# CLASHARNESS: BENCHMARKING MULTI-AGENT CONFLICT IN COMPUTER USE VIA A CONTROLLABLE AND VERIFIABLE GENERATION PIPELINE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision-language models (VLMs) are increasingly deployed in multi-agent systems to automate complex computer-use workflows, prompting numerous benchmarks to evaluate and advance agent capabilities. However, existing benchmarks mostly assume consistent, cooperative environments. In contrast, real-world collaboration rarely runs on consistent information: peer or sub-agents frequently report outdated states, issue contradictory instructions, or compete over shared permissions and files, forcing an agent to check others' outputs and verify ground truth in the operating environment. Evaluating such conflict resolution remains challenging: manual task authoring cannot scale across complex systems, while generating tasks via unconstrained LLM generation produces narrative drift, answer leakage, and unsatisfiable grading rubrics that leave tasks unsolvable. We introduce CLASHarness (**C**onflict-**L**oaded **A**gent **S**cenario **H**arness), a controllable framework for generating verifiable multi-agent conflicts in computer use, accompanied by a benchmark of 600 multi-round tasks. By combining hierarchical scenario planning with an eight-step verified micro-pipeline, CLASHarness pre-commits decisive visual facts before rendering and compiles hybrid code and semantic graders directly from executed reference trajectories, guaranteeing that every task is solvable by construction. Evaluating 15 frontier models on CLASHarness reveals three core bottlenecks. First, when outputs of sub-agents conflict with the environment, weaker models blindly follow misleading information, whereas stronger models will inspect visual evidence to verify ground truth. Second, resolving conflicts is much harder than finding them: while agents can spot inconsistencies, actively fixing the system state drops task success significantly. Third, coordination protocols remain a universal bottleneck: even frontier models fail over 58% of closing actions, frequently forgetting to release shared locks or hand off tasks to sub-agents. CLASHarness provides a scalable, verifiable foundation for diagnosing coordination failures and building reliable multi-agent systems. **Code and Dataset will be publicly released.**

## 1 INTRODUCTION

As vision-language models (VLMs) transition to tool-augmented computer-use scenarios, multi-agent systems are increasingly deployed to automate complex collaborative workflows (Li et al., 2023; Wu et al., 2023; Du et al., 2023), prompting numerous benchmarks to evaluate and advance agent capabilities (Zhou et al., 2024; Jimenez et al., 2024; Xie et al., 2024b; Xu et al., 2026). However, existing computer-use benchmarks mostly assume consistent, cooperative single-agent environments (Xie et al., 2024b; Zhou et al., 2024), while multi-agent evaluations remain largely confined to pure-text dialogue (Cemri et al., 2026; Pappu et al., 2026). In contrast, real-world collaboration rarely runs on consistent information: peer or sub-agents frequently report outdated states, issue contradictory instructions, or compete over shared permissions and files, triggering severe coordination failures (Xie et al., 2024a; Xu et al., 2024; Su et al., 2024; Khatua et al., 2026; Zhang et al., 2026). For example, when a sub-agent relies on a stale cache to report a deployment as successful and urges routing production traffic, while live telemetry shows failing health checks, an agent resolving the conflict cannot blindly trust the peer report. Instead, it must check others' outputs, inspect visual dashboards to verify ground truth in the operating environment, and execute a configuration rollback.



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

Evaluating such conflict resolution faces a fundamental challenge in benchmark construction: manual task curation is labor-intensive and cannot scale across complex operating environments (Mialon et al., 2024; Deng et al., 2023), while generating tasks via unconstrained LLM prompting produces uncontrollable scenarios suffering from narrative drift, answer leakage, ungrounded visual assets, and unsatisfiable grading rubrics that leave tasks unsolvable (Zheng et al., 2023; Lù et al., 2025; Peng et al., 2026b).

To address these challenges, we introduce CLASHarness (**C**onflict-**L**oaded **A**gent **S**cenario **H**arness), a controllable generation framework and benchmark for synthesizing verifiable multi-agent conflict tasks in computer use. Instead of relying on unconstrained prompting, CLASHarness follows three core design principles to guarantee scenario validity by design: (1) **Hierarchical Phased Synthesis:** scenario generation is decomposed into macro scenario planning and verified micro-execution, preventing context overflow and drift across rounds; (2) **Pre-Committed Multimodal Grounding:** decisive ground-truth values are committed prior to environment rendering, ensuring visual evidence is truly necessary rather than decorative; and (3) **Execution-Compiled Verification:** hybrid evaluation checkers, combining exact code checks and scoped semantic criteria, are compiled directly from executed reference trajectories, guaranteeing that every task is solvable.

We summarize our principal contributions as follows:

- **Controllable Generation Framework (CLASHarness):** A hierarchical synthesis pipeline combining declarative outline planning with verified micro-execution, compiling hybrid checkers from executed reference trajectories to guarantee every task is solvable without manual scripting (§3).
- **600-Round Multimodal Conflict Benchmark:** An extensive benchmark spanning 20 operational domains, 600 executable conflict rounds, 7,970 pre-committed visual assets, and 3,760 hybrid checkers enforcing grounded visual reasoning (§4).
- **Comprehensive Diagnostic Findings:** Across 15 evaluated models, we uncover three systemic bottlenecks: (1) *Visual Grounding vs. Blind Trust:* weaker models blindly follow misleading information, whereas stronger models inspect visual evidence to verify ground truth (§5.2); (2) *Resolution vs. Detection:* while agents can spot inconsistencies, actively fixing system state drops task success significantly (§5.4); and (3) *Coordination Breakdown:* even frontier models fail over 58% of closing actions, often forgetting to release shared locks or hand off tasks to sub-agents (§5.2).

## 2 Related Work

**Agentic Computer-Use and Multi-Agent Benchmarks.** Recent benchmarks evaluate tool-augmented agents on software engineering, OS execution, and web navigation (Mialon et al., 2024; Zhou et al., 2024; Jimenez et al., 2024; Xie et al., 2024b; Liu et al., 2024; Bonatti et al., 2024; Xu et al., 2026; Yuan et al., 2026; Li et al., 2026a; Jang et al., 2026; Li et al., 2026b), while multi-agent frameworks explore collaborative problem solving (Li et al., 2023; Wu et al., 2023; Du et al., 2023) alongside emerging coordination failure studies (Cemri et al., 2026; Pappu et al., 2026; Khatua et al., 2026; Zhang et al., 2026; Tessera et al., 2026). Whereas existing benchmarks assume internally consistent, cooperative environments, CLASHarness evaluates agents under explicit *factual and temporal* contradictions across tools, files, visual evidence, and peer dialogues.

**Knowledge Conflicts and Multimodal Grounding.** Prior work on knowledge conflicts studies contradictions between model memory and retrieved text in pure-text QA (Xu et al., 2024; Xie et al., 2024a; Su et al., 2024; Wang et al., 2023; 2025; Zhao et al., 2026; Peng et al., 2026a), while multimodal benchmarks emphasize graphical navigation without conflicting signals (Koh et al., 2024; Deng et al., 2023). CLASHarness advances beyond static unimodal evaluations by pre-committing visual ground truths across dashboards, terminal outputs, and system diagrams, and incorporating temporal dynamics where stale evidence and expiring authorizations require grounded action.

**Test-Time Compute and Reasoning Dynamics.** While scaling test-time reasoning compute consistently boosts performance on mathematical and formal domains (Snell et al., 2025; Guo et al., 2025; Muennighoff et al., 2025), open-ended agentic environments can trigger overthinking and repetitive action loops (Chen et al., 2025; Cuadron et al., 2025). CLASHarness provides an empirical testbed evaluating how reasoning effort impacts multi-turn execution under operational conflict.

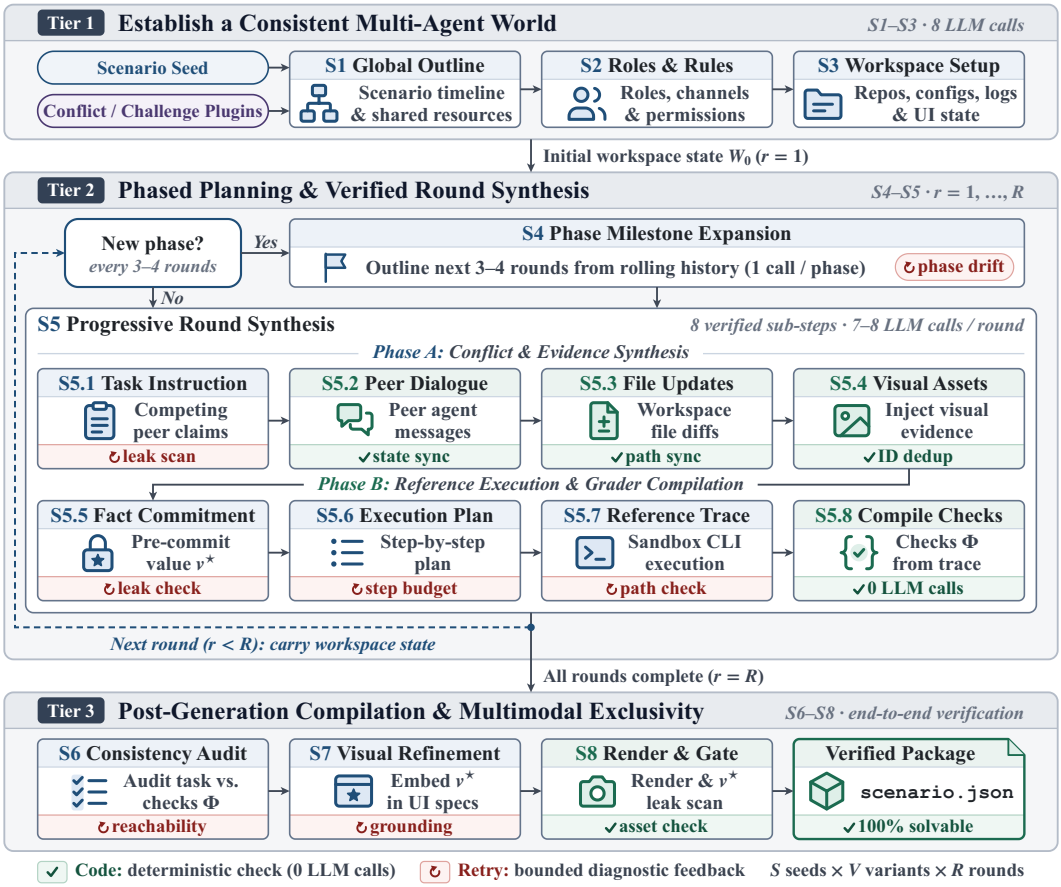



Figure 1: **Hierarchical generation pipeline in CLASHarness.** Tier 1 initializes a shared multi-agent world from seeds and plugins. Tier 2 refreshes phase milestones every 3–4 rounds and generates each round through eight explicit sub-steps. Ground truth $v^\star$ is committed before rendering; executed reference effects ground deterministic checks and scoped semantic criteria. The dashed return path carries workspace state to the next round ($r < R$); after $R = 10$ rounds, generation proceeds to Tier 3 for consistency auditing, visual refinement, and rendering. Visual evidence and checkers share the committed $v^\star$, with text-exclusivity gates enforced when visuals are required. Green tags mark deterministic processing; red tags mark bounded correction with diagnostic feedback. Detailed safeguards appear in Appendix Table B1.

## 3 The Generation Harness

### 3.1 Problem Formalization

We formalize a multi-agent conflict task as a *conflict round* defined by the tuple $\mathcal{R} = (\mathcal{W}_0, I, \mathcal{T}, v^\star, \Phi)$, where $\mathcal{W}_0$ denotes the initial computer-use workspace containing operating system state, file trees, asynchronous multi-agent communication transcripts, and rendered visual evidence. The instruction $I$ specifies the operational objective and explicitly frames competing claims from peer agents without revealing which assertion is grounded in fact. $\mathcal{T}$ represents the typed action space; executing $c \in \mathcal{T}$ transitions environment state and appends structured records to an audit effect log $\mathcal{E}_{\text{log}}$. The *decisive value* $v^\star$ is the ground-truth fact resolving the contradiction. Finally, $\Phi = \{\phi_1, \ldots, \phi_m\}$ is a hybrid evaluation suite combining exact code checks and targeted semantic criteria evaluated over the final workspace state and $\mathcal{E}_{\text{log}}$.

This formulation frames conflict resolution as active investigation and decision-making rather than passive text classification. At inference time, the agent is provided with $(\mathcal{W}_0, I, \mathcal{T})$ and alerted to the contradiction, but must autonomously investigate multimodal evidence across $\mathcal{W}$, identify $v^\star$, and

execute operational state changes (e.g., protocol handovers, configuration patches, rollbacks). Checks $\Phi$ are compiled directly from executed reference trajectories (§3.4), guaranteeing that every task is solvable. Rounds are grouped into 10-round *packages* sharing an evolving workspace, generated from a structured scenario *seed* (§4).

### 3.2 Hierarchical Phased Generation Pipeline

Synthesizing multi-round, multi-agent conflict scenarios poses a fundamental systems challenge. Under single-pass prompting, generator models must simultaneously author complex operational contexts, enforce conflicting agent identities, manipulate multi-file workspaces, and invent evaluation checkers. Single-pass generation frequently causes broken schemas, inconsistent storylines across rounds, leaked answers, and impossible grading checks. CLASHarness resolves these challenges through *hierarchical phased outline planning*, decomposing synthesis into an eight-stage pipeline across three coordinated tiers with in-loop self-correction (Figure 1).

**Tier 1: Establishing a Consistent Multi-Agent World (Stages 1–3).** Realistic conflict scenarios require a coherent operational environment before specific conflicts can be introduced. Rather than synthesizing individual rounds in isolation, Tier 1 establishes the global multi-agent foundation across the entire package from structured scenario seeds and declared plugin effects. First, in Stage 1 (*Global Scenario Outline*), the harness constructs the overall scenario progression, milestone timeline, and shared infrastructure allocations across all subsequent rounds. Building upon this global outline, Stage 2 (*Subagent Roles & Rules*) establishes agent roles, communication channels, and permission boundaries to prevent role drift and unauthorized tool access across rounds. Finally, Stage 3 (*Workspace Setup*) sets up the initial operating workspace, generating code repositories, configuration files, system logs, and UI environments establishing cross-round layout consistency.

**Tier 2: Phased Planning & Verified Round Synthesis (Stages 4–5).** Each scenario package consists of $R = 10$ sequential rounds. Accumulating the complete workspace and dialogue history across all rounds into a single prompt quickly exhausts context limits and causes inconsistencies. CLASHarness resolves this with a two-level planning hierarchy: **High-Level Phase Planning (Stage 4):** The $R$ rounds are partitioned into sequential phases of $P_s \in [3, 4]$ consecutive rounds, such as Rounds 1–3, 4–6, and 7–10. At the start of each phase, Stage 4 (*Phase Milestone Expansion*) outlines high-level milestones for the upcoming 3–4 rounds with a single model call using a rolling window of recent history, maintaining scenario coherence without overloading the prompt.

**Fine-Grained Round Synthesis (Stage 5):** Within each phase, the harness synthesizes each round sequentially through an eight-step verified micro-pipeline in Stage 5 (*Progressive Round Synthesis*): authoring task instructions that frame competing claims while strictly isolating the factual resolution, synthesizing peer agent dialogue history, applying workspace file updates, injecting visual assets, pre-committing the decisive ground-truth value $v^\star$ under automated leakage scanning, drafting step-by-step execution plans, executing reference command-line trajectories in a live sandbox, and compiling exact code checks $\Phi$ directly from recorded state changes without model calls. Workspace modifications from round $r$ persist and carry over to round $r + 1$. Decomposing round synthesis into isolated sub-steps confines potential failures to narrow execution scopes, enabling bounded in-loop retries without discarding valid package state.

**Tier 3: Post-Generation Compilation & Multimodal Exclusivity (Stages 6–8).** To guarantee that compiled benchmarks are verifiably solvable and genuinely multimodal, Tier 3 performs comprehensive end-to-end verification. A cross-stage consistency audit verifies natural-language task instructions against reference trajectories and compiled grading checks, eliminating mismatches between task goals and grading criteria in Stage 6 (*Consistency Audit*). Next, Stage 7 (*Multimodal Refinement*) synthesizes declarative HTML/Canvas specifications for visual evidence assets, adapting each asset to its platform's custom UI layout while embedding the decisive value $v^\star$ alongside realistic operational data. Stage 8 (*Render & Exclusivity Gate*) then renders high-resolution browser screenshots in a headless environment. To prevent agents from bypassing visual inspection via text shortcuts, automated text-leakage scans verify that the decisive value $v^\star$ appears exclusively within rendered visual assets for visually grounded tasks (§3.4), compiling the final verified `scenario.json`.

**Deterministic Normalization and Bounded Self-Correction.** Generating multi-round scenarios often introduces minor formatting inconsistencies or logical conflicts. Discarding and regenerating

an entire 10-round package upon encountering an error would be slow and expensive. Instead, CLASHarness resolves generation defects through a two-tier self-correction mechanism (Figure 1): First, **deterministic code normalization** corrects formatting, syntax, and schema errors—such as missing fields or invalid file paths—directly in code with zero model invocations (green tags in Figure 1). Second, **bounded semantic retries** resolve logical mismatches: when inconsistencies arise between task instructions and grading checks, the harness confines retries strictly to the affected sub-step, providing diagnostic error feedback (red tags in Figure 1) rather than restarting the package. Because retries are restricted to isolated sub-steps, total API calls scale linearly with round count: Total $= 8 + \lceil R/P_s \rceil + K \cdot R + A$, where $K \approx 7$–$8$ calls per round and $A$ is the visual asset count, averaging 95 calls per 10-round package (Appendix §B.2). This makes benchmark synthesis scalable and cost-effective while ensuring all generated tasks are solvable.

### 3.3 Declarative Controllability and Modular Plugins

**Structured Seed Specifications.** Scenario structure is defined via declarative seed schemas governing conflict types, interaction protocols, tool families, and evidence modalities, allowing users to customize parameters and synthesize new domains on demand (Appendix §B.4).

**Modular Plugins and State Effects.** To support modular extensions without modifying evaluation engines, CLASHarness separates domain mechanics into 23 modular plugins (16 conflict mechanics, 7 challenge plugins; Table C1). Each plugin $\mathcal{P}$ and tool command $c \in \mathcal{T}$ declares operational effects $\mathcal{E}(\mathcal{P})$ and $\mathcal{E}(c)$, such as file edits or permission changes. Valid actions and grading criteria match declared effects: $\mathcal{T}_{\text{valid}} = \{c \in \mathcal{T} \mid \mathcal{E}(c) \cap \mathcal{E}(\mathcal{P}) \neq \emptyset\}$. Users can synthesize new conflicts or custom APIs by authoring a seed schema and registering state effects, without procedural simulation.

### 3.4 Verifiable Grading and Multimodal Grounding

**Executable Reference Grader Compilation.** In unconstrained generation, natural-language rubrics often specify impossible requirements. CLASHarness compiles checkers directly from executed reference trajectories: exact-match checks verify specific identifiers, while semantic criteria evaluate scoped file modifications (Appendix §B.7), eliminating unreachable checks across the benchmark.

**Multimodal Pre-Commitment & Exclusivity Gates.** Mirroring cross-boundary ticket attachments, point-in-time incident snapshots, and physical sensor captures where underlying DOM or source access is unavailable, visual assets are provided as standalone image files. Committing $v^\star$ *prior* to rendering aligns visual evidence with compiled checkers. Across 600 rounds (§4), 346 require visual evidence (`role="required"`), with automated leakage scans ensuring $v^\star$ appears exclusively in rendered images rather than text.

## 4 The CLASHarness Benchmark: Scale, Diversity, and Validity

**Benchmark Scale and Structural Diversity.** Using CLASHarness, we construct a primary evaluation benchmark comprising 20 structured scenario seeds across diverse operational domains, 60 package variants (3 variants per seed), and 600 executable computer-use conflict rounds ($R = 10$). Across these 600 rounds, the benchmark spans 3,760 evaluation checks (3,284 required) and 7,970 rendered visual evidence assets. Reference trajectories average 10.28 steps against a communicated step budget $b = 20$ (Appendix §C.1). Task difficulty is empirically measured by reference trajectory length (7–18 steps; $b/\text{steps} \in [1.11, 2.86]$). Synthesizing the 20 seeds with an alternate model family yields an aligned parallel benchmark with low trajectory correlation (Pearson $r = 0.186$, avg. 12.48 steps; §6), confirming that the generator shapes execution complexity while seeds govern task semantics. Round diversity is strictly enforced: candidate $v^\star$ values cannot reuse past answers, and only 1.6% of semantic criteria repeat within a package (2,086 of 2,137 criteria are globally distinct; Appendix §B.9).

**Hybrid Rubric Composition and Evaluator Stability.** Of the 3,284 scored checks, 34.9% (1,147) are deterministic code checks and 65.1% (2,137) are targeted semantic criteria evaluated by an LLM judge (Table C2). Every semantic check is scored via 3-sample self-consistency majority voting, achieving 97.5% unanimous agreement across 19,419 recorded judgments. Re-grading all high-effort

Table 1: **Main evaluation results across 15 models on CLASHarness (600 rounds, reported as mean ± SE).** *Pass Rate*: percentage of tasks satisfying all required checks. *Mean Score*: average percentage of individual checks satisfied per round. Task difficulty is categorized by ground-truth reference trajectory length (*Easy*: $\leq 9$ steps, *Mid*: 10–11 steps, *Hard*: $\geq 12$ steps). Best results are in **bold**, second-best underlined. †Text-only models evaluated using OCR transcripts of visual assets. Appendix Tables D1–D3 provide full 95% CIs and bootstrap details.

| Model | Think Level | Pass Rate (%) | | | | Mean Score (%) | | | | Avg. Cost | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | All | Easy | Mid | Hard | All | Easy | Mid | Hard | $ | Tok (M) |
| Claude Opus 5 | High | 25.2±1.8 | 32.2±3.0 | 25.0±2.9 | 13.6±2.9 | 64.2±1.2 | 67.4±1.9 | 68.5±1.8 | 51.8±2.6 | 2,631 | 511 |
| | Med | 25.3±1.8 | 33.9±3.1 | 24.6±2.9 | 12.1±2.7 | 63.3±1.2 | 68.3±1.9 | 66.5±1.9 | 49.7±2.5 | 2,245 | 439 |
| | Low | 23.5±1.7 | 30.5±3.0 | 23.7±2.9 | 11.4±2.7 | 60.9±1.3 | 66.2±1.9 | 64.3±1.9 | 46.5±2.6 | 1,848 | 364 |
| Claude Opus 4.8 | High | 22.8±1.7 | 30.1±3.0 | 23.7±2.8 | 9.3±2.4 | 62.1±1.2 | 65.6±2.0 | 67.4±1.8 | 47.7±2.5 | 2,933 | 571 |
| | Med | 23.8±1.7 | 29.7±3.0 | 24.6±2.9 | 12.9±2.8 | 61.8±1.2 | 65.1±2.0 | 66.6±1.9 | 48.5±2.5 | 2,646 | 516 |
| | Low | 21.3±1.7 | 28.0±2.9 | 21.4±2.7 | 10.0±2.6 | 60.8±1.2 | 64.0±2.0 | 65.8±1.9 | 47.3±2.5 | 2,296 | 449 |
| Claude Sonnet 5 | High | 20.8±1.7 | 28.0±2.9 | 21.4±2.7 | 7.9±2.3 | 59.6±1.3 | 63.9±2.0 | 64.1±1.9 | 45.3±2.5 | 1,031 | 497 |
| | Med | 17.0±1.5 | 25.4±2.8 | 17.0±2.5 | 2.9±1.4 | 57.0±1.3 | 61.5±2.0 | 61.8±2.0 | 41.7±2.4 | 836 | 405 |
| | Low | 13.8±1.4 | 20.8±2.6 | 12.9±2.3 | 3.6±1.6 | 53.7±1.2 | 59.1±1.9 | 55.8±1.9 | 41.3±2.3 | 708 | 345 |
| GPT-6 Astra | High | **28.7**±1.9 | **35.2**±3.1 | **28.1**±3.1 | **18.6**±3.3 | **66.8**±1.2 | **68.9**±2.0 | **70.1**±1.8 | **58.2**±2.6 | 3,003 | 663 |
| GPT-5.6 Sol | High | 24.5±1.7 | 31.4±3.0 | 24.1±2.9 | 13.6±2.9 | 63.3±1.2 | 66.7±1.9 | 66.8±1.8 | 51.7±2.6 | 2,165 | 454 |
| | Med | 23.0±1.7 | 29.2±3.0 | 24.6±2.9 | 10.0±2.5 | 62.8±1.2 | 66.2±1.9 | 66.4±1.9 | 51.1±2.4 | 1,939 | 406 |
| | Low | 20.8±1.7 | 26.3±2.9 | 22.8±2.8 | 8.6±2.4 | 61.0±1.2 | 64.4±1.9 | 65.6±1.9 | 48.1±2.4 | 1,582 | 330 |
| GPT-5.6 Terra | High | 20.0±1.6 | 27.1±2.9 | 20.1±2.7 | 7.9±2.3 | 60.2±1.2 | 63.9±2.0 | 64.2±1.9 | 47.7±2.4 | 731 | 382 |
| | Med | 20.0±1.6 | 25.8±2.9 | 20.1±2.7 | 10.0±2.5 | 58.9±1.3 | 62.8±2.0 | 61.7±2.0 | 47.9±2.5 | 636 | 332 |
| | Low | 17.7±1.6 | 23.7±2.8 | 17.9±2.6 | 7.1±2.2 | 57.2±1.2 | 61.9±1.9 | 60.0±2.0 | 44.6±2.5 | 563 | 293 |
| GPT-5.6 Luna | High | 19.8±1.6 | 23.7±2.7 | 22.3±2.8 | 9.3±2.5 | 60.4±1.2 | 62.0±2.0 | 65.0±1.9 | 50.1±2.5 | 86 | 450 |
| | Med | 17.7±1.6 | 22.0±2.7 | 19.2±2.7 | 7.9±2.3 | 59.1±1.2 | 62.1±1.9 | 62.2±1.9 | 49.1±2.5 | 68 | 357 |
| | Low | 19.5±1.6 | 25.0±2.8 | 18.3±2.6 | 12.1±2.7 | 59.0±1.2 | 62.1±1.9 | 62.3±1.9 | 48.4±2.6 | 59 | 305 |
| Gemini 3.7 Flash | High | 20.3±1.6 | 26.7±2.9 | 21.9±2.8 | 7.1±2.2 | 58.1±1.3 | 60.8±2.1 | 63.0±2.0 | 45.9±2.3 | 307 | 395 |
| | Med | 21.5±1.7 | 27.1±2.9 | 24.1±2.9 | 7.9±2.3 | 61.8±1.2 | 65.1±1.9 | 65.5±1.9 | 50.2±2.4 | 250 | 324 |
| | Low | 20.5±1.6 | 27.1±2.9 | 20.5±2.7 | 9.3±2.4 | 60.0±1.2 | 64.5±2.0 | 63.4±1.9 | 47.0±2.5 | 203 | 265 |
| Gemini 3.6 Flash | High | 18.5±1.6 | 24.6±2.8 | 21.4±2.7 | 3.6±1.6 | 58.6±1.3 | 61.7±2.0 | 63.5±2.0 | 45.6±2.2 | 565 | 363 |
| | Med | 20.2±1.6 | 27.1±2.9 | 21.9±2.8 | 5.7±2.0 | 60.2±1.2 | 63.1±2.0 | 64.7±1.9 | 48.2±2.4 | 508 | 327 |
| | Low | 18.5±1.6 | 23.3±2.7 | 19.2±2.6 | 9.3±2.5 | 59.5±1.2 | 62.0±1.9 | 64.2±1.9 | 48.0±2.5 | 323 | 211 |
| Gemini 3.1 Pro | High | 12.0±1.3 | 15.7±2.4 | 12.1±2.2 | 5.7±2.0 | 48.7±1.3 | 49.3±2.1 | 53.6±2.0 | 39.9±2.3 | 578 | 280 |
| | Med | 9.8±1.2 | 11.4±2.1 | 10.3±2.1 | 6.4±2.1 | 47.9±1.2 | 47.9±2.0 | 53.5±2.0 | 39.0±2.4 | 490 | 239 |
| | Low | 8.3±1.1 | 10.2±2.0 | 8.5±1.8 | 5.0±1.9 | 46.3±1.2 | 48.0±1.9 | 50.3±1.9 | 36.9±2.3 | 436 | 213 |
| Gemini 3.5 Flash | High | 3.0±0.7 | 3.8±1.2 | 2.7±1.1 | 2.1±1.2 | 25.9±1.1 | 21.6±1.7 | 29.8±1.7 | 27.2±2.1 | 712 | 445 |
| | Med | 3.2±0.7 | 3.8±1.2 | 3.6±1.2 | 1.4±1.0 | 29.9±1.1 | 27.3±1.8 | 32.3±1.8 | 30.5±2.1 | 703 | 440 |
| | Low | 13.5±1.4 | 17.4±2.4 | 14.3±2.4 | 5.7±2.0 | 47.3±1.3 | 45.9±2.3 | 52.6±2.1 | 40.8±2.3 | 547 | 349 |
| Kimi K3 | High | 21.8±1.7 | 26.3±2.8 | 23.2±2.9 | 12.1±2.7 | 55.9±1.4 | 55.8±2.3 | 61.6±2.2 | 47.0±2.7 | 2,173 | 706 |
| DeepSeek V4 Pro† | High | 19.8±1.6 | 24.2±2.8 | 20.5±2.7 | 11.4±2.7 | 58.1±1.3 | 58.8±2.1 | 64.4±1.9 | 46.9±2.6 | 32 | 70 |
| GLM-5.2 FP8† | High | 14.7±1.4 | 18.2±2.5 | 16.5±2.5 | 5.7±2.0 | 48.7±1.3 | 48.2±2.3 | 55.0±2.1 | 39.4±2.4 | 120 | 82 |
| DeepSeek V4 Flash† | High | 8.5±1.1 | 8.5±1.8 | 11.2±2.1 | 4.3±1.7 | 37.5±1.3 | 34.5±2.2 | 42.7±2.2 | 34.4±2.3 | N/A | 95 |

rounds using independent frontier judges (Claude Sonnet 5, Claude Opus 5) shows high consistency across judges: relative model rankings remain nearly identical ($\rho \geq 0.98$, Cohen's $\kappa \geq 0.90$; Appendix Table F4). In addition, primary diagnostic findings (visual grounding F3, evidence anchoring F4, terminal actions F5, temporal deadlines F6) are evaluated strictly via deterministic code checks, ensuring conclusions hold independently of model judging.

**Cross-Metric Consistency and Capability Decoupling.** Model rankings under code and semantic rubrics closely agree ($\rho = +0.90$). Importantly, individual check evaluations reveal that visual evidence localization and semantic conflict resolution have almost zero correlation ($\rho = +0.02$). This confirms that visual grounding and conflict resolution are independent capabilities requiring separate evaluation rather than composite scores.

# 5 Behavioral Analysis

## 5.1 Main Evaluation Results: End-to-End Conflict Resolution Performance

**Why Conflict Scenarios Break Frontier Models: The Competence-Resolution Gap.** Table 1 shows why multi-agent conflicts are much harder than standard single-agent tasks. In our scenarios, peer or sub-agents give misleading or outdated text reports (e.g., claiming a broken deployment is healthy), shared resources are locked by competing agents, and system states change asynchronously. To pass a task, an agent must distrust false peer claims, verify ground truth from visual dashboards,

Table 2: **Failure concentration and error correlation dynamics (High Effort).** Measures whether execution errors cascade into task failure. *Co-Failure Lift* $> 1.0$ indicates correlated mistakes: on easy tasks, early errors typically trigger complete task failure, whereas hard tasks exhibit gradual, independent error accumulation (lift $\approx 1.0$). *Indep.* gives expected pass rates under independent check failures; asterisks ($^*$) denote statistically significant clustering.

| | Entire Benchmark (%) | | | Co-Failure Lift | | | | Lift − Peer Mean | |
|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Check** | **Pass** | **Indep.** | **All (Ceil.)** | **Easy** | **Mid** | **Hard** | **Δ** | **95% CI** |
| Claude Opus 5 | 63.0 | 25.2 | 10.8 | 1.49 (2.70) | 2.10$^*$ | 1.37$^*$ | 1.04 | +0.188 | [+0.110, +0.268] |
| Claude Opus 4.8 | 60.9 | 22.8 | 9.3 | 1.41 (2.56) | 1.95$^*$ | 1.04 | 1.02 | +0.102 | [+0.054, +0.153] |
| Claude Sonnet 5 | 58.2 | 20.8 | 7.7 | 1.39 (2.39) | 1.71$^*$ | 1.16$^*$ | 1.04$^*$ | +0.081 | [+0.034, +0.131] |
| GPT-6 Astra | **66.7** | **28.7** | **13.8** | 1.47 (3.00) | 1.42 | 0.96 | 1.01 | +0.149 | [+0.070, +0.229] |
| GPT-5.6 Sol | 62.2 | 24.5 | 10.2 | 1.42 (2.65) | 1.09 | 1.00 | 1.03 | +0.116 | [+0.059, +0.176] |
| GPT-5.6 Terra | 59.3 | 20.0 | 8.3 | 1.37 (2.46) | 1.15 | 0.99 | 1.02 | +0.062 | [+0.019, +0.106] |
| GPT-5.6 Luna | 59.8 | 19.8 | 8.6 | 1.36 (2.49) | 1.53$^*$ | 0.96 | 1.02 | +0.047 | [+0.002, +0.094] |
| Gemini 3.7 Flash | 57.5 | 20.3 | 7.3 | 1.30 (2.35) | 1.53$^*$ | 1.11$^*$ | 1.00 | −0.024 | [−0.081, +0.030] |
| Gemini 3.6 Flash | 58.0 | 18.5 | 7.6 | 1.28 (2.38) | 1.34$^*$ | 1.06 | 0.99 | −0.036 | [−0.092, +0.017] |
| Gemini 3.1 Pro | 48.8 | 12.0 | 3.8 | 1.19 (1.95) | 1.38$^*$ | 0.99 | 1.01 | −0.136 | [−0.175, −0.097] |
| Gemini 3.5 Flash | 28.1 | 3.0 | 0.5 | 1.05 (1.39) | 1.08$^*$ | 1.01 | 0.99$^*$ | −0.288 | [−0.338, −0.242] |
| Kimi K3 | 55.8 | 21.8 | 6.5 | 1.40 (2.26) | 2.26$^*$ | 1.17$^*$ | 1.06$^*$ | +0.087 | [+0.037, +0.141] |
| DeepSeek V4 Pro | 57.7 | 19.8 | 7.4 | 1.35 (2.36) | 1.29 | 1.09$^*$ | 1.02 | +0.031 | [−0.013, +0.074] |
| GLM-5.2 FP8 | 48.2 | 14.7 | 3.6 | 1.27 (1.93) | 1.81$^*$ | 1.16$^*$ | 1.00 | −0.053 | [−0.099, −0.008] |
| DeepSeek V4 Flash | 38.8 | 8.5 | 1.6 | 1.16 (1.63) | 1.49$^*$ | 1.06$^*$ | 1.00 | −0.177 | [−0.222, −0.133] |

override the faulty state, and complete strict closing handovers. Because models must satisfy every step in this chain, low pass rates do not mean complete failure: frontier models pass most individual checks ($55\%$–$67\%$), led by GPT-6 Astra at $66.8\%$. With deeper exploration (averaging 26.1 steps) and strong visual grounding, GPT-6 Astra handles complex conflict trajectories better than baseline models ($18.6\%$ on hard tasks vs. $13.6\%$ for Claude Opus 5). Yet even GPT-6 Astra solves only $28.7\%$ of tasks overall. Rarely hitting its execution limit ($1.0\%$), it frequently terminates prematurely on its own without finishing the required multi-agent cleanup—such as releasing shared locks or handing off tasks to sub-agents.

**Diminishing and Inconsistent Returns from Test-Time Compute.** While test-time reasoning compute boosts formal math benchmarks, Table 1 shows that in multi-agent conflict environments, scaling effort from Low to Medium to High yields diminishing returns or hurts performance. For top models (Claude Opus 5, GPT-5.6 Luna), increasing effort from Low to High brings minimal gains ($+1.7$ pp and $+0.3$ pp), leveling off at Medium effort. For flash models (Gemini 3.6 Flash, Gemini 3.7 Flash), Medium effort achieves peak accuracy ($20.2\%$, $21.5\%$) while High slightly drops. Worst of all, on Gemini 3.5 Flash, higher effort causes pass rates to drop sharply from $13.5\%$ to $3.0\%$, as internal thinking loops crowd out tool actions within the step budget.

### 5.2 Failure Taxonomy: Content Accuracy vs. Protocol Execution

To diagnose why agents fail under conflict, we map all $3{,}284$ evaluation checks per model into three distinct competencies spanning six diagnostic failure modes (Appendix §D.6): **Visual Grounding**, inspecting visual conflict evidence (F3); **Content Accuracy**, covering messaging (F1), file modifications (F2), and evidence citations (F4); and **Structural Protocol Compliance**, executing terminal actions (F5) and synchronizing time windows (F6).

**Resolving Text-Visual Contradictions vs. Multi-Agent Synchronization.** First, **visual grounding vs. blind trust separates strong and weak models**: when sub-agent outputs conflict with the environment, weaker models blindly follow misleading text, whereas stronger models inspect visual evidence to verify ground truth and cite exact conflict evidence (outperforming weaker baselines by 31–65 pp, with evidence citation F4 showing a 15.2 pp spread even among top models; Appendix §D.2). Second, **coordination protocols remain a universal bottleneck**: when conflicts involve competing locks, permission boundaries, or expiring holds, models struggle to execute closing actions (F5, $58\%$–$72\%$ failure rate) or synchronize time windows (F6, $83\%$–$97\%$ failure rate). Even GPT-6 Astra and Claude Opus 5 fail over $58\%$ of closing actions, frequently forgetting to release shared locks

or hand off tasks to sub-agents. Finally, statistical tests across conflict categories show no domain specialization (Bonferroni-corrected $p \geq 0.094$; Appendix §D.4), confirming that these failures reflect core multi-agent coordination bottlenecks.

### 5.3 Failure Concentration and Error Dynamics

While models achieve similar overall pass rates, they fail in fundamentally different ways: do errors trigger a chain reaction, or do models fail separate checks independently? To answer this, we evaluate *co-failure lift*—the likelihood that a second check fails given that one has already failed, relative to independent chance (Table 2; Appendix §D.3). Comparing tasks by ground-truth step length reveals two distinct patterns (Table 1): on short tasks (9 or fewer steps), errors trigger a chain reaction, with 11 of 15 models exhibiting co-failure lift significantly above 1.0, indicating that an early mistake causes the entire task to fail. In contrast, on long tasks (12 or more steps), lift drops near 1.0 (0.99–1.06), showing failures on long tasks stem from gradual mistakes rather than single fatal errors.

### 5.4 The Conflict-Resolution Gate: Isolating Resolution from Investigation

To separate contradiction detection from operational execution, CLASHarness evaluates two distinct task demands: rounds where agents must resolve conflicts through state-changing actions (*Resolve-gated*), and rounds with identical conflicts requiring only investigation (*Investigate-only*). Control tasks confirm that performance drops stem from operational execution rather than context growth.

As shown in Table 3, requiring active conflict resolution drops pass rates from $28.1\%$ to $11.0\%$ ($-17.1$ pp), while control performance is unchanged ($+1.1$ pp), yielding a net difference-in-differences penalty of $\mathbf{-18.2}$ pp (95% CI $[-29.1, -7.2]$, exact $p = .052$). Pass rates drop from $28.1\%$ under investigation alone, to $12.3\%$ under partial execution, falling to $4.4\%$ upon full protocol completion.

**Resolution vs. Detection.** On mean check score, the gap between resolving and investigating is only $-7.9$ pp ($p = .252$): while agents easily spot inconsistencies, task success drops by $-18.2$ pp when they must actively fix the system state by reverting changes, revoking permissions, or handing off locks (Appendix §D.5).

Table 3: **The conflict-resolution gate.** Pass rates when tasks require actively *resolving* a conflict compared to merely *investigating* it. Treated rounds require state-changing operational protocols, whereas control rounds require only passive diagnosis over identical round positions. High Effort, 15 evaluated models, 9,000 rounds (details in Table D5, Appendix §D.5).

| | Pass rate (%) | | |
|---|---|---|---|
| **Seed arm** | **Resolve** | **Investigate** | **Δ** |
| Treated (12 seeds) | 11.0 | 28.1 | −17.1 |
| Control (8 seeds) | 20.4 | 19.3 | +1.1 |
| **Difference-in-differences** | | | **−18.2** |
| 95% CI (seed bootstrap) | | | [−29.1, −7.2] |
| Exact test $p$ | | | .052 |

### 5.5 Compute Efficiency, Reasoning Effort, and Economic Costs

Examining test-time compute reveals an *effort inversion*: reasoning token share spans from $8.8\%$ for Claude Opus 5 to $90.5\%$ for Gemini 3.5 Flash (Table 1), yet correlates negatively with task success across the 20-step API models ($\rho = -0.697$). Rather than improving planning, extra thinking traps weaker models in repetitive loops: while top models rarely run out of steps (hitting the 20-step cap on only $0.5\%$ of rounds for Claude Opus 5 and $1.2\%$ for GPT-5.6 Sol), Gemini 3.5 Flash runs out of steps on $89.2\%$ of rounds (Appendix Table E1). This planning failure directly drives high economic cost: across all models at high effort ($11,737 across 87,493 calls), cost per solved task spans a 55-fold range, from $0.72 for GPT-5.6 Luna to $39.54 for Gemini 3.5 Flash (Appendix §E.1).

## 6 Ablations

**Cross-Generator Stability and Task Difficulty.** To test whether findings depend on the scenario generator, we synthesized an alternate suite using Claude Sonnet 5 alongside default Gemini 3.6 Flash (Figure 2c). Although Sonnet yields longer reference trajectories (12.5 vs. 10.3 steps) and lower pass rates, model rankings stay consistent ($\rho = 0.96$) without self-preference (Appendix §F.1).

**Action Budget Limits and Planning Horizons.** Varying step budgets from 20 to 40 steps (Figure 2a)

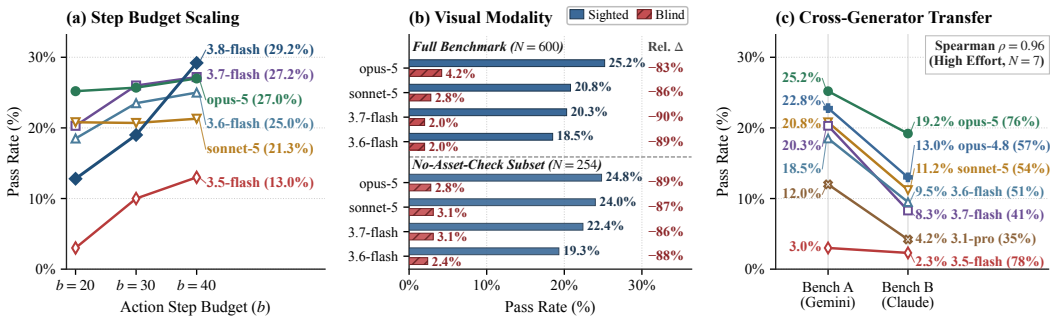



Figure 2: **Harness parameter ablations.** *(a) Step budget:* Expanding action horizons unlocks frontier models with high exploration demands while separating efficient planners from thinking loops. *(b) Visual modality:* Agent performance collapses without image inputs, confirming visual evidence is essential for conflict resolution. *(c) Generator model:* Relative rankings remain highly consistent across parallel benchmarks synthesized by different generator models.

reveals three planning patterns: First, **step-efficient models** (Claude Opus 5, Sonnet 5) solve tasks concisely (mean 10.3 steps), rarely hitting the 20-step cap ($< 3.3\%$) with minimal gain from extra steps ($< 2$ pp). Second, **loop-prone models** (Gemini 3.5 Flash) hit the cap on $89.2\%$ of rounds due to repetitive tool calls, inflating cost to \$8.69 per solved task. Third, **horizon-sensitive models** benefit from extended exploration: with deeper exploration, GPT-6 Astra reaches $\mathbf{28.7}\%$, while Gemini 3.8 Flash jumps from $12.8\%$ (20 steps) to $\mathbf{29.2}\%$ (40 steps) and $\mathbf{30.7}\%$ at 60 steps ($p = .0098$), cutting cost per solved task by $21\%$ to \$3.50 (Appendix §F.3).

**Visual Necessity: Sighted vs. Blind Evaluation.** Withholding visual evidence causes agent performance to collapse across all models (Figure 2b). Even on tasks where image inspection is not explicitly enforced by the rubric, blind agents retain only $11.1\%$–$13.1\%$ of sighted pass rates ($p < 10^{-11}$), confirming visual assets provide decisive rather than decorative evidence (Appendix §F.4).

## 7 Limitations

We note three scope limitations: (1) **Semantic Judging:** Semantic checks rely on an LLM judge without human studies, mitigated via 3-sample voting and multi-judge validation ($\rho \geq 0.98$, $\kappa \geq 0.90$; §4). (2) **Asynchronous Traces:** Conflict resolution is evaluated through logs and workspace traces rather than live chatting, ensuring reproducibility while mirroring ticket and review workflows. (3) **Diagnostic Focus:** Scenarios focus on conflict rounds without conflict-free baselines, isolating conflict handling via ablation gates (§5.4).

## 8 Conclusion

In this work, we introduced CLASHarness, a controllable generation framework and benchmark for synthesizing verifiable multi-agent conflict scenarios in computer use. By combining hierarchical outline planning with verified micro-execution, pre-committed visual ground truths, and execution-compiled hybrid checkers, CLASHarness guarantees task solvability by construction across 600 multi-round tasks and 20 operational domains. Our evaluation of 15 frontier models uncovers three fundamental bottlenecks: weaker models blindly trust misleading peer reports whereas stronger models actively inspect visual evidence to verify ground truth; actively fixing the system state is much harder than spotting inconsistencies; and closing coordination protocols—such as releasing shared locks and completing sub-agent handoffs—remain a universal failure mode across all model families. Beyond static evaluation, CLASHarness's modular plugin architecture and automated verification pipeline provide both a rigorous diagnostic testbed and a scalable data engine for post-training next-generation VLMs to build reliable multi-agent systems.

## AI USE STATEMENT

Language models served as generation engines in the CLASHarness hierarchical pipeline to expand structured scenario seeds for benchmark data generation under deterministic validation and reference-trajectory compilation (§3, §4). Language models were also used for language polishing during manuscript preparation, with all claims verified by the human authors.

## REPRODUCIBILITY STATEMENT

The generation harness, conflict plugins, asset generators, and evaluation checkers are fully specified with deterministic seeds and reproducible reference trajectories (Appendix B), with tasks and grading surfaces detailed in Appendix C. Every reported quantity is generated by released analysis scripts and verified mechanically: all table bodies are byte-diffed against script outputs, and every statistic in the prose of §4–§6 is programmatically asserted to prevent inconsistencies. A single validation command recomputes all counts from raw source files and flags any numerical drift.

## ETHICS STATEMENT

CLASHarness evaluates agent conflict resolution on synthetic technical and enterprise scenarios containing no private personal data, credentials, or harmful content. Evaluating how agents detect, investigate, and resolve conflicting information is critical for deploying reliable and safe AI in high-stakes multi-agent environments.

## REFERENCES

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, et al. Windows agent arena: Evaluating multi-modal os agents at scale. *arXiv preprint arXiv:2409.08264*, 2024.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38, 2026.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*, 2025.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Lawrence Keunho Jang, Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Odysseys: Benchmarking web agents on realistic long horizon tasks. *arXiv preprint arXiv:2604.24964*, 2026.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

Arpandeep Khatua, Hao Zhu, Peter Tran, Arya Prabhudesai, Frederic Sadrieh, Johann K Lieberwirth, Xinkai Yu, Yicheng Fu, Michael J Ryan, Jiaxin Pei, et al. Cooperbench: Why coding agents cannot be your teammates yet. *arXiv preprint arXiv:2601.13295*, 2026.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

Jinchao Li, Yunxin Li, Chenrui Zhao, Zhenran Xu, Baotian Hu, and Min Zhang. Windowsworld: A process-centric benchmark of autonomous gui agents in professional cross-application environments. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15262–15280, 2026a.

Wanli Li, Bowen Zhou, Yunyao Yu, Zhou Xu, Yifan Yang, Dongsheng Li, and Caihua Shan. Weavebench: A long-horizon, real-world benchmark for computer-use agents with hybrid interfaces. *arXiv preprint arXiv:2606.09426*, 2026b.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J Pal, and Siva Reddy. Agentrewardbench: Evaluating automatic evaluations of web agent trajectories. *arXiv preprint arXiv:2504.08942*, 2025.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pp. 9025–9049, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Aneesh Pappu, Batu El, Hancheng Cao, Carmelo di Nolfo, Yanchao Sun, Meng Cao, and James Zou. Multi-agent teams hold experts back. *arXiv preprint arXiv:2602.01011*, 2026.

Huang Peng, Jiuyang Tang, Weixin Zeng, Hao Xu, and Xiang Zhao. Navigating unreliable parametric and contextual knowledge: Explicit knowledge conflict resolution for llm inference. *arXiv preprint arXiv:2606.20245*, 2026a.

Yangda Peng, Yunjia Qi, Hao Peng, Haotian Xia, Guanzhong He, Xintong Shi, Richeng Xuan, Songyuanyi Lu, Yixian Liu, Zhichao Hu, et al. Can llm-as-a-judge reliably verify rubrics in agentic scenarios? *arXiv preprint arXiv:2606.29920*, 2026b.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llm. *arXiv preprint arXiv:2408.12076*, 2024.

Kale-ab Abebe Tessera, Andras Szecsenyi, Cameron Barker, Alexander Rutherford, Davide Paglieri, Aidan Scannell, Henry Gouk, Elliot J Crowley, Tim Rocktäschel, and Amos Storkey. Benchmarking open-ended multi-agent coordination in language agents. *arXiv preprint arXiv:2606.08340*, 2026.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. *arXiv preprint arXiv:2504.13079*, 2025.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Resolving knowledge conflicts in large language models. *arXiv preprint arXiv:2310.00935*, 2023.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, volume 2024, pp. 35623–35646, 2024a.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024b.

Frank Fangzheng Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *Advances in Neural Information Processing Systems*, 38, 2026.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565, 2024.

Mengqi Yuan, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, et al. Osworld2. 0: Benchmarking computer use agents on long-horizon real-world tasks. *arXiv preprint arXiv:2606.29537*, 2026.

Yuzhe Zhang, Feiran Liu, Yi Shan, Xinyi Huang, Xin Yang, Yueqi Zhu, Xuxin Cheng, Cao Liu, Ke Zeng, Terry Jingchen Zhang, et al. Silo-bench: A scalable environment for evaluating distributed coordination in multi-agent llm systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29379–29398, 2026.

Tianzhe Zhao, Jiaoyan Chen, Shuxiu Zhang, Haiping Zhu, Qika Lin, and Jun Liu. Exploring knowledge conflicts for faithful llm reasoning: Benchmark and method. *arXiv preprint arXiv:2604.11209*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

# A Overview and Table of Contents of Supplementary Material

This supplementary document provides comprehensive algorithmic specifications, formal derivations, dataset statistics, extended benchmark leaderboards, and ablation studies supporting the findings presented in the main paper. In addition to this PDF document, the accompanying Supplementary Material archive includes standalone interactive HTML scenario visualizers (`index.html`) for exploring multi-round conflict dependency graphs, workspace state diffs, sub-agent communication transcripts, native $3{,}840 \times 2{,}160$ visual assets, and compiled evaluation rubrics across all 20 benchmark seed families.

**Table of Contents of Supplementary Material:**


- **Appendix B: The CLASHarness Generation Pipeline: Algorithmic and Implementation Details**
  - §B.1: Problem Formalization and Grounded Locality Axioms
  - §B.2: Hierarchical Phased Generation Pipeline ($O(R)$ API Complexity)
  - §B.3: Harness Generation Safeguards (`M1`–`M14`; Table B1)
  - §B.4: Declarative Controllability and 3-Tier Constraint Enforcement
  - §B.5: Modular Plugins and State Effects for Extensibility
  - §B.6: Multimodal Asset Generation and Pre-Commitment (Figure B1)
  - §B.7: Verifiable Grader Compilation and Exclusivity Gates
  - §B.8: A Worked Round, end to end (Figures B3, B4; Table B2)
  - §B.9: Task Diversity and State Continuity
- **Appendix C: Benchmark Composition, Coverage, and Validity**
  - §C.1: Benchmark Coverage and Structural Diversity (Table C1)
  - §C.2: Continuous Measured Difficulty via Ground-Truth Trajectory Length
  - §C.3: Hybrid Grading Validity, Judge Integrity, and Competency Decoupling (Table C2)
- **Appendix D: Comprehensive Experimental Results and Baseline Diagnostics**
  - §D.1: Multi-Effort Evaluation Leaderboards (Tables D1, D2, D3)
  - §D.2: Pairwise Statistical Significance and Frontier Compression
  - §D.3: Failure Concentration Dynamics
  - §D.4: Capability Breakdown by Conflict Category (Table D4)
  - §D.5: The Conflict-Resolution Gate: Robustness, Replication, and Compliance (Tables D5, D6)
  - §D.6: Fine-Grained Diagnostic Failure Taxonomy (Figure D1)
- **Appendix E: Compute Efficiency, Token Economics, and Planning Horizons**
  - §E.1: Economic Cost and API Breakdown (Table E1)
  - §E.2: Reasoning Effort Inversion Dynamics ($\rho = -0.697$)
  - §E.3: Execution Step Budget Saturation
- **Appendix F: Extended Ablation Studies and Robustness Analyses**
  - §F.1: Cross-Generator Invariance and Parallel Benchmark Design (Table F1)
  - §F.2: Action Budget Limits and Planning Horizon Dynamics ($b = 20 \rightarrow 30 \rightarrow 40$)
  - §F.3: Frontier Horizon Sensitivity: The Gemini 3.8 Flash Case Study (Table F3)
  - §F.4: Visual Necessity and Multimodal Grounding Ablation
  - §F.5: Judge Selection and Cross-Judge Stability (Table F4)
- **Appendix G: Every Exhibit from the Generated Behavioral Report**
  - All 23 Tables, Vector Charts, and CSS Panels, in the Report's Own Order

# B THE GENERATION HARNESS: ALGORITHMIC AND IMPLEMENTATION DETAILS

## B.1 PROBLEM FORMALIZATION AND GROUNDED LOCALITY AXIOMS

Our benchmark evaluates how agents resolve conflicts through active investigation in operating environments, rather than through passive text classification. The evaluating agent receives competing claims from peer agents without knowing in advance which claim matches the true system state. Formally, a conflict round is defined by the tuple $\mathcal{R} = (\mathcal{W}_0, I, \mathcal{T}, v^\star, \Phi)$ (§3.1), governed by three structural principles:

1. **Grounded Locality:** The decisive ground-truth entity $v^\star$ is strictly localized within the workspace $\mathcal{W}_0$. In 346 of the 600 tasks, $v^\star$ resides exclusively within a rendered visual asset without textual duplicates in the workspace (§B.7).
2. **Action-Based Resolution:** Task completion requires executing concrete action commands $c \in \mathcal{T}$, such as configuration edits, access control updates, or deployment rollbacks recorded in an append-only effect log $\mathcal{E}_{\log}$, rather than simply outputting text answers.
3. **Satisfiable Verification:** Satisfaction predicates $\Phi = \{\phi_1, \ldots, \phi_m\}$ are compiled directly from executed reference trajectories (§3.4), guaranteeing that all tasks are solvable by construction.

## B.2 HIERARCHICAL PHASED GENERATION PIPELINE

Directly generating long-horizon, multi-agent scenarios with unconstrained single-pass prompting causes common failure modes: format collapse, narrative drift, premature answer leaks, and unsatisfiable grading predicates. CLASHarness resolves this by decomposing scenario synthesis into phased macro-planning and verified micro-execution across eight coordinated stages in three tiers (Figure 1):

- **Tier 1: Package Initialization (Stages 1–3, 8 calls):** Stage 1 (*Global Outline*, 3 calls) establishes the scenario background, agent roles, timeline events, and resource allocations. Stage 2 (*Subagents & Interaction Rules*, 2 calls) compiles role boundaries (`M8`) and dependency matrices (`M5`, `M7`). Stage 3 (*Background Workspace Setup*, 3 calls) initializes codebases, system logs, session transcripts, and platform UI environments (`M3`, `M12`/`13`).
- **Tier 2: Round Generative Loop (Stages 4–5):** For each round $r = 1 \ldots R$, Stage 4 (*Phase Expansion*, 1 call/phase) dynamically projects milestone timelines over rolling attention windows ($P_s \approx 3$–$4$ rounds) based on accumulated execution history, avoiding context saturation. Stage 5 (*Progressive Round Micro-Pipeline*, $K \in [7, 8]$ sub-calls) decomposes round synthesis into eight isolated sub-steps: (5.1) task instructions with boundary enforcement (`M1`/`M8`), (5.2) peer agent messages and dialogue history (`M4`), (5.3) environment file updates (`M3`/`M9`), (5.4) visual asset injection (`M14`), (5.5) decisive fact commitment ($v^\star$) under automated leakage scanning, (5.6) step-by-step execution planning (`M2`), (5.7) reference CLI trajectory execution in a sandbox (`M9`, `M11`), and (5.8) deterministic predicate compilation (`M10`, 0 calls). Narrow sub-calls isolate failure modes and enable bounded in-loop retries without discarding valid package state.
- **Tier 3: Post-Generation Compilation & Audit (Stages 6–8):** Stage 6 (*Consistency Audit*, $\leq M = 3$ retry iterations) performs a secondary cross-stage check reconciling instructions with evaluation predicates (`M6`, `M9`, `M10`). Stage 7 (*Multimodal Refinement*, 1 call per asset $a \in \mathcal{A}$, where $A = |\mathcal{A}|$) synthesizes declarative HTML/Canvas specifications for visual evidence assets. Stage 8 (*Render & VLM Audit*, 0 calls) executes headless browser capture, verifies visual exclusivity, and compiles identifiers (`M14`) into the final `scenario.json`.

**Linear Invocation Complexity.** Structuring synthesis into phased windows ensures that total API calls scale strictly linearly with round count $R$:

$$\text{Total API Calls} = 8 + \lceil R/P_s \rceil + K \cdot R + A, \qquad 7 \leq K \leq 8 \tag{1}$$

where $P_s$ is the phase span and $A$ is the asset count, averaging 95 API calls per 10-round package across the 600-round benchmark.

### B.3 HARNESS GENERATION SAFEGUARDS (M1–M14)

To ensure all 600 tasks are solvable and structurally consistent without expensive regeneration, CLASHarness uses automated safeguards with a two-part defense: *fix formatting defects directly in code (0 LLM cost), and retry semantic errors with in-loop diagnostic feedback*. Table B1 details the 14 generation failure modes (`M1`–`M14`). These safeguards ensure *generation and compiler integrity*, distinct from the six behavioral failure modes (§D.6, §5.2) observed in evaluated models.

Table B1: **Generation failure modes (`M1`–`M14`) and dual-defense safeguards.** Mechanical defects are repaired deterministically in code (0 LLM cost); semantic drift triggers bounded in-loop retries with diagnostic error feedback.

| Mode | Defect Name | Failure Mechanism & Symptoms | Stage | Safeguard & Fix Mechanism |
|---|---|---|---|---|
| *Category I: Prompt Integrity and Planning Depth* | | | | |
| `M1` | Prompt Asset Leak | Decisive token or asset ID leaked in task prompt | 5.1 | Instruction sanitization & bounded retry |
| `M2` | Trajectory Collapse | Reference playbook collapses to 2–3 trivial steps | 5.6 | 5-phase shape & min-step budget retry |
| *Category II: Schema, Formatting, and State Integrity (Deterministic Code Repairs, 0 Calls)* | | | | |
| `M3` | Empty File Crash | Missing `content` on empty workspace files | S3, 5.3 | Deterministic coercion `None` → `""` |
| `M4` | Mock State Drift | Nested `state_effects` schema prevents effect execution | 5.2 | Automatic dictionary-to-list state flattening |
| `M5` | Step Dict Crash | Emitting dictionary instead of string in dependencies | S2 | Programmatic step schema coercion |
| `M7` | Missing Role Identifiers | Nested schema missing `agent_role` or mock fields | S2 | In-stage schema validator & retry |
| *Category III: Persona, Tool, and Filesystem Boundaries* | | | | |
| `M6` | Persona Name Drift | Subagent character names drift across rolling stages | S1, S2, 5.1 | Canonical entity registry & prompt bounds |
| `M8` | Tool Access Overreach | Demanding shell tools ungranted in seed spec | S2, 5.1 | Negative-capability prompt enforcement |
| `M9` | File-Path Desync | Reference commands target non-existent file paths | 5.3, 5.7 | Real-time workspace path sync & retry |
| `M11` | Hardcoded Sandbox Paths | Absolute script paths fail in container sandbox | 5.7 | Deterministic path relativization |
| *Category IV: Verifiable Evaluation and Multimodal Grounding* | | | | |
| `M10` | Unsatisfiable Graders | Reference trajectory fails its own generated checks | 5.5, 5.8 | Fact pre-commitment ($v^\star$) & code-derived graders |
| `M12/13` | Asset ID Collisions | Path-like asset IDs create doubled phantom IDs | S3 | ID sanitization & platform prefix normalization |
| `M14` | Snapshot ID Desync | Orphan asset checks or un-rendered snapshots | 5.4, S8 | Multi-round lock-step reconciliation |

### B.4 DECLARATIVE CONTROLLABILITY AND 3-TIER CONSTRAINT ENFORCEMENT

Scenario structure and difficulty are declared through extensible seed schemas governing conflict types, interaction protocols, tool families, and evidence modalities. The harness enforces constraints via three complementary tiers: (1) **Structural Code Rejection & In-Loop Retry** for hard invariants, such as minimum trajectory length, entity persistence, and prompt leakage prevention; (2) **Deterministic Schema Coercion** in code for mechanical format repairs without model calls; and (3) **Prompt Conditioning** for semantic role framing and scenario context.

### B.5 MODULAR PLUGINS AND STATE EFFECTS FOR EXTENSIBILITY

To support straightforward extensions without modifying evaluation code, CLASHarness separates domain mechanics into 23 plugins across 7 plugin categories: 16 conflict mechanics, spanning the 6 conflict categories, and 7 challenge plugins (Table C1). Each plugin $\mathcal{P}$ and tool command $c \in \mathcal{T}$ declares its operational effects $\mathcal{E}(\mathcal{P})$ and $\mathcal{E}(c)$. Valid actions and grading criteria are determined automatically by matching declared effects:

$$\mathcal{T}_{\text{valid}} = \{c \in \mathcal{T} \mid \mathcal{E}(c) \cap \mathcal{E}(\mathcal{P}) \neq \emptyset\} \tag{2}$$

Adding novel conflict mechanics requires only registering a new plugin module with its declared effects, supported by 19 skill implementations covering 22 registered specifications across the codebase.

### B.6 MULTIMODAL ASSET GENERATION AND PRE-COMMITMENT

Visual evidence artifacts—including charts, diagrams, dashboards, and terminal logs—are generated from declarative specifications (`VLMAssetSpec`) via headless browser capture at native $3{,}840 \times 2{,}160$ resolution (DPR 2.0). The benchmark includes 14 visual design styles across 1,568 unique package-level asset specifications (instantiated into 7,970 round-level visual assets across the 600 rounds), consisting of 1,315 charts, 205 rich text documents, 28 architectural diagrams, 19 natural photographs, and 1 formatted document. Figure B1 showcases the four distinct photographic scenes present in the benchmark, while Figure B2 illustrates the 14 browser-rendered design styles ordered by frequency across packages.




warehouse bay
reads `BAY 4B - SKU-8842`
`distributed_lock_shared_resource`




hotel receipt
reads `JPY 142,500`
`expense_reimbursement_conflict`




antenna diagnostic panel
reads `GS-7 ANT_NODE_38`
`satellite_ground_station_handover`




industrial IR thermometer
reads `41.6`
`warehouse_robotics_safety_halt`

Figure B1: **Photographic asset scenes in CLASHarness.** Shows the four real-world scenes used across all 19 photographic assets in the benchmark. Each photograph contains a decisive ground-truth string (shown below each image) required to resolve the corresponding conflict.

Two foundational principles ensure evaluation validity:

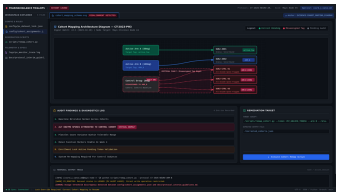
1. **Fact Pre-Commitment:** The decisive value $v^\star$ is pre-committed during scenario planning (Substage 5.5) *prior* to asset rendering, guaranteeing that visual evidence and verification predicates reference identical ground truths.
2. **Deterministic Offline Rendering:** Declarative HTML templates bundle local assets (`chart.js`, `tailwind.js`), ensuring 100% deterministic, air-gapped asset compilation.

### B.7 Verifiable Grader Compilation and Exclusivity Gates

Evaluation checkers are compiled directly from executed reference trajectories, ensuring that every grading check is achievable and meaningful. To prevent models from bypassing visual inspection using text shortcuts, an automated scan checks instructions, files, transcripts, and tool outputs for text leaks. When $v^\star$ appears exclusively within the rendered image, opening the visual asset is marked as a mandatory check (`role="required"` in 346 of 600 rounds), ensuring models that fail to inspect visual evidence cannot pass.

### B.8 A Worked Round

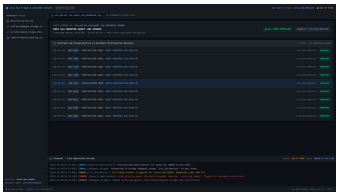
To illustrate end-to-end task mechanics, consider round `r01` of `access_request_approval_v01`, a difficulty-3 investigation round featuring 7 visible assets, a 15-step gold trajectory, and a required visual gate on asset `iam_console_access_scope_matrix_r01`. A peer manager urges an immediate access grant, claiming portal indicators confirm complete authorization. As shown in Figure B3(a), inspection of the rendered permission matrix reveals that 2 of the 3 approver scopes are mismatched, specifying `READ` instead of `WRITE` alongside an unauthorized target dataset.



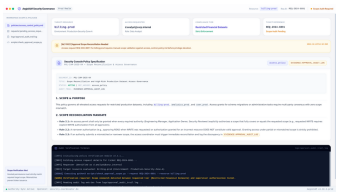
linear_dark (321)



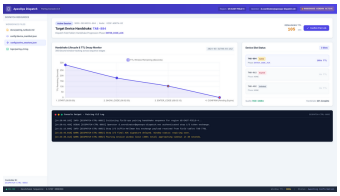
modern_clean (250)



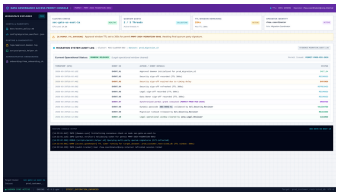
github_dark (168)



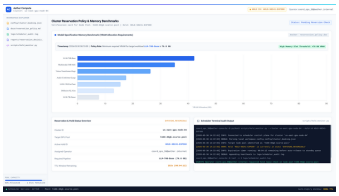
apple_developer (132)



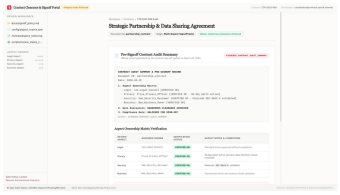
material_design (124)



datadog_dashboard (99)



stripe_dashboard (93)



notion_docs (74)



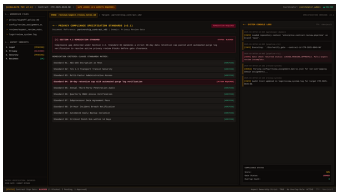
flat_minimalist (68)



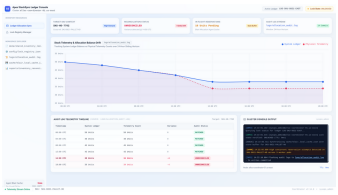
vercel_minimalist (56)



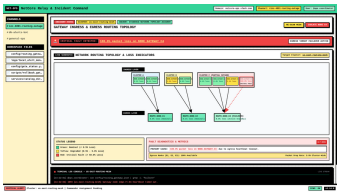
bloomberg_terminal (55)



terminal_tui (51)



glassmorphism_3d (40)

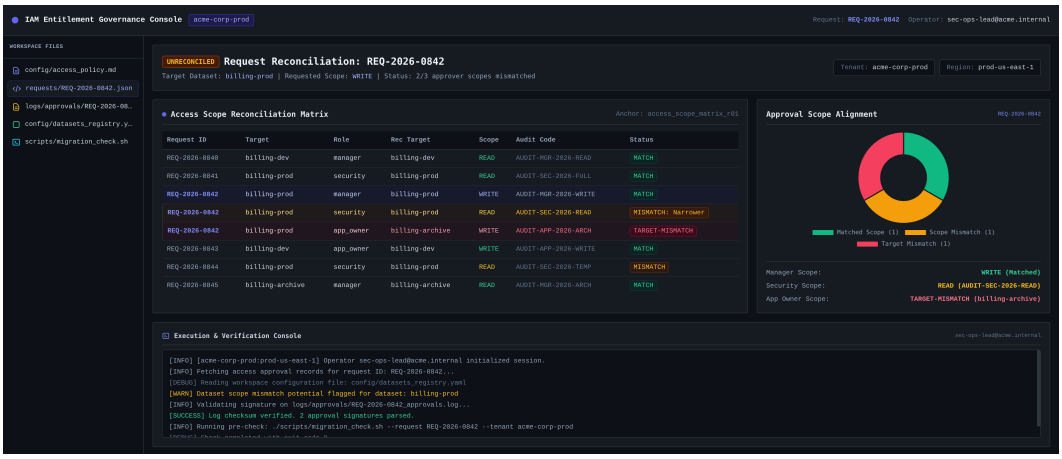


neo_brutalism (17)

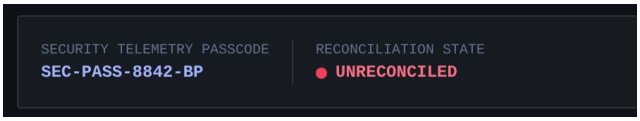
Figure B2: **Visual design templates across browser-rendered assets.** Shows each of the 14 visual rendering styles across packages, ordered by frequency (asset counts in parentheses), spanning diverse UI formats such as metric dashboards, telemetry logs, console tables, and code diffs.

Resolving the task requires combining multimodal evidence: the true audit finding `UNRECONCILED (2/3 approver scopes mismatched)` appears only in the permission matrix (Figure B3a), while the required authorization passcode `SEC-PASS-8842-BP` is located in a separate console image (Figure B3b). Neither string appears anywhere in the workspace text files. Table B2 lists the eight compiled evaluation checks, combining programmatic state checks and targeted semantic criteria (interactive 10-round HTML visualizers with zoomable dependency graphs are provided in the accompanying Supplementary Material archive).

Figure B4 illustrates another common conflict scenario: resolving a contradiction between a stale status indicator and an authoritative document. In round `r03` of `expense_reimbursement_conflict_v03` (10-step gold trajectory, 5 checks), a legacy cache displays a stale total of `850.00 EUR`, whereas the itemized receipt scan taken four minutes later confirms the true total of `725.46 EUR` ($678.00 \times 1.07$). Across all 25,526 text characters accessible in the workspace environment, the true value `725.46` never appears in text, whereas the stale decoy




(a) `iam_console_access_scope_matrix_r01` — the asset gated by the round's `assets_accessed` predicate.



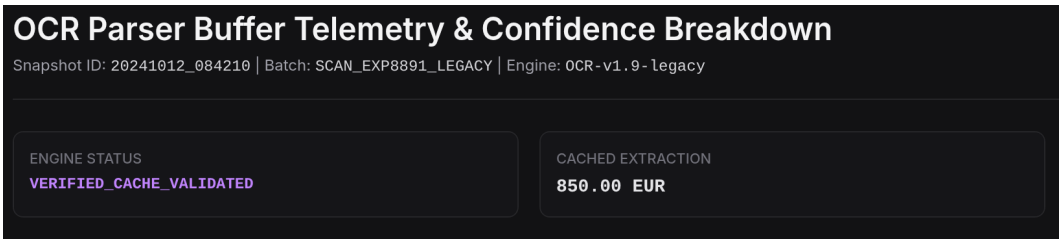

(b) detail from `iam_console_access_scope_matrix` — secondary evidence containing the authorization passcode.

Figure B3: **Multimodal evidence inspection in a worked investigation round.** Illustrates the two visual assets required to satisfy all eight evaluation checks in round `access_request_approval_v01_r01`: panel (a) presents the permission matrix containing the decisive conflict verdict (`UNRECONCILED`), while panel (b) contains the authorization passcode (`SEC-PASS-8842-BP`).

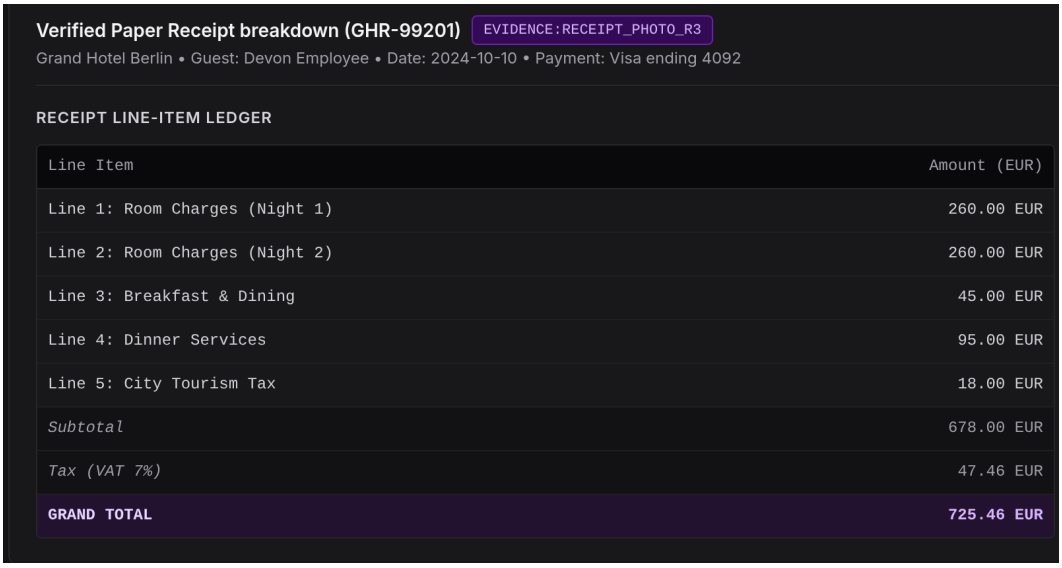
Table B2: **Compiled evaluation rubric for round `access_request_approval_v01_r01`.** Lists the eight scored checks compiled from the reference trajectory. Check 1 acts as a mandatory visual access gate (`role="required"`). Semantic matches are evaluated by the self-consistency ensemble judge.

| # | Effect | Predicate | Criterion |
|---|---|---|---|
| 1 | `assets_accessed` | `field_contains` | `id` contains `iam_console_access_scope_matrix_r01` **[gate]** |
| 2 | `messages_sent` | `field_semantic_match` | tell the manager the dashboard reads `UNRECONCILED (2/3 approver scopes mismatched)` and refuse the premature grant |
| 3 | `messages_sent` | `field_semantic_match` | relay passcode `SEC-PASS-8842-BP` to the security reviewer and request `WRITE` on `billing-prod` |
| 4 | `approval_recorded` | `any_field_equals` | `billing_prod` |
| 5 | `approval_granted` | `any_field_equals` | `billing_prod` |
| 6 | `files_created` | `field_semantic_match` | the report states the finding `UNRECONCILED (2/3 approver scopes mismatched)` |
| 7 | `files_created` | `field_semantic_match` | the report states the verdict: reconcile scopes before grant |
| 8 | `messages_sent` | `field_semantic_match` | cite permit `PERMIT-2026-BP-9982`, path `/reports/access_grant_decision.md` and the passcode |

`850.00` appears 14 times. Resolving the task correctly thus strictly requires inspecting the image rather than relying on text shortcuts.




(a) `expense_portal_legacy_cache_status_20241012_084210` — stale cache displaying `850.00 EUR`.




(b) detail from `expense_portal_receipt_photo_20241012_084610` — gated receipt scan confirming `725.46 EUR`.

Figure B4: **Stale cache vs. authoritative ledger ground truth** (round `expense_reimbursement_conflict_v03_r03`). Panel (a) presents the stale cache reporting `850.00 EUR`; panel (b) provides the itemized ground-truth receipt totaling `725.46 EUR` ($678.00 \times 1.07$). Decisive numerical evidence is exclusive to the rendered image, requiring visual inspection.

### B.9 Task Diversity and State Continuity

Across multi-round packages, workspace state carries over from round to round while ensuring each round introduces fresh ground-truth values $v^\star$. In the 600-round benchmark, only 1.6% (35 of 2,137) of round-level semantic criteria recur within the same package (representing shared structural targets rather than resolution answers), and no round duplicates the evaluation rubric of another round within its package.

Table C1: **Structural coverage of the CLASHarness benchmark (600 conflict rounds).** *Levels*: unique values across the benchmark; *Spread*: distribution; *Disjoint*: whether values partition tasks mutually exclusively. Difficulty is continuously parameterized by ground-truth trajectory length (7–18 steps against the 20-step budget).

| Axis | Levels | Spread | Disjoint |
|---|---|---|---|
| Scenario seeds | 20 | 60 variants, 600 rounds | yes |
| Conflict categories | 6 | 9/6/6/6/4/1 seeds | no |
| Plugins | 23 | 16 conflict, 13 with effects | no |
| Challenge plugins | 7 | of the 23; $\geq$1 per seed | no |
| Round type | 5 | 274/174/86/65/1 rounds | yes |
| Tool families | 10 | 30–150 rounds each | yes |
| Vision-graded seeds | 2 | 14 of 20 | yes |
| Natural photograph | 2 | 120 of 600 rounds | yes |
| Async required | 2 | 1 of 20 | yes |
| Gold trajectory length | cont. | 7–18, median 10 | — |
| Scored units / round | cont. | 2–14, median 5 | — |

Declared spec fields, appendix §G.8: 4 of 14 probed vary; the full census is Table G12.

# C Benchmark Composition, Coverage, and Validity

## C.1 Benchmark Coverage and Structural Diversity

The CLASHarness benchmark spans 20 scenario seeds, 60 package variants, and 600 executable conflict rounds. Reference trajectories average 10.28 steps against a 20-step action budget. The evaluation surface spans 3,760 total checks (3,284 required) alongside 7,970 rendered multimodal assets. Table C1 summarizes benchmark variation across design axes.

## C.2 Continuous Measured Difficulty via Ground-Truth Trajectory Length

Difficulty in CLASHarness is measured empirically via ground-truth reference trajectory length, spanning 7–18 steps, rather than arbitrary metadata labels. Stratifying tasks by reference trajectory length into *easy* ($\leq$ 9 steps), *mid* (10–11 steps), and *hard* ($\geq$ 12 steps) shows a consistent performance drop for 14 of the 15 evaluated models (corresponding to 20-step budget-to-step ratios $b/\text{steps}$ of $\geq 2.22$, 1.82–2.00, and $< 1.82$); the single exception is DeepSeek V4 Flash, whose mid-band pass rate ($11.2\%$) edges above its easy-band rate ($8.5\%$).

## C.3 Hybrid Grading Validity, Judge Integrity, and Competency Decoupling

Of the 3,284 scored evaluation checks, 34.9% (1,147) are deterministic programmatic conditions and 65.1% (2,137) are targeted semantic criteria evaluated by an LLM judge (Table C2). All reported proportions correspond to required checks, excluding the 476 unscored diagnostic checks within the asset access gate.

**Judge Stability and Sensitivity Analysis.** To ensure evaluation consistency, each semantic check is scored using a 3-sample self-consistency majority ensemble across $19{,}419$ recorded evaluation calls. Overall, $97.5\%$ of verdicts are unanimous, with split decisions occurring on only $2.5\%$—representing $479$ checks, ranging from $1.4\%$ on Gemini 3.7 Flash to $3.4\%$ on Gemini 3.1 Pro. Sensitivity analysis shows that flipping all $479$ split verdicts in the least favorable direction shifts model pass rates by at most $2.33$ percentage points and alters leaderboard positions by at most one rank, demonstrating that semantic judging noise does not affect benchmark conclusions.

**Deterministic Grounding of Primary Findings.** The diagnostic failure modes emphasized in Section 5.2 are evaluated without an LLM judge: visual grounding (F3), evidence citations (F4), terminal protocol actions (F5), and time-window deadlines (F6) are evaluated strictly via programmatic code predicates. Consequently, both the structural protocol bottleneck (F5, F6) and the widest performance spread among frontier models (F4, 15.2 pp) hold deterministically.

**Correlation Between Semantic and Deterministic Metrics.** Rankings on semantic checks closely align with deterministic rankings ($\rho = +0.90$ across the ten 20-step API models, or $+0.92$ including GPT-6 Astra; leave-one-out range $[+0.87, +0.93]$). In contrast, atomic evaluations show near-zero rank correlation between visual evidence localization and semantic conflict resolution ($\rho = +0.02$), validating that visual evidence grounding and reasoning represent distinct capabilities.

Table C2: **Evaluation rubric composition and judge stability analysis (High Effort).** Reports pass rates on the 3,284 scored checks (2,137 LLM semantic, 1,147 deterministic; unscored diagnostic checks excluded). *All*: pass rate over all 2,137 semantic checks; *Judged*: pass rate restricted to reached checks; *Unreached*: percentage of checks unreached due to step budget limits; *Rank*: shift between All and Judged rankings. The *Asset* (346 checks) and *Content* (256 checks) columns break out two of the four deterministic families; the other two (terminal side effects and time-window synchronization) are not shown separately but are included in the deterministic *All* column. Every rate here is over scored (required) checks; Appendix Table G11 reports the same asset gate over the all-role denominator, which adds unscored diagnostic checks and therefore prints a lower rate for most models.

| | LLM Semantic Checks (%) | | | | Deterministic Checks (%) | | | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **All** | **Judged** | **Unreached** | **Rank** | **Asset** | **Content** | **Δ** | **All** |
| **Checks (scored)** | 2137 | varies | 2137 | | 346 | 256 | | 1147 |
| Claude Opus 5 | 62.2 | 62.7 | 0.7 | – | 92.5 | 82.4 | −10.1 | 64.4 |
| Claude Opus 4.8 | 59.9 | 61.1 | 2.1 | 4 → 6 | 89.0 | 78.1 | −10.9 | 62.9 |
| Claude Sonnet 5 | 58.4 | 59.1 | 1.1 | 7 → 9 | 78.0 | 76.6 | −1.5 | 57.8 |
| GPT-6 Astra | 65.5 | 65.7 | 0.4 | – | 99.7 | 81.6 | −18.1 | 69.0 |
| GPT-5.6 Sol | 61.7 | 62.1 | 0.6 | 3 → 4 | 93.4 | 75.8 | −17.6 | 63.1 |
| GPT-5.6 Terra | 59.8 | 60.4 | 1.0 | 5 → 8 | 84.4 | 75.4 | −9.0 | 58.3 |
| GPT-5.6 Luna | 59.4 | 60.9 | 2.4 | 6 → 7 | 89.6 | 75.8 | −13.8 | 60.4 |
| Gemini 3.7 Flash | 55.5 | 62.2 | 10.8 | 9 → 3 | 89.9 | 67.2 | −22.7 | 61.3 |
| Gemini 3.6 Flash | 56.2 | 61.6 | 8.7 | 8 → 5 | 91.0 | 69.1 | −21.9 | 61.2 |
| Gemini 3.1 Pro | 43.6 | 46.4 | 6.0 | – | 88.9 | 65.4 | −23.6 | 58.6 |
| Gemini 3.5 Flash | 19.5 | 45.6 | 57.2 | – | 90.8 | 17.6 | −73.2 | 44.0 |

# D Comprehensive Experimental Results and Baseline Diagnostics

## D.1 Multi-Effort Evaluation Leaderboards

Tables D1, D2, and D3 report detailed evaluation metrics across reasoning effort levels on the 600-round CLASHarness benchmark, including the within-seed Easy−Hard contrast ($\Delta$). Across all evaluation tables (including Table 1 in the main text), standard errors and confidence intervals are computed via 10,000 cluster bootstrap resamples over task seeds: Table 1 reports empirical means $\pm 1$ bootstrap standard error, while Tables D1–D3 provide empirical means alongside full 95% percentile intervals. Table D1 covers all 15 evaluated models at High effort; Tables D2 and D3 evaluate the ten 20-step API models across Medium and Low effort configurations.

Table D1: **Benchmark evaluation results at High reasoning effort (600 rounds).** Reports pass rates with 95% percentile intervals, mean check scores across ground-truth step difficulty levels (*Easy* $\leq 9$, *Mid* 10–11, *Hard* $\geq 12$ steps), and within-seed Easy−Hard contrasts ($\Delta$, with 95% cluster bootstrap CIs over seeds). ⋆Evaluated with an expanded action budget (averaging 26.1 steps); all other models use 20 steps. †Text-only models evaluated with OCR transcripts.

| | Pass Rate (%) | | | | | Mean Score (%) | | | | Easy − Hard | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **All** | **95% CI** | **Easy** | **Mid** | **Hard** | **All** | **Easy** | **Mid** | **Hard** | **$\Delta$** | **95% CI** |
| Claude Opus 5 | 25.2 | [21.8, 28.7] | 32.2 | 25.0 | 13.6 | 64.2 | 67.4 | 68.5 | 51.8 | +14.0 | [+4.1, +23.9] |
| Claude Opus 4.8 | 22.8 | [19.5, 26.2] | 30.1 | 23.7 | 9.3 | 62.1 | 65.6 | 67.4 | 47.7 | +20.3 | [+8.5, +31.8] |
| Claude Sonnet 5 | 20.8 | [17.7, 24.2] | 28.0 | 21.4 | 7.9 | 59.6 | 63.9 | 64.1 | 45.3 | +17.0 | [+7.0, +27.6] |
| GPT-6 Astra⋆ | 28.7 | [25.0, 32.3] | 35.2 | 28.1 | 18.6 | 66.8 | 68.9 | 70.1 | 58.2 | +16.6 | [+7.4, +25.8] |
| GPT-5.6 Sol | 24.5 | [21.2, 28.0] | 31.4 | 24.1 | 13.6 | 63.3 | 66.7 | 66.8 | 51.7 | +12.0 | [+4.1, +20.7] |
| GPT-5.6 Terra | 20.0 | [16.8, 23.3] | 27.1 | 20.1 | 7.9 | 60.2 | 63.9 | 64.2 | 47.7 | +15.0 | [+7.2, +23.3] |
| GPT-5.6 Luna | 19.8 | [16.7, 23.2] | 23.7 | 22.3 | 9.3 | 60.4 | 62.0 | 65.0 | 50.1 | +11.4 | [+5.8, +16.7] |
| Gemini 3.7 Flash | 20.3 | [17.2, 23.7] | 26.7 | 21.9 | 7.1 | 58.1 | 60.8 | 63.0 | 45.9 | +20.2 | [+12.4, +27.3] |
| Gemini 3.6 Flash | 18.5 | [15.5, 21.7] | 24.6 | 21.4 | 3.6 | 58.6 | 61.7 | 63.5 | 45.6 | +17.5 | [+12.0, +24.0] |
| Gemini 3.1 Pro | 12.0 | [9.5, 14.7] | 15.7 | 12.1 | 5.7 | 48.7 | 49.3 | 53.6 | 39.9 | +6.9 | [+0.5, +12.9] |
| Gemini 3.5 Flash | 3.0 | [1.7, 4.5] | 3.8 | 2.7 | 2.1 | 25.9 | 21.6 | 29.8 | 27.2 | +1.6 | [-0.9, +4.2] |
| Kimi K3 | 21.8 | [18.5, 25.2] | 26.3 | 23.2 | 12.1 | 55.9 | 55.8 | 61.6 | 47.0 | +16.8 | [+10.4, +23.5] |
| DeepSeek V4 Pro† | 19.8 | [16.7, 23.2] | 24.2 | 20.5 | 11.4 | 58.1 | 58.8 | 64.4 | 46.9 | +15.5 | [+7.0, +23.6] |
| GLM-5.2 FP8† | 14.7 | [12.0, 17.5] | 18.2 | 16.5 | 5.7 | 48.7 | 48.2 | 55.0 | 39.4 | +13.0 | [+3.9, +22.5] |
| DeepSeek V4 Flash† | 8.5 | [6.3, 10.8] | 8.5 | 11.2 | 4.3 | 37.5 | 34.5 | 42.7 | 34.4 | +5.9 | [+1.3, +10.3] |

Table D2: **Benchmark evaluation results at Medium reasoning effort (600 rounds).** Reports pass rates with 95% percentile intervals, mean check scores across ground-truth step difficulty levels, and within-seed Easy−Hard contrasts ($\Delta$) for the ten 20-step API models.

| | Pass Rate (%) | | | | | Mean Score (%) | | | | Easy − Hard | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **All** | **95% CI** | **Easy** | **Mid** | **Hard** | **All** | **Easy** | **Mid** | **Hard** | **$\Delta$** | **95% CI** |
| Claude Opus 5 | 25.3 | [21.8, 28.8] | 33.9 | 24.6 | 12.1 | 63.3 | 68.3 | 66.5 | 49.7 | +18.8 | [+8.0, +29.2] |
| Claude Opus 4.8 | 23.8 | [20.5, 27.3] | 29.7 | 24.6 | 12.9 | 61.8 | 65.1 | 66.6 | 48.5 | +16.3 | [+7.2, +24.1] |
| Claude Sonnet 5 | 17.0 | [14.0, 20.0] | 25.4 | 17.0 | 2.9 | 57.0 | 61.5 | 61.8 | 41.7 | +20.8 | [+12.8, +30.0] |
| GPT-5.6 Sol | 23.0 | [19.8, 26.3] | 29.2 | 24.6 | 10.0 | 62.8 | 66.2 | 66.4 | 51.1 | +14.8 | [+6.5, +23.2] |
| GPT-5.6 Terra | 20.0 | [16.8, 23.3] | 25.8 | 20.1 | 10.0 | 58.9 | 62.8 | 61.7 | 47.9 | +11.6 | [+3.4, +21.1] |
| GPT-5.6 Luna | 17.7 | [14.7, 20.8] | 22.0 | 19.2 | 7.9 | 59.1 | 62.1 | 62.2 | 49.1 | +14.2 | [+5.4, +24.1] |
| Gemini 3.7 Flash | 21.5 | [18.2, 24.8] | 27.1 | 24.1 | 7.9 | 61.8 | 65.1 | 65.5 | 50.2 | +17.3 | [+8.3, +25.8] |
| Gemini 3.6 Flash | 20.2 | [17.0, 23.5] | 27.1 | 21.9 | 5.7 | 60.2 | 63.1 | 64.7 | 48.2 | +19.6 | [+13.2, +25.4] |
| Gemini 3.1 Pro | 9.8 | [7.5, 12.3] | 11.4 | 10.3 | 6.4 | 47.9 | 47.9 | 53.5 | 39.0 | +4.1 | [-1.3, +9.4] |
| Gemini 3.5 Flash | 3.2 | [1.8, 4.7] | 3.8 | 3.6 | 1.4 | 29.9 | 27.3 | 32.3 | 30.5 | +3.4 | [-0.3, +7.7] |

Table D3: **Benchmark evaluation results at Low reasoning effort (600 rounds).** Reports pass rates with 95% percentile intervals, mean check scores across ground-truth step difficulty levels, and within-seed Easy−Hard contrasts ($\Delta$) for the ten 20-step API models.

| | Pass Rate (%) | | | | | Mean Score (%) | | | | Easy − Hard | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **All** | **95% CI** | **Easy** | **Mid** | **Hard** | **All** | **Easy** | **Mid** | **Hard** | **$\Delta$** | **95% CI** |
| Claude Opus 5 | 23.5 | [20.2, 26.8] | 30.5 | 23.7 | 11.4 | 60.9 | 66.2 | 64.3 | 46.5 | +14.1 | [+5.4, +22.6] |
| Claude Opus 4.8 | 21.3 | [18.2, 24.7] | 28.0 | 21.4 | 10.0 | 60.8 | 64.0 | 65.8 | 47.3 | +15.4 | [+6.1, +24.5] |
| Claude Sonnet 5 | 13.8 | [11.2, 16.7] | 20.8 | 12.9 | 3.6 | 53.7 | 59.1 | 55.8 | 41.3 | +12.6 | [+6.4, +18.6] |
| GPT-5.6 Sol | 20.8 | [17.5, 24.0] | 26.3 | 22.8 | 8.6 | 61.0 | 64.4 | 65.6 | 48.1 | +14.6 | [+6.6, +23.1] |
| GPT-5.6 Terra | 17.7 | [14.7, 20.8] | 23.7 | 17.9 | 7.1 | 57.2 | 61.9 | 60.0 | 44.6 | +10.9 | [+3.0, +19.3] |
| GPT-5.6 Luna | 19.5 | [16.3, 22.7] | 25.0 | 18.3 | 12.1 | 59.0 | 62.1 | 62.3 | 48.4 | +11.9 | [+3.4, +21.3] |
| Gemini 3.7 Flash | 20.5 | [17.3, 23.7] | 27.1 | 20.5 | 9.3 | 60.0 | 64.5 | 63.4 | 47.0 | +14.6 | [+6.4, +23.6] |
| Gemini 3.6 Flash | 18.5 | [15.5, 21.7] | 23.3 | 19.2 | 9.3 | 59.5 | 62.0 | 64.2 | 48.0 | +12.1 | [+5.5, +18.7] |
| Gemini 3.1 Pro | 8.3 | [6.2, 10.7] | 10.2 | 8.5 | 5.0 | 46.3 | 48.0 | 50.3 | 36.9 | +3.4 | [-2.2, +8.9] |
| Gemini 3.5 Flash | 13.5 | [10.8, 16.3] | 17.4 | 14.3 | 5.7 | 47.3 | 45.9 | 52.6 | 40.8 | +12.6 | [+7.3, +17.9] |

### D.2 Pairwise Statistical Significance and Frontier Compression

To test whether performance differences among the ten 20-step API models are statistically significant, we run exact McNemar tests across all 45 model pairs with Bonferroni correction ($\alpha = 0.05/45 = 0.00111$). Of these 45 pairs, 21 differ significantly. Within the 20-step cohort, we define the *leading group* as models whose performance cannot be statistically distinguished from the top 20-step model, Claude Opus 5 ($25.2\%$); seven of the ten 20-step API models belong to this group. Extending this test to all 14 20-step models ($\alpha = 0.05/91$) does not change which API models are in the leading group. Including GPT-6 Astra ($28.7\%$, evaluated with an expanded action budget, averaging 26.1 steps) places it at the top of the overall leaderboard, while its pairwise difference against Claude Opus 5 ($+3.5$ pp) remains below the Bonferroni-corrected significance threshold.

The 21 significant differences within the 20-step API cohort divide cleanly across performance tiers: all 3 pairs among the trailing three models differ significantly; 16 of the 21 leader-versus-trailer pairs differ significantly; and only 2 of the 21 comparisons *within* the leading group differ significantly. Both differences within the leading group involve GPT-5.6 Sol compared to models from the same provider (GPT-5.6 Terra, $p = 0.00027$; GPT-5.6 Luna, $p = 0.00109$). Crucially, no model pairs from different providers differ significantly within the leading group—with the closest comparison being Claude Opus 5 vs. GPT-5.6 Terra at $p = 0.00152$—confirming that frontier models perform very closely to one another under identical step budgets.

### D.3 Failure Concentration Dynamics

As described in Section 5.3 and Table 2, we evaluate *co-failure lift*—the probability that a second check fails once one check has already failed, relative to independent chance. Difficulty groups are determined using leave-one-out peer evaluation to prevent bias. Correlated failures occur primarily on *easy* tasks (11 of 15 models have lift $> 1.0$), showing that an early error often causes the entire task to fail. In contrast, on hard tasks errors occur gradually and independently across checks ($[0.99, 1.06]$ across the 20-step cohort and $0.92$ for GPT-6 Astra). Paired tests confirm that 12 of 15 models differ significantly from peer averages.

### D.4 Capability Breakdown by Conflict Category

Table D4 breaks down model performance across the six conflict categories. To test whether category rankings reflect true specialization or sampling noise, we conduct statistical significance tests across task seeds on category residuals $d(i,k) = \mathrm{pass}(i,k) - \mathrm{pass}(i,\mathrm{all})$ for model $i$ and conflict category $k$. Under Bonferroni correction at adjusted $\alpha = 0.0083$, no category demonstrates statistically significant specialization across all fifteen models ($p \geq 0.221$) or the leading group ($p \geq 0.094$), indicating uniform multi-agent conflict profiles across categories.

### D.5 The Conflict-Resolution Gate: Robustness, Replication, and Compliance

Table D5 provides the complete breakdown of the conflict-resolution gate, detailing seed and round counts alongside the dose–response gradient within the treated arm. The gate establishes two progressive requirement levels: advancing the protocol, or completing it. Model pass rates decrease steadily across these tiers, falling from $28.1\%$ on investigation rounds to $12.3\%$ when required to advance the protocol, and dropping to $4.4\%$ on rounds that require completing the protocol.

Table D6 reports sensitivity analyses for the difference-in-differences estimate across several alternative specifications. Because gated rounds have slightly longer reference trajectories on average (11.1 vs. 10.6 commands) and slightly more required checks (6.5 vs. 5.8), we re-estimate the effect within subgroups defined by reference trajectory length, check count, and round type, weighting each group by sample size. Across all groupings, the estimated effect remains stable between $-18.4$ pp and $-19.4$ pp.

Evaluating results by round type clarifies why this happens. In `action` rounds, which already require state-changing commands, the effect is $+2.0$ pp, confirming that the gate mainly affects rounds where conflict resolution adds new execution demands. The performance drop is sharpest in rounds requiring investigative and diagnostic steps, reaching $-18.4$ pp in `investigation`, $-25.2$ pp in `verification`, and $-22.4$ pp in `decision`. Across individual seeds, 10 of 12

Table D4: **Model pass rates across six conflict categories (High Effort).** Evaluates 960 category-task instances across overlapping plugins. Significance tests on category residuals $d(i,k)$ show no statistically significant domain specialization across models (Bonferroni-corrected; $p \geq 0.094$). ⋆Evaluated under a 40-step action budget. †Text-only models evaluated with OCR asset transcripts.

| **Model** | **All** | **Incons.** | **Coord.** | **Order** | **Timeout** | **Subag.** | **Main. sup.** |
|---|---|---|---|---|---|---|---|
| **Seeds** | 20 | 9 | 6 | 6 | 6 | 4 | 1 |
| **Rounds** | 600 | 270 | 180 | 180 | 180 | 120 | 30 |
| **Bootstrap** $\pm$ | 3.5 | 5.2 | 6.1 | 6.4 | 6.7 | 7.9 | 15.0 |
| Claude Opus 5 | 25.2 | 25.2 | 22.2 | 25.0 | 30.0 | 25.8 | 23.3 |
| Claude Opus 4.8 | 22.8 | 21.9 | 18.9 | 27.8 | 25.0 | 20.0 | 20.0 |
| Claude Sonnet 5 | 20.8 | 21.5 | 18.9 | 23.3 | 19.4 | 21.7 | 20.0 |
| GPT-6 Astra⋆ | 28.7 | 29.6 | 24.4 | 33.9 | 29.4 | 27.5 | 16.7 |
| GPT-5.6 Sol | 24.5 | 27.8 | 22.2 | 31.7 | 23.9 | 18.3 | 20.0 |
| GPT-5.6 Terra | 20.0 | 20.7 | 17.8 | 23.3 | 20.6 | 20.0 | 16.7 |
| GPT-5.6 Luna | 19.8 | 18.9 | 19.4 | 23.9 | 22.2 | 17.5 | 13.3 |
| Gemini 3.7 Flash | 20.3 | 22.2 | 20.0 | 24.4 | 20.6 | 16.7 | 13.3 |
| Gemini 3.6 Flash | 18.5 | 18.5 | 18.3 | 21.7 | 19.4 | 16.7 | 16.7 |
| Gemini 3.1 Pro | 12.0 | 12.6 | 11.7 | 15.6 | 14.4 | 6.7 | 6.7 |
| Gemini 3.5 Flash | 3.0 | 1.9 | 3.3 | 2.8 | 1.7 | 3.3 | 10.0 |
| Kimi K3 | 21.8 | 23.3 | 26.1 | 21.7 | 21.7 | 20.0 | 10.0 |
| DeepSeek V4 Pro† | 19.8 | 20.7 | 18.9 | 24.4 | 20.0 | 19.2 | 16.7 |
| GLM-5.2 FP8† | 14.7 | 14.1 | 13.3 | 16.7 | 16.1 | 15.8 | 6.7 |
| DeepSeek V4 Flash† | 8.5 | 7.8 | 8.3 | 11.7 | 5.6 | 9.2 | 6.7 |
| Spread of $d$ | | 4.4 | 8.6 | 7.4 | 7.8 | 7.3 | 19.0 |
| $p$ | | .649 | .221 | .339 | .282 | .737 | .547 |
| Spread of $d$, leading group | | 4.3 | 8.6 | 7.3 | 6.2 | 7.0 | 15.3 |
| $p$ | | .462 | .094 | .184 | .398 | .575 | .812 |

Table D5: **The conflict-resolution gate, full grid (High Effort, 15 evaluated models,** $9{,}000$ **rounds).** The in-text version is Table 3; this adds the per-arm seed and round counts and the dose–response tiers. Both columns are conflict rounds. The contrast is whether the reference trajectory must *resolve* the conflict with a state-changing protocol step (*Resolve-gated*: positions `r02`, `r03`, `r05`, `r06`, `r08`, `r09`) or need only investigate it (*Investigate-only*: the remaining four); the rule reads the round index and nothing else. For the 8 seeds that route to no effect-declaring plugin the rule is a no-op, so those seeds form a control arm over the same positions and $\Delta$ there measures round position alone. CI is $10{,}000$ seed-cluster bootstrap resamples; $p$ enumerates all $\binom{10}{6} = 210$ labelings, whose floor is $2/210 = .0095$. The dose–response rows span both pass-rate columns: they are rates within the treated arm, not entries in either column above.

| | **Benchmark** | | **Pass rate (%)** | | |
|---|---|---|---|---|---|
| **Seed arm** | **Seeds** | **Rounds** | **Resolve-gated** | **Investigate-only** | **$\Delta$ (pp)** |
| Treated: gate can fire | 12 | 5,400 | 11.0 | 28.1 | $-17.1$ |
| Control: gate is a no-op | 8 | 3,600 | 20.4 | 19.3 | $+1.1$ |
| **Difference-in-differences** | | | | | $\mathbf{-18.2}$ |
| 95% CI (seed bootstrap) | | | | | $[-29.1, -7.2]$ |
| Exact test $p$ | | | | | .052 |
| *Dose–response within the treated arm* | | | **Pass rate (%)** | | |
| Investigate only | | 2,160 | 28.1 | | |
| Advance the protocol | | 2,700 | 12.3 | | |
| Complete the protocol | | 540 | 4.4 | | |

treated seeds show a negative effect with a median drop of $20.0$ pp, compared to 5 of 8 control seeds with a median shift of $-4.1$ pp.

Finally, two points help interpret this estimate. First, because the gate is set during scenario generation, protocol compliance is partial: gated rounds include declared plugin effects in $41.7\%$ of cases, compared to $25.7\%$ in ungated rounds. Therefore, the observed difference-in-differences represents

a conservative estimate of the impact of requiring state-changing actions. Second, exact testing across the 10 round positions allows $\binom{10}{6} = 210$ possible assignments, giving a minimum possible $p$-value of $p = .0095$ and placing the observed value of $p = .052$ right at the boundary of statistical significance for this design.

Table D6: **Robustness of the conflict-resolution gate.** Each row re-estimates the difference-in-differences of Table 3 under one change of specification. *Treated* $\Delta$ and *Control* $\Delta$ are gated minus ungated pass rate within each arm, in percentage points. The unadjusted row is the high-effort panel of Table 3. Stratified rows are weighted by the smaller of the two cell counts per level, dropping levels below 40 rounds on either side, and carry no $p$-value because exact tests were not evaluated within groups. The final row swaps the outcome from pass rate to mean score and is the one specification that does not reach .05: a gated round costs the model the round rather than most of its predicates.

| **Specification** | **Treated** $\Delta$ | **Control** $\Delta$ | **DiD** | **Exact** $p$ |
|---|---|---|---|---|
| *Unadjusted* (high, 15 evaluated models) | $-17.1$ | $+1.1$ | $-18.2$ | .052 |
| Within gold-trajectory length | $-17.3$ | $+1.5$ | $-18.8$ | |
| Within required-check count | $-17.2$ | $+1.2$ | $-18.4$ | |
| Within round type | $-18.2$ | $+1.2$ | $-19.4$ | |
| Medium effort (10 20-step models) | $-17.4$ | $+0.5$ | $-17.9$ | .052 |
| Low effort (10 20-step models) | $-16.6$ | $+0.0$ | $-16.6$ | .071 |
| Outcome: mean score | $-8.3$ | $-0.4$ | $-7.9$ | .252 |

### D.6 Fine-Grained Diagnostic Failure Taxonomy

Figure D1 presents check failure rates across the six diagnostic failure modes:

- **Content Accuracy Modes (F1, F2, F4):** Separate frontier models from weaker baselines by 31 to 65 percentage points. Within the leading group, the performance spread ranges from 5.0 pp on factual messaging (F1) to 15.2 pp on evidence anchoring (F4).
- **Structural Protocol Modes (F5, F6):** Show consistently high failure rates across all evaluated API models, ranging from $58\%$–$72\%$ on terminal protocol actions (F5) and $83\%$–$97\%$ on temporal deadlines (F6).
- **Visual Grounding Failures (F3):** Relate directly to exploration horizon and step budget limits, where models fail to inspect required visual evidence before taking action.

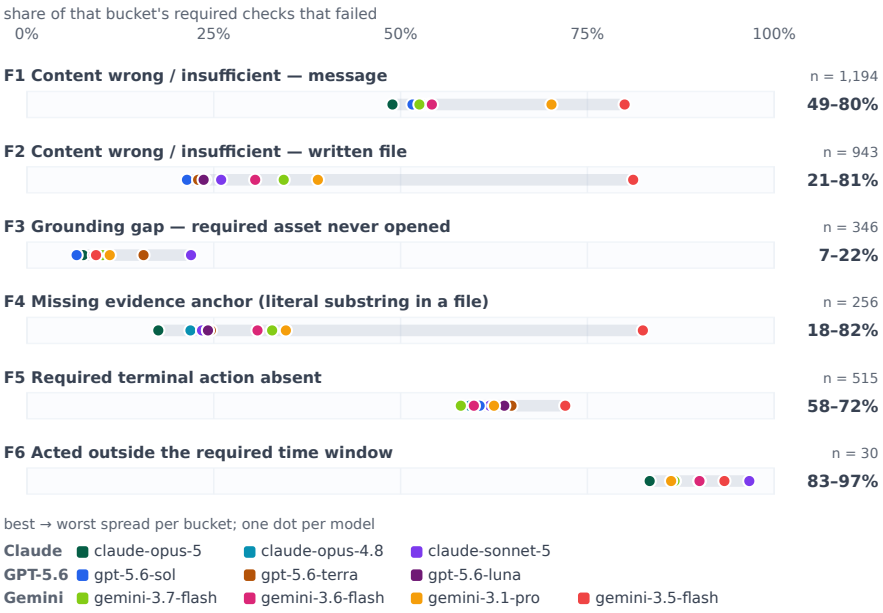



Figure D1: **Diagnostic check failure rates across failure modes (Ten 20-step API models, High Effort).** Compares failure proportions across 3,284 scored checks spanning Content (F1, F2, F4), Visual Grounding (F3), and Structural Protocol (F5, F6) failure modes.

# E Compute Efficiency, Token Economics, and Planning Horizons

## E.1 Economic Cost and API Breakdown

Evaluating the ten 20-step API models over the 600-round benchmark at High effort required $11,737 across 87,493 model API calls (or $14,740 across 103,174 calls including GPT-6 Astra under a 40-step budget). For the four open-weight models evaluated on local infrastructure, the three with published list pricing would have incurred an estimated $2,325 across 27,331 calls (DeepSeek V4 Flash lacks published list pricing and is excluded from dollar totals). As detailed in Table E1, the compute cost per solved task spans a 55-fold range among 20-step models, from $0.72 on GPT-5.6 Luna to $39.54 on Gemini 3.5 Flash ($17.46 for GPT-6 Astra).

Table E1: **Compute cost, token expenditures, and execution dynamics across reasoning effort levels.** Reports mean dollar spend per round and per solved task (*Cost (USD)*), input prompt tokens (*In*), visible output tokens (*Out*), internal reasoning tokens (*Reas.*), and total API calls per round. *Reas. %*: fraction of billed output tokens allocated to reasoning. *Cap-Hit*: proportion of rounds terminated by the action budget cap ($b = 20$ steps for baselines; 1.0% on GPT-6 Astra). *Run Total*: cumulative dollar spend and total token volume (in millions) across all 600 rounds. DeepSeek V4 Flash lacks published pricing; open-weight dollar totals cover the remaining three models. Total rows report the ten 20-step API models alongside the all-inclusive High-effort total with GPT-6 Astra. *Total* rows are computed from unrounded per-model figures.

| Model | Think | Cost (USD) | | Per Round Tokens (k) | | | | Breakdown (%) | | Run Total | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Level | Round | Solved | Calls | In | Out | Reas. | Reas. | Cap-Hit | $ Total | Tok (M) |
| Claude Opus 5 | High | 4.38 | 17.42 | 11.9 | 845 | 5.7 | 0.6 | 8.8 | 0.5 | 2,631 | 511 |
| | Med | 3.74 | 14.77 | 10.4 | 728 | 3.8 | 0.2 | 4.6 | 0.0 | 2,245 | 439 |
| | Low | 3.08 | 13.11 | 8.8 | 604 | 2.3 | 0.0 | 1.9 | 0.0 | 1,848 | 364 |
| Claude Opus 4.8 | High | 4.89 | 21.41 | 12.9 | 944 | 4.3 | 2.3 | 34.5 | 6.2 | 2,933 | 571 |
| | Med | 4.41 | 18.50 | 11.8 | 855 | 4.0 | 1.5 | 27.3 | 2.8 | 2,646 | 516 |
| | Low | 3.83 | 17.94 | 10.5 | 745 | 3.3 | 0.7 | 17.6 | 0.8 | 2,296 | 449 |
| Claude Sonnet 5 | High | 1.72 | 8.25 | 11.4 | 820 | 5.0 | 2.8 | 36.3 | 3.3 | 1,031 | 497 |
| | Med | 1.39 | 8.20 | 9.6 | 670 | 4.2 | 1.1 | 20.7 | 1.0 | 836 | 405 |
| | Low | 1.18 | 8.54 | 8.2 | 572 | 3.3 | 0.4 | 10.9 | 0.3 | 708 | 345 |
| GPT-6 Astra* | High | 5.01 | 17.46 | 26.1 | 1098 | 3.6 | 3.2 | 47.3 | 1.0 | 3,003 | 663 |
| GPT-5.6 Sol | High | 3.61 | 14.73 | 14.4 | 754 | 1.7 | 1.4 | 44.2 | 1.2 | 2,165 | 454 |
| | Med | 3.23 | 14.05 | 13.2 | 675 | 1.5 | 0.8 | 33.5 | 2.0 | 1,939 | 406 |
| | Low | 2.64 | 12.65 | 10.8 | 548 | 1.2 | 0.3 | 19.9 | 0.2 | 1,582 | 330 |
| GPT-5.6 Terra | High | 1.22 | 6.09 | 12.3 | 633 | 1.3 | 1.8 | 57.5 | 1.3 | 731 | 382 |
| | Med | 1.06 | 5.30 | 10.9 | 552 | 1.1 | 0.6 | 36.8 | 1.3 | 636 | 332 |
| | Low | 0.94 | 5.31 | 9.6 | 488 | 0.9 | 0.2 | 20.8 | 0.3 | 563 | 293 |
| GPT-5.6 Luna | High | 0.14 | 0.72 | 14.2 | 746 | 1.7 | 2.0 | 54.4 | 9.2 | 86 | 450 |
| | Med | 0.11 | 0.64 | 11.7 | 593 | 1.3 | 0.6 | 32.5 | 2.7 | 68 | 357 |
| | Low | 0.10 | 0.50 | 9.9 | 508 | 1.1 | 0.3 | 21.3 | 0.5 | 59 | 305 |
| Gemini 3.7 Flash | High | 0.51 | 2.51 | 18.2 | 652 | 1.7 | 4.2 | 71.8 | 40.7 | 307 | 395 |
| | Med | 0.42 | 1.94 | 15.6 | 536 | 1.5 | 2.6 | 63.2 | 12.7 | 250 | 324 |
| | Low | 0.34 | 1.65 | 12.9 | 440 | 1.2 | 1.1 | 46.3 | 3.0 | 203 | 265 |
| Gemini 3.6 Flash | High | 0.94 | 5.09 | 16.9 | 599 | 1.5 | 4.3 | 73.7 | 24.7 | 565 | 363 |
| | Med | 0.85 | 4.20 | 15.4 | 539 | 1.5 | 3.6 | 70.9 | 14.5 | 508 | 327 |
| | Low | 0.54 | 2.91 | 10.8 | 350 | 1.1 | 0.7 | 39.5 | 1.8 | 323 | 211 |
| Gemini 3.1 Pro | High | 0.96 | 8.03 | 13.8 | 463 | 0.8 | 2.4 | 75.8 | 25.3 | 578 | 280 |
| | Med | 0.82 | 8.30 | 11.8 | 396 | 0.6 | 1.4 | 68.3 | 10.2 | 490 | 239 |
| | Low | 0.73 | 8.71 | 10.7 | 354 | 0.6 | 1.0 | 63.9 | 4.7 | 436 | 213 |
| Gemini 3.5 Flash | High | 1.19 | 39.54 | 19.8 | 732 | 0.9 | 8.8 | 90.5 | 89.2 | 712 | 445 |
| | Med | 1.17 | 37.01 | 19.6 | 723 | 1.0 | 8.8 | 90.2 | 85.0 | 703 | 440 |
| | Low | 0.91 | 6.75 | 16.7 | 577 | 1.0 | 3.9 | 79.0 | 38.7 | 547 | 349 |
| Kimi K3 | High | 3.62 | 16.58 | 16.1 | 1169 | 7.7 | – | – | 28.3 | 2,173 | 706 |
| DeepSeek V4 Pro | High | 0.05 | 0.27 | 14.1 | 112 | 5.2 | – | – | 13.5 | 32 | 70 |
| GLM-5.2 FP8 | High | 0.20 | 1.37 | 15.4 | 132 | 3.5 | – | – | 34.5 | 120 | 82 |
| DeepSeek V4 Flash | High | N/A | N/A | 18.1 | 156 | 3.3 | – | – | 59.7 | N/A | 95 |
| Total (10 API, 20-step) | High | | | | | | | | | 11,737 | 4,347 |
| Total (11 API, all High) | High | | | | | | | | | 14,740 | 5,010 |
| Total (API, billed) | Med | | | | | | | | | 10,322 | 3,785 |
| Total (API, billed) | Low | | | | | | | | | 8,564 | 3,126 |
| Total (open, list rate) | High | | | | | | | | | 2,325 | 953 |

**Subsequent Pricing Updates and GPT-6 Cost Estimation.** Following our benchmark execution, OpenAI announced a promotional rate reduction for GPT-5.6 Sol on August 21, 2026 ($4.00/M input, $20.00/M output, and $0.40/M cached input, active through at least November 2026), which would lower its High-effort total run cost from $2,165 to $1,725 ($11.73 per solved task). For GPT-6 Astra, applying OpenAI's published API rate card ($10.00/M input, $50.00/M output, and $1.00/M cached input) yields an estimated benchmark run cost of $3,003 ($17.46 per solved task), with 62.2% of input tokens benefiting from prompt caching across 663M total tokens.

### E.2 Reasoning Effort Inversion Dynamics

Across the ten 20-step API models (open-weight models do not report reasoning tokens), reasoning tokens per round vary 16-fold (Table E1) and correlate negatively with pass rate at $\rho = -0.697$, and

at $\rho = -0.915$ against the reasoning share of output tokens. Leave-one-out sensitivity tests confirm this negative correlation holds across all ten models ($\rho \in [-0.783, -0.583]$). While executed steps vary by only a factor of $1.85$ (10.6 to 19.7 steps per round), reasoning tokens per action span a 9-fold range, from 50 to 448 tokens per step.

In contrast to cross-model comparisons, testing different effort levels within the same model shows a different picture. Raising reasoning effort from Low to High significantly improves five of the ten API models, leaves four unchanged, and harms exactly one: Gemini 3.5 Flash drops from $13.5\%$ at Low effort to $3.0\%$ at High effort, a drop of $10.5$ pp ($95\%$ CI $[-13.5, -7.7]$). This drop occurs because repetitive reasoning loops eat up steps, causing the model to hit the 20-step budget cap on $91.9\%$ of its failed rounds.

These results indicate that spending many reasoning tokens often signals planning inefficiency in interactive computer-use tasks: models fail when internal deliberation loops crowd out necessary actions in the environment.

### E.3 Execution Step Budget Saturation

Running out of steps (`cap-hit`) accounts for $0.7\%$–$11.4\%$ of failed rounds across six of the seven leading 20-step models (and only $1.0\%$ on GPT-6 Astra at its 40-step budget), but rises to $51.0\%$ on Gemini 3.7 Flash, and to $30.3\%$, $28.8\%$, and $91.9\%$ on Gemini 3.6 Flash, Gemini 3.1 Pro, and Gemini 3.5 Flash respectively. Gemini 3.7 Flash shows an interesting pattern: while achieving competitive overall pass rates, it spends $71.8\%$ of its output tokens on reasoning and runs out of steps on half of its failed rounds. Running out of steps is therefore a major bottleneck for reasoning-heavy models.

## F Extended Ablation Studies and Robustness Analyses

### F.1 Cross-Generator Invariance and Parallel Benchmark Design

To verify that benchmark difficulty and model rankings do not depend on the specific generator model, we regenerated the entire benchmark from the same 20 scenario seeds using Claude Sonnet 5 as the generator (Bench B) and compared it with Gemini 3.6 Flash (Bench A). Table F1 summarizes the structural parameters of both parallel benchmarks.

Table F1: **Structural comparison of parallel generation benchmarks.** Contrasts Bench A (Gemini 3.6 Flash) and Bench B (Claude Sonnet 5) across identical 20 scenario seeds and 600 aligned conflict tasks.

| | **Bench A** | **Bench B** |
|---|---|---|
| **Generator model** | Gemini 3.6 Flash | Claude Sonnet 5 |
| Step budget | 20 | 30 |
| GT step length, mean / median / max | 10.28 / 10 / 18 | 12.48 / 12 / 24 |
| Budget ratio (budget ÷ GT), mean | 2.02× | 2.51× |
| Rounds with budget ratio $< 1.5\times$ | 8.2% (49) | 1.2% (7) |
| Rounds with unexposed reference strings | 18 | 9 |
| *Pairing between the two benchmarks* | | |
| GT-length correlation (Pearson $r$) | 0.186 | |
| Bench B GT longer / equal / shorter | 430 / 72 / 98 | |
| Aligned rounds per benchmark | 600 | |
| Ratio-matched subset ($\lvert\Delta\rvert \leq 0.30$) | 164 | |

Evaluating seven models across 600 rounds per benchmark (Figure 2c, Table F2) confirms that relative model rankings remain highly stable. Rankings at high and low effort are consistent across Bench A and Bench B, with Spearman $\rho = 0.96$ at high effort, $1.00$ at low effort, and $0.97$ across all 19 comparisons, with no self-preference advantage for either generator model. Pass rate retention on Bench B relative to Bench A (**Ret%** in Figure 2c) places both generator models side-by-side in

the middle tier (0.51 for Gemini 3.6 Flash and 0.54 for Claude Sonnet 5), flanked by non-generator models Gemini 3.7 Flash at 0.41 and Claude Opus 4.8 at 0.57.

Table F2: **Comprehensive numerical results for harness parameter ablations.** *(a) Step budget:* Pass rates and cap exhaustion when expanding horizon $b = 20 \rightarrow 30 \rightarrow 40$ on Bench A ($\Delta = b_{40} - b_{20}$). *(b) Visual modality:* Performance of sighted vs. blind agents across full (600), asset-gated (346), and un-gated (254) task groups. *(c) Generator difference:* Pass rates across seven models evaluated on Bench A vs. Bench B (600 rounds each; $\Delta$ = B − A).

*(a) Step budget.* Same 600 rounds, high effort; only the cap moves.

| | Pass Rate (%) | | | | Ended at Cap (%) | |
|---|---|---|---|---|---|---|
| **Model** | **20** | **30** | **40** | **Δ** | **20** | **40** |
| Claude Opus 5 | 25.2 | 25.7 | 27.0 | +1.8 | 0.5 | 0.0 |
| Claude Sonnet 5 | 20.8 | 20.7 | 21.3 | +0.5 | 3.3 | 0.2 |
| Gemini 3.7 Flash | 20.3 | 26.0 | 27.2 | +6.8 | 40.7 | 2.5 |
| Gemini 3.6 Flash | 18.5 | 23.5 | 25.0 | +6.5 | 24.7 | 0.3 |
| Gemini 3.5 Flash | 3.0 | 10.0 | 13.0 | +10.0 | 89.2 | 44.0 |
| Gemini 3.8 Flash | 12.8 | 19.0 | 29.2 | +16.3 | 66.8 | 19.2 |

*(b) Visual modality.* Budget 20, high effort; assets withheld, task text unchanged.

| | All (600) | | Asset-Gated (346) | | No Asset Check (254) | |
|---|---|---|---|---|---|---|
| **Model** | **Sighted** | **Blind** | **Sighted** | **Blind** | **Sighted** | **Blind** |
| Claude Opus 5 | 25.2 | 4.2 | 25.4 | 5.2 | 24.8 | 2.8 |
| Claude Sonnet 5 | 20.8 | 2.8 | 18.5 | 2.6 | 24.0 | 3.1 |
| Gemini 3.6 Flash | 18.5 | 2.0 | 17.9 | 1.7 | 19.3 | 2.4 |

*(c) Generator model.* Pass rate (%) at high effort; 600 rounds per benchmark.

| **Benchmark** | **Opus 5** | **Opus 4.8** | **Sonnet 5** | **3.7 Flash** | **3.6 Flash** | **3.1 Pro** | **3.5 Flash** |
|---|---|---|---|---|---|---|---|
| A (Gemini 3.6 Flash) | 25.2 | 22.8 | 20.8 | 20.3 | 18.5 | 12.0 | 3.0 |
| B (Claude Sonnet 5) | 19.2 | 13.0 | 11.2 | 8.3 | 9.5 | 4.2 | 2.3 |
| Δ | −6.0 | −9.8 | −9.6 | −12.0 | −9.0 | −7.8 | −0.7 |

Paired McNemar, 19 model × effort cells (A easier / n.s. / B): 16/3/0 full, 14/5/0 ratio-matched ($n$=164).

### F.2 Action Budget Limits and Planning Horizon Dynamics

Increasing the action budget from $b = 20$ to 30 and 40 steps across identical scenarios (Figure 2a, Table F2) demonstrates that fixed step constraints do not penalize models uniformly. Instead, they act as an operational filter distinguishing three distinct agent planning behaviors:

1. **Concise Planners:** Models such as Claude Opus 5 and Claude Sonnet 5 rarely exhaust their budget even at $b = 20$ (cap exhaustion rates of 0.5% and 3.3%, respectively). Their trajectory lengths closely mirror human reference solutions (mean 10.28 steps, median 10, maximum 18; Table F1). Expanding the budget to $b = 40$ yields marginal pass rate gains (+1.8 pp for Claude Opus 5 and +0.5 pp for Claude Sonnet 5), establishing that a 20-step budget is generous for concise execution.
2. **Looping and Deliberation Traps:** Weaker reasoning architectures, most notably Gemini 3.5 Flash, hit the 20-step cap on 89.2% of rounds. While expanding the budget to $b = 40$ raises pass rate by +10.0 pp (from 3.0% to 13.0%), 44.1% of rounds still terminate at the cap. As detailed in Appendix §E, these extra steps are consumed by redundant file reading, hallucinated command repetitions, and excessive reasoning tokens (23,206 tokens/task at $b = 40$), inflating the cost per solved task to $8.69.
3. **Horizon-Constrained Frontier Competence:** In contrast to repetitive looping, Gemini 3.8 Flash demonstrates a capability profile characterized by deep investigative exploration and iterative hypothesis testing, discussed next.

### F.3 Frontier Horizon Sensitivity: The Gemini 3.8 Flash Case Study

To understand how frontier reasoning models interact with execution horizons, we conducted an extended evaluation of Gemini 3.8 Flash across four action budgets ($b = 20, 30, 40, 60$) over all 600 benchmark rounds under High reasoning effort on Vertex AI. All runs were evaluated by the corpus-standard Gemini 3.6 Flash judge. Table F3 summarizes empirical trajectory metrics, cap exhaustion, and token economics across the four budget arms.

Table F3: **Action horizon scaling for Gemini 3.8 Flash ($n = 600$ rounds, High effort, Gemini 3.6 Flash judge).** Metrics include task pass rate with 95% Wilson score confidence intervals, percentage of runs terminating at the step cap, average tool calls, average reasoning tokens, cost per task at introductory pricing, and effective cost per solved task.

| Budget ($b$) | Pass Rate (%) | 95% CI | Cap-Hit (%) | Calls/Task | Reason Tok | $/Solved |
|---|---|---|---|---|---|---|
| $b = 20$ | 12.8 (77/600) | [10.4, 15.7] | 66.8 | 19.4 | 6,699 | $4.44 |
| $b = 30$ | 19.0 (114/600) | [16.1, 22.3] | 46.8 | 27.3 | 8,976 | $4.40 |
| $b = 40$ | 29.2 (175/600) | [25.7, 32.9] | 19.2 | 32.2 | 11,233 | **$3.50** |
| $b = 60$ | **30.7** (184/600) | [27.1, 34.5] | **3.3** | 35.0 | 11,446 | $3.71 |
| *Comparator: Gemini 3.7 Flash (same judge, high effort)* | | | | | | |
| $b = 20$ | 20.3 (122/600) | [17.3, 23.7] | 40.7 | 17.8 | 4,026 | $2.51 |
| $b = 30$ | 26.0 (156/600) | [22.7, 29.7] | 13.7 | 23.4 | 5,341 | $2.60 |
| $b = 40$ | 27.2 (163/600) | [23.8, 30.9] | 2.5 | 26.0 | 5,879 | $2.80 |

**Threshold Effects Across Step Budgets.** The marginal return of expanding the budget for Gemini 3.8 Flash shows a clear threshold effect: increasing from $b = 20$ to $b = 30$ yields $+6.2$ pp ($12.8\% \rightarrow 19.0\%$), whereas expanding from $b = 30$ to $b = 40$ jumps by $+10.2$ pp ($19.0\% \rightarrow 29.2\%$). In complex multi-agent conflict scenarios, agents must first inspect contradictory logs, diagnose out-of-sync local states, and locate visual evidence before enacting coordinating fixes. High-capacity agents that conduct extensive diagnostic exploration require an operational threshold of steps before their solutions can be committed and verified. Once this threshold is surpassed, pass rates accelerate rapidly. Expanding further from $b = 40$ to $b = 60$ adds $+1.5$ pp to reach $30.7\%$, while cap exhaustion drops from $19.2\%$ to $3.3\%$. The saturation of gains alongside minimal cap exhaustion indicates that $30.7\%$ represents a genuine performance ceiling for the model architecture rather than an artificial truncation floor.

**Paired Head-to-Head Comparisons vs. Gemini 3.7 Flash.** Evaluating both models under identical prompts, seeds, and the corpus-standard Gemini 3.6 Flash judge allows rigorous paired McNemar tests ($n = 600$):

- At $b = 20$, Gemini 3.7 Flash leads by $+7.5$ pp ($20.3\%$ vs. $12.8\%$, $p < 10^{-4}$), driven by 3.7's lower step usage ($40.7\%$ vs. $66.8\%$ cap-hit).
- At $b = 30$, Gemini 3.7 Flash maintains a $+7.0$ pp lead ($26.0\%$ vs. $19.0\%$, discordant pairs 55/13, exact $p = 2.75 \times 10^{-7}$).
- At $b = 40$, Gemini 3.8 Flash overtakes Gemini 3.7 Flash by $+2.0$ pp ($29.2\%$ vs. $27.2\%$, discordant pairs 40/28, exact $p = 0.182$), becoming the highest-scoring model on the benchmark.
- When comparing both models at their respective saturation points ($b = 60$ for Gemini 3.8 Flash at $3.3\%$ cap-hit vs. $b = 40$ for Gemini 3.7 Flash at $2.5\%$ cap-hit), Gemini 3.8 Flash establishes a statistically significant $+3.5$ pp lead ($30.7\%$ vs. $27.2\%$, discordant pairs 41/20, exact $p = 0.00985$).

**Productive Execution vs. Repetitive Looping.** A critical distinction between Gemini 3.8 Flash and Gemini 3.5 Flash lies in their economic efficiency curves. For Gemini 3.5 Flash, budget expansion feeds repetitive thinking loops, driving reasoning tokens from 8,826 to 23,206 per task and resulting in a high cost of $8.69 per solved task at $b = 40$. In contrast, for Gemini 3.8 Flash, the additional budget is utilized for productive state verification and execution. Consequently, its effective cost per solved task drops from $4.44 at $b = 20$ to $3.50 at $b = 40$—a $21\%$ gain in economic efficiency. This confirms that Gemini 3.8 Flash's lower score at $b = 20$ is an artifact of horizon truncation rather than a lack of reasoning capability.

### F.4 Visual Necessity and Multimodal Grounding Ablation

Withholding visual images (Figure 2b, Table F2) causes pass rates to drop sharply across all models. Even on the 254 rounds without a mandatory image check, blind agents keep only $11.1\%$ to $13.1\%$ of their sighted pass rates ($24.8\% \rightarrow 2.8\%$ for Claude Opus 5, $24.0\% \rightarrow 3.1\%$ for Claude Sonnet 5, and $19.3\% \rightarrow 2.4\%$ for Gemini 3.6 Flash). With Bonferroni correction ($0.05/6$), all three drops are statistically significant (largest $p = 2.6 \times 10^{-12}$), with 60, 54, and 44 rounds solved only when images are available versus just 4, 1, and 1 when blind. This confirms that visual evidence is essential for resolving conflicts, rather than decorative background.

### F.5 Judge Selection and Cross-Judge Stability

**Offline Replay Protocol.** To test whether benchmark results depend on the specific evaluator model, we replayed all $9{,}000$ recorded high-effort rounds from the 14 20-step baseline models offline against original scenario data and reconstructed checker inputs from stored action trajectories. Deterministic checks were re-evaluated as a fidelity verification: all $24{,}331$ deterministic checks reproduced their recorded verdicts ($100.0\%$ fidelity), confirming that differences between evaluation runs stem entirely from semantic judging. The replay generated $28{,}009$ judge prompts, which were re-evaluated by three substitute judges using 3-sample majority voting: Gemini 3.6 Flash (the default judge re-run to estimate sampling variance), Claude Sonnet 5, and Claude Opus 5.

**Common Evaluation Subset.** Model endpoints occasionally decline prompts: Claude Sonnet 5 declined $161$ prompts ($0.62\%$) and Claude Opus 5 declined $3{,}089$ prompts ($11.8\%$), while the default judge completed all evaluations. To maintain fair comparisons across judges without missing-data bias, all reported metrics evaluate the common subset of $22{,}938$ checks ($87.9\%$) scored by all three judges, using recorded verdicts for un-scored prompts uniformly across columns.

**Evaluator Consistency.** Each judge ensemble demonstrates high internal consistency across its three voting samples. Disagreement rates among the three samples were $2.78\%$ for Gemini 3.6 Flash, $1.31\%$ for Claude Sonnet 5, and $0.34\%$ for Claude Opus 5, confirming stable majority verdicts across all evaluators.

**Results and Rank Invariance.** Table F4 summarizes evaluation results across substitute judges. Rankings on mean score show high stability across all evaluators, with rank correlations of $\rho = 0.987$ for re-run Gemini 3.6 Flash, $\rho = 0.991$ for Claude Sonnet 5, and $\rho = 0.987$ for Claude Opus 5. No model shifts by more than one rank position under either Claude judge.

Different judges affect overall score levels rather than ordinal rankings. On average, Claude Opus 5 gives slightly higher scores ($+0.021$ on mean score, two-sided sign test $p = 0.00012$) and Claude Sonnet 5 shows a modest increase ($+0.005$, $p = 0.0018$), whereas re-running the default judge produces no systematic shift ($p = 0.79$). The only rank change among top models occurs between GPT-5.6 Sol and Claude Opus 5, whose small pass rate difference of $0.67$ pp sits comfortably within bootstrap standard errors (Table 1). Overall, the top four and trailing four models remain identical across all judges, confirming that relative benchmark standings are robust to evaluator choice.

**Analysis of the Gemini 3.7 Flash Rollouts.** Gemini 3.7 Flash includes two evaluation rollouts: the first evaluated by Gemini 3.6 Flash, and the second evaluated by Gemini 3.7 Flash itself ($+1.67$ pp difference). Re-grading both rollouts with a common judge separates rollout variance from judge scoring differences: judge differences account for $+1.50$ pp of the gap, while rollout variance accounts for $+0.83$ pp (paired McNemar $p = 0.63$). We report the published run with an explicit self-scoring note for transparency.

Table F4: **Cross-judge stability ablation.** Every semantic check of the high-effort panel is re-evaluated offline by three substitute three-sample ensembles with agent trajectories held fixed: the replay reproduces $24{,}331/24{,}331$ deterministic verdicts, so the columns differ only in semantic opinion. The Gemini 3.6 Flash column re-runs the default judge, so it measures sampling noise rather than a judge change. Statistics use the $22{,}938$ of $26{,}100$ panel checks ($87.9\%$) all three substitutes scored; the recorded verdict fills the rest in every column and cancels in cross-judge comparisons. Mean score leads because pass rate requires passing all checks, so one flipped check can move a model several places. The substitute judges reorder the models no more than the default judge reorders itself; they primarily shift absolute score levels. Claude Opus 5 raises all fourteen 20-step baseline models, unevenly enough to hand first place to GPT-5.6 Sol—the only change of leader in the table. The self-scored row that the two *excluding* lines drop is Gemini 3.7 Flash, which issued its own recorded verdicts.

| **Substitute judge** | Gemini 3.6 Flash | Claude Sonnet 5 | Claude Opus 5 |
|---|---|---|---|
| *Mean score (primary).* | | | |
| Rank correlation $\rho$ | 0.987 | 0.991 | 0.987 |
| excluding the self-scored row | 0.995 | 0.995 | 0.989 |
| Maximum rank displacement | 2 | 1 | 1 |
| Models whose rank moves | 3/14 | 4/14 | 6/14 |
| Mean shift | $-0.001$ | $+0.005$ | $+0.021$ |
| Largest absolute shift | 0.007 | 0.018 | 0.045 |
| Models scored higher | 6/14 | 13/14 | 14/14 |
| sign test $p$ (two-sided) | 0.79 | 0.0018 | 0.00012 |
| *Pass rate.* | | | |
| Rank correlation $\rho$ | 0.975 | 0.991 | 0.949 |
| excluding the self-scored row | 0.982 | 0.990 | 0.974 |
| Maximum rank displacement | 3 | 1 | 3 |
| Models whose rank moves | 3/14 | 6/14 | 8/14 |
| Largest move (points) | $-0.7$ | $+1.3$ | $+3.7$ |
| *Per-check agreement with the recorded verdicts.* | | | |
| Agreement | 98.93% | 96.31% | 95.46% |
| Cohen's $\kappa$ | 0.977 | 0.922 | 0.902 |
| Fail $\rightarrow$ pass flips | 99 | 508 | 958 |
| Pass $\rightarrow$ fail flips | 147 | 339 | 83 |
| Declined prompts (of 26,100) | 0 | 161 | 3,089 |
| Non-unanimous ensembles | 2.78% | 1.31% | 0.34% |

# G Exhibits from the Cross-Model Behavioral Evaluation

This appendix provides the complete set of empirical exhibits, diagnostic tables, vector charts, and behavioral panels compiled during the benchmark evaluation across all 15 evaluated models.

**Experimental Configuration.** Unless otherwise indicated, all reported evaluations correspond to the default high reasoning effort configuration (`reasoning_effort=high`). The reasoning effort sweep across Low, Medium, and High levels for the ten API models is detailed in §G.9. Open-weight models are evaluated at their native high-effort configuration.

**Evaluation Runs.** All reported models are evaluated over the complete 600-round benchmark corpus. For Gemini 3.7 Flash, results correspond to run `r02` (20.3% vs. 18.7% on `r01`, paired McNemar $p = 0.27$). All comparative claims in the main text are supported by paired statistical significance tests or bootstrap confidence intervals.

**Exhibit Overview.** Tables G1–G23 (G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, G14, G15, G16, G17, G18, G19, G20, G21, G22, G23) reproduce empirical evaluation records across all tasks. Vector charts and heatmap panels in Figures G1–G22 (G1, G2, G3, G4, G6, G11, G12, G13, G14, G15, G16, G17, G18, G19, G20, G21, G22) visualize behavioral distributions, error profiles, and tool execution traces across the full 600-round corpus.

## Model Roster and Evaluation Configuration

| Model | model id | provider | reasoning_effort | run |
|---|---|---|---|---|
| Gemini 3.5 Flash | gemini-3.5-flash | Gemini | low / medium / high | r01 |
| Gemini 3.6 Flash | gemini-3.6-flash | Gemini | low / medium / high | r01 |
| Gemini 3.7 Flash | gemini-3.7-flash | Gemini | low / medium / high | **r02** |
| Gemini 3.1 Pro | gemini-3.1-pro-preview | Gemini | low / medium / high | r01 |
| Claude Opus 4.8 | claude-opus-4-8 | Claude | low / medium / high | r01 |
| Claude Opus 5 | claude-opus-5 | Claude | low / medium / high | r01 |
| Claude Sonnet 5 | claude-sonnet-5 | Claude | low / medium / high | r01 |
| GPT-5.6 Sol | openai.gpt-5.6-sol | OpenAI | low / medium / high | r01 |
| GPT-5.6 Luna | openai.gpt-5.6-luna | OpenAI | low / medium / high | r01 |
| GPT-5.6 Terra | openai.gpt-5.6-terra | OpenAI | low / medium / high | r01 |
| GPT-6 Astra | openai.gpt-6-astra | OpenAI | high only (not swept) | r01 |
| DeepSeek V4 Pro | deepseek-ai/DeepSeek-V4-Pro | Open-source | high only (not swept) | r01 |
| DeepSeek V4 Flash | deepseek-ai/DeepSeek-V4-Flash-0731 | Open-source | high only (not swept) | r01 |
| Kimi K3 | moonshotai/Kimi-K3 | Open-source | high only (not swept) | r01 |
| GLM-5.2 FP8 | zai-org/GLM-5.2-FP8 | Open-source | high only (not swept) | r01 |

Table G1: **Model evaluation roster and configuration coverage.** Details model identifiers, provider platforms, reasoning effort levels, and evaluation run identifiers across the 15 evaluated models.

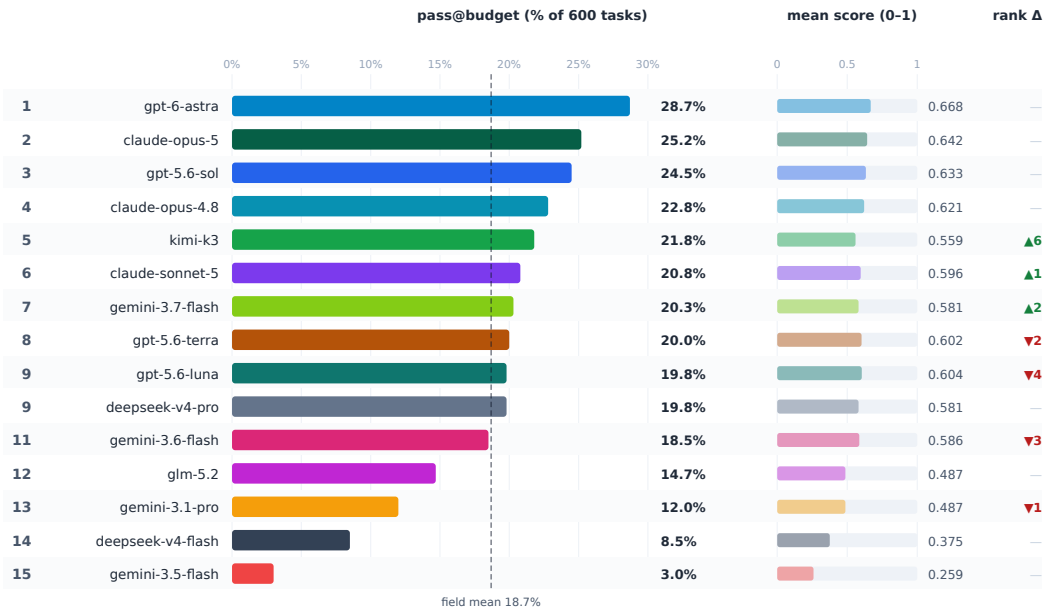



Figure G1: **Overall benchmark standings across the 15 evaluated models (High Effort).** Displays task pass rate (pass@budget, wide bar) and mean check score (narrow bar, 0–1 scale) across all 600 benchmark tasks. Models are ranked by pass rate; Rank Δ indicates the difference between mean score rank and pass rank.

| action category | sub-commands mapped into it (observed in this dataset) |
|---|---|
| open asset | `asset open` |
| inspect asset (list/info) | `asset describe`, `asset info`, `asset list`, `asset read-text` |
| file read | `file read` |
| file list | `file list` |
| file write/create | `file append`, `file create` |
| file delete | `file delete` |
| message send | `message broadcast`, `message send` |
| message search/read | `message poll`, `message search` |
| agent delegate | `agent delegate` |
| agent status | `agent add`, `agent list`, `agent set-identity`, `agent status` |
| wait | `wait 1`, `wait 10`, `wait 120`, `wait 15`, `wait 180`, `wait 2`, `wait 20`, `wait 200`, `wait 3`, `wait 30`, `wait 300`, `wait 4`, `wait 45`, `wait 5`, `wait 6`, `wait 60`, `wait 600`, `wait 8`, `wait 90` |
| other | domain operations: `approval (grant/deny/request)`, `booking`, `containment`, `deploy (gate-set/request/cutover)`, `lock (acquire/release)`, `pairing`, `review (submit/approve)`, `system`, `tasks`, `undo` — every verb enumerated in the command appendix |

Figure G2: **Action category glossary and invocation distribution.** Partitions all 88 sub-commands in the action DSL into 12 mutually exclusive functional categories (call rates normalized per task across all 600 rounds). The detailed breakdown of domain-specific operations is provided in Figure G7, and the full command inventory spans Figures G8–G10.

### G.1 Behavioral signatures at a glance

| Model | pass@ budget | passed | avg score | step p50 | mean steps | capped | opens/ task | writes/ task | zero-open | $/ task | $/ solved |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Gemini 3.5 Flash | **3.0%** | **18** | **0.259** | 20.0 | 19.70 | **535** | 3.56 | **0.31** | 18 | 1.1861 | **39.5374** |
| Gemini 3.6 Flash | 18.5% | 111 | 0.586 | 16.5 | 16.30 | 148 | 2.90 | 0.99 | 5 | 0.9414 | 5.0889 |
| Gemini 3.7 Flash | 20.3% | 122 | 0.581 | 19.0 | 17.80 | 244 | 2.25 | 0.91 | 60 | 0.5110 | 2.5130 |
| Gemini 3.1 Pro | 12.0% | 72 | 0.487 | 12.0 | 13.20 | 152 | 3.15 | 1.00 | 14 | 0.9630 | 8.0251 |
| Claude Opus 4.8 | 22.8% | 137 | 0.621 | 12.0 | 12.20 | 37 | 2.71 | 1.04 | 1 | 4.8879 | 21.4070 |
| Claude Opus 5 | 25.2% | 151 | 0.642 | 11.0 | 11.10 | **3** | 2.98 | 1.14 | 1 | 4.3843 | 17.4209 |
| Claude Sonnet 5 | 20.8% | 125 | 0.596 | **10.0** | **10.60** | 20 | **1.83** | 1.05 | **92** | 1.7188 | 8.2502 |
| GPT-5.6 Sol | 24.5% | 147 | 0.633 | 14.0 | 13.70 | 7 | 4.08 | 1.24 | 0 | 3.6082 | 14.7273 |
| GPT-5.6 Luna | 19.8% | 119 | 0.604 | 13.0 | 13.50 | 55 | 3.90 | **1.43** | 0 | 0.1429 | 0.7207 |
| GPT-5.6 Terra | 20.0% | 120 | 0.602 | 11.0 | 11.40 | 8 | 2.65 | 1.24 | 1 | 1.2180 | 6.0898 |
| GPT-6 Astra | **28.7%** | **172** | **0.668** | **27.0** | **26.09** | 6 | **9.66** | 1.21 | 0 | **5.0103** | 17.4605 |
| DeepSeek V4 Pro | 19.8% | 119 | 0.581 | 13.0 | 13.40 | 81 | 4.36 | 1.00 | 0 | **0.0534** | **0.2690** |
| DeepSeek V4 Flash | 8.5% | 51 | 0.375 | 20.0 | 17.80 | 358 | 6.00 | 0.58 | 0 | n/a | n/a |
| Kimi K3 | 21.8% | 131 | 0.559 | 16.0 | 15.40 | 170 | 6.72 | 0.88 | 3 | 3.6210 | 16.5846 |
| GLM-5.2 FP8 | 14.7% | 88 | 0.487 | 15.0 | 14.90 | 207 | 7.26 | 0.78 | 0 | 0.2006 | 1.3677 |

Table G2: **High-level behavioral signatures across models (High Effort).** Summary metrics across all 600 benchmark tasks per model, including pass rate (`pass@budget`), count of passed rounds, mean check score, step statistics (median, mean, and frequency of running out of steps), average image opens and file writes per task, zero-open frequency, and cost per round and per solved task.

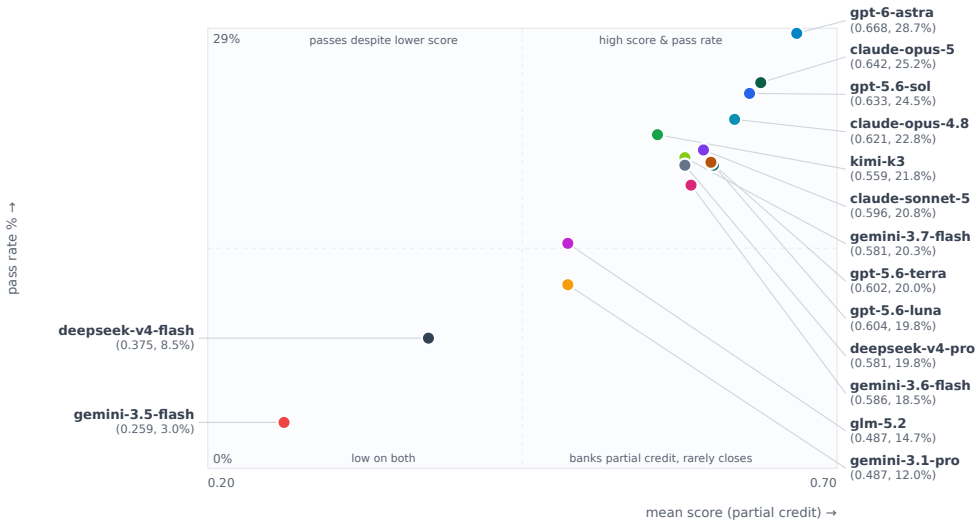



Figure G3: **Pass rate versus mean check score across models (High Effort).** Shows the relationship between all-or-nothing task completion (pass rate) and mean check score across all 15 models.

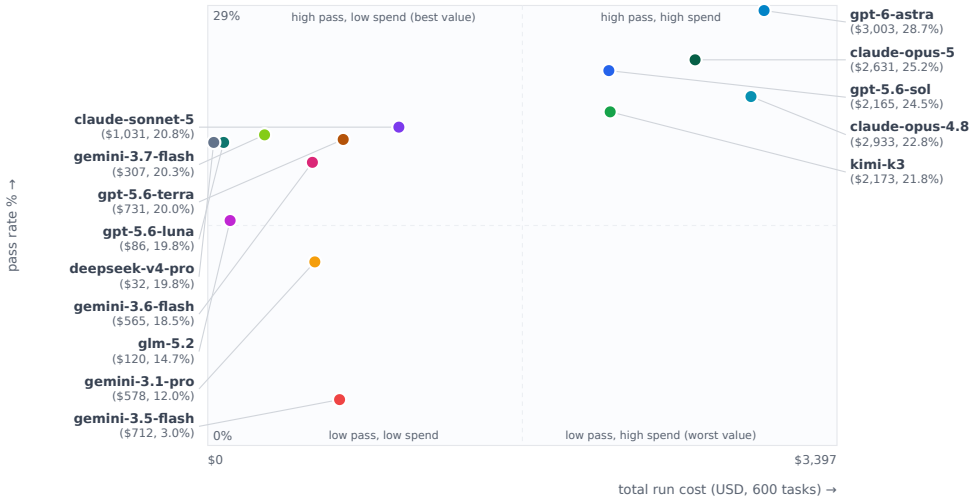



Figure G4: **Pass rate versus cumulative evaluation cost (High Effort).** Compares benchmark pass rate against total evaluation cost across models.

| model | input tok/task | imgs @ step 0 | tok / image (s0) | output tok/task | total tok/task | $ / task | total tok (run, 600) | $ total (run, 600) | $ / solved task | time per task (s) |
|---|---|---|---|---|---|---|---|---|---|---|
| gemini-3.5-flash | 732,244 | 13.3 | 2,207 | 9,751<br>91% reasoning | **741,995** | $1.186 | 445,196,979 | **$711.67** | $39.537 | 242.1 |
| gemini-3.6-flash | 598,689 | 13.3 | 2,207 | 5,788<br>74% reasoning | **604,477** | $0.941 | 362,685,881 | **$564.86** | $5.089 | 201.3 |
| gemini-3.7-flash | 651,786 | 13.3 | 2,207 | 5,902<br>72% reasoning | **657,688** | $0.511 | 394,612,164 | **$306.58** | $2.513 | 195.4 |
| gemini-3.1-pro | 462,842 | 13.3 | 2,207 | 3,111<br>76% reasoning | **465,953** | $0.963 | 279,571,681 | **$577.81** | $8.025 | 349.5 |
| claude-opus-4.8 | 944,427 | 13.3 | 4,740 | 6,632<br>35% reasoning | **951,059** | $4.888 | 570,635,183 | **$2,932.75** | $21.407 | 273.2 |
| claude-opus-5 | 845,483 | 13.3 | 4,740 | 6,273<br>9% reasoning | **851,756** | $4.384 | 511,054,388 | **$2,630.56** | $17.421 | 268.9 |
| claude-sonnet-5 | 820,212 | 13.3 | 4,740 | 7,836<br>36% reasoning | **828,048** | $1.719 | 496,829,091 | **$1,031.27** | $8.250 | 218.4 |
| GPT-6 Astra | 1,101,390 | 13.3 | 3,721 | 3,235<br>60% reasoning | 1,104,624 | $5.005 | 662,774,572 | $3,003.00 | $17.459 | 776.7 |
| gpt-5.6-sol | 754,231 | 13.3 | 3,721 | 3,063<br>44% reasoning | **757,294** | $3.608 | 454,376,026 | **$2,164.91** | $14.727 | 368.1 |
| gpt-5.6-luna | 746,156 | 13.3 | 3,721 | 3,666<br>54% reasoning | **749,822** | $0.143 | 449,893,547 | **$85.77** | $0.721 | 620.5 |
| gpt-5.6-terra | 633,397 | 13.3 | 3,721 | 3,062<br>57% reasoning | **636,459** | $1.218 | 381,875,371 | **$730.78** | $6.090 | 249.7 |
| deepseek-v4-pro | 112,166 | 0.0 | – | 5,239<br>split not reported | **117,405** | $0.053 | 70,443,269 | **$32.01** | **$0.269** | 213.0 |
| deepseek-v4-flash | 155,571 | 0.0 | – | 3,313<br>split not reported | **158,884** | N/A | 95,330,132 | N/A | N/A | **126.7** |
| kimi-k3 | 1,168,663 | 13.3 | 4,786 | 7,666<br>split not reported | **1,176,329** | $3.621 | 705,797,501 | **$2,172.59** | $16.585 | 963.8 |
| glm-5.2-fp8 | 132,361 | 0.0 | – | 3,474<br>split not reported | **135,835** | $0.201 | 81,500,981 | **$120.35** | $1.368 | 200.6 |

Figure G5: **Token volume, monetary spend, and execution latency across models (High Effort).** Breaks down token usage per task (input prompt, visible generation, internal reasoning), cost per task and per solved task, and execution latency across evaluated models.

**Compute Cost Accounting.** Across all 13 priced baseline models, evaluating the 600-round benchmark at High effort required 5,204,472,062 tokens ($\sim$\$14,061.91 across 7,800 model runs; \$17,065.11 across 14 priced models including GPT-6 Astra). Cost per solved task measures the average dollar spend required to successfully complete a task, spanning a 55-fold range from \$0.72 on GPT-5.6 Luna to \$39.54 on Gemini 3.5 Flash (\$17.46 on GPT-6 Astra). Pricing is based on published API pricing at the time of evaluation.

**Multimodal Tokenization and Context Accumulation.** Input tokens make up most of the token volume due to multi-turn execution history and high-resolution screenshots ($3{,}840 \times 2{,}160$). Different model tokenizers encode these images into varying token counts ($\sim$2.2k tokens on Gemini, $\sim$3.7k on OpenAI, $\sim$4.7k on Claude, and $\sim$4.8k on Kimi).

**Deliberation Allocation and Output Density.** Models differ sharply in how many tokens they spend on reasoning: Claude Opus 5 spends only $8.8\%$ of its output tokens on reasoning, whereas Gemini models allocate $71.8\%$–$90.5\%$ of their output tokens to internal reasoning.

## G.2 ACTION-CATEGORY MIX

| Model | total | asset_open | asset_inspect | file_read | file_list | file_write | file_delete | message_send | message_read | agent_delegate | agent_status | wait | other |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gemini 3.5 Flash | 11,820 | 18.1% | 8.3% | **28.3%** | 5.2% | **1.6%** | 0.0% | **6.5%** | **22.9%** | 0.1% | 1.7% | 2.3% | 5.1% |
| Gemini 3.6 Flash | 9,786 | 17.8% | 7.6% | 25.9% | **6.1%** | 6.1% | 0.0% | 16.4% | 12.3% | 0.1% | 0.6% | **0.6%** | 6.4% |
| Gemini 3.7 Flash | 10,674 | **12.7%** | 8.0% | 26.4% | 6.0% | 5.1% | 0.0% | 19.1% | 11.1% | 0.2% | **3.3%** | 0.7% | 7.4% |
| Gemini 3.1 Pro | 7,898 | 24.0% | 12.7% | 12.2% | 2.0% | 7.6% | 0.0% | 16.9% | 10.9% | 0.4% | 1.4% | **5.1%** | 7.0% |
| Claude Opus 4.8 | 7,303 | 22.3% | 9.6% | 19.4% | 3.0% | 8.6% | 0.0% | 21.6% | 5.8% | 0.4% | 0.5% | 1.7% | 7.2% |
| Claude Opus 5 | 6,672 | 26.8% | 9.1% | 16.7% | 1.4% | 10.3% | 0.0% | **23.5%** | **2.0%** | 0.2% | 0.3% | 1.3% | **8.3%** |
| Claude Sonnet 5 | **6,385** | 17.2% | **15.6%** | 17.6% | 1.1% | 9.9% | 0.0% | 20.9% | 6.2% | 0.2% | 0.5% | 3.4% | 7.3% |
| GPT-5.6 Sol | 8,191 | 29.9% | 7.8% | 15.2% | 0.9% | 9.1% | 0.0% | 16.3% | 10.7% | 0.4% | 0.8% | 2.9% | 6.0% |
| GPT-5.6 Luna | 8,125 | 28.8% | 8.3% | 17.7% | 1.2% | 10.5% | 0.0% | 18.3% | 6.4% | 0.2% | 0.6% | 2.5% | 5.4% |
| GPT-5.6 Terra | 6,812 | 23.4% | 9.6% | 17.7% | 1.0% | **10.9%** | 0.0% | 19.1% | 7.6% | 0.3% | 0.9% | 3.9% | 5.7% |
| GPT-6 Astra | **15,656** | 37.0% | 9.3% | 14.8% | 2.7% | 4.6% | 0.0% | 14.0% | 9.3% | 0.3% | 0.6% | 3.0% | **4.5%** |
| DeepSeek V4 Pro | 8,051 | 32.5% | 7.6% | 13.9% | 1.6% | 7.5% | 0.0% | 19.1% | 9.5% | 0.3% | 0.6% | 2.0% | 5.6% |
| DeepSeek V4 Flash | 10,674 | 33.7% | **5.9%** | 15.7% | 5.7% | 3.3% | 0.0% | 9.0% | 20.1% | 0.2% | 0.6% | 0.8% | 4.9% |
| Kimi K3 | 9,251 | 43.6% | 7.4% | 16.0% | 1.7% | 5.7% | 0.0% | 10.7% | 6.7% | 0.2% | 0.4% | 1.4% | 6.1% |
| GLM-5.2 FP8 | 8,933 | **48.8%** | 6.9% | **11.3%** | **0.8%** | 5.3% | 0.0% | 14.2% | 6.6% | 0.1% | **0.2%** | 1.0% | 5.0% |

Table G3: **Action category distribution across models (High Effort).** Reports the percentage allocation of tool calls across 12 functional action verbs over all 600 tasks per model.




Figure G6: **Empirical action-category distributions across models (High Effort).** Relative proportion of tool calls across the 12 action categories. Each bar sums to 100% of that model's executed actions. The domain-specific operations grouped under `other` are detailed in Figure G7.

| domain family (inside `other`) | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| deploy (gate-set/request/cutover) | 177 | 190 | 225 | 152 | 153 | 138 | 140 | 183 | 133 | 123 | 114 | 115 | 157 | 163 | 121 |
| booking | 107 | 112 | 114 | 121 | 86 | 101 | 90 | 151 | 81 | 81 | 64 | 91 | 75 | 127 | 90 |
| approval (grant/deny/request) | 66 | 93 | 124 | 76 | 60 | 72 | 47 | 93 | 73 | 65 | 59 | 66 | 60 | 65 | 70 |
| pairing | 61 | 64 | 69 | 65 | 63 | 55 | 51 | 61 | 58 | 56 | 47 | 57 | 52 | 59 | 38 |
| containment | 31 | 51 | 63 | 53 | 75 | 99 | 65 | 71 | 63 | 47 | 42 | 44 | 57 | 61 | 43 |
| review (submit/approve) | 44 | 68 | 79 | 33 | 44 | 41 | 35 | 65 | 37 | 29 | 29 | 32 | 39 | 41 | 48 |
| lock (acquire/release) | 47 | 37 | 42 | 32 | 39 | 44 | 38 | 59 | 40 | 31 | 28 | 40 | 39 | 45 | 31 |
| tasks | 26 | 12 | 75 | 6 | 2 | 2 | 2 | 5 | 3 | 2 | 2 | 1 | 35 | 4 | 4 |
| system | 38 | 0 | 2 | 7 | 3 | 2 | 1 | 11 | 1 | 4 | 4 | 2 | 9 | 2 | 1 |
| undo | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| OTHER TOTAL | 597 | 627 | 793 | 549 | 525 | 554 | 469 | 701 | 489 | 438 | 390 | 448 | 523 | 567 | 446 |

Figure G7: **Itemized distribution of domain-specific operations (`other` category).** Breaks down specialized environment operations (including containment locks, deployment approvals, resource pairing, and process scheduling) across models.

*Not every row here is like-for-like:* DeepSeek V4 Pro, DeepSeek V4 Flash and GLM-5.2 FP8 read their assets as text rather than as images, and Gemini 3.7 Flash is reported on `r02`, the higher of its two runs. Full run conditions under Figure G5.

### G.3 COMMAND-SURFACE DEEP-DIVE

| Model | total | asset open | asset list | file read | message search | file create | message send |
|---|---|---|---|---|---|---|---|
| Gemini 3.5 Flash | 11,820 | 2,137 | 604 | **3,342** | **2,707** | **187** | **747** |
| Gemini 3.6 Flash | 9,786 | 1,740 | 606 | 2,538 | 1,206 | 593 | 1,593 |
| Gemini 3.7 Flash | 10,674 | 1,351 | 603 | 2,814 | 1,188 | 547 | 1,704 |
| Gemini 3.1 Pro | 7,898 | 1,892 | 662 | **966** | 861 | 581 | 1,283 |
| Claude Opus 4.8 | 7,303 | 1,625 | 604 | 1,416 | 427 | 617 | 1,534 |
| Claude Opus 5 | 6,672 | 1,786 | 604 | 1,117 | **134** | 669 | 1,502 |
| Claude Sonnet 5 | **6,385** | **1,100** | 616 | 1,125 | 397 | 612 | 1,334 |
| GPT-5.6 Sol | 8,191 | 2,448 | 619 | 1,248 | 877 | 708 | 1,337 |
| GPT-5.6 Luna | 8,125 | 2,341 | 629 | 1,437 | 522 | **773** | 1,480 |
| GPT-5.6 Terra | 6,812 | 1,592 | 610 | 1,206 | 516 | 727 | 1,286 |
| GPT-6 Astra | **15,656** | **5,799** | **1,439** | 2,310 | 1,452 | 642 | **2,136** |
| DeepSeek V4 Pro | 8,051 | 2,618 | 612 | 1,117 | 760 | 595 | 1,533 |
| DeepSeek V4 Flash | 10,674 | 3,601 | 607 | 1,679 | 2,137 | 335 | 961 |
| Kimi K3 | 9,251 | 4,030 | **601** | 1,476 | 621 | 514 | 992 |
| GLM-5.2 FP8 | 8,933 | 4,355 | 606 | 1,013 | 587 | 468 | 1,265 |

Table G4: **Command surface invocation frequencies (High Effort).** Raw execution counts for individual CLI sub-commands across all 600 tasks per model.

| sub-command | category | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 | total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| asset open | open asset | 2137 | 1740 | 1351 | 1892 | 1625 | 1786 | 1100 | 5799 | 2448 | 2341 | 1592 | 2618 | 3601 | 4030 | 4355 | 38415 |
| asset list | inspect asset (list/info) | 604 | 606 | 603 | 662 | 604 | 604 | 616 | 1439 | 619 | 629 | 610 | 612 | 607 | 601 | 606 | 10022 |
| asset info | inspect asset (list/info) | 306 | 140 | 245 | 242 | 79 | 5 | 330 | 15 | 16 | 40 | 45 | 1 | 17 | 59 | 5 | 1545 |
| asset describe | inspect asset (list/info) | 48 | 1 | 2 | 61 | 11 | 0 | 44 | 0 | 1 | 3 | 1 | 2 | 5 | 13 | 0 | 192 |
| asset read-text | inspect asset (list/info) | 24 | 0 | 0 | 36 | 5 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 1 | 82 |
| file read | file read | 3342 | 2538 | 2814 | 966 | 1416 | 1117 | 1125 | 2310 | 1248 | 1437 | 1206 | 1117 | 1679 | 1476 | 1013 | 24804 |
| file list | file list | 612 | 598 | 640 | 157 | 220 | 92 | 73 | 419 | 76 | 101 | 66 | 125 | 612 | 158 | 72 | 4021 |
| file create | file write/create | 187 | 593 | 547 | 581 | 617 | 669 | 612 | 642 | 708 | 773 | 727 | 595 | 335 | 514 | 468 | 8568 |
| file append | file write/create | 0 | 0 | 0 | 17 | 9 | 16 | 20 | 85 | 35 | 82 | 16 | 5 | 16 | 13 | 2 | 316 |
| file delete | file delete | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| message send | message send | 747 | 1593 | 1704 | 1283 | 1534 | 1502 | 1334 | 2136 | 1337 | 1480 | 1286 | 1533 | 961 | 992 | 1265 | 20687 |
| message broadcast | message send | 25 | 14 | 338 | 50 | 47 | 69 | 2 | 52 | 2 | 6 | 13 | 1 | 0 | 1 | 5 | 625 |
| message search | message search/read | 2707 | 1206 | 1188 | 861 | 427 | 134 | 397 | 1452 | 877 | 522 | 516 | 760 | 2137 | 621 | 587 | 14392 |
| message poll | message search/read | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 9 | 0 | 0 | 10 |
| agent delegate | agent delegate | 16 | 14 | 19 | 32 | 26 | 14 | 14 | 52 | 30 | 19 | 19 | 24 | 25 | 21 | 9 | 334 |
| agent list | agent status | 181 | 36 | 325 | 41 | 12 | 2 | 4 | 32 | 10 | 8 | 8 | 9 | 30 | 8 | 5 | 711 |
| agent status | agent status | 18 | 22 | 30 | 65 | 24 | 18 | 26 | 57 | 46 | 41 | 43 | 33 | 29 | 30 | 8 | 490 |
| agent add | agent status | 2 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 7 | 3 | 3 | 3 | 0 | 1 | 1 | 24 |
| agent set-identity | agent status | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 7 |
| wait 10 | wait | 130 | 29 | 13 | 290 | 0 | 0 | 10 | 42 | 35 | 15 | 63 | 37 | 32 | 23 | 20 | 739 |
| wait 5 | wait | 99 | 23 | 46 | 43 | 0 | 0 | 17 | 32 | 94 | 48 | 124 | 106 | 33 | 3 | 20 | 688 |
| wait 30 | wait | 27 | 0 | 5 | 29 | 77 | 11 | 127 | 183 | 13 | 2 | 8 | 7 | 5 | 70 | 7 | 571 |
| wait 60 | wait | 5 | 0 | 0 | 6 | 25 | 54 | 39 | 105 | 4 | 1 | 1 | 1 | 3 | 11 | 0 | 255 |
| wait 3 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 40 | 23 | 0 | 0 | 0 | 0 | 90 |
| wait 15 | wait | 0 | 0 | 5 | 2 | 0 | 0 | 4 | 9 | 5 | 0 | 6 | 8 | 7 | 6 | 31 | 83 |
| wait 2 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 31 | 38 | 4 | 0 | 0 | 0 | 0 | 75 |
| wait 1 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 46 | 10 | 46 | 1 | 0 | 0 | 0 | 0 | 103 |
| wait 300 | wait | 3 | 4 | 4 | 5 | 4 | 6 | 4 | 13 | 4 | 4 | 4 | 4 | 3 | 6 | 2 | 70 |
| wait 20 | wait | 2 | 1 | 0 | 21 | 1 | 0 | 3 | 1 | 9 | 1 | 6 | 1 | 5 | 1 | 5 | 57 |

Figure G8: **Complete command inventory across all evaluated models (Part 1: Rows 1–30 of 89).** Displays individual sub-command invocation counts across all 600 tasks per model, mapped to functional categories. Continued in Figure G9.

| sub-command | category | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 | total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| wait 120 | wait | 1 | 0 | 0 | 6 | 8 | 9 | 3 | 21 | 2 | 0 | 0 | 0 | 0 | 3 | 0 | 53 |
| wait 8 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 17 | 0 | 0 | 0 | 0 | 19 |
| wait 45 | wait | 0 | 0 | 0 | 0 | 0 | 3 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 15 |
| wait 4 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 6 | 0 | 0 | 0 | 0 | 12 |
| wait 180 | wait | 0 | 0 | 2 | 0 | 2 | 3 | 1 | 2 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 13 |
| wait 90 | wait | 0 | 0 | 0 | 0 | 4 | 4 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 12 |
| wait 6 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 4 |
| wait 200 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| wait 600 | wait | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| deploy status | other | 87 | 90 | 108 | 53 | 58 | 70 | 59 | 87 | 43 | 39 | 32 | 37 | 79 | 61 | 47 | 950 |
| booking status | other | 83 | 72 | 77 | 72 | 51 | 61 | 56 | 109 | 46 | 36 | 34 | 51 | 51 | 81 | 42 | 922 |
| deploy gate-set | other | 59 | 63 | 80 | 65 | 66 | 38 | 53 | 66 | 60 | 51 | 52 | 51 | 49 | 73 | 53 | 879 |
| containment status | other | 21 | 24 | 35 | 20 | 35 | 42 | 26 | 38 | 22 | 16 | 14 | 19 | 35 | 24 | 20 | 391 |
| approval status | other | 35 | 35 | 64 | 22 | 15 | 23 | 9 | 43 | 21 | 14 | 10 | 17 | 26 | 16 | 25 | 375 |
| approval approve | other | 14 | 32 | 32 | 24 | 25 | 23 | 19 | 23 | 25 | 23 | 25 | 23 | 15 | 21 | 22 | 346 |
| lock status | other | 35 | 24 | 28 | 16 | 21 | 24 | 22 | 36 | 23 | 17 | 16 | 23 | 26 | 25 | 16 | 352 |
| pairing status | other | 28 | 26 | 28 | 16 | 18 | 16 | 13 | 22 | 13 | 12 | 6 | 15 | 21 | 20 | 6 | 260 |
| review status | other | 24 | 29 | 34 | 8 | 16 | 18 | 12 | 28 | 12 | 4 | 3 | 9 | 19 | 13 | 26 | 255 |
| booking hold | other | 9 | 16 | 15 | 13 | 14 | 19 | 14 | 18 | 15 | 13 | 12 | 13 | 12 | 19 | 12 | 214 |
| deploy cutover | other | 10 | 15 | 14 | 16 | 9 | 13 | 13 | 9 | 12 | 16 | 14 | 12 | 14 | 10 | 8 | 185 |
| review assign | other | 8 | 20 | 24 | 12 | 13 | 13 | 10 | 17 | 11 | 10 | 10 | 9 | 6 | 16 | 11 | 190 |
| booking confirm | other | 7 | 12 | 14 | 15 | 11 | 14 | 10 | 14 | 12 | 18 | 11 | 15 | 8 | 14 | 10 | 185 |
| pairing show-code | other | 10 | 11 | 11 | 12 | 13 | 11 | 10 | 10 | 13 | 12 | 13 | 14 | 10 | 14 | 10 | 174 |
| containment block-token | other | 5 | 9 | 10 | 12 | 15 | 19 | 13 | 13 | 15 | 11 | 11 | 9 | 8 | 13 | 8 | 171 |
| approval request | other | 9 | 10 | 12 | 11 | 10 | 12 | 10 | 11 | 11 | 11 | 11 | 11 | 9 | 14 | 12 | 164 |
| review submit | other | 10 | 15 | 15 | 10 | 11 | 7 | 11 | 15 | 11 | 11 | 12 | 10 | 10 | 10 | 9 | 167 |
| approval grant | other | 6 | 12 | 12 | 14 | 8 | 10 | 7 | 12 | 13 | 14 | 11 | 13 | 9 | 9 | 10 | 160 |
| deploy request | other | 11 | 12 | 12 | 11 | 12 | 10 | 9 | 13 | 12 | 12 | 11 | 9 | 8 | 11 | 7 | 160 |
| pairing enter-code | other | 8 | 9 | 11 | 14 | 12 | 11 | 9 | 10 | 12 | 10 | 12 | 10 | 8 | 9 | 8 | 153 |
| tasks list | other | 24 | 10 | 72 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 33 | 0 | 0 | 144 |

Figure G9: **Complete command inventory across all evaluated models (Part 2: Rows 31–59 of 89).** Continued from Figure G8; continued in Figure G10.

| sub-command | category | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 | total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pairing start | other | 10 | 10 | 11 | 11 | 11 | 10 | 10 | 11 | 11 | 10 | 10 | 11 | 8 | 11 | 8 | 153 |
| lock acquire | other | 9 | 8 | 8 | 8 | 12 | 12 | 10 | 13 | 9 | 9 | 7 | 9 | 9 | 11 | 9 | 143 |
| containment isolate-device | other | 3 | 6 | 7 | 8 | 10 | 15 | 9 | 8 | 10 | 8 | 6 | 6 | 6 | 9 | 7 | 118 |
| pairing confirm | other | 5 | 8 | 8 | 12 | 9 | 7 | 9 | 8 | 9 | 12 | 6 | 7 | 5 | 5 | 6 | 116 |
| booking search | other | 6 | 5 | 4 | 19 | 4 | 3 | 6 | 4 | 5 | 7 | 6 | 4 | 4 | 9 | 24 | 110 |
| containment disable-forwarding | other | 1 | 7 | 6 | 7 | 9 | 13 | 10 | 7 | 9 | 7 | 7 | 5 | 5 | 9 | 4 | 106 |
| lock release | other | 3 | 5 | 6 | 8 | 6 | 8 | 6 | 10 | 8 | 5 | 5 | 8 | 4 | 9 | 6 | 97 |
| containment revoke-sessions | other | 1 | 5 | 5 | 6 | 6 | 10 | 7 | 5 | 7 | 5 | 4 | 5 | 3 | 6 | 4 | 79 |
| deploy approve | other | 6 | 7 | 8 | 4 | 5 | 4 | 4 | 6 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 73 |
| booking release | other | 2 | 7 | 4 | 2 | 6 | 4 | 4 | 6 | 3 | 7 | 1 | 8 | 0 | 4 | 2 | 60 |
| review sign | other | 2 | 4 | 6 | 3 | 4 | 3 | 2 | 5 | 3 | 4 | 4 | 4 | 4 | 2 | 2 | 52 |
| system status | other | 19 | 0 | 2 | 5 | 2 | 1 | 1 | 8 | 1 | 3 | 2 | 1 | 4 | 1 | 1 | 51 |
| approval escalate | other | 2 | 4 | 4 | 5 | 2 | 4 | 2 | 4 | 3 | 3 | 2 | 2 | 1 | 4 | 1 | 43 |
| tasks add | other | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 4 | 3 | 2 | 2 | 1 | 1 | 4 | 4 | 36 |
| deploy monitor | other | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 3 | 1 | 22 |
| deploy execute | other | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 15 |
| system devices | other | 9 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 14 |
| system config | other | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 0 | 0 | 5 |
| system health | other | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 5 |
| system setup | other | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| system cron | other | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| system doctor | other | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| system security | other | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 |
| undo 1 | other | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| approval provisional | other | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| tasks search | other | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| undo | other | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| undo 2 | other | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| undo 4 | other | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Figure G10: **Complete command inventory across all evaluated models (Part 3: Rows 60–89 of 89).** Final part of the 88-command execution inventory across all 15 evaluated models.

### G.4 GROUNDING × ACTION MAP

| Model | opens/ task | zero-open tasks | pass |
|---|---|---|---|
| Claude Sonnet 5 | **1.83** | **92** | 20.8% |
| Gemini 3.7 Flash | 2.25 | 60 | 20.3% |
| GPT-5.6 Terra | 2.65 | 1 | 20.0% |
| Claude Opus 4.8 | 2.71 | 1 | 22.8% |
| Gemini 3.6 Flash | 2.90 | 5 | 18.5% |
| Claude Opus 5 | 2.98 | 1 | 25.2% |
| Gemini 3.1 Pro | 3.15 | 14 | 12.0% |
| Gemini 3.5 Flash | 3.56 | 18 | **3.0%** |
| GPT-5.6 Luna | 3.90 | 0 | 19.8% |
| GPT-5.6 Sol | 4.08 | 0 | 24.5% |
| DeepSeek V4 Pro | 4.36 | 0 | 19.8% |
| DeepSeek V4 Flash | 6.00 | 0 | 8.5% |
| Kimi K3 | 6.72 | 3 | 21.8% |
| GLM-5.2 FP8 | 7.26 | 0 | 14.7% |
| GPT-6 Astra | **9.66** | 0 | **28.7%** |

Table G5: **Visual asset grounding frequency versus benchmark pass rate (High Effort).** Compares average image opens per task, zero-open task frequency, and overall pass rate across the 15 evaluated models.




Figure G11: **Visual asset grounding volume versus task pass rate (High Effort).** Plots mean asset opens per task against benchmark pass rate, illustrating the decoupling between visual inspection volume and reasoning accuracy.

### G.5 OUTCOME AND STOP-REASON MIX

| Model | none | missing_required_side_effect | incomplete_required_submission | reached max_steps | 3-identical loop | empty reply | provider_failure |
|---|---|---|---|---|---|---|---|
| Gemini 3.5 Flash | **18** | **26** | **19** | **535** | 1 | 1 | 0 |
| Gemini 3.6 Flash | 111 | 171 | 170 | 148 | 0 | 0 | 0 |
| Gemini 3.7 Flash | 122 | 128 | 106 | 244 | 0 | 0 | 0 |
| Gemini 3.1 Pro | 72 | 203 | 168 | 152 | 0 | 0 | **5** |
| Claude Opus 4.8 | 137 | 233 | 193 | 37 | 0 | 0 | 0 |
| Claude Opus 5 | 151 | 239 | 204 | **3** | 0 | 3 | 0 |
| Claude Sonnet 5 | 125 | 255 | 200 | 20 | 0 | 0 | 0 |
| GPT-5.6 Sol | 147 | 238 | 208 | 7 | 0 | 0 | 0 |
| GPT-5.6 Luna | 119 | 226 | 194 | 55 | 6 | 0 | 0 |
| GPT-5.6 Terra | 120 | **261** | **210** | 8 | 1 | 0 | 0 |
| GPT-6 Astra | **172** | 220 | 199 | 6 | 0 | 3 | 0 |
| DeepSeek V4 Pro | 119 | 218 | 182 | 81 | 0 | 0 | 0 |
| DeepSeek V4 Flash | 51 | 103 | 85 | 358 | 2 | 1 | 0 |
| Kimi K3 | 131 | 165 | 132 | 170 | 0 | 2 | 0 |
| GLM-5.2 FP8 | 88 | 156 | 139 | 207 | 6 | **4** | 0 |

Table G6: **Episode termination reasons across models (High Effort).** Reports how each model's 600 episodes ended: successful completion (`none`), missing required environment actions, incomplete deliverables, reaching the step limit, or execution loops.

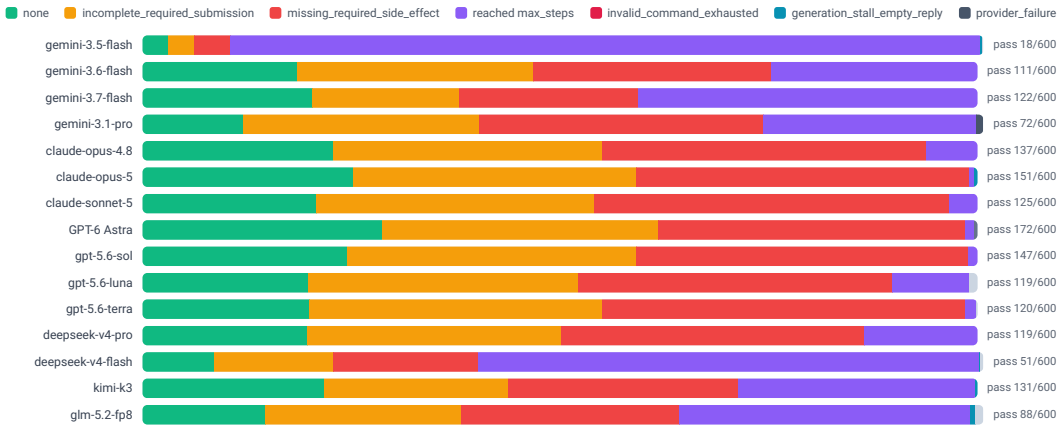



Figure G12: **Episode outcome and termination reason breakdown (High Effort).** Visualizes the proportion of episodes ending successfully versus failing from missing actions, incomplete deliverables, or reaching the step limit.

### G.6 Loops and termination

| Model | mean steps | p50 | p50 on success | p50 on failure | tail waste (mean) | step errors |
|---|---|---|---|---|---|---|
| Gemini 3.5 Flash | 19.7 | 20.0 | 18.0 | 20.0 | **6.6** | **0.3%** |
| Gemini 3.6 Flash | 16.3 | 16.5 | 15.0 | 17.0 | 1.6 | 0.7% |
| Gemini 3.7 Flash | 17.8 | 19.0 | 17.0 | 20.0 | 1.9 | 0.6% |
| Gemini 3.1 Pro | 13.2 | 12.0 | 10.0 | 13.0 | 3.0 | 1.1% |
| Claude Opus 4.8 | 12.2 | 12.0 | 11.0 | 12.0 | 1.6 | 0.7% |
| Claude Opus 5 | 11.1 | 11.0 | 10.0 | 11.0 | 1.5 | 0.9% |
| Claude Sonnet 5 | **10.6** | **10.0** | 10.0 | **10.0** | **1.3** | 0.9% |
| GPT-5.6 Sol | 13.7 | 14.0 | 12.0 | 15.0 | 1.6 | 1.0% |
| GPT-5.6 Luna | 13.5 | 13.0 | 11.0 | 14.0 | 2.5 | 1.2% |
| GPT-5.6 Terra | 11.4 | 11.0 | 10.0 | 11.0 | 1.8 | 1.0% |
| GPT-6 Astra | **26.1** | **27.0** | **26.0** | **27.0** | 2.3 | **1.6%** |
| DeepSeek V4 Pro | 13.4 | 13.0 | 12.0 | 13.0 | 1.6 | 1.2% |
| DeepSeek V4 Flash | 17.8 | 20.0 | 15.0 | 20.0 | 4.9 | 0.5% |
| Kimi K3 | 15.4 | 16.0 | 14.0 | 17.0 | 2.8 | 0.4% |
| GLM-5.2 FP8 | 14.9 | 15.0 | 13.0 | 16.0 | 3.3 | 0.7% |

Table G7: **Step execution dynamics and tail error rates (High Effort).** Breaks down step counts by outcome (successful vs. failed rounds), wasted steps taken after the last required environment action, and tool execution error rates.

### G.7 Failure clustering: the shape of a model's losses

#### G.7.1 Concentration of losses

| Model | per-check pass | observed pass | if independent | excess | lift | 95% CI | overdispersion |
|---|---|---|---|---|---|---|---|
| GPT-6 Astra | **66.7%** | **28.7%** | **13.8%** | +14.9pp | 1.47 | [1.38, 1.57] | 1.91 |
| Claude Opus 5 | 63.0% | 25.2% | 10.8% | +14.4pp | **1.49** | [1.39, 1.59] | **2.23** |
| GPT-5.6 Sol | 62.2% | 24.5% | 10.2% | +14.3pp | 1.42 | [1.33, 1.51] | 2.05 |
| Claude Opus 4.8 | 60.9% | 22.8% | 9.3% | +13.5pp | 1.41 | [1.33, 1.50] | 2.08 |
| Kimi K3 | 55.8% | 21.8% | 6.5% | **+15.3pp** | 1.40 | [1.33, 1.47] | 2.11 |
| Claude Sonnet 5 | 58.2% | 20.8% | 7.7% | +13.1pp | 1.39 | [1.32, 1.47] | 2.13 |
| GPT-5.6 Terra | 59.3% | 20.0% | 8.3% | +11.7pp | 1.37 | [1.30, 1.45] | 2.01 |
| GPT-5.6 Luna | 59.8% | 19.8% | 8.6% | +11.2pp | 1.36 | [1.28, 1.44] | 1.93 |
| DeepSeek V4 Pro | 57.7% | 19.8% | 7.4% | +12.4pp | 1.35 | [1.28, 1.41] | 1.95 |
| Gemini 3.7 Flash | 57.5% | 20.3% | 7.3% | +13.0pp | 1.30 | [1.24, 1.36] | 1.83 |
| Gemini 3.6 Flash | 58.0% | 18.5% | 7.6% | +10.9pp | 1.28 | [1.23, 1.34] | 1.79 |
| GLM-5.2 FP8 | 48.2% | 14.7% | 3.6% | +11.0pp | 1.27 | [1.22, 1.32] | 1.97 |
| Gemini 3.1 Pro | 48.8% | 12.1% | 3.8% | +8.3pp | 1.19 | [1.15, 1.24] | 1.66 |
| DeepSeek V4 Flash | 38.8% | 8.5% | 1.6% | +6.9pp | 1.16 | [1.12, 1.19] | 1.56 |
| Gemini 3.5 Flash | **28.1%** | **3.0%** | **0.5%** | **+2.5pp** | **1.05** | [1.03, 1.08] | **1.13** |

Table G8: **Failure clustering across models (High Effort).** Reports per-check pass rates, observed task pass rates, expected pass rates if check failures were independent, excess pass rate, empirical co-failure lift (with 95% bootstrap CIs), and error clustering overdispersion.

| decomposition | per-check component | clustering component | total |
|---|---|---|---|
| pooled | +0.58 | −0.07 | 0.50 |
| stratified | −2.33 | +2.83 | 0.50 |

Table G9: **Performance difference decomposition between top models (Claude Opus 5 vs. GPT-5.6 Sol).** Decomposes the score difference into individual per-check accuracy and joint failure clustering components ($n = 598$, 4,000 bootstrap draws).

more tasks here than independence predicts · fewer than independence predicts · each cell: **signed gap in points** · bar length · observed vs independent

| model | lost 0<br>task passed | lost 1 | lost 2 | lost 3 | lost 4+ |
|---|---|---|---|---|---|
| gemini-3.5-flash | +2.5<br>3.0 vs 0.5 | -0.5<br>4.0 vs 4.5 | -3.9<br>10.8 vs 14.8 | +0.0<br>24.0 vs 24.0 | +1.9<br>58.2 vs 56.3 |
| gemini-3.6-flash | +10.9<br>18.5 vs 7.6 | -0.9<br>22.0 vs 22.9 | -11.0<br>18.3 vs 29.4 | -5.2<br>16.8 vs 22.0 | +6.2<br>24.3 vs 18.2 |
| gemini-3.7-flash | +13.0<br>20.3 vs 7.3 | -4.0<br>18.5 vs 22.5 | -9.3<br>20.0 vs 29.3 | -7.2<br>15.0 vs 22.2 | +7.5<br>26.2 vs 18.7 |
| gemini-3.1-pro | +8.3<br>12.1 vs 3.8 | -1.6<br>13.9 vs 15.6 | -3.9<br>22.5 vs 26.4 | -6.6<br>18.7 vs 25.2 | +3.8<br>32.8 vs 29.0 |
| claude-opus-4.8 | +13.5<br>22.8 vs 9.3 | -2.9<br>22.5 vs 25.4 | -11.0<br>18.7 vs 29.7 | -6.8<br>13.7 vs 20.5 | +7.3<br>22.3 vs 15.1 |
| claude-opus-5 | +14.4<br>25.2 vs 10.8 | -3.9<br>23.3 vs 27.2 | -11.5<br>18.2 vs 29.7 | -5.6<br>13.7 vs 19.3 | +6.6<br>19.7 vs 13.0 |
| claude-sonnet-5 | +13.1<br>20.8 vs 7.7 | -1.6<br>21.5 vs 23.1 | -9.6<br>19.8 vs 29.4 | -8.7<br>13.2 vs 21.9 | +6.8<br>24.7 vs 17.9 |
| GPT-6 Astra | +14.9<br>28.7 vs 13.8 | -8.5<br>21.8 vs 30.4 | -10.0<br>19.2 vs 29.2 | -3.5<br>13.3 vs 16.8 | +7.2<br>17.0 vs 9.8 |
| gpt-5.6-sol | +14.3<br>24.5 vs 10.2 | -6.0<br>20.5 vs 26.5 | -9.4<br>20.3 vs 29.7 | -5.1<br>14.7 vs 19.7 | +6.2<br>20.0 vs 13.8 |
| gpt-5.6-luna | +11.2<br>19.8 vs 8.6 | -1.6<br>22.8 vs 24.4 | -9.1<br>20.5 vs 29.6 | -6.8<br>14.3 vs 21.1 | +6.3<br>22.5 vs 16.2 |
| gpt-5.6-terra | +11.7<br>20.0 vs 8.3 | -1.0<br>23.0 vs 24.0 | -9.7<br>19.8 vs 29.6 | -7.4<br>14.0 vs 21.4 | +6.4<br>23.2 vs 16.8 |
| deepseek-v4-pro | +12.4<br>19.8 vs 7.4 | -3.1<br>19.5 vs 22.6 | -8.8<br>20.5 vs 29.3 | -5.5<br>16.7 vs 22.2 | +5.0<br>23.5 vs 18.5 |
| deepseek-v4-flash | +6.9<br>8.5 vs 1.6 | +1.8<br>11.0 vs 9.2 | -8.1<br>13.0 vs 21.1 | -3.4<br>22.5 vs 25.9 | +2.8<br>45.0 vs 42.2 |
| kimi-k3 | +15.3<br>21.8 vs 6.5 | -4.2<br>16.8 vs 21.0 | -11.2<br>17.7 vs 28.9 | -7.5<br>15.5 vs 23.0 | +7.6<br>28.2 vs 20.6 |
| glm-5.2-fp8 | +11.0<br>14.7 vs 3.6 | +0.7<br>15.8 vs 15.1 | -10.0<br>16.2 vs 26.1 | -7.2<br>18.2 vs 25.3 | +5.4<br>35.2 vs 29.8 |

Figure G13: **Check failure distribution compared to an independent error baseline (High Effort).** Plots the difference between observed check failure frequencies and independent binomial predictions, showing error clustering across models.

**Error Clustering Patterns.** Twelve of the 15 models show systematic error clustering: failures are pulled away from intermediate outcomes (failing 1–3 checks) and concentrated at the extremes (passing all checks or failing $\geq 4$ checks). All 15 models show this pattern, with differences across models driven mainly by strength rather than direction.

| model | all 595 descriptive | easy tercile | middle tercile | hard tercile | vs other 13 the one to quote |
|---|---|---|---|---|---|
| claude-opus-5 | 1.50 | 2.10 ▲ [1.53, 2.70] | 1.37 ▲ [1.14, 1.55] | 1.04 [1.00, 1.09] | +0.188 ▲ [+0.110, +0.268] |
| GPT-6 Astra | 1.47 | 1.42 [1.12, 1.85] | 0.96 [0.88, 1.04] | 1.01 [0.98, 1.04] | +0.149 ▲ [+0.070, +0.229] |
| gpt-5.6-sol | 1.43 | 1.09 [0.79, 1.40] | 1.00 [0.92, 1.08] | 1.03 [0.99, 1.07] | +0.116 ▲ [+0.059, +0.176] |
| claude-opus-4.8 | 1.42 | 1.95 ▲ [1.16, 2.66] | 1.04 [0.95, 1.13] | 1.02 [0.98, 1.05] | +0.102 ▲ [+0.054, +0.153] |
| kimi-k3 | 1.40 | 2.26 ▲ [1.95, 2.64] | 1.17 ▲ [1.08, 1.26] | 1.06 ▲ [1.02, 1.10] | +0.087 ▲ [+0.037, +0.141] |
| claude-sonnet-5 | 1.40 | 1.71 ▲ [1.19, 2.19] | 1.16 ▲ [1.05, 1.26] | 1.04 ▲ [1.00, 1.07] | +0.081 ▲ [+0.034, +0.131] |
| gpt-5.6-terra | 1.38 | 1.15 [0.81, 1.48] | 0.99 [0.91, 1.06] | 1.02 [0.99, 1.05] | +0.062 ▲ [+0.019, +0.106] |
| gpt-5.6-luna | 1.36 | 1.53 ▲ [1.10, 1.98] | 0.96 [0.89, 1.03] | 1.02 [0.99, 1.05] | +0.047 ▲ [+0.002, +0.094] |
| deepseek-v4-pro | 1.35 | 1.29 [1.00, 1.58] | 1.09 ▲ [1.01, 1.17] | 1.02 [0.99, 1.04] | +0.031 [-0.013, +0.074] |
| gemini-3.7-flash | 1.30 | 1.53 ▲ [1.29, 1.80] | 1.11 ▲ [1.04, 1.18] | 1.00 [0.97, 1.02] | -0.024 [-0.081, +0.030] |
| gemini-3.6-flash | 1.29 | 1.34 ▲ [1.08, 1.60] | 1.06 [0.97, 1.15] | 0.99 [0.97, 1.01] | -0.036 [-0.092, +0.017] |
| glm-5.2-fp8 | 1.27 | 1.81 ▲ [1.58, 2.07] | 1.16 ▲ [1.08, 1.23] | 1.00 [0.98, 1.02] | -0.053 ▼ [-0.099, -0.008] |
| gemini-3.1-pro | 1.19 | 1.38 ▲ [1.16, 1.61] | 0.99 [0.95, 1.04] | 1.01 [0.99, 1.04] | -0.136 ▼ [-0.175, -0.097] |
| deepseek-v4-flash | 1.16 | 1.49 ▲ [1.36, 1.63] | 1.06 ▲ [1.02, 1.09] | 1.00 [0.98, 1.02] | -0.177 ▼ [-0.222, -0.133] |
| gemini-3.5-flash | 1.05 | 1.08 ▲ [1.03, 1.15] | 1.01 [0.99, 1.04] | 0.99 ▼ [0.98, 1.00] | -0.288 ▼ [-0.338, -0.242] |

Figure G14: **Co-failure lift across task difficulty levels (High Effort).** Displays empirical co-failure lift recomputed within leave-one-out peer difficulty groups (Easy, Medium, Hard). The final column isolates model-specific clustering after controlling for shared task difficulty across the 595 universally graded rounds (4,000 bootstrap draws).

**Error Clustering Across Difficulty Levels.** After controlling for baseline task difficulty through peer grouping, co-failure lift remains elevated on easy and intermediate tasks for top models (GPT-6 Astra, Claude Opus 5, GPT-5.6 Sol, Kimi K3). This indicates that agents fail due to all-or-nothing misunderstandings rather than isolated slips on single checks.

### G.7.2 CHECK-TYPE SPLIT

| Model | `field_semantic_match` | `field_contains` | `other` |
|---|---|---|---|
| GPT-6 Astra | **65.7** | **94.7** | **43.5** |
| Kimi K3 | **65.7** | 84.9 | 42.6 |
| Claude Opus 5 | 62.7 | 87.9 | 38.2 |
| Gemini 3.7 Flash | 62.2 | 80.7 | 40.4 |
| GPT-5.6 Sol | 62.1 | 87.6 | 38.0 |
| Gemini 3.6 Flash | 61.6 | 83.4 | 38.5 |
| Claude Opus 4.8 | 61.1 | 83.2 | 39.3 |
| GPT-5.6 Luna | 60.9 | 83.8 | 34.7 |
| GPT-5.6 Terra | 60.4 | 79.6 | 33.8 |
| DeepSeek V4 Pro | 60.0 | 84.1 | 33.2 |
| Claude Sonnet 5 | 59.1 | 73.6 | 36.1 |
| GLM-5.2 FP8 | 58.9 | 83.3 | 28.3 |
| DeepSeek V4 Flash | 53.9 | 79.1 | 27.7 |
| Gemini 3.1 Pro | 46.4 | 79.9 | 36.2 |
| Gemini 3.5 Flash | **45.6** | **70.8** | **26.8** |

Table G10: **Check-type pass rate breakdown across models (High Effort).** Reports pass rates across LLM-judged semantic matches (`field_semantic_match`), deterministic literal substrings (`field_contains`), and auxiliary state assertions (`other`).

### G.7.3 FIELD_CONTAINS, DECOMPOSED

| Model | asset gate | literal content | pooled | content − asset |
|---|---|---|---|---|
| GPT-6 Astra | **98.8** | 81.6 | **94.7** | −17.2 |
| Claude Opus 5 | 89.7 | **82.4** | 87.9 | −7.3 |
| GPT-5.6 Sol | 91.2 | 75.8 | 87.6 | −15.4 |
| Kimi K3 | 92.5 | 60.5 | 84.9 | −32.0 |
| DeepSeek V4 Pro | 88.9 | 68.8 | 84.1 | −20.1 |
| GPT-5.6 Luna | 86.3 | 75.8 | 83.8 | −10.5 |
| Gemini 3.6 Flash | 87.8 | 69.1 | 83.4 | −18.7 |
| GLM-5.2 FP8 | 93.2 | 51.6 | 83.3 | −41.6 |
| Claude Opus 4.8 | 84.8 | 78.1 | 83.2 | −6.7 |
| Gemini 3.7 Flash | 84.9 | 67.2 | 80.7 | −17.7 |
| Gemini 3.1 Pro | 84.4 | 65.4 | 79.9 | −19.0 |
| GPT-5.6 Terra | 80.9 | 75.4 | 79.6 | −5.5 |
| DeepSeek V4 Flash | 91.8 | 38.3 | 79.1 | −53.5 |
| Claude Sonnet 5 | **72.6** | 76.6 | 73.6 | **+4.0** |
| Gemini 3.5 Flash | 87.3 | **17.6** | **70.8** | **−69.7** |

Table G11: **Decomposition of `field_contains` into visual gating and literal transcription (High Effort).** Separates substring evaluation into the visual asset access gate (`asset gate`) and faithful deliverable transcription (`literal content`). Both components are verified via deterministic checkers. Rates are over all roles, including unscored diagnostic checks (which sit entirely in the asset gate); Table C2 reports the asset gate over scored checks only and therefore prints higher rates.

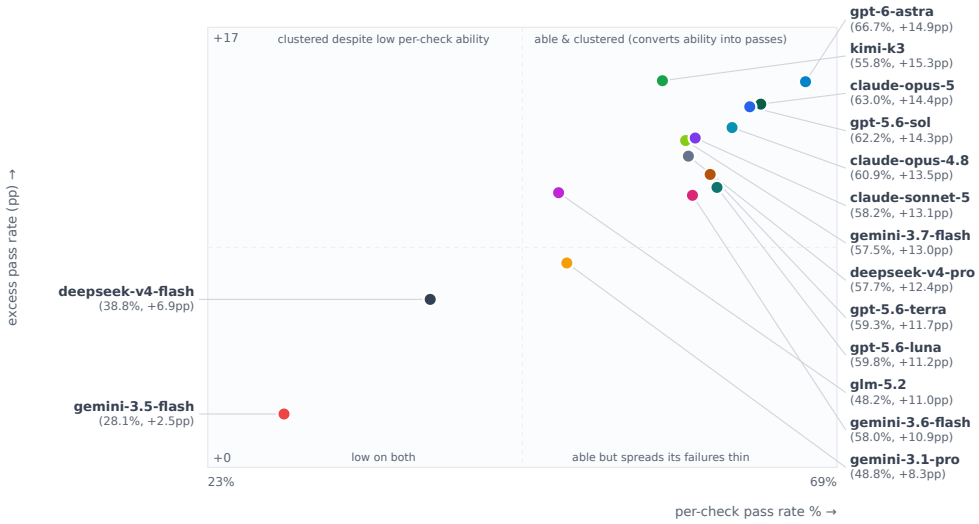



Figure G15: **Per-check accuracy versus failure clustering lift across models (High Effort).** Compares average check accuracy against task-level error clustering, showing distinct failure patterns across models with similar average check pass rates.

### G.8 DIFFICULTY LANDSCAPE

Table G12: **Inventory and suitability of candidate task difficulty axes.** Summarizes potential difficulty dimensions across the 600 task specifications, evaluating their variety of values and usefulness for analysis.

| field | cardinality | usable? |
|---|---|---|
| `difficulty` | 1 (every task is `3`) | no |
| `domains` | 1 (`composite`, `conflict`) | no |
| `template_id` | 1 | no |
| `weight` | 1 (`1.0`) | no |
| `hidden_constraints` / `visible_constraints` | 0 on every task | no |
| `decision_requirements` | 1 on every task | no |
| `availability_tier` / `online_requirement` | null on every task | no |
| `realism_tags` | 5, but = `conflict_vlm_v1` + `round_type` | redundant |
| `conflict_taxonomy` | **absent from v8 entirely** | no |
| `round_type` | 5 (one with n=1) | **yes** |
| `required_modalities` | 2 | yes, with caveats |
| `allowed_tool_families` | 10 | yes, confounded with scenario |
| **scored-unit count** | 2–14 per task | **yes — the best axis** |

#### G.8.1 SCORED-UNIT COUNT IS THE ONE CLEAN DIFFICULTY AXIS

| Model | 2–4 pass | 5–6 pass | 7+ pass | 2–4 score | 5–6 score | 7+ score |
|---|---|---|---|---|---|---|
| Gemini 3.5 Flash | **3.1** | **3.5** | **2.0** | **0.160** | **0.303** | **0.311** |
| Gemini 3.6 Flash | 29.7 | 17.1 | 6.7 | 0.585 | 0.614 | 0.538 |
| Gemini 3.7 Flash | 31.8 | 19.4 | 7.3 | 0.572 | 0.619 | 0.528 |
| Gemini 3.1 Pro | 20.0 | 10.1 | 5.4 | 0.469 | 0.523 | 0.464 |
| Claude Opus 4.8 | 33.3 | 21.7 | 11.3 | 0.638 | 0.652 | 0.543 |
| Claude Opus 5 | 34.4 | 24.0 | 15.3 | 0.655 | 0.670 | 0.576 |
| Claude Sonnet 5 | 32.8 | 18.2 | 10.0 | 0.621 | 0.623 | 0.518 |
| GPT-5.6 Sol | 35.9 | 21.3 | 15.3 | 0.654 | 0.645 | 0.584 |
| GPT-5.6 Luna | 28.6 | 19.8 | 8.7 | 0.606 | 0.627 | 0.561 |
| GPT-5.6 Terra | 28.6 | 19.8 | 9.3 | 0.618 | 0.619 | 0.553 |
| GPT-6 Astra | **39.6** | **24.8** | **21.3** | **0.661** | **0.685** | **0.647** |
| DeepSeek V4 Pro | 27.6 | 18.2 | 12.7 | 0.571 | 0.608 | 0.548 |
| DeepSeek V4 Flash | 9.9 | 9.3 | 5.3 | 0.279 | 0.444 | 0.380 |
| Kimi K3 | 29.2 | 21.7 | 12.7 | 0.536 | 0.594 | 0.529 |
| GLM-5.2 FP8 | 22.4 | 12.8 | 8.0 | 0.468 | 0.518 | 0.457 |
| **pooled** | 27.1 | 17.5 | 10.1 | 0.540 | 0.583 | 0.515 |

Table G13: **Benchmark pass rate and check score grouped by scored-unit count (High Effort).** Compares performance across low (2–4 units, $n = 192$), medium (5–6 units, $n = 258$), and high (7+ units, $n = 150$) check count groups (3,284 total evaluated units).

### G.8.2 TASK DIFFICULTY VARIES BY ROUND TYPE INDEPENDENT OF CHECK COUNT

| round_type | n | mean units | 2–4 band | 5–6 band | 7+ band | marginal |
|---|---|---|---|---|---|---|
| investigation | 274 | 4.91 | 27.5 [119] | 13.4 [112] | **2.3** [43] | 17.8 |
| verification | 174 | 5.83 | **28.6** [39] | 18.1 [80] | 11.2 [55] | 18.3 |
| action | 86 | 6.55 | 21.0 [17] | **26.7** [31] | 13.7 [38] | **19.9** |
| decision | 65 | 5.49 | 17.4 [16] | 16.7 [35] | 11.2 [14] | 15.7 |

Table G14: **Pass rate distribution across round operational types.** Compares pass rates across investigation, verification, action, and decision task types, broken down by rubric unit count.

### G.8.3 NATURAL PHOTOGRAPHS DO NOT INCREASE TASK DIFFICULTY

| band | n (no-img) | n (img) | no-img pass | img pass | diff |
|---|---|---|---|---|---|
| 2–4 | 120 | 72 | 24.8 | 28.6 | +3.8 |
| 5–6 | 223 | 35 | 17.1 | 16.0 | −1.1 |
| 7+ | 137 | 13 | 9.8 | 4.4 | **−5.4** |

Table G15: **Stratified comparison of natural-image versus chart-only tasks.** Reports pass rates for tasks requiring natural photographic grounding versus structured charts across rubric unit bands ($n = 600$).

### G.8.4 TOOL FAMILY (CONFOUNDED, REPORTED FOR COMPLETENESS)

| family | n | pooled pass |
|---|---|---|
| lock + review | 30 | **32.2%** |
| shell_scripting | 150 | 22.4% |
| approval | 90 | 20.0% |
| deploy | 90 | 19.5% |
| *(none)* | 60 | 19.0% |
| pairing | 30 | 16.2% |
| lock | 30 | 10.5% |
| containment | 30 | 10.0% |
| booking | 60 | 9.3% |
| tasks | 30 | **4.1%** |

Table G16: **Pooled pass rate across domain tool families.** Summarizes aggregate pass rates across specialized domain CLI interfaces beyond the core toolset.

### G.8.5 GOLD-TRAJECTORY LENGTH

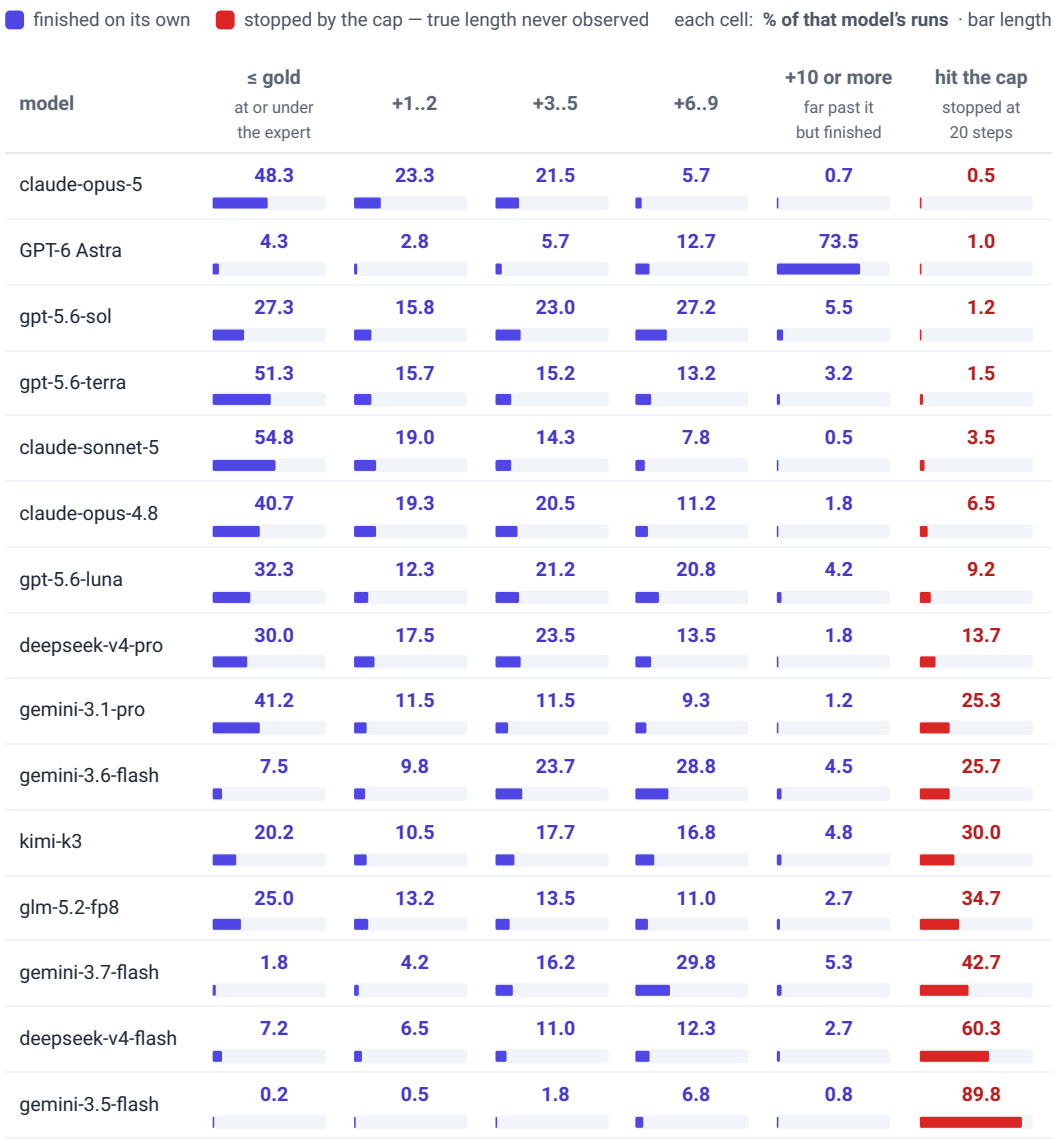
| Model | short | mid | long | all |
|---|---|---|---|---|
| Gemini 3.5 Flash | **3.5** | **3.2** | **2.1** | **3.0** |
| Gemini 3.6 Flash | 26.5 | 21.9 | 3.6 | 18.5 |
| Gemini 3.7 Flash | 30.1 | 22.5 | 7.1 | 20.3 |
| Gemini 3.1 Pro | 18.6 | 12.4 | 5.7 | 12.0 |
| Claude Opus 4.8 | 32.7 | 25.1 | 9.3 | 22.8 |
| Claude Opus 5 | 36.3 | 26.2 | 13.6 | 25.2 |
| Claude Sonnet 5 | 32.7 | 22.2 | 7.9 | 20.8 |
| GPT-5.6 Sol | **37.2** | 24.8 | 13.6 | 24.5 |
| GPT-5.6 Luna | 23.0 | 23.1 | 9.3 | 19.8 |
| GPT-5.6 Terra | 28.3 | 22.2 | 7.9 | 20.0 |
| GPT-6 Astra | 35.2 | **28.1** | **18.6** | **28.7** |
| DeepSeek V4 Pro | 24.8 | 21.6 | 11.4 | 19.8 |
| DeepSeek V4 Flash | 9.7 | 9.8 | 4.3 | 8.5 |
| Kimi K3 | 30.1 | 23.1 | 12.1 | 21.8 |
| GLM-5.2 FP8 | 19.5 | 16.7 | 5.7 | 14.7 |

Table G17: **Pass rate by reference trajectory length (High Effort).** Divides tasks into short, medium, and long reference trajectory groups ($n = 113/347/140$, maximum reference length 18 steps).

| gold band<br>tasks · gold median · range | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **short (≤8)**<br>113 tasks · gold 8 (7–8) | **4%**<br>*14 st · 1.75× n1*<br>**96% capped** | **26%**<br>15 st · 1.88×<br>28% capped | **30%**<br>17 st · 2.12×<br>**45% capped** | **19%**<br>12 st · 1.50×<br>31% capped | **33%**<br>10 st · 1.25×<br>7% capped | **36%**<br>9 st · 1.14×<br>2% capped | **33%**<br>9 st · 1.25×<br>5% capped | **42%**<br>25 st · 3.12×<br>2% capped | **37%**<br>14 st · 1.75×<br>3% capped | **23%**<br>12.5 st · 1.62×<br>16% capped | **28%**<br>10.5 st · 1.33×<br>4% capped | **25%**<br>11 st · 1.46×<br>23% capped | **10%**<br>*16.5 st · 2.06× n10*<br>**67% capped** | **30%**<br>14 st · 1.86×<br>32% capped | **20%**<br>13.5 st · 1.80×<br>**34% capped** |
| **mid (9–11)**<br>347 tasks · gold 10 (9–11) | **3%**<br>*17 st · 1.70× n11*<br>**88% capped** | **22%**<br>15 st · 1.50×<br>21% capped | **22%**<br>16 st · 1.67×<br>**37% capped** | **12%**<br>9 st · 0.90×<br>24% capped | **25%**<br>10.5 st · 1.10×<br>4% capped | **26%**<br>10 st · 1.00×<br>0% capped | **22%**<br>10 st · 0.91×<br>2% capped | **29%**<br>26 st · 2.60×<br>1% capped | **25%**<br>11.5 st · 1.19×<br>1% capped | **23%**<br>10.5 st · 1.10×<br>7% capped | **22%**<br>9 st · 0.89×<br>1% capped | **22%**<br>12 st · 1.20×<br>10% capped | **10%**<br>15 st · 1.50×<br>**60% capped** | **23%**<br>13 st · 1.33×<br>28% capped | **17%**<br>12 st · 1.20×<br>32% capped |
| **long (≥12)**<br>140 tasks · gold 13 (12–18) | **2%**<br>*16.5 st · 1.31× n2*<br>**89% capped** | **4%**<br>*16 st · 1.33× n5*<br>**34% capped** | **7%**<br>*16 st · 1.25× n9*<br>**55% capped** | **6%**<br>*15 st · 1.08× n8*<br>24% capped | **9%**<br>*13.5 st · 1.11× n12*<br>13% capped | **14%**<br>*12 st · 0.92× n19*<br>0% capped | **8%**<br>*12 st · 0.92× n10*<br>6% capped | **19%**<br>29 st · 2.23×<br>1% capped | **14%**<br>*13 st · 1.00× n19*<br>1% capped | **9%**<br>*12 st · 0.85× n13*<br>9% capped | **8%**<br>*11 st · 0.85× n11*<br>1% capped | **11%**<br>*13 st · 1.00× n15*<br>14% capped | **4%**<br>*13 st · 1.00× n5*<br>**56% capped** | **12%**<br>*13 st · 1.03× n16*<br>33% capped | **6%**<br>*15 st · 1.23× n8*<br>**42% capped** |
| **all tasks (overall)**<br>600 tasks · gold 10 (7–18) | **3%**<br>*17 st · 1.70× n14*<br>**90% capped** | **18%**<br>15 st · 1.60×<br>26% capped | **20%**<br>16 st · 1.73×<br>**43% capped** | **12%**<br>10 st · 1.00×<br>25% capped | **23%**<br>11 st · 1.11×<br>6% capped | **25%**<br>10 st · 1.00×<br>0% capped | **21%**<br>10 st · 1.00×<br>4% capped | **29%**<br>27 st · 2.70×<br>1% capped | **24%**<br>12 st · 1.25×<br>1% capped | **20%**<br>11 st · 1.10×<br>9% capped | **20%**<br>10 st · 1.00×<br>2% capped | **20%**<br>12 st · 1.24×<br>14% capped | **8%**<br>15 st · 1.55×<br>**60% capped** | **22%**<br>13 st · 1.40×<br>30% capped | **15%**<br>13 st · 1.25×<br>**35% capped** |

Figure G16: **Agent step expenditure relative to reference gold trajectories (High Effort).** Reports task pass rate, median steps, step overshoot, and budget cap exhaustion rate across reference gold trajectory length strata.

**Step Usage Relative to Reference Solutions.** Comparing steps taken against expert reference solutions shows that successful episodes complete tasks close to the reference length, whereas failed runs frequently run out of steps at the 20-step limit. On longer tasks, models run out of steps much more often.




| model | ≤ gold<br>at or under the expert | +1..2 | +3..5 | +6..9 | +10 or more<br>far past it but finished | hit the cap<br>stopped at 20 steps |
|---|---|---|---|---|---|---|
| claude-opus-5 | 48.3 | 23.3 | 21.5 | 5.7 | 0.7 | 0.5 |
| GPT-6 Astra | 4.3 | 2.8 | 5.7 | 12.7 | 73.5 | 1.0 |
| gpt-5.6-sol | 27.3 | 15.8 | 23.0 | 27.2 | 5.5 | 1.2 |
| gpt-5.6-terra | 51.3 | 15.7 | 15.2 | 13.2 | 3.2 | 1.5 |
| claude-sonnet-5 | 54.8 | 19.0 | 14.3 | 7.8 | 0.5 | 3.5 |
| claude-opus-4.8 | 40.7 | 19.3 | 20.5 | 11.2 | 1.8 | 6.5 |
| gpt-5.6-luna | 32.3 | 12.3 | 21.2 | 20.8 | 4.2 | 9.2 |
| deepseek-v4-pro | 30.0 | 17.5 | 23.5 | 13.5 | 1.8 | 13.7 |
| gemini-3.1-pro | 41.2 | 11.5 | 11.5 | 9.3 | 1.2 | 25.3 |
| gemini-3.6-flash | 7.5 | 9.8 | 23.7 | 28.8 | 4.5 | 25.7 |
| kimi-k3 | 20.2 | 10.5 | 17.7 | 16.8 | 4.8 | 30.0 |
| glm-5.2-fp8 | 25.0 | 13.2 | 13.5 | 11.0 | 2.7 | 34.7 |
| gemini-3.7-flash | 1.8 | 4.2 | 16.2 | 29.8 | 5.3 | 42.7 |
| deepseek-v4-flash | 7.2 | 6.5 | 11.0 | 12.3 | 2.7 | 60.3 |
| gemini-3.5-flash | 0.2 | 0.5 | 1.8 | 6.8 | 0.8 | 89.8 |

Figure G17: **Distribution of step count overshoot beyond reference gold length.** Plots the empirical distribution of additional steps expended relative to each task's reference gold trajectory length ($n = 600$).

**Step Count Overshoot Distribution.** Successful runs closely match the reference length with near-zero extra steps, whereas search loops and repeated polling cause models to hit the step limit.

**Designed Difficulty Levers and Capability Shifts.** Three designed environment levers (`async_required`, `coordinated_containment`, and `credential_lock`) reduce pass rates across all 15 evaluated models (with `gap_completion` reducing pass rates across 14 of 15 models). Tasks requiring asynchronous polling and active security containment cause the largest performance drops across all models.

| designed lever | tasks | seeds | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 | below baseline |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| all tasks (baseline) | 600 | 20 | 3.0% | 18.5% | 20.3% | 12.0% | 22.8% | 25.2% | 20.8% | 28.7% | 24.5% | 19.8% | 20.0% | 19.8% | 8.5% | 21.8% | 14.7% | – |
| async_required | 30 | 1 | 0.0% -3.0 pp | 3.3% -15.2 pp | 3.3% -17.0 pp | 3.3% -8.7 pp | 13.3% -9.5 pp | 13.3% -11.8 pp | 3.3% -17.5 pp | 6.7% -22.0 pp | 6.7% -17.8 pp | 3.3% -16.5 pp | 3.3% -16.7 pp | 0.0% -19.8 pp | 0.0% -8.5 pp | 3.3% -18.5 pp | 0.0% -14.7 pp | 15/15 |
| no_async | 570 | 19 | 3.2% +0.2 pp | 19.3% +0.8 pp | 21.2% +0.9 pp | 12.5% +0.5 pp | 23.3% +0.5 pp | 25.8% +0.6 pp | 21.8% +0.9 pp | 29.8% +1.1 pp | 25.4% +0.9 pp | 20.7% +0.9 pp | 20.9% +0.9 pp | 20.9% +1.0 pp | 8.9% +0.4 pp | 22.8% +1.0 pp | 15.4% +0.8 pp | 0/15 |
| vision_graded | 420 | 14 | 3.3% +0.3 pp | 16.7% -1.8 pp | 17.6% -2.7 pp | 10.0% -2.0 pp | 21.2% -1.6 pp | 23.6% -1.6 pp | 17.9% -3.0 pp | 25.7% -3.0 pp | 21.0% -3.5 pp | 18.8% -1.0 pp | 17.6% -2.4 pp | 18.6% -1.3 pp | 8.1% -0.4 pp | 21.2% -0.6 pp | 14.0% -0.6 pp | 14/15 |
| not_vision_graded | 180 | 6 | 2.2% -0.8 pp | 22.8% +4.3 pp | 26.7% +6.3 pp | 16.7% +4.7 pp | 26.7% +3.8 pp | 28.9% +3.7 pp | 27.8% +6.9 pp | 35.6% +6.9 pp | 32.8% +8.3 pp | 22.2% +2.4 pp | 25.6% +5.6 pp | 22.8% +2.9 pp | 9.4% +0.9 pp | 23.3% +1.5 pp | 16.1% +1.4 pp | 1/15 |
| evidence_mismatch | 240 | 8 | 1.7% -1.3 pp | 20.8% +2.3 pp | 24.6% +4.2 pp | 14.2% +2.2 pp | 23.8% +0.9 pp | 27.1% +1.9 pp | 23.8% +2.9 pp | 31.2% +2.6 pp | 30.8% +6.3 pp | 21.2% +1.4 pp | 22.5% +2.5 pp | 22.1% +2.2 pp | 8.3% -0.2 pp | 24.6% +2.8 pp | 15.0% +0.3 pp | 2/15 |
| temporal_staleness | 210 | 7 | 1.9% -1.1 pp | 19.0% +0.5 pp | 23.3% +3.0 pp | 14.3% +2.3 pp | 23.8% +1.0 pp | 24.8% -0.4 pp | 23.8% +3.0 pp | 32.4% +3.7 pp | 29.0% +4.5 pp | 20.0% +0.2 pp | 22.4% +2.4 pp | 21.0% +1.1 pp | 7.6% -0.9 pp | 22.9% +1.0 pp | 14.3% -0.4 pp | 4/15 |
| dependency_ordering | 180 | 6 | 2.8% -0.2 pp | 21.7% +3.2 pp | 24.4% +4.1 pp | 15.6% +3.6 pp | 27.8% +4.9 pp | 25.0% -0.2 pp | 23.3% +2.5 pp | 33.9% +5.2 pp | 31.7% +7.2 pp | 23.9% +4.1 pp | 23.3% +3.3 pp | 24.4% +4.6 pp | 11.7% +3.2 pp | 21.7% -0.2 pp | 16.7% +2.0 pp | 3/15 |
| time_window_expiration | 180 | 6 | 1.7% -1.3 pp | 19.4% +0.9 pp | 20.6% +0.2 pp | 14.4% +2.4 pp | 25.0% +2.2 pp | 30.0% +4.8 pp | 19.4% -1.4 pp | 29.4% +0.7 pp | 23.9% -0.6 pp | 22.2% +2.4 pp | 20.6% +0.6 pp | 20.0% +0.2 pp | 5.6% -2.9 pp | 21.7% -0.2 pp | 16.1% +1.4 pp | 5/15 |
| credential_lock | 60 | 2 | 1.7% -1.3 pp | 11.7% -6.8 pp | 18.3% -2.0 pp | 6.7% -5.3 pp | 11.7% -11.2 pp | 11.7% -13.5 pp | 16.7% -4.2 pp | 18.3% -10.4 pp | 15.0% -9.5 pp | 10.0% -9.8 pp | 10.0% -10.0 pp | 15.0% -4.8 pp | 6.7% -1.8 pp | 18.3% -3.5 pp | 10.0% -4.7 pp | 15/15 |
| deadlock_dependency | 60 | 2 | 6.7% +3.7 pp | 25.0% +6.5 pp | 23.3% +3.0 pp | 16.7% +4.7 pp | 31.7% +8.8 pp | 35.0% +9.8 pp | 23.3% +2.5 pp | 33.3% +4.6 pp | 28.3% +3.8 pp | 30.0% +10.2 pp | 23.3% +3.3 pp | 25.0% +5.2 pp | 10.0% +1.5 pp | 36.7% +14.8 pp | 18.3% +3.7 pp | 0/15 |
| barrier_synchronization | 30 | 1 | 3.3% +0.3 pp | 26.7% +8.2 pp | 20.0% -0.3 pp | 23.3% +11.3 pp | 23.3% +0.5 pp | 20.0% -5.2 pp | 23.3% +2.5 pp | 26.7% -2.0 pp | 26.7% +2.2 pp | 30.0% +10.2 pp | 30.0% +10.0 pp | 20.0% +0.2 pp | 10.0% +1.5 pp | 23.3% +1.5 pp | 16.7% +2.0 pp | 3/15 |
| controlled_deployment_gate | 30 | 1 | 6.7% +3.7 pp | 20.0% +1.5 pp | 30.0% +9.7 pp | 13.3% +1.3 pp | 33.3% +10.5 pp | 26.7% +1.5 pp | 30.0% +9.2 pp | 40.0% +11.3 pp | 36.7% +12.2 pp | 20.0% +0.2 pp | 30.0% +10.0 pp | 36.7% +16.8 pp | 23.3% +14.8 pp | 23.3% +1.5 pp | 23.3% +8.7 pp | 0/15 |
| coordinated_containment | 30 | 1 | 0.0% -3.0 pp | 6.7% -11.8 pp | 10.0% -10.3 pp | 3.3% -8.7 pp | 10.0% -12.8 pp | 13.3% -11.8 pp | 13.3% -7.5 pp | 16.7% -12.0 pp | 16.7% -7.8 pp | 13.3% -6.5 pp | 16.7% -3.3 pp | 13.3% -6.5 pp | 6.7% -1.8 pp | 6.7% -15.2 pp | 10.0% -4.7 pp | 15/15 |

**(a)** Rows 1–16 of 32 (levers 1–16)

| designed lever | tasks | seeds | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 | below baseline |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| distributed_lock | 30 | 1 | 0.0% -3.0 pp | 10.0% -8.5 pp | 16.7% -3.7 pp | 0.0% -12.0 pp | 3.3% -19.5 pp | 20.0% -5.2 pp | 10.0% -10.8 pp | 16.7% -12.0 pp | 20.0% -4.5 pp | 6.7% -13.2 pp | 10.0% -10.0 pp | 13.3% -6.5 pp | 6.7% -1.8 pp | 23.3% +1.5 pp | 6.7% -8.0 pp | 14/15 |
| duplicate_hold_stale | 30 | 1 | 3.3% +0.3 pp | 0.0% -18.5 pp | 3.3% -17.0 pp | 0.0% -12.0 pp | 6.7% -16.2 pp | 10.0% -15.2 pp | 3.3% -17.5 pp | 16.7% -12.0 pp | 3.3% -21.2 pp | 0.0% -19.8 pp | 6.7% -13.3 pp | 10.0% -9.8 pp | 3.3% -5.2 pp | 13.3% -8.5 pp | 6.7% -8.0 pp | 14/15 |
| multi_device_pairing | 30 | 1 | 3.3% +0.3 pp | 13.3% -5.2 pp | 16.7% -3.7 pp | 16.7% +4.7 pp | 23.3% +0.5 pp | 23.3% -1.8 pp | 16.7% -4.2 pp | 30.0% +1.3 pp | 23.3% -1.2 pp | 20.0% +0.2 pp | 13.3% -6.7 pp | 20.0% +0.2 pp | 3.3% -5.2 pp | 16.7% -5.2 pp | 16.7% +2.0 pp | 8/15 |
| multi_party_synchronization | 30 | 1 | 3.3% +0.3 pp | 20.0% +1.5 pp | 23.3% +3.0 pp | 6.7% -5.3 pp | 23.3% +0.5 pp | 33.3% +8.2 pp | 20.0% -0.8 pp | 30.0% +1.3 pp | 16.7% -7.8 pp | 26.7% +6.8 pp | 16.7% -3.3 pp | 30.0% +10.2 pp | 6.7% -1.8 pp | 20.0% -1.8 pp | 16.7% +2.0 pp | 6/15 |
| non_overlapping_assignment | 30 | 1 | 6.7% +3.7 pp | 40.0% +21.5 pp | 30.0% +9.7 pp | 16.7% +4.7 pp | 40.0% +17.2 pp | 46.7% +21.5 pp | 50.0% +29.2 pp | 46.7% +18.0 pp | 36.7% +12.2 pp | 30.0% +10.2 pp | 40.0% +20.0 pp | 23.3% +3.5 pp | 20.0% +11.5 pp | 40.0% +18.2 pp | 30.0% +15.3 pp | 0/15 |
| reversible_action_hold | 30 | 1 | 0.0% -3.0 pp | 16.7% -1.8 pp | 13.3% -7.0 pp | 6.7% -5.3 pp | 16.7% -6.2 pp | 26.7% +1.5 pp | 6.7% -14.2 pp | 20.0% -8.7 pp | 13.3% -11.2 pp | 16.7% -3.2 pp | 16.7% -3.3 pp | 13.3% -6.5 pp | 3.3% -5.2 pp | 20.0% -1.8 pp | 20.0% +5.3 pp | 13/15 |
| scope_reconciliation | 30 | 1 | 10.0% +7.0 pp | 16.7% -1.8 pp | 13.3% -7.0 pp | 6.7% -5.3 pp | 20.0% -2.8 pp | 23.3% -1.8 pp | 20.0% -0.8 pp | 16.7% -12.0 pp | 20.0% -4.5 pp | 13.3% -6.5 pp | 16.7% -3.3 pp | 16.7% -3.2 pp | 6.7% -1.8 pp | 10.0% -11.8 pp | 6.7% -8.0 pp | 14/15 |
| workflow_completion | 540 | 18 | 3.1% +0.1 pp | 19.3% +0.8 pp | 20.6% +0.2 pp | 12.6% +0.6 pp | 24.1% +1.2 pp | 26.7% +1.5 pp | 21.3% +0.5 pp | 29.8% +1.1 pp | 25.6% +1.1 pp | 20.9% +1.1 pp | 21.1% +1.1 pp | 20.4% +0.5 pp | 8.7% +0.2 pp | 22.2% +0.4 pp | 15.2% +0.5 pp | 0/15 |
| information_transfer | 450 | 15 | 2.9% -0.1 pp | 16.9% -1.6 pp | 18.4% -1.9 pp | 10.4% -1.6 pp | 21.3% -1.5 pp | 22.2% -2.9 pp | 18.9% -1.9 pp | 25.6% -3.1 pp | 21.6% -2.9 pp | 17.6% -2.3 pp | 17.8% -2.2 pp | 19.3% -0.5 pp | 8.4% -0.1 pp | 19.1% -2.7 pp | 14.7% +0.0 pp | 14/15 |
| multi_source_reasoning | 360 | 12 | 3.6% +0.6 pp | 17.8% -0.7 pp | 18.9% -1.4 pp | 11.1% -0.9 pp | 21.9% -0.9 pp | 26.4% +1.2 pp | 21.4% +0.6 pp | 27.8% -0.9 pp | 23.3% -1.2 pp | 18.6% -1.2 pp | 20.0% +0.0 pp | 17.8% -2.1 pp | 7.5% -1.0 pp | 22.5% +0.7 pp | 13.6% -1.1 pp | 10/15 |
| state_repair | 300 | 10 | 3.0% +0.0 pp | 20.0% +1.5 pp | 22.0% +1.7 pp | 12.7% +0.7 pp | 23.0% +0.2 pp | 24.7% -0.5 pp | 20.3% -0.5 pp | 27.3% -1.4 pp | 26.3% +1.8 pp | 20.7% +0.8 pp | 20.7% +0.7 pp | 22.7% +2.8 pp | 9.7% +1.2 pp | 25.7% +3.8 pp | 16.7% +2.0 pp | 3/15 |
| gap_completion | 90 | 3 | 1.1% -1.9 pp | 13.3% -5.2 pp | 20.0% -0.3 pp | 8.9% -3.1 pp | 17.8% -5.1 pp | 17.8% -7.4 pp | 15.6% -5.3 pp | 28.9% +0.2 pp | 17.8% -6.7 pp | 17.8% -2.1 pp | 10.0% -10.0 pp | 16.7% -3.2 pp | 7.8% -0.7 pp | 14.4% -7.4 pp | 8.9% -5.8 pp | 14/15 |
| duplicate_avoidance | 60 | 2 | 1.7% -1.3 pp | 28.3% +9.8 pp | 33.3% +13.0 pp | 25.0% +13.0 pp | 35.0% +12.2 pp | 40.0% +14.8 pp | 38.3% +17.5 pp | 53.3% +24.6 pp | 45.0% +20.5 pp | 33.3% +13.5 pp | 38.3% +18.3 pp | 21.7% +1.8 pp | 8.3% -0.2 pp | 26.7% +4.8 pp | 15.0% +0.3 pp | 2/15 |
| vision_decoy | 60 | 2 | 0.0% -3.0 pp | 15.0% -3.5 pp | 18.3% -2.0 pp | 6.7% -5.3 pp | 15.0% -7.8 pp | 21.7% -3.5 pp | 11.7% -9.2 pp | 18.3% -10.4 pp | 25.0% +0.5 pp | 18.3% -1.5 pp | 13.3% -6.7 pp | 20.0% +0.2 pp | 5.0% -3.5 pp | 28.3% +6.5 pp | 11.7% -3.0 pp | 12/15 |

**(b)** Rows 17–32 of 32 (levers 17–32)

Figure G18: **Designed difficulty levers and perturbation axes across evaluated models.** Displays empirical pass rate shifts induced by designed environment levers across models. Three of the 25 levers push every one of the 15 models below its own baseline (`async_required`, `coordinated_containment`, and `credential_lock`, with `gap_completion` below baseline for 14 of 15 models). **(a)** Rows 1–16 of 32; **(b)** Rows 17–32 of 32.

| scenario (hardest → easiest) | ALL | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| warehouse_robotics_safety_halt_v03 | 0.26 | 0.07 | 0.21 | 0.21 | 0.20 | 0.29 | 0.47 | 0.36 | 0.23 | 0.29 | 0.27 | 0.27 | 0.37 | 0.15 | 0.28 | 0.30 |
| clinical_trial_dosage_conflict_v01 | 0.33 | 0.07 | 0.38 | 0.43 | 0.26 | 0.51 | 0.46 | 0.35 | 0.40 | 0.26 | 0.26 | 0.26 | 0.40 | 0.37 | 0.29 | 0.30 |
| distributed_lock_shared_resource_v02 | 0.35 | 0.12 | 0.35 | 0.36 | 0.27 | 0.36 | 0.41 | 0.32 | 0.50 | 0.36 | 0.30 | 0.30 | 0.38 | 0.34 | 0.46 | 0.36 |
| multimodal_temporal_conflict_01_v01 | 0.36 | 0.21 | 0.42 | 0.40 | 0.32 | 0.38 | 0.40 | 0.35 | 0.48 | 0.42 | 0.33 | 0.42 | 0.34 | 0.19 | 0.44 | 0.34 |
| clinical_trial_dosage_conflict_v02 ⋆ | 0.39 | 0.00 | 0.31 | 0.49 | 0.34 | 0.32 | 0.45 | 0.39 | 0.55 | 0.55 | 0.44 | 0.44 | 0.60 | 0.23 | 0.38 | 0.30 |
| clinical_trial_dosage_conflict_v03 | 0.38 | 0.19 | 0.39 | 0.44 | 0.21 | 0.46 | 0.39 | 0.54 | 0.35 | 0.40 | 0.44 | 0.44 | 0.45 | 0.31 | 0.27 | 0.48 |
| coordinated_containment_v01 | 0.40 | 0.12 | 0.35 | 0.42 | 0.38 | 0.45 | 0.53 | 0.50 | 0.55 | 0.51 | 0.44 | 0.40 | 0.35 | 0.27 | 0.45 | 0.24 |
| incident_commander_selection_v01 | 0.40 | 0.00 | 0.46 | 0.44 | 0.33 | 0.41 | 0.58 | 0.56 | 0.58 | 0.48 | 0.53 | 0.48 | 0.46 | 0.00 | 0.35 | 0.38 |
| distributed_lock_shared_resource_v03 | 0.41 | 0.33 | 0.43 | 0.38 | 0.25 | 0.42 | 0.53 | 0.43 | 0.52 | 0.44 | 0.41 | 0.40 | 0.38 | 0.43 | 0.46 | 0.32 |
| multimodal_temporal_conflict_01_v02 | 0.40 | 0.06 | 0.50 | 0.35 | 0.37 | 0.48 | 0.50 | 0.53 | 0.47 | 0.52 | 0.36 | 0.47 | 0.34 | 0.13 | 0.59 | 0.38 |
| multi_tab_slot_monitoring_v02 | 0.42 | 0.17 | 0.47 | 0.51 | 0.30 | 0.51 | 0.51 | 0.43 | 0.59 | 0.49 | 0.45 | 0.47 | 0.38 | 0.23 | 0.51 | 0.26 |
| sla_support_response_v03 | 0.42 | 0.31 | 0.48 | 0.43 | 0.39 | 0.46 | 0.43 | 0.44 | 0.33 | 0.51 | 0.48 | 0.43 | 0.45 | 0.30 | 0.52 | 0.34 |
| payment_chargeback_dispute_v03 | 0.46 | 0.03 | 0.53 | 0.57 | 0.27 | 0.63 | 0.60 | 0.47 | 0.73 | 0.77 | 0.47 | 0.73 | 0.43 | 0.07 | 0.23 | 0.30 |
| payment_chargeback_dispute_v02 | 0.44 | 0.07 | 0.48 | 0.42 | 0.41 | 0.48 | 0.58 | 0.58 | 0.50 | 0.42 | 0.46 | 0.42 | 0.49 | 0.42 | 0.43 | 0.46 |
| multi_tab_slot_monitoring_v01 | 0.48 | 0.16 | 0.45 | 0.41 | 0.37 | 0.61 | 0.70 | 0.48 | 0.76 | 0.54 | 0.54 | 0.51 | 0.59 | 0.29 | 0.54 | 0.26 |
| multimodal_temporal_conflict_01_v03 | 0.47 | 0.24 | 0.47 | 0.55 | 0.30 | 0.56 | 0.52 | 0.57 | 0.50 | 0.67 | 0.54 | 0.48 | 0.35 | 0.35 | 0.59 | 0.36 |
| satellite_ground_station_handover_v01 | 0.50 | 0.03 | 0.47 | 0.37 | 0.50 | 0.56 | 0.78 | 0.64 | 0.69 | 0.64 | 0.57 | 0.61 | 0.48 | 0.10 | 0.49 | 0.52 |
| employee_onboarding_workflow_v03 | 0.50 | 0.44 | 0.57 | 0.62 | 0.38 | 0.52 | 0.49 | 0.43 | 0.57 | 0.46 | 0.56 | 0.53 | 0.49 | 0.44 | 0.44 | 0.52 |
| employee_onboarding_workflow_v01 | 0.51 | 0.23 | 0.55 | 0.57 | 0.52 | 0.58 | 0.57 | 0.52 | 0.64 | 0.56 | 0.58 | 0.53 | 0.50 | 0.32 | 0.55 | 0.49 |
| multi_party_access_permit_v01 | 0.52 | 0.23 | 0.51 | 0.54 | 0.48 | 0.59 | 0.69 | 0.47 | 0.59 | 0.56 | 0.57 | 0.54 | 0.61 | 0.40 | 0.41 | 0.56 |
| multi_tab_slot_monitoring_v03 | 0.52 | 0.26 | 0.55 | 0.49 | 0.37 | 0.57 | 0.66 | 0.58 | 0.58 | 0.59 | 0.55 | 0.58 | 0.57 | 0.27 | 0.62 | 0.51 |
| multi_device_pairing_v01 | 0.52 | 0.33 | 0.63 | 0.59 | 0.33 | 0.53 | 0.56 | 0.54 | 0.53 | 0.50 | 0.57 | 0.49 | 0.58 | 0.49 | 0.66 | 0.54 |
| incident_commander_selection_v03 | 0.54 | 0.09 | 0.55 | 0.67 | 0.40 | 0.73 | 0.70 | 0.66 | 0.77 | 0.70 | 0.67 | 0.67 | 0.47 | 0.32 | 0.56 | 0.20 |
| warehouse_robotics_safety_halt_v01 | 0.53 | 0.24 | 0.57 | 0.55 | 0.34 | 0.63 | 0.75 | 0.57 | 0.53 | 0.75 | 0.50 | 0.55 | 0.64 | 0.40 | 0.44 | 0.44 |
| warehouse_robotics_safety_halt_v02 ⋆ | 0.52 | 0.10 | 0.48 | 0.52 | 0.39 | 0.65 | 0.80 | 0.59 | 0.45 | 0.58 | 0.75 | 0.64 | 0.58 | 0.10 | 0.68 | 0.54 |
| collaborative_contract_review_v01 | 0.55 | 0.29 | 0.69 | 0.56 | 0.49 | 0.53 | 0.62 | 0.59 | 0.63 | 0.63 | 0.61 | 0.62 | 0.65 | 0.33 | 0.61 | 0.39 |
| sla_support_response_v01 | 0.55 | 0.39 | 0.53 | 0.55 | 0.49 | 0.61 | 0.65 | 0.53 | 0.64 | 0.60 | 0.57 | 0.55 | 0.57 | 0.51 | 0.59 | 0.50 |
| employee_onboarding_workflow_v02 | 0.56 | 0.32 | 0.59 | 0.51 | 0.51 | 0.55 | 0.61 | 0.63 | 0.71 | 0.60 | 0.59 | 0.53 | 0.57 | 0.47 | 0.63 | 0.59 |
| expiring_resource_hold_v02 | 0.57 | 0.39 | 0.60 | 0.60 | 0.65 | 0.70 | 0.71 | 0.65 | 0.60 | 0.47 | 0.49 | 0.47 | 0.53 | 0.50 | 0.54 | 0.65 |
| incident_commander_selection_v02 | 0.58 | 0.34 | 0.64 | 0.70 | 0.45 | 0.72 | 0.69 | 0.62 | 0.76 | 0.75 | 0.67 | 0.68 | 0.55 | 0.29 | 0.40 | 0.47 |

**(a)** Rows 1–30 of 60 (most challenging to median)

| scenario (hardest → easiest) | ALL | gemini-3.5-flash | gemini-3.6-flash | gemini-3.7-flash | gemini-3.1-pro | claude-opus-4.8 | claude-opus-5 | claude-sonnet-5 | GPT-6 Astra | gpt-5.6-sol | gpt-5.6-luna | gpt-5.6-terra | deepseek-v4-pro | deepseek-v4-flash | kimi-k3 | glm-5.2-fp8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| coordinated_containment_v02 ⋆ | 0.58 | 0.14 | 0.65 | 0.58 | 0.55 | 0.65 | 0.66 | 0.63 | 0.66 | 0.74 | 0.69 | 0.71 | 0.60 | 0.38 | 0.48 | 0.53 |
| satellite_ground_station_handover_v03 | 0.58 | 0.07 | 0.60 | 0.72 | 0.58 | 0.69 | 0.74 | 0.81 | 0.76 | 0.72 | 0.74 | 0.76 | 0.54 | 0.19 | 0.62 | 0.20 |
| access_request_approval_v03 | 0.57 | 0.56 | 0.70 | 0.62 | 0.54 | 0.45 | 0.62 | 0.47 | 0.60 | 0.60 | 0.54 | 0.57 | 0.68 | 0.62 | 0.54 | 0.50 |
| multi_device_pairing_v03 | 0.60 | 0.37 | 0.57 | 0.61 | 0.56 | 0.68 | 0.74 | 0.70 | 0.75 | 0.72 | 0.65 | 0.61 | 0.61 | 0.42 | 0.49 | 0.45 |
| multi_party_access_permit_v03 | 0.60 | 0.26 | 0.69 | 0.67 | 0.64 | 0.68 | 0.72 | 0.63 | 0.70 | 0.70 | 0.69 | 0.66 | 0.63 | 0.40 | 0.45 | 0.46 |
| barrier_sync_deployment_v03 | 0.60 | 0.30 | 0.67 | 0.60 | 0.64 | 0.59 | 0.58 | 0.57 | 0.68 | 0.67 | 0.71 | 0.73 | 0.69 | 0.39 | 0.60 | 0.62 |
| expiring_resource_hold_v01 | 0.61 | 0.25 | 0.70 | 0.62 | 0.61 | 0.65 | 0.66 | 0.67 | 0.76 | 0.66 | 0.64 | 0.63 | 0.66 | 0.45 | 0.61 | 0.54 |
| barrier_sync_deployment_v01 | 0.61 | 0.33 | 0.69 | 0.64 | 0.64 | 0.67 | 0.61 | 0.63 | 0.72 | 0.66 | 0.67 | 0.62 | 0.61 | 0.36 | 0.68 | 0.59 |
| it_deployment_gate_v01 | 0.61 | 0.35 | 0.58 | 0.57 | 0.46 | 0.72 | 0.72 | 0.70 | 0.68 | 0.64 | 0.70 | 0.67 | 0.66 | 0.50 | 0.58 | 0.57 |
| multi_device_pairing_v02 | 0.62 | 0.25 | 0.49 | 0.61 | 0.64 | 0.69 | 0.68 | 0.72 | 0.81 | 0.74 | 0.72 | 0.69 | 0.66 | 0.32 | 0.62 | 0.62 |
| sla_support_response_v02 | 0.60 | 0.38 | 0.67 | 0.63 | 0.48 | 0.65 | 0.71 | 0.60 | 0.47 | 0.62 | 0.68 | 0.64 | 0.67 | 0.60 | 0.63 | 0.55 |
| it_deployment_gate_v02 | 0.61 | 0.49 | 0.63 | 0.66 | 0.55 | 0.61 | 0.48 | 0.55 | 0.67 | 0.67 | 0.61 | 0.64 | 0.66 | 0.67 | 0.69 | 0.64 |
| expense_reimbursement_conflict_v02 ⋆ | 0.62 | 0.32 | 0.71 | 0.70 | 0.59 | 0.75 | 0.69 | 0.59 | 0.64 | 0.67 | 0.72 | 0.57 | 0.64 | 0.46 | 0.79 | 0.48 |
| deadlock_multi_approval_v01 | 0.63 | 0.47 | 0.67 | 0.62 | 0.60 | 0.70 | 0.72 | 0.70 | 0.72 | 0.65 | 0.67 | 0.66 | 0.62 | 0.43 | 0.64 | 0.60 |
| barrier_sync_deployment_v02 | 0.64 | 0.32 | 0.60 | 0.63 | 0.51 | 0.73 | 0.70 | 0.68 | 0.78 | 0.76 | 0.76 | 0.76 | 0.61 | 0.55 | 0.62 | 0.52 |
| it_deployment_gate_v03 | 0.64 | 0.30 | 0.69 | 0.64 | 0.62 | 0.72 | 0.60 | 0.62 | 0.84 | 0.80 | 0.65 | 0.75 | 0.71 | 0.42 | 0.64 | 0.61 |
| access_request_approval_v01 | 0.64 | 0.32 | 0.70 | 0.72 | 0.60 | 0.70 | 0.72 | 0.71 | 0.79 | 0.73 | 0.67 | 0.73 | 0.61 | 0.52 | 0.49 | 0.58 |
| expense_reimbursement_conflict_v01 | 0.64 | 0.22 | 0.69 | 0.53 | 0.55 | 0.78 | 0.72 | 0.67 | 0.82 | 0.82 | 0.84 | 0.79 | 0.72 | 0.22 | 0.62 | 0.65 |
| distributed_lock_shared_resource_v01 ⋆ | 0.63 | 0.27 | 0.68 | 0.65 | 0.50 | 0.63 | 0.79 | 0.72 | 0.68 | 0.72 | 0.62 | 0.64 | 0.64 | 0.45 | 0.79 | 0.73 |
| access_request_approval_v02 | 0.64 | 0.28 | 0.71 | 0.69 | 0.47 | 0.80 | 0.80 | 0.79 | 0.74 | 0.77 | 0.70 | 0.73 | 0.68 | 0.45 | 0.58 | 0.44 |
| satellite_ground_station_handover_v02 ⋆ | 0.65 | 0.00 | 0.77 | 0.73 | 0.77 | 0.80 | 0.68 | 0.80 | 0.87 | 0.87 | 0.77 | 0.80 | 0.67 | 0.20 | 0.60 | 0.47 |
| expense_reimbursement_conflict_v03 | 0.64 | 0.31 | 0.69 | 0.73 | 0.48 | 0.74 | 0.71 | 0.56 | 0.66 | 0.68 | 0.69 | 0.66 | 0.70 | 0.61 | 0.77 | 0.62 |
| expiring_resource_hold_v03 ⋆ | 0.65 | 0.24 | 0.65 | 0.75 | 0.61 | 0.70 | 0.79 | 0.62 | 0.81 | 0.77 | 0.73 | 0.78 | 0.81 | 0.34 | 0.72 | 0.46 |
| coordinated_containment_v03 | 0.65 | 0.51 | 0.65 | 0.76 | 0.61 | 0.70 | 0.64 | 0.65 | 0.69 | 0.73 | 0.75 | 0.73 | 0.69 | 0.46 | 0.67 | 0.45 |
| collaborative_contract_review_v03 | 0.67 | 0.30 | 0.74 | 0.70 | 0.65 | 0.80 | 0.81 | 0.80 | 0.79 | 0.80 | 0.75 | 0.81 | 0.63 | 0.38 | 0.62 | 0.43 |
| deadlock_multi_approval_v03 | 0.69 | 0.39 | 0.71 | 0.76 | 0.68 | 0.76 | 0.70 | 0.72 | 0.82 | 0.76 | 0.74 | 0.73 | 0.70 | 0.57 | 0.69 | 0.65 |
| multi_party_access_permit_v02 | 0.73 | 0.45 | 0.73 | 0.72 | 0.62 | 0.79 | 0.83 | 0.76 | 0.88 | 0.73 | 0.84 | 0.73 | 0.83 | 0.51 | 0.79 | 0.75 |
| collaborative_contract_review_v02 | 0.74 | 0.52 | 0.80 | 0.71 | 0.65 | 0.83 | 0.77 | 0.79 | 0.81 | 0.79 | 0.72 | 0.83 | 0.73 | 0.67 | 0.75 | 0.77 |
| payment_chargeback_dispute_v01 ⋆ | 0.75 | 0.27 | 0.83 | 0.83 | 0.79 | 0.88 | 0.72 | 0.83 | 0.93 | 0.90 | 0.78 | 0.87 | 0.91 | 0.49 | 0.50 | 0.72 |
| deadlock_multi_approval_v02 | 0.76 | 0.47 | 0.86 | 0.82 | 0.76 | 0.86 | 0.86 | 0.77 | 0.91 | 0.84 | 0.79 | 0.72 | 0.75 | 0.50 | 0.84 | 0.68 |

**(b)** Rows 31–60 of 60 (median to easiest)

Figure G19: **Scenario-level mean score heatmap across evaluated models.** Displays mean scores per scenario (10 rounds each), sorted from most challenging to easiest by cross-model mean score. Starred scenarios are analyzed in qualitative walkthroughs (§G.10). **(a)** Rows 1–30 of 60; **(b)** Rows 31–60 of 60.

## G.9 THE REASONING-EFFORT SWEEP

### G.9.1 REASONING TOKEN CONSUMPTION ACROSS EFFORT LEVELS

| Model | reasoning tok/ call (l/m/h) | reasoning tok/ task (l/m/h) | low→high | zero-reasoning calls (l/m/h) |
|---|---|---|---|---|
| Gemini 3.5 Flash | 236.3 / 449.3 / **445.8** | 3,945 / 8,800 / 8,826 | 1.89× | 0.6% / 0.4% / 0.4% |
| Gemini 3.6 Flash | 65.3 / 231.3 / 252.7 | 704 / 3,568 / 4,266 | 3.87× | **57.3%** / 0.0% / 0.0% |
| Gemini 3.7 Flash | 83.4 / 165.6 / 233.0 | 1,079 / 2,578 / 4,237 | 2.80× | 39.1% / 0.0% / 0.0% |
| Gemini 3.1 Pro | 93.6 / 118.0 / 171.2 | 1,002 / 1,394 / 2,359 | 1.83× | 1.0% / 0.1% / 0.2% |
| Claude Opus 4.8 | 68.6 / 126.1 / 177.8 | 717 / 1,493 / 2,291 | 2.59× | 64.1% / 57.2% / 51.4% |
| Claude Opus 5 | **5.1 / 17.6 / 46.5** | **45 / 184 / 553** | **9.07**× | 95.3% / 88.7% / 76.3% |
| Claude Sonnet 5 | 49.6 / 114.4 / 249.8 | 406 / 1,095 / 2,847 | 5.03× | 72.7% / 64.5% / 53.0% |
| GPT-5.6 Sol | 27.7 / 57.8 / 94.0 | 301 / 760 / 1,354 | 3.39× | 10.6% / 9.7% / 10.2% |
| GPT-5.6 Luna | 29.1 / 54.0 / 140.0 | 289 / 632 / 1,994 | 4.82× | 23.4% / 26.6% / 29.0% |
| GPT-5.6 Terra | 25.3 / 57.2 / 142.5 | 243 / 625 / 1,760 | 5.63× | 0.4% / 0.1% / 0.1% |

Table G18: **Reasoning token generation across effort levels.** Reports internal reasoning tokens generated per tool call, per task, scaling factor (Low → High), and frequency of zero-reasoning calls across the ten swept API models.

### G.9.2 PERFORMANCE IMPACT OF REASONING EFFORT SCALING

| Model | pass low/ med/high | score low/ med/high | mean steps l/m/h | capped% l/m/h |
|---|---|---|---|---|
| Claude Opus 5 | 23.5 / **25.3** / 25.2 | 0.609 / 0.633 / **0.642** | 7.99 / 9.65 / 11.12 | 0.0 / 0.0 / 0.5 |
| GPT-5.6 Sol | 20.8 / 23.0 / **24.5** | 0.610 / 0.628 / **0.633** | 10.04 / 12.41 / 13.65 | 0.2 / 2.0 / 1.2 |
| Claude Opus 4.8 | 21.3 / **23.8** / 22.8 | 0.608 / 0.618 / **0.621** | 9.67 / 11.11 / 12.17 | 0.8 / 2.8 / 6.2 |
| Claude Sonnet 5 | 13.8 / 17.0 / **20.8** | 0.537 / 0.570 / **0.596** | 7.31 / 8.76 / 10.64 | 0.3 / 1.0 / 3.3 |
| Gemini 3.7 Flash | 20.5 / **21.5** / 20.3 | 0.600 / **0.618** / 0.581 | 12.18 / 14.91 / 17.79 | 3.0 / 12.7 / 40.7 |
| GPT-5.6 Terra | 17.7 / 20.0 / 20.0 | 0.572 / 0.589 / **0.602** | 8.65 / 9.98 / 11.35 | 0.3 / 1.3 / 1.3 |
| GPT-5.6 Luna | 19.5 / 17.7 / **19.8** | 0.590 / 0.591 / **0.604** | 9.12 / 10.91 / 13.54 | 0.5 / 2.7 / 9.2 |
| Gemini 3.6 Flash | 18.5 / **20.2** / 18.5 | 0.595 / **0.602** / 0.586 | 9.99 / 14.77 / 16.31 | 1.8 / 14.5 / 24.7 |
| Gemini 3.1 Pro | 8.3 / 9.8 / **12.0** | 0.463 / 0.479 / **0.487** | 9.84 / 11.02 / 13.16 | 4.7 / 10.2 / 25.3 |
| Gemini 3.5 Flash | **13.5** / 3.2 / 3.0 | **0.473** / 0.299 / 0.259 | 16.22 / 19.46 / 19.70 | 38.7 / 85.0 / 89.2 |

Table G19: **Performance metrics across reasoning effort configurations.** Reports pass rate, mean check score, average steps per episode, and budget cap exhaustion rate across Low, Medium, and High effort levels for all ten API models.

| Model | Δpp | lost | gained | p | verdict |
|---|---|---|---|---|---|
| Claude Sonnet 5 | **+7.0** | 11 | **53** | 1.0e-07 | **real** |
| Gemini 3.1 Pro | +3.7 | **9** | 31 | 6.8e-04 | **real** |
| GPT-5.6 Sol | +3.7 | 12 | 34 | 1.6e-03 | **real** |
| GPT-5.6 Terra | +2.3 | 15 | 29 | 0.049 | **real** |
| Claude Opus 5 | +1.7 | 23 | 33 | 0.23 | n.s. |
| Claude Opus 4.8 | +1.5 | 18 | 27 | 0.23 | n.s. |
| GPT-5.6 Luna | +0.3 | 24 | 26 | 0.89 | n.s. |
| Gemini 3.6 Flash | 0.0 | 34 | 34 | 1.00 | n.s. |
| Gemini 3.7 Flash | −0.2 | 34 | 33 | 1.00 | n.s. |
| Gemini 3.5 Flash | **−10.5** | **63** | **0** | 2.2e-19 | **real (negative)** |

Table G20: **Paired McNemar significance tests across reasoning effort scaling (Low → High).** Evaluates task-level changes over identical 600 rounds per model ($n = 600$). Significance verdicts reflect uncorrected ($\alpha = 0.05$) and Bonferroni-corrected family-wise thresholds ($\alpha = 0.05/30 = 0.00167$).

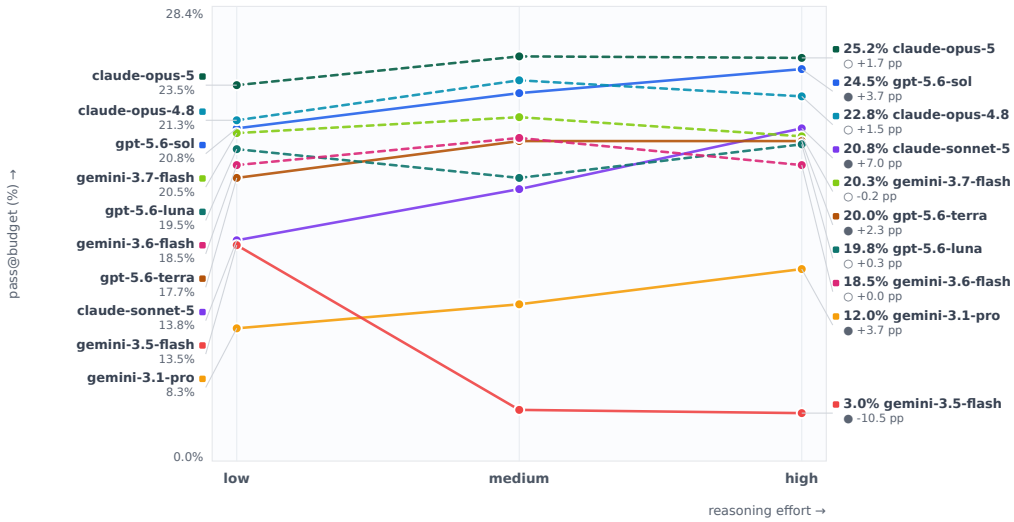



Figure G20: **Pass rate scaling across reasoning effort configurations (Low, Medium, High).** Visualizes task completion rate trajectories across effort levels for all ten swept API models.

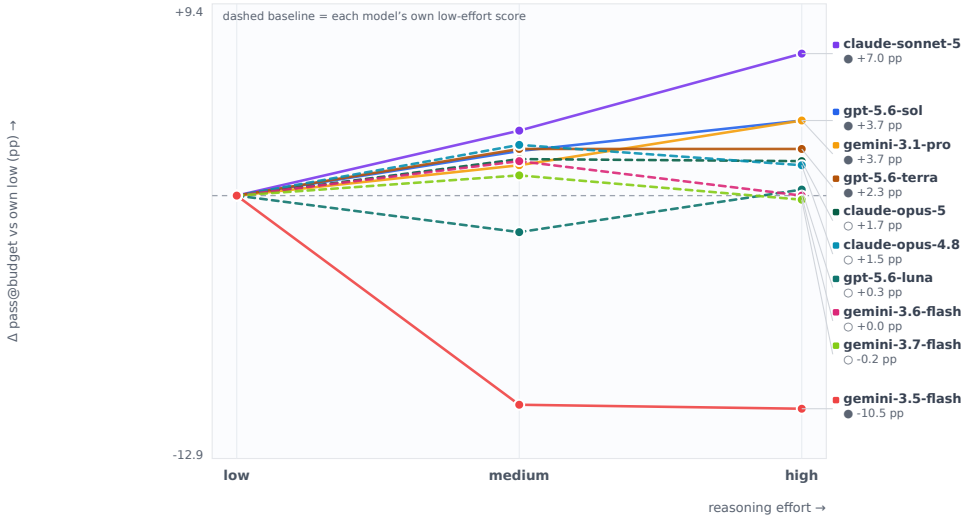



Figure G21: **Marginal pass rate delta relative to low effort baseline.** Displays pass rate differences ($\Delta$pp) achieved at medium and high effort relative to each model's native low-effort performance.




Figure G22: **Mean step expenditure across reasoning effort levels against the 20-step budget.** Illustrates how internal deliberation expands action trajectory length, driving horizon exhaustion on bounded models.

### G.10 Qualitative Analysis: Representative Conflict Scenarios and Model Signatures

To examine detailed agent behavior and where models diverge, this section provides step-level walkthroughs across eight conflict scenarios with large score differences across the 14 20-step baseline models, followed by 14 case studies showing each baseline model's primary failure mode.

#### Paired Tool-Execution Chains across Eight Representative Scenarios

Each exhibit shows all 14 20-step baseline models tested on the exact same task instance. Rows show sequential tool calls (colored by action type; red circles indicate tool execution errors), along with final check results, step counts, visual assets opened, files written, and termination reasons.

● `asset open` ● `file read` ● `message broadcast/send` ● `message poll/search` ● `asset describe/info/list/read-text` ● `file append/create` ● `booking hold/status`, `containment block-token/disable-forwarding/status`, `lock acquire/status`, `system cron/status`, `tasks list` ● `wait` ● `file list` ● `agent list/status` ● `agent delegate` ○ a red ring marks a step whose tool returned an error




Figure G23: **Representative Execution Trace: Payment Chargeback Dispute (`v01_r04`; Figure G23).** Compares agent action traces on credit verification, report delivery, and notifications under conflicting financial ledgers.

**Case Analysis: Payment Chargeback Dispute (`v01_r04`).** This scenario evaluates multi-agent coordination under conflicting billing records across three scored checks: (1) extracting the validation code `PASSPORT-VERIFY-9982`, (2) reporting the chargeback dispute in `reports/support_escrow_key_verification.txt`, and (3) sending the code to `Casey_Support_Specialist`.

All 14 baseline models send a message to `Casey_Support_Specialist` and write to the correct report path. However, only three models (GPT-5.6 Sol at step 5, DeepSeek V4 Pro at step 4, and GPT-5.6 Luna at step 10) include the exact required code in their message. GPT-5.6 Sol is the only model to pass completely (11 steps), correctly writing the report and sending the notification. Eight models achieve partial credit (0.67) by writing both required findings in a single report but missing the exact message content. Notably, Claude Sonnet 5 finishes in 7 steps without opening any images by skipping the diagnostic escrow panel, whereas Kimi K3 uses all 20 steps and opens 5 images before writing its report at step 19.

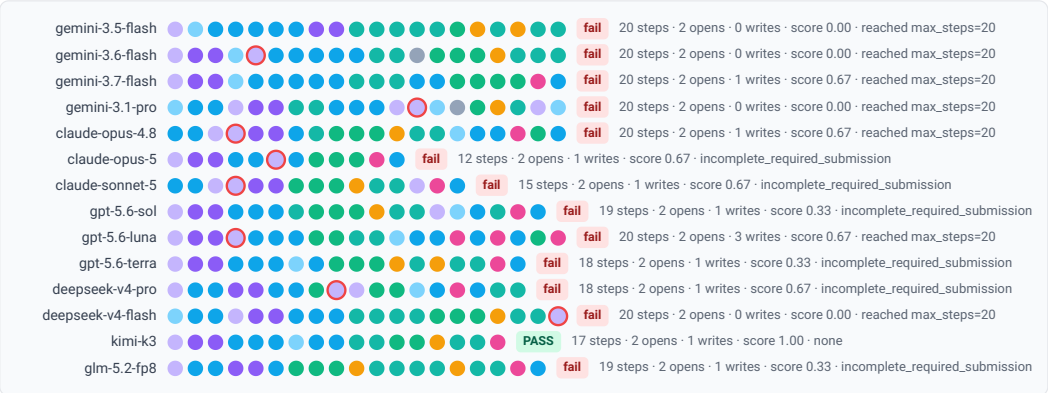



Figure G24: **Representative Execution Trace: Warehouse Robotics Safety Halt (`v02_r01`; Figure G24).** Compares emergency threshold verification, thermal sensor reading extraction (`88.5°C`), and safety channel notification across models.

**Case Analysis: Warehouse Robotics Safety Halt (`v02_r01`).** This scenario requires identifying thermal sensor anomalies across three scored units: (1) extracting the temperature reading `88.5°C`, (2) filing the safety halt verdict in the incident report, and (3) broadcasting the emergency alert to `Avery_Safety_Officer` in channel `#ops-safety-emergency-b7`.

Kimi K3 is the sole model to pass (17 steps), issuing the alert at step 11 and committing the report at step 16. Six models (Claude Opus 5, Claude Sonnet 5, DeepSeek V4 Pro, Gemini 3.7 Flash, Claude Opus 4.8, GPT-5.6 Luna) score 0.67 by satisfying both report checks while missing the channel notification. Four models (Gemini 3.5 Flash, Gemini 3.6 Flash, Gemini 3.1 Pro, DeepSeek V4 Flash) score 0.00, exhausting the 20-step budget with zero file writes. Failure modes illustrate three distinct loss mechanisms: Gemini 3.1 Pro misidentifies the recipient (messaging `Blair_Robotics_Engineer` instead), GPT-5.6 Luna substitutes `message broadcast` for direct messaging, and the remaining models fail on message payload syntax.

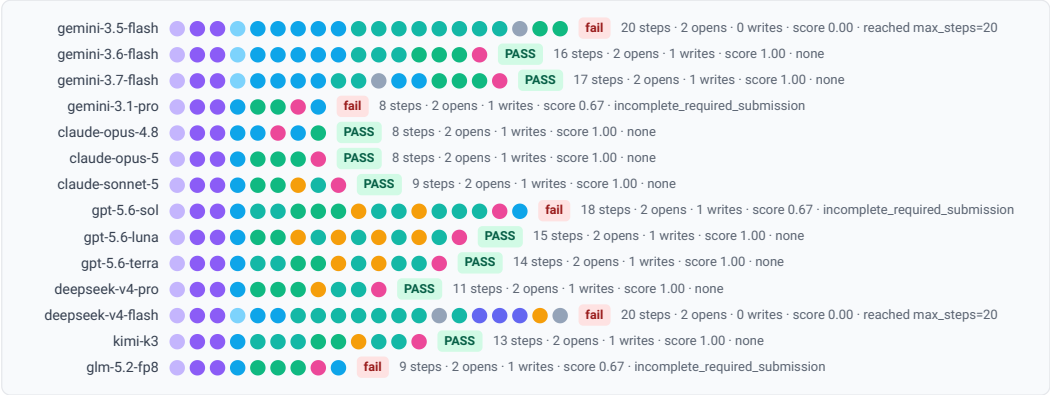



Figure G25: **Representative Execution Trace: Satellite Ground Station Handover (`v02_r01`; Figure G25).** Illustrates telemetry alarm resolution (`ERR-LOCK-049-DRV`) and operations handover messaging under uniform visual grounding.

**Case Analysis: Satellite Ground Station Handover (`v02_r01`).** The rubric comprises three units: (1) transcribing the diagnostic alert `ERR-LOCK-049-DRV`, (2) stating the handover status verdict, and (3) notifying `Devon_Ground_Ops_Lead` in channel `#ops-handover`.

Nine of the 14 baseline models successfully pass. Crucially, all 14 baseline models open exactly 2 visual assets, providing an empirical control where grounding volume is identical across the entire field. The primary distinguishing factor is the final verdict inference: three models (Gemini 3.1 Pro, GLM-5.2 FP8, GPT-5.6 Sol) extract the code and notify the operator but fail the conclusion check (0.67), while two models (Gemini 3.5 Flash, DeepSeek V4 Flash) exhaust the action budget without writing (0.00). Execution horizon spans 8 steps (Claude Opus 4.8, Claude Opus 5) to 16 steps (Gemini 3.6 Flash) among passing trajectories.




Figure G26: **Representative Execution Trace: Clinical Trial Dosage Conflict (`v02_r03`; Figure G26).** Highlights divergence in causal interpretation (control cohort mislabeling vs. toxicity) despite identical numerical extraction (`142 U/L`).

**Case Analysis: Clinical Trial Dosage Conflict (`v02_r03`).** This scenario tests causal conflict resolution across three units: (1) extracting the liver enzyme reading `142 U/L` into `reports/cohort_audit.md`, (2) alerting `Avery_Trial_Safety_Monitor` in `#pv-safety-governance`, and (3) correctly identifying that elevated readings stem from control-cohort mislabeling rather than true drug toxicity.

Eight models successfully pass. Literal extraction and recipient routing are universal among all models that generate deliverables. However, four models (Claude Opus 4.8, GPT-5.6 Terra, GPT-5.6 Sol, GPT-5.6 Luna) score 0.67 due to misattributing the anomaly to pharmaceutical toxicity. Grounding volume does not predict accuracy: Gemini 3.7 Flash passes with 2 asset opens, whereas Claude Opus 4.8 fails with 3 opens, and Gemini 3.5 Flash scores 0.00 with zero opens and zero writes.

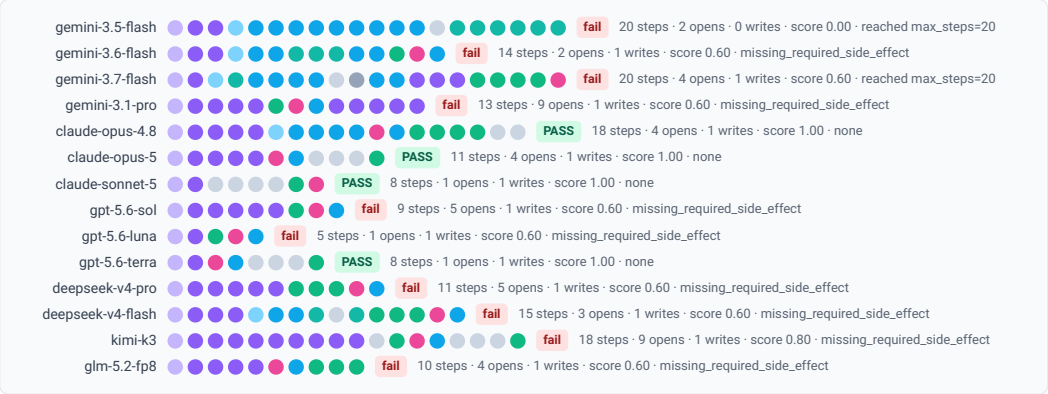



Figure G27: **Representative Execution Trace: Coordinated Security Containment (`v02_r04`; Figure G27).** Shows the gap between writing diagnostic reports and performing required actions in the environment (revoking an API token and blocking email forwarding).

**Case Analysis: Coordinated Security Containment (`v02_r04`).** This scenario requires coordinated incident response across five checks: two environment actions (blocking `standard_user_449` and revoking API token `tok_live_sec_99321`), reporting the anomaly rate `142 msgs/hr`, reporting the incident verdict, and notifying the on-call engineer.

While nearly all models write the report, taking real actions in the environment is the main challenge. Exactly four models (Claude Sonnet 5, GPT-5.6 Terra, Claude Opus 5, Claude Opus 4.8) execute both containment actions with correct parameters, accounting for all four passes. Eight models execute neither action, stopping with a `missing_required_side_effect` error after writing an accurate report about a breach they failed to contain (0.60). Kimi K3 gets partial credit (0.80) due to wrong arguments (disabling forwarding for `j.doe@company.com` instead of `standard_user_449`).

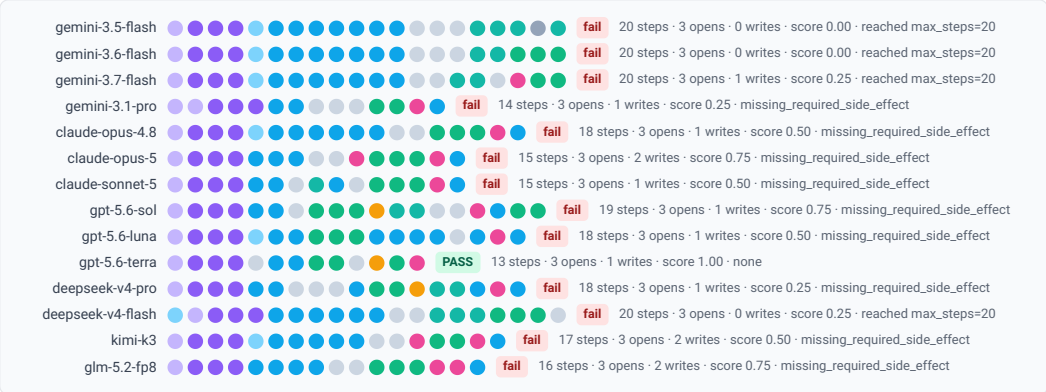



Figure G28: **Representative Execution Trace: Distributed Lock Conflict Resolution (`v01_r02`; Figure G28).** Examines identifying race conditions and acquiring the required lock on `inventory_ledger`.

**Case Analysis: Distributed Lock Conflict Resolution (`v01_r02`).** The rubric evaluates conflict resolution across four checks: identifying the state `UNLOCKED (Un-serialized Write Blocked)`, writing the audit report, acquiring the lock on `inventory_ledger`, and sending the confirmation notification.

GPT-5.6 Terra is the only model to pass (13 steps). All 14 baseline models open all 3 images. However, acquiring the shared ledger lock is difficult: Claude Sonnet 5 acquires the lock at step 9 but scores 0.50 due to incomplete report and message contents, showing that acquiring locks is necessary but not sufficient without accurate deliverables.

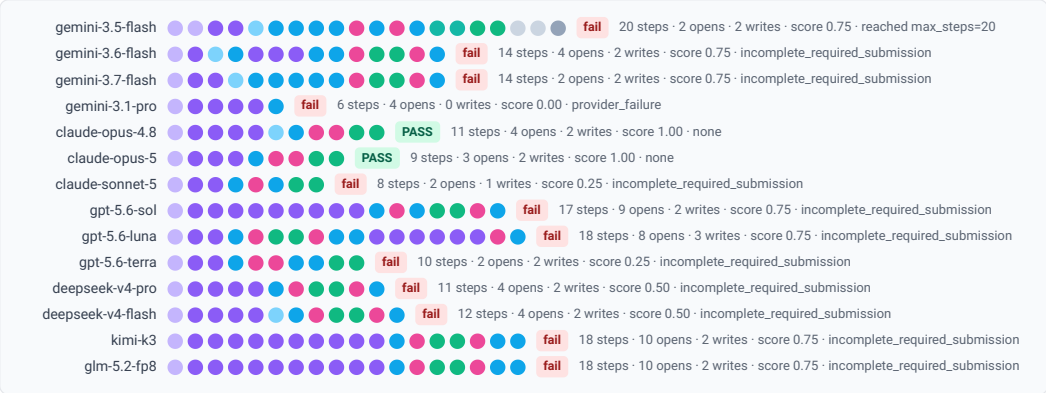



Figure G29: **Representative Execution Trace: Expense Reimbursement Audit (`v02_r05`; Figure G29).** Contrasts telemetry configuration changes, audit passcode delivery, and multi-recipient stakeholder notifications.

**Case Analysis: Expense Reimbursement Audit (`v02_r05`).** This scenario requires modifying environment configuration across four units: reading `config/telemetry_settings.json`, rewriting it with audit passcode `AUDIT-8842-REV`, generating the report, and notifying both `Blair_Audit_Investigator` and `Casey_Approver`.

Claude Opus 5 (9 steps) and Claude Opus 4.8 (11 steps) achieve clean passes. Seven models form a 0.75 plateau by successfully rewriting the configuration file and sending notifications but missing report precision. Claude Sonnet 5 acts without modifying the configuration file (scoring 0.25 in 8 steps), while GPT-5.6 Terra scores 0.25 via the inverse pattern (rewriting configuration but failing on notification delivery).

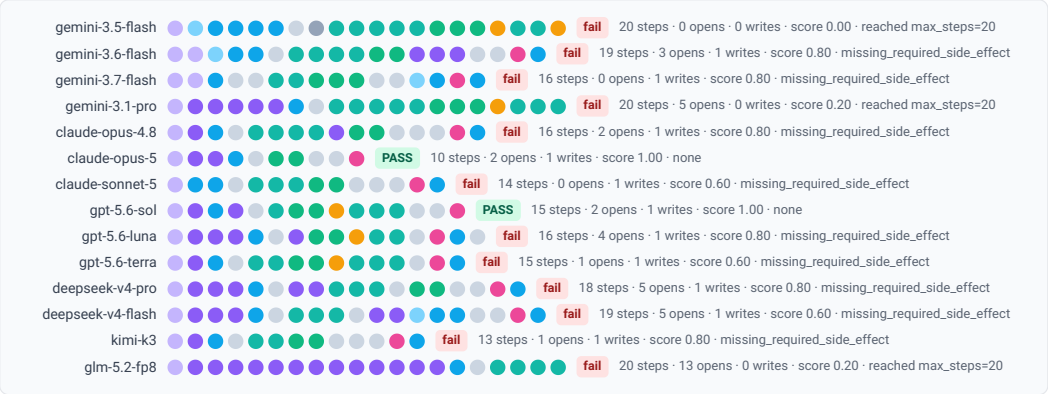



Figure G30: **Representative Execution Trace: Expiring GPU Cluster Resource Hold (`v03_r04`; Figure G30).** Evaluates inspecting the reservation board, acquiring the provisional hold, and reporting the lease expiration.

**Case Analysis: Expiring GPU Cluster Resource Hold (`v03_r04`).** This scenario tests visual inspection and resource management across five checks: (1) inspecting the visual dashboard `cluster_dashboard_reservation_board_active_hold_r04`, (2) acquiring hold token `HOLD-80G-9412-PROVISIONAL` on slot `A100-80G-02` with a 300 s duration, (3) reporting the lease expiration, (4) stating the report verdict, and (5) notifying the stakeholder.

Claude Opus 5 (10 steps) and GPT-5.6 Sol (15 steps) pass all five checks. Six models score 0.80 by opening the image and acquiring the hold, but failing on notification details. Claude Sonnet 5 misses the visual check (0 images opened) but still acquires the hold and writes the report (0.60), while GLM-5.2 FP8 opens 13 images but runs out of steps without writing any files (0.20).

### SIGNATURE FAILURE MODES (ONE EXEMPLAR PER MODEL)

One round was chosen to clearly illustrate each model's characteristic behavior in Figures G31–G44 (G31, G32, G33, G34, G35, G36, G37, G38, G39, G40, G41, G42, G43, G44). We use the same format as above—all 14 20-step baseline models tested on the same task, with action dots and the highlighted model shown first.

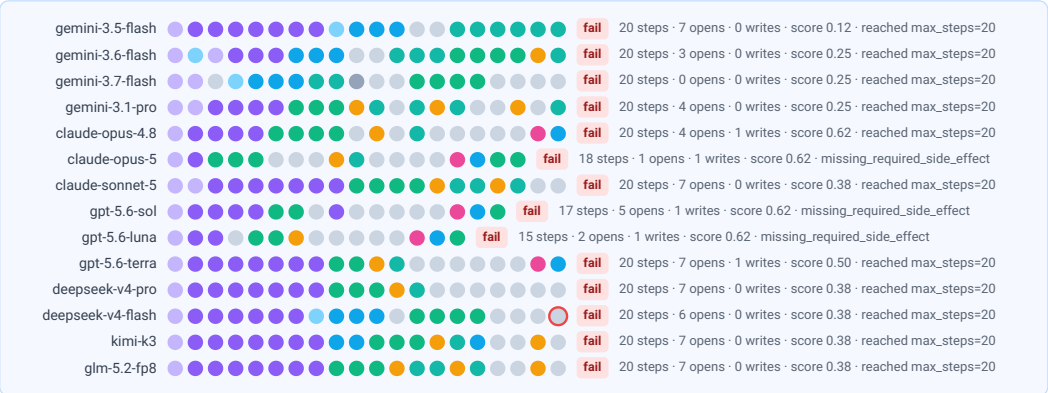



Figure G31: **Gemini 3.5 Flash: Running out of steps due to search loops.** Gets trapped in repetitive search loops and reaches the 20-step limit on **89%** of tasks (highest across all models). Task: `access_request_approval_v01_r01`.

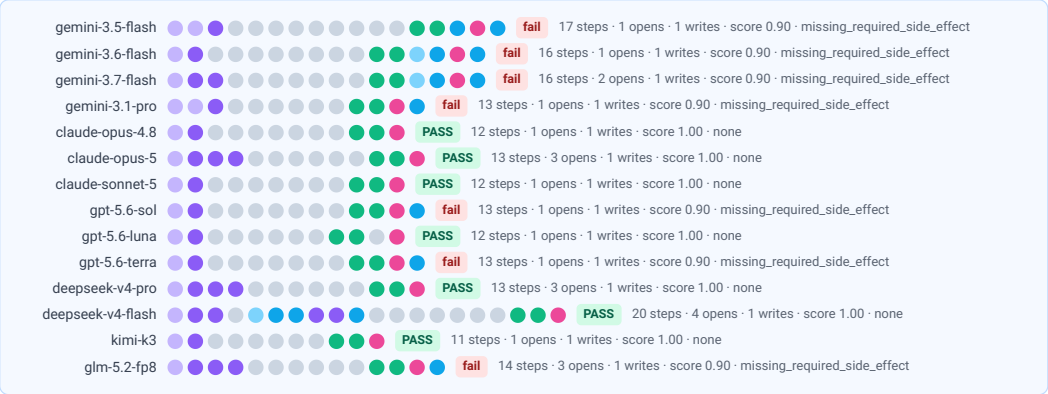



Figure G32: **Gemini 3.6 Flash: Near-miss deliverable submission.** Quickly inspects images and writes reports without looping, but makes small precision errors on required text. Task: `employee_onboarding_workflow_v01_r08`.

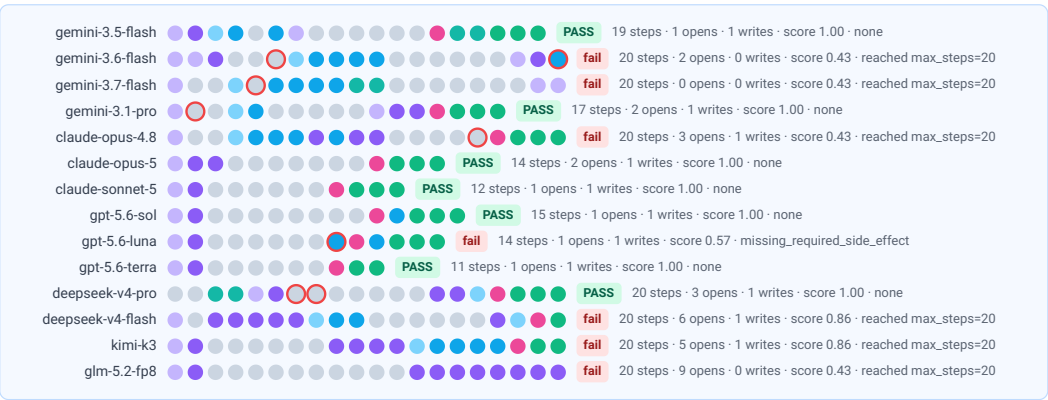



Figure G33: **Gemini 3.7 Flash: Running out of steps by repeatedly checking agent status.** Inspects few images (**2.25** opens/task) while hitting the 20-step limit on **41%** of tasks, spending **3.3%** of all actions polling `agent status`. Task: `access_request_approval_v03_r07`.

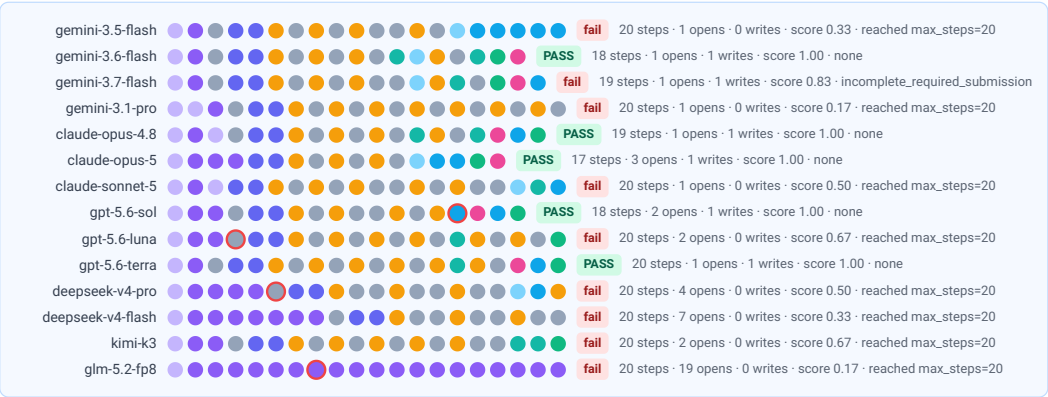



Figure G34: **Gemini 3.1 Pro: Excessive waiting after sub-agent delegation.** Spends **5.1%** of all actions on `wait` commands (highest across all models) after delegating work to sub-agents. Task: `barrier_sync_deployment_v02_r07`.

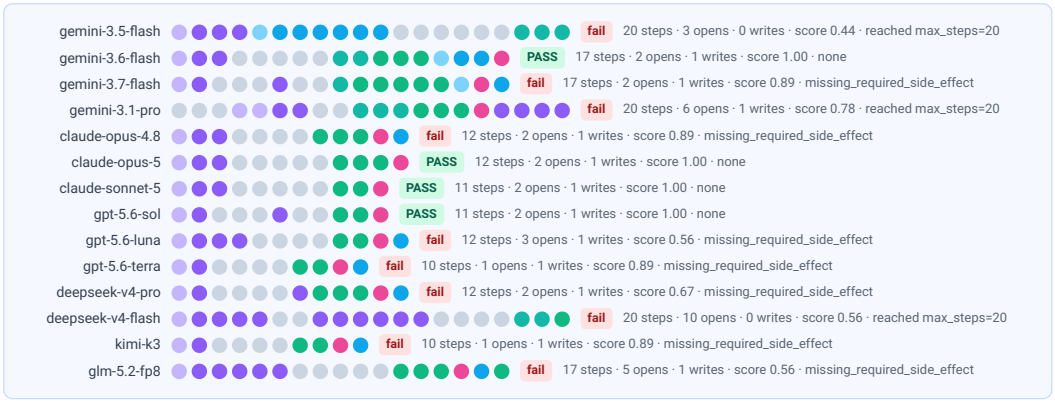



Figure G35: **Claude Opus 4.8: Content precision errors under minimal image inspection.** Writes deliverables after inspecting few visual assets, missing fine-grained verification details. Task: `it_deployment_gate_v03_r05`.

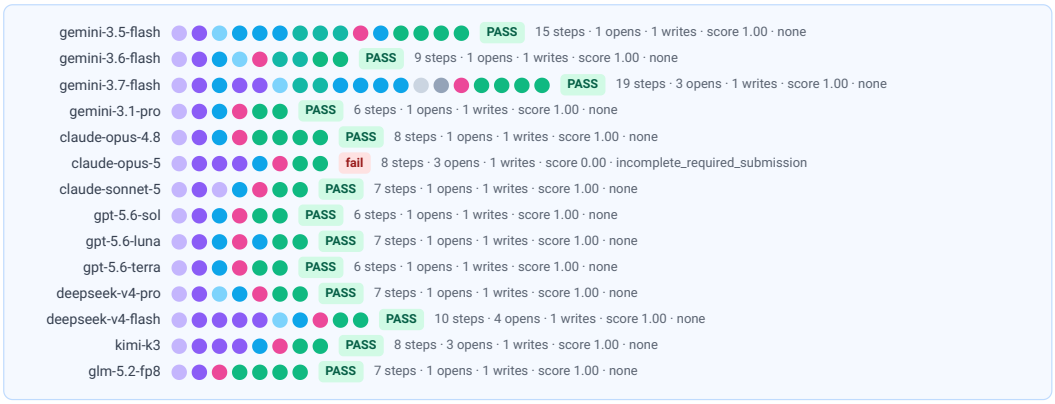



Figure G36: **Claude Opus 5: High-concentration all-or-nothing task resolution.** Shows the highest failure clustering (**1.49**) and overdispersion (**2.23**), passing all checks simultaneously on **25.2%** of tasks versus **10.8%** expected under independence. Task: `it_deployment_gate_v02_r10`.

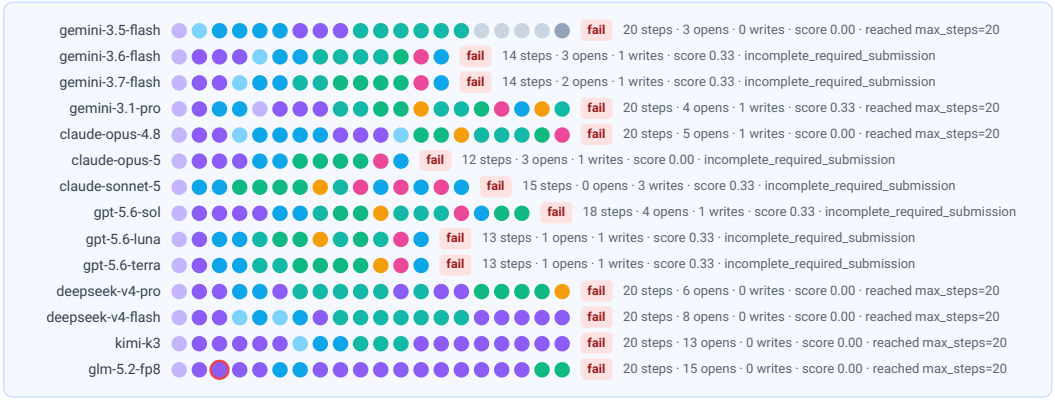



Figure G37: **Claude Sonnet 5: Relying on text alone without inspecting images.** Skips image inspection entirely on **92** of 600 tasks, attempting to write reports from text context alone. Task: `satellite_ground_station_handover_v03_r09`.

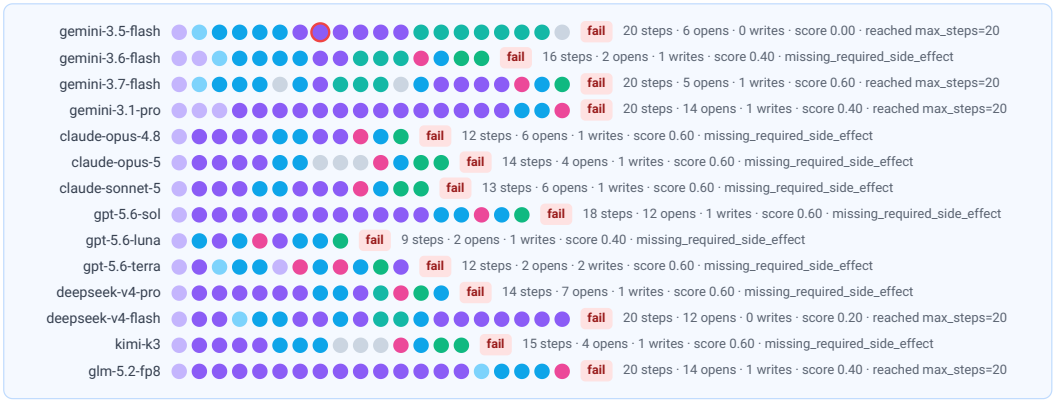



Figure G38: **GPT-5.6 Sol: Thorough visual inspection with text transcription limits.** Inspects visual evidence on **100%** of tasks and writes deliverables on 599/600 rounds, with failures driven by subtle text transcription errors. Task: `multi_tab_slot_monitoring_v02_r05`.

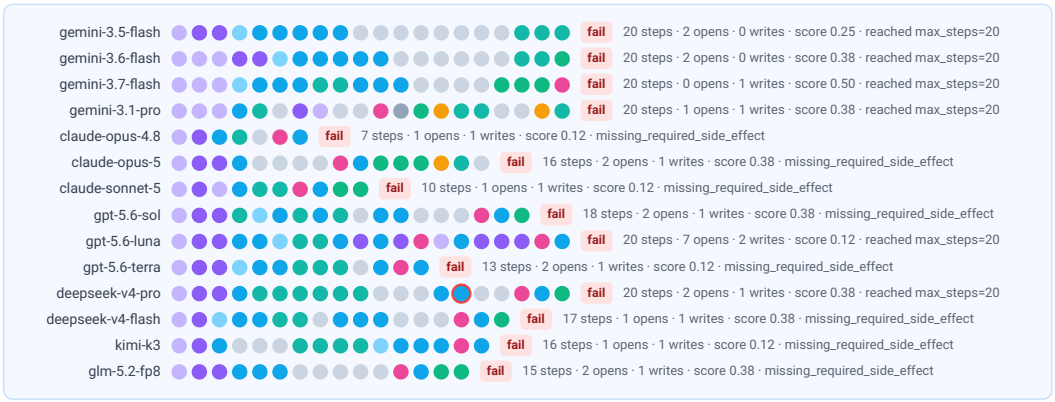



Figure G39: **GPT-5.6 Luna: Re-opening images instead of finishing deliverables.** Repeatedly re-opens already-inspected images rather than writing outputs, leading to more step limit hits than other OpenAI models. Task: `multi_tab_slot_monitoring_v01_r01`.

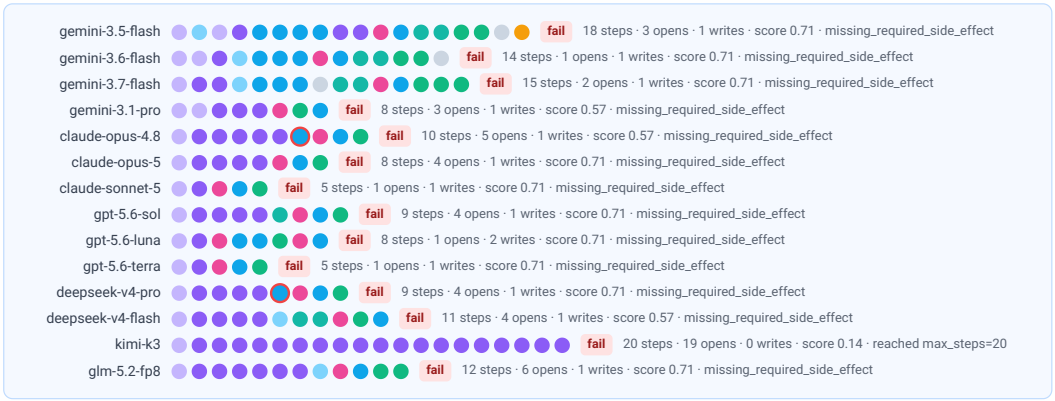



Figure G40: **GPT-5.6 Terra: Stopping too early.** Concludes episodes early without looping, sometimes stopping before completing all required actions or deliverables. Task: `employee_onboarding_workflow_v03_r08`.

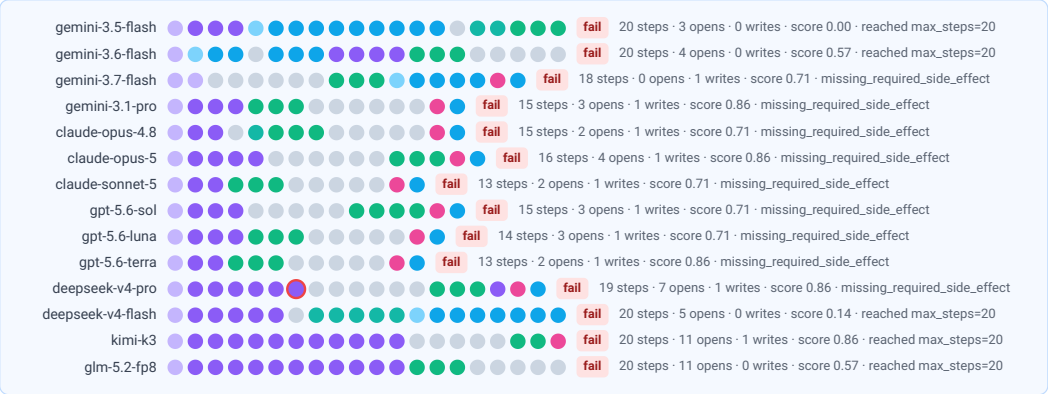



Figure G41: **DeepSeek V4 Pro: Reliable completion among open-weight models.** Finishes and writes deliverables on **92.8%** of tasks, hitting the step limit on only **13.5%** of episodes. Task: `coordinated_containment_v01_r05`.

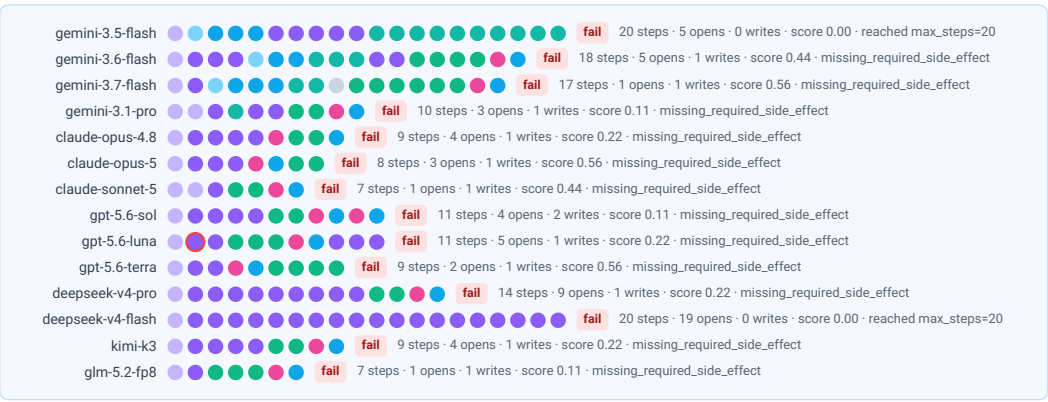



Figure G42: **DeepSeek V4 Flash: Inspecting images without writing deliverables.** Opens **6.00** images per task but fails to write any deliverable on **47.3%** of episodes. Task: `multi_device_pairing_v02_r08`.

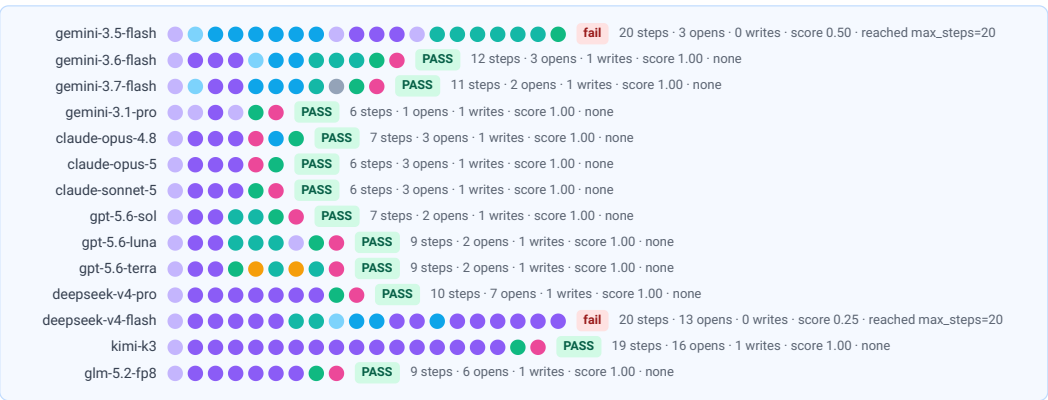



Figure G43: **Kimi K3: Strong visual reasoning and image inspection.** Combines active image inspection (**6.72** opens/task) with high semantic check accuracy, achieving a **21.8%** overall pass rate. Task: `multi_party_access_permit_v03_r07`.




Figure G44: **GLM-5.2 FP8: Heavy image inspection without writing deliverables.** Spends **48.8%** of actions inspecting images (highest share across all models, **7.26** assets/task) but achieves a **14.7%** pass rate because it often fails to write final deliverables. Task: `employee_onboarding_workflow_v03_r09`.

## G.11 Limitations

### G.11.3 Notification checks depend on message body text rather than routing

Table G21: **Notification check results across representative rounds.** Compares sending messages to the right person versus receiving grading credit. Across all rounds, models almost always message the right recipient (10–14 of the 14 baseline models), but scoring credit depends on matching the required body text.

| round | named recipient | route to them | credited |
|---|---|---|---|
| `payment_chargeback_dispute_v01_r04` | `Casey_Support_Specialist` | 14/14 | 3/14 |
| `warehouse_robotics_safety_halt_v02_r01` | `Avery_Safety_Officer` | 12/14 | **1/14** |
| `satellite_ground_station_handover_v02_r01` | `Devon_Ground_Ops_Lead` | 12/14 | 12/14 |
| `clinical_trial_dosage_conflict_v02_r03` | `Avery_Trial_Safety_Monitor` | 13/14 | 13/14 |
| `coordinated_containment_v02_r04` | `Owen_Platform_On_Call` | 11/14 | 13/14 |
| `distributed_lock_shared_resource_v01_r02` | `Casey_Inventory_Controller` | 12/14 | 4/14 |
| `expense_reimbursement_conflict_v02_r05` | `Blair_Audit_Investigator` | 12/14 | 11/14 |
| `expense_reimbursement_conflict_v02_r05` | `Casey_Approver` | 12/14 | 11/14 |
| `expiring_resource_hold_v03_r04` | `Rowan_Research_Lead` | 10/14 | 2/14 |

### G.11.4 EIGHTEEN EXACT-MATCH CHECKS LACK READABLE REFERENCE STRINGS

Table G22: **Pass rates of exact-match string checks.** Compares pass rates on checks where reference strings appear in `visible_refs` ($n = 238$, $75.8\%$ pass) versus checks where they are unexposed ($n = 18$, $0.7\%$ pass).

| exact-match checks | n | this check passes | sibling checks, same rounds | never passed |
|---|---|---|---|---|
| value readable from `visible_refs` | 238 | 75.8% | 67.6% | 17 |
| value **not** readable | 18 | **0.7%** | 37.7% | **15** |

## G.12 WHAT EACH MODEL WOULD NEED

Table G23: **Primary bottlenecks and diagnostic evidence for each evaluated model.** Summarizes the main bottlenecks limiting task completion across models, ordered by High-effort pass rate.

| Model | binding constraint | evidence |
|---|---|---|
| GPT-6 Astra | Multi-agent closing protocols and shared state cleanup. Leads the benchmark with **28.7%** pass rate, **66.8%** mean score, and **98.8%** visual gate compliance (averaging 26.1 steps/task). Main failures stem from terminating prematurely before completing closing protocols: incomplete deliverable submissions (**199** tasks) and missing environment side effects (**220** tasks), hitting its step limit on only **6** tasks. | §G.1, §G.5, §G.7.1, §G.7.3 |
| Claude Opus 5 | Executing environment actions, not more reasoning. Achieves strong per-check accuracy (63.0%) and high failure clustering (lift **1.49**), yet **239** episodes end without required environment actions while only **3** hit the step limit. **140** rounds finish exactly one check short (47 on missing environment actions). Higher reasoning effort provides negligible gain (+1.7pp, n.s.). | §G.5, §G.7.1, §G.9.2 |
| GPT-5.6 Sol | Search efficiency and deliverable accuracy. Issues **877** `message search` calls and opens **4.08** images/task, reaching a **91.2%** visual gate rate but lower text transcription accuracy (**75.8%** vs. Claude Opus 5's 82.4%). Shows steady gains from higher reasoning effort (+3.7pp, $p = 0.0016$). | §G.3, §G.7.3, §G.9.2 |
| Claude Opus 4.8 | High token cost and compute efficiency. Highest cost per solved task among frontier models (**$21.41**, and **$4.89**/task) with a 2.4pp lower pass rate than Claude Opus 5, driven by **4.1**× higher reasoning token usage (2,291 vs. 553 tokens/task). Higher reasoning effort yields no significant improvement (+1.5pp, n.s.). | §G.1, §G.9.1, §G.9.2 |

*continued on the next page*

*Table G23 (continued)*

| Model | binding constraint | evidence |
|---|---|---|
| Kimi K3 | Writing deliverables before timing out. Achieves strong semantic accuracy (**65.7%** on `field_semantic_match` and **42.6%** on `other`), but **437** semantic checks are never tested because the model wrote no files on **128** rounds and hit the step limit on **170** rounds. | §G.7.2, §G.7.3, §G.5 |
| Claude Sonnet 5 | Opening and inspecting visual assets. Opens the fewest images (**1.83**/task), skips images entirely on **92** tasks, has the lowest visual gate rate (**72.6%**), and fails **76** image checks. When it does open images, transcription accuracy is high (**76.6%**), and higher reasoning effort yields the largest gain in the field (**+7.0pp**, $p = 1.0 \times 10^{-7}$). | §G.7.2, §G.7.3, §G.9.2 |
| Gemini 3.7 Flash | Running out of steps. Hits the 20-step limit on **244** rounds (**40.7%**, highest among competitive models) with a median of 19 steps. Higher reasoning effort increases limit exhaustion ($3.0\% \rightarrow 12.7\% \rightarrow 40.7\%$) with no accuracy gain ($-0.2$pp). | §G.5, §G.3, §G.9.2 |
| GPT-5.6 Terra | Stopping too early. Has the highest rates of missing environment actions (**261**) and incomplete deliverables (**210**), while hitting the step limit on only **8** rounds (mean 11.4 steps). Higher reasoning effort yields modest gains (+2.3pp, $p = 0.049$). | §G.5, §G.6 |
| GPT-5.6 Luna | High latency and slow execution. Has low financial cost ($0.14/task, $0.72/solved), but very slow runtimes (**620.5 s**/task) and **6** stalls in repetitive loops. | §G.1, §G.9.2, §G.5 |
| DeepSeek V4 Pro | Reasoning about final verdicts. Operates at the lowest cost ($0.05/task, $0.27/solved) with balanced error types (218 missing actions, 182 incomplete deliverables, 81 step-limited), but scores lower on final verdict checks (`other` at **33.2%**). | §G.1, §G.7.2 |
| Gemini 3.6 Flash | Outperformed by newer models. Gemini 3.7 Flash achieves a higher pass rate (20.3% vs. **18.5%**) at half the cost per solved task ($2.51 vs. **$5.09**). Higher reasoning effort yields no accuracy gain (+0.0pp, $p = 1.0$) while causing more step limit hits ($1.8\% \rightarrow 14.5\% \rightarrow 24.7\%$). | §G.1, §G.9.2 |
| GLM-5.2 FP8 | Turning visual evidence into written deliverables. Opens **7.26** images/task with **93.2%** visual gate compliance, but writes no files on **157** rounds and achieves only **51.6%** text transcription accuracy (a **41.6pp** gap). | §G.3, §G.7.3 |
| Gemini 3.1 Pro | Notification content precision and excessive waiting. Attains **46.4%** semantic match accuracy and fails **831** message content checks, spending **402** actions on `wait` (highest across all models). | §G.7.1, §G.7.2, §G.3 |

*continued on the next page*

*Table G23 (continued)*

| Model | binding constraint | evidence |
|---|---|---|
| DeepSeek V4 Flash | Running out of steps and omitting deliverables. Ends **358** rounds (**59.7%**) at the 20-step limit and writes no deliverables on **284** rounds (0.58 writes/task). | §G.5, §G.3, §G.7.3 |
| Gemini 3.5 Flash | Trapped in search loops. Executes **2,707** search actions (4.51/task) and writes files only **0.31** times/task with **440** zero-write episodes, causing **535** step limit hits (**89.2%**). High reasoning effort worsens search loops ($-10.5$pp, $p = 2.2 \times 10^{-19}$), whereas Low effort recovers a **13.5%** pass rate. | §G.3, §G.6, §G.7.1, §G.9.2 |