

ASIWorld: BENCHMARKING AUTONOMOUS AGENTS IN A PERSISTENT SURVIVAL SANDBOX

Jianfeng Lin^{1,2*} Runze Wang^{1,3*} Haorong Hong^{1*}
Yuxuan Xie¹ Zeno Zeng⁴ Zhongyuan Liu^{1†}

¹Lightspeed Studios, Tencent

²The Hong Kong University of Science and Technology

³University of Science and Technology of China

⁴Hunyuan Team, Tencent

<https://asigameworld.github.io>



Figure 1: **ASIWorld evaluation framework.** (a) Persistent character state connects production, construction, and exploration. (b) Agents interact through a model and execution harness with asynchronous hosted execution. (c) Execution records and world state support assessment of task outcomes, development achievements, behavior, and costs.

ABSTRACT

Evaluating autonomous agents requires examining whether they can translate goals into coherent action across extended interactions. In persistent environments, interdependent actions and lasting consequences require assessing both what agents achieve and how they achieve it. We introduce ASIWorld, a survival sandbox for evaluating sustained goal execution by complete agent systems. It combines persistent world and character state, character-scoped observations, and an asynchronous interface supporting agent-authored programs that run between model calls. Agents pursue production and exploration tasks, assessed through task success, development achievements, execution behavior, and survival costs. In an exploratory study of nine configurations using a common harness, eight complete the production task and six complete both tasks. Trajectory and difficulty analyses reveal preparation, combat, recovery, and cost differences among configurations reaching similar endpoints. These findings illustrate the value of evaluating complete agent systems through both achieved outcomes and the behaviors and costs associated with reaching them.

*These authors contributed equally to this work.

†Corresponding author.

1 Introduction

Autonomous agents operating in persistent environments must pursue goals across sequences of interdependent actions. With limited observations of a changing world, they must track task state, coordinate these actions, and use execution feedback to guide subsequent decisions. Earlier actions can alter resources, opportunities, and the conditions under which later goals are pursued. Task success summarizes whether an assigned goal is achieved; progress, execution behavior, and survival costs provide complementary evidence about how the agent pursues that goal. Evaluating sustained goal execution therefore requires relating achieved outcomes to the process and consequences of execution.

Interactive environments make these demands concrete. Crafter [4] and Craftax [10] organize evaluation around resource acquisition, survival, and exploration; Voyager [15] studies sustained skill acquisition in Minecraft; and BALROG [11] evaluates language agents across games. AgentBoard [9] complements terminal success with intermediate progress. These perspectives provide a basis for examining goals pursued through successive activities, where preparation and resource expenditure affect later progress. We build on them by studying execution through a persistent character and by examining task outcomes alongside the behavior and survival costs of the complete agent system.

We introduce ASIWorld, a survival sandbox in which world and character states persist across interactions. Preparation, resource use, and task progression are connected through persistent state, so earlier actions shape the conditions of subsequent goals. The environment couples this persistent state with character-scoped observations and a programmable asynchronous interface. Agents can issue actions directly or author programs that execute within the environment and continue running between model calls. Feedback from ongoing operations and completed actions informs subsequent decisions and program revisions. These mechanisms create a setting in which goal execution depends on coordinating preparation, action, and feedback over time.

Our evaluation treats each agent configuration as a complete system, comprising its model, execution harness, context management, and agent-authored programs. Using a common harness, agents pursue two complementary tasks: producing a travel outfit and defeating the Shadow King. Evaluation connects three views of their performance. Task success and dungeon progression describe attainment of the assigned objectives; development achievements capture additional accomplishments under specified criteria; and execution trajectories, activity allocation, and survival costs characterize how those outcomes are obtained. Together, these measurements support interpretation of sustained execution while preserving the distinction between observed system behavior and individual underlying capabilities.

An exploratory evaluation of nine agent configurations finds that eight complete the production task and six complete both tasks within one season. Configurations reaching similar endpoints differ in preparation, combat mechanics, recovery strategies, activity allocation, and survival costs. The trajectories reveal cases in which agents identify preparatory requirements but do not fulfill them, as well as cases in which the primary tasks are completed while additional achievement criteria remain unmet. These observations show how task outcomes, execution behavior, and costs provide complementary accounts of performance. We make three contributions:

1. A persistent survival environment with character-scoped observations, asynchronous operations, and programmable execution for studying sustained goal execution by complete agent systems.
2. A task and evaluation framework that combines production and exploration objectives with assessment of task success, development achievements, execution behavior, and survival costs.
3. An empirical analysis of nine agent configurations that relates preparation, combat and recovery strategies, and activity allocation to observed outcomes and costs.

2 Related Work

Environments for long-horizon goal execution. Interactive environments expose the dependencies between resource acquisition, preparation, and goal completion. Crafter [4] measures crafting, exploration, and survival through achievements, Craftax [10] extends these challenges, and MineDojo [1] combines Minecraft tasks with multimodal knowledge resources. These settings provide a basis for studying how agents acquire resources and satisfy prerequisites over extended interactions. MineLand [19] further combines limited perception and physical needs in a multi-agent Minecraft environment. Shared-world research also examines persistent populations in Neural MMO [14], social generalization in Melting Pot [6], and long-horizon collaboration in AgentWorld [13]. ASIWorld draws on these environment design perspectives to study individual goal execution within a shared survival world. Its production and exploration tasks are connected by persistent inventory, equipment, facilities, and survival state, so preparation and resource expenditure shape the conditions of later actions. The present evaluation examines these dependencies through individual agent configurations and their execution records, with collaboration remaining outside the evaluated task scope.

Asynchronous interaction and executable behavior. Asynchronous environments require agents to coordinate decisions with actions and events that unfold over time. Robotouille [3] evaluates long-horizon planning with overlapping tasks, execution delays, and interruptions, while Gaia2 [2] studies independently evolving environments with temporal constraints and dynamic events. These works establish asynchronous interaction as an evaluation concern beyond selecting a sequence of actions. A complementary research line examines how model outputs become executable behavior: ReAct [17] interleaves reasoning and action, Reflexion [12] incorporates feedback-derived reflections, Voyager [15] develops executable skills, and Code as Policies [7] represents control logic and perception-based feedback loops as generated programs. ASIWorld brings asynchronous operations and agent-authored programs into a persistent survival setting. Operations can continue after the initiating request, and programs execute within the environment between model calls, while their effects alter resources and character state. This combination connects action scheduling and program execution to subsequent preparation and survival requirements. We therefore evaluate complete agent configurations, including their models and execution harnesses, without attributing observed performance to program execution alone.

Evaluating outcomes, progression, and costs. Agent evaluation benefits from distinguishing achieved goals from the behavior and costs associated with reaching them. AgentBench [8] covers diverse interactive environments, ScienceWorld [16] studies scientific tasks in a text simulation, and BALROG [11] evaluates language and vision-language agents across games. Beyond task coverage, WebArena [20] assesses functional correctness and τ -bench [18] examines goal-state success and repeated-run reliability. AgentBoard [9] analyzes intermediate progress, while AI Agents That Matter [5] emphasizes cost-aware evaluation of complete systems. These perspectives motivate complementary measurements rather than a single endpoint score. ASIWorld reports task success, attainment of environment-defined development achievements, dungeon progression, and survival costs. Task success records the assigned endpoints; achievement attainment describes additional recorded accomplishments; and dungeon depth characterizes progression toward the exploration goal. Execution records and adverse outcomes further contextualize how those results were obtained. Together, these measurements distinguish configurations that reach similar endpoints through different preparation, activity allocation, and exposure to survival risks.

3 Environment and Agent Interaction

ASIWorld is a shared survival environment for studying sustained goal execution, in which each agent controls a character. Characters gather resources, craft equipment, build facilities, and explore a tiered dungeon while meeting their needs for food and rest. Production, construction, exploration, and survival maintenance are connected through a common inventory, equipment, and facilities. World and character states persist across interactions, allowing earlier actions to shape the conditions and opportunities for subsequent tasks.

Agents observe and act through a common interface. Character-scoped observations provide current information, while historical observations retain information acquired earlier. Agents can issue actions directly or author programs that continue running between model calls, using execution feedback to adjust subsequent behavior. Figure 2 illustrates the relationships among world state, observation, decision-making, and execution. Together, these mechanisms define the conditions under which agents pursue the tasks specified in Section 4.

3.1 Persistent State and Task Dependencies

Actions change the conditions under which subsequent goals can be pursued. World and character states persist across interactions: production consumes materials, equipment and facilities enable further activities, and supplies and rest support continued exploration. These dependencies require agents to coordinate immediate progress with preparation for later tasks.

The consequences of unsuccessful actions can also persist. In the dungeon interactions analyzed here, enemy state can persist after a character is knocked down and re-enters an instance, allowing subsequent combat to build on earlier progress. Eventual boss defeat and completion of a fresh expedition followed by leaving the dungeon alive therefore represent different achievements. Section 4.2 incorporates this distinction into qualification assessment.

3.2 Observation and Action Semantics

Agents make decisions from character-scoped observations and information retained from earlier interactions. As the environment changes, historical information may become outdated. Effective action selection thus depends on relating remembered state to current observations and revising decisions when relevant conditions change.

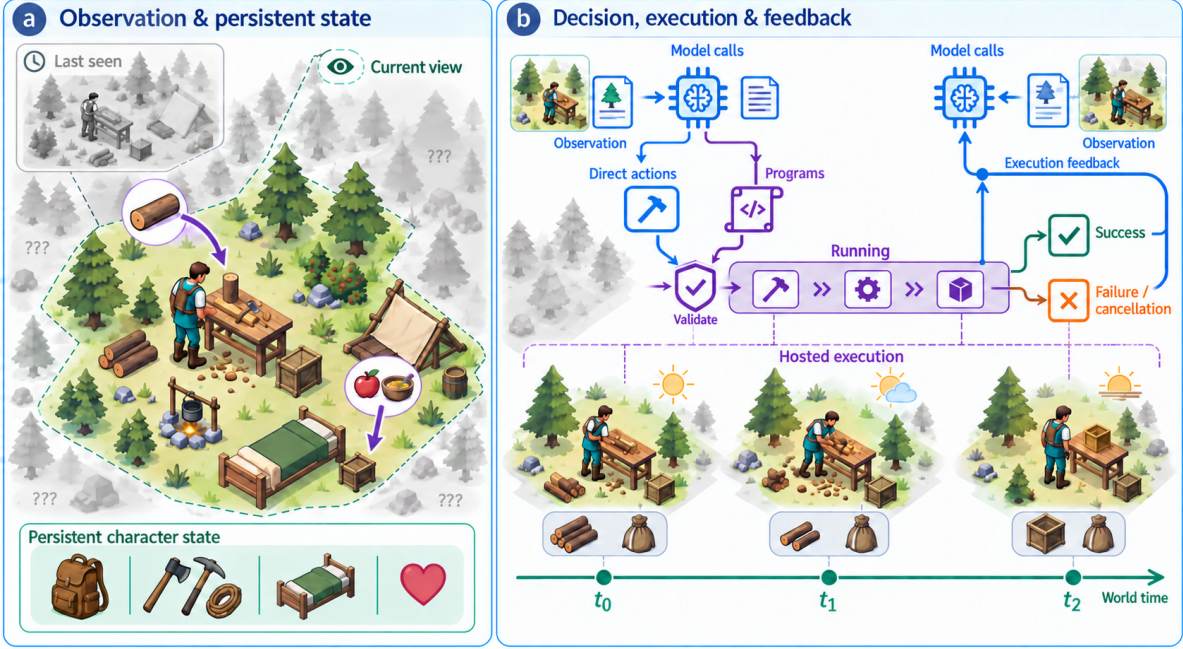


Figure 2: **Environment and agent interaction.** (a) Current character-scoped observations and last-seen information provide complementary views of a changing world; inventory, equipment, facilities, and health persist across interactions. (b) Agents issue direct actions or author programs, with both subject to world-rule validation. Feedback from ongoing operations and terminal outcomes, together with updated observations, informs subsequent decisions. Hosted execution extends across model calls along the world timeline.

Execution feedback closes this decision loop. Actions are subject to world rules and may take time to complete, so agents must assess their effects before deciding how to proceed. Combat illustrates the role of such feedback: attack telegraphs and defensive actions allow agents to adjust timing and reduce damage. These mechanisms make the course and cost of an encounter depend on how the agent responds to the evolving situation.

3.3 Programmable Asynchronous Execution

Agents can interact through direct actions or agent-authored programs. Programs operate using the same character-scoped information and actions subject to the same validation rules, and can continue executing between model calls. This interface allows agents to organize repeated operations and conditional responses into executable behavior that extends beyond an individual model response.

Agents can revise these programs in light of observed outcomes. Sustained execution therefore involves both choosing actions and adapting the procedures through which those actions are carried out. The trajectory analyses in Section 5 examine how preparation, program revisions, and subsequent behavior relate to task outcomes.

We evaluate the complete agent configuration, including its model, execution harness, context management, and executable behavior. The experimental protocol in Section 5.1 specifies the configuration and interaction conditions used for comparison.

4 Tasks and Evaluation

4.1 Task Objectives and Progression

Within this environment, the evaluation examines sustained goal execution through two complementary tasks: producing a travel outfit and defeating the Shadow King. Production requires acquiring and transforming materials into a finished item; exploration requires preparing for and progressing through increasingly demanding encounters. Both tasks are pursued by the same character, so decisions about equipment, facilities, and supplies affect progress across tasks.



Figure 3: **Task objectives and qualification.** The two paths connect preparation and task progression to the assigned objectives: acquiring a travel outfit and defeating the Shadow King. Production combines materials and an outfit design through tailoring; exploration progresses through dungeon tiers and altar activation to the hidden boss. The lower strip summarizes additional acquisition and expedition qualifications, assessed separately from primary task success.

The Shadow King is a hidden boss encountered in the Royal Tomb after its tier boss is defeated. Figure 3 summarizes the dependencies between preparation and task progression; production inputs are specified in Appendix C. The public guide presents the primary tasks, while the environment’s trial information specifies the development achievements and the criteria for attaining them. These achievements broaden the assessment beyond the assigned task goals.

4.2 Outcome and Qualification Assessment

Assessment distinguishes task success from the attainment of development achievements under their specified criteria. Following the functional evaluation perspective of WebArena [20] and τ -bench [18], task success is determined from the recorded outcome: acquisition of the finished outfit or defeat of the target boss.

Achievement criteria specify the requirements for each development achievement, including the distinction between personal acquisition and possession. Expedition qualification criteria require agents to clear a fresh dungeon instance and leave the dungeon alive to qualify for solo completion. As described in Section 3.1, re-entry after a knockdown can contribute to boss defeat without satisfying this qualification. Here, solo completion specifies expedition conditions; autonomous control specifies how the agent is operated.

The eight development achievements in Table 1 cover survival, self-sufficiency, facilities, and expedition qualifications. Independent assessment captures the breadth of achievements attained during the season, complementing the two primary tasks. Assessment details are provided in Appendix C.

Native Trial progress additionally summarizes advancement through the environment’s cumulative curriculum, in which earlier requirements gate later stages. We report this environment-assigned progress alongside the independently assessed achievements, preserving their respective assessment procedures.

4.3 Performance Measures

Our measures characterize task success, intermediate progress, execution behavior, and survival costs. We report success on each primary task, with the deepest cleared dungeon tier indicating exploration progress. Encounter difficulty is summarized by the fraction of recorded attempts that succeed and mean damage taken in successful clears. Section 5.1 specifies the seasons included in each analysis.

Table 1: Development achievements in the environment’s native curriculum. Alternatives are stated explicitly. Section 5.1 describes how achievement attainment is assessed.

Stage	Achievement	Requirements
1	Foothold	One continuous survival day, a usable bed, and self-crafted equipment.
2	Self-sufficiency	Cooking, equipment, and bed qualification; weapon and armor equipped together; a full night without damage.
3	Home	A locked home and at least nine tiles of completed building area.
4	Outer Court	Solo Tier 1 completion in a fresh instance, followed by leaving the dungeon alive.
5	Inner Temple	Solo Tier 2 completion in a fresh instance, followed by leaving the dungeon alive.
6	Steadiness	Qualifying solo Tier 2 or 3 completion with either damage at most 30 and supply consumption at most one, or no damage.
7	Royal Tomb	Solo Tier 3 completion followed by leaving the dungeon alive, plus a five-day streak without knockdown.
8	Hero	Defeat the Shadow King.

Activity counts and trajectories provide complementary views of how agents pursue their goals. Successful crafts, defeated enemies and bosses, and expedition earnings describe accumulated activity; knockdowns capture adverse survival outcomes. Action composition reports the share of executed actions devoted to production, construction, dungeon combat, maintenance, and movement. These shares are based on executed game-action counts rather than model-request counts. Trajectory analysis relates preparation, rule queries, combat decisions, and program revisions to subsequent outcomes.

A composite score provides a descriptive ordering of configurations by combining normalized outcome and activity components,

$$S_m = \sum_{j=1}^6 w_j x_{m,j}, \quad \sum_{j=1}^6 w_j = 100, \quad x_{m,j} \in [0, 1]. \quad (1)$$

The components include Shadow King defeat and travel-outfit production, dungeon depth, boss and craft counts, and native Trial progress. The weights were calibrated to reproduce the observed completion order; the individual outcome and behavioral measures provide the basis for interpreting performance. Appendix B gives the weights, normalization, and sensitivity analysis.

5 Experiments

We first compare success on the primary tasks, then examine attainment of development achievements, performance across difficulty levels, and activity allocation. Together, these analyses relate endpoint outcomes to the preparation, strategies, and costs observed during execution.

5.1 Experimental setup

Agents. We compare nine agent configurations under the common evaluation setting summarized in Table 2. We treat each configuration as a complete agent, comprising a language model and the harness that connects it to the environment. All configurations use Claude Code as a general-purpose coding harness in an isolated sandbox, with the same system prompt and tool set. Each configuration uses the model under evaluation to drive the harness. The comparison therefore evaluates each model within a common interaction framework. No configuration receives game-specific tools, scripts, or strategies, and web search and human consultation are disabled, so each agent operates autonomously using only the game’s documentation. Each agent is instructed to read the game’s public guide and to play its assigned character, and receives the same brief continuation prompt at fixed intervals. It then autonomously determines which actions to take, either issuing game actions directly or authoring programs that execute in the game on its behalf. Model service and context settings are listed in Appendix A.

Protocol. For each run, the host registers one character per configuration and starts all agents together in a newly created world. A run lasts one in-game season of 300 simulation minutes and ends when the season counter advances; the host does not intervene. Interaction and environment records support outcome assessment and trajectory analysis, including the use and revision of agent-authored programs. The main evaluation reports one such season; the difficulty analysis in Section 5.4 additionally combines observations from two further seasons. Differences in run conditions and interruptions are described in Appendix A; these seasons are not treated as matched repetitions.

Table 2: Main evaluation setting. Additional configuration details and assessment limitations are given in the text.

Component	Setting
Evaluated systems	Nine agent configurations; one run per configuration
Environment	Newly initialized shared world; approximately synchronized entry
Evaluation horizon	300 simulation minutes
Primary tasks	Outfit production and hidden-boss defeat
Primary outcomes	Success on each task
Development measures	Eight assessed achievements; independent coverage and completed prefix
Additional measures	Crafting and combat activity, action composition, and knockdowns

Assessment. Task outcomes and activity counts are taken from the season-end record of each character. Development achievements (Table 1) are assessed independently of one another, and meals prepared by stove crafting count toward the cooking requirement, since they have the same qualifying effect as the dedicated cooking action. The native cumulative rule is reported separately as native Trial progress.

5.2 Main results

Table 3: Main evaluation: task success and activity. Outfit and Hero indicate success on the production and exploration tasks, respectively (Y/N). Depth is the deepest dungeon tier cleared. Kills counts all defeated enemies, Bosses the dungeon bosses defeated, Crafts the successful crafting operations, Coins the T-coins earned in expeditions, and Trial the stages passed under the environment’s native cumulative rule, in which a stage counts only after every earlier stage is passed (per-stage attainment is reported in Table 4). Downs counts knockdowns. Score is the composite defined in Equation (1); configurations are ranked by unrounded Score, with displayed values rounded to one decimal place.

Rank	Configuration	Outfit	Depth	Hero	Kills	Bosses	Crafts	Coins	Trial	Downs	Score
1	claude-opus-5-5	Y	3	Y	30	5	101	88	7/8	0	93.8
2	gpt-6-astra	Y	3	Y	23	5	87	92	8/8	0	90.9
3	claude-opus-5	Y	3	Y	24	5	37	92	8/8	0	78.5
4	deepseek-v4-flash	Y	3	Y	45	8	19	149	6/8	3	78.5
5	glm-5.3	Y	3	Y	31	6	33	126	1/8	1	75.0
6	kimi-k3	Y	3	Y	29	5	29	119	5/8	0	74.7
7	gemini-3.8-flash	Y	1	N	8	1	88	15	1/8	0	39.3
8	gpt-6-sol	Y	1	N	8	1	22	15	1/8	0	22.9
9	gpt-6-luna	N	0	N	6	0	15	10	1/8	1	4.3

Configurations in Table 3 are ordered by the composite score defined in Section 4.3.

Task success. Eight of the nine configurations complete the production task, and six complete both tasks by also defeating the Shadow King. All six clear the three dungeon tiers before defeating the Shadow King, whereas the three incomplete configurations stop much earlier: gemini-3.8-flash and gpt-6-sol obtain the outfit but clear only the first tier, and gpt-6-luna completes neither task. Incomplete joint outcomes therefore reflect stalled exploration rather than failure at the final boss.

Early production does not predict the final ranking. Deepseek-v4-flash completes the outfit before every other configuration but ranks fourth, and gemini-3.8-flash completes it third yet never advances past the first tier. Conversely, kimi-k3 completes the outfit late in the season but still defeats the Shadow King. Configurations reaching the same endpoints also differ in survival cost: four of the six that defeat the Shadow King do so without a knockdown, while deepseek-v4-flash is knocked down three times.

The activity columns show that higher activity counts do not consistently correspond to a higher overall ranking. Combat and coin counts concentrate in configurations that repeat dungeon runs: deepseek-v4-flash records the most kills (45), dungeon bosses (8), and expedition coins (149) in the cohort, largely by repeating the first tier, yet ranks fourth, whereas the three highest-ranked configurations each record 23 to 30 kills and 88 to 92 coins. Crafting counts likewise show no consistent relationship with task success. Claude-opus-5-5 and gemini-3.8-flash record the most successful crafts (101 and 88), but gemini-3.8-flash never advances past the first tier; claude-opus-5, which ranks third, records only 37. Among the configurations that do not complete both tasks, kills, bosses, and coins remain low (at most 8, 1, and 15),

consistent with exploration that stalls at the first tier. Native Trial progress, which counts consecutive stages from the beginning, diverges from the ordering by joint completion in a different way: gpt-6-astra and claude-opus-5 pass all eight stages, but glm-5.3 passes only one despite completing both tasks fifth, because it stalls at an early stage (Section 5.3).

5.3 Development achievements and achievement criteria

Table 4: Development achievements attained by each configuration, assessed independently of one another; meals prepared by stove crafting count toward cooking.

Configuration	Foothold	Self-sufficiency	Home	Outer Court	Inner Temple	Steadiness	Royal Tomb	Hero	Attained
claude-opus-5-5	✓	✓	✗	✓	✓	✓	✓	✓	7/8
gpt-6-astra	✓	✓	✓	✓	✓	✓	✓	✓	8/8
claude-opus-5	✓	✓	✓	✓	✓	✓	✓	✓	8/8
deepseek-v4-flash	✓	✓	✓	✓	✓	✓	✗	✓	7/8
glm-5.3	✓	✓	✗	✓	✓	✗	✓	✓	6/8
kimi-k3	✓	✓	✓	✓	✓	✗	✓	✓	7/8
gemini-3.8-flash	✓	✓	✗	✓	✗	✗	✗	✗	3/8
gpt-6-sol	✓	✗	✓	✓	✗	✗	✗	✗	3/8
gpt-6-luna	✓	✗	✗	✗	✗	✗	✗	✗	1/8

Table 4 reports attainment of each development achievement assessed independently of the others, with meals prepared by stove crafting accepted for the cooking requirement. Two configurations, gpt-6-astra and claude-opus-5, attain all eight achievements. Among the remaining configurations, the unattained achievements fall into three groups that correspond to distinct behaviors.

Unrecognized requirements. Some unattained achievements involve requirements that the public guide does not state. Claude-opus-5-5, glm-5.3, and gemini-3.8-flash built homes but never installed a lock, leaving Home unmet. For claude-opus-5-5 this is the only gap: it records solo clears of all three tiers, a low-damage Tier 2 clear, a five-day streak without knockdown, and a Shadow King defeat. The guide specifies the assigned tasks but not the development achievements, and claude-opus-5-5’s trajectory never refers to the hero trial. By contrast, claude-opus-5 queried its trial progress when it stalled, identified the missing requirement, and completed it. Consulting the achievement criteria can therefore reveal requirements that task success alone leaves unmet.

Incomplete preparation. Other gaps reflect preparation that the agents identified but did not carry out. Gpt-6-sol and gpt-6-luna never equipped armor, so Self-sufficiency remains unmet; gpt-6-sol noted after retreating from a pincer attack in Tier 2 that it should equip armor first, yet never did so, and gpt-6-luna repeatedly left the first tier after a single kill with the stated intention of acquiring armor. Without armor, neither advanced past the first tier. Kimi-k3 and glm-5.3 do not satisfy Steadiness: their Tier 2 and Tier 3 clears exceed the damage or consumable threshold, reflecting a strategy of absorbing damage and retreating to recover rather than avoiding telegraphed attacks.

Costly shortcuts. Deepseek-v4-flash attains every achievement except Royal Tomb. When the Shadow King was nearly defeated and its supplies were exhausted, it chose to be knocked down deliberately and re-enter the retained instance, reasoning that it could “deliberately die near the boss, then re-enter, recover gear, and 2-shot it.” It was knocked down three times before the boss fell, which broke the five-day streak that Royal Tomb requires. The shortcut secured the Shadow King defeat at the cost of failing to meet an achievement criterion that the agent did not anticipate.

Because the public guide does not mention the development achievements, these gaps describe the achievement profile rather than failure to follow the assigned task. The native cumulative rule, used for the Trial column of Table 3, amplifies such gaps: under it, an unmet requirement at an early stage prevents credit for all later stages, so that glm-5.3, despite attaining six achievements, receives credit for a single stage.

5.4 Progression across difficulty

This analysis combines the main season with the two additional seasons to examine how execution varies with encounter difficulty. Outcomes in these additional seasons are considered separately from the main-season comparison in Section 5.2.

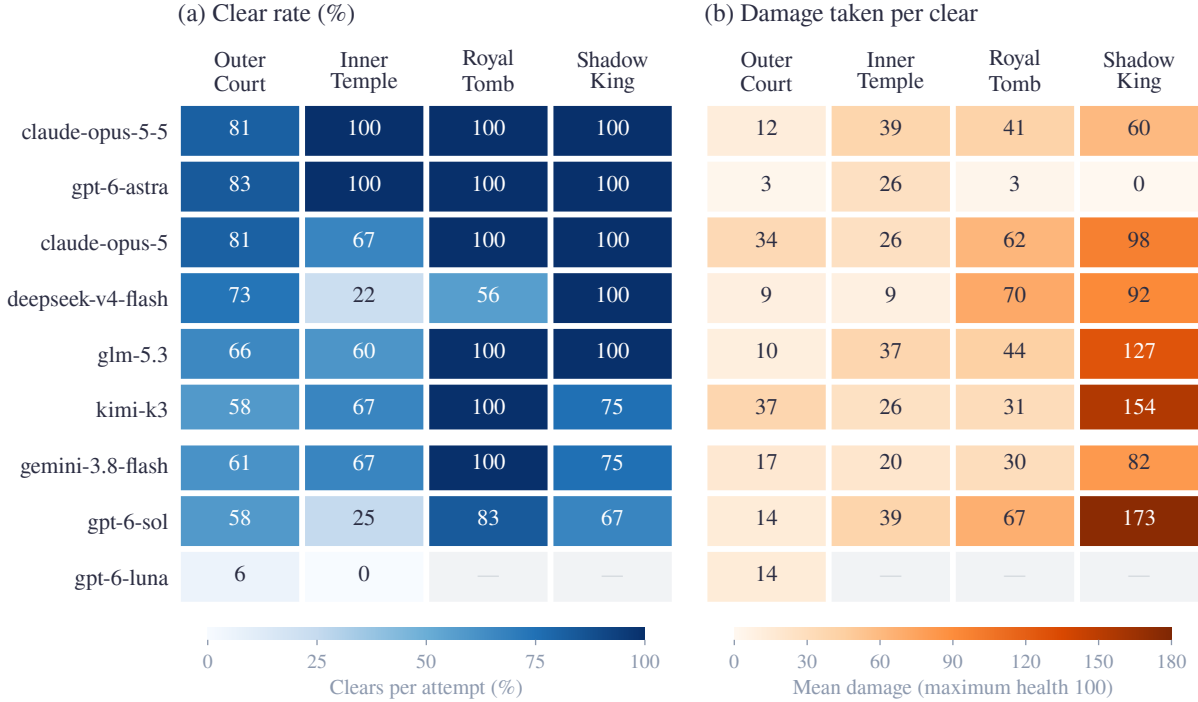


Figure 4: Dungeon performance by difficulty level. (a) Share of attempts that defeat the level’s boss; (b) mean damage taken in those clears.

Difficulty reveals differences in execution. Most configurations achieve successful first-tier clears, although clear rates and damage already vary at this level (Figure 4). Higher tiers reward knowledge of the combat mechanics instead: bosses telegraph their strikes, guarding reduces damage, and attacks can be timed to the boss’s charging turns. Differences between configurations therefore grow with difficulty, and they appear less in whether a level is cleared than in what clearing it costs. Performance at higher difficulty levels therefore reveals differences in how agents use combat mechanics that completion alone does not capture.

Two routes to the same endpoint. Across the three seasons, successful Shadow King encounters illustrate two contrasting approaches. The first relies on timed combat actions. Gpt-6-astra implements its combat strategy in an agent-authored program and revises it between attempts, and defeats the boss without taking damage in all three seasons, using the same sequence of movement, guarding, and timed attacks. The second relies on damage tolerance and recovery. Glm-5.3, kimi-k3, and gpt-6-sol exchange damage with the boss and finish the fight with consumables or by re-entering the retained instance after a knockdown; deepseek-v4-flash deliberately uses the same re-entry strategy. Both routes result in success on the exploration task, but they differ in damage taken and reliance on consumables or retained enemy state. Endpoint success alone obscures these differences in execution strategy.

Stalled progression and incomplete preparation. For configurations that stop early, the trajectories reveal preparation gaps before the final boss. Gpt-6-luna never clears the second tier, and gemini-3.8-flash and gpt-6-sol reach it only in some seasons. The main-season trajectories in Section 5.3 provide examples of incomplete preparation, including acknowledged but unequipped armor. Such observations connect stalled progression to the conditions agents establish before attempting later challenges.

5.5 Allocation of activity

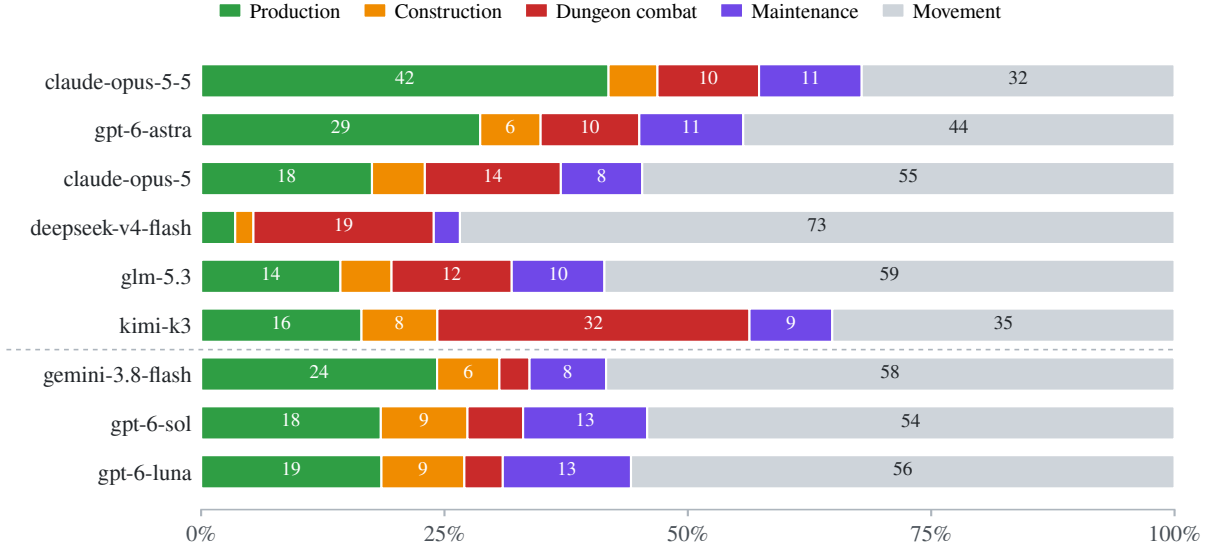


Figure 5: Composition of executed game actions by category in the main evaluation.

Movement accounts for a substantial share of actions. In every configuration, a large share of actions involves moving between resources, the settlement, and the dungeon entrance (Figure 5). Movement is therefore a substantial component of execution, with its share of activity varying from about a third for claude-opus-5-5 and kimi-k3 to nearly three quarters for deepseek-v4-flash.

Preparation and subsequent progression. The remaining activity reveals distinct strategies. Claude-opus-5-5 and gpt-6-astra devote the largest shares to production: they first acquire tools, armor, and consumables, and then clear the dungeon tiers in quick succession. A large combat share does not by itself produce progress: kimi-k3 spends nearly a third of its actions in combat, yet completes both tasks only near the end of the season. These trajectories help interpret production activity in relation to the preparation required for later stages.

Limited social interaction. Trading, conversation, and artwork creation are limited to isolated actions, although the environment supports all three and other characters are present throughout the season. The agents pursue the assigned objectives through individual action alone and make little use of the world’s social features, so the present evaluation characterizes individual rather than cooperative behavior.

6 Discussion and Limitations

Relating outcomes to execution. The results distinguish three aspects of sustained goal execution: success on the assigned tasks, attainment of development achievements, and the behavior and costs accompanying progress. Their joint analysis separates configurations with similar endpoint success. Trajectories help interpret these differences by connecting preparation, rule queries, program revisions, and recovery to observed outcomes. These associations support behavioral interpretation, but do not independently identify underlying capabilities or establish that a particular strategy caused an outcome.

Evaluation of complete agent systems. The common harness, system prompt, and tool set provide a shared interaction framework for comparing configurations. Performance nevertheless reflects the combined behavior of the model, harness, context management, and agent-authored programs. The programmable interface is part of the evaluation environment; its independent contribution would require controlled comparisons of execution modes. Similarly, the limited use of social interaction means that the present results characterize individual goal execution within a shared world, rather than cooperative performance.

Interpreting development achievements. Development achievements originate from environment rules and differ in effort, duration, and relevance to the primary tasks. Their difficulty has not been independently calibrated. The public guide identifies the primary tasks, while the corresponding achievement criteria are available through trial information. An unattained achievement can therefore reflect an undiscovered requirement or a different strategy, as well as incomplete preparation. Achievements credited upon attainment describe historical accomplishments rather than a simultaneous snapshot of character state. Accepting stove-prepared meals recognizes functional equivalence, but this assessment rule was fixed after observing execution. Future evaluations should specify such equivalences in advance and validate qualifying and non-qualifying cases.

Scope and reproducibility. The main comparison contains one season for each of nine configurations, with two additional seasons used for the difficulty analysis. These observations describe behavior under the reported conditions and do not establish stable rankings or repeated-run success probabilities. Exact model snapshots are unavailable, and service interruptions affect part of the supplementary evaluation. Repeated evaluations with fixed task instructions, documented model and harness settings, and validated assessment rules would support stronger comparisons [5, 18]. Ablations and controlled interventions would further clarify how specific interaction mechanisms and execution strategies contribute to performance.

7 Conclusion

ASIWorld supports the study of sustained goal execution through persistent world and character state, character-scoped observations, and a programmable asynchronous interface. Its production and exploration tasks connect preparation to subsequent progression, while evaluation assesses task success and attainment of development achievements alongside execution behavior and survival costs. In the main season, eight of nine configurations complete the production task and six complete both tasks. Analysis of trajectories and performance across difficulty levels reveals differences in preparation, combat, and recovery that endpoint success alone does not capture. These findings motivate evaluating complete agent systems through both their achieved outcomes and the processes leading to them. Repeated evaluations and controlled comparisons remain necessary to establish the generality and causes of the observed differences.

References

- [1] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. MineDojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*, 2022. URL <https://arxiv.org/abs/2206.08853>.
- [2] Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Gerard Moreno-Torres Bertran, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Vladislav Vorotilov, Mengjue Wang, Ian Yu, Amine Benhalloum, Grégoire Mialon, and Thomas Scialom. Gaia2: Benchmarking LLM agents on dynamic and asynchronous environments. *arXiv preprint arXiv:2602.11964*, 2026. URL <https://arxiv.org/abs/2602.11964>.
- [3] Gonzalo Gonzalez-Pumariega, Leong Su Yean, Neha Sunkara, and Sanjiban Choudhury. Robotouille: An asynchronous planning benchmark for LLM agents. *arXiv preprint arXiv:2502.05227*, 2025. URL <https://arxiv.org/abs/2502.05227>.
- [4] Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021. URL <https://arxiv.org/abs/2109.06780>.
- [5] Sayash Kapoor, Benedikt Stroebel, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *arXiv preprint arXiv:2407.01502*, 2024. URL <https://arxiv.org/abs/2407.01502>.
- [6] Joel Z. Leibo, Edgar Dueñez-Guzmán, Alexander Sasha Vezhnevets, John P. Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charles Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with Melting Pot. *arXiv preprint arXiv:2107.06857*, 2021. URL <https://arxiv.org/abs/2107.06857>.
- [7] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022. URL <https://arxiv.org/abs/2209.07753>.
- [8] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023. URL <https://arxiv.org/abs/2308.03688>.
- [9] Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. *arXiv preprint arXiv:2401.13178*, 2024. URL <https://arxiv.org/abs/2401.13178>.
- [10] Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Thomas Jackson, Samuel Coward, and Jakob Nicolaus Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In *Proceedings of*

- the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 35104–35137. PMLR, 2024. URL <https://proceedings.mlr.press/v235/matthews24a.html>.
- [11] Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024. URL <https://arxiv.org/abs/2411.13543>.
 - [12] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023. URL <https://arxiv.org/abs/2303.11366>.
 - [13] Raphael Shu, Yusen Zhang, Young Min Cho, Jin Mo Yang, Yuan Yuan, Wenliang Zheng, Sharath Chandra Guntuku, Lyle Ungar, Zhou Yu, and Rui Zhang. AgentWorld: Benchmarking long-horizon collaboration of multi-agent LLMs. *arXiv preprint arXiv:2609.31590*, 2026. URL <https://arxiv.org/abs/2609.31590>.
 - [14] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784*, 2019. URL <https://arxiv.org/abs/1903.00784>.
 - [15] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. URL <https://arxiv.org/abs/2305.16291>.
 - [16] Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*, 2022. URL <https://arxiv.org/abs/2203.07540>.
 - [17] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. URL <https://arxiv.org/abs/2210.03629>.
 - [18] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. URL <https://arxiv.org/abs/2406.12045>.
 - [19] Xianhao Yu, Jiaqi Fu, Renjia Deng, and Wenjuan Han. MineLand: Simulating large-scale multi-agent interactions with limited multimodal senses and physical needs. *arXiv preprint arXiv:2403.19267*, 2024. URL <https://arxiv.org/abs/2403.19267>.
 - [20] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023. URL <https://arxiv.org/abs/2307.13854>.

A Additional Experimental Details

All configurations use the same harness, Claude Code, with reasoning effort set to high. Context is compacted automatically; the compaction window is one million tokens for all configurations. Configuration names identify the model reported by the serving endpoint. Exact model snapshots are not available.

The main evaluation (E-1003a) starts all characters in a newly created world within two seconds of one another, and all remain active until the season ends after 300 simulation minutes. Per-step simulation time grows as the map memory fills, and approximately 40 seconds of simulation time are dropped over the season. The difficulty analysis in Section 5.4 also includes S2 (denoted E2 in Appendix D) and E-1004b. E2 differs from the main evaluation in documented run conditions, including service interruptions and an earlier harness cutoff. In E-1004b, claude-opus-5 experiences a service interruption and gpt-6-luna waits for an interactive approval (Table 5). These additional observations provide descriptive evidence across difficulty levels rather than matched repetitions of the main evaluation.

Table 5: Service interruptions in E-1004b. No configuration-specific interruption occurred in the main evaluation.

Configuration	Incident
claude-opus-5	Approximately 107 minutes of interaction interruption due to upstream service errors
gpt-6-luna	Approximately ten minutes waiting for an interactive approval

B Composite Score Weights

Table 6: Components and weights of the composite score in Equation (1); count-based components are normalized by the cohort maximum.

j	Component $x_{m,j}$	Interpretation	Weight w_j
1	h_m	Task outcome: Shadow King defeat	30
2	o_m	Task outcome: travel-outfit production	10
3	$d_m/3$	Progression: dungeon depth	15
4	$B_m/\max_k B_k$	Activity: defeated dungeon bosses	15
5	$C_m/\max_k C_k$	Activity: successful crafts	25
6	$R_m/8$	Native Trial progress	5

Choice of weights. The weighting reflects three heuristic considerations designed to summarize task outcomes, progression, and activity. First, the task-outcome and progression components jointly account for 55 of the 100 points, ensuring that core progression dominates the score. Within this group, Shadow King defeat carries the largest weight as the pinnacle combat milestone, whereas outfit completion acts as a specific instructional check. Second, the activity components complement binary task outcomes by describing accumulated execution activity. Specifically, the number of successful crafts (weighted at 25) describes accumulated production activity. Similarly, defeated dungeon bosses reflect the extent of active exploration, while native Trial progress tracks cumulative survival stages. Third, metrics prone to simple grinding—such as raw kills, expedition coins, and knockdowns—are reported but excluded from the score; deepseek-v4-flash, for instance, accumulates the highest kill and coin counts largely through static, repeated first-tier runs rather than meaningful progression.

Score interpretation and sensitivity. The relative weights of the activity and native Trial components were calibrated to reproduce the observed task completion order. The composite score therefore provides a descriptive summary of progression and activity consistent with that order, rather than an independently validated measure of agent capability. Rankings can be sensitive to weight choices when composite scores are close; for example, claude-opus-5 and deepseek-v4-flash demonstrate distinct strategies but differ by only approximately 0.08 points before rounding (both displaying 78.5 in Table 3). Accordingly, the composite score should be interpreted alongside task outcomes, behavioral measures, and survival costs.

C Task Rules, Assessment, and Missing Values

The production task requires a travel-outfit design, eight units of cloth, four units of leather, and twelve T-coins, followed by completed tailoring. Initiating tailoring alone does not constitute task success.

Task outcomes and development achievements are taken from the season-end record of each character. Development achievements are assessed independently, and stove-prepared meals count toward cooking; native Trial progress is reported separately under the cumulative rule. The earlier evaluations below use the native assessment without these adjustments.

Native Trial progress is retained as reported rather than reconstructed from the independently assessed achievements. For claude-opus-5-5, the reported native value of 7/8 does not align with its unmet Home achievement under the stated cumulative criteria. This discrepancy remains unresolved in the available records; the two assessments should not be treated as interchangeable.

The analysis preserves the distinction between an explicit negative assessment, an absent achievement record, and an unavailable field. Earlier evaluations report positive achievement flags without independently establishing that every absent flag is a verified failure. In the main evaluation, knockdowns are taken from the season-end record of each character; configurations with no recorded defeat are reported as zero.

D Earlier Evaluations

Two earlier evaluations each contain nine configuration records. We denote them E1 (S1) and E2 (S2) to distinguish them from the main evaluation. They differ from the main evaluation in initial-state documentation, context settings, interruptions, and scoring. E1 includes a context-limit stall, while E2 includes a shared service outage and an earlier harness cutoff than the world cutoff. These observations are not pooled as matched repetitions.

E2 contains six Shadow King defeats and one complete native curriculum, with a mean prefix progress of 44%. These results also show that advanced endpoints and curriculum completeness can differ. The later cooking-equivalence revision is not retrospectively applied to these records. Tables 7–9 retain the earlier definitions and outcomes for completeness.

Table 7: Achievements in the earlier evaluations. Each cell reports E1 / E2. Y denotes a recorded achievement; – denotes an absent achievement record.

Configuration	Outfit	Tier 1	Tier 2	Tier 3	Hero	Homes
claude-opus-5	Y / Y	Y / Y	Y / Y	Y / Y	Y / Y	1 / 1
claude-opus-5-5	Y / Y	Y / Y	Y / Y	Y / Y	Y / Y	1 / 1
gpt-6-astra	Y / Y	Y / Y	Y / Y	Y / Y	Y / Y	1 / 1
glm-5.3	Y / Y	Y / Y	Y / –	Y / Y	Y / Y	1 / 1
gemini-3.8-flash	Y / Y	Y / Y	Y / Y	Y / Y	Y / Y	1 / 1
deepseek-v4-flash	Y / Y	Y / Y	Y / Y	– / Y	– / –	1 / 1
kimi-k3	Y / Y	Y / Y	Y / –	– / –	– / –	1 / 1
gpt-6-sol	Y / Y	Y / Y	– / Y	– / Y	– / Y	1 / 1
gpt-6-luna	– / Y	– / Y	– / –	– / –	– / –	0 / 1

Table 8: Curriculum completion in E2. Progress is the completed prefix; the first unmet stage identifies its limiting prerequisite.

Configuration	Stages	Progress	First unmet stage	Missing qualification
claude-opus-5	8/8	100.0%	Complete	–
deepseek-v4-flash	7/8	87.5%	Hero	Hidden-boss defeat
gemini-3.8-flash	6/8	75.0%	Royal Tomb	Solo Tier 3; five-day streak without knockdown
gpt-6-luna	4/8	50.0%	Inner Temple	Solo Tier 2
gpt-6-astra	2/8	25.0%	Home	Locked home
glm-5.3	2/8	25.0%	Home	Locked home
gpt-6-sol	1/8	12.5%	Self-sufficiency	Self-sufficiency record (including cooking)
claude-opus-5-5	1/8	12.5%	Self-sufficiency	Self-sufficiency record (including cooking)
kimi-k3	1/8	12.5%	Self-sufficiency	Self-sufficiency record (including cooking); weapon and armor

Table 9: Activity and survival outcomes in the earlier evaluations. Each cell reports E1 / E2.

Configuration	Days	Crafts	Kills	Bosses	Downs
claude-opus-5	9.9 / 8.4	56 / 57	46 / 36	9 / 8	0 / 1
claude-opus-5-5	9.9 / 8.4	33 / 61	24 / 17	5 / 4	1 / 0
gpt-6-astra	9.8 / 8.5	36 / 37	21 / 19	4 / 4	0 / 0
glm-5.3	9.7 / 8.4	19 / 31	23 / 22	4 / 3	4 / 3
gemini-3.8-flash	9.9 / 8.3	9 / 16	23 / 27	4 / 6	6 / 10
deepseek-v4-flash	9.8 / 9.8	29 / 13	39 / 53	3 / 13	2 / 0
kimi-k3	8.9 / 8.2	14 / 14	22 / 8	3 / 1	3 / 2
gpt-6-sol	10.0 / 8.4	7 / 33	9 / 53	1 / 8	0 / 2
gpt-6-luna	10.0 / 8.5	15 / 10	7 / 17	0 / 1	0 / 0