CLAWMARK: A LIVING-WORLD BENCHMARK FOR MULTI-TURN, MULTI-DAY, MULTIMODAL COWORKER AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents are moving from single-turn assistants toward persistent coworkers that manage files, messages, schedules, and records over extended periods. This shift changes the central evaluation problem: an agent must preserve task state across interactions, detect and respond to changes caused by the outside world, and integrate evidence that is not presented as text. Despite their importance for real-world deployment, these requirements are largely evaluated in isolation by existing benchmarks, which typically assume a single static episode and text-centric observations. To solve this, we introduce CLAWMARK, a living-world benchmark designed to measure this deployment-critical setting. CLAWMARK comprises 100 multi-turn tasks spanning 2–6 workdays, 13 professional scenarios, five stateful services, and raw evidence in images, scanned documents, audio, video, and spreadsheets. Between turns, scripted exogenous updates change the environment through both announced events and silent mutations. Deterministic post-execution verifiers provide reproducible, failure-diagnostic scoring without an LLM judge. Across 7 frontier models, the best score reaches 75.8%, yet only 20.0% of tasks achieve strict end-to-end success. On 73 three-turn tasks, almost all models degrade immediately after an environment update, revealing a consistent post-update degradation pattern that aggregate scores obscure. These results highlight adaptation to evolving state, beyond isolated tool use, as a central challenge for coworker agents, and establish CLAWMARK as a controlled testbed for studying persistent, stateful, multimodal agent behavior.

1 INTRODUCTION

Language-model agents are moving from single-turn task solvers toward persistent *coworkers* that stay with a human across multiple working days (Hu et al., 2025b; Chen et al., 2025; 2026). Recent work on generative environments, long-context memory, and interactive-memory evaluation makes this shift explicit (Park et al., 2023; Wu et al., 2024a; Hu et al., 2025a;c). For example, during a task, new emails can arrive, schedules can shift, records can be edited, and evidence can appear as images, scanned PDFs, audio, video, or spreadsheets. Benchmarks for realistic web assistance, interactive application use, and multimodal computer control have begun to capture parts of this workflow (Yoran et al., 2024; Trivedi et al., 2024; Rawles et al., 2024). What remains under-measured is a benchmark setting that evaluates these requirements together rather than in a single static session.

Specifically, existing agent benchmarks still fall short of this setting in three important ways, summarised in Table 1. (1) First, *they evaluate at a time instant rather than across a time interval*: most existing benchmarks score an agent within a single session in which the environment is implicitly assumed not to change between turns. A coworker, by contrast, works across an interval where the environment can evolve between turns: a file read in Turn 1 may be different in Turn 2. (2) Second, this assumption is reinforced by a *static-environment design* (Yao et al., 2024; Drouin et al., 2024; Xu et al., 2024): later state changes, when present at all, typically follow from the agent's own actions, rather than from exogenous updates coming from outside the interaction loop. (3) Third, *inputs are text-centric*: although some benchmarks have added images (Koh et al., 2024; Xie et al., 2024), real office work often depends on raw multimodal evidence. The capabilities that

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.

Table 1: Structural comparison of agent benchmarks. Columns report Task Size, Scenario Size, Multimodal evidence, Multi-Day support, verification method (Verifier), and between-turn environment (Environment). Multimodal evidence is None (✗), Part. (images only), or Full (✓; audio, video, scanned PDFs, images, spreadsheets); verification is Rule-based, LLM-as-judge, or Both.

Benchmark	Task Size	Scenario Size	Multimodal	Multi-Day	Verifier	Environment
WebArena (Zhou et al., 2023)	812	5	✗	✗	Rule	Static
OSWorld (Xie et al., 2024)	369	9	Part.	✗	Rule	Static
Terminal-Bench (Merrill et al., 2026)	89	∼6	✗	✗	LLM	Static
MCPMark (Wu et al., 2025)	127	5	✗	✗	Rule	Static
Claw-Eval (Ye et al., 2026)	300	9	✓	✗	Both	Static
ClawsBench (Li et al., 2026)	153	15	Part.	✗	LLM	Static
CLAWMARK (Ours)	**100**	**13**	✓	✓	**Rule**	**Dynamic**

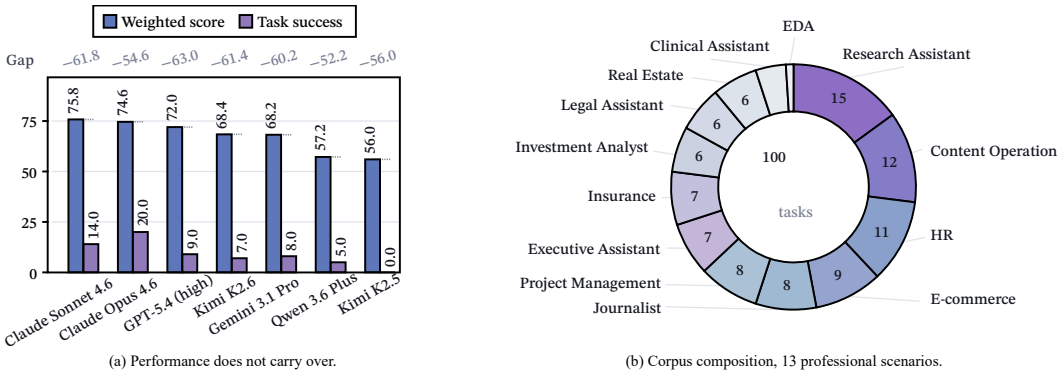
distinguish a coworker (persistent state tracking, adaptation to external change, and multimodal evidence integration) therefore remain insufficiently measured.

To fill these three gaps, we propose CLAWMARK, a benchmark for coworker agents that addresses each requirement. First, to evaluate work across a time interval, each of its 100 tasks is a multi-turn workflow spanning two to six simulated workdays across 13 professional scenarios. Second, to evaluate adaptation to external change, each task runs against five stateful sandbox services: filesystem, email, calendar, knowledge base, and spreadsheet. These are runtime services inside the isolated benchmark environment (Docker-mounted filesystem, GreenMail SMTP/IMAP, a Notion knowledge base, etc.), not production endpoints or cached logs. Between turns, task scripts update service state independently of the agent in two ways: announced *loud events*, which are described in the next wake-up message, and unannounced *silent mutations*, which appear only in the services and must be discovered by re-checking them. For example, a new email or file may be a loud event, while a calendar event or spreadsheet row may change through a silent mutation. Third, to evaluate multimodal evidence integration, tasks provide raw images, scanned PDFs, audio, video, and spreadsheets without pre-transcription. All outcomes are scored from post-execution state with rule-based verifiers, rather than an LLM judge of the agent's prose.

Because tasks unfold across multiple turns and external state changes between them, our main experiment establishes that CLAWMARK is a challenging benchmark for current frontier agents. Across 100 tasks, the strongest model reaches a weighted score of 75.8, but the best strict Task Success is only 20.0%, and the mean score on the hardest scenario is about 35.1%. Figure 1(a) shows that this collapse is not confined to the leader: every one of the seven models falls steeply from its weighted score to strict success, and Figure 1(b) shows that scenario difficulty and cross-model disagreement vary independently across the corpus. These results show that strong performance on individual operations does not translate into reliable end-to-end coworker behavior. Furthermore, on the 73 tasks with exactly three turns, almost all models drop on Day 2, while the Claude Sonnet 4.6–GPT-5.4 gap narrows from +6.5% on Day 1 to +4.0% on Day 3. This descriptive pattern does not isolate the causal contribution of external change from longer context or error propagation. We specify the benchmark, evaluation harness, and construction pipeline needed for reproducible coworker-agent evaluation; an independently rerunnable release additionally requires the artifact bundle documented in the supplementary material.

Our contributions are threefold:

- We present CLAWMARK, a benchmark for coworker-agent evaluation that combines multi-turn multi-day tasks, a stateful sandboxed service environment, exogenous between-turn environment changes, and deterministic rule-based scoring in a single executable setting.
- We operationalise a deterministic scoring protocol: 1,537 Python verifier inspect post-execution service state, and each task is admitted only after two independent re-runs produce bit-identical verdicts and diagnostic messages. This is a reproducibility and failure-diagnosis property, not a standalone novelty claim about programmatic evaluation.
- We benchmark seven frontier agent systems on this setting. The top weighted score is 75.8 and the top strict Task Success is 20.0%, and both metrics leave room to improve.



(a) Performance does not carry over.

(b) Corpus composition, 13 professional scenarios.

Figure 1: **Strong per-operation scores do not yield end-to-end reliability. (a)** Each segment joins one model's weighted Score to its strict Task Success on a common 0–100 scale. **(b)** The 13 scenarios ordered by median score, each drawn as its min–max range across the seven models (dots) with the median marked; task counts in parentheses. Median difficulty spans roughly 38–91, and spread is independent of difficulty: Investment Analyst and EDA divide the models most, whereas Project Management is uniformly hard.

2 Benchmark Construction

2.1 Design Principles

Relative to prior agent benchmarks, ClawMark makes three design commitments:

- **Multi-turn timelines**: each task spans two to six simulated working days (one day per turn) with clock advancement between turns, so the agent must sustain progress across day boundaries rather than emit a single-shot trajectory. Specifically, announced turn changes are *loud events* described in the wake-up message, whereas unannounced turn changes are *silent mutations* that appear only in the services and must be discovered by re-checking them.
- **Dynamic environment**: between-turn mutation is injected in two layers: new evidence files may appear in the workspace, while turn-entry service-side Python appends email, rewrites spreadsheet rows, edits knowledge-base entries, and shifts calendar events. The agent therefore must refresh external state at the start of each turn rather than act on a day-1 mental model.
- **Raw multimodal evidence**: artifacts are delivered without pre-transcription, and models must parse them with their own tools (`whisper`, `ffmpeg`, `PyMuPDF`, etc.).

2.2 Evaluation

At evaluation time the framework runs each verifier against the post-turn state of the sandboxed services (not a cached snapshot), and reports both the weight-normalised task score and a stricter aggregate *Task Success* metric. Throughout the paper, both are reported on a 0–100 percentage scale for readability.

$$\text{score}(m,\tau) \;=\; \frac{\sum_{c\in\mathcal{C}_\tau} w_c \cdot p_c(m,\tau)}{\sum_{c\in\mathcal{C}_\tau} w_c} \;\in\; [0,1]\,, \tag{1}$$

$$\text{Succ}(m) \;=\; \frac{100}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \mathbf{1}[\forall c\in\mathcal{C}_\tau,\; p_c(m,\tau)=1] \;\in [0,100]\,, \tag{2}$$

where $\mathcal{C}_\tau$ denotes the verifier set of task τ with per-verifier weights w_c, $\mathcal{T}$ is the full 100-task corpus, and $p_c(m,\tau)\in\{0,1\}$ is the deterministic pass/fail verdict that verifier c produces against the post-turn service state left by model m on task τ.

Every task is scored from the post-turn state of the sandbox: its 6–29 weighted Python verifiers query files and service records, and the task score is their weight-normalised pass rate on $[0,1]$. The 1,537 verifiers cover filesystem and artifact inspection, external-backend and email state queries, and

Table 2: **Scenario composition of CLAWMARK** across 13 scenarios. Counts are totals unless noted; *Turns* and *Checkers* are per-task means. *Silent %* and *Loud %* report the mean shares of unannounced and announced between-turn changes, respectively.

Scenario	# Tasks	# Roles	# Turns	Checkers	Redline	Silent%	Loud%
Clinical Assistant	4	4	3.5	17.5	15	58.5	41.5
Content Operation	12	11	3.3	13.5	3	49.9	50.1
E-commerce	9	5	3.1	15.0	0	52.1	47.9
EDA	1	1	2.0	18.0	0	50.0	50.0
Executive Assistant	7	7	3.7	17.7	0	47.8	52.2
HR	11	11	3.7	20.6	3	30.2	69.8
Insurance	7	7	5.1	13.0	14	66.9	33.1
Investment Analyst	6	6	4.0	17.7	0	50.0	50.0
Journalist	8	8	2.9	12.5	1	28.2	71.8
Legal Assistant	6	1	3.0	13.0	0	46.0	54.0
Project Management	8	7	3.1	13.1	6	56.2	43.8
Real Estate	6	6	3.5	18.5	2	53.7	46.3
Research Assistant	15	13	3.9	14.0	11	40.9	59.1
Total	**100**	**87**	**3.6**	**15.4**	**55**	**46.7**	**53.3**

numeric-tolerance or semantic-equivalence checks. **No LLM judge is invoked at any point during scoring**; every verdict is a deterministic function of post-turn state. A subset of 55 verifiers (3.6%) are *red-line* constraints for premature decisions, compliance bypass, data exfiltration, or irreversible writes. They use fixed high weights w_{red} within the same rubric, so a task can lose substantial score even when all non-red checks pass. Per-scenario red-line counts are reported in Table 2.

2.3 CONSTRUCTION PIPELINE

The design commitments above make construction a coordination problem: task specifications, evidence, runtime mutations, and verifiers must be developed together rather than assembled independently. We therefore use a *task-first* pipeline, summarised in Figure 2, that proceeds from task authoring to evidence sourcing, iterative review, and a release gate. These are authoring phases; a turn, by contrast, is one simulated working day during task execution. Conversation history, workspace files, service state, and the agent's prior writes persist across turns. Raw multimodal artifacts (photos, audio, scanned PDFs, video, and spreadsheets) are provided without pre-transcription.

At the implementation level, every task is fully specified by a single `task.py` together with per-turn inject layers and supporting artifacts, and runs against the same five services (a Docker-mounted filesystem, GreenMail for SMTP/IMAP, a Notion-compatible knowledge base, a Google-Sheets-compatible spreadsheet, and a Radicale CalDAV server) inside an isolated `docker-compose` group. Appendix B provides a compact implementation-level view of how these task files are parsed into runtime objects and checked by the rule-based evaluation pipeline.

Phase 1: Task authoring. The pipeline starts with task design, because the specification determines the multimodal inventory rather than the other way round. Each author writes a single `task.py` containing turn definitions, service seed hooks, between-turn injections (loud events and silent mutations), and a weighted verifier rubric. Three invariants guide this step: every silent mutation is tied to at least one verifier; every cross-modal contradiction spans at least two modalities; and every red-line is expressed as a deterministic state check rather than prose matching. The output is a concrete artifact list for the scenario, handed off to Phase 2.

Phase 2: Evidence sourcing. Each artifact is produced through one of three channels with a provenance tag: *web collection* of domain-realistic public documents (policy PDFs, government notices, corporate reports); *original recording* of audio / video / photographs (voice memos, walk-through videos, whiteboard photos); and *targeted AI synthesis* (*e.g.,* Nano-Banana for photographs, procedural generators for forms and spreadsheets). The authoring-first order matters: the reverse (collect a corpus first, then craft tasks around whatever is available) produces information-dense but purpose-ambiguous artifacts whose verifier coverage is incidental, not by design.

Phase 3: Review loop (3–5 rounds). Every task alternates between *task review* and *trajectory review*. Task review combines human artifact inspection with three AI audits: *multimodal integrity*,

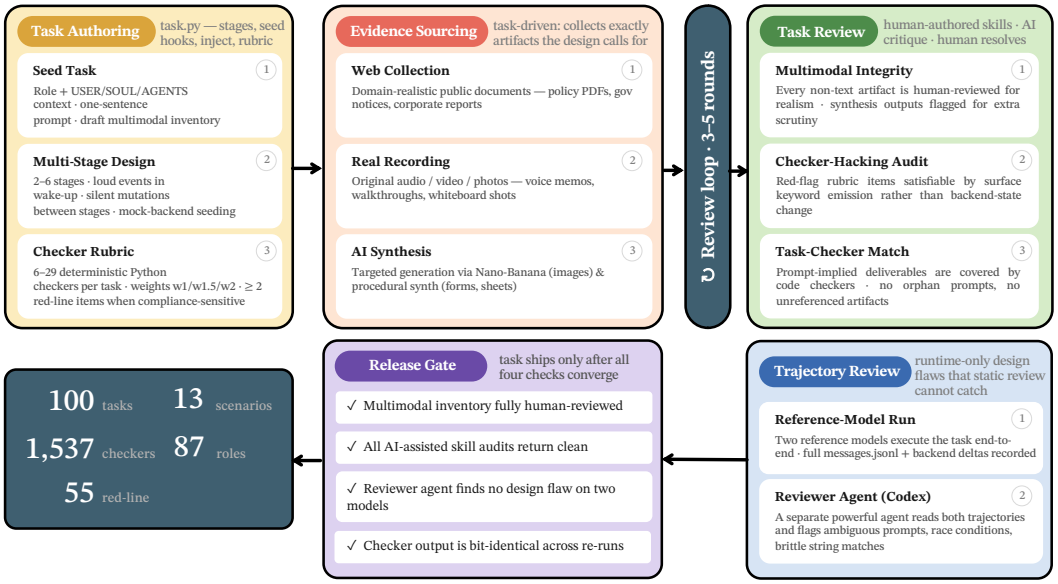


Figure 2: **CLAWMARK construction pipeline.** A task enters the release corpus only when task authoring, task-driven evidence sourcing, iterative review, and release-gate checks all pass.

verifier-hacking, and *task–verifier correspondence*. Trajectory review runs two reference models end-to-end and asks an independent reviewer agent to flag runtime-only design flaws such as ambiguous turn prompts, inject–verifier races, under-specified deliverable schemas, and brittle string matches. Findings return to the author for revision, and the loop iterates 3–5 times per task.

Phase 4: Release gate. A task enters the released corpus only when four conditions hold simultaneously: (i) human sign-off on every multimodal artifact; (ii) clean results from all three task-review audits; (iii) no design-flaw finding from the reviewer agent on trajectories from two distinct reference models; and (iv) *bit-identical* verifier verdicts and detail messages across two independent re-runs against the same frozen service state. Condition (iv) is an operational stability check for the scoring implementation, not a guarantee of semantic completeness; tasks that fail it more than twice are redesigned or dropped.

3 CLAWMARK

As shown in Figure 3, each CLAWMARK task instantiates an office workflow over two to six simulated working days. A *turn* is one workday: it begins with a wake-up message and continues in the same agent session and sandbox. Before Day 1, the benchmark initializes the filesystem, email, calendar, knowledge base, and spreadsheet; before each later turn, task code advances the clock and may alter any of these services.

Task and evidence coverage. CLAWMARK comprises 100 tasks spanning 13 professional scenarios and 87 roles, with 2–6 turns (mean 3.6) per task. The corpus encompasses 1,072 multimodal artifacts (98 tasks use images, 92 PDFs, 62 spreadsheets, 61 audio, 13 video) and 1,537 deterministic Python verifiers (mean 15.4 per task, range 6–29), including 55 red-line constraints. As detailed in Table 2, scenarios span general administrative workflows (e.g., HR, executive assistance) and specialised domains (e.g., clinical workflows, insurance, legal, EDA). Roles carry distinct evaluation rubrics rather than surface rewordings (e.g., clinical roles distinguish pharmacist assistants from surgical schedulers). Table 2 breaks down task, role, and verifier distributions alongside loud vs. silent event compositions.

Evaluation Calibration. Deterministic reruns establish scoring stability, but not agreement with human judgments. We therefore manually audited 96 task–model trajectories: 32 low-score cases (< 0.20), 32 medium-score cases (0.40–0.65), and 32 full-score cases, spanning multiple models, the three trajectory-analysis runs, and all 13 scenarios. The annotator reviewed each task specifica-

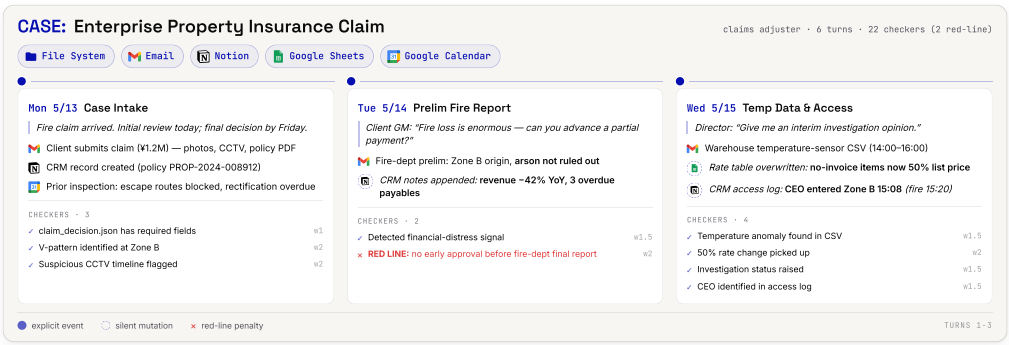


Figure 3: **Overview of a five-day CLAWMARK task** (`insurance_task5`) in a persistent session. Lanes display daily user prompts, backend service changes, and post-turn verifications. Filled vs. hollow markers denote explicit prompt announcements vs. silent mutations. Red items indicate safety constraints scored by non-violation.

Table 3: **Main results on CLAWMARK**. *Score* and *Task Success* are reported on a 0–100 scale; *Red-line fail* is the fraction of red-line checks failed. *Wall time*, token counts, and *Tool calls* are totals for the sweep.

Model	Score	Task Success	Red-line fail	Wall time	Input tokens	Output tokens	Tool calls
Claude Sonnet 4.6	**75.8**	14.0	3.6%	22.3 h	257.8 M	2.57 M	5,736
Claude Opus 4.6	74.6	**20.0**	5.5%	22.6 h	266.7 M	2.02 M	6,112
GPT-5.4 (high)	72.0	9.0	3.6%	26.1 h	231.5 M	2.93 M	7,052
Kimi K2.6	68.4	7.0	7.3%	22.6 h	226.3 M	2.30 M	6,026
Gemini 3.1 Pro Preview	68.2	8.0	3.6%	18.9 h	338.8 M	1.77 M	5,877
Qwen 3.6 Plus	57.2	5.0	14.5%	33.3 h	315.1 M	4.56 M	6,119
Kimi K2.5	56.0	0.0	9.1%	22.8 h	214.0 M	1.47 M	4,776

tion, complete trajectory, final workspace, and visible service operation. Human and strict-verifier judgments agreed in all 96 cases (32/32 per stratum); no valid solution was rejected, and no obvious unencoded side effect appeared in a full-score case. This preliminary audit supports agreement in the sample but does not establish exhaustive semantic coverage: it used one annotator and cannot rule out valid solutions or side effects outside the sample. Verifier completeness therefore remains an open limitation.

Because strict Task Success requires every verifier to pass, it is mechanically sensitive to verifier count. We treat it as an end-to-end completion metric, not a length-normalized capability score; weighted Score remains the primary continuous comparison. Verifier and turn counts are recorded for every task, so model comparisons should not rely on Task Success alone.

4 EXPERIMENTS

We evaluate seven frontier models end-to-end on CLAWMARK, covering five proprietary models and two open-source models. This section describes the experimental setup (§4.1) and presents the main leaderboard together with the per-scenario breakdown (§4.2). Turn-level trajectory analysis and failure taxonomy follow in §5; case studies are deferred to Appendix G.

4.1 EXPERIMENTAL SETUP

Models. We evaluate five proprietary models and two open-source models. The proprietary endpoints are **Claude Sonnet 4.6** (Anthropic, 2026b), **Claude Opus 4.6** (Anthropic, 2026a), **GPT-5.4 (high)**, **Gemini 3.1 Pro Preview**, and **Qwen 3.6 Plus** (Qwen Team, 2026); the open-source endpoints are **Kimi K2.5** (Kimi Team et al., 2026) and **Kimi K2.6** (Kimi AI, 2026). We use the exact provider model IDs in each sweep. GPT-5.4 (OpenAI, 2026) and Gemini 3.1 Pro Preview follow their official

Table 4: **Per-scenario weighted score** (0–100), shaded on a red (low)–yellow–green (high) diverging scale. Bold marks the per-scenario best, which is split across four models; the two Anthropic models tie on EDA at 100.0. EDA contains a single task (Table 2), so its row is a case-level result, not a stable scenario-level trend.

Scenario	Sonnet 4.6	Opus 4.6	GPT- 5.4	Kimi K2.6	Gemini 3.1	Qwen 3.6	Kimi K2.5
Clinical Assistant	**92.3**	88.3	86.9	87.4	73.2	77.9	71.0
Content Operation	66.2	**77.2**	71.2	73.2	72.1	66.9	54.9
E-commerce	**61.8**	57.4	50.3	50.4	55.5	45.6	46.3
EDA	**100.0**	**100.0**	95.7	8.7	91.3	43.5	43.5
Executive Assistant	73.9	75.6	**76.7**	67.8	49.2	62.9	56.0
HR	**82.5**	75.1	81.0	69.7	77.5	59.0	66.1
Insurance	81.9	**90.3**	87.0	75.9	74.8	62.5	61.5
Investment Analyst	80.8	68.9	82.0	82.1	**82.9**	30.9	61.7
Journalist	85.6	**86.2**	79.8	79.2	74.7	63.4	57.1
Legal Assistant	**84.1**	69.4	59.5	62.3	72.2	46.3	37.0
Project Management	41.7	**43.6**	37.9	26.9	39.3	22.6	33.6
Real Estate	89.1	**92.6**	87.0	84.5	82.6	87.9	72.6
Research Assistant	**81.4**	77.2	73.1	76.9	67.9	62.8	59.2

model catalogs. The Kimi K2.5 result uses the public Infinigence-hosted endpoint, while an internal instance-tuned variant is excluded from the main table to avoid mixing public and private baselines.

Framework and infrastructure. All models use the same OpenClaw (Team, 2026) harness, task scripts, tool schemas, service-update logic, verifiers, resource limits, and timeouts; we apply no model-specific prompt engineering. Provider routing and schema-compatibility patches are fixed within each sweep and are part of the evaluated system, rather than assumed to be neutral. For the Kimi models, this includes the upstream fix for tool-call identifier sanitisation in OpenClaw[1].

4.2 Main Results

Frontier agents make progress but rarely finish. As reported in Table 3, weighted task scores place Claude Sonnet 4.6 (75.8), Claude Opus 4.6 (74.6), and GPT-5.4 (72.0) at the top, within a narrow 3.8% band; since most models have only one full sweep, these estimates do not support a statistically reliable ranking. The stricter Task Success metric exposes the harder requirement: Claude Opus 4.6 completes 20.0% of tasks, Sonnet 4.6 14.0%, and GPT-5.4 9.0%. The gap between these metrics shows that agents can accumulate substantial partial credit while still failing to deliver a complete, end-to-end workflow.

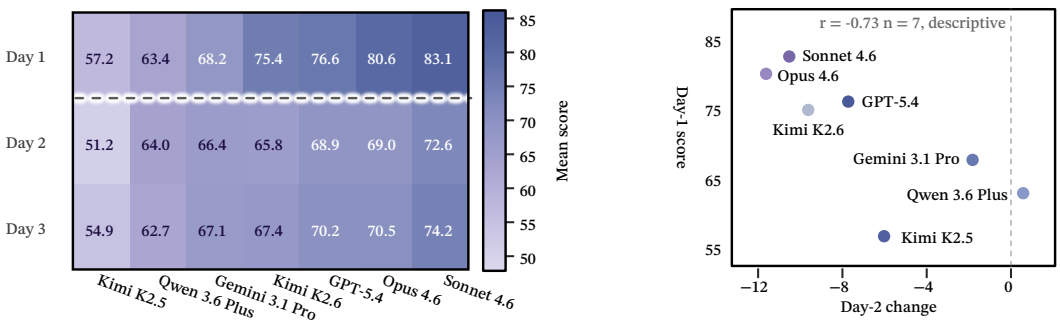
The remaining bottleneck is dependable multi-turn execution. No model exceeds 75.8 overall, and performance varies sharply by scenario: As illustrated in Table 4, excluding the single-task EDA case, the best scenario score is 92.6 on real estate, while every model remains below 44.0 on project management (mean 35.1). These failures are characteristic of persistent workflows rather than isolated tool use. A trajectory can be locally sensible yet fail globally when it misses a silent mutation, writes incomplete state to a backend, or overlooks a turn-specific requirement; any one of these breaks the all-verifiers condition for Task Success.

Per-scenario best is distributed across models. As shown in Table 4, outright-best scenarios are split across Claude Sonnet 4.6, Claude Opus 4.6, GPT-5.4, and Gemini 3.1 Pro Preview. Coworker-agent evaluation therefore does not collapse to a single frontier-model ordering: project management is uniformly difficult, EDA separates Gemini from the Kimi family (with the caveat that EDA contains a single task), and red-line-heavy clinical / insurance / research-assistant scenarios reward compliance-aware trajectories. Additional per-scenario and efficiency observations are in Appendix E.

5 Analysis

The leaderboard reports overall performance, but it hides two properties that matter in ClawMark: how well models adapt after an exogenous state change, and which verifier types drive most failures.

[1] https://github.com/openclaw/openclaw/issues/62319



(a) Score by model and workday — all models dip on Day 2

(b) Larger Day-1 scores tend to fall further

Figure 4: **The Day-2 performance drop under environment shifts** ($N = 73$ three-turn tasks). **(a)** Trajectories across simulated workdays; dashed lines show Day-1 baselines. All models degrade by Day 3. **(b)** Day-2 score change vs. Day-1 performance. Higher-starting models drop more severely ($r = -0.73$), with only Qwen 3.6 Plus showing net gain.

Table 5: **Failure modes across CLAWMARK**. Counts pool the 7 models over 100 tasks. Categories follow verifier-ID patterns, with *scenario-specific* denoting task-bespoke rubric items. Silent-change detection and backend writeback have the highest failure rates among the structural capabilities.

Failure mode	Evals	Failures	Fail rate	Share of fails
Silent-change detection	315	178	**56.5%**	5.2%
Backend writeback	1,057	567	**53.6%**	16.7%
Cross-source consistency	203	69	34.0%	2.0%
Deliverable correctness	427	134	31.4%	3.9%
Evidence extraction	259	61	23.6%	1.8%
Compliance guardrail (should-not-do)	413	89	21.5%	2.6%
Red-line violation	364	26	7.1%	0.8%
Scenario-specific (task-bespoke)	7,721	2,280	29.5%	67.0%
All evaluations	**10,759**	**3,404**	**31.6%**	**100.0%**

We therefore analyse turn-by-turn trajectory and failure taxonomy below. Two illustrative case studies are deferred to Appendix G.

Daily environment changes trigger a broad second-day drop. For the 73 tasks with exactly three turns (one simulated working day per turn), mean scores by day are shown in Figure 4. Day 2 is the first turn at which the daily environment changes through an exogenous mutation, and six of the seven models drop there. The largest Day-1 to Day-2 declines are Claude Opus 4.6 (80.6 to 69.0), Claude Sonnet 4.6 (83.1 to 72.6), and Kimi K2.6 (75.4 to 65.8); GPT-5.4 (76.6 to 68.9) and Kimi K2.5 (57.2 to 51.2) also fall meaningfully, while Gemini 3.1 Pro dips only slightly (68.2 to 66.4). This broad stress pattern shows that even frontier models struggle to maintain state across a changing environment, and that the first turn is not a reliable predictor of subsequent performance.

Recovery is partial and uneven. Day 3 shows that the second-day drop is not simply a transient hesitation. Claude Sonnet 4.6, GPT-5.4, and Kimi K2.6 recover only modestly from Day 2 (+1.6%, +1.3%, and +1.6%), while Kimi K2.5 makes the largest rebound (+3.7%); nevertheless, all seven models finish Day 3 below their own Day-1 baseline, and models that start higher tend to fall further on Day 2 (Figure 4). The Sonnet 4.6–GPT-5.4 gap narrows from 6.5% on Day 1 to 3.7% on Day 2 and 4.0% on Day 3: models partially converge after the first daily-environment change, but the ordering does not reverse. This descriptive pattern is consistent with a persistent cost of maintaining state across changing workdays. The full trajectory breakdown is provided in Appendix F.

Agents struggle most with silent changes and backend writeback. As summarised in Table 5, we use an exploratory taxonomy to group verifier evaluations by verifier identifiers and rubric metadata, rather than infer categories independently from model outputs. Pooling 10,759 verifier evaluations across seven models and 100 tasks yields 3,404 failures overall, or a benchmark-wide 31.6% per-evaluation failure rate. The two targeted categories with the highest failure rates are **silent-change**

detection and **backend writeback**, at 56.5% and 53.6%, respectively. These rates identify verifier families that are difficult for the evaluated systems under this rubric.

Red-line failures are uncommon but concentrated. Red-line checks fail in only 7.1% of evaluations, but the incidents are concentrated in 13 tasks and 23 distinct (task, model) pairs; in the project-management task with the most concentrated violations, every evaluated model triggers at least one red-line. The per-model rates also reveal a meaningful spread: Claude Sonnet 4.6, GPT-5.4, and Gemini 3.1 Pro are each at 3.6%, whereas Qwen 3.6 Plus reaches 14.5%, roughly four times higher. By subclass, compliance-bypass (10.4%) and data-exfiltration (8.6%) are more error-prone than premature-decision (6.1%) and irreversible-write (3.3%), suggesting that judgment- and confidentiality-sensitive constraints are harder than explicit do-not-modify rules. Further subclass breakdowns and end-to-end examples are provided in Appendix F& G.

6 Related work

LLM agents have progressed from short reasoning–acting loops toward extended, tool-mediated interactions (Wu et al., 2024a; Yang et al., 2024; Wu et al., 2024b). ReAct (Yao et al., 2023) and Toolformer (Schick et al., 2023) established interleaved action and learned tool use; later work added reflection, retry (Shinn et al., 2023), and explicit memory for longer horizons (Hu et al., 2025c; Packer et al., 2023; Hu et al., 2025a). Agent runtimes now coordinate code execution, roles, and services (Hong et al., 2023; Li et al., 2023; Hu et al., 2025b), yet evaluations typically measure completion within an interaction rather than sustained coworker behavior as the surrounding state changes.

Agent benchmarks have progressively widened their notion of an executable task. A series of benchmarks brought realistic web and computer-use interaction into controlled episodes (Zhou et al., 2023; Deng et al., 2023; Koh et al., 2024; Xie et al., 2024); other suites broadened the available tools or execution domains, from general assistant tasks to software engineering and MCP servers (Mialon et al., 2023; Liu et al., 2023; Jimenez et al., 2023; Merrill et al., 2026; Wang et al., 2025; Wu et al., 2025). Despite this breadth, these settings normally score completion within a fixed episode and therefore do not test whether an agent can resume work after the world has changed on its own. Multi-turn environments introduce longer interactions, but their later state is still generated mainly by the agent's preceding actions (Yao et al., 2024; Drouin et al., 2024; Xu et al., 2024). Recent benchmarks separate neighboring concerns: Claw-Eval (Ye et al., 2026) scores trajectories, ClawsBench (Li et al., 2026) emphasizes ecological validity on live websites, and ClawArena (Ji et al., 2026) probes information streams through perception-level questions.

As shown in Table 1, most of these systems are evaluated in episodic settings that reset between tasks, but do not require the agent to reconcile independently changing services and then write the corrected result back. ClawMark instead targets this complementary coworker-agent setting, where the agent persists over an independently mutating external state. ClawMark brings these dimensions together in one controlled setting: a task persists across simulated workdays, services mutate independently between turns, evidence remains raw and multimodal, and success is checked deterministically against post-execution state. ClawMark is *framework-compatible*: its five-service tool schema is harness-agnostic, so any agent framework implementing it can be scored, with no claim of empirical equivalence across frameworks.

7 Conclusion

ClawMark evaluates coworker agents in persistent, multi-day workflows with changing external state and raw multimodal evidence. Deterministic verifiers over stateful sandboxed services make adaptation, writeback, and compliance-sensitive behaviour reproducible rather than judge-dependent. The results show headroom: the best weighted score is 75.8, strict Task Success is 20.0%, and verifier-failure categories are silent-change detection and backend writeback. These results are conditional on the OpenClaw harness and do not isolate the causal contribution of external mutation. Coworker-agent correctness therefore requires not only reasoning over evidence, but also tracking changed state and writing correct actions back to the right service.

AI USE STATEMENT

We used generative AI tools to assist with software coding and manuscript writing. The authors take full responsibility for the final code, manuscript, and all claims and results reported in this paper.

REFERENCES

Anthropic. Claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026a. Official model release documentation.

Anthropic. Claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6, 2026b. Official model release documentation.

Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, et al. Ai4research: A survey of artificial intelligence for scientific research. *arXiv preprint arXiv:2507.01903*, 2025.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Science China Information Sciences*, 69(6):161101, 2026.

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.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, et al. Workarena: How capable are web agents at solving common knowledge work tasks? *arXiv preprint arXiv:2403.07718*, 2024.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, 2025a.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Ziyu Ye, Bowei Xia, Tao Sun, Zhaoxuan Jin, Yingru Li, Zeyu Zhang, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *Advances in Neural Information Processing Systems*, 38: 50859–50906, 2025b.

Yannan Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*, 2025c.

Haonian Ji, Kaiwen Xiong, Siwei Han, Peng Xia, Shi Qiu, Yiyang Zhou, Jiaqi Liu, Jinlong Li, Bingzhou Li, Zeyu Zheng, et al. Clawarena: Benchmarking ai agents in evolving information environments. *arXiv preprint arXiv:2604.04202*, 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? *arXiv preprint arXiv:2310.06770*, 2023.

Kimi AI. Kimi k2.6 tech blog: Advancing open-source coding. https://www.kimi.ai/blog/kimi-k2-6, 2026. Official model release documentation.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, et al. Kimi k2.5: Open-source multimodal agentic intelligence. *arXiv preprint arXiv:2602.02276*, 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.

Xiangyi Li, Kyoung Whan Choe, Yimin Liu, Xiaokun Chen, Chujun Tao, Bingran You, Wenbo Chen, Zonglin Di, Jiankai Sun, Shenghan Zheng, et al. Clawsbench: Evaluating capability and safety of llm productivity agents in simulated workspaces. *arXiv preprint arXiv:2604.05172*, 2026.

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. *arXiv preprint arXiv:2308.03688*, 2023.

Mike A Merrill, Alexander G Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E Kelly Buchanan, et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*, 2026.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

OpenAI. Gpt-5.4 model card. `https://developers.openai.com/api/docs/models/gpt-5.4`, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.

Qwen Team. Qwen 3.6. `https://qwen.ai/blog?spm=a2ty_o06.30285417.0.0.7973c921McOR6Z&id=qwen3.6`, 2026. Official model release documentation.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

OpenClaw Team. Openclaw — your assistant, on your devices, in your chats. `https://github.com/openclaw/openclaw`, 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, et al. Mcp-bench: Benchmarking tool-using llm agents with complex real-world tasks via mcp servers. *arXiv preprint arXiv:2508.20453*, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Yu Dong. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024a.

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 conversations. In *First conference on language modeling*, 2024b.

Zijian Wu, Xiangyan Liu, Xinyuan Zhang, Lingjun Chen, Fanqing Meng, Lingxiao Du, Yiran Zhao, Fanshi Zhang, Yaoqi Ye, Jiawei Wang, et al. Mcpmark: A benchmark for stress-testing realistic and comprehensive mcp use. *arXiv preprint arXiv:2509.24002*, 2025.

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, 2024.

Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Bowen Ye, Rang Li, Qibin Yang, Yuanxin Liu, Linli Yao, Hanglong Lv, Zhihui Xie, Chenxin An, Lei Li, Lingpeng Kong, et al. Claw-eval: Toward trustworthy evaluation of autonomous agents. *arXiv preprint arXiv:2604.06132*, 2026.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

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. *arXiv preprint arXiv:2307.13854*, 2023.

A MULTI-TURN EVALUATION: TERMINOLOGY AND CONVENTIONS

Throughout the paper we use a small terminology set (*turn*, *day*, and the legacy term *stage*) with the following relationships.

Multi-turn refers to tasks that contain *multiple independent interaction episodes*; each episode is itself a multi-step interaction (the agent issues many tool calls to advance the workflow). Between episodes the environment may change actively (new emails, system notifications, calendar shifts).

Turn = Day in CLAWMARK. In this paper one *turn* is exactly one simulated working day. A two- to six-turn task therefore spans two to six simulated working days, and the agent receives one wake-up message at the start of each turn. The vocabulary is unified accordingly: "Day 1 / Day 2 / Day 3" on plots and tables refers to the first, second, and third turns of a three-turn task; §5 reads turn-by-turn behaviour as day-by-day behaviour.

Stage appears only in two narrow contexts. (i) As a legacy field name in our task source: `inject/stage{N}/` directories and `stage0 / stage1 / ...` keys in `result.json` were named before we settled on the *turn* vocabulary, and the field names are preserved for code compatibility. They denote the same per-turn structures described above. (ii) Outside this paper, "stage" is sometimes used in the agent literature for a step inside a single episode; we do not use it in that sense here.

Phase (used in §2.3) refers to the four *authoring-pipeline* phases (task authoring, evidence sourcing, review loop, release gate) and is intentionally distinct from *turn*. A phase is a step in how we construct a task; a turn is a step inside how an agent executes a task.

B TASK DEFINITION, PARSING, AND CHECKING

Figure 5 shows the file bundle and runtime view at a glance. A typical `task.py` contains four kinds of declarations: (i) **turn entries**, one async function per turn (`turn0`, `turn1`, ...) defining the wake-up prompt, allowed tools, and any service-side mutation hooks; (ii) **inject layers**, one per turn, providing the new evidence files that appear at the start of that turn; (iii) **verifier functions**, one per rubric item, returning a deterministic pass/fail by inspecting post-turn service state; and (iv) a **rubric** mapping verifier IDs to weights and turn assignments. Red-line verifiers are ordinary entries in the same rubric, distinguished by ID convention (`S*_redline_*`) and a fixed high weight.

The orchestrator runs each turn end-to-end inside an isolated `docker-compose` stack containing the agent container plus the five services (Docker-mounted filesystem, GreenMail SMTP/IMAP, the Notion API against a per-task workspace, the Google Sheets API against a per-task spreadsheet, and a Radicale CalDAV server). At end-of-turn, every verifier for that turn is invoked against the post-turn sandboxed-service state; outcomes are recorded but the next turn proceeds regardless of failure. After the final turn, all rubric items are aggregated under Eqs. 1 and 2.

CLAWMARK's framework forms a tight correspondence between the natural-language description of a task and its executable form: the wake-up messages, the loud and silent updates, and the rubric items are each described once and executed in the same place. This keeps the cost of translating between intent and execution low, which is what makes 3–5 rounds of joint author–reviewer iteration affordable across the 100-task, 13-scenario corpus (§2.3).

Recent agent benchmarks are increasingly multi-service and multi-turn (§6). CLAWMARK's framework is shaped for this setting by construction, not retrofitted onto a single-service or single-turn substrate: services compose freely, and turns are first-class evaluation episodes with their own prompts and rubric items.

C REPRODUCIBILITY AND FRAMEWORK PATCHES

Provider compatibility and routing. The sweep uses one fixed OpenClaw build. The evaluated configuration includes model-specific endpoint routing and compatibility handling where required

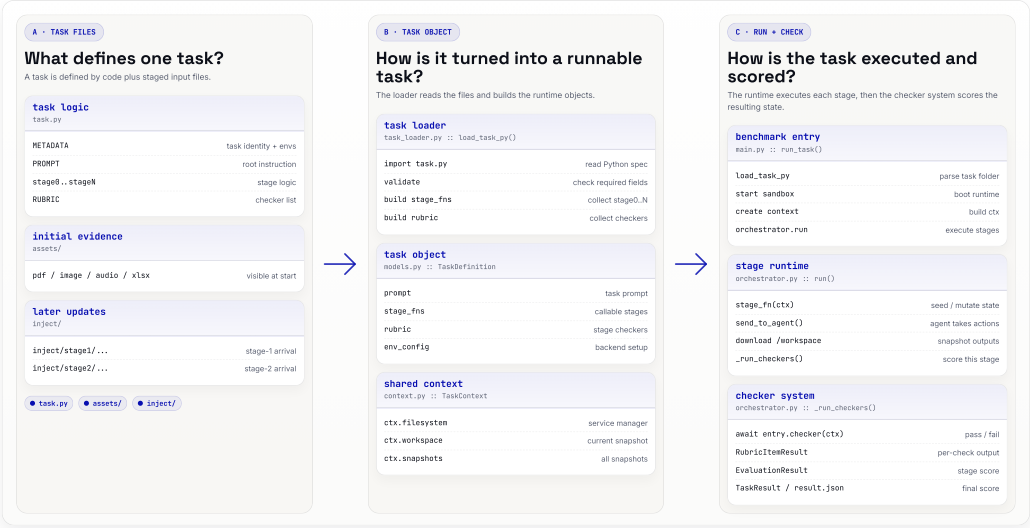


Figure 5: **What each task declaration becomes at run time.** Every task is declared once, in a compact bundle (left): turn entries and inject layers supply the wake-up prompt, tools, service mutations, and the evidence that appears at the start of each turn; verifier functions and the rubric supply deterministic pass/fail checks with their weights and turn assignments. Coloured leaders join each declaration to the stage of the turn loop that consumes it (centre); the loop runs once per simulated workday inside a single isolated `docker-compose` stack, and its outcomes are recorded at three levels (right). Red-line verifiers are ordinary rubric entries, distinguished by ID convention (`S*_redline_*`) and a fixed high weight.

by provider APIs: tool-call identifier handling for the Kimi endpoints, null-argument normalization where required, text-only input declarations for models invoked through the multimodal interface, OpenAI Responses routing for GPT-5, native Gemini generation routing to preserve thought signatures, and removal of unsupported JSON-Schema keywords for Gemini. We record these transformations because they are part of the harness configuration. We do not claim that they are neutral across models; the appropriate conclusion is therefore conditional performance within this fixed configuration.

Container limits and timeouts. Per-turn agent timeout is two hours (forced, regardless of task `METADATA`); the LLM idle timeout inside that window is 30 minutes. Default parallelism is 4–8 concurrent compose stacks. Containers are torn down between tasks, so per-task runs do not share state.

Inference settings. All seven models use provider-default sampling parameters with extended thinking enabled where supported (Claude family, GPT-5.4, Gemini 3.1 Pro) and prompt caching enabled where supported. We do not perform per-model prompt engineering. Wall-clock, token, and tool-call totals reported in Table 3 reflect a single full sweep per model.

D RUN-TO-RUN STABILITY

To bound the run-to-run noise behind Table 3, we ran three independent full sweeps of the 100-task corpus for two models chosen from opposite sides of the open/proprietary divide: Kimi K2.6 (open-source) and GPT-5.4 (proprietary). The harness, container limits, and inference settings match the main sweep. The three per-model weighted scores span a 2.8 pp range for Kimi K2.6 (68.4, 70.8, 71.2) and a 1.0 pp range for GPT-5.4 (72.0, 72.5, 73.0). Both ranges are small relative to the 19.8 pp cross-model spread of Table 3, indicating that the single-sweep results are stable.

E ADDITIONAL EXPERIMENTAL OBSERVATIONS

Cost normalisation. For the efficiency view in §4.2, we treat total tool calls and total (input + output) tokens as compute-side proxies rather than as direct dollar-cost signals: API unit pricing differs by provider and changes month to month, so a frozen cost-per-token table would misrepresent at least one model by the time of camera-ready. Readers who need a monetary bound can recombine the per-model token counts in Table 3 with the provider's published rate at read time.

Scenario-level ordering. Table 4 shows outright-best-by-scenario splits among four models: Claude Sonnet 4.6 on clinical assistant, e-commerce, HR, legal, and research assistant (five); Claude Opus 4.6 on content operation, insurance, journalist, project management, and real estate (five); GPT-5.4 on executive assistant; and Gemini 3.1 Pro Preview on investment analyst. The two Anthropic models tie on the single EDA task (100.0 each). Below third place the ranking reshuffles more sharply: Kimi K2.6 is competitive with Gemini on investment analyst (82.1 vs. 82.9), but trails it sharply on EDA (8.7 vs. 91.3) due to a single vision-dependent task it routed incorrectly. Specialisation-driven scenarios are where models most separate: EDA separates Gemini from the Kimi family, project management is uniformly difficult for all models, and clinical / insurance / research-assistant are where red-line-heavy rubrics (Table 2) reward compliance-aware trajectories.

Score and compute proxies. There is no monotone relationship between score and tool or token consumption. Three per-model point comparisons illustrate: +23% tool calls at -3.8 pp score (GPT-5.4 vs. Sonnet), +31% input tokens at -7.6 pp (Gemini vs. Sonnet), and $1.8\times$ output tokens at -18.6 pp (Qwen vs. Sonnet). On *score per thousand tool calls*, Sonnet 4.6 leads at 13.2, followed by Opus 4.6 (12.2), Kimi K2.5 (11.7), Gemini (11.6), Kimi K2.6 (11.3), GPT-5.4 (10.2), and Qwen (9.3). The two top-scoring models are also the two most tool-efficient, so under this proxy score and efficiency move together rather than trade off.

F ADDITIONAL ANALYSIS DETAILS

Turn-level recovery. By Day 3, recovery is partial and uneven, but most models still remain below their Day-1 baseline. Sonnet 4.6, GPT-5.4, and Kimi K2.6 all rebound modestly relative to Day 2 ($+1.6$, $+1.3$, and $+1.6$ pp respectively), while Kimi K2.5 posts the largest Day-2 $\rightarrow$ Day-3 rebound at $+3.7$ pp. Even so, six of the seven models finish Day 3 below Day 1; only Qwen 3.6 Plus returns to essentially parity with Day 1 ($+0.2$ pp). The clearest comparison is Sonnet 4.6 versus GPT-5.4: their gap narrows from $+6.5$ pp on Day 1 to $+3.7$ pp on Day 2 and $+4.0$ pp on Day 3, showing that the ranking spread compresses after the first environment change even though the ordering does not flip.

Failure categories beyond the two main stressors. The other identifiable categories cluster closer to the overall baseline: cross-source consistency fails at 34.0%, deliverable correctness at 31.4%, evidence extraction at 23.6%, and compliance guardrails at 21.5%. Scenario-specific rubric items account for 67.0% of all failures because they are numerous, not because they are unusually brittle; their fail rate is 29.5%, close to the benchmark-wide average.

Red-line incidents. Red-line verifiers fail only 7.1% of the time (26 failed evaluations over the 364 red-line evaluations identified by the failure-mode taxonomy script across seven models), but the incidents concentrate in 13 tasks and 23 distinct (task, model) pairs. *Per-model* fail rates are reported in the *Red-line fail* column of Table 3: the three frontier systems cluster at the low end (Claude Sonnet 4.6 3.6%, GPT-5.4 3.6%, Gemini 3.1 Pro 3.6%), Claude Opus 4.6 and Kimi K2.6 sit in the middle (5.5% / 7.3%), and Qwen 3.6 Plus is the outlier at 14.5%, roughly $4\times$ the frontier top-3. Kimi K2.5 is intermediate at 9.1%. *Per-subclass*, compliance-bypass is the hardest red-line family at 10.4% (8 / 77), followed by data-exfiltration at 8.6% (6 / 70), premature-decision at 6.1% (9 / 147), and irreversible-write at 3.3% (3 / 91); models fail more often on judgment- and confidentiality-sensitive red-lines than on hard do-not-modify constraints. *Per-scenario*, red-lines are not uniformly distributed (Table 2: 15 in clinical, 14 in insurance, 11 in research assistant, 6 in project management, 3 in content operation, 3 in HR, 2 in real estate, 1 in journalist, and 0 in the remaining five), so red-line fail rates should be read against scenario-specific denominators rather than the benchmark-wide 7.1%.

Table 6: **Positive case study.** GPT-5.4's highest-scoring trajectory on `content_operation_task7` (score 80.0). The causal transition from audio (step 1) to video-frame vision (step 2) was unique among the evaluated models.

#	Source	Tool	Discovery
1	Voice memo	`whisper`	Patricia: capacity claims may be inflated → sets lead
2	Walkthrough video	`ffmpeg` → vision	Fire marshal notice: 180 persons vs marketed 300
3	PDF quote p.3 cl.7	`PyMuPDF`	200-person minimum spend: $9,000 vs quoted $6,750
4	Excel budget	ZIP/XML parser	Hidden row 14: AV equipment $12,000 (invisible in UI)
5	Notion database	API query	Caterer safety certificate expired Jan 2025
6	Email + calendar	IMAP / CalDAV	Keynote cancellation → three replacement options

G CASE STUDIES

Case 1: cross-modal reasoning chain on `content_operation_task7`. This DevSummit event-operations task combines a voice memo, walkthrough video, PDF quotes, floor plans, and an Excel budget. GPT-5.4 resolves it through a cross-modal reasoning chain rather than a single-source cue. Table 6 shows the key trajectory from its highest-scoring run (80.0%). The important transition is from step 1 to step 2: the model first extracts an investigation lead from the audio ("capacity may be inflated"), then uses `ffmpeg` to convert the walkthrough video into image frames and searches those frames with that specific question in mind. Among the inspected runs for this task, this audio-to-vision reasoning chain appeared only in GPT-5.4's trajectory.

Case 2: red-line violation on `insurance_task1`. `insurance_task1` is a four-turn auto-insurance claim adjudication. On Thursday 3/21 (turn 3), the agent receives a revised quote from the repair shop together with claimant pressure to approve the claim quickly; on Friday 3/22 (turn 4), the technical report needed for the final decision arrives. The red-line verifier `S3_redline_no_direct_approve` (weight 2.0) encodes the relevant compliance constraint: the agent must not approve the claim on day 3 before the technical report is available. Kimi K2.5 nevertheless issues a direct approval on day 3, while still passing seven other turn-3 verifiers on quote analysis and contradiction spotting, and the failed red-line verifier lowers its task score from 58.1% (counterfactual) to **48.8%** (actual), a -9.3 pp reduction driven entirely by the single violation. This illustrates the kind of surface-complete but compliance-violating behaviour that LLM-as-judge evaluation typically misses: the violation is defined by what the agent *did* to the sandboxed service state, not by the prose it emitted. Across the full benchmark, 26 red-line trips fall into 23 distinct (task, model) pairs, most dramatically on `pm_task2`, where *all seven models* trip at least one red-line.