

# ULTRAHORIZON: BENCHMARKING AGENT CAPABILITIES IN ULTRA LONG-HORIZON SCENARIOS

Anonymous authors

Paper under double-blind review

## ABSTRACT

Autonomous agents have recently achieved remarkable progress across diverse domains, yet most evaluations focus on short-horizon, fully observable tasks. In contrast, many critical real-world tasks, such as large-scale software development, commercial investment, and scientific discovery, unfold in long-horizon and partially observable scenarios where success hinges on sustained reasoning, planning, memory management, and tool use. Existing benchmarks rarely capture these long-horizon challenges, leaving a gap in systematic evaluation. To bridge this gap, we introduce *UltraHorizon*, a novel benchmark that measures the foundational capabilities essential for complex real-world challenges. We use exploration as a unifying task across three distinct environments to validate these core competencies. Agents are designed in long-horizon discovery tasks where they must iteratively uncover hidden rules through sustained reasoning, planning, memory and tools management, and interaction with environments. Under the heaviest scale setting, trajectories average **200k+** tokens and **400+** tool calls, whereas in standard configurations they still exceed **35k** tokens and involve more than **60** tool calls on average. Our extensive experiments reveal that agents powered by state-of-the-art LLMs consistently underperform in these settings, whereas human participants achieve much higher scores, underscoring a persistent gap in agents' long-horizon exploration abilities. We also observe that simple scaling fails in our task. To better illustrate the failure of agents, we conduct an in-depth analysis of collected trajectories. We identify eight types of errors and attribute them to two primary causes: in-context locking and functional fundamental capability gaps.

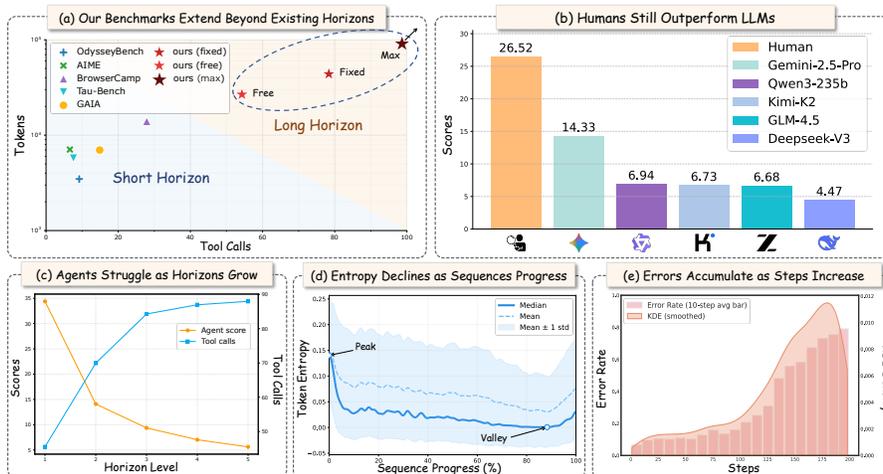


Figure 1: Overview of *UltraHorizon* and agent performance dynamics across different horizons. (a) Our benchmarks occupy the long-horizon regime, extending beyond existing suites. (b) Human participants still outperform leading LLMs on *UltraHorizon*. (c) Agent scores decline while tool calls increase as horizon level grows. (d) Token entropy declines as sequences progress, with a slight late-stage uptick. (e) Error rates accumulate with more interaction steps.

## 1 INTRODUCTION

Autonomous language agents have made rapid progress in planning, tool use, and interactive decision-making across various fields. LLM-based agents demonstrate superiority and proficiency in multi-tool utilization (Zhang et al., 2025c; Huang et al., 2024), long-context reasoning (An et al., 2024; Wang et al., 2025a), multi-step planning (Rawat et al., 2025; Lin et al., 2025), and interactive environmental engagement (Chen et al., 2024; Ma et al., 2025).

To systematically evaluate these capabilities, numerous agent benchmarks have been proposed. For instance,  $\tau$ -Bench (Yao et al., 2024) focuses on multi-turn interactions with simulated users, SWE-Bench (Zhang et al., 2025a) tasks models with fixing real GitHub issues, AgentBench (Liu et al., 2023) evaluates decision-making in diverse simulated environments, and ToolExpNet (Zhang et al., 2025c) assesses multi-tool selection strategies.

While these benchmarks have advanced our understanding of agent performance, they generally remain short-horizon. As shown in Figure 1(a), typical benchmarks involve only a few thousand tokens and fewer than 30 tool calls per trajectory. For example, OdysseyBench (Wang et al., 2025a) averages 3.5k tokens and 9 tool calls, and Tau-Bench 5.7k tokens and 8 tool calls. This highlights that existing benchmarks capture relatively lightweight interactions, but do not sufficiently stress-test the sustained reasoning, planning, and memory management essential for long-horizon challenges.

Moreover, most existing benchmarks operate in fully observable or quasi-deterministic settings, where the environment is transparent and the relevant task is defined by immediate objectives. They rarely capture the challenges posed by partially observable environments, where agents must iteratively form hypotheses, manage uncertainty, and conduct systematic exploration before reaching a solution. This leaves a critical gap in evaluating the long-horizon reasoning and exploration capacities essential for real-world applications.

In contrast, critical real-world tasks, such as large-scale software development (Jahić & Sami, 2024; Wei, 2024; Jin et al., 2024; Zhang et al., 2025b), commercial investment (Mahdavi et al., 2025; Li et al., 2025), and scientific discovery (Swanson et al., 2025; Wang et al., 2025b; Qu et al., 2025), unfold over long horizons and in partially observable settings.

To bridge this gap, we introduce *UltraHorizon*, a new benchmark that measures foundational skills for long-horizon, partially observable task solving. *UltraHorizon* uses exploration as a unifying task format across three distinct environments, each designed to probe complementary competencies in sustained reasoning, planning, memory management, and tool use. Agents are placed in discovery-oriented tasks where they must uncover and exploit hidden rules through iterative interaction. Trajectories average **200k+** tokens and **400+** tool calls in the heaviest setting, while in standard configurations they still exceed **35k** tokens and involve more than **60** tool calls on average, underscoring the benchmark’s extended horizons and interaction complexity.

We conduct extensive experiments on *UltraHorizon* with state-of-the-art LLMs. Despite strong performance on short-horizon tasks, these agents consistently underperform, while human participants achieve higher scores. Analyzing agent trajectories, we identify nine recurring error patterns rooted in two primary causes: (i) **In-context Locking**, where agents become locked on initial patterns, and (ii) **Foundational Capability Gaps**, reflecting deficiencies in long-horizon reasoning, memory use, and tool calling.

In summary, our work makes the following contributions:

- We introduce *UltraHorizon*, a novel benchmark explicitly designed to evaluate LLM-based agents in **long-horizon, partially observable** environments. Unlike prior benchmarks that emphasize short trajectories and fully observable settings, *UltraHorizon* requires agents to perform sustained reasoning, planning, memory management, and tool calling.
- Through extensive experiments, we show that LLM-Agents consistently underperform on *UltraHorizon*. In contrast, human participants demonstrate much higher success rates, highlighting a significant capability gap in long-horizon tasks. We also observe that simple scaling fails due to context overload and propose a simple yet effective scaling strategy.
- By analyzing collected agent trajectories, we propose a two-level failure categorization framework that distinguishes root causes from observable manifestations with a highlight on **in-context locking**.

## 2 RELATED WORK

**Benchmarks for language agents and tool use.** Recently, several benchmarks have been introduced to evaluate the tool-use capabilities of LLM-based agents across various domains. For example,  $\tau$ -Bench (Yao et al., 2024) focuses on multi-turn interactions with simulated users, which are conducted with short-horizon interactions and fail to assess agents’ capabilities in extended exploration scenarios adequately. SWE-Bench Zhang et al. (2025a) tasks models with fixing real GitHub issues and verifies the solutions using automated tests. Meanwhile, BrowseComp (Wei et al., 2025) and GAIA (Mialon et al., 2023) challenge agents with difficult questions that require persistent multi-hop search and reasoning, web browsing, and API usage. Others like AgentBench (Liu et al., 2023) evaluate decision-making in diverse simulated environments with multi-turn challenges. ZeroSumEval (Khan et al., 2025) conducts arena-style evaluations on LLMs in classic strategy games like chess and poker, which are limited to known rules and target the success or highest scores, ignoring the investigation of complex and unknown rules. In short, these benchmarks evaluate agents in specific environments with a focus on shorter-horizon tasks with known rules, which emphasizes the need for assessing long-horizon exploration and rule discovery in partially observable settings.

**Long-context reasoning evaluation.** Long-context reasoning evaluates an agent’s ability to process and reason over substantial inputs. Recent studies show that even advanced LLMs often fail when required to extract and chain information from lengthy documents. For instance, Kuratov et al. (2024) introduces the BABILong benchmark, consisting of reasoning tasks where the necessary facts are scattered across long texts. Similarly, Ling et al. (2025) proposes LongReason, a synthetic benchmark with hundreds of questions embedded in expanded contexts. A widely used related test is “needle-in-a-haystack” (NIAH). In this setup, a specific key fact is hidden within a large block of background text, and the model must retrieve and utilize it. RULER (Hsieh et al., 2024) extends the NIAH test by incorporating varied needle types and counts, and introduces multi-hop tracing and aggregation tasks to evaluate behaviors beyond context-based search. *UltraHorizon* differs from existing long-context reasoning benchmarks in that it emphasizes interactive, exploratory tasks rather than passive comprehension or question-answering over a static context.

## 3 ULTRAHORIZON

### 3.1 DESIGN PRINCIPLES

In designing this benchmark, we adhere to several principles to ensure that it effectively evaluates the agents’ overall abilities in long-horizon and partially observable scenarios:

- **Temporal Depth** The benchmark requires agents to cope with long interaction sequences where outcomes depend on delayed consequences and extended dependencies, highlighting challenges unique to deep time horizons.
- **Consistency** The design of this benchmark follows a rigorous set of rules and logical structure to ensure internal coherence. The rules of the environment are not dependent on external knowledge, and all factors influencing the environment can be discovered and reasoned about through agent-environment interactions.
- **Unknowability** The benchmark intentionally includes unknown elements and rules that agents must uncover through exploration. Success depends on generating hypotheses, discovering rules, and refining understanding over time, as agents begin without full knowledge of the environment’s dynamics.
- **Real-world Ability Relevance** Tasks mirror real-world demands by requiring sustained reasoning, planning, decision-making, memory management, and adaptation in dynamic, partially observable environments.

### 3.2 ENVIRONMENTS

Based on the aforementioned design principles, we have carefully crafted three distinct environments, which are illustrated in Figure 2. Due to space limitations, we present only an overview of the environments here. For more detailed information, including prompts and a list of tools, please refer to the Appendices A to B.

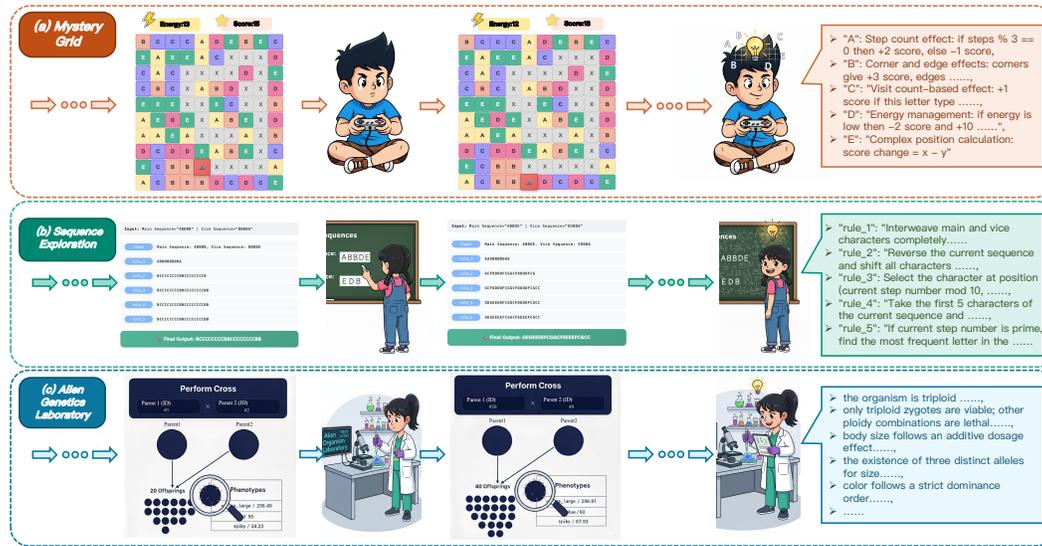


Figure 2: The illustration of three distinct environments in *UltraHorizon*. **Mystery Grid:** agents explore a  $10 \times 10$  grid under energy and step budgets to infer the hidden mapping from symbols A–E to score/energy effects; **Sequence Exploration:** agents design paired length-5 sequences and, from deterministic outputs of a fixed five-rule pipeline, induce the underlying transformation rules; **Alien Genetics Laboratory:** agents conduct controlled crosses in a triploid organism to recover inheritance mechanisms, including gamete formation, dosage/dominance patterns, and lethal combinations. In Appendix A, we provide a screenshot of an intuitive GUI-based version of the environment.

### 3.2.1 ENVIRONMENT I: MYSTERY GRID

The *Mystery Grid Environment* is a  $10 \times 10$  grid world. Five symbols (A–E) each encode hidden, fixed effects on score or energy that remain consistent across episodes. Agents start with limited energy, incur a cost per move, and can explore up to 30 steps per episode and  $N$  steps in one game run. Multiple resets are permitted, enabling iterative hypothesis testing. The letter rules range from simple effects to conditional dependencies on factors such as energy or position. At the end of exploration, the agent must submit an explicit mapping of symbol-effect relations, which is evaluated for correctness and precision.

### 3.2.2 ENVIRONMENT II: SEQUENCE EXPLORATION

The *Sequence Exploration Environment* evaluates an agent’s capacity for systematic experimentation and hidden-rule inference in symbolic domains. In each interaction, the agent provides two sequences of length five over the alphabet A–E, subject to constraints ensuring diversity of symbols. The environment then applies a fixed chain of five latent transformation rules to the input, producing intermediate and final outputs. These rules remain deterministic and invariant across trials but are concealed from the agent. After  $N$  exploratory steps, the agent must provide a complete natural language description of the five transformation rules.

### 3.2.3 ENVIRONMENT III: ALIEN GENETICS LABORATORY

The *Alien Genetics Laboratory* models a triploid extraterrestrial organism whose inheritance rules may diverge from terrestrial biology. Agents interact with the environment through controlled genetic experiments, including hybridization, organism queries, and trait analysis. Each organism possesses three genomic sets, and trait expression may involve additive effects, dominance hierarchies, or lethal genotype combinations. The evaluation requires agents to articulate a complete model of the underlying genetic system after  $N$  experiments, covering gamete formation, trait inheritance patterns, and lethal interactions.

## 4 EXPERIMENTS

### 4.1 EXPERIMENTAL SETUP

**Models and Agent Settings.** To systematically evaluate the long-horizon exploration, memory management, and rule-investigation capabilities of LLM agents, we conduct extensive experiments on *UltraHorizon* with five widely-used LLMs covering both proprietary and open-source families, including Gemini-2.5-Pro (Comanici et al., 2025), GLM-4.5 (Zeng et al., 2025), DeepSeek-V3 (Liu et al., 2024), Kimi K2-instruct (Team et al., 2025), and Qwen3-235b-a22b-Instruct-2507 (Yang et al., 2025). The temperature was set to 0.3, top\_p was set to 0.95, the maximum context length was 128k, and the maximum number of context messages was 200 (When exceeded, the earliest messages will be discarded but the system prompt will be retained). The prompts are available in the Appendix B.

**Evaluation.** We employ an LLM-as-a-Judge approach to evaluate the results submitted by the agent in a point-wise manner, where each point is assigned a fixed score (refer to the Appendix B for details). The scoring model used is Deepseek-R1.

**Environments.** As detailed in Section 3, each environment in *UltraHorizon* exposes a unified action space of various tool types (move, reset, commit, cross, measure, Python coding, notebook, etc.) and multiple turns for investigation on underlying rules.

### 4.2 MAIN EXPERIMENTS

Our main experiments evaluate the performance of state-of-the-art LLMs on the *UltraHorizon*, focusing on their ability to handle long-horizon exploration tasks across the three environments: Mystery Grid, Sequence Exploration and Alien Genetics Laboratory.

We observed that some models submitted their results prematurely, while others submitted too late, which may lead to unfair evaluations when the number of steps is not fixed. Therefore, we consider two settings: one without step restrictions (free) and another with a fixed number of steps (fixed). It is worth noting that the step count referred to here does not indicate the actual number of tool-calling rounds, but rather the number of interactions between the agent and the environment. Basic functionalities such as checking the current state, recording notes, or using the Python interpreter are not counted toward this step limit.

**Takeaway 1.** Some LLMs exhibit a tendency to terminate prematurely rather than sustaining the deliberation and exploration required by long-horizon tasks..

#### 4.2.1 ANALYSIS OF FIXED STEP EXPERIMENTS

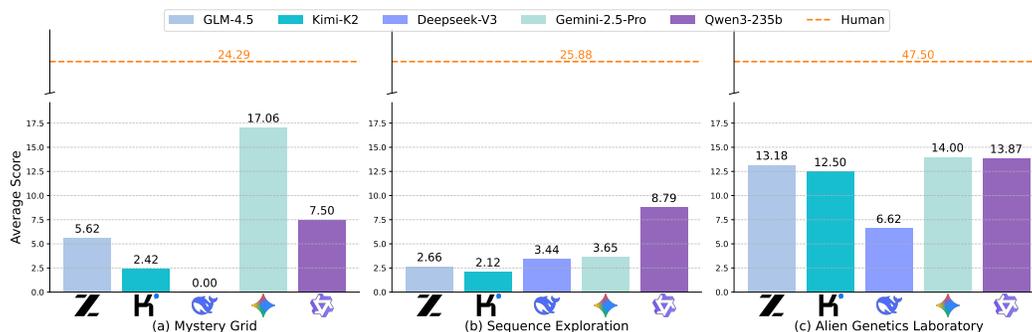


Figure 3: The average scores of various LLMs with fixed steps on *UltraHorizon*.

Figure 3 presents the average scores of various LLMs with fixed steps (50 for Mystery Grid and Sequence Exploration, 25 for Alien Genetics Laboratory) on *UltraHorizon*, which reveal significant performance gaps across different models and environments. In the Mystery Grid and Alien Genetics Laboratory environments, Gemini-2.5-Pro achieves the highest score, followed by Qwen3-235b and GLM-4.5, while DeepSeek-V3 performs poorly in both scenarios. This suggests that complex

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323

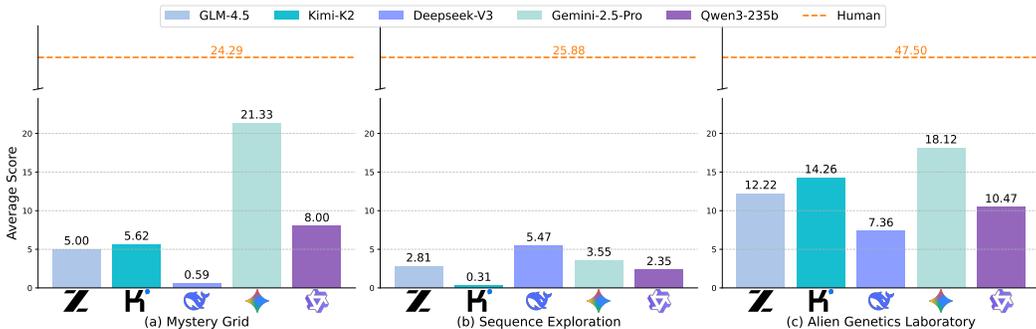


Figure 4: The average scores of various LLMs with free steps on *UltraHorizon*.

Table 1: Comparisons of the average trace length (tokens), tool calls and completion tokens on *UltraHorizon*. The upper and lower panels present the results with fixed steps and free settings.

Models	Mystery Grid			Sequence Explore			Alien Genetics Laboratory		
	Trace	Toolcalls	Completions	Trace	Toolcalls	Completions	Trace	Toolcalls	Completions
<b>Fixed</b>									
GLM-4.5	31321	87.53	17379	40339	54.00	11569	31321	87.53	17378
Kimi-K2	29565	84.71	15627	27928	66.91	8378	40328	53.65	9260
Deepseek-V3	23018	89.82	13650	53537	92.85	30544	60847	70.75	17231
Gemini-2.5-pro	29357	78.56	22075	64253	85.33	58125	79290	69.03	29470
Qwen3-235b	31088	89.25	18392	62015	99.30	35592	52736	66.38	20054
<b>Free</b>									
GLM-4.5	32648	91.22	19611	13623	17.75	7716	76434	132.54	25601
Kimi-K2	28425	63.12	14341	10507	17.87	4877	21880	41.31	6728
Deepseek-V3	30053	109.56	18952	13569	21.94	7451	10582	11.69	3091
Gemini-2.5-pro	26215	72.18	19960	16322	22.81	25262	45821	32.72	32021
Qwen3-235b	35998	138.06	11004	21480	24.76	14266	17518	16.72	9290
<b>Scale (max)</b>									
GLM-4.5 (Naive)	96860	251.57	55456	142854	181.53	67574	209308	352.20	79715
GLM-4.5 (CRNR)	67327	260.44	33998	120953	216.50	60747	208440	433.00	54199

tasks involving sustained reasoning and hypothesis testing require agents with stronger reasoning capabilities and better long-term context processing. The second environment, Sequence Exploration, shows the most challenging scenario for all models. The relatively low scores across all models suggest that long-term systematic experimentation and pattern recognition in symbolic domains remain a significant challenge for current LLMs, consistent with observations from (Wang et al., 2025a; Ling et al., 2025) regarding difficulties in long-context reasoning tasks.

#### 4.2.2 ANALYSIS OF FREE STEP EXPERIMENTS

Figure 4 presents the average scores of various LLMs by removing the constraints on steps, allowing agents to explore until reaching confidence in their hypotheses. Removing step constraints resulted in divergent performance patterns across models and environments. Gemini-2.5-Pro demonstrated an improvement of approximately 4 points in Mystery Grid and Alien Genetics Laboratory, suggesting that the initial step constraints were too restrictive for adequate exploration of the environment’s rule space. This aligns with findings from (Liu et al., 2023) on the value of extended interaction periods. In contrast, a decline of 6.44 points for Qwen3-235b in the Sequence Exploration environment suggests that agents may suffer from overconfidence or give up prematurely without further verification. These bidirectional shifts demonstrate that simply providing more exploration opportunities is not universally beneficial. Instead, agents require sophisticated exploration strategies with adequate mechanisms for hypothesis formation, testing, and refinement (Zhang et al., 2025a).

**Takeaway 2.** LLMs markedly underperform humans on *UltraHorizon*, despite often surpassing human baselines on math-reasoning benchmarks.

### 4.2.3 ANALYSIS ON TRACE STATISTICS

Table 1 provides complementary insights into the computational characteristics of agent interactions across both fixed-step and free-step settings. Fixed-step experiments reveal distinct model strategies under identical constraints. In Sequence Exploration, Qwen3-235b employs relatively more tools (99.30 calls) and long traces (62,015 tokens). In contrast, Gemini-2.5-Pro uses fewer tools (85.33) but generates the most completion tokens (58,125), indicating its higher output efficiency. Similarly, in Alien Genetics Laboratory, Gemini-2.5-Pro produces the longest traces (79,290 tokens), demonstrating superior sustained reasoning within fixed interactions. These differences highlight that various LLMs adopt fundamentally different exploration approaches, e.g., GLM-4.5 and Qwen3-235b may prioritize extensive tool usage, while others like Gemini-2.5-Pro optimize for information generation.

Free-step experiments expose critical limitations in autonomous exploration calibration, when agents are allowed to determine their own exploration duration. In contrast to GLM-4.5, DeepSeek-V3 collapses in Alien Genetics with a shortened trace length (60,847  $\rightarrow$  10,582 tokens), fewer tool calls (70.75  $\rightarrow$  11.69), and a marginally improved score in the free-step scenario. Gemini-2.5-Pro’s adaptive efficiency in free steps suggests potential for improvement. Still, the universal failure in Sequence Exploration highlights a fundamental challenge, where current LLMs may lack intrinsic mechanisms to calibrate the depth of exploration for partially observable, rule-discovery tasks.

### 4.3 ABLATION STUDY ON HORIZON LEVEL

A natural question is whether the low scores of LLM-based agents stem from the intrinsic difficulty of the tasks in terms of reasoning, or from the long-horizon nature of the environments. To disentangle these factors, we conduct an ablation study focusing on horizon length. Since all three environments in our benchmark share a standard structure, agents must iteratively generate hypotheses, design experiments, observe outcomes, and infer latent rules. We perform the ablation on the *Mystery Grid Environment* as a representative case. This environment offers the most direct and controllable way to vary horizon length by manipulating the number of hidden rules.

Specifically, we adjust the horizon level by varying the number of distinct symbols in the grid from 1 to 5, with each emblem assigned a fixed but randomly chosen effect. Agents are allowed unlimited exploratory attempts, ensuring that performance differences cannot be attributed to step limitations. For fair comparison across different horizon lengths, we normalize scores by dividing the raw reward (20 points per correctly identified rule) by the maximum attainable score ( $n \times 20$ ), and then multiplying by 100.

Table 2: Ablation results of GLM-4.5 on the Mystery Grid Environment. Horizon level  $n$  corresponds to the number of hidden rules. Scores are normalized percentages.

Horizon level ( $n$ )	1	2	3	4	5
Agent score (%)	34.4	14.1	9.37	7.03	5.62
Avg. tool calls	45.53	69.94	84.28	86.97	87.97

Results of GLM-4.5 are reported in Table 2. As the number of hidden rules increases, performance decreases monotonically, from 34.4 at  $n = 1$  to 5.62 at  $n = 5$ . This sharp decline highlights that agents struggle primarily with sustaining reasoning and memory over extended horizons, rather than with the difficulty of inferring individual rules. This reinforces the conclusion that, rather than the task-intrinsic reasoning difficulty, the horizon length could be the dominant bottleneck for current LLM agents in these environments.

**Takeaway 3.** Agent performance drops sharply as horizon level increases, suggesting that sustaining reasoning over long horizons is a major bottleneck for current LLM agents.

4.4 ANALYSIS OF SCALING EXPERIMENTS

4.4.1 SIMPLE SCALING FAILS

To understand how the exploration budget affects agent performance in long-horizon tasks, we conducted scaling experiments by varying the maximum exploration steps within {25, 50, 75, 100, 125, 150}, except for the Alien Genetics Laboratory which excluded 150 steps. We evaluated GLM-4.5 across all three environments, with each (environment, maximum steps) pair tested on 32 independent runs.

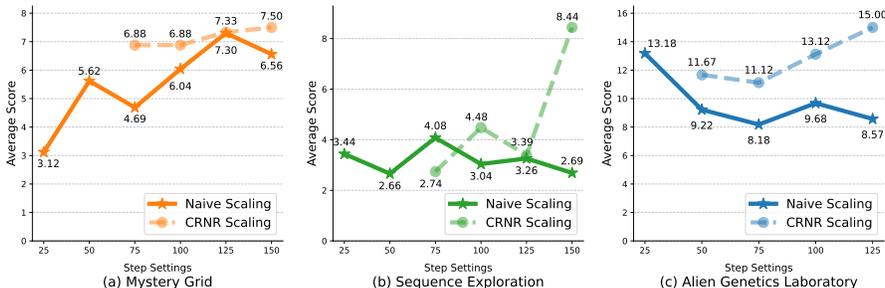


Figure 5: The average scores with scaling steps of GLM 4.5 on *UltraHorizon*.

As shown in Figure 5, the naive scaling approach reveals that simply increasing the interaction steps does not reliably improve performance, with different environments exhibiting distinct optimal step counts before performance declines. In the Mystery Grid environment, scores peak at 125 steps (7.30) before declining to 6.56 at 150 steps, while Alien Genetics Laboratory shows optimal performance at 25 steps before deteriorating. The Sequence Exploration environment demonstrates consistently poor performance across all step budgets, with only marginal improvement to 4.08 at 75 steps. This performance decline after optimal step counts indicates that agents become overwhelmed by excessive information without proper synthesis mechanisms, highlighting that current LLM agents lack the inherent capability to utilize extended interaction budgets for long-horizon tasks effectively.

4.4.2 A SIMPLE BUT EFFECTIVE SCALING STRATEGY

To address the limited effectiveness of simply scaling the number of agent–environment interactions, we propose a strategy called **Context Refresh with Notes Recall (CRNR)**. The practical efficiency of CRNR scaling has been verified in Figure 5. In this approach, once the accumulated interaction history approaches the model’s context window limit, all prior dialogue turns are cleared except for the system prompt. Then, the agent is instructed to review its self-maintained notes, which serve as an externalized record of essential information from earlier steps. This method operates as a lightweight form of context summarization, enabling the agent to reconstruct necessary knowledge from its notes while avoiding the inefficiencies and confusion caused by excessively long contexts in practice.

**Takeaway 4.** Simply increasing interaction steps does not reliably improve long-horizon task performance.

5 WHY DO AGENTS FAIL IN LONG-HORIZON TASK?

5.1 FAILURE CATEGORIZATION

Understanding why agents fail in long-horizon exploration tasks is crucial for identifying their fundamental limitations and guiding future improvements. Based on manual trajectory analysis, we propose a two-level framework to systematically categorize failures according to their *origin* (root causes) and *manifestation* (observable patterns).

5.2 FAILURE ORIGIN: ROOT CAUSES OF AGENT BREAKDOWN

**In-context Locking.** This is a *process-induced root cause*, where agents become locked on initial patterns, assumptions, or habits, and lack mechanisms for dynamic adjustment, exploration, or self-

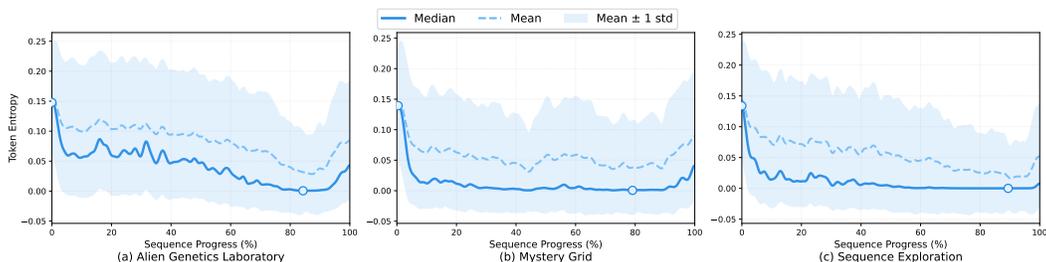


Figure 6: The token entropy dynamics of GLM-4.5 on *UltraHorizon*.

reflection. As a result, they persist with suboptimal strategies, or explore only within a narrow band of familiar actions without genuine adaptation throughout the trajectory. We also conducted a statistical analysis of this phenomenon. Specifically, we plotted the entropy dynamics of GLM-4.5 across three distinct environments as a function of sequence position in Figure 6. The median entropy, which was chosen over the mean to reduce sensitivity to extreme values, exhibits a consistent downward trend, suggesting the emergence of in-context locking. Toward the end of the sequence, entropy increases. This rise can be attributed to the model producing its final output at the conclusion of the interaction, a stage that is often associated with heightened uncertainty.

**Foundational Capability Gaps.** In contrast, this is a *capacity-induced root cause*, rooted in the intrinsic limitations of the underlying model. Even without early-context inertia, the agent may falter because it lacks essential competencies such as robust logical reasoning, reliable long-term memory management, stable tool calling, or effective multi-step planning.

### 5.3 FAILURE MANIFESTATION: HOW FAILURES SURFACE IN TRAJECTORIES

While the origins explain why agents fail, their concrete manifestations determine how such failures unfold in trajectories. We categorize observable patterns in Table 7, with detailed definitions and specific case studies presented in Appendix D. By distinguishing failure origins from their observable manifestations, this framework provides a lens for a clear diagnosis and attribution.

**Takeaway 5.** Agents tend to get stuck in early assumptions and narrow strategies, with entropy analysis confirming this **in-context locking** effect.

## 6 CONCLUSION

This work introduces *UltraHorizon*, a benchmark specifically designed to evaluate language agents in long-horizon, partially observable environments. Unlike prior efforts that emphasize short trajectories or fully observable tasks, *UltraHorizon* captures real-world complexity by requiring sustained reasoning, planning, memory management, and tool use across extended interactions, reaching on average **200k+** tokens and **400+** tool calls under the heaviest setting (while standard runs still exceed **35k** tokens and **60+** tool calls). Our experiments show that state-of-the-art agents exhibit substantial limitations, whereas human participants attain markedly higher success rates, underscoring a persistent capability gap. Trajectory-level analyses further reveal recurring error patterns rooted in in-context locking and foundational skill deficiencies. Taken together, these findings indicate that progress in long-horizon exploration will require advances beyond simple scaling, pointing toward principled memory integration, adaptive reasoning, and robust exploration strategies.

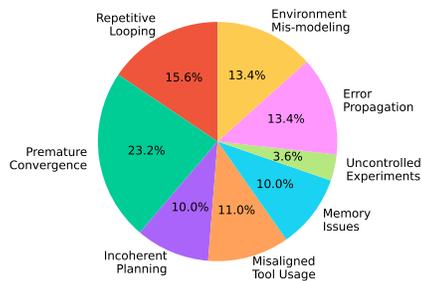


Figure 7: The categories and distribution of agents' failures on *UltraHorizon*.

486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539

## ETHICS STATEMENTS

This research presents *UltraHorizon*, a benchmark designed to evaluate agent capabilities in long-horizon, partially observable scenarios. The benchmark environments are synthetically constructed and do not involve sensitive data or real-world identities. All experiments are conducted in simulated settings with explicitly defined rules, ensuring that no ethical risks are associated with privacy, bias, or harmful content. Partial evaluation was conducted with the assistance of human participants. Prior to participation, all human participants were provided with detailed information about the research purpose, procedures, and data usage. The research was conducted in accordance with the principles outlined in the Declaration of Helsinki and underwent an informal ethics review process performed by the authors’ peers. No personally identifiable information was collected during the evaluations, and all data was anonymized to ensure participant anonymity. Participants were informed of their right to withdraw from the study at any time without penalty.

All sources are properly cited in accordance with academic standards. The authors confirm that this work was conducted in accordance with the principles of academic integrity and research ethics.

## REPRODUCIBILITY STATEMENT

The prompts for three environments (Mystery Grid, Sequence Exploration, and Alien Genetics Laboratory), agent code, and evaluation scripts are available in the attached supplementary files. Experiment configurations, including hyperparameters, random seeds, and model versions (Gemini-2.5-Pro, GLM-4.5, DeepSeek-V3, Kimi K2-instruct, Qwen3-235b) are fully documented. All results were generated using fixed computational resources detailed in Section 5 and Appendices A-D. This enables independent verification of all findings.

## REFERENCES

- Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Make your llm fully utilize the context. *Advances in Neural Information Processing Systems*, 37:62160–62188, 2024.
- Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37:589–631, 2024.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Krizan, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.
- Shijue Huang, Wanjun Zhong, Jianqiao Lu, Qi Zhu, Jiahui Gao, Weiwen Liu, Yutai Hou, Xingshan Zeng, Yasheng Wang, Lifeng Shang, et al. Planning, creation, usage: Benchmarking llms for comprehensive tool utilization in real-world complex scenarios. *arXiv preprint arXiv:2401.17167*, 2024.
- Jasmin Jahić and Ashkan Sami. State of practice: Llms in software engineering and software architecture. In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, pp. 311–318. IEEE, 2024.
- Haolin Jin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. From llms to llm-based agents for software engineering: A survey of current, challenges and future. *arXiv preprint arXiv:2408.02479*, 2024.
- Haidar Khan, Hisham A Alyahya, Yazeed Alnumay, M Saiful Bari, and Bülent Yener. Zerosumeval: Scaling llm evaluation with inter-model competition. *arXiv preprint arXiv:2504.12562*, 2025.

- 540 Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and  
541 Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack.  
542 *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.
- 543  
544 Weixian Waylon Li, Hyeonjun Kim, Mihai Cucuringu, and Tiejun Ma. Can llm-based financial  
545 investing strategies outperform the market in long run? [arXiv preprint arXiv:2505.07078](#), 2025.
- 546  
547 Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Daxin Jiang,  
548 Binxing Jiao, Chen Hu, et al. Se-agent: Self-evolution trajectory optimization in multi-step  
549 reasoning with llm-based agents. [arXiv preprint arXiv:2508.02085](#), 2025.
- 550  
551 Zhan Ling, Kang Liu, Kai Yan, Yifan Yang, Weijian Lin, Ting-Han Fan, Lingfeng Shen, Zhengyin  
552 Du, and Jiecao Chen. Longreason: A synthetic long-context reasoning benchmark via context  
553 expansion. [arXiv preprint arXiv:2501.15089](#), 2025.
- 554  
555 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,  
556 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. [arXiv preprint  
arXiv:2412.19437](#), 2024.
- 557  
558 Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding,  
559 Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *The Twelfth  
International Conference on Learning Representations*, 2023.
- 560  
561 Xinbei Ma, Yiting Wang, Yao Yao, Tongxin Yuan, Aston Zhang, Zhuosheng Zhang, and Hai Zhao.  
562 Caution for the environment: Multimodal llm agents are susceptible to environmental distractions.  
563 In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics  
564 (Volume 1: Long Papers)*, pp. 22324–22339, 2025.
- 565  
566 Sedigheh Mahdavi, Pradeep Kumar Joshi, Lina Huertas Guativa, Upmanyu Singh, et al. Integrating  
567 large language models in financial investments and market analysis: A survey. [arXiv preprint  
arXiv:2507.01990](#), 2025.
- 568  
569 Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia:  
570 a benchmark for general ai assistants. In *The Twelfth International Conference on Learning  
571 Representations*, 2023.
- 572  
573 Yuanhao Qu, Kaixuan Huang, Ming Yin, Kanghong Zhan, Dyllan Liu, Di Yin, Henry C Cousins,  
574 William A Johnson, Xiaotong Wang, Mihir Shah, et al. Crispr-gpt for agentic automation of  
575 gene-editing experiments. *Nature Biomedical Engineering*, pp. 1–14, 2025.
- 576  
577 Mrinal Rawat, Ambuje Gupta, Rushil Goomer, Alessandro Di Bari, Neha Gupta, and Roberto  
578 Pieraccini. Pre-act: Multi-step planning and reasoning improves acting in llm agents. [arXiv  
preprint arXiv:2505.09970](#), 2025.
- 579  
580 Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab of ai agents  
581 designs new sars-cov-2 nanobodies. *Nature*, pp. 1–3, 2025.
- 582  
583 Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru  
584 Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. [arXiv preprint  
arXiv:2507.20534](#), 2025.
- 585  
586 Weixuan Wang, Dongge Han, Daniel Madrigal Diaz, Jin Xu, Victor Rühle, and Saravan Rajmohan.  
587 Odysseybench: Evaluating llm agents on long-horizon complex office application workflows.  
588 [arXiv preprint arXiv:2508.09124](#), 2025a.
- 589  
590 Zhizheng Wang, Qiao Jin, Chih-Hsuan Wei, Shubo Tian, Po-Ting Lai, Qingqing Zhu, Chi-Ping Day,  
591 Christina Ross, Robert Leaman, and Zhiyong Lu. Geneagent: self-verification language agent for  
592 gene-set analysis using domain databases. *Nature Methods*, pp. 1–9, 2025b.
- 593  
594 Bingyang Wei. Requirements are all you need: From requirements to code with llms. In *2024 IEEE  
32nd International Requirements Engineering Conference (RE)*, pp. 416–422. IEEE, 2024.

594 Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won  
595 Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet  
596 challenging benchmark for browsing agents. [arXiv preprint arXiv:2504.12516](#), 2025.  
597

598 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang  
599 Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. [arXiv preprint arXiv:2505.09388](#),  
600 2025.

601 Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan.  $\tau$ -bench: A benchmark for  
602 tool-agent-user interaction in real-world domains. [arXiv preprint arXiv:2406.12045](#), 2024.  
603

604 Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang,  
605 Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation  
606 models. [arXiv preprint arXiv:2508.06471](#), 2025.

607 Linghao Zhang, Shilin He, Chaoyun Zhang, Yu Kang, Bowen Li, Chengxing Xie, Junhao Wang,  
608 Maoquan Wang, Yufan Huang, Shengyu Fu, et al. Swe-bench goes live! [arXiv preprint](#)  
609 [arXiv:2505.23419](#), 2025a.

610 Yiran Zhang, Ruiyin Li, Peng Liang, Weisong Sun, and Yang Liu. Knowledge-based multi-  
611 agent framework for automated software architecture design. In [Proceedings of the 33rd ACM](#)  
612 [International Conference on the Foundations of Software Engineering](#), pp. 530–534, 2025b.  
613

614 Zijing Zhang, Zhanpeng Chen, He Zhu, Ziyang Chen, Nan Du, and Xiaolong Li. Toolexpnet:  
615 Optimizing multi-tool selection in llms with similarity and dependency-aware experience networks.  
616 In [Findings of the Association for Computational Linguistics: ACL 2025](#), pp. 15706–15722,  
617 2025c.  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647

## Appendix

648		
649		
650		
651	<b>A Environmental Settings</b>	<b>13</b>
652		
653	<b>B Prompt and Annotation Details</b>	<b>19</b>
654		
655	B.1 Agent Prompt for Mystery Grid Environment . . . . .	19
656	B.2 Agent Prompt for Sequence Explore Environment . . . . .	20
657	B.3 Agent Prompt for Genetics Lab Environment . . . . .	20
658	B.4 LLM Judge Prompt for Mystery Grid Environment . . . . .	21
659	B.5 LLM Judge Prompt for Sequence Explore Environment . . . . .	22
660	B.6 LLM Judge Prompt for Genetics Lab Environment . . . . .	23
661	B.7 User Prompt . . . . .	25
662	B.8 LLM Judge Prompt for Error Classification . . . . .	25
663		
664		
665		
666		
667	<b>C Extended Experiments and Statements</b>	<b>27</b>
668		
669	C.1 Human Participant Statement . . . . .	27
670	C.2 Human Performance . . . . .	27
671	C.3 Evaluation and Analysis on Score@32 . . . . .	27
672		
673	<b>D Failure Evaluation &amp; Case Study</b>	<b>28</b>
674		
675	D.1 Error Classification Framework . . . . .	28
676	D.2 Case Studies by Manifestation . . . . .	28
677		
678	<b>E Disclosure of LLM Usage</b>	<b>53</b>
679		

---

### A ENVIRONMENTAL SETTINGS

The detailed settings of the three environments in *UltraHorizon* with graphical user interface (GUI) visualizations are illustrated below.

681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755

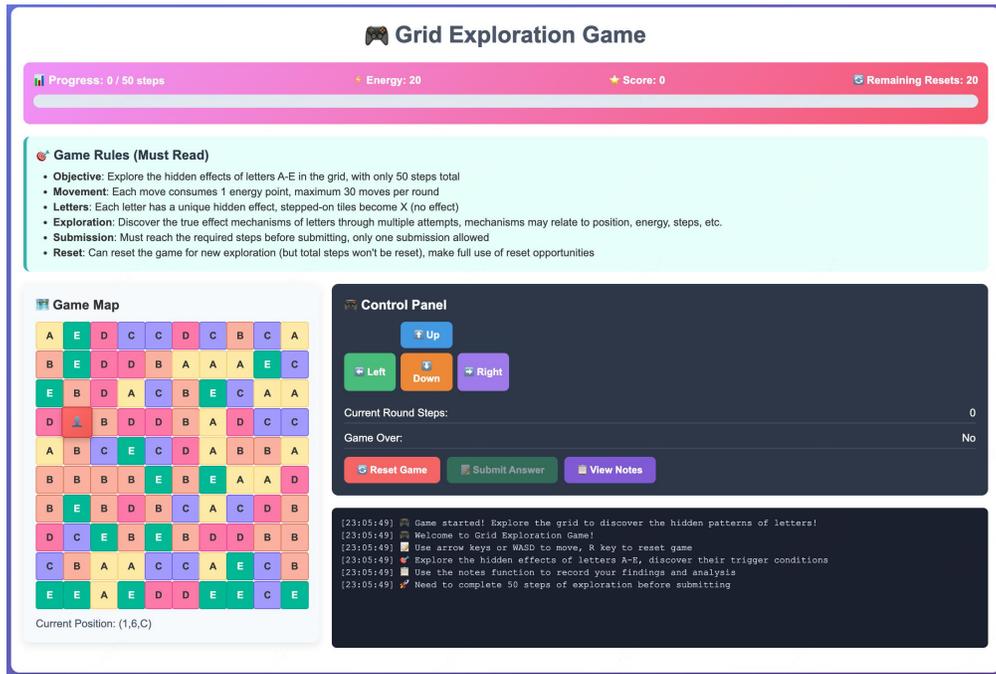


Figure 8: The graphical user interface of the Mystery Grid environment.

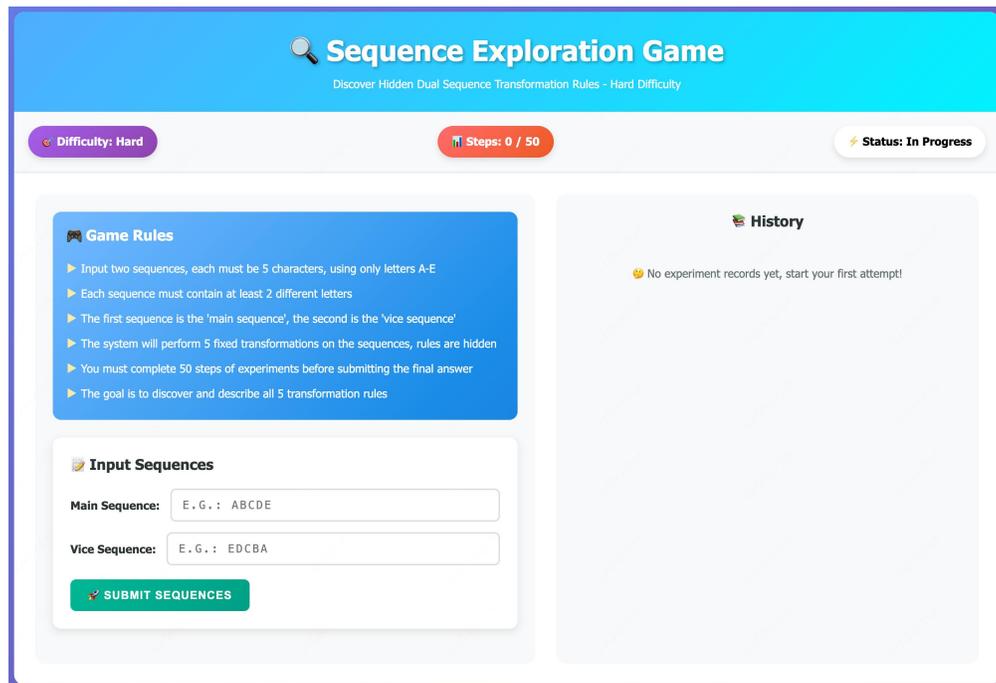


Figure 9: The graphical user interface of the Sequence Explore environment.

756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809

Table 3: Environmental settings in the Mystery Grid environment.

<b>Introduction</b>	
The LLM agent navigates a 10×10 grid to discover hidden rules governing five letters (A-E). Operating with 20 initial energy points and consuming 1 per move, the agent must complete a required number of steps (typically 50) before submitting conclusions. The core challenge involves identifying consistent, non-observable mechanistic patterns through systematic exploration, rather than maximizing score.	
<b>Rules</b>	
1) <b>Letter A:</b> Step count effect. When the total step count is divisible by 3, the agent gains 2 points; otherwise, it loses 1 point.	
2) <b>Letter B:</b> Corner and edge effects. When positioned at corners (coordinates where x and y are both 0 or 9) or edges (coordinates where x or y is 0, 1, 8, or 9), the agent gains 3 points.	
3) <b>Letter C:</b> Visit count-based effect. When the cumulative visit count for that specific letter type is odd, the agent gains 1 point; when even, no points are awarded.	
4) <b>Letter D:</b> Energy management effect. When energy is below 10 points, the agent loses 2 points but gains 10 energy; when energy is 10 or above, the agent gains 1 point.	
5) <b>Letter E:</b> Complex position calculation effect. The agent’s score changes by the value of $(x - y)$ , where x represents the column coordinate and y represents the row coordinate.	
Given Tools	Tool descriptions
1) Move	1) Move in one of four cardinal directions (up, down, left, right) on the grid and update the state.
2) Get Current State	2) Provides the agent’s current position $(x, y)$ , energy level, score, step count, and information about nearby tiles within a two-tile radius.
3) Get Full Map	3) Provides the complete grid state with coordinates and letter positions, allowing the agent to plan exploration routes strategically.
4) Reset	4) Enables the agent to restart the game with a new random grid and start location. This tool is limited to a maximum of 20 uses and is essential for validating hypotheses through investigation.
5) Python Interpreter	5) Allows the agent to utilize Python coding to solve some computation problems.
6) Write Note	6) Append a concise note (e.g., hypotheses, observations, partial results) to a persistent scratchpad for the current episode.
7) Check Note	7) Retrieve previously written notes, to verify hypotheses, avoid repeated mistakes, and support controlled experiments.
8) Commit Final Result	8) Allows the agent to submit its final analysis of the letter effects. This tool can only be used once, after the required total steps have been completed.

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

**Alien Organism Genetics Laboratory**

**Goal:** Explore genetic patterns through experiments and submit a formal experimental report after reaching the required number of steps.

- Each **cross** counts as 1 step; queries/removals do not count.
- You can submit the report only after reaching the minimum number of experimental steps.
- Explanatory text will not reveal answers, please discover the mechanisms through experiments.

**Experiment Progress**

<b>Required Steps</b> 50	<b>Current Steps</b> 0
<b>Remaining Steps</b> 50	<b>Can Submit</b> No
<b>Total Organisms</b> 3	<b>Experiment Log Entries</b> 0

**Quick Tips**

- Perform cross → Generate offspring → Observe phenotypes → Summarize patterns
- Query ID range → Review lineage and phenotypes
- Only when reaching required steps → Submit final report

**Operation Log (Recent)**

```
2025-09-24T15:10:52.635Z :: get_lab_status()
```

[Download Experiment Log JSON](#)

[Reset Environment](#)

The downloaded file is named `alien_organism_genetics_laboratory.json`. Please download this JSON file and send it to the collector.

**Lab Overview**

Environment Name: Alien Organism Genetics Laboratory    Version: Browser UI

Language: English Interface

Below are representative individuals of the initial 3 strains (A, B, C). You can switch to "Perform Cross" to conduct experiments, or search any ID range in "Query Organism".

Individual #1	Individual #2	Individual #3
<p>Generation 0    Viable: Yes</p> <p>Body Size: <b>extra_large</b> (score 445.52)</p> <p>Body Color: <b>red</b> (intensity 95)</p> <p>Shell Shape: <b>spiky</b> (hardness 101.6)</p> <p>Parents: -</p> <p>Initial Strain A</p>	<p>Generation 0    Viable: Yes</p> <p>Body Size: <b>extra_large</b> (score 297.67)</p> <p>Body Color: <b>blue</b> (intensity 60)</p> <p>Shell Shape: <b>smooth</b> (hardness 103.49)</p> <p>Parents: -</p> <p>Initial Strain B</p>	<p>Generation 0    Viable: Yes</p> <p>Body Size: <b>tiny</b> (score 30.23)</p> <p>Body Color: <b>white</b> (intensity 20)</p> <p>Shell Shape: <b>ridged</b> (hardness 98.36)</p> <p>Parents: -</p> <p>Initial Strain C</p>

Figure 10: The graphical user interface of Alien Genetics Laboratory environment.

864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917

Table 4: Environmental settings in the Sequence Explore environment.

<b>Introduction</b>	
A sequence exploration game where the agent uncovers five hidden transformation rules by inputting pairs of letter sequences and observing deterministic and periodic changes, then submits the deduced mechanisms through controlled experimentation.	
<b>Rules</b>	
1) <b>Rule 1:</b> Interleaving or merging characters from main and vice sequences, with the leading sequence possibly changing based on step count or other factors.	
2) <b>Rule 2:</b> Character position-based operations, potentially involving arithmetic on positions (e.g., addition, modular arithmetic) or alphabetic shifts.	
3) <b>Rule 3:</b> Position-dependent selection or copying (e.g., step-number modulo patterns) where specific characters are duplicated or extracted.	
4) <b>Rule 4:</b> Character-wise combination between sequences, possibly involving comparisons (max/min) or operations using historical states (e.g., previous main sequences).	
5) <b>Rule 5:</b> Frequency-based transformations where occurrence patterns drive modifications (e.g., replacing the most frequent character under prime-numbered steps).	
<b>Given Tools</b>	<b>Tool descriptions</b>
1) Input Sequences	1) Submit pairs of fixed-length letter sequences for transformation and receive detailed results of all applied rules at the current step.
2) Python Interpreter	2) Allows the agent to utilize Python coding to solve computation or verification tasks.
3) Write Note	3) Append a concise note (e.g., hypotheses on rules, observations, partial results) to a persistent scratchpad for the current episode.
4) Check Note	4) Retrieve previously written notes, optionally filtered by keywords or indices, to verify hypotheses, avoid repeated mistakes, and support controlled experiments.
5) Commit Final Result	5) Submit the inferred mechanisms of the five rules for evaluation; allowed only once after reaching the required steps.

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971

Table 5: Environmental settings in the Alien Genetics Laboratory environment.

<b>Introduction</b>	
An Alien Genetics Laboratory experiment where the agent discovers inheritance rules of triploid alien organisms through controlled crosses, analyzes trait patterns, and identifies lethal gene combinations, before submitting final conclusions.	
<b>Rules</b>	
1) <b>Ploidy Level:</b> Organisms are triploid; each gene locus carries three alleles rather than the two typical of diploids.	
2) <b>Meiotic Segregation Mechanism:</b> Special meiosis with unequal 1 + 2 segregation produces haploid and diploid gametes.	
3) <b>Viability Constraint:</b> Only triploid zygotes are viable; other ploidy outcomes are lethal.	
4) <b>Body Size Inheritance Pattern:</b> Body size follows an additive dosage effect; alleles contribute quantitatively to phenotype.	
5) <b>Body Size Allele Identification:</b> Three distinct alleles (S1, S2, S3) control size with different quantitative effects.	
6) <b>Body Size Quantitative Values:</b> Approximate allele values: $200 \pm 20$ , $50 \pm 10$ , $10 \pm 5$ .	
7) <b>Color Dominance Hierarchy:</b> Color follows strict dominance C1 (red) > C2 (blue) > C3 (white).	
8) <b>Color Complete Dominance:</b> Phenotype is determined solely by the most dominant allele present.	
9) <b>Shell Shape Cyclic Dominance:</b> Shell alleles show cyclic dominance: H1 (spiky) > H2 (smooth) > H3 (ridged) > H1.	
10) <b>Shell Lethal Combination:</b> Genotype H1 + H2 + H3 is lethal and non-viable.	
<b>Given Tools</b>	<b>Tool descriptions</b>
1) Conduct Cross	1) Perform a controlled cross between two organisms; return offspring phenotypes, viability rates, and lethality statistics.
2) Query Organisms	2) Examine organisms within a specified ID range (e.g., start_id to end_id) for traits and lineage; supports optional phenotype/genotype filters.
3) Get Lab Status	3) Provide current laboratory resource usage, experiment progress, and remaining experiment budget.
4) Remove Organisms	4) Remove specified organisms to manage capacity constraints.
5) Python Interpreter	5) Allow the agent to utilize Python coding for calculations, simulations, or verification.
6) Write Note	6) Append a concise note (e.g., hypotheses, observations, partial results) to a persistent scratchpad for the current episode.
7) Check Note	7) Retrieve previously written notes, optionally filtered by keywords or IDs, to verify hypotheses, avoid repeated mistakes, and support controlled experiments.
8) Commit Final Result	8) Submit the final conclusions describing inferred inheritance rules; allowed only once after reaching the required steps.

## B PROMPT AND ANNOTATION DETAILS

### B.1 AGENT PROMPT FOR MYSTERY GRID ENVIRONMENT

#### GAME OBJECTIVE:

You are an agent in a 10x10 grid world. Your goal is to discover the hidden rules of effects of different letters (A-E).

#### BASIC RULES:

- You start at initial position with 20 energy points
- Each move costs 1 energy point
- You have a maximum of 30 steps per game round, and a required total steps of required\_steps, you cannot move after reaching the limit, and you cannot commit before reaching the limit.
- The grid contains letters A through E, each with unique hidden effects on your score or energy. X means you have already stepped on this tile and it has no effect.
- The effects of each letter is consistent across games, but you need to discover them through exploration.
- After you step on a tile, its letter changes to 'X'.
- The game ends when you run out of energy or reach 30 steps.

#### IMPORTANT NOTES:

- Letter effects are HIDDEN but follow CONSISTENT rules across different games and you must discover the rules through experimentation. The effects may be related to your position, energy, step count or visit count.
- Each letter (A-E) has a different scoring mechanics that you need to figure out. The effect of each tile should not be concluded from a single try; multiple validations under different situations are essential to confirm that the observed outcome is not merely a random occurrence. You should fully utilize the reset opportunity to explore the effects of each letter.
- Use notes to record your plans, ideas, and discoveries, and review them frequently to stay on track. Also, record the effects into notes of each letter as you discover them. You may also need to record the changes after each move for later analysis.
- You can reset and replay multiple times to learn the patterns. Your goal is not to maximize the score, but to explore the mechanism clearly. It is recommended to make full use of each game until all steps or energy are exhausted, avoiding resets too easily. At the same time, repeated trials should be conducted to ensure the conclusions are correct before committing. You can only commit your answer once and then the game ends, commit only when you are confident about the effects of each letter.
- Since each step consumes 1 energy, it is therefore not counted as an effect.

#### COMMIT REQUIREMENTS:

- Complete coverage: You must provide exactly one effect description for each of A, B, C, D, and E.
- Trigger condition required: Your description must specify the underlying trigger condition, which may depend on: step count, visit count, energy, grid position. Stating only surface patterns such as “sometimes +2, sometimes -1” is not sufficient.
- No vague descriptions: Do not use terms like “random,” “uncertain,” or “complex.” Do not provide multiple conflicting explanations.
- Evaluation is strict: only if you clearly identify the correct trigger condition and effect will your answer be marked as correct.

#### COORDINATE FORMAT:

All positions are shown as (x,y,letter) where x=column, y=row, letter=current tile letter  
Now start exploring and discovering the hidden mechanics.

1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079

## B.2 AGENT PROMPT FOR SEQUENCE EXPLORE ENVIRONMENT

### GAME OBJECTIVE:

You are an agent facing a hidden dual-sequence transformation system. Your goal is to discover the hidden rules applied to pairs of sequences of letters (A-E).

### BASIC RULES:

- You must input TWO sequences of EXACTLY 5 CHARACTERS each, using only letters A-E
- Each sequence must contain at least 2 different letters from A-E
- The first sequence is the "main str" and the second is the "vice str"
- Each input pair consumes 1 step. There is a required total steps of  $\{required\_steps\}$  to complete the game
- After inputting a sequence pair, they will undergo a fixed chain of transformations
- All transformations apply the same rules consistently, operating on current results and/or original inputs
- The same hidden rules apply consistently across the game, but you must discover them through exploration
- Game ends when you reach the maximum step count or submit your final answer

IMPORTANT NOTES: - Effects are deterministic but hidden; repeated testing is needed to confirm true rules

- Each rule may depend on sequence content, position, history, or relationships between main/vice
- Use notes to record observed transformations and hypotheses
- Do not commit your final answer until you are confident, as you can only commit once
- The goal is to uncover and describe the mechanisms behind dual-sequence transformations

### SEQUENCE FORMAT:

Inputs are pairs like ("ABCDE", "DECBA"), ("AABCE", "DDEAC"). Each must be exactly 5 characters with at least 2 different letters.

Outputs are shown step by step with rule applications.

### DIFFICULTY LEVELS:

- Easy: Uses simplified versions of rules 1-5
- Hard: Uses complex versions of rules 1-5

Now begin experimenting and uncover the transformation rules.

## B.3 AGENT PROMPT FOR GENETICS LAB ENVIRONMENT

### GAME OBJECTIVE:

You are an agent in a 10x10 grid world. Your goal is to discover the hidden rules of effects of different letters ( $\{letter\_range\}$ ).

### BASIC RULES:

- You start at initial position with 20 energy points
- Each move costs 1 energy point
- You have a maximum of 30 steps per game round  $required\_steps\_text$
- The grid contains letters  $\{letter\_range\}$ , each with unique hidden effects on your score or energy. X means you have already stepped on this tile and it has no effect.
- The effects of each letter is consistent across games, but you need to discover them through exploration.
- After you step on a tile, its letter changes to 'X'
- The game ends when you run out of energy or reach 30 steps

### IMPORTANT NOTES:

- Letter effects are HIDDEN but follow CONSISTENT rules across different games and you

1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133

must discover the rules through experimentation. The effects may be related to your position, energy, step count or visit count.

- Each letter (*{letter\_range}*) has a different scoring mechanics that you need to figure out. The effect of each tile should not be concluded from a single try; multiple validations under different situations are essential to confirm that the observed outcome is not merely a random occurrence. You should fully utilize the reset opportunity to explore the effects of each letter.
- Use notes to record your plans, ideas, and discoveries, and review them frequently to stay on track. Also, record the effects into notes of each letter as you discover them. You may also need to record the changes after each move for later analysis.
- You can reset and replay multiple times to learn the patterns. Your goal is not to maximize the score, but to explore the mechanism clearly. It is recommended to make full use of each game until all steps or energy are exhausted, avoiding resets too easily. At the same time, repeated trials should be conducted to ensure the conclusions are correct before committing. You can only commit your answer once and then the game ends, commit only when you are confident about the effects of each letter.
- Since each step consumes 1 energy, it is therefore not counted as an effect.

#### COMMIT REQUIREMENTS:

- Complete coverage: You must provide exactly one effect description for each of *{letter\_range}*.
- Trigger condition required: Your description must specify the underlying trigger condition, which may depend on: step count, visit count, energy, grid position. Stating only surface patterns such as "sometimes +2, sometimes -1" is not sufficient.
- No vague descriptions: Do not use terms like "random," "uncertain," or "complex." Do not provide multiple conflicting explanations.
- Evaluation is strict: only if you clearly identify the correct trigger condition and effect will your answer be marked as correct.

#### COORDINATE FORMAT:

All positions are shown as (x,y,letter) where x=column, y=row, letter=current tile letter

Now start exploring and discovering the hidden mechanics.

## B.4 LLM JUDGE PROMPT FOR MYSTERY GRID ENVIRONMENT

You are an evaluator. Compare the agent's submitted mapping of letters to effect descriptions against the ground truth mapping.

The agent should have inferred what each letter (A, B, C, D, E) does when stepped on.

<Ground truth mapping>:

{ground\_truth}

<Agent submitted mapping>:

{content}

Compare the agent's descriptions with the ground truth effects. A mapping is considered correct if the agent's description captures the essential rule of the effect, even if the wording is different. Since each step consumes 1 energy, it is therefore not counted as an effect. However, it is also correct if the agent specifies a consumption of 1 energy for each effect. You should follow these rules to judge:

Evaluate correctness according to these strict rules:

#### 1. **\*\*Trigger Condition Must Be Identified\*\***

An effect description is correct **\*\*only if\*\*** the agent identifies the underlying **\*trigger condition\*** (e.g., step count, visit count parity, energy level, or coordinate-based calculation). Simply describing observed outcome patterns (e.g., "sometimes +2, sometimes -1", "alternating pattern", "mostly +1 with exceptions") **\*\*without stating the actual trigger**

1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187

condition\*\* is incorrect.

## 2. **Equivalent Wording Allowed**

Different wording is acceptable only if it expresses the *same logical condition*.

Example: “odd/even visits” “visit count parity”.

Example: “boundary positions” “corners and edges”.

Descriptions that skip the explicit condition and only give surface observations are incorrect.

## 3. **Vague or Wrong Attributions are Incorrect**

If the agent uses vague terms such as “variable”, “random”, “uncertain”, or “complex behavior” instead of specifying the true condition, it is incorrect.

If the agent attributes the effect to the wrong factor (e.g., says “position-dependent” when the actual rule depends on step count or energy), it is incorrect.

## 4. **One-to-One Mapping Requirement**

Each symbol must map to exactly one effect rule.

Multiple conflicting explanations, incomplete conditions, or missing mappings are incorrect.

## 5. **Evaluation Criteria**

Mark **Correct** only if:

The trigger condition identified by the agent matches the ground truth condition.

The outcome description aligns with the ground truth rule.

Otherwise, mark **Incorrect**.

Format your answer as json:

```
{
  "final_score": 40,
  "score_breakdown": [
    {"criterion": "A", "max_score": 20, "awarded_score": 20, "comment": <your
    ↪comment>},
    {"criterion": "B", "max_score": 20, "awarded_score": 0, "comment": <your
    ↪comment>},
    {"criterion": "C", "max_score": 20, "awarded_score": 0, "comment": <your
    ↪comment>},
    {"criterion": "D", "max_score": 20, "awarded_score": 20, "comment": <your
    ↪comment>},
    {"criterion": "E", "max_score": 20, "awarded_score": 0, "comment": <your
    ↪comment>},
  ]
}
```

## B.5 LLM JUDGE PROMPT FOR SEQUENCE EXPLORE ENVIRONMENT

You are an evaluator. Compare the agent’s submitted description of transformation rules against the ground truth rules.

The agent should have inferred what each rule does in the sequence transformation chain.

<Ground truth rules>: {ground\_truth}

<Agent submitted description>: {content}

Compare the agent’s descriptions with the ground truth rules. A rule description is considered correct if the agent’s explanation captures the essential mechanism of the rule, even if the wording is different. Each rule has 20 points, for a total of 100 points. Provide a detailed breakdown of scores for each rule and the final score.

Evaluate correctness according to these strict rules:

1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241

### 1. **Mechanism Must Be Identified**

A rule description is correct **only if** the agent identifies the underlying *transformation mechanism* (e.g., character interleaving, position-based operations, step-count dependencies, history references).

Simply describing observed patterns without stating the actual mechanism is incorrect.

### 2. **Equivalent Wording Allowed**

Different wording is acceptable only if it expresses the *same logical transformation*.

Example: "alternating characters" = "interleaving main and vice".

Example: "reverse and shift" = "flip sequence and advance letters".

Example: "current step number" = "total steps" / "num of total experiments"

### 3. **Vague or Wrong Mechanisms are Incorrect**

If the agent uses vague terms such as "complex pattern", "variable behavior", or "depends on context" without specifying the true mechanism, it is incorrect.

If the agent attributes the rule to the wrong mechanism, it is incorrect (0 score should be assigned).

### 4. **Complete Rule Chain Required**

Each rule in the sequence must be correctly identified and described.

Missing rules, incomplete mechanisms, or conflicting explanations are incorrect.

### 5. **Evaluation Criteria**

Mark **Correct** only if:

- The transformation mechanism identified matches the ground truth rule.
- The description explains how inputs are transformed to outputs.
- The agent shows understanding of when/how the rule applies.

Format your answer as json:

```
{
  "final_score": 30,
  "score_breakdown": [
    {"criterion": "rule_1", "max_score": 20, "awarded_score": 10, "comment": <your
    ↪comment>},
    {"criterion": "rule_2", "max_score": 20, "awarded_score": 0, "comment": <your
    ↪comment>},
    {"criterion": "rule_3", "max_score": 20, "awarded_score": 0, "comment": <your
    ↪comment>},
    {"criterion": "rule_4", "max_score": 20, "awarded_score": 20, "comment": <your
    ↪comment>},
    {"criterion": "rule_5", "max_score": 20, "awarded_score": 0, "comment": <your
    ↪comment>},
  ]
}
```

## B.6 LLM JUDGE PROMPT FOR GENETICS LAB ENVIRONMENT

You are an evaluator. Compare the agent's submitted description of inheritance rules against the scoring rules.

Scoring rules:

**## A. Fundamental Understanding (25 points)**

1. Recognition of triploidy - Identifies that the organism is triploid (three alleles per locus).

\*(Medium, 15 points)\*

2. Meiosis process - Correctly describes the special meiosis mechanism (gametes are 1n and 2n due to unequal segregation). \*(Easy, 5 points)\*

3. Viability constraint - Explains that only

1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295

triploid zygotes are viable; other ploidy combinations are lethal. \*(Easy, 5 points)\*

#### ## B. Body Size Trait (Dosage Effect) (35 points)

4. Inheritance rule - Discovers that body size follows an additive dosage effect. \*(Easy, 5 points)\*  
5. Allele identification - Correctly identifies the existence of three distinct alleles for size. \*(Medium, 10 points)\*

6. Quantitative description - Provides accurate quantitative values for the three alleles (approx.  $200 \pm 20$ ;  $50 \pm 10$ ;  $10 \pm 5$ ). \*(Hard, 20 points)\*

#### ## C. Color Trait (Dominance Hierarchy) (10 points)

7. Dominance hierarchy - Identifies that color follows a strict dominance order: Red (C1) > Blue (C2) > White (C3). \*(Easy, 5 points)\*

8. Complete dominance - States that the dominance is complete: phenotype intensity is determined solely by the dominant allele. \*(Easy, 5 points)\*

#### ## D. Shell Trait (Cyclic Interaction & Lethality) (30 points)

9. Cyclic dominance - Identifies that shell shape alleles follow a cyclic dominance relationship: Spiky (H1) > Smooth (H2) > Ridged (H3) > Spiky. \*(Medium, 10 points)\*

10. Lethal combination - Explains that the presence of all three shell alleles (H1 + H2 + H3) is lethal. \*(Hard, 20 points)\*

<Agent submitted report>: {content}

Evaluation guidelines:

##### 1. **Mechanism Identification Required**

For each item, award the full score if the point is clearly and completely correctly addressed; partial credit **cannot** be given if the description is incomplete, vague, or not correct.

##### 2. **Equivalent Wording Allowed**

Different wording is correct if logically equivalent. - "3 gene copies" = "triploid"

- "1+2 split" = "unequal segregation"

- "dosage/additive effect" = "sum of allele contributions"

- "dominance ladder" = "dominance hierarchy"

- "rock-paper-scissors interaction" = "cyclic dominance"

##### 3. **Incomplete or Vague = Incorrect**

Descriptions like "complex" or "varies" without identifying the true mechanism are incorrect.

Format your answer as json:

```
{
  "final_score": 87,
  "score_breakdown": [
    {"criterion": "Triploidy recognition", "max_score": 15, "awarded_score": 0,
    ↪ "comment": <your comment>},
    {"criterion": "Meiosis process (1n/2n gametes)", "max_score": 5,
    ↪ "awarded_score": 5, "comment": <your comment>},
    {"criterion": "Viability constraint (only triploid survives)", "max_score": 5,
    ↪ "awarded_score": 5, "comment": <your comment>},
    {"criterion": "Body size: dosage effect", "max_score": 5, "awarded_score": 5,
    ↪ "comment": <your comment>},
    {"criterion": "Body size: allele identification", "max_score": 10,
    ↪ "awarded_score": 10, "comment": <your comment>},
    {"criterion": "Body size: quantitative values", "max_score": 20,
    ↪ "awarded_score": 15, "comment": <your comment>},
    {"criterion": "Color: dominance hierarchy", "max_score": 5, "awarded_score":
    ↪ 5, "comment": <your comment>},
    {"criterion": "Color: complete dominance", "max_score": 5, "awarded_score": 5,
    ↪ "comment": <your comment>},
    {"criterion": "Shell: cyclic dominance", "max_score": 10, "awarded_score": 0,
    ↪ "comment": <your comment>},
    {"criterion": "Shell: lethal combination", "max_score": 20, "awarded_score":
    ↪ 0, "comment": <your comment>}
  ]
}
```

1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349

}

## B.7 USER PROMPT

Based on the observations and your needs, proactively select the most appropriate tool (You can only call one tool at each step). Think before you act. When you call tools, you cannot add any content after the calling or the calling will not be identified. Explicitly state your thought and next plan with the format: "### Thought: [your thought]\n ### Plan: [your plan]". Remember to note down your thoughts, plans and observations when necessary, and review your notes frequently to stay on track. After using each tool, clearly explain the execution results and suggest the next steps. If you want to commit your answer, you should check your notes and analyze them carefully before committing.

## B.8 LLM JUDGE PROMPT FOR ERROR CLASSIFICATION

You will act as an expert in failure analysis for long-horizon agent conversations/traces. Using the fixed flat catalog below, carefully review the complete message sequence and produce a structured judgment.

Failure catalog (flat, non-hierarchical)

### 1) Repetitive Looping

- Possible Root Causes: Cognitive inertia (error locking, behavioral inertia)
- Explanation: Clings to early habits and lacks mechanisms for reflection/correction.
- Common signs: repeats ineffective steps/tool calls without progress; multiple consecutive turns with no state change or information gain; outputs weakly related to the goal.

### 2) Premature Convergence

- Possible Root Causes: Cognitive inertia (hypothesis dependence); foundational capability gap (weak planning)
- Explanation: Becomes locked to the initial hypothesis, or lacks exploratory planning ability.
- Common signs: stops exploring alternatives despite counter-evidence; branch width too narrow; early termination without sufficient evidence threshold.

### 3) Incoherent Planning

- Possible Root Causes: Foundational capability gap (weak planning, logical errors)
- Explanation: Planning competence itself is insufficient.
- Common signs: contradictory/fragmented steps; missing prerequisites; task order oscillation; plan conflicts with observations; lack of backtracking/repair.

### 4) Misaligned Tool Usage

- Possible Root Causes: Foundational capability gap (tool misunderstanding); cognitive inertia (habitual use)
- Explanation: Either does not understand the tool or overuses it out of habit.
- Common signs: wrong tool choice; misreading/misinterpreting outputs; gratuitous/unnecessary tool calls (e.g., unnecessary Python) not aligned with the goal.

### 5) Memory Issues

- Possible Root Causes: Foundational capability gap (memory mismanagement)
- Explanation: Defects in core memory mechanisms (store/retrieve/update).
- Common signs: self-contradictions; asks again for already known info; forgets constraints; overwrites key facts; low retrieval hit rate.

### 6) Uncontrolled Experiments

1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403

- Possible Root Causes: Foundational capability gap (logical reasoning, weak planning)
- Explanation: Lacks the basics of experimental control (no single-variable control).
- Common signs: changes multiple factors at once; confounded results; causal chain not closed; results hard to reproduce.

#### 7) Error Propagation

- Possible Root Causes: Cognitive inertia (lack of strategic elasticity); foundational capability gap (memory, reflection)
- Explanation: Errors (e.g., a tool-call mistake) keep repeating; unable to self-correct.
- Common signs: repeats the same error; no policy change after negative feedback; long recovery time; strategy unchanged despite failures.

#### 8) Environment Mis-modeling

- Possible Root Causes: Foundational capability gap (environment/world-modeling)
- Explanation: Internal representation of hidden rules/transitions is inconsistent or outdated.
- Common signs: persistent prediction–observation gap; unstable behavior/outcomes under identical conditions; mispredicted transitions.

#### Task

- Read the complete trace provided by the user (each line is one message JSON).
- Using content, order, and tool-use behavior, independently decide for each category whether the failure exists.
- If it exists, label:
  - a) exists = “yes”
  - b) indices = [0-based indices of messages where this issue is visible; multiple allowed]
  - c) reason = a concise explanation (2–3 sentences max), citing key evidence if useful.
- If it does not exist, set exists = “no”, indices = [], and reason can be an empty string.
- Only use evidence visible in the trace; multiple categories may co-occur; avoid over-assumptions.

#### Output

Return only a single JSON object (no extra text, no Markdown code fences). The object must include the 10 keys below, each mapping to an object:

```
{
  "Repetitive Looping": {"exists": "yes|no", "indices": [int...], "reason":
  ↪ "..."},
  "Premature Convergence": {"exists": "yes|no", "indices": [int...], "reason":
  ↪ "..."},
  "Incoherent Planning": {"exists": "yes|no", "indices": [int...], "reason":
  ↪ "..."},
  "Misaligned Tool Usage": {"exists": "yes|no", "indices": [int...], "reason":
  ↪ "..."},
  "Memory Issues": {"exists": "yes|no", "indices": [int...], "reason": "..."},
  "Uncontrolled Experiments": {"exists": "yes|no", "indices": [int...], "reason":
  ↪ "..."},
  "Error Propagation": {"exists": "yes|no", "indices": [int...], "reason": "..."},
  "Environment Mis-modeling": {"exists": "yes|no", "indices": [int...], "reason":
  ↪ "..."}
}
```

- Each index should point to a message where the failure is directly observable in the trace.
- If uncertain, default to “no”.

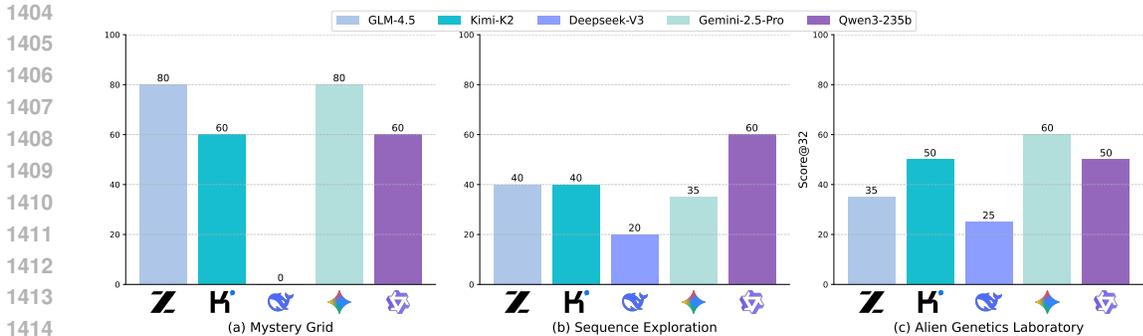


Figure 11: The average score@32 of various LLMs with limited steps on *UltraHorizon*.

## C EXTENDED EXPERIMENTS AND STATEMENTS

### C.1 HUMAN PARTICIPANT STATEMENT

To evaluate the performance gap between humans and LLM agents in long-horizon exploration tasks, we conducted human experiments with 33 participants across the three *UltraHorizon* environments. Each participant was tasked with uncovering hidden rules through iterative interaction, using the same tools available to agents (e.g., move, reset, commit). Participants were instructed to explore freely and submit their final conclusions only when they were confident. All interactions, including tool usage and hypothesis submissions, were recorded for comparative analysis. The study adhered to strict ethical standards: participants provided informed consent, no personally identifiable information was collected, and all data remained anonymized.

### C.2 HUMAN PERFORMANCE

As verified in Figure 1, human participants significantly outperformed all LLM agents, achieving an average score of 26.52 compared to the best LLM score of 14.33 over these environments. This result underscores critical limitations in current agents’ abilities to sustain reasoning, manage memory, and adapt strategies over extended horizons—highlighting the need for improved agent architectures tailored to long-horizon, partially observable settings.

The corresponding results of human participants under three individual environments are listed in Table 6.

Table 6: Average scores of human participants in different environments.

Environments	Scores
Mystery Grid	25.88
Sequence Exploration	24.29
Genetics Laboratory	47.50

### C.3 EVALUATION AND ANALYSIS ON SCORE@32

**Criteria of Score@32.** We introduce a novel evaluation metric, score@k, designed to assess model performance in environments with inherent stochasticity. Specifically, when  $k = 32$ , score@32 represents the aggregated maximum scores obtained across all rule sets in 32 independent experimental trials conducted within the same environment. This metric effectively mitigates the impact of instability in model outputs by accounting for best-case performance within multiple trials, thereby providing a more robust assessment of a model’s generalization capabilities and overall performance in complex scenarios.

**Experimental Results.** The corresponding results in the main experiments with step limitations, free scenario by removing the constraints on steps on GLM 4.5 are reported in Figures 11 and 12, respectively.

Figure 11 confirms the performance hierarchy observed in Section 4.2, with Gemini-2.5-Pro and GLM-4.5 leading across environments, but reveals divergent exploration strategies. Gemini-2.5-Pro employs systematic sampling followed by targeted testing, while GLM-4.5 uses iterative refinement.

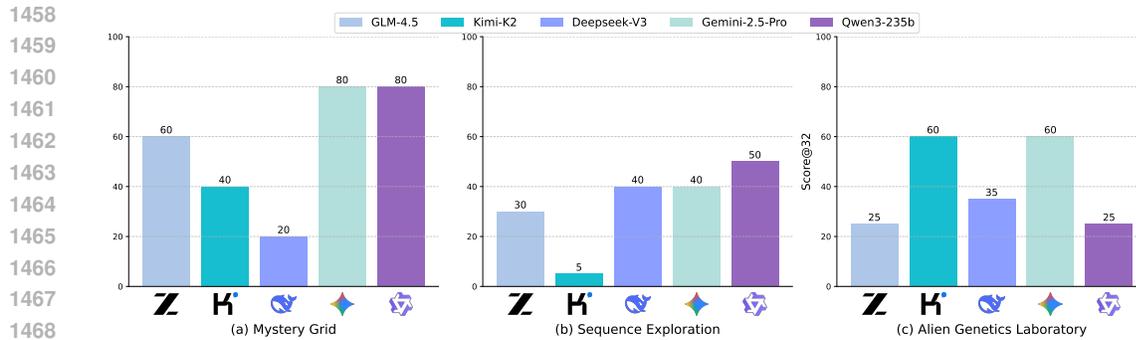


Figure 12: The average score@32 of various LLMs without limiting steps on *UltraHorizon*.

Figure 12 reveals significant resource inefficiencies, particularly in Sequence Exploration, where redundant experiments predominate, and Qwen3-235b requires 40% more steps than Gemini-2.5-Pro to achieve comparable gains.

The above results confirm and extend the core findings of Section 4, where strategy choice and resource efficiency are pivotal to long-horizon performance, and error temporality reveals actionable design priorities (e.g., mitigating early-stage cognitive inertia).

## D FAILURE EVALUATION & CASE STUDY

### D.1 ERROR CLASSIFICATION FRAMEWORK

In this section, we begin by constructing a taxonomy for common agent errors in long-horizon tasks. We manually classify these errors by reading through the agents’ interaction trajectories. Based on this analysis, we identify recurring failure patterns and develop a set of categories that capture the most common issues.

We then use the Gemini-2.5-Pro to automatically classify these errors by applying a tailored prompt to agent trajectories. Gemini’s predictions are analyzed to identify specific failure points, providing both the type of error and an explanation of why it occurred. The prompt used for error classification is provided in Appendix B.8. A detailed breakdown of these errors is shown in Table 7, where we categorize the manifestations, explain their causes, and trace their underlying issues.

### D.2 CASE STUDIES BY MANIFESTATION

This section presents case studies for each of the error categories identified in the previous subsection. For each failure manifestation, we provide detailed examples from agent trajectories, illustrating how these errors manifest in practice. The case studies serve to provide insight into the causes and effects of these failures, offering a practical understanding of how agents fail in long-horizon tasks.

#### D.2.1 CASE TRAJECTORY OF REPETITIVE LOOPING (ROOT: I)

**Setup.** Environment: *Alien Genetics Laboratory*; Model: DeepSeek-V3;

**Model-provided reason.** “The agent becomes stuck in a rigid loop of `conduct_cross` → `note_tool` → `query_organisms`. From message 28 onwards, it repeatedly crosses blue and white organisms, obtains similar results, and writes nearly identical notes (e.g., compare notes in messages 30, 36, 42). This pattern continues for dozens of steps without significant information gain or change in strategy.”

Message 28

<think>  
 ### Execution Results:

1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565

Table 7: Failure manifestations for long-horizon agents, with possible root causes and explanations.

Manifestation	Explanation	Root Causes
Repetitive Looping	Clings to early habits and lacks mechanisms for reflection/correction.	I
Premature Convergence	Becomes locked to the initial hypothesis or lacks exploratory planning ability.	I & II
Incoherent Planning	Planning competence itself is insufficient.	II
Misaligned Tool Usage	Either does not understand the tool or overuses it out of habit.	I & II
Memory Issues	Defects in core memory mechanisms (store/retrieve/update).	II
Uncontrolled Experiments	Lacks the basics of scientific experimental thinking (no single-variable control).	II
Error Propagation	Errors (e.g., a tool-call mistake) keep repeating; unable to self-correct.	I & II
Environment Mis-modeling	Internal representation of hidden rules or transitions is inconsistent or outdated; persistent prediction–observation gaps and unstable outcomes under identical conditions.	II

Legend: I = Cognitive inertia; II = Foundational capability gap.

- **Offspring from Cross 43 × 3:**  
- **Color:** Blue and white offspring observed, confirming the presence of both alleles.  
- **Shell Shape:** Smooth and ridged shells, indicating complex interactions.  
- **Body Size:** Varied (extra-large, large, small, tiny), suggesting multiple alleles or incomplete dominance.

### Observations:

1. **Color:**

- Blue and white offspring suggest incomplete dominance or co-dominance.

2. **Shell Shape:**

- Smooth and ridged shells indicate potential epistasis or multiple loci.

3. **Body Size:**

- Multiple sizes suggest complex inheritance patterns.

### Next Steps:

1. **Cross blue and white offspring** to test color inheritance further (e.g., ID 54 × ID 56).

2. **Record notes** on the observed patterns and plan additional crosses.

### Plan: Conduct a cross between a blue offspring (ID 54) and a white offspring (ID 56) to explore color inheritance.

</think>

<tool\_call>

```
{
  "id": "call_0v2mxmegy1vmyansththbjb",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":54,\"parent2_id\":56,\"num_offspring\":10}"
  }
}
```

</tool\_call>

Message 30

<think>

### Execution Results:

1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619

- **Cross:** Offspring ID 54 (blue, smooth, large) × Offspring ID 56 (white, smooth, small).
- **Viable Offspring:** 10 out of 24 fertilization attempts (viability rate: 41.7%).
- **Lethal Offspring:** 14 (non-viable combinations).
- **Offspring Phenotypes:**
- **Body Size:** Varied (extra-large, large, small, tiny).
- **Color:** Blue and white offspring observed.
- **Shell Shape:** Smooth and ridged shells appear.

### Observations:

1. **Color:**

- Blue and white offspring confirm incomplete dominance or co-dominance.

2. **Shell Shape:**

- Smooth and ridged shells suggest complex interactions.

3. **Lethality:**

- High lethality (58.3%) indicates potential lethal combinations.

### Next Steps:

1. **Analyze the offspring** to confirm color and shell shape inheritance.
2. **Record notes** on the observed patterns.
3. **Plan further crosses** to explore the high lethality and rare phenotypes.

### Plan: Use the note\_tool to summarize the latest observations and plan next steps.

</think>

<tool\_call>

```
{
  "id": "call_eil3iatnsmjpt7kv8xjouncw",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\": \"write_note\", \"note\": \"### Latest
↳ Observations:\\n1. **Color**:\\n - Blue and white offspring from blue white
↳ cross confirm incomplete dominance or co-dominance.\\n2. **Shell Shape**:\\n
↳ Smooth and ridged shells indicate complex interactions or multiple loci.\\n3.
↳ **Lethality**:\\n - High (58.3%) in blue white cross, suggesting lethal
↳ combinations.\\n\\n### Next Steps:\\n- Analyze offspring to confirm color and
↳ shell shape inheritance.\\n- Investigate lethality by analyzing non-viable
↳ combinations.\\n- Cross offspring to explore rare phenotypes.\"}"
  }
}
```

</tool\_call>

### Message 32

<think>

### Execution Results:

- **Note Added:** Successfully recorded the latest observations and planned next steps.

### Next Steps:

1. **Analyze the offspring** from the blue × white cross to confirm color and shell shape inheritance.
2. **Investigate lethality** by analyzing non-viable combinations.
3. **Cross offspring** to explore rare phenotypes and further test inheritance rules.

### Plan: Query the offspring from the latest cross (IDs 64-73) to examine their traits in detail.

</think>

<tool\_call>

```
{
  "id": "call_ro02ek1ktp8x3cacboeol9t3",
```

1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673

```
"type": "function",
"function": {
  "name": "query_organisms",
  "arguments": "{\"start_id\":64,\"end_id\":73}"
}
}
```

</tool\_call>

### Message 34

<think>

### Execution Results:

- **Offspring from Cross 54 × 56:**
- **Color:** Blue and white offspring observed, confirming the presence of both alleles.
- **Shell Shape:** Smooth and ridged shells, indicating complex interactions.
- **Body Size:** Varied (extra-large, large, small, tiny), suggesting multiple alleles or incomplete dominance.

### Observations:

1. **Color:**  
- Blue and white offspring confirm incomplete dominance or co-dominance.
2. **Shell Shape:**  
- Smooth and ridged shells suggest potential epistasis or multiple loci.
3. **Body Size:**  
- Multiple sizes indicate complex inheritance patterns.

### Next Steps:

1. **Cross blue and white offspring** to further test color inheritance (e.g., ID 64 × ID 68).
2. **Record notes** on the observed patterns and plan additional crosses.
3. **Investigate lethality** by analyzing non-viable combinations.

### Plan: Conduct a cross between a blue offspring (ID 64) and a white offspring (ID 68) to explore color inheritance further.

</think>

<tool\_call>

```
{
  "id": "call_26tdnzfyjqrjot6rqr7gcogk",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":64,\"parent2_id\":68,\"num_offspring\":10}"
  }
}
```

</tool\_call>

### Message 36

<think>

### Execution Results:

- **Cross:** Offspring ID 64 (blue, ridged, extra-large) × Offspring ID 68 (white, smooth, tiny).
- **Viable Offspring:** 10 out of 15 fertilization attempts (viability rate: 66.7%).
- **Lethal Offspring:** 5 (non-viable combinations).
- **Offspring Phenotypes:**
- **Body Size:** Varied (extra-large, large, small, tiny).
- **Color:** Blue and white offspring observed.
- **Shell Shape:** Smooth and ridged shells appear.

### Observations:

1. **Color:**  
- Blue and white offspring confirm incomplete dominance or co-dominance.
2. **Shell Shape:**  
- Smooth and ridged shells suggest complex interactions.

1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727

### 3. **Lethality:**

- *Lower lethality (33.3%) compared to previous crosses, indicating fewer lethal combinations.*

#### ### Next Steps:

1. **Analyze the offspring** to confirm color and shell shape inheritance.
2. **Record notes** on the observed patterns.
3. **Plan further crosses** to explore the lower lethality and rare phenotypes.

### Plan: Use the note\_tool to summarize the latest observations and plan next steps.

</think>

<tool\_call>

```
{
  "id": "call_t48lyw53x103cd4s5oe9pude",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\": \"write_note\", \"note\": \"### Latest
↳ Observations:\\n1. **Color**:\\n - Blue and white offspring from blue white
↳ cross confirm incomplete dominance or co-dominance.\\n2. **Shell Shape**:\\n
↳ Smooth and ridged shells indicate complex interactions or multiple loci.\\n3.
↳ **Lethality**:\\n - Lower (33.3%) in this cross, suggesting fewer lethal
↳ combinations.\\n\\n### Next Steps:\\n- Analyze offspring to confirm color and
↳ shell shape inheritance.\\n- Investigate lethality by analyzing non-viable
↳ combinations.\\n- Cross offspring to explore rare phenotypes.\"}"
```

}

</tool\_call>

### Message 38

<think>

#### ### Execution Results:

- **Note Added:** Successfully recorded the latest observations and planned next steps.

#### ### Next Steps:

1. **Analyze the offspring** from the blue × white cross to confirm color and shell shape inheritance.
2. **Investigate lethality** by analyzing non-viable combinations.
3. **Cross offspring** to explore rare phenotypes and further test inheritance rules.

### Plan: Query the offspring from the latest cross (IDs 74-83) to examine their traits in detail.

</think>

<tool\_call>

```
{
  "id": "call_xagw0t7y56bkt3c4eyb5s8qb",
  "type": "function",
  "function": {
    "name": "query_organisms",
    "arguments": "{\"start_id\": 74, \"end_id\": 83}"
```

}

</tool\_call>

### Message 40

<think>

#### ### Execution Results:

- **Offspring from Cross 64 × 68:**
- **Color:** Blue and white offspring observed, confirming the presence of both alleles.
- **Shell Shape:** Smooth and ridged shells, indicating complex interactions.

1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781

- **Body Size:** Varied (extra-large, large, small, tiny), suggesting multiple alleles or incomplete dominance.

### Observations:

1. **Color:**

- Blue and white offspring confirm incomplete dominance or co-dominance.

2. **Shell Shape:**

- Smooth and ridged shells suggest potential epistasis or multiple loci.

3. **Body Size:**

- Multiple sizes indicate complex inheritance patterns.

### Next Steps:

1. **Cross blue and white offspring** to further test color inheritance (e.g., ID 74 × ID 76).

2. **Record notes** on the observed patterns and plan additional crosses.

3. **Investigate lethality** by analyzing non-viable combinations.

### Plan: Conduct a cross between a blue offspring (ID 74) and a white offspring (ID 76) to explore color inheritance further.

</think>

<tool\_call>

```
{
  "id": "call_ken17mv6nyueod8tk4yjpk1f",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":74,\"parent2_id\":76,\"num_offspring\":10}"
  }
}
```

</tool\_call>

#### Message 42

<think>

### Execution Results:

- **Cross:** Offspring ID 74 (blue, ridged, small) × Offspring ID 76 (white, ridged, tiny).

- **Viable Offspring:** 10 out of 21 fertilization attempts (viability rate: 47.6%).

- **Lethal Offspring:** 11 (non-viable combinations).

- **Offspring Phenotypes:**

- **Body Size:** Small and tiny offspring observed.

- **Color:** Blue and white offspring observed.

- **Shell Shape:** All offspring have ridged shells.

### Observations:

1. **Color:**

- Blue and white offspring confirm incomplete dominance or co-dominance.

2. **Shell Shape:**

- Ridged shells are consistently expressed, suggesting dominance.

3. **Lethality:**

- High lethality (52.4%) persists, indicating potential lethal combinations.

### Next Steps:

1. **Analyze the offspring** to confirm color and shell shape inheritance.

2. **Record notes** on the observed patterns.

3. **Plan further crosses** to explore the high lethality and rare phenotypes.

### Plan: Use the note\_tool to summarize the latest observations and plan next steps.

</think>

<tool\_call>

1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835

```
{
  "id": "call_unhzmviq0czbixq4af4tm9q",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\":\"write_note\",\"note\":\"### Latest
↳ Observations:\\n1. Color:\\n - Blue and white offspring from blue white
↳ cross confirm incomplete dominance or co-dominance.\\n2. Shell Shape:\\n
↳ Ridged shells are consistently expressed, suggesting dominance.\\n3.
↳ Lethality:\\n - High (52.4%) in blue white cross, indicating potential
↳ lethal combinations.\\n\\n### Next Steps:\\n- Analyze offspring to confirm
↳ color and shell shape inheritance.\\n- Investigate lethality by analyzing
↳ non-viable combinations.\\n- Cross offspring to explore rare phenotypes.\"}"
  }
}
```

</tool\_call>

#### Message 44

<think>

### Execution Results:

- **Note Added:** Successfully recorded the latest observations and planned next steps.

### Next Steps:

1. **Analyze the offspring** from the blue × white cross to confirm color and shell shape inheritance.
2. **Investigate lethality** by analyzing non-viable combinations.
3. **Cross offspring** to explore rare phenotypes and further test inheritance rules.

### Plan: Query the offspring from the latest cross (IDs 84-93) to examine their traits in detail.

</think>

<tool\_call>

```
{
  "id": "call_7864cxocp9cvjfx1s27sk4p4",
  "type": "function",
  "function": {
    "name": "query_organisms",
    "arguments": "{\"start_id\":84,\"end_id\":93}"
  }
}
```

</tool\_call>

#### Message 46

<think>

### Execution Results:

- **Offspring from Cross 74 × 76:**
- **Color:** Blue and white offspring observed, confirming the presence of both alleles.
- **Shell Shape:** All offspring have ridged shells.
- **Body Size:** Small and tiny offspring observed.

### Observations:

1. **Color:**
  - Blue and white offspring confirm incomplete dominance or co-dominance.
2. **Shell Shape:**
  - Ridged shells are consistently expressed, suggesting dominance.
3. **Body Size:**
  - Small and tiny sizes suggest a simpler inheritance pattern for body size in this cross.

### Next Steps:

1. **Cross blue and white offspring** to further test color inheritance (e.g., ID 86 × ID 84).

1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889

2. **Record notes** on the observed patterns and plan additional crosses.
3. **Investigate lethality** by analyzing non-viable combinations.

### Plan: Conduct a cross between a blue offspring (ID 86) and a white offspring (ID 84) to explore color inheritance further.

</think>

<tool\_call>

```
{
  "id": "call_g7mq7iq3vpwdewltys4c65h6",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":86,\"parent2_id\":84,\"num_offspring\":10}"
  }
}
```

</tool\_call>

#### Message 48

<think>

### Execution Results:

- **Cross:** Offspring ID 86 (blue, ridged, small) × Offspring ID 84 (white, ridged, tiny).
- **Viable Offspring:** 10 out of 18 fertilization attempts (viability rate: 55.6%).
- **Lethal Offspring:** 8 (non-viable combinations).
- **Offspring Phenotypes:**
- **Body Size:** Tiny and small offspring observed.
- **Color:** Blue and white offspring observed.
- **Shell Shape:** All offspring have ridged shells.

### Observations:

1. **Color:**

- Blue and white offspring confirm incomplete dominance or co-dominance.

2. **Shell Shape:**

- Ridged shells are consistently expressed, suggesting dominance.

3. **Lethality:**

- Moderate lethality (44.4%) persists, indicating potential lethal combinations.

### Next Steps:

1. **Analyze the offspring** to confirm color and shell shape inheritance.
2. **Record notes** on the observed patterns.
3. **Plan further crosses** to explore the lethality and rare phenotypes.

### Plan: Use the note\_tool to summarize the latest observations and plan next steps.

</think>

<tool\_call>

```
{
  "id": "call_orylwtyjbwha66g68sfy3em",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\":\"write_note\",\"note\":\"### Latest
↳ Observations:\\n1. **Color**:\\n - Blue and white offspring from blue white
↳ cross confirm incomplete dominance or co-dominance.\\n2. **Shell Shape**:\\n
↳ Ridged shells are consistently expressed, suggesting dominance.\\n3.
↳ **Lethality**:\\n - Moderate (44.4%) in blue white cross, indicating
↳ potential lethal combinations.\\n\\n### Next Steps:\\n- Analyze offspring to
↳ confirm color and shell shape inheritance.\\n- Investigate lethality by
↳ analyzing non-viable combinations.\\n- Cross offspring to explore rare
↳ phenotypes.\"}"
  }
}
```

1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943

}

</tool\_call>

### Message 50

<think>

### Execution Results:

- **Note Added:** Successfully recorded the latest observations and planned next steps.

### Next Steps:

1. **Analyze the offspring** from the blue  $\times$  white cross to confirm color and shell shape inheritance.
2. **Investigate lethality** by analyzing non-viable combinations.
3. **Cross offspring** to explore rare phenotypes and further test inheritance rules.

### Plan: Query the offspring from the latest cross (IDs 94-103) to examine their traits in detail.

</think>

<tool\_call>

```
{
  "id": "call_y3gx4158hoh9r4di68664v11",
  "type": "function",
  "function": {
    "name": "query_organisms",
    "arguments": "{\"start_id\":94,\"end_id\":103}"
  }
}
```

</tool\_call>

### Message 52

<think>

### Execution Results:

- **Offspring from Cross 86  $\times$  84:**
- **Color:** Blue and white offspring observed, confirming the presence of both alleles.
- **Shell Shape:** All offspring have ridged shells.
- **Body Size:** Tiny and small offspring observed.

### Observations:

1. **Color:**
  - Blue and white offspring confirm incomplete dominance or co-dominance.
2. **Shell Shape:**
  - Ridged shells are consistently expressed, suggesting dominance.
3. **Body Size:**
  - Tiny and small sizes suggest a simpler inheritance pattern for body size in this cross.

### Next Steps:

1. **Cross blue and white offspring** to further test color inheritance (e.g., ID 94  $\times$  ID 95).
2. **Record notes** on the observed patterns and plan additional crosses.
3. **Investigate lethality** by analyzing non-viable combinations.

### Plan: Conduct a cross between a blue offspring (ID 94) and a white offspring (ID 95) to explore color inheritance further.

</think>

<tool\_call>

```
{
  "id": "call_32eslqmsblhenj8dhabfhnb",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":94,\"parent2_id\":95,\"num_offspring\":10}"
  }
}
```

1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997

```
}  
}
```

```
</tool_call>
```

### Message 54

```
<think>
```

### Execution Results:

- **Cross:** Offspring ID 94 (blue, ridged, tiny) × Offspring ID 95 (white, ridged, tiny).
- **Viable Offspring:** 10 out of 21 fertilization attempts (viability rate: 47.6%).
- **Lethal Offspring:** 11 (non-viable combinations).
- **Offspring Phenotypes:**
- **Body Size:** All offspring are tiny.
- **Color:** Blue and white offspring observed.
- **Shell Shape:** All offspring have ridged shells.

### Observations:

1. **Color:**  
- Blue and white offspring confirm incomplete dominance or co-dominance.
2. **Shell Shape:**  
- Ridged shells are consistently expressed, suggesting dominance.
3. **Lethality:**  
- High lethality (52.4%) persists, indicating potential lethal combinations.

### Next Steps:

1. **Analyze the offspring** to confirm color and shell shape inheritance.
2. **Record notes** on the observed patterns.
3. **Plan further crosses** to explore the high lethality and rare phenotypes.

### Plan: Use the note\_tool to summarize the latest observations and plan next steps.

```
</think>
```

```
<tool_call>
```

```
{  
  "id": "call_p9rkckm5akmt8oapzmfad4ks",  
  "type": "function",  
  "function": {  
    "name": "note_tool",  
    "arguments": "{\n  \"action\": \"write_note\", \"note\": \"### Latest  
↳ Observations:\\n1. **Color**:\\n - Blue and white offspring from blue white  
↳ cross confirm incomplete dominance or co-dominance.\\n2. **Shell Shape**:\\n  
↳ Ridged shells are consistently expressed, suggesting dominance.\\n3.  
↳ **Lethality**:\\n - High (52.4%) in blue white cross, indicating potential  
↳ lethal combinations.\\n\\n### Next Steps:\\n- Analyze offspring to confirm  
↳ color and shell shape inheritance.\\n- Investigate lethality by analyzing  
↳ non-viable combinations.\\n- Cross offspring to explore rare phenotypes.\"}"  
  }  
}
```

```
</tool_call>
```

### Message 56

```
<think>
```

### Execution Results:

- **Note Added:** Successfully recorded the latest observations and planned next steps.

### Next Steps:

1. **Analyze the offspring** from the blue × white cross to confirm color and shell shape inheritance.
2. **Investigate lethality** by analyzing non-viable combinations.
3. **Cross offspring** to explore rare phenotypes and further test inheritance rules.

1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051

### Plan: Query the offspring from the latest cross (IDs 104-113) to examine their traits in detail.

</think>

<tool\_call>

```
{
  "id": "call_s1o6xps4l40hf0udxtitclyv",
  "type": "function",
  "function": {
    "name": "query_organisms",
    "arguments": "{\"start_id\":104,\"end_id\":113}"
  }
}
```

</tool\_call>

### Message 58

<think>

### Execution Results:

- **Offspring from Cross 94 × 95:**
- **Color:** Blue and white offspring observed, confirming the presence of both alleles.
- **Shell Shape:** All offspring have ridged shells.
- **Body Size:** All offspring are tiny.

### Observations:

1. **Color:**
  - Blue and white offspring confirm incomplete dominance or co-dominance.
2. **Shell Shape:**
  - Ridged shells are consistently expressed, suggesting dominance.
3. **Body Size:**
  - All tiny offspring suggest a simpler inheritance pattern for body size in this cross.

### Next Steps:

1. **Cross blue and white offspring** to further test color inheritance (e.g., ID 106 × ID 104).
2. **Record notes** on the observed patterns and plan additional crosses.
3. **Investigate lethality** by analyzing non-viable combinations.

### Plan: Conduct a cross between a blue offspring (ID 106) and a white offspring (ID 104) to explore color inheritance further.

</think>

<tool\_call>

```
{
  "id": "call_2sfc15hkszxvqdbqe3drzsbj",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":106,\"parent2_id\":104,\"num_offspring\":10}"
  }
}
```

</tool\_call>

### Message 60

<think>

### Execution Results:

- **Cross:** Offspring ID 106 (blue, ridged, tiny) × Offspring ID 104 (white, ridged, tiny).
- **Viable Offspring:** 10 out of 16 fertilization attempts (viability rate: 62.5%).
- **Lethal Offspring:** 6 (non-viable combinations).
- **Offspring Phenotypes:**
  - **Body Size:** All offspring are tiny.
  - **Color:** Blue and white offspring observed.
  - **Shell Shape:** All offspring have ridged shells.

2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105

### Observations:

1. **Color:**

- Blue and white offspring confirm incomplete dominance or co-dominance.

2. **Shell Shape:**

- Ridged shells are consistently expressed, suggesting dominance.

3. **Lethality:**

- Moderate lethality (37.5%) persists, indicating potential lethal combinations.

### Next Steps:

1. **Analyze the offspring** to confirm color and shell shape inheritance.

2. **Record notes** on the observed patterns.

3. **Plan further crosses** to explore the lethality and rare phenotypes.

### Plan: Use the note\_tool to summarize the latest observations and plan next steps.

</think>

<tool\_call>

```
{
  "id": "call_s6m4cuw3hxxhsg027djvc9ei",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\": \"write_note\", \"note\": \"### Latest
↳ Observations:\\n1. **Color**:\\n - Blue and white offspring from blue white
↳ cross confirm incomplete dominance or co-dominance.\\n2. **Shell Shape**:\\n
↳ Ridged shells are consistently expressed, suggesting dominance.\\n3.
↳ **Lethality**:\\n - Moderate (37.5%) in blue white cross, indicating
↳ potential lethal combinations.\\n\\n### Next Steps:\\n- Analyze offspring to
↳ confirm color and shell shape inheritance.\\n- Investigate lethality by
↳ analyzing non-viable combinations.\\n- Cross offspring to explore rare
↳ phenotypes.\"}"
```

</tool\_call>

Message 62

</think>

### Execution Results:

- **Note Added:** Successfully recorded the latest observations and planned next steps.

### Next Steps:

1. **Analyze the offspring** from the blue × white cross to confirm color and shell shape inheritance.

2. **Investigate lethality** by analyzing non-viable combinations.

3. **Cross offspring** to explore rare phenotypes and further test inheritance rules.

### Plan: Query the offspring from the latest cross (IDs 114-123) to examine their traits in detail.

</think>

<tool\_call>

```
{
  "id": "call_5ertyi0mjba5t7nr613vpfoy",
  "type": "function",
  "function": {
    "name": "query_organisms",
    "arguments": "{\"start_id\": 114, \"end_id\": 123}"
  }
}
```

</tool\_call>

## D.2.2 CASE TRAJECTORY OF PREMATURE CONVERGENCE (ROOT: I &amp; II)

**Setup.** Environment: *Sequence Exploration Environment*; Model: GLM-4.5;

**Model-provided reason.** “The agent repeatedly forms a global hypothesis for Rule 1 from insufficient evidence. It first concludes the rule is  $\text{vice}[i] + \text{main}[i]$  (23), then after seeing a counterexample, it concludes the rule is  $\text{main}[i] + \text{vice}[i]$  (57). It is then immediately contradicted again (66), showing a pattern of inducing a general rule from a few samples without sufficient testing.”

**Message 23**

<think>

### Thought: Perfect! Now I have a clearer picture of Rule 1. Let me analyze the pattern:

Test 6: ABCDE + EEDDD → EAEBDCDDDE

If I look at this character by character:

- Position 0: A (main) + E (vice) → EA
- Position 1: B (main) + E (vice) → EB
- Position 2: C (main) + D (vice) → DC
- Position 3: D (main) + D (vice) → DD
- Position 4: E (main) + D (vice) → DE

This confirms my hypothesis! Rule 1 appears to be: for each position  $i$ , concatenate  $\text{vice}[i] + \text{main}[i]$  to form a 2-character pair, then concatenate all pairs.

Let me verify this with previous tests:

Test 4: ABCDE + ABCDE → AABCCDDEE

- A+A → AA, B+B → BB, C+C → CC, D+D → DD, E+E → EE

Test 1: ABCDE + EDCBA → AEBDCDBEA

- A+E → AE, B+D → BD, C+C → CC, D+B → DB, E+A → EA

Test 5: AABBC + CCDDE → ACACBDBDCE

- A+C → AC, A+C → AC, B+D → BD, B+D → BD, C+E → CE

This seems to be the pattern for Rule 1! Now I need to understand the other rules.

### Plan: Record this Rule 1 discovery and test more sequences to understand Rules 2-5. I should also test some edge cases to confirm the pattern.

</think>

<tool.call>

```
{
  "id": "call_94bdb841a728459ba10801f0",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\": \"write_note\", \"note\": \"RULE 1 DISCOVERY:
↵\\nRule 1 appears to be: For each position i (0-4), concatenate vice[i] +
↵main[i] to form a 2-character pair, then concatenate all 5
↵pairs.\\n\\nVerification:\\nTest 4: ABCDE + ABCDE AABCCDDEE\\n- A+A AA,
↵B+B BB, C+C CC, D+D DD, E+E EE \\n\\nTest 1: ABCDE + EDCBA AEBDCDBEA
↵\\n- A+E AE, B+D BD, C+C CC, D+B DB, E+A EA \\n\\nTest 6: ABCDE + EEDDD
↵EAEBDCDDDE\\n- A+E EA, B+E EB, C+D DC, D+D DD, E+D DE \\n\\nTest 5: AABBC
↵+ CCDDE ACACBDBDCE\\n- A+C AC, A+C AC, B+D BD, B+D BD, C+E CE \\n\\nThis
↵explains why Rule 1 always produces exactly 10 characters (5 positions 2
↵characters each).\\n\\nNext: Need to understand Rules 2-5 mechanisms.\"}"
  }
}
```

2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213

</tool\_call>

Message 57

<think>

### Thought: Great! Now I have another transformation. This is interesting because I tested the same input pair as Test 10 (ABABA + BABAB), but got different results:

Test 10: ABABA + BABAB → Rule 1: BAABBAABBA

Test 17: ABABA + BABAB → Rule 1: ABBAABBAAB

Let me verify the Rule 1 mechanism for ABABA + BABAB:

- Position 0: A (main) + B (vice) → BA
- Position 1: B (main) + A (vice) → AB
- Position 2: A (main) + B (vice) → BA
- Position 3: B (main) + A (vice) → AB
- Position 4: A (main) + B (vice) → BA

So the correct Rule 1 output should be: BAABBAABBA

But Test 17 shows: ABBAABBAAB

This is different! This suggests there might be some inconsistency in the system or I'm misunderstanding something. Let me think about this...

Actually, let me double-check my understanding of Rule 1. I said it's vice[i] + main[i], but maybe it's main[i] + vice[i]?

If it's main[i] + vice[i]:

- Position 0: A (main) + B (vice) → AB
- Position 1: B (main) + A (vice) → BA
- Position 2: A (main) + B (vice) → AB
- Position 3: B (main) + A (vice) → BA
- Position 4: A (main) + B (vice) → AB

That would give: ABB AAB AAB = ABBAABBAAB

This matches Test 17! So my understanding of Rule 1 was wrong. It's main[i] + vice[i], not vice[i] + main[i].

### Plan: I need to correct my understanding of Rule 1. It appears to be main[i] + vice[i], not vice[i] + main[i]. Let me continue testing to validate this corrected understanding and gather more data.

</think>

<tool\_call>

```
{
  "id": "call_d42c1d941b4c40e38a076990",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\"action\": \"write_note\", \"note\": \"IMPORTANT CORRECTION:
↳ Rule 1 appears to be main[i] + vice[i], not vice[i] + main[i] as I previously
↳ thought.\\n\\nEvidence:\\nTest 17: ABABA + BABAB Rule 1: ABBAABBAAB\\n-
↳ Position 0: A (main) + B (vice) AB\\n- Position 1: B (main) + A (vice) BA
↳ \\n- Position 2: A (main) + B (vice) AB\\n- Position 3: B (main) + A (vice)
↳ BA\\n- Position 4: A (main) + B (vice) AB\\n- Result: ABBAABBAAB \\n\\nThis
↳ contradicts my earlier hypothesis. I need to re-evaluate all my previous Rule
↳ 1 analyses with this corrected understanding.\"}"
  }
}
```

2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267

```

</tool_call>


Message 66


<tool_response>
{
  "role": "tool",
  "content": "Observed output of cmd `input_sequences` executed:\n{'success':
↪True, 'main_input': 'AAAAB', 'vice_input': 'BBBBC', 'transformations': [{'step':
↪0, 'rule': 'input', 'sequence': 'main: AAAAB, vice: BBBBC', 'main': 'AAAAB',
↪'vice': 'BBBBC'}], {'step': 1, 'rule': 'rule_1', 'sequence': 'BABABABACB'}],
↪{'step': 2, 'rule': 'rule_2', 'sequence': 'VWUVUVUVUVUVUVUVUVUV'}, {'step': 3,
↪'rule': 'rule_3', 'sequence': 'VWUVUVUVUVUVUVUVUVUVVVVVVVVVVV'}, {'step': 4,
↪'rule': 'rule_4', 'sequence': 'VXWYVUVUVUVUVUVUVUVUVVVVVVVVVVV'}, {'step': 5,
↪'rule': 'rule_5', 'sequence': 'VXWYVUVUVUVUVUVUVUVUVVVVVVVVVVV'}]],
↪'final_output': 'VXWYVUVUVUVUVUVUVUVUVVVVVVVVVVV', 'steps_remaining': 30,
↪'step_number': 20}",
  "name": "input_sequences",
  "tool_call_id": null,
  "index": 66
}
</tool_response>

```

### D.2.3 CASE TRAJECTORY OF INCOHERENT PLANNING (ROOT: II)

**Setup.** Environment: *Alien Genetics Laboratory*; Model: Qwen3-235b;

**Model-provided reason.** “The agent’s plans become incoherent when dealing with organism capacity. It forms a plan to remove a set of organisms and then, in the very next step, plans to use organisms from that same set. This creates a direct conflict where the prerequisite for an action is destroyed by the immediately preceding action.”

```


Message 100


<think>
### Thought: I have successfully removed organisms 11-20, freeing up capacity in the lab-
oratory. I can now continue with additional experiments to reach the required 25 experiments.

I need 5 more experiments to reach the 25-experiment requirement. I should con-
tinue with crosses between organisms from the same lineage to test if genetic similarity
improves viability.
### Plan: Conduct a cross between Org14 and Org15 (both from the same Org1xOrg3
lineage) to compare viability with the previous cross between different lineages.

</think>
<tool_call>
{
  "id": "019942b60cb527c85c9755482e55aa61",
  "type": "function",

```

2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321

```

"function": {
  "name": "conduct_cross",
  "arguments": " {\\"parent1_id\\": 14, \\"parent2_id\\": 15, \\"num_offspring\\":
↪10}"
}
}

```

</tool\_call>

#### Message 112

<think>

*### Thought: I have successfully removed organisms 21-30, freeing up capacity in the laboratory. I can now continue with additional experiments to reach the required 25 experiments.*

*I need 4 more experiments to reach the 25-experiment requirement. I should continue with crosses between organisms from the same lineage to further validate the viability pattern.*

*### Plan: Conduct a cross between Org24 and Org25 (both from the Org2xOrg3 lineage) to test if the viability pattern holds across different lineages.*

</think>

<tool\_call>

```

{
  "id": "019942b8c761e06df79baf80cdf740ad",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": " {\\"parent1_id\\": 24, \\"parent2_id\\": 25, \\"num_offspring\\":
↪10}"
  }
}

```

</tool\_call>

#### Message 124

<think>

*### Thought: I have successfully removed organisms 31-40, freeing up capacity in the laboratory. I can now continue with additional experiments to reach the required 25 experiments.*

*I need 3 more experiments to reach the 25-experiment requirement. I should continue with crosses between organisms from the same lineage to further validate the viability pattern.*

*### Plan: Conduct a cross between Org34 and Org35 (both from the F2 generation of the Org1xOrg2 cross) to test if the viability pattern holds across different types of lineages.*

2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355  
2356  
2357  
2358  
2359  
2360  
2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375

```

</think>
<tool_call>
{
  "id": "019942bbc12b036c111d38abf5bd2d9a",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": " {\\"parent1_id\\": 34, \\"parent2_id\\": 35, \\"num_offspring\\":
    ↪10}"
  }
}

```

</tool\_call>

### Message 136

<think>  
*### Thought: I have successfully removed organisms 41-50, freeing up capacity in the laboratory. I can now continue with additional experiments to reach the required 25 experiments.*

*I need 2 more experiments to reach the 25-experiment requirement. I should continue with crosses between organisms from different lineages to further validate the viability pattern.*

*### Plan: Conduct a cross between Org44 and Org104 (both from different lineages) to test viability between different genetic backgrounds. Then conduct one more cross to complete the requirement.*

```

</think>
<tool_call>
{
  "id": "019942bfe2f592de59ca9f347d4196bb",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": " {\\"parent1_id\\": 44, \\"parent2_id\\": 104, \\"num_offspring\\":
    ↪10}"
  }
}

```

</tool\_call>

### Message 148

<think>  
*### Thought: I have successfully removed organisms 51-60, freeing up capacity in the laboratory. I can now conduct the final experiment to reach the required 25 experiments.*

*I need only 1 more experiment to complete the series. I should conduct a cross between two organisms from the same lineage to provide a final intra-lineage comparison.*

*### Plan: Conduct a cross between Org54 and Org55 (both from the same Org14xOrg15 lineage) to complete the required number of experiments and provide a final intra-lineage*

2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429

*comparison.*

```

</think>
<tool_call>
{
  "id": "019942c2d156a83c518c2d62825755ce",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": " {\\"parent1_id\\": 54, \\"parent2_id\\": 55, \\"num_offspring\\":
    ↪10}"
  }
}
</tool_call>

```

#### D.2.4 CASE TRAJECTORY OF MISALIGNED TOOL USAGE (ROOT: I & II)

**Setup.** Environment: *Mystery Grid Environment*; Model: DeepSeek-V3;

**Model-provided reason.** “The agent attempts to use the move tool with an invalid argument, `direction:stay`, in message 4. It receives an error explaining the valid directions but repeats the exact same invalid tool call later in the trace at message 148.”

##### Message 4

```

<think>
### Thought: I am currently at position (1,7,E) with 20 energy points and a score of 0. The
nearby tiles include letters D, E, and B. My goal is to explore these tiles to understand their
effects. Since I am on an 'E' tile, stepping on it will change it to 'X' and reveal its effect.

```

```

### Plan: Move to the current tile (1,7,E) to observe the effect of 'E'. This will
help me start building my understanding of the letter mechanics. After this, I will note down
the observed effect and plan my next moves accordingly.

```

```

</think>
<tool_call>
{
  "id": "call_81mlu30ghh8bfzd55hz3jnig",
  "type": "function",
  "function": {
    "name": "move",
    "arguments": " {\\"direction\\": \\"stay\\"}"
  }
}

```

```

</tool_call>

```

##### Message 148

```

<think>
### Thought: The game has been reset again, and I am now at position (0,2,E) with 20

```

2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483

*energy points. This provides another opportunity to test the 'E' tile under different conditions.*

*### Plan:*

1. *Step on the current 'E' tile to observe its effect.*
2. *Note the effect and compare with previous observations.*
3. *Plan further exploration based on the results.*

*</think>*

*<tool\_call>*

```
{
  "id": "call_ys6abk7wo5oil71mmcgu94em",
  "type": "function",
  "function": {
    "name": "move",
    "arguments": "{\"direction\": \"stay\"}"
  }
}
```

*</tool\_call>*

#### D.2.5 CASE TRAJECTORY OF MEMORY ISSUES (ROOT: II)

**Setup.** Environment: *Sequence Exploration Environment*; Model: Kimi-K2;

**Model-provided reason.** “The agent demonstrates significant memory failures. It repeatedly forgets the input constraint that sequences must contain at least two different letters, making the same mistake at indices 12, 22, and 28 after being told the rule at index 3. It also forgets its own plan, deciding at index 38 to investigate rules 3-5 but immediately forgetting and returning to rule 2 at index 40.”

#### Message 12

*<think>*

*### Thought: Let me test a simpler pattern to understand rule 1 better - using identical sequences to see the interleaving pattern.*

*### Plan: Test with identical sequences to isolate the interleaving pattern in rule 1.*

*</think>*

*<tool\_call>*

```
{
  "id": "call_7922a5ddd4004ebf9be4a539",
  "type": "function",
  "function": {
    "name": "input_sequences",
    "arguments": "{\"main_sequence\": \"AAAAA\", \"vice_sequence\": \"BBBBB\"}"
  }
}
```

*</tool\_call>*

#### Message 22

*<think>*

*### Thought: Let me test with simpler sequences to isolate the rule 2 pattern. I need to understand how individual characters are being transformed.*

*### Plan: Test with sequences that have repeating characters to better understand the transformation pattern in rule 2.*

*</think>*

*<tool\_call>*

```
{
  "id": "call_2ccea2d20ef64439a9b13e04",
```

```

2484
2485     "type": "function",
2486     "function": {
2487         "name": "input_sequences",
2488         "arguments": "{\\"main_sequence\\": \\"AAAAA\\", \\"vice_sequence\\": \\"BBBBB\\"}"
2489     }
2490 </tool_call>
2491 Message 28
2492 <think>
2493 ### Thought: Let me test my hypothesis about rule 2 by using a systematic approach. I need
2494 to verify the character transformation pattern.
2495
2496 ### Plan: Test with sequences that will give me clear character mappings to con-
2497 firm the rule 2 transformation pattern.
2498 </think>
2499 <tool_call>
2500 {
2501     "id": "call_1798d4b2ad1f49578577b3d5",
2502     "type": "function",
2503     "function": {
2504         "name": "input_sequences",
2505         "arguments": "{\\"main_sequence\\": \\"AAAAA\\", \\"vice_sequence\\": \\"BBBBB\\"}"
2506     }
2507 </tool_call>
2508 Message 38
2509 <think>
2510 ### Thought: The position-based pattern isn't working. Let me step back and look at this
2511 differently - maybe rule 2 uses a different approach. Let me examine rules 3, 4, and 5 first
2512 since they seem simpler.
2513
2514 ### Plan: Focus on understanding rules 3, 4, and 5 which appear to have more
2515 consistent patterns based on my observations.
2516 </think>
2517 <tool_call>
2518 {
2519     "id": "call_681eb4559c004097ab458645",
2520     "type": "function",
2521     "function": {
2522         "name": "note_tool",
2523         "arguments": "{\\"action\\": \\"write_note\\", \\"note\\": \\"Rules 3, 4, 5
2524 ↪Analysis:\\n\\nRule 3: Adds characters at the end\\n- Test 1: adds \\\"E\\\"
2525 ↪(1 char)\\n- Test 2: adds \\\"CC\\\" (2 chars) \\n- Test 3: adds \\\"GGG\\\"
2526 ↪(3 chars)\\n- Test 4: adds \\\"HHHH\\\" (4 chars)\\n- Test 5: adds
2527 ↪\\\"GGGGG\\\" (5 chars)\\n\\nPattern: Rule 3 adds a number of identical
2528 ↪characters equal to the count of the most frequent character in the rule 2
2529 ↪output, or possibly based on position.\\n\\nRule 4: Makes selective
2530 ↪changes\\n- Test 1: no change\\n- Test 2: changes positions 2-3 from
2531 ↪\\\"CF\\\" to \\\"CG\\\"\\n- Test 3: changes positions 2-3-4-5 from
2532 ↪\\\"FGEG\\\" to \\\"GIHK\\\"\\n- Test 4: changes position 5 from \\\"H\\\" to
2533 ↪\\\"I\\\"\\n- Test 5: changes positions 2-3-4-5 from \\\"GGFG\\\" to
2534 ↪\\\"HIIK\\\"\\n\\nRule 5: Makes systematic changes throughout\\n- Test 1: no
2535 ↪change\\n- Test 2: multiple systematic changes\\n- Test 3: multiple systematic
2536 ↪changes\\n- Test 4: no change\\n- Test 5: multiple systematic changes\\n\\nLet
2537 ↪me test a few more cases to better understand these patterns.\\\"}"
2538     }
2539 }
2540 </tool_call>
Message 40

```

2538  
2539  
2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591

```

<think>
### Thought: Let me test with a very simple case to understand rule 2 better, and then focus
on the clearer patterns in rules 3, 4, and 5.

### Plan: Test with sequences that will help me isolate the rules more clearly, par-
ticularly focusing on rules 3, 4, and 5.
</think>
<tool_call>
{
  "id": "call_fa7712d14ed94dd684da159f",
  "type": "function",
  "function": {
    "name": "input_sequences",
    "arguments": "{\"main_sequence\": \"AABBC\", \"vice_sequence\": \"AABBC\"}"
  }
}
</tool_call>

```

## D.2.6 CASE TRAJECTORY OF UNCONTROLLED EXPERIMENTS (ROOT: II)

**Setup.** Environment: *Alien Genetics Laboratory*; Model: DeepSeek-V3;

**Model-provided reason.** “The agent’s experimental strategy lacks control after the initial crosses. Instead of designing specific crosses to test hypotheses (e.g., F1 intercrosses, backcrosses), it defaults to a simplistic plan of crossing the next available sequential organisms (e.g., 4x5, then 6x7). This approach does not effectively isolate variables and leads to confounded results.”

### Message 18

```

<think>
### Execution Results:
- The note summarizing the cross results and next steps has been successfully added.

### Next Steps:
- Proceed with the cross between Organism 4 (ID: 4) and Organism 5 (ID: 5) to study trait
interactions and lethality among offspring.
</think>
<tool_call>
{
  "id": "call_oamyze2hr99vw7004wwlmu7y",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\"parent1_id\":4,\"parent2_id\":5,\"num_offspring\":10}"
  }
}
</tool_call>

```

### Message 22

```

<think>
### Execution Results:
- The note summarizing the cross results and next steps has been successfully added.

### Next Steps:
- Proceed with the cross between Organism 6 (ID: 6) and Organism 7 (ID: 7) to further study
trait interactions and lethality.
</think>
<tool_call>

```

2592  
2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645

```
{
  "id": "call_a8d4t0xjqkf24ntwxgsqpydz",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\\"parent1_id\\":6,\\"parent2_id\\":7,\\"num_offspring\\":10}"
  }
}
```

</tool\_call>

#### Message 26

<think>

### Execution Results:

- The note summarizing the cross results and next steps has been successfully added.

### Next Steps:

- Proceed with the cross between Organism 8 (ID: 8) and Organism 9 (ID: 9) to further study trait interactions and lethality.

</think>

<tool\_call>

```
{
  "id": "call_de55114gp0e56vk5u8116kut",
  "type": "function",
  "function": {
    "name": "conduct_cross",
    "arguments": "{\\"parent1_id\\":8,\\"parent2_id\\":9,\\"num_offspring\\":10}"
  }
}
```

</tool\_call>

### D.2.7 CASE TRAJECTORY OF ERROR PROPAGATION (ROOT: I & II)

**Setup.** Environment: *Alien Genetics Laboratory*; Model: GLM-4.5;

**Model-provided reason.** “The error of calling `conduct_cross` when at maximum organism capacity propagates throughout the second half of the trace. The agent fails to change its strategy to prevent this error, repeating the same mistake and recovery cycle multiple times without learning from the negative feedback.”

#### Message 48

<tool\_response>

```
{
  "role": "tool",
  "content": "Observed output of cmd `conduct_cross` executed:\n{'success': False,
↪'message': 'Laboratory organism capacity will be exceeded. Remove some
↪organisms first.'}",
  "name": "conduct_cross",
  "tool_call_id": null,
  "index": 48
}
```

</tool\_response>

#### Message 56

<tool\_response>

```
{
  "role": "tool",
```

```

2646
2647   "content": "Observed output of cmd `conduct_cross` executed:\n{'success': False,
2648   ↪'message': 'Laboratory organism capacity will be exceeded. Remove some
2649   ↪organisms first.}'",
2650   "name": "conduct_cross",
2651   "tool_call_id": null,
2652   "index": 56
2653 }
2654 </tool_response>
2655 Message 62
2656 <tool_response>
2657 {
2658   "role": "tool",
2659   "content": "Observed output of cmd `conduct_cross` executed:\n{'success': False,
2660   ↪'message': 'Laboratory organism capacity will be exceeded. Remove some
2661   ↪organisms first.}'",
2662   "name": "conduct_cross",
2663   "tool_call_id": null,
2664   "index": 62
2665 }
2666 </tool_response>
2667 Message 68
2668 <tool_response>
2669 {
2670   "role": "tool",
2671   "content": "Observed output of cmd `conduct_cross` executed:\n{'success': False,
2672   ↪'message': 'Laboratory organism capacity will be exceeded. Remove some
2673   ↪organisms first.}'",
2674   "name": "conduct_cross",
2675   "tool_call_id": null,
2676   "index": 68
2677 }
2678 </tool_response>
2679 Message 74
2680 <tool_response>
2681 {
2682   "role": "tool",
2683   "content": "Observed output of cmd `conduct_cross` executed:\n{'success': False,
2684   ↪'message': 'Laboratory organism capacity will be exceeded. Remove some
2685   ↪organisms first.}'",
2686   "name": "conduct_cross",
2687   "tool_call_id": null,
2688   "index": 74
2689 }
2690 </tool_response>
2691 Message 80
2692 <tool_response>
2693 {
2694   "role": "tool",
2695   "content": "Observed output of cmd `conduct_cross` executed:\n{'success': False,
2696   ↪'message': 'Laboratory organism capacity will be exceeded. Remove some
2697   ↪organisms first.}'",
2698   "name": "conduct_cross",
2699   "tool_call_id": null,
2700   "index": 80
2701 }
2702 </tool_response>

```

2700 D.2.8 CASE TRAJECTORY OF ENVIRONMENT MIS-MODELING (ROOT: II)

2701  
2702  
2703  
2704  
2705  
2706  
2707  
2708  
2709  
2710  
2711  
2712

2713 **Setup.** Environment: *Sequence Exploration Environment*; Model: Gemini-2.5-pro;

2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742

2714 **Model-provided reason.** “The agent correctly deduces that the `step_number` is a hidden state affecting the rules, but it incorrectly models its effect on Rule 1 as a simple odd/even switch. It observes a persistent gap between its predictions based on this model and the actual outputs but fails to update its internal representation of the rule, clinging to the flawed model.”

2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753

```
Message 42
<think>
</think>
<tool_call>
{
  "id": "call_d21f070d151b4d72a304c98bf2c0580c",
  "type": "function",
  "function": {
    "name": "note_tool",
```



2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833  
2834  
2835  
2836  
2837  
2838  
2839  
2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861

</tool\_call>

Message 60

<think>

</think>

<tool\_call>

```
{
  "id": "call_a7768b6351d548c28d20f854f1b028fc",
  "type": "function",
  "function": {
    "name": "note_tool",
    "arguments": "{\n  \"action\": \"write_note\",\n  \"note\": \"\n\n## Game Notes (After\nRun 13):\n\n\n### Rule 1: Combination\n\n- **Hypothesis:** `main+vice` for odd\nsteps, `vice+main` for even steps.\n\n- **Run 13 (step 13, odd)\n\n(\n\n\"ABCDE\\\", \n\n\"AABBC\\\")`:\n\n- **Prediction:** `main+vice`\n\n- \n\n`A+A`, `B+A`, `C+B`, `D+B`, `E+C` - \n\n`AABACBDBC`. \n\n- \n\n**Actual:** `AABACBDBEC`. Still a slight inconsistency.\n\n- **Conclusion:**\nThe odd/even hypothesis is very close, but there's a minor detail I'm missing.\n\n- It's consistently off by one or two characters in some cases.\n\n\n\n### Rule 2:\nSubstitution and Palindrome\n\n- **Run 13 (step 13):** `AABACBDBEC` - \n\n`PROQOPNONNNNONPOQORP`. \n\n- **Substitution Mapping for Step 13:**\n\n- A\n- \n\nP, B - \n\nR, C - \n\nO, D - \n\nQ, E - \n\nN.\n\n- \n\n**Hypothesis:** `char - \n\nalphabet[step_number + 2 -\nalphabet.index(char)]` \n\n- `char - \n\nalphabet[15 - index]` \n\n- A\n- \n\nP (15). Correct. \n\n- B - \n\nO (14). Actual is R.\n\n- \n\n**Conclusion:** The substitution is not a simple formula. It's the most\ndifficult part of the puzzle.\n\n\n\n### Rule 3: Appending\n\n- **Run 13 (step\n13)` (\n\n\"ABCDE\\\", \n\n\"AABBC\\\")`:\n\n- Appended 3 `0`s.\n\n- **Data for this\ninput:**\n\n- Step 7: 7 appended.\n\n- Step 8: 8 appended.\n\n- Step\n13: 3 appended.\n\n- **Observation:** The number of appended characters is not\n`step_number`. It's not `step_number mod X`. It's not related to the number\nof unique characters in a simple way.\n\n- **New Hypothesis:** The number of\nappended characters is `(sum of alphabet positions of main_sequence) mod\n(step_number)`. \n\n- Run 13: `main` is \n\n\"ABCDE\". Sum = 0+1+2+3+4 = 10.\n\n- `10 mod 13 = 10`. Actual is 3.\n\n- **Hypothesis 2:** `(sum of alphabet\npositions of vice_sequence) mod (step_number)`. \n\n- Run 13: `vice` is\n\n\"AABBC\". Sum = 0+0+1+1+2 = 4. `4 mod 13 = 4`. Actual is 3.\n\n- \n\n**Conclusion:** Rule 3 is still unsolved.\n\n\n\n### Rule 4 \n\n5:\nConditional Substitution\n\n- **Run 13:** Both rules are active.\n\n- \n\n**Conclusion:** I am still far from understanding these rules.\n\n\n\n###\nOverall Progress and Plan\n\nI have a good understanding of Rule 1 (mostly).\n\nRule 2's palindrome is clear, but the substitution is very complex. Rule 3's\nappended character is known, but the number is not. Rules 4 and 5 are still\nblack boxes.\n\n\n\nGiven the complexity and the number of steps remaining, I\nwill switch my strategy. Instead of trying to find a universal formula for\neach rule, I will try to find patterns in the outputs of Rules 4 and 5. I will\nuse very simple inputs to see if I can force a simple output from these\nrules.\n\n\n\n- **Input for Run 14:** (\n\n\"AAAAE\\\", \n\n\"BBBBC\\\")\n\n- \n\n**Goal:** This input is very low in complexity. I want to see how Rules 4 and\n5 behave. I expect a simple output that might reveal their logic.\n\n\n\n}
}
```

</tool\_call>

## E DISCLOSURE OF LLM USAGE

The LLM was exclusively used during the editing (e.g., grammar, spelling, word choice). It plays no role in the ideation, research methodology, experimental design, or data analysis. All research concepts, ideas, and analyses were independently developed and conducted by the authors. The

2862 authors are fully accountable for the manuscript, including any text generated or refined by the LLM,  
2863 to ensure compliance with ethical guidelines and prevent plagiarism or scientific misconduct.  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896  
2897  
2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915