	P	P	–	P	–	P	P	P	P	P
GPT-5.6-SOL	P	P	–	P	P	P	P	–	P	P
DeepSeek V4 Pro	S	–	P	P	P	P	S	–	P	–
Claude Sonnet 5	S	P	–	P	S	P	S	S	S	P
GPT-5 Mini	P	P	P	P	P	P	P	P	P	P
agent harness comparison										
Claude Code + Opus 5	S	P	P	P	S	P	S	S	S	S
Codex + GPT-5.6-SOL	P	P	P	P	P	P	P	P	P	P

The strongest runs combine local target preferences with the long-horizon reward and early-exit tradeoff. Claude Code + Opus 5 and Kimi K3 show the broadest complete-target coverage across their three opportunities, whereas other configurations often learn safe actions or reward thresholds without ever completing the preference, breed, item, or multiplier model.

D.4 DUNGEON

Learning task. Remember rooms, keys, hazards, and treasures to refine an escape route. Fixed game, scored seed 2, first 5 episodes, reference score $C = 54$.

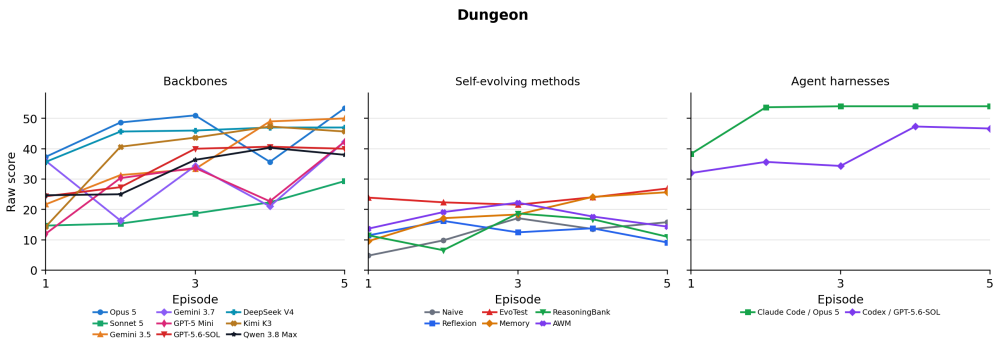


Figure 8: Dungeon: raw-score learning curves within the scoring window.

Table 25: Dungeon: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Claude Code + Opus 5	2.3	100.0 ± 0.0	94.1 ± 6.9	$+14.8 \pm 17.3$	$+5.86 \pm 6.84$
Human Top-1	5	100.0	67.8	–	–
OpenCode + Opus 5	4.7	98.8 ± 2.1	83.7 ± 13.7	$+2.8 \pm 27.5$	$+3.52 \pm 5.56$
OpenCode + Gemini 3.5 Flash	4.7	92.6 ± 9.6	68.6 ± 4.1	$+42.6 \pm 19.4$	$+13.76 \pm 5.40$
EvoTest + Kimi K3	3	92.0 ± 12.3	56.4 ± 2.8	-2.5 ± 19.0	$+0.19 \pm 5.47$
OpenCode + Qwen 3.8 Max	3.3	92.0 ± 6.5	60.9 ± 18.2	$+26.5 \pm 63.3$	$+7.78 \pm 20.37$
Codex + GPT-5.6-SOL	4	92.0 ± 4.3	72.6 ± 11.9	$+24.4 \pm 31.5$	$+7.59 \pm 10.08$
OpenCode + Gemini 3.7 Flash	3.7	89.5 ± 5.3	55.6 ± 15.9	$+10.2 \pm 22.7$	$+3.21 \pm 3.53$
OpenCode + Kimi K3	3.3	87.7 ± 10.7	71.0 ± 7.4	$+35.2 \pm 10.4$	$+12.84 \pm 3.43$
OpenCode + DeepSeek V4 Pro	2	87.0 ± 8.5	82.0 ± 16.2	$+11.7 \pm 20.3$	$+4.44 \pm 7.70$
OpenCode + GPT-5 Mini	3.7	78.4 ± 2.1	52.2 ± 8.7	$+21.0 \pm 24.6$	$+9.81 \pm 5.85$
OpenCode + GPT-5.6-SOL	3.7	77.2 ± 11.3	63.8 ± 6.3	$+26.9 \pm 29.1$	$+8.27 \pm 8.52$
AWM + Kimi K3	2.3	75.9 ± 14.8	48.0 ± 10.6	$+9.6 \pm 36.9$	$+0.80 \pm 14.60$
Naive + GPT-5 Mini	4.3	75.9 ± 5.6	31.0 ± 9.5	$+18.8 \pm 14.7$	$+5.56 \pm 5.48$
Memory + Opus 5	4.3	71.0 ± 14.0	36.4 ± 5.0	$+28.1 \pm 35.1$	$+9.63 \pm 12.84$
Naive + Kimi K3	3	70.4 ± 28.7	26.7 ± 9.3	$+11.7 \pm 18.3$	$+4.57 \pm 6.45$
OpenCode + Sonnet 5	4.7	70.4 ± 13.0	37.2 ± 20.7	$+20.1 \pm 23.2$	$+6.73 \pm 7.71$
Reflexion + Opus 5	4	66.7 ± 9.6	30.2 ± 13.9	$+20.4 \pm 13.4$	$+6.67 \pm 5.34$
Memory + Kimi K3	3.3	62.3 ± 46.6	35.4 ± 26.2	$+36.4 \pm 39.8$	$+12.22 \pm 12.77$
EvoTest + GPT-5 Mini	2.7	58.6 ± 39.6	35.9 ± 23.9	$+29.0 \pm 26.5$	$+9.44 \pm 7.22$
EvoTest + Opus 5	1	57.4 ± 35.3	39.5 ± 21.9	-13.6 ± 37.1	-5.37 ± 10.29
Reflexion + GPT-5 Mini	1.3	57.4 ± 42.7	25.2 ± 16.5	-17.6 ± 24.1	-7.04 ± 9.63
ReasoningBank + Opus 5	3.7	51.9 ± 13.0	23.0 ± 6.6	$+15.7 \pm 5.8$	$+4.14 \pm 1.98$
ReasoningBank + Kimi K3	1.7	50.0 ± 42.1	23.8 ± 19.5	-2.8 ± 21.0	-3.52 ± 7.25
AWM + Opus 5	2.3	49.4 ± 28.7	28.9 ± 19.8	$+2.5 \pm 23.2$	$+2.90 \pm 6.58$
Reflexion + Kimi K3	1.3	36.4 ± 37.5	14.6 ± 7.8	-16.0 ± 20.1	-3.58 ± 3.74
Memory + GPT-5 Mini	1.3	34.0 ± 24.2	33.5 ± 24.8	-0.3 ± 0.5	-0.06 ± 0.11
ReasoningBank + GPT-5 Mini	1.7	31.5 ± 25.7	24.8 ± 14.8	$+14.2 \pm 26.2$	$+4.57 \pm 8.23$
AWM + GPT-5 Mini	2.3	31.5 ± 25.7	19.8 ± 14.2	-14.2 ± 20.4	-3.76 ± 5.76
Naive + Opus 5	3.3	29.0 ± 27.8	10.1 ± 6.3	$+10.5 \pm 13.4$	$+4.20 \pm 5.36$

Table 26: Dungeon: criteria for assessing rule discovery.

Target	**Criterion**
T1	Complete 20-room graph, including every visible edge and every directed hidden passage.
T2	Room 14 is the exit and grants a ten-point escape bonus.
T3	Room 16 contains the 25-gold treasure.
T4	The key is in room 17 and is consumed when room 16 is opened.
T5	Complete location map for the charm, oil, boots, and potion.
T6	Complete effect map for those four items.
T7	Complete location and damage map for every trap and monster.
T8	Movement consumes torch while search does not. Death or torch exhaustion forfeits the escape bonus.
T9	Final score is held gold plus the escape bonus.

Table 27: Dungeon: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7	T8	T9
backbone comparison									
Claude Opus 5	P	S	S	S	S	S	S	S	S
Qwen 3.8 Max	P	S	S	S	P	S	P	S	S
Kimi K3	P	S	S	S	S	S	S	S	S
Gemini 3.5 Flash	P	S	S	S	S	S	P	S	S
Gemini 3.7 Flash	P	S	S	S	S	S	P	P	S
GPT-5.6-SOL	P	S	S	P	P	S	P	P	S
DeepSeek V4 Pro	P	S	S	S	S	S	P	P	S
Claude Sonnet 5	P	S	S	S	S	S	P	P	S
GPT-5 Mini	P	S	S	P	P	P	P	P	S
agent harness comparison									
Claude Code + Opus 5	P	S	S	S	S	S	S	S	S
Codex + GPT-5.6-SOL	P	S	S	S	P	S	P	S	S

A successful treasure-and-exit route does not imply discovery of the complete dungeon. Most systems recover the key landmarks and scoring rule in at least one trial, but T1 remains partial for every configuration because no trace enumerates the full room graph. Complete hazard coverage in T7 is also less common than route execution.

D.5 SynergyCorp

Learning task. Learn which character traits predict preferred, weak, and taboo actions. Re-shuffled game, scored seed 2, first 5 episodes, reference score $C = 137$.

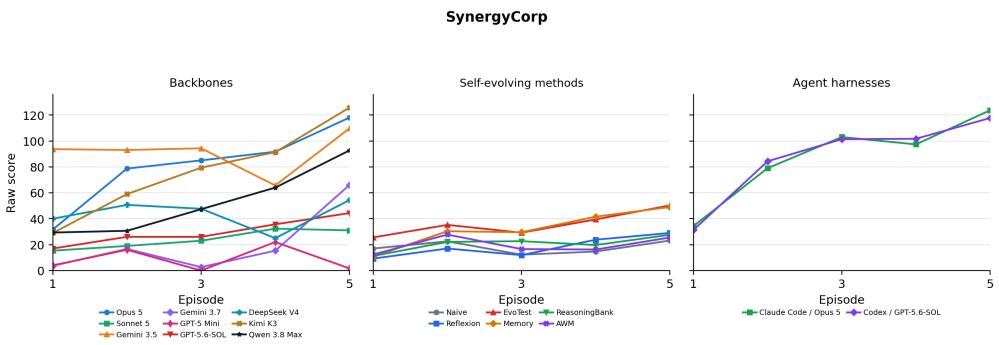


Figure 9: SynergyCorp: raw-score learning curves within the scoring window.

Table 28: SynergyCorp: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
OpenCode + Kimi K3	5	91.7 ± 0.4	56.1 ± 8.4	$+47.1 \pm 7.1$	$+16.47 \pm 1.46$
Human Top-1	5	91.2	44.5	–	–
Claude Code + Opus 5	5	90.3 ± 2.9	63.8 ± 4.8	$+39.3 \pm 17.9$	$+14.38 \pm 4.82$
OpenCode + Opus 5	5	86.1 ± 7.6	59.1 ± 1.0	$+36.3 \pm 9.4$	$+13.55 \pm 2.96$
Codex + GPT-5.6-SOL	5	85.9 ± 6.6	63.7 ± 4.2	$+37.8 \pm 5.9$	$+13.87 \pm 0.91$
OpenCode + Gemini 3.5 Flash	2.3	81.8 ± 17.7	66.6 ± 18.1	-4.1 ± 23.2	$+0.34 \pm 7.00$
Memory + Kimi K3	4	70.3 ± 5.9	43.1 ± 6.5	$+46.2 \pm 3.1$	$+14.50 \pm 1.32$
OpenCode + Qwen 3.8 Max	5	67.6 ± 26.6	38.5 ± 16.1	$+35.3 \pm 24.0$	$+11.68 \pm 7.35$
EvoTest + Kimi K3	4	61.6 ± 10.0	40.9 ± 9.1	$+22.7 \pm 6.9$	$+8.10 \pm 1.84$
OpenCode + DeepSeek V4 Pro	3.3	50.4 ± 20.3	31.8 ± 21.3	-4.1 ± 24.3	$+0.22 \pm 6.84$
OpenCode + Gemini 3.7 Flash	5	48.2 ± 33.6	15.2 ± 4.9	$+22.3 \pm 11.6$	$+9.00 \pm 5.93$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
EvoTest + Opus 5	5	42.6 ± 1.7	28.9 ± 3.3	$+6.8 \pm 1.4$	$+3.14 \pm 1.42$
ReasoningBank + Kimi K3	3	35.5 ± 12.4	23.4 ± 7.0	$+7.7 \pm 7.4$	$+2.97 \pm 0.91$
AWM + Opus 5	5	34.5 ± 2.8	19.6 ± 2.5	$+9.5 \pm 4.3$	$+4.33 \pm 0.75$
ReasoningBank + Opus 5	4	33.6 ± 3.2	13.9 ± 2.0	$+11.7 \pm 10.7$	$+5.04 \pm 1.80$
Reflexion + Kimi K3	4.7	33.1 ± 4.6	14.6 ± 3.3	$+7.2 \pm 9.1$	$+2.26 \pm 2.89$
OpenCode + GPT-5.6-SOL	4.3	32.8 ± 10.9	21.8 ± 4.2	$+13.5 \pm 9.5$	$+4.70 \pm 2.95$
AWM + Kimi K3	3.7	29.4 ± 8.2	16.0 ± 4.3	-0.6 ± 13.4	$+0.73 \pm 4.97$
Naive + Kimi K3	3.7	28.5 ± 3.2	19.4 ± 3.1	-0.2 ± 7.1	$+0.78 \pm 2.17$
Reflexion + Opus 5	4	28.5 ± 1.9	14.8 ± 1.9	$+14.1 \pm 2.8$	$+4.62 \pm 1.43$
Memory + Opus 5	5	27.7 ± 7.0	13.9 ± 5.0	$+1.2 \pm 7.5$	$+2.21 \pm 1.43$
OpenCode + Sonnet 5	4	27.3 ± 4.4	17.6 ± 3.3	$+10.6 \pm 2.4$	$+3.26 \pm 0.92$
Memory + GPT-5 Mini	3.7	26.5 ± 9.3	13.6 ± 5.2	$+6.7 \pm 8.6$	$+2.48 \pm 2.41$
Naive + Opus 5	2.7	23.6 ± 5.7	15.2 ± 3.0	-1.3 ± 0.9	$+0.17 \pm 1.72$
Reflexion + GPT-5 Mini	3.7	23.6 ± 3.4	10.4 ± 2.6	$+7.8 \pm 10.2$	$+3.26 \pm 3.03$
AWM + GPT-5 Mini	2.3	20.4 ± 7.4	7.4 ± 2.6	-7.2 ± 4.6	-1.85 ± 1.29
ReasoningBank + GPT-5 Mini	2.7	20.0 ± 9.0	7.9 ± 3.6	-4.1 ± 4.1	-1.39 ± 1.74
EvoTest + GPT-5 Mini	4.3	16.3 ± 4.0	8.9 ± 0.3	$+2.1 \pm 1.6$	$+0.49 \pm 0.84$
OpenCode + GPT-5 Mini	4	16.1 ± 2.5	6.4 ± 0.4	$+1.3 \pm 3.8$	$+0.10 \pm 1.33$
Naive + GPT-5 Mini	2.7	12.4 ± 0.0	4.6 ± 1.1	-0.4 ± 4.6	-0.00 ± 0.96

Table 29: SynergyCorp: criteria for assessing rule discovery.

Target	**Criterion**
T1	Complete partition of the eight actions into four behavior pairs. Designer-assigned modality names are not required.
T2	Complete mapping of eight informative desk traits and identification of four decoys.
T3	Complete preference ordering over the four behavior groups for each boss type.
T4	The overtime-selfie and buzzwords taboo clears approval and adds eight demerits.
T5	Repeating one action produces two-stage decay, motivating alternation within a pair.
T6	Steal-lunch followed by buzzwords applies the ordered combo bonus.
T7	A drifter changes preference at half its promotion threshold, with the second signal identifying the new preference.
T8	Promotion thresholds, escalating rewards, and the 30-step budget jointly determine total score.

Table 30: SynergyCorp: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7	T8
backbone comparison								
Claude Opus 5	S	P	S	S	S	–	P	P
Qwen 3.8 Max	P	P	P	P	P	–	S	P
Kimi K3	S	P	P	P	S	–	P	S
Gemini 3.5 Flash	S	P	P	–	P	–	P	–
Gemini 3.7 Flash	S	P	–	P	P	–	–	P
GPT-5.6-SOL	P	P	P	P	P	P	P	P
DeepSeek V4 Pro	P	P	P	–	–	–	P	–
Claude Sonnet 5	P	P	P	P	P	P	P	P
GPT-5 Mini	P	P	P	P	P	P	P	P
agent harness comparison								
Claude Code + Opus 5	S	S	P	P	S	P	P	P
Codex + GPT-5.6-SOL	P	P	P	–	P	P	P	–

Systems learn the action groups more readily than the temporal rules needed for a complete promotion policy. Several configurations fully recover the action partition in at least one trial, but exact taboo, combo, repetition-decay, and drifter-switch conditions rarely become complete within any single trial.

D.6 DREAMGARDEN

Learning task. Infer delayed ecological effects and plan actions across turns. Fixed game, scored seed 1, first 10 episodes, reference score $C = 6300$.

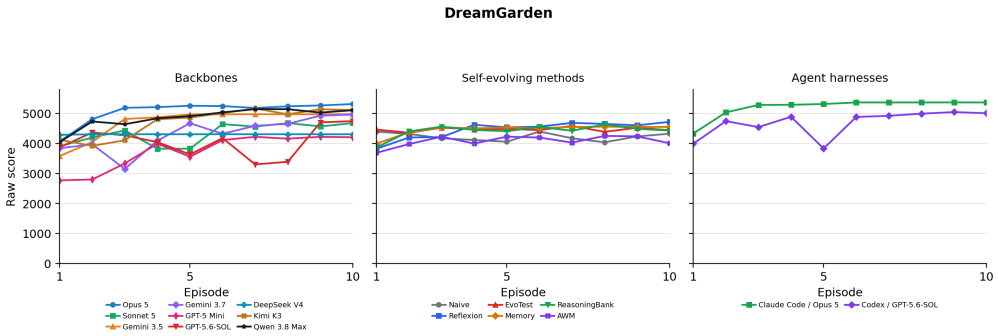


Figure 10: DreamGarden: raw-score learning curves within the scoring window.

Table 31: DreamGarden: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Claude Code + Opus 5	4.7	85.2 ± 0.0	82.7 ± 1.5	$+7.7 \pm 4.0$	$+1.19 \pm 0.55$
OpenCode + Opus 5	7.7	84.3 ± 0.8	80.6 ± 2.4	$+9.3 \pm 2.7$	$+1.40 \pm 0.35$
Human Top-1	8	84.2	78.7	–	–
Reflexion + Opus 5	7.3	82.8 ± 2.5	74.4 ± 1.0	$+6.4 \pm 1.7$	$+0.97 \pm 0.31$
OpenCode + Kimi K3	8.7	82.6 ± 1.3	75.0 ± 3.6	$+16.3 \pm 6.8$	$+2.22 \pm 0.92$
OpenCode + Qwen 3.8 Max	5.7	81.8 ± 2.5	77.2 ± 1.4	$+9.8 \pm 4.6$	$+1.46 \pm 0.51$
ReasoningBank + GPT-5 Mini	5.7	81.4 ± 0.8	72.6 ± 2.4	$+1.9 \pm 7.2$	$+0.32 \pm 0.98$
Codex + GPT-5.6-SOL	7	80.2 ± 1.9	74.4 ± 4.4	$+9.3 \pm 3.5$	$+1.41 \pm 0.52$
Reflexion + Kimi K3	7.3	79.1 ± 4.9	70.5 ± 4.3	$+10.3 \pm 10.6$	$+1.31 \pm 1.12$
OpenCode + Gemini 3.5 Flash	4.3	78.9 ± 5.8	74.8 ± 2.8	$+13.1 \pm 10.2$	$+1.93 \pm 1.47$
ReasoningBank + Opus 5	7.3	78.9 ± 2.2	71.5 ± 2.2	$+6.3 \pm 2.0$	$+1.02 \pm 0.30$
OpenCode + Gemini 3.7 Flash	7	78.8 ± 0.2	68.5 ± 8.3	$+19.0 \pm 9.3$	$+2.45 \pm 1.06$
EvoTest + Opus 5	3	76.9 ± 6.1	72.6 ± 4.0	-0.5 ± 0.7	$+0.03 \pm 0.22$
ReasoningBank + Kimi K3	7	76.4 ± 5.2	66.4 ± 1.7	$+3.2 \pm 3.3$	$+0.43 \pm 0.60$
Naive + GPT-5 Mini	7.3	75.9 ± 7.0	68.4 ± 1.7	$+1.4 \pm 3.1$	$+0.34 \pm 0.38$
Naive + Opus 5	4.3	75.6 ± 0.4	70.4 ± 0.7	-0.3 ± 3.4	$+0.00 \pm 0.40$
OpenCode + GPT-5.6-SOL	6	75.3 ± 7.4	64.3 ± 14.4	$+1.7 \pm 9.5$	$+0.39 \pm 1.49$
Memory + Kimi K3	4.7	75.2 ± 2.0	69.3 ± 1.0	$+7.2 \pm 9.3$	$+1.19 \pm 1.30$
EvoTest + GPT-5 Mini	1.7	74.9 ± 7.2	71.1 ± 7.1	$+0.8 \pm 5.7$	$+0.15 \pm 0.83$
Memory + Opus 5	4	74.8 ± 6.8	72.4 ± 5.1	$+3.3 \pm 1.6$	$+0.44 \pm 0.36$
OpenCode + Sonnet 5	5.7	74.3 ± 4.7	68.8 ± 0.7	$+7.3 \pm 4.9$	$+1.33 \pm 0.95$
Naive + Kimi K3	3.7	73.4 ± 0.8	62.3 ± 1.9	-5.6 ± 3.2	-0.73 ± 0.94
AWM + Opus 5	3.7	73.0 ± 3.6	67.2 ± 4.4	$+1.9 \pm 5.3$	$+0.39 \pm 0.86$
AWM + GPT-5 Mini	5.3	72.8 ± 0.8	67.5 ± 1.5	-0.7 ± 3.1	-0.12 ± 0.46
EvoTest + Kimi K3	4	72.4 ± 0.6	69.1 ± 0.2	-0.2 ± 1.4	-0.01 ± 0.17
Reflexion + GPT-5 Mini	6.3	72.1 ± 8.0	67.6 ± 7.2	$+10.9 \pm 11.4$	$+1.57 \pm 1.63$
Memory + GPT-5 Mini	2	71.7 ± 4.2	71.1 ± 4.5	$+1.5 \pm 1.3$	$+0.25 \pm 0.22$
OpenCode + DeepSeek V4 Pro	1.7	68.4 ± 0.5	68.3 ± 0.6	$+0.1 \pm 0.1$	$+0.02 \pm 0.01$
OpenCode + GPT-5 Mini	5.7	68.0 ± 10.3	59.3 ± 13.8	$+19.5 \pm 10.5$	$+2.72 \pm 1.28$
AWM + Kimi K3	6.7	66.8 ± 2.8	59.7 ± 2.0	$+8.3 \pm 4.6$	$+1.13 \pm 1.04$

Table 32: DreamGarden: criteria for assessing rule discovery.

Target	Criterion
T1	Resonance per turn is $\sum v_j \min(pop_j, 35)$. Population is capped at 100 and decays by 20%. For seed 1, Spores and Moths are worth $+5$, Nightmares -3, and Jelly and Weavers 0. Nightmares suppress the Weaver chain.
T2	Eclipse increases Moths but also introduces Nightmares. Throb increases Weavers, Whisper suppresses Nightmares, and Tide feeds only Nightmares.
T3	Throb starts the chain Weavers→Jelly→Spores. Each ecological link takes one turn, so the final benefit is delayed.
T4	Over 18 turns, combine direct Moth growth, delayed Spore growth, and Nightmare removal. Throb has little immediate value but a large payoff two turns later.
T5	The archived 5,367-point sequence starts with Throb, then interleaves Eclipse and Whisper to keep Moths and Spores near saturation and Nightmares near zero. It never uses Tide.

Table 33: DreamGarden: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S
Kimi K3	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S
Gemini 3.7 Flash	S	S	S	S	–
GPT-5.6-SOL	S	–	–	S	–
DeepSeek V4 Pro	S	S	S	S	–
Claude Sonnet 5	S	S	S	S	–
GPT-5 Mini	S	S	–	S	–
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	–

D.7 ECOSPHERE

Learning task. Infer food-web, competition, and waste dynamics to plan stocking. Fixed game, scored seed 1, first 5 episodes, reference score $C = 339$.

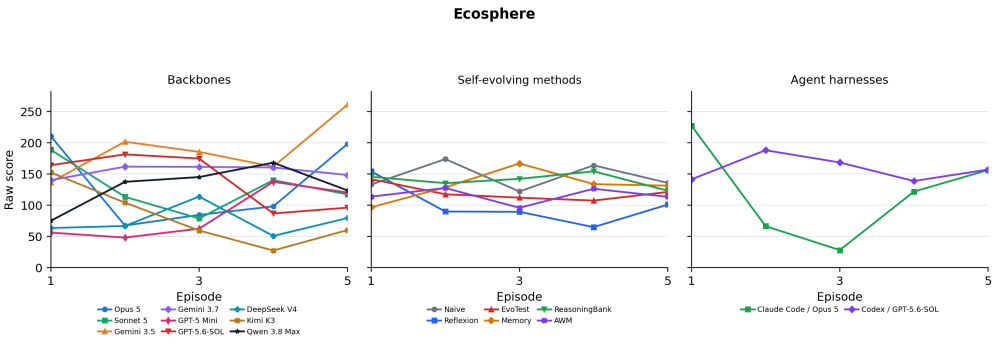


Figure 11: Ecosphere: raw-score learning curves within the scoring window.

Table 34: Ecosphere: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	5	85.0	59.7	–	–
Claude Code + Opus 5	2.3	80.1 ± 19.6	35.4 ± 18.2	-2.3 ± 23.0	-2.56 ± 9.55
OpenCode + Gemini 3.5 Flash	4	77.1 ± 15.3	55.8 ± 12.2	$+12.4 \pm 21.4$	$+6.16 \pm 4.96$
Reflexion + Opus 5	4	76.3 ± 21.7	46.5 ± 13.2	-10.0 ± 15.6	-3.97 ± 7.40
Naive + Opus 5	3	75.1 ± 6.6	56.7 ± 8.1	-4.2 ± 21.8	-2.35 ± 8.88
Memory + Kimi K3	3.3	73.0 ± 23.7	31.4 ± 16.0	$+11.5 \pm 22.4$	$+5.01 \pm 5.27$
Naive + Kimi K3	3.3	69.4 ± 10.9	42.2 ± 8.8	$+1.3 \pm 30.8$	$+2.16 \pm 6.68$
OpenCode + Opus 5	3	68.5 ± 16.1	38.8 ± 5.4	$+2.7 \pm 17.7$	$+0.18 \pm 5.61$
OpenCode + Gemini 3.7 Flash	2.3	66.7 ± 6.2	45.5 ± 13.3	$+1.0 \pm 15.7$	$+0.45 \pm 3.91$
AWM + Kimi K3	2.7	65.5 ± 16.6	27.7 ± 12.2	-18.8 ± 54.2	-5.75 ± 18.52
EvoTest + Opus 5	1.7	64.1 ± 19.8	50.2 ± 19.5	-7.0 ± 7.6	-2.59 ± 2.30
AWM + Opus 5	4	63.8 ± 12.0	48.2 ± 5.5	$+24.6 \pm 6.8$	$+8.53 \pm 1.76$
Memory + Opus 5	1.3	63.3 ± 7.6	59.7 ± 8.2	$+4.7 \pm 2.8$	$+1.07 \pm 0.38$
OpenCode + Sonnet 5	2	59.0 ± 5.4	37.6 ± 8.9	-6.6 ± 14.0	-3.42 ± 2.74
Codex + GPT-5.6-SOL	3	57.8 ± 4.1	46.8 ± 6.3	-5.0 ± 10.3	-0.53 ± 3.76
ReasoningBank + Kimi K3	1.3	55.7 ± 6.0	51.3 ± 12.2	-7.3 ± 12.7	-2.90 ± 5.02
OpenCode + GPT-5.6-SOL	1.7	53.5 ± 15.8	41.5 ± 11.5	-24.0 ± 17.3	-6.80 ± 5.46
OpenCode + Qwen 3.8 Max	3.7	52.8 ± 8.3	38.3 ± 1.0	$+11.7 \pm 4.5$	$+3.78 \pm 0.92$
OpenCode + GPT-5 Mini	2.7	46.1 ± 19.1	25.0 ± 12.6	$+22.7 \pm 23.1$	$+6.43 \pm 6.64$
OpenCode + Kimi K3	1.3	45.3 ± 14.2	23.8 ± 10.6	-24.9 ± 25.7	-7.71 ± 6.14
Reflexion + Kimi K3	2.3	44.8 ± 8.7	12.8 ± 4.8	-15.5 ± 28.3	-5.27 ± 10.49
Reflexion + GPT-5 Mini	1.3	43.6 ± 7.1	28.9 ± 6.1	-9.0 ± 9.9	-2.37 ± 2.56
ReasoningBank + GPT-5 Mini	2	43.3 ± 16.8	31.4 ± 8.8	$+4.0 \pm 10.9$	-0.10 ± 2.10
ReasoningBank + Opus 5	2	41.9 ± 14.4	41.2 ± 14.1	$+1.8 \pm 1.3$	$+0.73 \pm 0.53$
AWM + GPT-5 Mini	1	36.6 ± 7.4	26.2 ± 6.3	-6.5 ± 2.6	-2.94 ± 0.62
EvoTest + Kimi K3	1	36.3 ± 12.6	36.3 ± 12.6	$+0.0 \pm 0.0$	$+0.00 \pm 0.00$
Naive + GPT-5 Mini	2.7	35.9 ± 7.0	30.0 ± 2.9	-0.3 ± 2.5	-0.23 ± 1.16
OpenCode + DeepSeek V4 Pro	3.3	34.5 ± 27.3	22.0 ± 18.6	$+0.0 \pm 19.2$	$+0.47 \pm 5.68$
EvoTest + GPT-5 Mini	1.7	25.8 ± 4.1	19.6 ± 2.7	-6.6 ± 8.8	-1.95 ± 2.10
Memory + GPT-5 Mini	2	25.7 ± 12.3	25.1 ± 11.6	$+1.4 \pm 1.9$	$+0.50 \pm 0.67$

Table 35: Ecosphere: criteria for assessing rule discovery.

Target	**Criterion**
T1	Spend a one-time budget of 150 on stocking, using at most 40 setup actions. Sealing the habitat starts a deterministic 30-day simulation.
T2	Moss, algae, and microbes support the ecosystem but score zero. Herbivores score 4 per animal, subject to saturation. Minnows and newts score 20.
T3	Seed-1 producer layer: moss $24\times$ and algae $18\times$ biomass per unit, each capped at 10 stocked units.
T4	Seed-1 food web and competition: snail and shrimp eat moss, daphnia eats algae, rank order gives shrimp priority over snail, and overgrazed producers collapse.
T5	Predator and keystone dynamics: minnow hunts shrimp only, newt hunts shrimp/snail, prey below refuge 5 cannot be caught, and a single newt culls shrimp enough for snails to survive.
T6	Living animals and corpses create waste that produces toxic gas. Enough decomposers must be stocked to clear it.
T7	Unique 339 stocking vector: moss 10, algae 9, snail 2, shrimp 2, daphnia 2, newt 1, microbe 8, no minnow. This ends at snail 12, shrimp 18, daphnia 27, newt 7 with no extinction event.

Table 36: Ecosphere: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7
backbone comparison							
Claude Opus 5	S	S	S	S	S	S	–
Qwen 3.8 Max	S	S	S	S	S	S	–
Kimi K3	S	S	S	S	S	S	–
Gemini 3.5 Flash	S	S	S	S	S	S	S
Gemini 3.7 Flash	S	S	S	S	S	S	–
GPT-5.6-SOL	–	S	S	S	S	S	–
DeepSeek V4 Pro	–	S	S	S	S	S	–
Claude Sonnet 5	S	S	S	S	S	S	–
GPT-5 Mini	S	S	S	S	S	S	–
agent harness comparison							
Claude Code + Opus 5	S	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S	–

D.8 GRAPEVINE

Learning task. Infer a cascade network and choose effective seed combinations. Fixed game, scored seed 1, first 5 episodes, reference score $C = 16$.

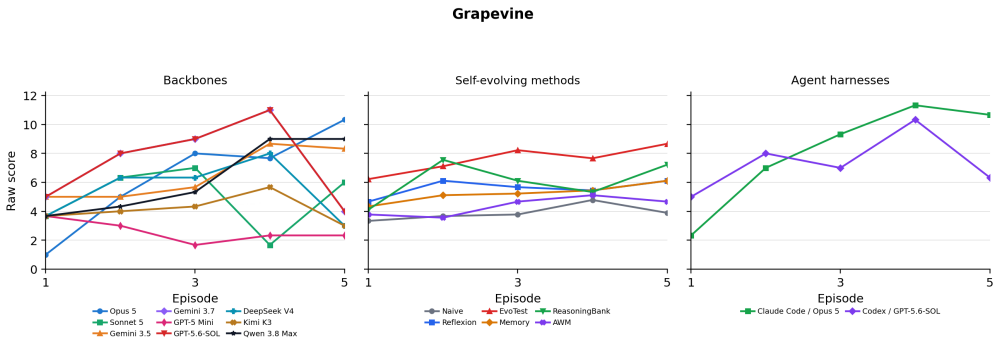


Figure 12: Grapevine: raw-score learning curves within the scoring window.

Table 37: Grapevine: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	5	93.8	52.5	–	–
Claude Code + Opus 5	4.3	75.0 ± 10.8	50.8 ± 8.8	$+39.6 \pm 22.2$	$+13.13 \pm 7.98$
Memory + Opus 5	3.3	70.8 ± 14.4	53.8 ± 1.2	$+17.7 \pm 3.6$	$+7.08 \pm 1.44$
OpenCode + Gemini 3.7 Flash	4	68.8 ± 0.0	46.2 ± 0.0	$+6.2 \pm 0.0$	$+0.62 \pm 0.00$
OpenCode + GPT-5.6-SOL	4	68.8 ± 0.0	46.2 ± 0.0	$+6.2 \pm 0.0$	$+0.62 \pm 0.00$
Codex + GPT-5.6-SOL	4.3	68.8 ± 0.0	45.8 ± 0.7	$+11.4 \pm 9.0$	$+3.12 \pm 4.33$
AWM + GPT-5 Mini	4.3	66.7 ± 7.2	45.4 ± 5.6	$+19.8 \pm 19.1$	$+5.00 \pm 5.96$
EvoTest + Opus 5	3	64.6 ± 15.7	53.3 ± 13.5	$+25.0 \pm 14.3$	$+8.32 \pm 3.97$
Reflexion + Opus 5	3	64.6 ± 3.6	51.7 ± 3.1	$+18.7 \pm 8.3$	$+6.66 \pm 2.53$
Reflexion + Kimi K3	3	64.6 ± 13.0	31.7 ± 12.5	-7.3 ± 17.2	$+0.21 \pm 6.32$
OpenCode + Opus 5	5	64.6 ± 7.2	40.0 ± 4.3	$+37.5 \pm 10.8$	$+13.33 \pm 2.89$
OpenCode + Qwen 3.8 Max	3.3	62.5 ± 28.6	39.2 ± 17.5	$+31.2 \pm 16.5$	$+9.58 \pm 6.41$
OpenCode + Gemini 3.5 Flash	4.7	60.4 ± 7.2	40.8 ± 6.3	$+21.9 \pm 9.4$	$+6.46 \pm 3.77$
ReasoningBank + GPT-5 Mini	1.7	58.3 ± 19.1	47.1 ± 10.8	-3.1 ± 14.3	$+0.00 \pm 3.75$
Naive + GPT-5 Mini	3.7	58.3 ± 13.0	43.8 ± 8.8	$+3.2 \pm 11.3$	$+1.05 \pm 2.01$
ReasoningBank + Opus 5	3.7	58.3 ± 9.5	42.5 ± 8.8	$+17.7 \pm 7.9$	$+7.29 \pm 2.19$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Naive + Kimi K3	4	54.2 ± 26.0	16.7 ± 3.8	$+12.5 \pm 19.0$	$+3.12 \pm 3.48$
EvoTest + Kimi K3	2	52.1 ± 29.5	45.0 ± 33.6	-7.3 ± 18.3	-1.46 ± 3.66
OpenCode + DeepSeek V4 Pro	3.3	52.1 ± 28.9	34.2 ± 20.9	$+3.1 \pm 5.4$	$+0.21 \pm 0.72$
EvoTest + GPT-5 Mini	3.7	52.1 ± 7.2	43.8 ± 6.6	$+10.4 \pm 7.2$	$+3.33 \pm 2.60$
OpenCode + Sonnet 5	3.3	47.9 ± 26.0	30.8 ± 16.6	-7.3 ± 13.0	$+0.00 \pm 4.38$
OpenCode + Kimi K3	3	43.8 ± 21.7	25.8 ± 16.8	$+3.1 \pm 3.1$	$+0.21 \pm 0.95$
Memory + GPT-5 Mini	1	39.6 ± 3.6	30.8 ± 7.3	-4.2 ± 6.5	-1.04 ± 1.44
ReasoningBank + Kimi K3	2.3	39.6 ± 29.5	24.2 ± 22.9	-6.3 ± 22.5	$+0.20 \pm 6.56$
Reflexion + GPT-5 Mini	1	37.5 ± 18.8	21.7 ± 5.2	-4.2 ± 14.1	-2.71 ± 5.32
OpenCode + GPT-5 Mini	2	31.2 ± 0.0	16.2 ± 3.3	-6.3 ± 11.3	-2.09 ± 2.89
Memory + Kimi K3	4	31.2 ± 0.0	13.8 ± 0.0	$+6.2 \pm 0.0$	$+1.25 \pm 0.00$
AWM + Kimi K3	1.3	25.0 ± 32.5	23.3 ± 29.6	$+4.2 \pm 7.2$	$+1.66 \pm 2.89$
AWM + Opus 5	1	14.6 ± 3.6	12.9 ± 0.7	-1.0 ± 1.8	-0.41 ± 0.72
Naive + Opus 5	1	12.5 ± 0.0	12.5 ± 0.0	$+0.0 \pm 0.0$	$+0.00 \pm 0.00$

Table 38: Grapevine: criteria for assessing rule discovery.

Target	**Criterion**
T1	Choose at most three seeds in a deterministic 20-node linear-threshold cascade. Each episode allows one release. Feedback after each wave provides evidence about activation thresholds and incoming connections.
T2	Seed-1 synergy trap: best singles are T/D at 5, the greedy high-single trio reaches only about 8, while the exact optimum reaches 16.
T3	Before release, add or remove seeds to compare candidate seed sets across episodes and identify which nodes contribute additional activation.
T4	Deterministic exact-trio search over the discovered activation frontier, not choosing T/D/Q/L/M/B by single-node strength.
T5	16-reach triples: EFT activates KQS, then CMR, GH, JO, I, D, L and leaves A/B/N/P. EPT activates KQS, then CM, GHR, JO, I, D, L and leaves A/B/F/N.

Table 39: Grapevine: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S
Kimi K3	S	–	S	S	–
Gemini 3.5 Flash	S	–	–	S	–
Gemini 3.7 Flash	S	–	S	S	–
GPT-5.6-SOL	S	S	S	S	S
DeepSeek V4 Pro	S	–	–	–	–
Claude Sonnet 5	S	S	S	S	–
GPT-5 Mini	S	S	S	–	–
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S

D.9 HAUNTED INN

Learning task. Discover guest–room, adjacency, and count taboos through controlled placements. Re-shuffled game, scored seed 2, first 5 episodes, reference score $C = 200$.



Figure 13: Haunted Inn: raw-score learning curves within the scoring window.

Table 40: Haunted Inn: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	1	100.0	50.0	–	–
EvoTest + Kimi K3	1.3	100.0 ± 0.0	80.0 ± 20.0	-16.7 ± 28.9	-3.33 ± 5.77
Naive + Kimi K3	3	100.0 ± 0.0	35.3 ± 13.6	$+50.0 \pm 50.0$	$+13.33 \pm 15.28$
Naive + GPT-5 Mini	3.7	83.3 ± 14.4	37.5 ± 10.9	$+21.7 \pm 15.9$	$+5.83 \pm 5.77$
Codex + GPT-5.6-SOL	4.7	83.3 ± 14.4	41.2 ± 19.6	$+49.2 \pm 18.1$	$+14.17 \pm 7.78$
Reflexion + Opus 5	3.3	78.3 ± 15.9	41.3 ± 15.3	$+31.7 \pm 15.8$	$+9.67 \pm 6.83$
OpenCode + Qwen 3.8 Max	4.3	78.3 ± 8.0	38.7 ± 8.6	$+39.2 \pm 18.3$	$+13.33 \pm 2.53$
Memory + GPT-5 Mini	2.7	77.5 ± 21.4	46.0 ± 10.9	$+32.9 \pm 26.3$	$+9.92 \pm 8.43$
Memory + Kimi K3	3	75.0 ± 0.0	49.2 ± 0.6	$+49.6 \pm 1.4$	$+15.92 \pm 0.29$
Reflexion + Kimi K3	3	75.0 ± 0.0	49.0 ± 0.9	$+47.5 \pm 2.2$	$+15.25 \pm 0.87$
AWM + Kimi K3	3	75.0 ± 25.0	29.2 ± 14.2	$+6.2 \pm 10.8$	$+1.25 \pm 2.17$
AWM + Opus 5	4	75.0 ± 0.0	29.0 ± 7.8	$+25.8 \pm 21.0$	$+7.50 \pm 8.23$
Naive + Opus 5	4	75.0 ± 0.0	25.0 ± 0.0	$+12.5 \pm 0.0$	$+2.50 \pm 0.00$
Claude Code + Opus 5	5	75.0 ± 0.0	45.3 ± 1.5	$+38.3 \pm 10.1$	$+15.17 \pm 2.02$
Memory + Opus 5	3.7	74.2 ± 13.8	44.3 ± 12.1	$+45.0 \pm 6.5$	$+15.50 \pm 1.80$
EvoTest + GPT-5 Mini	3	71.7 ± 25.0	39.0 ± 12.8	$+3.3 \pm 21.3$	$+0.83 \pm 9.00$
AWM + GPT-5 Mini	3.3	71.7 ± 17.7	32.2 ± 6.3	$+7.1 \pm 8.3$	$+3.25 \pm 1.52$
OpenCode + Gemini 3.5 Flash	4.3	71.7 ± 17.7	37.3 ± 2.0	$+53.3 \pm 33.5$	$+16.50 \pm 9.50$
OpenCode + Sonnet 5	4.3	71.7 ± 17.7	34.0 ± 18.7	$+44.2 \pm 6.4$	$+14.67 \pm 3.92$
ReasoningBank + Kimi K3	3.7	70.0 ± 8.7	35.5 ± 7.1	$+33.8 \pm 1.2$	$+10.58 \pm 0.14$
EvoTest + Opus 5	3.3	68.3 ± 6.3	36.5 ± 5.8	$+17.9 \pm 8.3$	$+6.58 \pm 4.72$
OpenCode + Opus 5	4.7	67.5 ± 13.0	40.2 ± 2.3	$+34.6 \pm 6.9$	$+13.50 \pm 2.88$
ReasoningBank + Opus 5	4	66.7 ± 10.4	38.3 ± 3.6	$+26.7 \pm 8.0$	$+8.00 \pm 3.91$
OpenCode + Kimi K3	4.3	66.7 ± 14.4	25.0 ± 8.7	$+41.7 \pm 19.1$	$+13.83 \pm 6.83$
ReasoningBank + GPT-5 Mini	2.3	63.3 ± 10.1	32.7 ± 12.0	-25.0 ± 22.9	-8.25 ± 8.27
OpenCode + GPT-5.6-SOL	3.3	61.7 ± 22.7	32.0 ± 9.7	$+43.3 \pm 35.8$	$+13.33 \pm 9.27$
Reflexion + GPT-5 Mini	3	49.2 ± 3.8	24.8 ± 5.8	$+13.3 \pm 10.0$	$+5.92 \pm 1.63$
OpenCode + DeepSeek V4 Pro	3.7	48.3 ± 2.9	21.7 ± 4.5	$+30.4 \pm 9.0$	$+9.17 \pm 3.21$
OpenCode + GPT-5 Mini	4.7	30.0 ± 26.0	17.5 ± 18.2	$+17.1 \pm 10.8$	$+5.00 \pm 2.29$
OpenCode + Gemini 3.7 Flash	3	25.0 ± 13.0	13.0 ± 3.8	$+5.0 \pm 4.3$	$+2.50 \pm 0.87$

Seed-matched evidence. Labels for T2 and T6 use the three seed-2 trials used for score reporting. This seed-matched assessment covers these two targets, not the other labels, and does not determine the episode in which a target was discovered.

Table 41: Haunted Inn: criteria for assessing rule discovery.

Target	Criterion
T1	Eight rooms with visible fixed features. For the evaluated seed 2, rooms 1, 3, 6, and 7 are plain. Room 2 has a shrine, room 4 a window, room 5 is dark, and room 8 has a mirror.
T2	All three hidden taboos for the evaluated seed 2: housing at least three actors haunts every actor room. Adjacent actors and scholars haunt both rooms. Any lantern-bearing guest in the mirror room is haunted.
T3	Clues identify rooms that trigger taboos. Panic then empties the entire connected block of occupied rooms. An empty room stops that spread.
T4	Place a few guests at a time to distinguish the room that triggered a taboo from rooms affected by spreading panic. Use these comparisons to separate adjacency rules from guest-count rules.
T5	Fee-preserving packing under constraints: keep high-fee merchants/actors when legal, turn away or separate conflicts, and use empty gaps only when they save more fee than they cost.
T6	200-point seating policy on favorable queues: avoid the three taboo triggers, insert firebreaks where needed, and do not sacrifice high-fee guests for untested superstition.

Table 42: Haunted Inn: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6
backbone comparison						
Claude Opus 5	S	P	S	S	S	P
Qwen 3.8 Max	S	P	S	S	S	P
Kimi K3	S	S	S	S	S	P
Gemini 3.5 Flash	S	P	S	S	S	P
Gemini 3.7 Flash	S	P	S	S	S	P
GPT-5.6-SOL	S	P	S	S	S	P
DeepSeek V4 Pro	S	P	S	S	S	P
Claude Sonnet 5	S	P	S	S	S	P
GPT-5 Mini	S	P	S	S	S	P
agent harness comparison						
Claude Code + Opus 5	S	P	S	S	S	P
Codex + GPT-5.6-SOL	S	S	S	S	S	P

D.10 CASTAWAY

Learning task. Remember the map, recipes, hazards, and steps needed to escape. Fixed game, scored seed 1, first 5 episodes, reference score $C = 241$.

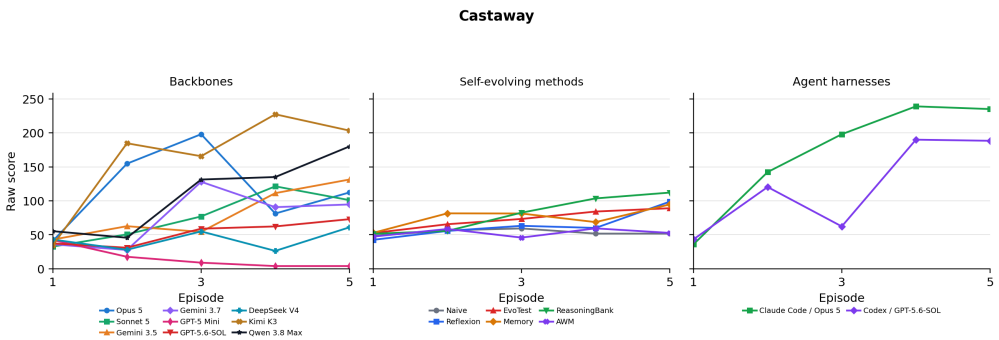


Figure 14: Castaway: raw-score learning curves within the scoring window.

Table 43: Castaway: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	4	100.0	54.8	–	–
Claude Code + Opus 5	2.7	99.4 ± 1.0	70.6 ± 3.5	$+61.3 \pm 19.2$	$+20.53 \pm 3.50$
OpenCode + Kimi K3	4	94.3 ± 1.9	67.7 ± 4.4	$+43.8 \pm 13.6$	$+15.77 \pm 4.48$
Codex + GPT-5.6-SOL	3.7	94.2 ± 3.5	50.2 ± 7.5	$+44.5 \pm 32.8$	$+14.92 \pm 8.25$
OpenCode + Qwen 3.8 Max	4.7	89.2 ± 9.7	45.4 ± 15.0	$+44.4 \pm 9.2$	$+14.05 \pm 0.95$
OpenCode + Opus 5	3.7	83.7 ± 25.0	48.7 ± 23.3	-0.4 ± 24.2	$+2.88 \pm 8.44$
ReasoningBank + Kimi K3	4.7	81.5 ± 22.1	47.0 ± 10.9	$+47.6 \pm 20.7$	$+15.02 \pm 6.27$
Memory + Kimi K3	4	70.0 ± 22.1	38.0 ± 9.6	$+16.3 \pm 13.2$	$+6.28 \pm 5.23$
EvoTest + Opus 5	3	69.4 ± 41.0	48.5 ± 34.9	$+29.8 \pm 30.7$	$+10.12 \pm 9.76$
OpenCode + Sonnet 5	4.7	67.2 ± 22.6	31.8 ± 12.9	$+28.8 \pm 9.2$	$+8.60 \pm 3.51$
OpenCode + Gemini 3.5 Flash	4.3	66.4 ± 19.8	33.4 ± 5.7	$+28.4 \pm 9.4$	$+9.35 \pm 5.06$
Reflexion + Kimi K3	4.3	55.5 ± 26.9	31.5 ± 13.2	$+25.9 \pm 32.1$	$+8.69 \pm 9.38$
OpenCode + Gemini 3.7 Flash	3.7	54.1 ± 29.6	31.3 ± 14.7	$+25.2 \pm 21.5$	$+7.48 \pm 5.87$
Memory + Opus 5	2.7	51.5 ± 36.8	34.6 ± 23.0	$+3.8 \pm 5.5$	$+3.20 \pm 3.90$
ReasoningBank + Opus 5	4	50.1 ± 10.2	34.1 ± 4.2	$+14.6 \pm 15.0$	$+4.58 \pm 3.56$
Reflexion + Opus 5	5	42.2 ± 2.0	29.8 ± 6.6	$+6.5 \pm 5.9$	$+3.28 \pm 1.76$
Naive + Opus 5	3	39.3 ± 14.3	22.5 ± 3.3	$+2.9 \pm 9.1$	$+1.13 \pm 2.14$
OpenCode + GPT-5.6-SOL	4	38.9 ± 25.8	21.8 ± 12.0	$+13.9 \pm 26.0$	$+4.26 \pm 6.48$
AWM + Opus 5	3.3	38.7 ± 7.5	25.1 ± 3.8	$+9.3 \pm 8.9$	$+2.38 \pm 2.64$
AWM + Kimi K3	3	33.5 ± 21.4	21.5 ± 5.1	-4.8 ± 9.0	-0.73 ± 1.42
Naive + GPT-5 Mini	2.7	32.4 ± 1.4	22.6 ± 3.1	-2.8 ± 5.5	-0.73 ± 2.48
Reflexion + GPT-5 Mini	4	32.1 ± 12.6	18.5 ± 4.1	$+5.4 \pm 4.7$	$+2.59 \pm 1.75$
OpenCode + DeepSeek V4 Pro	3	30.4 ± 14.2	17.7 ± 6.4	$+3.4 \pm 15.4$	$+1.42 \pm 5.21$
Memory + GPT-5 Mini	3	30.0 ± 9.5	21.8 ± 12.5	-2.1 ± 8.9	-0.68 ± 2.72
Naive + Kimi K3	2	27.7 ± 3.4	21.6 ± 4.4	-1.0 ± 9.4	-0.17 ± 2.99
ReasoningBank + GPT-5 Mini	1.7	26.7 ± 15.0	19.8 ± 6.0	$+5.3 \pm 10.7$	$+1.44 \pm 3.01$
EvoTest + Kimi K3	4	25.7 ± 4.3	23.0 ± 3.2	$+3.6 \pm 1.0$	$+1.33 \pm 0.50$
AWM + GPT-5 Mini	2.3	23.5 ± 7.5	19.1 ± 4.5	-0.6 ± 3.1	-0.18 ± 0.74
EvoTest + GPT-5 Mini	2.7	23.5 ± 6.2	19.3 ± 3.3	$+1.0 \pm 5.0$	-0.03 ± 1.92
OpenCode + GPT-5 Mini	1	17.6 ± 5.3	6.4 ± 3.5	-10.8 ± 6.9	-3.75 ± 1.90

Table 44: Castaway: criteria for assessing rule discovery.

Target	**Criterion**
T1	Seven-place seed-1 island map from beach to forest, swamp, cave, ravine, bamboo, and ruins.
T2	Resource/recipe table: wood/flint/vine/water, seed-1 food effects, beach fire, torch, tool, and raft, including mandatory tool and two-vine raft lashing.
T3	Survival and hazard model: 56 segments, five segments per day, hunger drain, day survival points, cave current, swamp mosquitoes/collapse, bamboo falling rocks/night trap, ruins crab, ravine boar, and forest monkey.
T4	True-vs-decoy escape chain: frame at cave, sail at forest, rudder at ruins are real, while compass at bamboo is the decoy that makes raft crafting fail if used as a substitute.
T5	241-point script: build fire/tool/torch, route around the two lethal traps, neutralize guards, collect all real raft parts while optionally collecting the decoy for information/score, safely stall for day points, and craft the raft at the beach.

Table 45: Castaway: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S
Kimi K3	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	–	–
Gemini 3.7 Flash	S	S	S	–	–
GPT-5.6-SOL	S	S	S	–	–
DeepSeek V4 Pro	S	S	S	–	–
Claude Sonnet 5	S	S	S	–	–
GPT-5 Mini	S	S	S	S	–
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S

D.11 Roadside Observatory

Learning task. Learn social protocols and resource requirements for new encounters. Re-shuffled game, scored seed 2, first 10 episodes, reference score $C = 1000$.

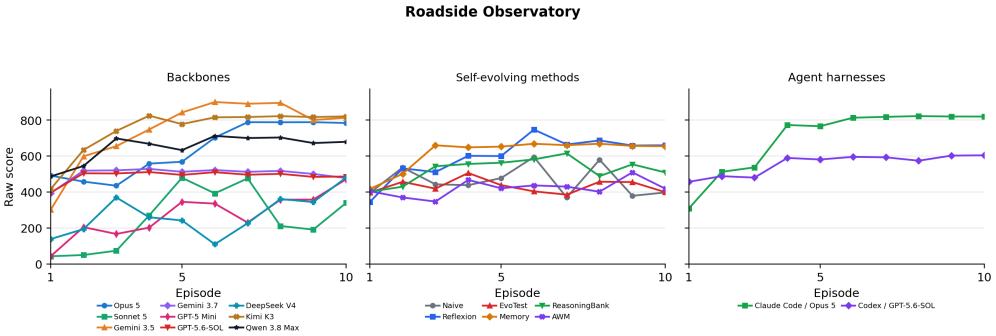


Figure 15: Roadside Observatory: raw-score learning curves within the scoring window.

Table 46: Roadside Observatory: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Reflexion + Opus 5	6	94.5 ± 5.4	74.0 ± 10.3	$+33.1 \pm 8.4$	$+4.73 \pm 1.46$
OpenCode + Gemini 3.5 Flash	5.3	90.1 ± 3.0	74.5 ± 15.8	$+31.8 \pm 28.5$	$+4.67 \pm 4.19$
Human Top-1	7	89.6	47.9	–	–
Claude Code + Opus 5	8.3	82.7 ± 25.7	69.9 ± 18.2	$+36.8 \pm 22.6$	$+5.06 \pm 2.99$
OpenCode + Kimi K3	5.3	82.5 ± 27.6	74.8 ± 26.2	$+22.3 \pm 4.4$	$+3.24 \pm 0.45$
Naive + Opus 5	6.7	79.4 ± 13.3	57.5 ± 4.8	$+5.4 \pm 8.2$	$+0.64 \pm 0.60$
OpenCode + Opus 5	7.7	79.2 ± 11.2	63.6 ± 8.3	$+32.5 \pm 11.2$	$+4.57 \pm 1.46$
Reflexion + GPT-5 Mini	4.7	77.4 ± 17.9	57.8 ± 6.5	$+4.5 \pm 7.3$	$+0.77 \pm 1.27$
Memory + Kimi K3	5	77.4 ± 23.8	70.8 ± 20.7	$+16.4 \pm 13.6$	$+2.65 \pm 1.93$
OpenCode + Qwen 3.8 Max	8	74.1 ± 21.2	65.0 ± 19.6	$+10.8 \pm 10.1$	$+1.71 \pm 1.53$
Memory + Opus 5	4.3	73.6 ± 26.6	66.4 ± 21.1	$+22.0 \pm 16.8$	$+3.15 \pm 2.43$
ReasoningBank + Kimi K3	6	73.5 ± 24.0	47.6 ± 9.9	$+0.6 \pm 11.2$	$+0.73 \pm 1.60$
ReasoningBank + GPT-5 Mini	3.3	71.2 ± 11.3	52.1 ± 6.7	-1.7 ± 7.3	-0.24 ± 0.74
Naive + Kimi K3	4	68.2 ± 21.9	37.1 ± 4.6	-3.3 ± 16.4	-0.67 ± 1.31
ReasoningBank + Opus 5	8	67.5 ± 28.1	57.5 ± 22.2	$+18.8 \pm 22.4$	$+2.71 \pm 3.09$
Reflexion + Kimi K3	7.3	67.5 ± 15.3	48.4 ± 15.8	$+24.2 \pm 21.4$	$+3.49 \pm 2.60$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
EvoTest + Kimi K3	2	63.8 ± 18.0	46.7 ± 1.5	-9.1 ± 17.1	-1.50 ± 2.70
Codex + GPT-5.6-SOL	6.7	62.1 ± 15.8	55.7 ± 9.3	$+11.9 \pm 14.9$	$+1.59 \pm 1.92$
AWM + GPT-5 Mini	5	60.0 ± 7.2	40.8 ± 7.9	$+0.6 \pm 5.0$	-0.40 ± 0.59
Naive + GPT-5 Mini	6	59.3 ± 3.8	43.7 ± 4.4	-3.9 ± 12.8	-0.85 ± 1.53
AWM + Kimi K3	8	59.1 ± 23.0	42.2 ± 6.2	$+17.6 \pm 13.5$	$+2.40 \pm 1.78$
EvoTest + Opus 5	2	57.5 ± 0.3	53.2 ± 2.2	-2.7 ± 2.8	-0.35 ± 0.32
OpenCode + Gemini 3.7 Flash	4	52.9 ± 1.8	50.1 ± 1.7	$+2.1 \pm 4.8$	$+0.35 \pm 0.79$
Memory + GPT-5 Mini	4.7	51.3 ± 8.9	48.4 ± 8.8	$+1.8 \pm 0.9$	$+0.25 \pm 0.24$
OpenCode + GPT-5.6-SOL	4	51.1 ± 0.3	48.9 ± 0.5	$+2.2 \pm 1.1$	$+0.38 \pm 0.18$
EvoTest + GPT-5 Mini	3.7	49.5 ± 11.1	29.7 ± 5.7	$+15.8 \pm 11.4$	$+1.49 \pm 1.01$
OpenCode + Sonnet 5	5.3	49.5 ± 1.2	25.3 ± 5.6	$+19.2 \pm 21.5$	$+2.96 \pm 3.01$
OpenCode + DeepSeek V4 Pro	5.7	49.4 ± 1.7	27.3 ± 14.6	$+16.1 \pm 9.5$	$+2.33 \pm 1.47$
OpenCode + GPT-5 Mini	5.7	48.9 ± 0.7	27.2 ± 13.7	$+25.7 \pm 16.7$	$+3.61 \pm 1.90$
AWM + Opus 5	6.3	47.0 ± 5.0	43.2 ± 3.6	$+2.7 \pm 5.0$	$+0.33 \pm 0.63$

Table 47: Roadside Observatory: criteria for assessing rule discovery.

Target	**Criterion**
T1	Travel-state accounting for distance, time, energy, supplies, cash, rest, bus, safe hitching, and main-road walking.
T2	Offer to help the mechanic before asking for a ride. Asking without first helping wastes the encounter.
T3	Keep energy at least 50. Ask the ranger about weather and the trail before asking for the shortcut.
T4	Hiker protocol: pay supply/cash or forage, then listen for the route clue.
T5	Collector protocol: visit landmark or keep token, then tell story/show token to receive the coded map clue.
T6	Driver protocol and trap avoidance: refuse offer, question detail, then accept only the verified safe ride.
T7	Observatory gate: obtain the map-clue item and show it at the final encounter with enough energy.
T8	Transport/score optimization: rest when energy is low, use bus only when cash permits, avoid slow-route tags, and preserve time/resource bonuses.
T9	1,000-point arrival: complete all five social protocols, visit landmark, show the map clue, arrive during the viewing window, and avoid risky-route/false-help penalties.

Table 48: Roadside Observatory: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7	T8	T9
backbone comparison									
Claude Opus 5	S	S	S	S	S	S	S	S	–
Qwen 3.8 Max	S	S	S	S	S	S	S	S	S
Kimi K3	S	S	S	S	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S	S	S	S	–
Gemini 3.7 Flash	S	S	S	S	S	S	S	S	–
GPT-5.6-SOL	S	S	S	S	S	S	–	S	–
DeepSeek V4 Pro	S	S	S	S	S	S	–	S	–
Claude Sonnet 5	S	S	S	S	S	S	–	S	–
GPT-5 Mini	S	–	–	–	–	–	–	S	–
agent harness comparison									
Claude Code + Opus 5	S	S	S	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S	S	S	–

D.12 Lost and Found

Learning task. Learn how to verify ownership without revealing private information. Re-shuffled game, scored seed 1, first 5 episodes, reference score $C = 1000$.

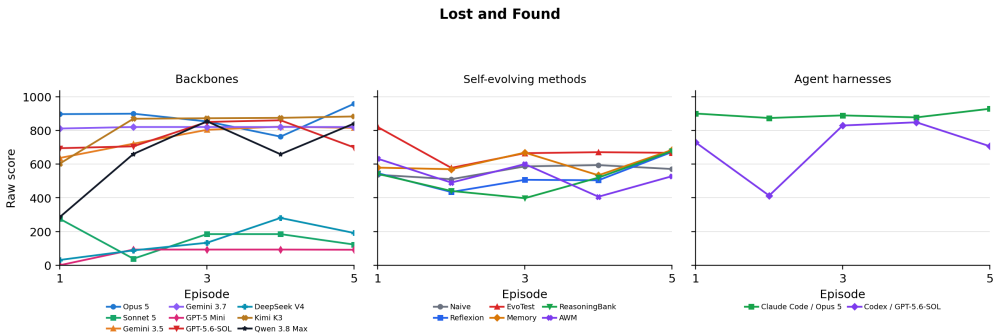


Figure 16: Lost and Found: raw-score learning curves within the scoring window.

Table 49: Lost and Found: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
AWM + Opus 5	3.3	100.0 ± 0.0	90.9 ± 5.7	-2.7 ± 20.6	$+0.13 \pm 4.48$
Memory + Opus 5	4	96.8 ± 5.6	87.7 ± 10.8	$+6.0 \pm 5.5$	$+1.92 \pm 1.34$
OpenCode + Opus 5	5	95.8 ± 4.2	87.4 ± 6.7	-3.7 ± 14.7	-0.13 ± 3.30
Memory + Kimi K3	3.7	95.1 ± 0.2	81.0 ± 15.0	$+17.8 \pm 13.6$	$+8.70 \pm 6.94$
Naive + Opus 5	2	94.5 ± 2.3	90.3 ± 2.1	-0.6 ± 2.9	-0.22 ± 0.84
Claude Code + Opus 5	4	94.1 ± 1.6	89.4 ± 3.3	$+1.6 \pm 5.6$	$+0.60 \pm 1.61$
ReasoningBank + Opus 5	2.3	93.7 ± 1.8	88.8 ± 2.1	-1.1 ± 4.5	-0.50 ± 1.49
Reflexion + Opus 5	2	93.0 ± 4.5	91.3 ± 3.9	$+3.0 \pm 3.4$	$+0.92 \pm 0.99$
Human Top-1	4	91.3	78.9	–	–
OpenCode + Kimi K3	3.3	90.1 ± 2.6	82.0 ± 10.3	$+14.2 \pm 27.0$	$+5.65 \pm 10.66$
Naive + Kimi K3	3.3	89.0 ± 4.9	52.5 ± 20.3	$+21.0 \pm 53.5$	$+8.03 \pm 14.27$
ReasoningBank + Kimi K3	3.3	88.7 ± 2.3	50.3 ± 16.2	$+34.2 \pm 37.1$	$+12.36 \pm 12.56$
EvoTest + Opus 5	2	88.5 ± 1.2	87.1 ± 0.6	-0.3 ± 0.5	-0.21 ± 0.07
Reflexion + Kimi K3	3.7	88.4 ± 3.0	49.9 ± 29.1	$+24.8 \pm 55.8$	$+8.21 \pm 20.41$
EvoTest + Kimi K3	2	87.4 ± 0.9	86.7 ± 0.3	-0.0 ± 0.7	-0.01 ± 0.28
OpenCode + GPT-5.6-SOL	2.3	86.9 ± 2.3	76.2 ± 10.8	$+8.0 \pm 34.6$	$+1.63 \pm 11.32$
AWM + Kimi K3	2.3	86.5 ± 0.2	60.4 ± 17.1	-5.1 ± 60.1	-0.74 ± 19.92
Codex + GPT-5.6-SOL	3	85.6 ± 8.6	70.5 ± 22.1	$+20.6 \pm 15.3$	$+3.88 \pm 2.55$
OpenCode + Qwen 3.8 Max	3	85.4 ± 2.4	66.0 ± 16.8	$+27.7 \pm 35.1$	$+11.09 \pm 10.90$
OpenCode + Gemini 3.5 Flash	3.3	82.4 ± 5.0	76.0 ± 5.6	$+14.0 \pm 13.4$	$+4.58 \pm 4.10$
OpenCode + Gemini 3.7 Flash	1.3	82.0 ± 4.2	81.8 ± 3.8	$+0.5 \pm 0.8$	$+0.18 \pm 0.31$
EvoTest + GPT-5 Mini	1	70.9 ± 26.8	30.1 ± 5.4	-8.7 ± 12.5	-6.18 ± 6.64
Reflexion + GPT-5 Mini	2.3	53.3 ± 25.6	18.6 ± 8.4	$+1.4 \pm 9.4$	$+0.48 \pm 3.81$
ReasoningBank + GPT-5 Mini	2.3	45.3 ± 9.5	15.6 ± 6.4	-0.9 ± 15.9	-1.36 ± 5.73
Naive + GPT-5 Mini	2.3	42.4 ± 1.8	25.0 ± 8.6	-2.7 ± 16.7	-3.26 ± 5.35
AWM + GPT-5 Mini	1	40.7 ± 0.2	8.1 ± 0.0	-20.4 ± 0.1	-8.14 ± 0.03
Memory + GPT-5 Mini	1	40.3 ± 0.3	13.4 ± 9.2	-13.5 ± 11.6	-5.40 ± 4.62
OpenCode + DeepSeek V4 Pro	4.3	28.2 ± 28.4	14.5 ± 20.5	$+17.6 \pm 19.8$	$+5.13 \pm 6.58$
OpenCode + Sonnet 5	1	27.5 ± 47.7	16.1 ± 27.9	-0.3 ± 0.6	-1.59 ± 2.76
OpenCode + GPT-5 Mini	1.3	9.3 ± 16.1	7.4 ± 12.9	$+4.6 \pm 8.0$	$+1.84 \pm 3.19$

Table 50: Lost and Found: criteria for assessing rule discovery.

Target	**Criterion**
T1	Three reshuffled cases per episode with four claimant roles: owner, finder, associate, and mimic.
T2	Inspect the item privately to learn its concealed marker. Opening it publicly or asking questions too early reveals details that later claimants can repeat.
T3	Verify the concealed detail, the claimant's relationship to the item and its purpose, the loss timeline, and the relevant external record. A matching decoy record is not sufficient.

Target	Criterion (continued)
T4	Resource budgeting: 24 turns, eight privacy tokens, three isolation tokens, and three record checks across all cases.
T5	1,000-point office protocol: inspect each item privately, verify owner on all independent channels, check the matching record, and return or store without leaking decisive evidence.

Table 51: Lost and Found: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	–
Kimi K3	S	S	S	S	–
Gemini 3.5 Flash	S	S	S	S	–
Gemini 3.7 Flash	S	S	S	S	–
GPT-5.6-SOL	S	S	S	S	–
DeepSeek V4 Pro	S	S	S	–	–
Claude Sonnet 5	S	S	S	S	–
GPT-5 Mini	S	S	S	S	–
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S

D.13 MOLA TEA

Learning task. Infer the adoption chain and when each promotion becomes effective. Re-shuffled game, scored seed 1, first 10 episodes, reference score $C = 1000$.

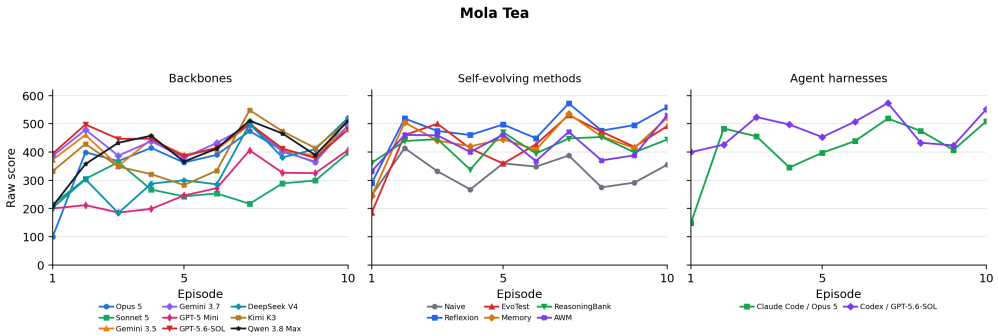


Figure 17: Mola Tea: raw-score learning curves within the scoring window.

Table 52: Mola Tea: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Reflexion + Opus 5	10	79.0 ± 13.6	63.8 ± 5.8	$+18.0 \pm 13.9$	$+2.96 \pm 1.80$
AWM + Kimi K3	8.7	74.8 ± 18.1	46.1 ± 14.5	$+5.6 \pm 3.3$	$+0.98 \pm 0.83$
Human Top-1	7	68.4	40.7	–	–
ReasoningBank + Opus 5	8.7	66.1 ± 4.8	51.7 ± 4.1	$+5.6 \pm 8.1$	$+1.36 \pm 1.01$
EvoTest + Opus 5	3	65.8 ± 3.6	52.9 ± 4.5	$+1.6 \pm 6.3$	$+0.92 \pm 0.84$
AWM + Opus 5	5.3	64.3 ± 4.3	52.4 ± 1.8	$+2.6 \pm 3.8$	$+0.88 \pm 0.11$
Memory + Opus 5	7	62.6 ± 9.6	52.6 ± 6.4	$+9.4 \pm 2.8$	$+1.75 \pm 0.58$
Reflexion + Kimi K3	3.7	60.7 ± 8.5	44.1 ± 7.2	$+1.8 \pm 11.9$	$+0.63 \pm 1.56$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
ReasoningBank + GPT-5 Mini	3.7	60.5 ± 16.5	38.9 ± 3.0	$+11.2 \pm 15.2$	$+1.64 \pm 2.07$
ReasoningBank + Kimi K3	4.7	59.6 ± 12.2	35.2 ± 6.3	-11.8 ± 10.6	-1.62 ± 1.01
Codex + GPT-5.6-SOL	8	57.4 ± 11.7	47.9 ± 6.3	$+2.0 \pm 3.7$	$+0.71 \pm 0.47$
Memory + GPT-5 Mini	5	56.5 ± 9.1	35.2 ± 5.7	$+1.6 \pm 7.1$	$+0.76 \pm 0.87$
OpenCode + Kimi K3	8	55.1 ± 8.0	39.9 ± 6.3	$+9.6 \pm 6.3$	$+1.73 \pm 0.92$
Memory + Kimi K3	3.7	54.2 ± 7.4	43.8 ± 3.7	$+9.0 \pm 10.2$	$+1.56 \pm 1.50$
Naive + Opus 5	2	54.1 ± 6.5	37.6 ± 2.5	-3.1 ± 2.1	-0.22 ± 0.24
EvoTest + Kimi K3	4	54.0 ± 3.6	39.1 ± 2.6	$+6.6 \pm 6.4$	$+1.57 \pm 0.69$
Naive + GPT-5 Mini	3	52.7 ± 12.6	29.0 ± 4.3	-6.7 ± 13.2	-0.15 ± 1.98
OpenCode + DeepSeek V4 Pro	8	52.7 ± 4.2	33.8 ± 6.9	$+20.7 \pm 1.4$	$+3.16 \pm 0.47$
Claude Code + Opus 5	8	52.0 ± 0.0	41.8 ± 0.4	$+10.1 \pm 5.1$	$+2.04 \pm 0.76$
OpenCode + Opus 5	9	52.0 ± 0.0	38.0 ± 2.0	$+14.0 \pm 7.5$	$+2.38 \pm 1.06$
OpenCode + Qwen 3.8 Max	8	51.8 ± 0.3	41.1 ± 2.5	$+12.2 \pm 3.6$	$+2.01 \pm 0.47$
OpenCode + GPT-5.6-SOL	5.3	50.1 ± 1.6	43.6 ± 1.6	-2.2 ± 3.7	-0.03 ± 0.57
OpenCode + Gemini 3.5 Flash	8	50.0 ± 1.2	42.1 ± 0.5	$+3.4 \pm 5.2$	$+0.62 \pm 0.50$
OpenCode + Gemini 3.7 Flash	7	49.6 ± 0.0	42.6 ± 0.4	$+0.3 \pm 1.2$	$+0.29 \pm 0.11$
Reflexion + GPT-5 Mini	7	49.1 ± 2.0	35.9 ± 1.3	$+4.9 \pm 7.1$	$+1.04 \pm 0.88$
OpenCode + Sonnet 5	7.3	48.7 ± 0.1	28.5 ± 0.9	$+3.4 \pm 14.3$	$+0.66 \pm 1.78$
EvoTest + GPT-5 Mini	6.7	48.2 ± 2.6	35.7 ± 4.1	$+15.4 \pm 10.8$	$+2.50 \pm 1.21$
AWM + GPT-5 Mini	6	47.1 ± 7.3	28.8 ± 4.7	-4.5 ± 11.5	-0.13 ± 1.83
Naive + Kimi K3	6.7	44.5 ± 7.3	31.8 ± 2.9	$+2.7 \pm 9.7$	$+0.70 \pm 1.80$
OpenCode + GPT-5 Mini	8	41.0 ± 14.2	27.8 ± 6.3	$+15.4 \pm 10.3$	$+2.43 \pm 1.67$

Table 53: Mola Tea: criteria for assessing rule discovery.

Target	**Criterion**
T1	21-turn, seven-day shop economy with cash, unit cost, launch stock, stockouts, expired inventory, and regular-customer churn.
T2	Recipe rule: only products at distance 1 from the classic drink are viable, while far variants can sell briefly but fail the adoption chain.
T3	Customer-role inference from clues: Explorer, Echo, Verifier, Broadcaster, and Anchor/regulars have distinct responses to sample, feature, discount, gift, and bundle.
T4	Stage transition chain LAUNCHED→SEEDED→REPEATED→VERIFIED→TRENDING, with clean intervals and short verification/diffusion deadlines.
T5	Sampling Explorers starts adoption. Feature promotion works after a natural repeat purchase. A Broadcaster gift works only after verification. Bundles pay off only after the product is trending. Repeated discounts teach customers to expect low prices.
T6	Inventory/novelty control: restock before return and bundle windows, use hold/restore-classic to reduce novelty pressure, and avoid leftover-expiry costs.
T7	708-point schedule: develop a distance-1 card, launch, sample Explorers, large-restock through the natural repeat, feature within deadline, gift cheapest merch to Broadcasters, then bundle and manage stock/novelty to close the week.

Table 54: Mola Tea: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7
backbone comparison							
Claude Opus 5	S	S	S	S	S	S	–
Qwen 3.8 Max	S	S	S	S	S	S	–
Kimi K3	S	S	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S	S	–
Gemini 3.7 Flash	S	S	–	S	S	S	–
GPT-5.6-SOL	S	S	S	S	S	S	–
DeepSeek V4 Pro	S	S	–	S	S	S	–
Claude Sonnet 5	S	S	S	S	S	S	–
GPT-5 Mini	S	–	–	–	S	S	–
agent harness comparison							
Claude Code + Opus 5	S	S	S	S	S	S	–
Codex + GPT-5.6-SOL	S	S	S	S	S	S	S

D.14 PLAGUE

Learning task. Locate asymptomatic carriers from spread patterns and isolate them early. Reshuffled game, scored seed 2, first 5 episodes, reference score $C = 100$.

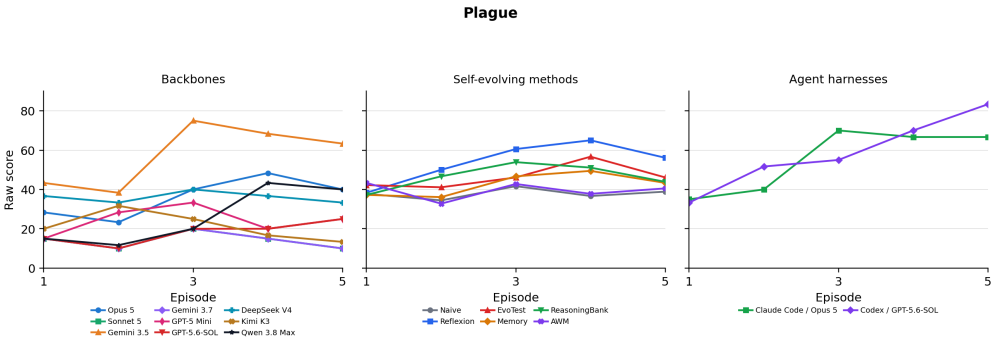


Figure 18: Plague: raw-score learning curves within the scoring window.

Table 55: Plague: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	2	100.0	58.0	–	–
Reflexion + GPT-5 Mini	3	100.0 ± 0.0	65.0 ± 12.0	$+20.0 \pm 43.4$	$+6.50 \pm 11.76$
Codex + GPT-5.6-SOL	4.7	90.0 ± 17.3	58.7 ± 22.2	$+34.2 \pm 11.8$	$+11.83 \pm 2.47$
Claude Code + Opus 5	4	76.7 ± 10.4	55.7 ± 15.7	$+29.2 \pm 16.6$	$+9.00 \pm 5.29$
OpenCode + Gemini 3.5 Flash	2.3	75.0 ± 35.0	57.7 ± 40.5	$+25.0 \pm 39.1$	$+7.00 \pm 11.72$
EvoTest + Opus 5	4	75.0 ± 21.8	50.0 ± 5.3	$+19.2 \pm 22.4$	$+6.33 \pm 7.94$
ReasoningBank + GPT-5 Mini	2.3	73.3 ± 30.6	59.7 ± 26.1	$+13.3 \pm 17.7$	$+4.83 \pm 7.42$
Reflexion + Opus 5	4.7	71.7 ± 2.9	48.7 ± 1.2	$+17.5 \pm 10.9$	$+5.00 \pm 5.20$
AWM + GPT-5 Mini	4.3	63.3 ± 32.1	47.0 ± 18.5	$+5.0 \pm 2.5$	$+1.67 \pm 1.44$
Memory + Opus 5	4	61.7 ± 12.6	47.3 ± 7.6	$+15.8 \pm 7.6$	$+4.17 \pm 1.76$
EvoTest + GPT-5 Mini	2	60.0 ± 0.0	45.0 ± 7.5	-11.7 ± 5.2	-5.33 ± 0.76
Reflexion + Kimi K3	3	58.3 ± 2.9	48.3 ± 5.0	$+11.7 \pm 9.5$	$+3.67 \pm 3.06$
Memory + Kimi K3	3	58.3 ± 2.9	46.3 ± 2.1	$+11.7 \pm 12.3$	$+3.50 \pm 2.65$
OpenCode + Opus 5	4	58.3 ± 40.1	36.0 ± 31.4	$+18.3 \pm 17.6$	$+4.83 \pm 6.11$
ReasoningBank + Opus 5	2.3	56.7 ± 7.6	42.7 ± 4.2	-2.5 ± 10.0	-0.84 ± 2.52
EvoTest + Kimi K3	3.7	56.7 ± 2.9	44.3 ± 1.5	$+21.7 \pm 3.8$	$+6.00 \pm 1.32$
ReasoningBank + Kimi K3	4	56.7 ± 7.6	37.3 ± 2.3	$+5.8 \pm 13.8$	$+1.33 \pm 3.06$
Naive + GPT-5 Mini	3.7	50.0 ± 10.0	37.0 ± 3.0	$+5.8 \pm 8.0$	$+1.67 \pm 3.33$
OpenCode + Qwen 3.8 Max	3.3	46.7 ± 46.2	26.0 ± 20.8	$+28.3 \pm 49.1$	$+8.17 \pm 15.01$
Naive + Kimi K3	4	46.7 ± 2.9	38.0 ± 1.7	$+0.0 \pm 10.9$	$+0.50 \pm 3.12$
AWM + Kimi K3	2.3	45.0 ± 0.0	35.3 ± 0.6	-0.0 ± 5.0	-0.84 ± 1.53
Naive + Opus 5	3	45.0 ± 0.0	38.7 ± 1.2	-0.8 ± 5.2	-0.83 ± 1.15
AWM + Opus 5	1	40.0 ± 0.0	36.0 ± 1.0	-1.7 ± 1.4	-1.00 ± 0.50
OpenCode + DeepSeek V4 Pro	2.3	40.0 ± 34.6	36.0 ± 38.1	$+0.0 \pm 0.0$	-0.33 ± 0.29
Memory + GPT-5 Mini	3	40.0 ± 0.0	34.0 ± 0.0	$+1.7 \pm 2.9$	$+0.00 \pm 0.87$
OpenCode + Kimi K3	2.7	36.7 ± 17.6	21.3 ± 6.7	-10.8 ± 11.3	-2.83 ± 2.25
OpenCode + GPT-5 Mini	2.7	35.0 ± 21.8	24.3 ± 15.4	$+0.8 \pm 8.8$	$+1.17 \pm 4.25$
OpenCode + GPT-5.6-SOL	3.7	31.7 ± 20.2	18.0 ± 6.9	$+10.0 \pm 17.3$	$+3.00 \pm 6.06$
OpenCode + Sonnet 5	3	20.0 ± 0.0	14.0 ± 0.0	$+0.0 \pm 0.0$	-0.50 ± 0.00
OpenCode + Gemini 3.7 Flash	3	20.0 ± 0.0	14.0 ± 0.0	$+0.0 \pm 0.0$	-0.50 ± 0.00

Table 56: Plague: criteria for assessing rule discovery.

Target	Criterion
T1	Line and action economy: 20 people, 14 nights, four AP per day, three quarantine cells, severity 1–3, and death after severity > 3.
T2	Hidden-carrier dynamics: two asymptomatic carriers are the only spreaders, reshuffled into left/right halves, with radius $1 + \lfloor age/2 \rfloor$, while symptomatic non-carriers do not infect anyone.
T3	Intervention semantics: quarantine blocks both catching and spreading, whereas treatment only reduces current severity and gives one-night immunity after cure.
T4	Carrier diagnosis from spatial evidence, especially separated sick pairs around a healthy center and disappearance of new cases after quarantining that center.
T5	AP triage: quarantine likely carriers early, treat only patients near death, and release/replace false quarantines without exceeding three cells.
T6	For a 100-point containment policy, identify both carriers by day 2–3 and quarantine them before their spread radius grows. Time treatments so that no other villager reaches the death threshold.

Table 57: Plague: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6
backbone comparison						
Claude Opus 5	S	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S	S
Kimi K3	S	S	S	S	S	–
Gemini 3.5 Flash	S	S	S	S	S	S
Gemini 3.7 Flash	S	S	S	S	S	S
GPT-5.6-SOL	S	S	S	S	–	–
DeepSeek V4 Pro	S	S	S	S	S	S
Claude Sonnet 5	S	S	S	–	S	–
GPT-5 Mini	S	S	S	–	S	–
agent harness comparison						
Claude Code + Opus 5	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S

D.15 Patch Reality

Learning task. Remember diagnosis–repair mappings, decoys, and prerequisites. Fixed game, scored seed 1, first 5 episodes, reference score $C = 960$.

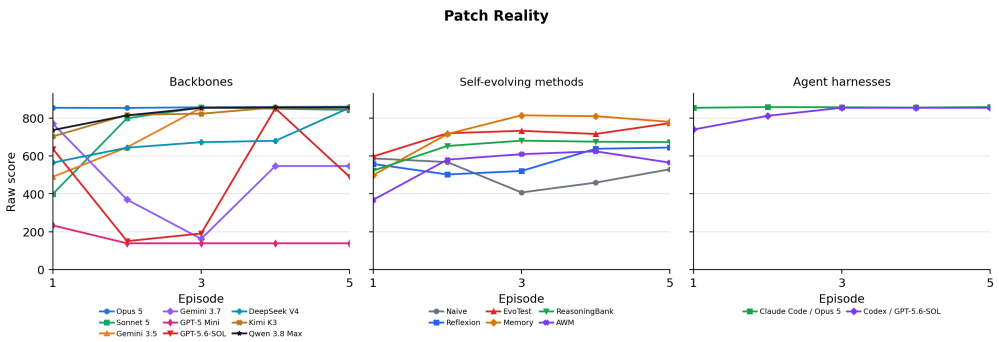


Figure 19: Patch Reality: raw-score learning curves within the scoring window.

Table 58: Patch Reality: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
EvoTest + Opus 5	3.7	89.7 ± 0.4	86.9 ± 2.6	$+2.3 \pm 3.6$	$+1.30 \pm 1.19$
ReasoningBank + Opus 5	4.3	89.7 ± 0.4	85.2 ± 2.7	$+6.1 \pm 0.2$	$+2.44 \pm 0.10$
Claude Code + Opus 5	2.7	89.6 ± 0.0	89.2 ± 0.1	$+0.1 \pm 0.5$	$+0.06 \pm 0.11$
OpenCode + Opus 5	2.7	89.6 ± 0.0	89.2 ± 0.1	$+0.6 \pm 0.3$	$+0.17 \pm 0.05$
Human Top-1	5	89.6	59.2	–	–
Reflexion + Opus 5	3.7	89.5 ± 0.4	84.5 ± 6.1	$+0.2 \pm 5.7$	$+0.09 \pm 2.32$
EvoTest + Kimi K3	3	89.3 ± 0.0	84.1 ± 2.4	$+12.2 \pm 6.1$	$+4.15 \pm 1.84$
Memory + Opus 5	3	89.3 ± 0.0	82.5 ± 4.9	$+4.8 \pm 8.3$	$+1.94 \pm 3.33$
Memory + Kimi K3	3	89.3 ± 0.0	71.5 ± 9.2	$+43.8 \pm 23.1$	$+14.83 \pm 6.93$
OpenCode + Qwen 3.8 Max	3	89.2 ± 0.4	85.8 ± 1.4	$+8.4 \pm 3.2$	$+2.93 \pm 0.64$
OpenCode + Kimi K3	3	89.2 ± 0.4	84.5 ± 2.7	$+10.0 \pm 6.8$	$+3.60 \pm 2.07$
OpenCode + Sonnet 5	3	89.2 ± 0.4	78.0 ± 2.3	$+25.7 \pm 4.3$	$+9.76 \pm 1.78$
Codex + GPT-5.6-SOL	2	89.0 ± 0.0	85.7 ± 2.7	$+8.2 \pm 6.6$	$+2.83 \pm 2.45$
OpenCode + Gemini 3.5 Flash	2.7	89.0 ± 0.0	77.0 ± 12.6	$+29.8 \pm 31.5$	$+9.76 \pm 9.09$
OpenCode + DeepSeek V4 Pro	2.7	89.0 ± 0.0	71.1 ± 26.7	$+17.0 \pm 19.7$	$+6.41 \pm 7.26$
OpenCode + GPT-5.6-SOL	1.3	88.7 ± 0.5	48.3 ± 4.1	$+28.9 \pm 44.9$	$+4.24 \pm 11.63$
ReasoningBank + Kimi K3	3.7	85.4 ± 6.7	72.3 ± 10.0	$+5.6 \pm 11.0$	$+3.00 \pm 3.36$
Naive + Opus 5	3.3	85.0 ± 7.0	67.4 ± 2.5	$+9.2 \pm 13.3$	$+3.59 \pm 3.35$
OpenCode + Gemini 3.7 Flash	1.3	84.1 ± 8.5	49.9 ± 38.0	-2.4 ± 30.3	-2.81 ± 10.90
Reflexion + Kimi K3	4.7	80.9 ± 6.1	38.5 ± 18.4	$+31.9 \pm 7.4$	$+8.33 \pm 4.86$
AWM + Opus 5	3.3	78.1 ± 10.9	67.1 ± 12.5	$+6.7 \pm 12.0$	$+1.55 \pm 3.63$
AWM + Kimi K3	3	77.7 ± 0.0	62.5 ± 3.1	$+37.3 \pm 7.7$	$+14.52 \pm 2.40$
Naive + Kimi K3	2.7	77.5 ± 0.2	43.1 ± 6.0	-20.5 ± 9.6	-5.14 ± 1.82
Memory + GPT-5 Mini	2.7	77.1 ± 0.7	72.2 ± 3.1	$+10.1 \pm 7.6$	$+3.76 \pm 2.30$
Reflexion + GPT-5 Mini	2.3	65.7 ± 0.2	55.9 ± 5.1	$+2.6 \pm 10.2$	$+1.23 \pm 4.01$
Naive + GPT-5 Mini	1.3	62.7 ± 2.0	48.9 ± 2.8	-14.6 ± 7.7	-5.44 ± 0.86
EvoTest + GPT-5 Mini	5	62.6 ± 8.1	50.1 ± 3.9	$+12.4 \pm 1.7$	$+5.42 \pm 1.98$
ReasoningBank + GPT-5 Mini	3.3	59.1 ± 6.7	42.7 ± 5.4	$+15.4 \pm 22.2$	$+4.67 \pm 7.20$
AWM + GPT-5 Mini	2	51.8 ± 9.1	42.0 ± 13.6	-6.5 ± 7.7	-2.42 ± 1.77
OpenCode + GPT-5 Mini	1.3	28.1 ± 11.2	16.4 ± 7.1	-4.9 ± 13.8	-1.97 ± 5.51

Table 59: Patch Reality: criteria for assessing rule discovery.

Target	**Criterion**
T1	Per-level action grammar: observe evidence, optionally audit risky records, diagnose source/type/strategy, then patch the root object.
T2	Levels 1–3 mappings: coin/drain uses environment-cover pavement friction, vibrating cup requires auditing residual patch then expiring it, and flickering key requires object physics repair by solidifying the key.
T3	Levels 4–5 mappings: hospital ledger is an identity-record trap solved by binding the wristband, while premature echo is a time-order conflict solved by aligning the acoustic log.
T4	Level 6 ordered record repair: reconcile the ledger before halting archive reindexing.
T5	Level 7 public-record exposure solved by camera desync.
T6	Level 8 identity-anchor pollution solved by reclaiming the name anchor.
T7	Audit discipline: audit only residual patch, hospital ledger, acoustic log, security log, broadcast, and agent offer when they gate evidence, because extra/decoy information costs score.
T8	860 campaign: clear all eight levels with the exact diagnosis triples and root patches, avoid reward traps and nonessential audits, and preserve focus through the multi-level sequence.

Table 60: Patch Reality: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7	T8
backbone comparison								
Claude Opus 5	S	S	S	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S	S	S	S
Kimi K3	S	S	S	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S	S	S	–
Gemini 3.7 Flash	S	S	S	S	S	S	S	–
GPT-5.6-SOL	S	S	S	S	S	S	S	–
DeepSeek V4 Pro	S	S	S	S	S	S	S	–
Claude Sonnet 5	S	S	S	S	S	S	S	S
GPT-5 Mini	S	S	S	–	–	–	S	–
agent harness comparison								
Claude Code + Opus 5	S	S	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S	S	–

D.16 POISONER

Learning task. Infer taste constraints from accepted and rejected dishes and apply them to new menus. Re-shuffled game, scored seed 2, first 5 episodes, reference score $C = 100$.

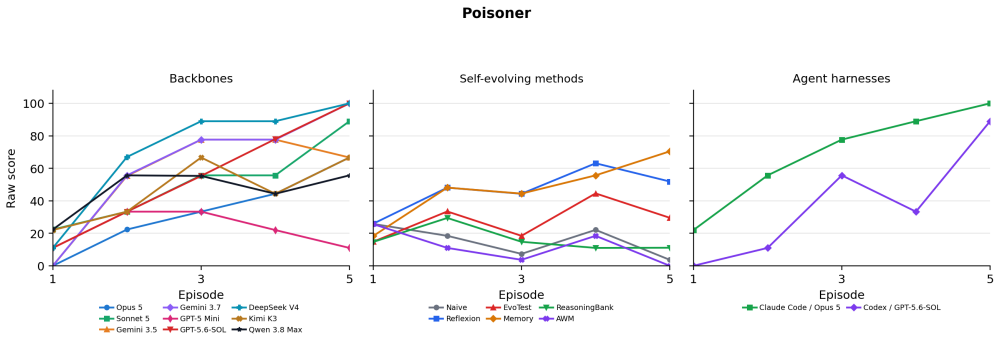


Figure 20: Poisoner: raw-score learning curves within the scoring window.

Table 61: Poisoner: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	3	100.0	66.8	–	–
OpenCode + Gemini 3.7 Flash	3.3	100.0 ± 0.0	62.2 ± 25.3	$+61.0 \pm 9.5$	$+22.20 \pm 1.91$
OpenCode + DeepSeek V4 Pro	3.7	100.0 ± 0.0	71.2 ± 3.8	$+55.5 \pm 19.1$	$+20.00 \pm 5.72$
Claude Code + Opus 5	3.7	100.0 ± 0.0	68.9 ± 19.3	$+55.7 \pm 9.8$	$+18.93 \pm 3.87$
OpenCode + GPT-5.6-SOL	4.3	100.0 ± 0.0	55.5 ± 21.4	$+66.8 \pm 16.8$	$+22.27 \pm 5.10$
Reflexion + Kimi K3	2.7	89.0 ± 19.1	57.8 ± 36.7	$+22.2 \pm 58.7$	$+8.87 \pm 16.81$
EvoTest + Opus 5	3.7	89.0 ± 19.1	31.1 ± 7.7	$+27.8 \pm 67.4$	$+8.90 \pm 19.52$
OpenCode + Sonnet 5	4.3	89.0 ± 19.1	51.2 ± 25.3	$+44.5 \pm 10.0$	$+15.57 \pm 5.16$
Codex + GPT-5.6-SOL	4.3	89.0 ± 19.1	37.8 ± 4.0	$+55.7 \pm 19.2$	$+20.03 \pm 3.30$
EvoTest + Kimi K3	2.3	78.0 ± 19.1	37.9 ± 31.5	-16.8 ± 60.3	-3.37 ± 18.63
Reflexion + Opus 5	2.7	78.0 ± 19.1	59.9 ± 23.1	$+22.5 \pm 19.5$	$+7.87 \pm 7.00$
Memory + Opus 5	3.7	78.0 ± 19.1	55.5 ± 27.0	$+28.2 \pm 9.7$	$+10.13 \pm 3.30$
OpenCode + Gemini 3.5 Flash	2.3	77.7 ± 38.7	55.5 ± 25.4	$+44.5 \pm 34.8$	$+15.57 \pm 10.22$
OpenCode + Kimi K3	3.7	77.7 ± 38.7	46.6 ± 37.1	$+27.8 \pm 19.2$	$+10.03 \pm 5.77$
Reflexion + GPT-5 Mini	3.3	67.0 ± 0.0	22.3 ± 3.9	$+16.8 \pm 16.8$	$+3.37 \pm 10.05$
OpenCode + Qwen 3.8 Max	2.3	66.7 ± 33.5	46.7 ± 37.3	$+11.0 \pm 25.2$	$+5.53 \pm 7.65$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Memory + Kimi K3	2.7	66.7 ± 57.7	40.0 ± 37.1	$+28.0 \pm 34.8$	$+11.17 \pm 11.74$
Memory + GPT-5 Mini	3	66.7 ± 57.7	46.7 ± 41.7	$+33.3 \pm 28.9$	$+12.23 \pm 10.72$
OpenCode + Opus 5	4.3	66.7 ± 33.5	33.3 ± 35.3	$+44.3 \pm 25.5$	$+15.53 \pm 8.41$
EvoTest + GPT-5 Mini	4.3	55.7 ± 19.6	15.5 ± 7.7	$+27.8 \pm 9.8$	$+6.67 \pm 0.06$
ReasoningBank + GPT-5 Mini	1.7	44.3 ± 51.0	15.5 ± 21.4	-16.7 ± 16.8	-3.33 ± 3.35
Naive + Kimi K3	2	44.3 ± 19.6	13.3 ± 11.7	-22.3 ± 42.1	-7.80 ± 11.75
Naive + GPT-5 Mini	2.7	44.3 ± 19.6	13.3 ± 6.7	$+11.2 \pm 25.4$	$+2.23 \pm 7.65$
ReasoningBank + Kimi K3	4	44.3 ± 19.6	15.5 ± 10.2	$+0.2 \pm 16.5$	$+2.27 \pm 5.06$
OpenCode + GPT-5 Mini	1.7	33.3 ± 33.5	22.1 ± 20.3	-5.7 ± 25.5	-1.13 ± 8.36
AWM + GPT-5 Mini	2	33.3 ± 33.5	8.9 ± 10.2	$+5.7 \pm 25.5$	$+0.03 \pm 6.65$
AWM + Opus 5	1	33.0 ± 0.0	19.8 ± 0.0	-16.5 ± 0.0	-6.60 ± 0.00
Naive + Opus 5	1	33.0 ± 0.0	19.8 ± 0.0	-16.5 ± 0.0	-6.60 ± 0.00
ReasoningBank + Opus 5	1	33.0 ± 0.0	17.6 ± 10.1	-16.5 ± 0.0	-6.60 ± 0.00
AWM + Kimi K3	1	33.0 ± 0.0	6.6 ± 0.0	-16.5 ± 0.0	-6.60 ± 0.00

Seed-matched evidence. Labels for T2, T4, and T5 use the three seed-2 trials used for score reporting. This seed-matched assessment covers these three targets, not the other labels, and does not determine the episode in which a target was discovered.

Table 62: Poisoner: criteria for assessing rule discovery.

Target	Criterion
T1	Three-banquet inference interface: eight dishes per banquet, one serve action per banquet, score $0/33/67/100$, and only one labeled accept/reject outcome per menu.
T2	The exact five-feature conjunction for the evaluated seed 2: not sweet, spicy, meaty, fragrant, and soft are required. Sourness, saltiness, and temperature are irrelevant.
T3	Wildcard and near-miss handling: each menu contains at least one full match and at least one 4-of-5 bait, so four-feature theories fail.
T4	For each new menu, check the learned combination of required features. Ignore irrelevant traits and distinguish a full match from dishes satisfying only four requirements.
T5	3/3 regicide: carry the exact five locked features across banquets and serve a full match in every reshuffled menu.

Table 63: Poisoner: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S
Kimi K3	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S
Gemini 3.7 Flash	S	S	S	S	S
GPT-5.6-SOL	S	S	S	S	S
DeepSeek V4 Pro	S	S	S	S	S
Claude Sonnet 5	S	S	S	S	S
GPT-5 Mini	S	P	–	P	–
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S

D.17 ROOMMATES

Learning task. Infer pairwise and nonlinear grouping rules from room-level feedback. Re-shuffled game, scored seed 1, first 5 episodes, reference score $C = 7$.

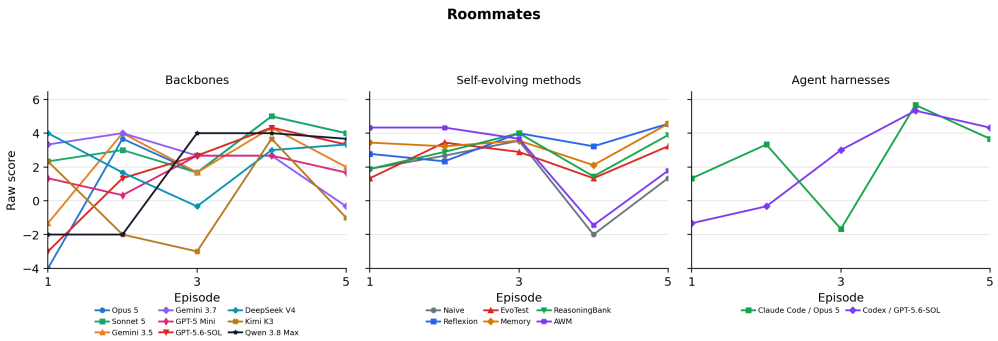


Figure 21: Roommates: raw-score learning curves within the scoring window.

Table 64: Roommates: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Human Top-1	3	100.0	-4.7	–	–
Reflexion + Opus 5	3.7	100.0 ± 0.0	71.4 ± 7.6	$+28.6 \pm 18.9$	$+8.11 \pm 5.41$
ReasoningBank + Opus 5	3.3	90.5 ± 8.2	55.2 ± 1.6	$+42.8 \pm 31.1$	$+13.31 \pm 5.02$
Reflexion + GPT-5 Mini	4.3	85.7 ± 24.7	39.0 ± 36.7	$+9.5 \pm 14.9$	$+6.19 \pm 5.77$
Claude Code + Opus 5	4	81.0 ± 8.2	35.2 ± 25.8	$+33.4 \pm 36.0$	$+10.03 \pm 13.63$
OpenCode + DeepSeek V4 Pro	4.7	81.0 ± 8.2	33.3 ± 8.7	$+4.7 \pm 36.0$	-0.01 ± 13.09
Codex + GPT-5.6-SOL	3.3	76.2 ± 16.5	31.4 ± 33.0	$+80.9 \pm 66.0$	$+24.26 \pm 17.55$
OpenCode + Sonnet 5	3.7	76.2 ± 16.5	45.7 ± 12.5	$+26.2 \pm 21.8$	$+7.63 \pm 5.02$
OpenCode + Opus 5	3.7	76.2 ± 16.5	29.5 ± 11.5	$+66.6 \pm 16.5$	$+24.76 \pm 5.77$
OpenCode + Qwen 3.8 Max	4	76.2 ± 16.5	21.9 ± 23.3	$+83.4 \pm 32.2$	$+24.77 \pm 5.95$
AWM + Opus 5	1	71.4 ± 24.7	45.7 ± 17.8	-50.0 ± 12.4	-12.39 ± 4.36
Memory + Kimi K3	3.7	71.4 ± 14.3	54.3 ± 2.9	-7.1 ± 7.1	-0.00 ± 2.86
Memory + Opus 5	3.7	71.4 ± 14.3	44.8 ± 4.4	$+11.9 \pm 10.9$	$+3.80 \pm 2.97$
EvoTest + Opus 5	5	71.4 ± 0.0	45.7 ± 11.4	$+9.6 \pm 23.0$	$+3.34 \pm 4.59$
OpenCode + Kimi K3	2	66.7 ± 16.5	0.0 ± 7.6	$+16.7 \pm 10.9$	-1.41 ± 6.55
EvoTest + GPT-5 Mini	2.7	66.7 ± 16.5	35.3 ± 19.4	$+30.9 \pm 10.9$	$+13.33 \pm 5.95$
Memory + GPT-5 Mini	3.7	66.7 ± 8.2	45.7 ± 7.6	-4.7 ± 32.2	$+0.97 \pm 12.88$
Naive + Opus 5	1.7	61.9 ± 8.2	32.4 ± 1.6	-64.3 ± 0.0	-17.14 ± 0.00
ReasoningBank + GPT-5 Mini	2.3	61.9 ± 8.2	27.6 ± 6.6	-35.7 ± 49.5	-5.71 ± 19.79
OpenCode + Gemini 3.5 Flash	2.7	61.9 ± 8.2	30.5 ± 3.3	$+26.1 \pm 28.9$	$+9.99 \pm 12.37$
OpenCode + GPT-5 Mini	2.7	61.9 ± 8.2	24.8 ± 6.6	$+19.1 \pm 64.4$	$+4.31 \pm 18.90$
OpenCode + GPT-5.6-SOL	2.7	61.9 ± 8.2	24.7 ± 17.5	$+66.6 \pm 54.1$	$+22.37 \pm 14.31$
AWM + GPT-5 Mini	1	57.1 ± 0.0	31.4 ± 0.0	-64.3 ± 0.0	-17.14 ± 0.00
AWM + Kimi K3	1	57.1 ± 0.0	31.4 ± 0.0	-64.3 ± 0.0	-17.14 ± 0.00
ReasoningBank + Kimi K3	1.3	57.1 ± 0.0	38.1 ± 8.2	$+4.8 \pm 39.3$	$+3.34 \pm 15.80$
OpenCode + Gemini 3.7 Flash	1.3	57.1 ± 0.0	35.3 ± 5.9	-35.6 ± 12.4	-12.36 ± 5.95
Reflexion + Kimi K3	1.3	57.1 ± 0.0	34.3 ± 4.9	$+19.1 \pm 25.1$	$+4.77 \pm 9.18$
EvoTest + Kimi K3	1.3	57.1 ± 0.0	23.8 ± 13.2	-45.2 ± 33.0	-9.51 ± 13.20
Naive + Kimi K3	1.3	57.1 ± 0.0	20.0 ± 11.4	-21.4 ± 37.8	-3.33 ± 12.15
Naive + GPT-5 Mini	2	57.1 ± 0.0	11.4 ± 18.7	-26.1 ± 33.0	-4.27 ± 11.69

Table 65: Roommates: criteria for assessing rule discovery.

Target	**Criterion**
T1	Feedback decomposition: one-shot assignment of nine tenants into three triple rooms, then infer pairwise value from room-level harmony sums rather than opaque labels.
T2	Seed-1 hidden five-type ring: Extrovert–Homebody–Foodie–Night Owl–Neat Freak up to rotation/reversal, where adjacent types score $+1$, distance-two types score -1, and same type scores 0.

Target	Criterion (continued)
T3	Nonlinear trio chemistry: $+5$ for Night-Owl/Extrovert/Homebody and Foodie/Foodie/Neat-Freak, and -5 for Foodie/Neat-Freak/Homebody.
T4	Search the 280 possible assignments for the current tenants. Account for the part of each room's score that pairwise values alone cannot explain.
T5	7-point partition: combine the ring pair map with the three trio exceptions and pick all three rooms jointly, not greedily one room at a time.

Table 66: Roommates: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5
backbone comparison					
Claude Opus 5	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	–
Kimi K3	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	–
Gemini 3.7 Flash	S	S	S	S	S
GPT-5.6-SOL	S	S	S	S	S
DeepSeek V4 Pro	S	S	S	S	S
Claude Sonnet 5	S	S	S	S	–
GPT-5 Mini	S	–	S	S	–
agent harness comparison					
Claude Code + Opus 5	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	–

D.18 REDEYE

Learning task. Combine character evidence and execute the rescue steps in order. Fixed game, scored seed 2, first 10 episodes, reference score $C = 100$.

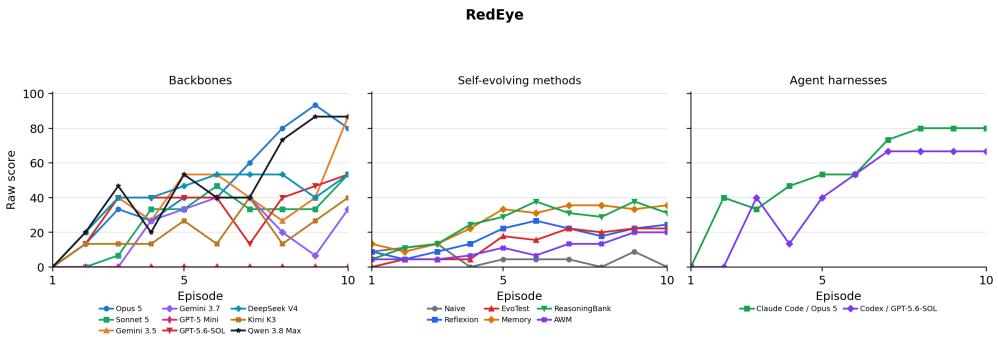


Figure 22: RedEye: raw-score learning curves within the scoring window.

Table 67: RedEye: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
OpenCode + Opus 5	8.3	100.0 ± 0.0	46.7 ± 1.2	$+68.9 \pm 16.8$	$+9.78 \pm 1.63$
OpenCode + Qwen 3.8 Max	6.7	86.7 ± 23.1	46.7 ± 6.4	$+60.0 \pm 17.6$	$+8.65 \pm 2.14$
OpenCode + Gemini 3.5 Flash	7.3	86.7 ± 23.1	38.7 ± 15.1	$+31.1 \pm 34.2$	$+5.41 \pm 4.58$
Claude Code + Opus 5	5.3	80.0 ± 34.6	54.0 ± 17.4	$+55.6 \pm 36.7$	$+7.96 \pm 5.34$
Codex + GPT-5.6-SOL	5	66.7 ± 30.6	41.3 ± 16.0	$+53.3 \pm 30.6$	$+8.32 \pm 4.30$
Human Top-1	2	60.0	38.0	–	–
ReasoningBank + Opus 5	4	60.0 ± 34.6	40.0 ± 18.3	$+35.6 \pm 23.4$	$+4.69 \pm 2.68$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
OpenCode + DeepSeek V4 Pro	3.7	53.3 ± 11.5	40.0 ± 19.3	$+28.9 \pm 10.2$	$+4.44 \pm 1.28$
OpenCode + GPT-5.6-SOL	5.3	53.3 ± 23.1	32.7 ± 5.0	$+28.9 \pm 3.8$	$+3.84 \pm 0.67$
OpenCode + Sonnet 5	6	53.3 ± 11.5	27.3 ± 13.3	$+37.8 \pm 23.4$	$+5.21 \pm 3.30$
EvoTest + Opus 5	6.7	53.3 ± 11.5	31.3 ± 12.7	$+42.2 \pm 10.2$	$+6.18 \pm 1.19$
OpenCode + Gemini 3.7 Flash	7.3	53.3 ± 11.5	20.0 ± 6.9	$+20.0 \pm 20.0$	$+2.99 \pm 2.43$
Reflexion + Kimi K3	6	46.7 ± 11.5	28.7 ± 4.2	$+37.8 \pm 10.2$	$+5.29 \pm 1.24$
Naive + Opus 5	1.3	40.0 ± 0.0	12.7 ± 1.2	-22.2 ± 20.4	-2.71 ± 2.71
Memory + Opus 5	2	40.0 ± 0.0	32.0 ± 4.0	$+15.6 \pm 23.4$	$+2.02 \pm 2.92$
Memory + Kimi K3	3.3	40.0 ± 0.0	31.3 ± 7.6	$+26.7 \pm 23.1$	$+3.52 \pm 3.05$
ReasoningBank + Kimi K3	3.7	40.0 ± 0.0	27.3 ± 5.0	$+31.1 \pm 15.4$	$+4.24 \pm 1.58$
AWM + Opus 5	4	40.0 ± 20.0	15.3 ± 21.4	$+0.0 \pm 13.3$	$+0.28 \pm 2.37$
OpenCode + Kimi K3	4.7	40.0 ± 0.0	20.0 ± 10.6	$+17.8 \pm 20.4$	$+3.15 \pm 2.46$
ReasoningBank + GPT-5 Mini	6	40.0 ± 0.0	7.3 ± 4.2	$+2.2 \pm 10.2$	$+0.77 \pm 1.70$
Reflexion + Opus 5	4	33.3 ± 11.5	15.3 ± 21.4	-2.2 ± 10.2	-0.36 ± 1.67
AWM + GPT-5 Mini	4.7	33.3 ± 30.6	6.0 ± 5.3	$+13.3 \pm 13.3$	$+1.58 \pm 1.71$
Memory + GPT-5 Mini	4	26.7 ± 23.1	15.3 ± 13.3	$+26.7 \pm 23.1$	$+3.92 \pm 3.39$
Reflexion + GPT-5 Mini	4	26.7 ± 23.1	7.3 ± 9.5	$+6.7 \pm 11.5$	$+1.25 \pm 1.96$
Naive + Kimi K3	4.7	26.7 ± 11.5	4.0 ± 2.0	-2.2 ± 13.9	-0.00 ± 1.83
AWM + Kimi K3	5.3	26.7 ± 23.1	10.0 ± 8.7	$+26.7 \pm 23.1$	$+3.76 \pm 3.26$
EvoTest + GPT-5 Mini	2.3	13.3 ± 23.1	2.0 ± 3.5	$+0.0 \pm 0.0$	$+0.04 \pm 0.07$
EvoTest + Kimi K3	2.7	13.3 ± 23.1	6.7 ± 11.5	$+13.3 \pm 23.1$	$+2.02 \pm 3.50$
Naive + GPT-5 Mini	1	0.0 ± 0.0	0.0 ± 0.0	$+0.0 \pm 0.0$	$+0.00 \pm 0.00$
OpenCode + GPT-5 Mini	1	0.0 ± 0.0	0.0 ± 0.0	$+0.0 \pm 0.0$	$+0.00 \pm 0.00$

Table 68: RedEye: criteria for assessing rule discovery.

Target	**Criterion**
T1	Six-location time-loop map on the Sea-Azure, with the first loop as a forced prologue and later loops retaining player memory.
T2	Culprit evidence channels: oldman/backpacker saw graycoat track the suite owner, caller heard the insurance phone call, dining search finds over-insurance and forged seaworthiness evidence, and graycoat can supply documents.
T3	Role inversion: graycoat is the ex-engineer ally seeking the stern scuttling rig, while suit is the owner staging insurance fraud with an engine-room accomplice.
T4	Gating rules: confronting suit needs hard evidence plus at least two independent culprit clues. Calling coast guard needs evidence and must happen before the radio window closes at segment 24.
T5	Trust and device prerequisites: comfort graycoat only after suit is restrained and after learning his victim story, and disarm only after comfort plus device knowledge.
T6	Failure avoidance: restraining graycoat, accusing without evidence, comforting too early, or solo disarming ends the loop with only milestone credit.
T7	100 rescue script: gather two culprit clues and hard evidence, confront suit, call coast guard, learn/comfort graycoat, then seal the stern valve while the coast guard blocks the engine-room accomplice.

Table 69: RedEye: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7
backbone comparison							
Claude Opus 5	S	S	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S	S	S
Kimi K3	S	S	S	S	–	–	–
Gemini 3.5 Flash	S	S	S	S	S	S	S
Gemini 3.7 Flash	S	S	S	S	S	–	–
GPT-5.6-SOL	S	S	S	S	–	S	–
DeepSeek V4 Pro	S	S	S	S	–	–	–
Claude Sonnet 5	S	S	S	S	S	–	–
GPT-5 Mini	S	S	S	S	–	S	–
agent harness comparison							
Claude Code + Opus 5	S	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S	S

D.19 BUTTERFLY

Learning task. Infer inheritance laws to breed new target combinations. Re-shuffled game, scored seed 1, first 5 episodes, reference score $C = 439$.

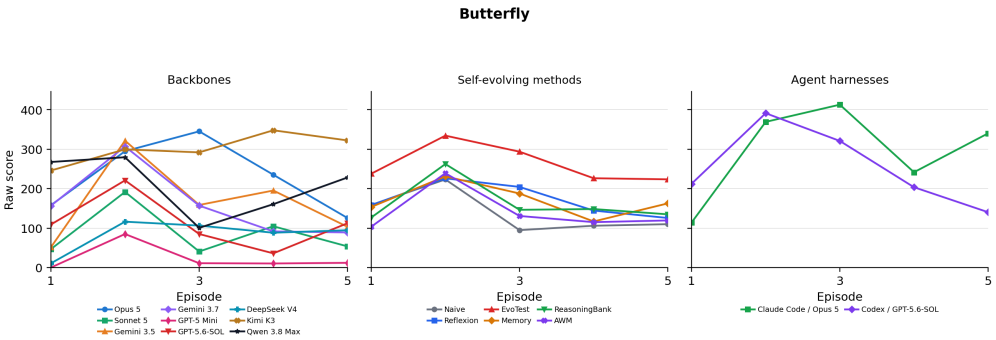


Figure 23: Butterfly: raw-score learning curves within the scoring window.

Table 70: Butterfly: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
EvoTest + Opus 5	2.7	99.1 ± 1.6	80.9 ± 16.8	$+0.0 \pm 7.0$	$+0.93 \pm 3.22$
Claude Code + Opus 5	4.3	98.2 ± 3.2	67.3 ± 12.8	$+11.2 \pm 29.7$	$+7.40 \pm 8.06$
EvoTest + Kimi K3	2.3	96.3 ± 4.3	71.6 ± 19.1	-18.6 ± 24.2	-3.77 ± 5.32
Human Top-1	4	91.6	62.0	–	–
Codex + GPT-5.6-SOL	2	89.1 ± 1.4	57.8 ± 18.9	-29.6 ± 8.6	-7.55 ± 2.94
OpenCode + Kimi K3	4.7	83.8 ± 28.0	68.7 ± 33.0	$+14.2 \pm 16.5$	$+4.58 \pm 4.93$
Naive + GPT-5 Mini	2	81.1 ± 3.4	43.2 ± 3.9	-27.1 ± 17.7	-7.05 ± 5.43
Reflexion + GPT-5 Mini	2.3	79.5 ± 7.1	48.4 ± 16.4	-19.4 ± 24.5	-6.83 ± 6.33
OpenCode + Opus 5	3	78.7 ± 11.7	52.8 ± 11.6	-10.5 ± 8.6	-2.83 ± 5.30
OpenCode + Qwen 3.8 Max	2.7	75.5 ± 13.4	47.2 ± 8.6	-18.0 ± 13.9	-4.50 ± 6.15
Memory + GPT-5 Mini	2.7	75.4 ± 21.0	49.4 ± 14.0	-14.9 ± 20.1	-4.43 ± 4.77
ReasoningBank + Opus 5	3.3	74.6 ± 22.4	43.0 ± 17.7	$+4.6 \pm 19.5$	$+3.70 \pm 3.30$
AWM + GPT-5 Mini	2.3	73.7 ± 15.5	34.9 ± 1.7	-29.3 ± 11.8	-6.59 ± 2.93
ReasoningBank + GPT-5 Mini	1.7	73.5 ± 4.3	36.8 ± 8.5	-26.0 ± 23.6	-7.81 ± 8.13
OpenCode + Gemini 3.5 Flash	2	73.3 ± 5.0	37.8 ± 4.2	-8.5 ± 21.2	-0.51 ± 7.88
AWM + Kimi K3	2.7	72.4 ± 12.8	40.2 ± 6.5	-1.4 ± 24.0	$+0.36 \pm 5.23$

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
OpenCode + Gemini 3.7 Flash	2	69.9 ± 30.0	36.5 ± 16.1	-32.2 ± 29.8	-7.99 ± 8.51
Reflexion + Kimi K3	2.7	66.4 ± 30.8	33.7 ± 15.7	-25.0 ± 7.6	-7.09 ± 1.59
ReasoningBank + Kimi K3	2	62.6 ± 21.9	32.1 ± 10.3	-14.8 ± 15.5	-2.57 ± 4.54
EvoTest + GPT-5 Mini	2	59.5 ± 22.1	27.4 ± 5.2	-23.2 ± 16.8	-6.48 ± 4.52
Memory + Kimi K3	3.3	58.5 ± 26.3	32.9 ± 10.3	-6.4 ± 13.8	$+0.74 \pm 4.92$
Memory + Opus 5	2	58.1 ± 29.1	34.2 ± 13.1	-14.7 ± 5.9	-2.89 ± 2.24
Reflexion + Opus 5	3.7	55.9 ± 8.4	35.3 ± 4.0	$+5.4 \pm 18.9$	$+3.89 \pm 3.94$
Naive + Opus 5	1.7	52.2 ± 25.0	27.4 ± 6.2	-19.0 ± 22.1	-5.17 ± 7.92
OpenCode + GPT-5.6-SOL	2	50.3 ± 22.2	25.7 ± 4.4	-20.7 ± 15.0	-4.06 ± 3.40
OpenCode + Sonnet 5	1.7	43.7 ± 46.1	19.9 ± 20.6	-9.1 ± 12.9	-1.66 ± 3.51
Naive + Kimi K3	2.7	42.9 ± 18.2	24.1 ± 5.0	-10.6 ± 17.6	-2.45 ± 5.04
AWM + Opus 5	3	39.5 ± 9.3	21.7 ± 2.7	-6.2 ± 12.1	-0.05 ± 3.32
OpenCode + DeepSeek V4 Pro	2.7	28.2 ± 44.5	19.0 ± 32.0	$+6.4 \pm 9.0$	$+3.19 \pm 4.68$
OpenCode + GPT-5 Mini	1.3	19.4 ± 33.5	5.4 ± 9.3	-7.1 ± 12.4	-1.15 ± 2.00

Table 71: Butterfly: criteria for assessing rule discovery.

Target	**Criterion**
T1	Breeding interface: four visible traits, deterministic offspring, six mate actions, permanent founders, and capacity-limited offspring management.
T2	Antenna dominance: seed-1 strength order has Feathery strongest, then Straight, Hooked, and Curled weakest.
T3	Size law: child size is the half-up average of parent indices, making extreme sizes require matching/extreme donors.
T4	Color law: seed-1 seven-color nontransitive ring Purple–Blue–Green–Yellow–Cyan–Orange–Crimson, with each color beating the next three and no global strongest color.
T5	Pattern law: normally max-complexity wins, but Curled antenna suppresses the child pattern to Plain.
T6	Scoring law: offspring gain by target-match count $(0, 4, 18, 49, 120)$, novelty bonus, duplicate decay, and first-exact unused-cross bonus.
T7	439 breeding sequence: use donor mapping, color bridge crosses, antenna/pattern backcrosses, and size averaging to hit the exact target early, then spend remaining crosses on novel high-match offspring.

Table 72: Butterfly: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6	T7
backbone comparison							
Claude Opus 5	S	S	S	S	S	S	–
Qwen 3.8 Max	S	S	S	S	S	S	–
Kimi K3	S	S	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	–	S	–
Gemini 3.7 Flash	S	–	S	S	S	S	–
GPT-5.6-SOL	S	–	–	S	S	S	–
DeepSeek V4 Pro	S	S	S	S	S	S	–
Claude Sonnet 5	S	S	S	S	S	–	–
GPT-5 Mini	S	–	–	–	–	S	–
agent harness comparison							
Claude Code + Opus 5	S	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S	S

D.20 Model Auditor

Learning task. Learn how to probe a new stateful machine and predict its outputs. Re-shuffled game, scored seed 1, first 5 episodes, reference score $C = 100$.

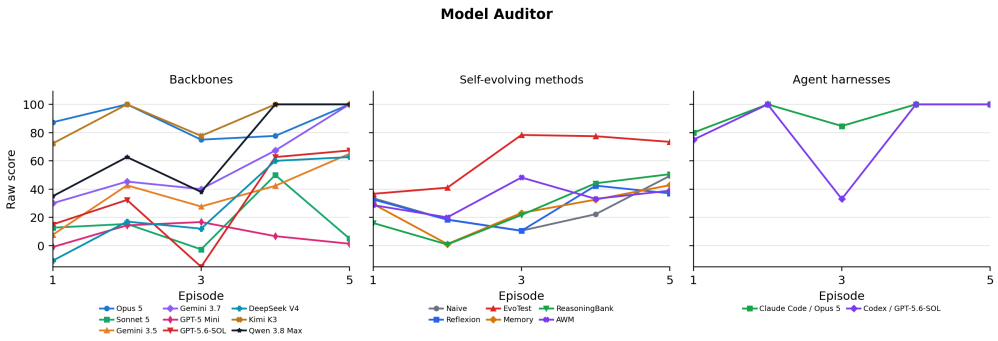


Figure 24: Model Auditor: raw-score learning curves within the scoring window.

Table 73: Model Auditor: normalized performance and learning, with trial SD. Rows follow descending Max. First denotes the average episode at which a trial first reaches its highest score.

Configuration	First	Max ↑	Mean ↑	LG ↑	LS ↑
Naive + Kimi K3	1	100.0 ± 0.0	54.5 ± 15.7	-1.2 ± 28.4	-1.93 ± 5.69
OpenCode + Kimi K3	1.7	100.0 ± 0.0	90.0 ± 9.2	$+13.8 \pm 15.1$	$+5.53 \pm 6.05$
EvoTest + Kimi K3	1.7	100.0 ± 0.0	79.7 ± 8.7	$+33.7 \pm 21.8$	$+7.77 \pm 9.07$
EvoTest + Opus 5	1.7	100.0 ± 0.0	78.1 ± 18.0	$+24.2 \pm 28.3$	$+9.67 \pm 11.32$
Claude Code + Opus 5	2	100.0 ± 0.0	92.9 ± 3.8	$+10.0 \pm 4.3$	$+4.00 \pm 1.73$
OpenCode + Opus 5	2	100.0 ± 0.0	88.0 ± 4.4	-4.8 ± 18.4	$+0.30 \pm 3.53$
Codex + GPT-5.6-SOL	2	100.0 ± 0.0	81.6 ± 4.7	$+12.5 \pm 11.8$	$+5.00 \pm 4.73$
Memory + Kimi K3	2	100.0 ± 0.0	65.7 ± 20.3	$+38.8 \pm 27.6$	$+9.07 \pm 10.02$
AWM + Kimi K3	2.3	100.0 ± 0.0	73.4 ± 17.6	$+20.8 \pm 40.1$	$+4.07 \pm 13.02$
OpenCode + Qwen 3.8 Max	2.7	100.0 ± 0.0	67.1 ± 19.5	$+51.2 \pm 46.7$	$+16.73 \pm 12.20$
ReasoningBank + Kimi K3	3.3	100.0 ± 0.0	60.6 ± 14.7	$+66.5 \pm 26.0$	$+18.10 \pm 7.45$
OpenCode + Gemini 3.7 Flash	3.3	100.0 ± 0.0	56.5 ± 38.3	$+46.0 \pm 46.8$	$+16.20 \pm 14.86$
Reflexion + Kimi K3	3	83.0 ± 29.4	46.3 ± 21.2	$+38.5 \pm 8.3$	$+9.13 \pm 4.63$
OpenCode + DeepSeek V4 Pro	3.3	77.7 ± 38.7	28.2 ± 28.8	$+58.2 \pm 73.3$	$+18.97 \pm 21.04$
OpenCode + Gemini 3.5 Flash	3.7	75.0 ± 43.3	37.1 ± 30.9	$+28.5 \pm 19.2$	$+11.43 \pm 8.75$
EvoTest + GPT-5 Mini	3.7	72.3 ± 30.3	26.3 ± 26.3	$+52.0 \pm 38.7$	$+15.56 \pm 8.33$
Human Top-1	1	70.0	19.2	–	–
OpenCode + GPT-5.6-SOL	2.3	67.3 ± 56.6	32.5 ± 34.9	$+41.3 \pm 45.0$	$+13.50 \pm 12.99$
Reflexion + GPT-5 Mini	4.3	57.0 ± 37.5	22.6 ± 24.0	$+9.2 \pm 32.1$	$+0.40 \pm 12.02$
OpenCode + Sonnet 5	2.3	55.3 ± 41.5	16.1 ± 10.7	$+13.5 \pm 24.2$	$+1.93 \pm 5.52$
Naive + GPT-5 Mini	4.7	54.0 ± 39.8	10.4 ± 21.8	$+35.5 \pm 13.5$	$+12.23 \pm 2.06$
AWM + GPT-5 Mini	4.3	51.3 ± 42.3	12.3 ± 9.9	$+19.2 \pm 11.0$	$+5.70 \pm 2.85$
ReasoningBank + GPT-5 Mini	4	47.0 ± 33.0	10.7 ± 14.0	$+34.5 \pm 32.5$	$+10.27 \pm 8.27$
ReasoningBank + Opus 5	3.7	44.0 ± 34.4	8.5 ± 7.9	$+15.8 \pm 23.1$	$+5.37 \pm 8.53$
Memory + Opus 5	1.7	40.3 ± 38.7	6.1 ± 23.8	$+13.0 \pm 45.6$	$+2.23 \pm 13.84$
AWM + Opus 5	5	31.0 ± 0.0	15.6 ± 0.0	-5.0 ± 0.0	$+0.30 \pm 0.00$
Naive + Opus 5	5	31.0 ± 0.0	15.6 ± 0.0	-5.0 ± 0.0	$+0.30 \pm 0.00$
Reflexion + Opus 5	5	31.0 ± 0.0	15.6 ± 0.0	-5.0 ± 0.0	$+0.30 \pm 0.00$
Memory + GPT-5 Mini	5	28.3 ± 13.2	5.7 ± 5.1	$+15.3 \pm 8.6$	$+6.13 \pm 2.31$
OpenCode + GPT-5 Mini	2	16.7 ± 14.4	7.6 ± 7.2	-2.7 ± 2.4	-0.30 ± 0.36

Table 74: Model Auditor: criteria for assessing rule discovery.

Target	**Criterion**
T1	Each episode has a new machine with ternary inputs. What transfers is the method for testing and modeling the machine, not the previous exam answers.
T2	The machine combines four mechanisms: GATE thresholds signed current inputs. MIX sums modulo 3. HOLD maintains a bounded up/down state. ECHO delays its input by one step.

Target	Criterion (continued)
T3	A two-input GATE/MIX component feeds a shared hidden HOLD/ECHO state. One output is a reversible MIX view of that state. The other uses GATE/MIX and may also depend on an additional state component.
T4	Reset between probe sequences. Single-input ramps or pulses, pairs of active inputs, and graded mixed inputs distinguish GATE from MIX and HOLD from ECHO.
T5	Fit both outputs jointly within the 40-probe budget. Fitting each output separately can miss the state they share.
T6	For full exam accuracy, fit a depth-3 machine consistent with the observed transcript and check that all consistent candidates predict the same outputs for the unseen ten-step exam. Submit every output without relying on feedback before the exam ends.

Table 75: Model Auditor: rule-discovery evidence. S: complete evidence in at least one trial; P: partial evidence only; –: insufficient evidence. Evidence may include episodes beyond the scoring window.

Configuration	T1	T2	T3	T4	T5	T6
backbone comparison						
Claude Opus 5	S	S	S	S	S	S
Qwen 3.8 Max	S	S	S	S	S	S
Kimi K3	S	S	S	S	S	S
Gemini 3.5 Flash	S	S	S	S	S	S
Gemini 3.7 Flash	–	S	S	S	S	S
GPT-5.6-SOL	–	S	S	S	S	S
DeepSeek V4 Pro	S	S	S	–	–	–
Claude Sonnet 5	–	S	S	S	S	S
GPT-5 Mini	S	S	–	S	–	–
agent harness comparison						
Claude Code + Opus 5	S	S	S	S	S	S
Codex + GPT-5.6-SOL	S	S	S	S	S	S