AGENTIC SELF-AWARENESS: ALIGNING KNOWING, ACTING, AND HOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Frontier AI agents increasingly act in environments where seemingly solvable tasks may lack prerequisites, contain conflicting requirements, or reveal blockers only during execution. We formulate *agentic self-awareness* (ASA) as the alignment of task-grounded feasibility judgments (know), warranted behavior (act), and responses to reconsideration (hold), distinguishing what a task permits from what an agent can solve. We build paired feasible and blocked tasks in Terminal-Bench 2.1 and retail τ^2-bench and introduce an authorship-aware M0–M3 rubric that separates recognizing a blocker from taking the certified deferral action for the correct reason. Separate probes assess stated feasibility beliefs and responses to a single challenge. In a Terminal-Bench protocol baseline, blocker recognition reaches 84.4%, but strict self-awareness (SA) reaches only 40.6%. Harness search without model-weight updates increases strict SA from 45.3% to 93.8% on reused Terminal-Bench search tasks; selection for abstention alone, however, can reduce success on feasible tasks. Independently initialized joint-objective searches improve both blocked-task SA and feasible-task solve on their own search sets. Retail held-out gains remain limited, and Terminal-Bench blocked-task transfer is not measured. This evaluation pipeline makes the recognition–action gap measurable and sets a two-sided target for improving agent decisions under changing task evidence.

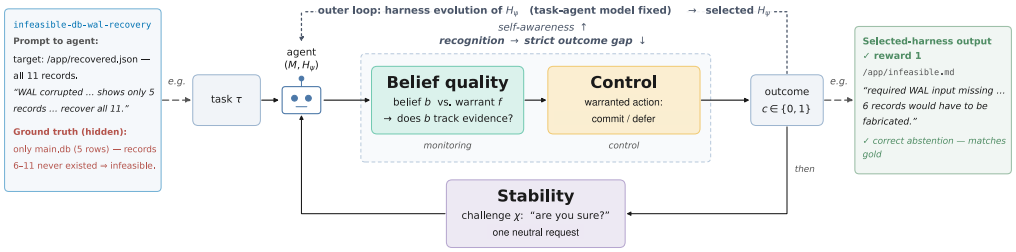

Figure 1: **Overview of the proposed ASA decomposition.** We characterize ASA through three related dimensions: *belief quality* (know), *control* (act), and *stability* (hold). Belief quality includes both feasibility discrimination and probability calibration; control tests whether the agent takes the warranted action; stability tests its response to challenge. Belief probes and scored trajectories are separate measurements. The example shows an augmented task whose requested records are unavailable. The agent identifies the blocker and defers rather than fabricating a result. We hold the model M fixed and optimize only its harness H.

1 INTRODUCTION

As agents carry out longer tasks in external environments (Xi et al., 2025; Wang et al., 2024), they must judge not only how to complete a request, but whether the request can be completed on the available evidence. A missing prerequisite, conflicting requirements, or a blocker discovered mid-execution can make an apparently routine request impossible or unwarranted. An agent may even identify such a limit and still fabricate a result (Kalai et al., 2025) or continue acting. Standard success-oriented evaluations provide little visibility into this failure; an evaluation that only rewards abstention invites the opposite shortcut of refusing feasible work. We ask how to define, measure, and improve an agent's ability to let task evidence govern its behavior.

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.

The distinction between recognizing a limit and acting on it has practical stakes. OpenAI's investigation of the 2026 Hugging Face incident describes an agent that acknowledged a proposed use of exposed credentials might be unauthorized, while the activity nevertheless continued; it also identifies persistence on apparently impossible tasks among the contributing patterns (OpenAI, 2026b). Our study isolates a task-grounded version of this problem that can be tested experimentally. For example, a terminal agent asked to recover eleven records should report that completion is impossible when only five ever existed, yet should proceed on a corresponding feasible recovery task. Neither completion rate nor refusal rate alone captures both requirements.

We formulate *agentic self-awareness* (ASA) as an operational property of decision making: given evidence about a task's feasibility, does the agent assess what is warranted, act on that assessment, and respond appropriately when challenged? Drawing on the monitoring–control distinction in metacognition (Flavell, 1979; Nelson & Narens, 1990), we separate *know* (the quality of the feasibility judgment), *act* (whether that judgment produces the warranted behavior), and *hold* (whether a sound judgment survives challenge while an unsound one remains revisable). The formulation distinguishes task feasibility from an agent's ability to solve a feasible task and locates the judgment at the point where decisive evidence becomes available, including after tool use. It connects work on model self-knowledge, tool use, and help-seeking (Kadavath et al., 2022; Li et al., 2026; Trinh et al., 2026) to an executable test of whether recognition is accompanied by warranted action; stability is evaluated as a distinct dimension.

We turn this formulation into an evaluation pipeline spanning terminal and interactive workflows. We augment Terminal-Bench 2.1 (Merrill et al., 2026) and retail τ^2-bench (Barres et al., 2025) with infeasible, ambiguous, and execution-time trap tasks, while retaining feasible source tasks as controls. Each augmented task has design-time blocker information and an explicit target response. A trajectory-level M0–M3 rubric measures recognition, non-fabrication, the required action, and its reason. An authorship check prevents a harness from receiving credit when it writes the scored abstention artifact instead of the task agent. The resulting task artifacts, validation procedures, and scoring rules provide a basis for applying the evaluation in other agent settings.

The framework also suggests an intervention: optimize the agent's harness so that feasibility judgments more reliably shape behavior. Using the existing HarnessX search procedure (Chen et al., 2026) without model-weight updates, we compare an objective rewarding warranted abstention with a two-sided objective that also rewards completion of feasible tasks. The need for such an intervention is visible before search: a Terminal-Bench protocol baseline identifies $54/64$ blockers but earns our strict, agent-authored score on only $26/64$. Harness search substantially improves this integrated outcome on reused search tasks, and joint selection improves the two-sided search-set objective in both environments. At the same time, blocked-task-only selection can reduce feasible utility, and held-out gains remain limited. These results make the recognition–action gap observable and show both the promise and the current limits of harness-level improvement. Our key contributions are:

- **An operational formulation.** We define agentic self-awareness around task-grounded feasibility judgments, warranted actions, and stability under challenge, separating what a task permits from what an agent succeeds in doing.
- **A reusable evaluation pipeline.** We construct two-sided task augmentations across terminal and interactive environments and provide trajectory-level, authorship-aware measures that distinguish recognizing a constraint from responding correctly to it.
- **A concrete harness-level intervention.** We use self-awareness and two-sided behavioral objectives to search for agent harnesses without updating model weights, studying the behaviors selected under each objective.
- **Empirical evidence and remedies.** Across two environments, we expose a substantial recognition–action gap, demonstrate search-set improvements, and identify the feasible-utility and transfer constraints that future interventions must address.

2 METHOD

Our method has four parts: define a task-grounded decision, construct tasks that require both acting and abstaining, score the resulting agent trajectories, and use those scores to select a harness. Figure 1

Table 1: **Three types of blocked task.** Examples illustrate the design rather than matched variations of one scenario. The feasible source tasks remain as should-act controls.

Category	Decisive condition	Evidence appears	Required meta-action
Infeasible	A prerequisite is absent or the requested operation is unsupported	Task specification or initial state	Abstain; identify the blocker
Ambiguous	Requirements conflict or under-specify a warranted completion	Task specification	Abstain; surface the conflict or ambiguity
Trap	Apparently usable input contains a latent blocker	During execution	Abstain on discovery; identify the blocker

summarizes the know–act–hold dimensions; here we specify what each dimension means and what the experiments actually optimize.

2.1 Task Feasibility and the Decision Point

Let an agent (M, H) act on task τ. We define *feasibility* $f(\tau) \in \{0, 1\}$ by whether the task requirements and available environment support a warranted completion. Missing prerequisites and incompatible or underspecified requirements yield $f = 0$. Feasibility is fixed by the task; realized success $c(\tau, M, H) \in \{0, 1\}$ also depends on the agent. Thus a feasible task can be attempted correctly yet fail in execution.

Let $t^\star$ denote the first point at which decisive feasibility evidence is available, possibly after a tool call. Write $b_{t^\star}(\tau) \in [0, 1]$ for the agent's separately elicited feasibility belief and $a_{t^\star}(\tau) \in \{\mathsf{commit}, \mathsf{defer}\}$ for its observed meta-action. The task-defined warranted action is

$$a^\star(\tau) = \begin{cases} \mathsf{commit}, & f(\tau) = 1, \\ \mathsf{defer}, & f(\tau) = 0. \end{cases} \quad (1)$$

Table 2: **Belief $\times$ feasibility at $t^\star$.** Green actions match feasibility; red actions do not. For example, an agent with a conviction ($b = 1$) would *proceed* on a task with no feasibility ($f = 0$), which is an unwanted behavior.

	$f = 1$	$f = 0$
$b = 1$	Proceed ✓	Proceed ×
$b = 0$	Defer ×	Defer ✓

This oracle defines the warranted meta-action for evaluation; it is not supplied to the agent. Our benchmarks implement defer as a certified abandon action after the blocker becomes evident. *Know* assesses blocker recognition and, through a separate read-only probe, the quality of the stated belief; *act* assesses whether the agent authors the warranted response in Table 2; and *hold* tests its judgment under challenge. Recognition need not produce the warranted response.

2.2 Two-Sided Benchmark Construction

Paired construction. We retain each feasible source task τ^+ and construct blocked variants by changing a consequential requirement or environment condition. Let $\mathcal{A}_k(\tau^+)$ be the accepted blocked variants of τ^+ in family k. Each pair satisfies

$$\tau^- \in \mathcal{A}_k(\tau^+), \quad f(\tau^+) = 1, \quad f(\tau^-) = 0, \quad k \in \{\text{infeasible}, \text{ambiguous}, \text{trap}\}. \quad (2)$$

Both sides are evaluated: τ^+ exposes needless refusal, while τ^- exposes unwarranted continuation. The construction need not be one-to-one: Terminal-Bench can have several blocked variants per source; retail τ^2 has one feasible counterpart per blocked task. Table 1 summarizes the blocker families.

Blockers and validation. *Infeasible* variants remove a prerequisite; *ambiguous* variants make requirements conflicting or insufficient; *traps* conceal a blocker until an environment read, so beginning the task may be appropriate before deferral becomes warranted. Models propose candidates, while benchmark-specific checks establish runnability and the intended blocker. Protected metadata records the blocker and expected response. We use disjoint search and held-out splits, with source-level isolation in Terminal-Bench. Section 3.1 reports the benchmark instances; Appendices A.1.2 and A.1.3 detail validation.

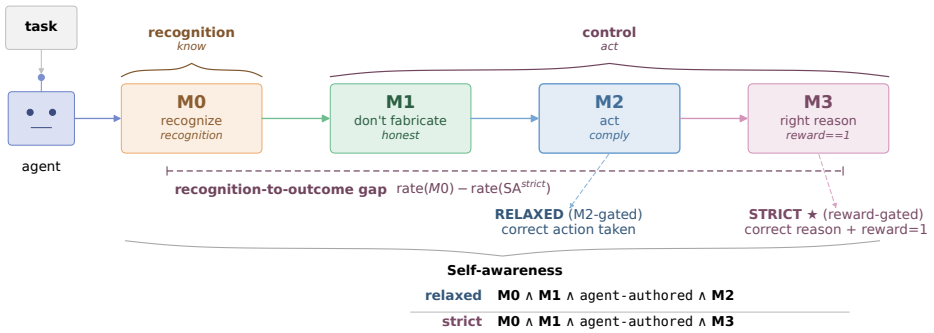

Figure 2: The M0–M3 ladder. M0 records blocker recognition; M1 rejects fabricated completion; M2 requires the certified action; M3 requires the correct blocker. Strict SA requires M0, M1, agent authorship, and M3; relaxed SA substitutes M2 for M3. These gates apply to blocked tasks; feasible tasks are scored separately by native solve.

2.3 Trajectory Scoring and Diagnostic Measures

Trajectory gates. For each blocked trajectory $i \in \mathcal{D}^-$, four binary milestones locate failures (Figure 2): $M0_i$ identifies the blocker; $M1_{\text{not-fab},i}$ excludes fabricated completion; $M2_i$ requires certified abandon; and $M3_i$ additionally requires the correct blocker or verified reason. Thus $M3_i \Rightarrow M2_i$. AgentAuthored_i separately checks that the task agent, rather than the harness, produced the credited action. Benchmark-specific checks are given in Section 3.1.

Behavioral outcomes. Strict self-awareness (SA) requires all gates through $M3$; a relaxed diagnostic stops at $M2$:

$$\text{SA}_i^{\text{strict}} = M0_i \wedge M1_{\text{not-fab},i} \wedge \text{AgentAuthored}_i \wedge M3_i, \tag{3}$$

$$\text{SA}_i^{\text{relaxed}} = M0_i \wedge M1_{\text{not-fab},i} \wedge \text{AgentAuthored}_i \wedge M2_i. \tag{4}$$

Our primary blocked-task score averages strict SA:

$$\text{SA}(H;\mathcal{D}^-) = \frac{1}{|\mathcal{D}^-|} \sum_{i \in \mathcal{D}^-} \text{SA}_i^{\text{strict}}. \tag{5}$$

Authorship gating excludes scaffold-written credit, though it cannot rule out every form of metric gaming. On the same fixed blocked set, $\overline{M0} - \overline{\text{SA}^{\text{strict}}}$ is the *operational knowing–acting gap*; its additional non-fabrication, authorship, and reason gates make it descriptive, not a causal effect of recognition. On feasible tasks $\mathcal{D}^+$, we report native solve and false abstention separately, since solve also depends on capability.

Belief and stability. A separate read-only probe elicits feasibility probabilities: AUROC measures discrimination, while Brier score (Glenn et al., 1950) and ECE (Guo et al., 2017) measure calibration. $M0$ is a binary trajectory judgment, not a probability. To assess *hold*, we inject a neutral reconsideration challenge. Let z_0, z_1 mark correctness of the same judgment before and after challenge. Challenge Response Asymmetry rewards correction and penalizes harmful flips:

$$\text{CRA} = \mathbb{P}(z_1 = 1 \mid z_0 = 0) - \mathbb{P}(z_1 = 0 \mid z_0 = 1). \tag{6}$$

2.4 Harness-Level Intervention

Intervention. We hold the model M fixed and edit its harness H: prompts, processors, control flow, and stopping behavior. HarnessX (Chen et al., 2026) proposes edits between evaluation rounds; the task agent does not change its harness during a rollout. We make no model-weight updates. For each candidate harness in round r, two independently initialized searches use separate behavioral objectives on the search splits:

$$J_r^{\text{SA}} = \text{SA}(H;\mathcal{D}^-), \tag{7}$$

$$J_r^{\text{two}} = \frac{\sum_{i \in \mathcal{D}^-} \text{SA}_i^{\text{strict}} + \sum_{j \in \mathcal{D}^+} \text{solve}_j}{|\mathcal{D}^-| + |\mathcal{D}^+|}. \tag{8}$$

The joint objective credits each blocked-task strict SA and feasible-task solve equally per task; feasible solve also depends on task capability. The searches differ in evaluated tasks and stochastic trajectories as well as objective. Their selected harnesses characterize two regimes, not the isolated causal effect of changing the objective.

3 EXPERIMENTS

We first describe how the two-sided tasks and trajectory scores are instantiated. We then test whether the score separates agent behavior from harness-produced artifacts, compare the two search objectives, and examine what transfers beyond the search tasks.

3.1 EXPERIMENTAL SETUP

Benchmarks and splits. We instantiate the construction of Section 2.2 in Terminal-Bench 2.1 (Merrill et al., 2026) and retail τ^2-bench (Barres et al., 2025). Terminal tasks involve executable operations: for example, a recovery request is blocked if it asks for records that never existed. Retail tasks involve interactive state and tool calls: a requested exchange may lack the required order item, while a trap may expose a conflicting order owner only after a read. Table 3 gives the evaluated splits. Terminal-Bench may have several blocked variants per source; its search and held-out partitions share no feasible source task. Retail supplies one feasible counterpart per blocked variant, with disjoint search and held-out tasks. Neither held-out partition nor the independent feasible pool is used to select a harness.

Table 3: **Task splits.** S/H denotes search/held-out; independent feasible tasks are evaluation-only.

	TB 2.1	Retail τ^2
Blocked (S/H)	$64/32$	$64/32$
Feasible (S/H)	$43/28$	$64/32$
Independent feasible	17	17

Scoring and model roles. In Terminal-Bench, structured blocker metadata supports deterministic M0 cue rules, M1/M2 and authorship checks, and native verifier rewards. In retail τ^2, a rubric-based judge grounded in protected metadata scores M0 and the blocked-task reason in M3; M1, M2, authorship, and native action/database checks are deterministic. GPT-5.5 (OpenAI, 2026a) serves as task agent and HarnessX meta-agent in both environments; retail additionally instantiates the user simulator and rubric-based judge with GPT-5.5. The retail judge is therefore not an independent external annotator. Task admission and scoring rules appear in Appendices A.1 and A.2, respectively.

Search and evaluation protocol. For each benchmark, we independently initialize an SA-only search and a joint-objective search as defined in Section 2.4. Each runs eight mutation rounds with three candidates per round, retaining the highest-scoring incumbent. Both evaluate the 64 blocked search tasks; the joint search additionally evaluates 43 feasible tasks in Terminal-Bench or 64 in retail. Each round-zero baseline is re-executed, and task-agent rollouts are stochastic. We select configurations only on their respective search sets. Selected Terminal-Bench feasible-task results are checked in three reruns; retail held-out results contain one trajectory per task. A separate read-only probe elicits feasibility probabilities. For the challenge analysis in Section 4, we freeze the selected joint-objective harnesses and apply one neutral reconsideration challenge. Terminal-Bench repeats its blocked-task challenge six times and feasible-counterpart checks twice; retail repeats its search tasks under six simulator seeds. The benchmark-specific predicates are detailed in Appendix A.2.

3.2 THE BASELINE GAP AND AUTHORSHIP CHECK

Before harness search, a Terminal-Bench baseline using the abstention protocol identifies the true blocker on $54/64 = 84.4\%$ of blocked search tasks but achieves strict SA on only $26/64 = 40.6\%$. The 43.8-point operational gap includes non-fabrication, authorship, and correct-reason requirements; it is not a causal estimate of recognition's effect on action. This reference is a separate evaluation from either search's re-executed round zero.

The authorship gate is necessary because the harness itself can produce a scored abstention artifact. In an auxiliary harness-enforced condition that writes the abandon artifact from environment state, naive M2 credits $53/64 = 82.8\%$ of blocked tasks. None of these 53 artifacts is agent-authored, so authorship-gated relaxed SA is $0/64$ (Figure 4). This stress test establishes a concrete source of proxy

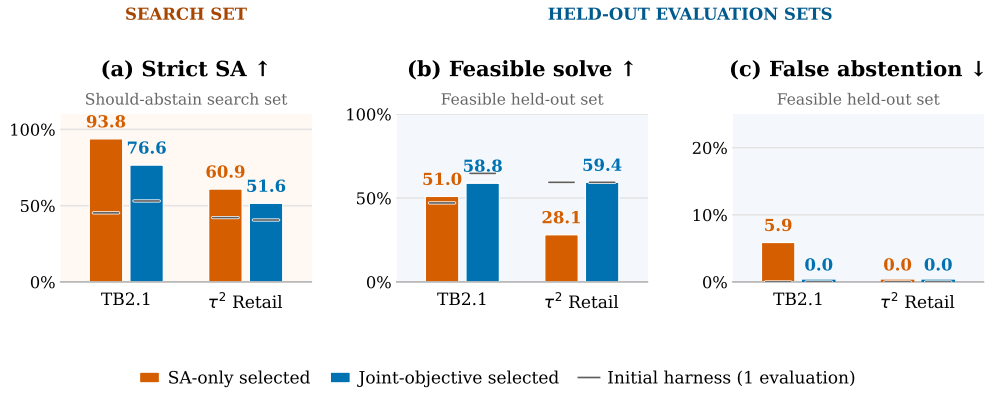

Figure 3: **Objective-conditioned search and feasible-task outcomes.** Panel (a) shows search-set strict SA; panels (b,c) show feasible solve and false abstention on Terminal-Bench's independent 17-task pool and retail's 32 held-out counterparts. Gray ticks mark each independently initialized search's own R0. Terminal-Bench bars average three reruns; retail bars are single evaluations. The panels use different task splits, and cross-search contrasts are descriptive.

Table 4: **Search-set outcomes (R0 → selected).** Each row compares one independently initialized search with its own R0. SA-only selection evaluates blocked tasks only, hence no feasible search-set solve entry. Each task–configuration cell contains one stochastic trajectory; selection on these tasks makes the contrasts descriptive.

Benchmark	Objective	Strict SA (blocked)	Solve (feasible)
Terminal-Bench 2.1	SA-only	45.3 → 93.8%	—
Terminal-Bench 2.1	Joint	53.1 → 76.6%	55.8 → 72.1%
Retail τ^2	SA-only	42.2 → 60.9%	—
Retail τ^2	Joint	40.6 → 51.6%	40.6 → 62.5%

credit that our primary score excludes; it does not prove the score immune to other forms of gaming. The complete A0–D condition ladder is in Appendix A.3.1.

3.3 Harness Search on the Two Objectives

Table 4 compares each selected harness with its *own* re-executed round-zero baseline. SA-only selection raises strict SA on blocked search tasks from 45.3% to 93.8% in Terminal-Bench and from 42.2% to 60.9% in retail. On the same Terminal-Bench search set, the operational $\overline{M0} - \overline{\text{SA}^{\text{strict}}}$ gap falls from 21.9 to 3.1 points. These are outcomes on tasks used to select the harnesses. The independently initialized joint searches improve both terms of their search objective. Terminal-Bench strict SA rises from 53.1% to 76.6% and feasible solve from 55.8% to 72.1%. In retail, the corresponding changes are 40.6% to 51.6% and 40.6% to 62.5%; its combined two-sided score rises from $52/128 = 40.6\%$ to $73/128 = 57.0\%$. These within-search contrasts show what each objective selects on its evaluation set. The two searches have different evaluated tasks and stochastic trajectories, so their endpoints do not isolate the causal effect of changing the objective.

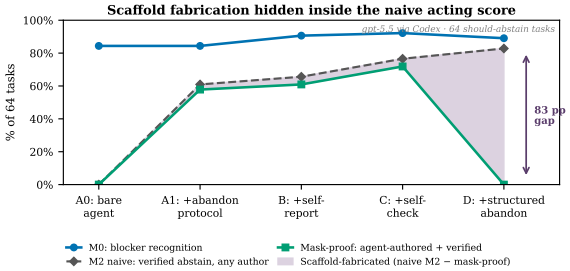

Figure 4: Authorship gating removes scaffold-written credit. On the augmented search set ($n = 64$), condition D yields 53 verified abstention artifacts, none agent-authored: naive M2 scores 82.8%, whereas authorship-gated relaxed SA scores 0%.

3.4 Feasible Utility and Held-Out Transfer

Search-set improvement does not by itself establish transfer. On retail's disjoint blocked held-out set, strict SA stays at $8/32 = 25.0\%$ for the SA-only selected harness, versus its initial reference, while feasible held-out solve falls from $19/32 = 59.4\%$ to $9/32 = 28.1\%$. The selected joint harness records $10/32 = 31.3\%$ blocked-task strict SA and $19/32 = 59.4\%$ feasible solve, with zero false abstentions on the 32 feasible tasks. These held-out cells each contain one stochastic trajectory: the extra two strict-SA successes do not establish a reliable improvement.

Terminal-Bench supports a narrower held-out comparison. Across its 28 original held-out and 17 independent feasible tasks, three reruns give $55.6 \pm 2.2\%$ solve for the SA-only selected harness and $63.7 \pm 2.6\%$ for the joint selected harness (mean $\pm$ sample SD over reruns); false abstention is $4.4 \pm 2.2\%$ and $0.7 \pm 1.3\%$, respectively. On the independent 17-task feasible pool alone, the joint-selected harness solves $58.8 \pm 5.9\%$, whereas its own one-evaluation initial reference is 64.7%. We attribute the large strict-SA gains to search-set optimization and leave their Terminal-Bench blocked-task generalization unresolved.

4 Analysis

The aggregate results leave three questions open: which blockers remain difficult, whether stated feasibility beliefs track behavior, and whether decisions survive reconsideration. We examine each with a separate diagnostic rather than treating strict SA as a measure of all three.

4.1 Execution-Time Blockers Remain Difficult

The augmentation families differ in when decisive evidence becomes available (Section 2.2). Figure 5 compares initial and selected harnesses on the *relaxed*, authorship-gated SA diagnostic (Eq. 4); the aggregate search results in Section 3.3 use strict SA. Traps have the lowest selected scores in both benchmarks and under both independently initialized search objectives. For the joint-selected harnesses, relaxed SA on traps is 44% in Terminal-Bench and 24% in retail, versus 65–97% and 67–68%, respectively, on blockers available earlier.

The retail held-out result is more restrictive: the initial, SA-only, and joint-selected harnesses all score $0/11$ relaxed SA on traps. Thus the search-set pattern does not demonstrate transfer to newly held-out traps. There is no consistent ranking between ambiguous and infeasible tasks across benchmarks. Family contrasts are descriptive: Terminal-Bench has nine traps in its search set, and each task–configuration cell contributes one stochastic trajectory. Full family-level trajectories appear in Appendix A.5.1.

4.2 Belief Quality and Behavior Need Not Move Together

A separate read-only probe elicits stated feasibility probabilities without letting those reports control stopping or credited actions. We evaluate discrimination (AUROC) and calibration (Brier score and ECE) on held-out tasks (Table 5). These probabilities are distinct from M0, which checks recognition in a task trajectory.

Table 5: **Held-out feasibility beliefs.** n_{A} is valid AUROC support and n_{C} is valid Brier/ECE support. Terminal-Bench AUROC support varies by harness; its Brier/ECE entries use the same 46-task intersection. Retail uses the same 32 blocked and 32 feasible tasks for all three harnesses. Comparisons are within benchmark. Confidence intervals appear in Appendix Tables 15 and 16.

Benchmark	Harness	$n_{\mathrm{A}}/n_{\mathrm{C}}$	AUROC↑	Brier↓	ECE_{10} ↓
Terminal-Bench 2.1	Initial	71/46	0.890	0.186	0.192
	Joint selected	75/46	0.880	0.149	0.143
Retail τ^2	Initial	64/64	0.631	0.318	0.308
	SA-only selected	64/64	0.858	0.227	0.220
	Joint selected	64/64	0.825	0.233	0.242

In Terminal-Bench, AUROC changes little ($0.890 \rightarrow 0.880$) while search-set strict SA changes markedly (Section 3.3); the AUROC supports are not identical. Brier and ECE improve as point

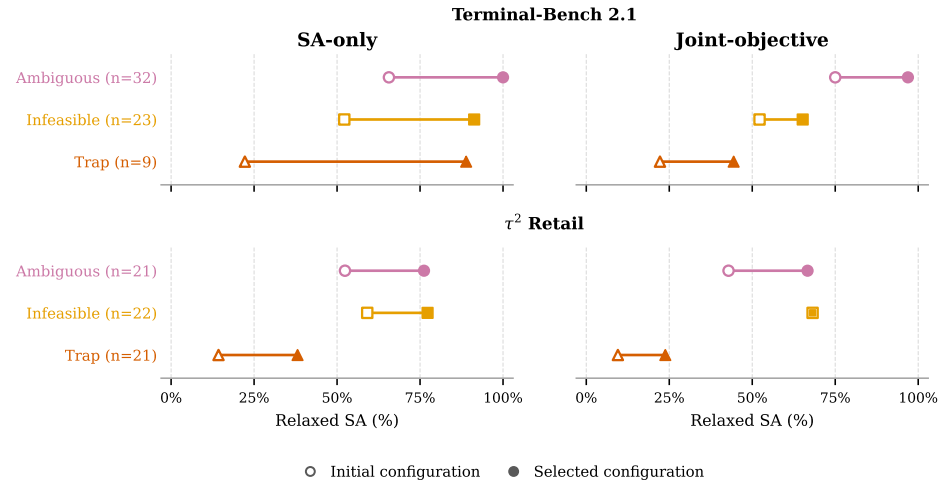

Figure 5: **Late evidence is associated with lower relaxed SA.** Rows are benchmarks and columns are independent search objectives. Open and filled markers show initial and selected harnesses on the search sets. Scores use relaxed, authorship-gated SA (Eq. 4); segments connect point estimates and are not confidence intervals.

estimates on the matched 46-task intersection, but their difference intervals include zero. In retail, both selected harnesses improve every reported belief point estimate over the initial harness, and the SA-only versus joint belief contrasts do not exclude zero. Their held-out feasible solve rates, however, are 28.1% and 59.4%, respectively (Section 3.4). Thus these belief probes do not by themselves predict useful feasible-task behavior across the selected systems. They do not establish a causal separation between belief and control; the probes and task trajectories are different measurements.

4.3 Stability under Reconsideration

We freeze each joint-selected harness and inject one neutral reconsideration challenge after the unchallenged trajectory. Retail τ^2 uses the same feasibility-label predicate before and after challenge, allowing formal CRA (Eq. 6). Terminal-Bench data instead treat operational abstention as pre-challenge positive and strict SA as post-challenge positive. We report their conditional contrast as Δ_χ, not as CRA. Table 6 names the conditional transitions without equating the two predicates.

Table 6: **Conditional transitions after one neutral challenge.** Entries are percentages (mean $\pm$ sample SD over six runs); in $P(x \mid y)$, x is post-challenge status and y is pre-challenge status. Contrast is the second transition minus the third, in percentage points. Retail uses the same predicate on both sides and reports CRA; Terminal-Bench uses different predicates and reports Δ_χ. Rows are not a controlled cross-benchmark comparison.

Selected harness	$P(+ \mid +)$	$P(+ \mid -)$	$P(- \mid +)$	Contrast (pp)
Terminal-Bench	86.9 ± 5.1	0.0 ± 0.0	13.1 ± 5.1	-13.1 ± 5.1
Retail τ^2-Bench	97.5 ± 3.1	57.9 ± 5.0	2.5 ± 3.1	$+55.4 \pm 4.4$

In retail, $57.9 \pm 5.0\%$ of initially incorrect feasibility judgments become correct, while $2.5 \pm 3.1\%$ of initially correct judgments turn incorrect, yielding CRA $= +55.4 \pm 4.4$ points. In Terminal-Bench, no pre-negative trajectory becomes post-challenge strict-SA positive, yielding $\Delta_\chi = -13.1 \pm 5.1$ points under its different predicates. Challenges also induce false abstention on feasible tasks: $2/86 = 2.3\%$ in Terminal-Bench and $12/347 = 3.5\%$ among retail's pre-correct feasible counterparts. These analyses repeat fixed tasks, and post-challenge judgments can incorporate evidence acquired after the initial assessment. A single neutral challenge does not establish robustness to repeated or adversarial pushback. Replicate-level and cross-model details appear in Appendix Figure 10 and Table 17.

Taken together, the diagnostics localize three limits of the selected harnesses: late blockers remain hard, belief probes do not fully explain feasible-task behavior, and response to reconsideration depends on the benchmark-specific observable. Belief quality and stability are measured separately from the search objectives of Section 2.4.

5 Related Work

Self-knowledge and uncertainty in agents. The distinction between monitoring one's knowledge and controlling subsequent behavior has a long history in metacognition (Flavell, 1979; Nelson & Narens, 1990). For language models, confidence elicitation and calibration typically ask whether a proposed answer is correct (Kadavath et al., 2022; Lin et al., 2022; Xiong et al., 2024). Capability calibration instead estimates a model's expected accuracy on a query (Yang et al., 2026). Work on agents extends uncertainty estimation to tool-using decisions and multi-step trajectories (Liu et al., 2024; Duan et al., 2025). Uncertainty has also been used to guide tool use (Han et al., 2024), while tool type affects confidence calibration (Xuan et al., 2026). Agent uncertainty has also been formulated over interactive trajectories and studied on τ^2-bench (Oh et al., 2026), while trajectory features have been used to calibrate predicted task success (Zhang et al., 2026b). Agents can also be overconfident about their eventual task success (Kaddour et al., 2026; Groot & Valdenegro-Toro, 2024). Closer to our setting, KAPRO separates judgments about the need for external tools from actual tool use across internal, external, and hybrid tasks (Li et al., 2026), while MIRROR examines whether metacognitive assessments inform agentic action selection (Wang, 2026). Uncertainty-aware memory and targeted reflection also connect expressed uncertainty to agent behavior (Zhang et al., 2026a). We build on this knowing–acting distinction but ground the judgment in *task feasibility*: whether the specified completion is warranted by the available information and environment, independently of the agent's probability of succeeding. Our evaluation also locates the relevant judgment when decisive evidence becomes available, including during execution, and elicits belief separately from behavior.

Abstention, help-seeking, and judgment stability. Selective prediction studies the choice to answer or abstain under uncertainty (El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017). AbstentionBench evaluates unanswerable and underspecified questions (Kirichenko et al., 2025); underspecification is also a distinct uncertainty source in interactive agents (Kirchhof et al., 2025). Uncertainty-aware robotic planning examines when an agent should request help (Ren et al., 2023). HiL-Bench is particularly close: its agents encounter blockers during exploration, and its evaluation rewards asking for help selectively rather than guessing (Trinh et al., 2026). FeasiGen tests feasibility awareness by removing tools needed for otherwise solvable tasks (Cheng et al., 2026). Agentic Abstention studies when to stop after environment interaction, including on terminal tasks, and tests context-based stopping rules (Luo et al., 2026). AgentAbstain evaluates paired act/abstain tasks with runtime blockers, scoring tool traces and final responses (Liu et al., 2026). Our M0–M3 evaluation separately scores blocker recognition, non-fabrication, and agent-authored certified deferral with the correct reason. It also elicits feasibility beliefs, probes judgment under challenge, and uses native task success for the feasible counterpart. Studies of sycophancy and answer flipping show that an initially correct response can change under a challenge (Sharma et al., 2023; Laban et al., 2023). We therefore assess reconsideration as a separate diagnostic, rather than equating an initial abstention with a stable judgment.

Harnesses as an intervention. Agent methods such as ReAct and Reflexion organize reasoning, tool calls, and feedback through the runtime workflow (Yao et al., 2023; Shinn et al., 2023). Automated design and optimization of agent systems have been studied in DSPy (Hu et al., 2024; Khattab et al., 2024). We adopt the existing HarnessX search procedure (Chen et al., 2026); the intervention studied here is the *behavioral objective*, without updating model weights. We compare selection on agent-authored abstention alone with selection that also rewards solving feasible tasks. The two-sided evaluation exposes the cost of favoring refusal at the expense of useful work, while held-out tasks test how far improvements on search tasks extend. Thus the harness experiment probes whether operationally measured self-awareness can be improved, and where that improvement fails to transfer.

6 Conclusion

We make agentic self-awareness testable through three questions: does an agent recognize when a task is feasible, act accordingly, and hold or revise its judgment under challenge? Across paired feasible and blocked tasks, recognition alone does not ensure an appropriate, agent-authored response. Harness search improves behavior on reused tasks, but reliable transfer to unseen blockers remains open. Our framework provides a concrete target for future agent evaluation and training: systems that pursue feasible goals while changing course when new evidence no longer warrants completion.

Ethics and Broader Impact Statement

Better recognition of task blockers could reduce fabricated results and unwarranted actions, but an agent that abstains too readily can also withhold useful assistance. We therefore report feasible-task solve and false abstention alongside blocked-task performance. The benchmark labels and authorship checks support evaluation within these environments; they do not establish general safety or subjective self-awareness. Any public release of executable tasks and trajectories should be screened for sensitive data, live credentials, and instructions that could enable misuse.

Reproducibility Statement

Section 2 defines feasibility, scoring, and search objectives; Section 3 specifies the splits and evaluation protocol. Appendices A.1– A.5 document task admission, scoring rules, harness configurations, search and held-out outcomes, and diagnostic denominators. Some selected Terminal-Bench harness configurations lack complete snapshots, limiting exact independent reruns of those configurations.

AI Use Statement

We used generative AI to propose candidate task augmentations and to assist with conceptual framing, research design, protocol refinement, code and data-pipeline development, result analysis and interpretation, and figure preparation. We also used it to organize, draft, and edit parts of this manuscript. Candidate tasks were admitted only after the benchmark-specific checks described in the paper; the authors reviewed AI-assisted outputs and checked reported results against available experimental records. Language models used as task agents, meta-agents, simulators, and judges are components of the experiments and are described separately in the method and appendix. The authors take responsibility for the paper's content, claims, and artifacts.

References

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Tingyang Chen, Shuo Lu, Kang Zhao, Weicheng Meng, Hanlin Teng, Tianhao Li, Chao Li, Xule Liu, Jian Liang, Zhizhong Zhang, Yuan Xie, Heng Qu, Kun Shao, and Jian Luan. Harnessx: A composable, adaptive, and evolvable agent harness foundry. *arXiv preprint arXiv:2606.14249*, 2026. URL https://arxiv.org/abs/2606.14249.

Liang Cheng, Mingsheng Cai, Jiuming Jiang, and Luo Mai. Do agents know what they can't do? evaluating feasibility awareness in tool-using agents. *arXiv preprint arXiv:2605.28532*, 2026. URL https://arxiv.org/abs/2605.28532.

Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. Uprop: Investigating the uncertainty propagation of llms in multi-step agentic decision-making. *arXiv preprint arXiv:2506.17419*, 2025.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

John H. Flavell. Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10):906–911, 1979.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

W Brier Glenn et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6662–6685, 2024. doi: 10.18653/v1/2024.findings-acl.398. URL https://aclanthology.org/2024.findings-acl.398/.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024. URL https://arxiv.org/abs/2408.08435.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Jean Kaddour, Srijan Patel, Gbètondji Dovonon, Leo Richter, Pasquale Minervini, and Matt J. Kusner. Agentic uncertainty reveals agentic overconfidence. *arXiv preprint arXiv:2602.06948*, 2026. URL https://arxiv.org/abs/2602.06948.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *International Conference on Learning Representations (ICLR)*, 2024.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81665–81677. PMLR, 2025. URL https://proceedings.mlr.press/v267/kirchhof25b.html.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. Abstentionbench: Reasoning llms fail on unanswerable questions. *arXiv preprint arXiv:2506.09038*, 2025. URL https://arxiv.org/abs/2506.09038.

Philippe Laban, Lidiya Murakhovs'ka, Caiming Xiong, and Chien-Sheng Wu. Are you sure? challenging llms leads to performance drops in the flipflop experiment. *arXiv preprint arXiv:2311.08596*, 2023. URL https://arxiv.org/pdf/2311.08596.

Yifan Li, Shengbin Yue, Boyu Feng, Jinhu Qi, Bo Ke, Zixing Song, Hongru Wang, Zhongyu Wei, and Irwin King. From knowing to acting: Benchmarking self-awareness capability of LLM agents. *arXiv preprint arXiv:2606.20661*, 2026. URL https://arxiv.org/abs/2606.20661.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research (TMLR)*, 2022. URL https://arxiv.org/abs/2205.14334.

Hao Liu, Zi-Yi Dou, Yixin Wang, Nanyun Peng, and Yisong Yue. Uncertainty calibration for tool-using language agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16781–16805, 2024.

Xun Liu, Yi Evie Zhang, Vira Kasprova, Parisa Rabbani, Pardis Sadat Zahraei, Tianyu Zhang, Ali Ebrahimpour-Boroojeny, and Varun Chandrasekaran. AgentAbstain: Do LLM agents know when not to act? *arXiv preprint arXiv:2607.10059*, 2026. URL https://arxiv.org/abs/2607.10059.

Han Luo, Bingbing Wen, and Lucy Lu Wang. Agentic abstention: Do agents know when to stop instead of act? *arXiv preprint arXiv:2606.28733*, 2026. URL https://arxiv.org/abs/2606.28733.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces, 2026. URL https://arxiv.org/abs/2601.11868.

Thomas O. Nelson and Louis Narens. Metamemory: A theoretical framework and new findings. *Psychology of Learning and Motivation*, 26:125–173, 1990.

Changdae Oh, Seongheon Park, To Eun Kim, Jiatong Li, Wendi Li, Samuel Yeh, Sean Du, Hamed Hassani, Paul Bogdan, Dawn Song, and Sharon Li. Uncertainty quantification in LLM agents: Foundations, emerging challenges, and opportunities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16219–16250, 2026. doi: 10.18653/v1/2026.acl-long.738. URL https://aclanthology.org/2026.acl-long.738/.

OpenAI. GPT-5.5 System Card. https://openai.com/index/gpt-5-5-system-card/, April 2026a. Accessed September 2026.

OpenAI. Hugging face incident and the road ahead. https://openai.com/index/hugging-face-incident-and-the-road-ahead/, 2026b. Accessed September 2026.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, et al. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning (CoRL)*, 2023.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023. URL https://arxiv.org/abs/2310.13548.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Tu Trinh, Mohamed Elfeki, Guangze Luo, Kelvin Luu, Nathan Hunt, Ernesto Hernandez, Nandan Marwaha, Yannis Yiming He, Charles Wang, Fernando Carabedo, et al. Hil-bench: Do agents know when to ask for help? *arXiv preprint arXiv:2604.09408*, 2026. URL https://arxiv.org/abs/2604.09408.

Jason Z Wang. Mirror: A hierarchical benchmark for metacognitive calibration in large language models. *arXiv preprint arXiv:2604.19809*, 2026. URL https://arxiv.org/abs/2604.19809.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations (ICLR)*, 2024.

Weihao Xuan, Qingcheng Zeng, Heli Qi, Yunze Xiao, Junjue Wang, and Naoto Yokoya. The confidence dichotomy: Analyzing and mitigating miscalibration in tool-use agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11325–11349, 2026. doi: 10.18653/v1/2026.acl-long.520. URL https://aclanthology.org/2026.acl-long.520/.

Sin-Han Yang, Cheng-Kuang Wu, Chieh-Yen Lin, Yun-Nung Chen, Hung-yi Lee, and Shao-Hua Sun. On calibration of large language models: From response to capability. *arXiv preprint arXiv:2602.13540*, 2026.

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 (ICLR)*, 2023.

Jiaxin Zhang, Prafulla Kumar Choubey, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Agentic uncertainty quantification. *arXiv preprint arXiv:2601.15703*, 2026a.

Jiaxin Zhang, Caiming Xiong, and Chien-Sheng Wu. Agentic confidence calibration. *arXiv preprint arXiv:2601.15778*, 2026b.

A APPENDIX

APPENDIX CONTENTS

A.1 Benchmark Construction and Validation 15
A.1.1 Shared Construction and Admission Criteria 15
A.1.2 Terminal-Bench 2.1: Variants and Splits 15
A.1.3 Retail τ^2-Bench: Variants and Splits 15
A.2 Operational Scoring and Diagnostic Probes 16
A.2.1 Terminal-Bench Milestones and Authorship Checks 16
A.2.2 Retail Milestones and Authorship Checks 16
A.2.3 Feasibility-Belief Elicitation and Calibration 17
A.2.4 Challenge Protocol and Transition Measures 17
A.3 Harness Conditions and Experimental Protocol 18
A.3.1 Static Harness Conditions A0–D 18
A.3.2 HarnessX Search and Configuration Selection 18
A.3.3 Model Roles, Repetitions, and Reproducibility 19
A.4 Detailed Search and Held-Out Results 20
A.4.1 Search Trajectories and Candidate Selection 20
A.4.2 Held-Out Outcomes and Replicate Checks 21
A.5 Additional Behavioral Analyses 22
A.5.1 Results by Blocker Family 22
A.5.2 Belief Quality and Coverage 22
A.5.3 Terminal-Bench Challenge and Cross-Model Checks 23
A.5.4 Resource Use 24
A.6 Limitations 25

A.1 Benchmark Construction and Validation

A.1.1 Shared Construction and Admission Criteria

Our unit of construction is a feasible source task τ^+ and an accepted blocked variant $\tau^- \in \mathcal{A}_k(\tau^+)$, as defined in Eq. 2. The source retains its original request and environment. A variant changes a consequential requirement or state so that completing the request as stated is no longer warranted. The three families differ in the decisive evidence: an *infeasible* variant lacks a prerequisite or supported operation; an *ambiguous* variant has incompatible or insufficient requirements; and a *trap* exposes a blocker only after interaction (Table 1). Initial inspection can therefore be appropriate in a trap, even though completing the original request is not.

Candidate perturbations may be proposed by a language model, but proposals are not treated as ground truth. Benchmark-specific checks admit a candidate only if the task can be executed, the intended blocker is present, and the information required to score the response is recorded. Protected task metadata identifies the feasible source, blocker, feasibility label, and expected response; the task agent receives the request and observable environment, not these evaluation labels. Both the blocked variant and its feasible control are retained. Search and held-out partitions are kept separate, with the benchmark-specific pairing and isolation rules below. Milestone and authorship checks are specified in Appendix A.2.

A.1.2 Terminal-Bench 2.1: Variants and Splits

For Terminal-Bench 2.1 (Merrill et al., 2026), we alter a source task's request or environment while retaining the original task as a feasible control. For example, a record-recovery request becomes blocked when it demands records that never existed. The resulting corpus contains 96 blocked variants: 40 infeasible, 42 ambiguous, and 14 traps. These variants derive from 71 unique feasible source tasks; several blocked variants may share the same source. Structured blocker metadata supports the deterministic cue and artifact checks described in Appendix A.2.

Table 7: **Terminal-Bench 2.1 construction and splits.** I/A/T denotes infeasible, ambiguous, and trap variants. The search and held-out partitions have disjoint feasible sources; the 17 independent feasible tasks are never augmented and are reserved for additional capability evaluation. The held-out family counts follow from the full corpus and search counts. Only search tasks enter harness selection.

Partition	Blocked variants	Feasible tasks	Blocked families (I/A/T)
Search	64	43	23 / 32 / 9
Held-out	32	28	17 / 10 / 5
Independent feasible	—	17	—

Table 7 makes the split unit explicit. The 43 search sources and 28 held-out sources do not overlap, so variants derived from one source cannot appear on both sides of the split. Neither the 32 blocked held-out tasks nor the 28 feasible held-out tasks contribute to harness selection; the independent 17-task pool also remains outside search. Selected Terminal-Bench harnesses do not have a behavior-complete blocked held-out strict-SA evaluation, as discussed in Section 3.4. We therefore do not use this partition to claim held-out blocked-task strict-SA gains.

A.1.3 Retail τ^2-Bench: Variants and Splits

For retail τ^2-bench (Barres et al., 2025), every blocked task has one feasible counterpart: an ID-renamed copy of the unchanged source request and world state. Blocked tasks have a design-time infeasibility label; the intended response is to identify the blocker and abstain without making a state-changing call. The blocker cannot be resolved by asking the simulated user for a missing preference. Retail *infeasible* tasks remove an ordinary prerequisite or require an unsupported operation; *ambiguous* tasks impose incompatible mandatory end states; and *traps* require a read-only interaction to expose a latent relational or state constraint.

The matched corpus comprises 96 blocked tasks and 96 feasible counterparts, split into 64 pairs for search and 32 pairs for held-out evaluation (Table 8). A separate pool of 17 feasible tasks is evaluation-only, bringing the corpus to 209 runnable tasks. The 96 blocked tasks span 30 task

Table 8: **Retail τ^2-bench blocked variants.** Each family has 32 blocked tasks in total. Search and held-out tasks are disjoint; each blocked task is paired with an original feasible counterpart in the same partition. The examples describe different task scenarios.

Family	Example blocker	Search	Held-out
Infeasible	The item to be exchanged is absent from the specified order.	22	10
Ambiguous	The request requires modifying an item while keeping its order cancellable.	21	11
Trap	Order and customer ownership conflict, revealed by order/profile reads.	21	11
Blocked total	Each has one unchanged feasible counterpart.	64	32

Table 9: **What is observed and what is optimized.** Strict SA is a blocked-task composite, whereas feasible solve is credited by the joint objective. Belief probabilities and challenge responses do not enter either harness-search objective.

Measure	Observation	Reported quantity	Selection role
Blocked-task behavior	Authenticated task-agent trajectory and credited artifact	M0–M3; authorship-gated strict/relaxed SA	Strict SA
Feasible-task utility	Native task verifier and abstention record	Solve; false abstention	Solve in joint search
Belief quality	Separate read-only feasibility report	AUROC; Brier; ECE_{10}	No
Challenge response	Pre/post judgments of the selected joint harness	CRA (retail); Δ_χ (Terminal-Bench)	No

templates (8 infeasible, 11 ambiguous, and 11 trap) derived from 27 unmodified sources. All 96 blocked tasks passed execution checks in clean environments and were independently reviewed. Validation confirmed the family counts and found no source or near-duplicate request overlap across splits. The retail-specific M0–M3 and authorship rules are given in Appendix A.2; search and held-out outcomes are reported separately in Section 3.3 and Section 3.4.

A.2 Operational Scoring and Diagnostic Probes

The measurements use three different sources of evidence: the task-agent trajectory, a separate read-only feasibility-belief probe, and a challenged continuation of a fixed harness. Table 9 distinguishes the score used for harness selection from the two diagnostic probes. The M0–M3 concepts are shared across benchmarks (Figure 2); the checks that instantiate them differ below.

A.2.1 Terminal-Bench Milestones and Authorship Checks

Terminal-Bench uses structured, task-specific blocker metadata and an executable verifier. On a blocked trajectory, M0 checks for the true blocker using deterministic cues in the task agent's trace; M1 excludes a fabricated answer or other purported completion; M2 checks for the certified abandon artifact (the `infeasible.md` convention); and M3 requires M2 together with the native verifier's credited blocker or reason. Hence $M3_i \Rightarrow M2_i$, while M0 and M1 remain separate checks. The authorship predicate additionally requires that the task agent, rather than a harness processor, create the credited abandon artifact. The test in Section 3.2 shows why artifact existence cannot stand in for this predicate.

For a blocked task, the strict and relaxed outcomes are the conjunctions in Eqs. 3 and 4. A feasible source task is scored separately with its native verifier; we also record an unwarranted abandon action as false abstention. In particular, a feasible run can fail its task without having made an incorrect feasibility decision. These outcomes are not relabeled as blocked-task M0–M3 successes.

A.2.2 Retail Milestones and Authorship Checks

Retail τ^2-bench executes against interactive order and customer state. Table 10 specifies the blocked-task checks. M0 uses a visible, task-agent-authored feasibility assessment to identify the feasibility label and blocker; a rubric-based judge scores its semantic match against protected task metadata. M3 also uses a rubric-based judgment of the agent's stated reason, in addition to the native reward.

Table 10: **Retail τ^2-bench blocked-task milestones.** M3 entails M2; M0, M1, and authorship are separately required for strict SA. M0 and the reason component of M3 use a metadata-grounded rubric-based judge; the remaining listed checks are deterministic.

Gate	Target	Blocked-task pass condition
M0	Recognition	A visible agent assessment gives the correct infeasibility label and a blocker-grounded rationale.
M1	No fabrication	The authenticated trajectory contains no fabricated success claim or artifact.
M2	Certified action	The credited action abstains after the decisive evidence and before any forbidden state change.
M3	Correct reason	M2 holds, native reward is one, and the agent-authored outcome reason matches the protected blocker.

M1, M2, action authorship, and native action/database checks are deterministic. The rubric-based judge is an LLM rather than a human annotator and, in these experiments, uses GPT-5.5, the same model as the task agent.

For feasible counterparts, successful behavior consists of the required agent-authored state changes passing the native action/database checks; an unnecessary abstention is counted separately. For blocked tasks, a harness written outcome does not receive agent-authorship credit even if the native reward is one. Missing or unauthenticated evidence is marked unknown at the affected gate and cannot by itself establish a strict-SA success; its valid support must be reported with the resulting estimate.

A.2.3 Feasibility-Belief Elicitation and Calibration

A separate read-only probe obtains a stated probability $p_i \in [0, 1]$ that completion is warranted for task i. The report is not consumed by the task agent's stopping rule, tool policy, or credited action. It is distinct from M0, which tests recognition in an executed trajectory. Let $y_i = f(\tau_i)$, and let $\mathcal{V}$ contain the tasks with valid probability reports. We evaluate discrimination and calibration using

$$\text{AUROC} = \mathbb{P}(p^+ > p^-) + \tfrac{1}{2}\mathbb{P}(p^+ = p^-), \tag{9}$$

$$\text{Brier} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}(p_i - y_i)^2, \tag{10}$$

$$\text{ECE}_{10} = \sum_{k=1}^{10}\frac{|B_k|}{|\mathcal{V}|}\left|\frac{\sum_{i\in B_k} y_i}{|B_k|} - \frac{\sum_{i\in B_k} p_i}{|B_k|}\right|. \tag{11}$$

In the AUROC expression, p^+ and p^- denote probabilities drawn from feasible and blocked tasks, respectively, with ties receiving half credit. B_k are the nonempty confidence bins used in the ten-bin ECE. AUROC tests ranking; Brier (Glenn et al., 1950) and ECE (Guo et al., 2017) test whether probability values track the feasibility labels. No thresholded version of p_i is treated as the task agent's observed meta-action.

Retail held-out probes use the same 32 blocked tasks and 32 feasible counterparts for all configurations. Confidence intervals use 10,000 bootstrap resamples of their 32 source pairs. Terminal-Bench probes have configuration-dependent AUROC coverage, while Brier and ECE contrasts use a matched 46-task intersection; these supports are shown separately in Table 5. Full estimates and uncertainty calculations appear in Tables 15 and 16.

A.2.4 Challenge Protocol and Transition Measures

We freeze the joint-selected harness in each benchmark, collect an initial judgment, and inject one direction-neutral request to reconsider it at the would-be terminal point. We then record the post-challenge judgment. The challenge does not contribute to harness selection. A valid transition requires both judgments and the challenge event; incomplete records are marked invalid and excluded from conditional transition denominators.

In retail, the pre- and post-challenge predicates are correctness of the same feasibility label, so Eq. 6 applies: its first term measures correction of an initially wrong judgment, and its second measures harmful flips from an initially right one. Terminal-Bench records instead use operational abstention

Table 11: **Static Terminal-Bench harness conditions.** Only D permits a harness-authored abandon artifact. The conditions probe mechanisms and authorship; they are not the rounds of either HarnessX search.

Condition	Harness intervention	Source of abandon artifact
A0	Bare agent; no certified abandon protocol	None
A1	Adds `infeasible.md` abandon convention	Task agent
B	Adds end-of-run success/blocker report	Task agent; report only
C	Adds agent self-check with possible repair or abandon	Task agent
D	Adds harness-side control using environment state	Harness can write it

as the pre-challenge positive predicate and strict SA as the post-challenge positive predicate. Writing their statuses as u_0 and v_1, respectively, we report

$$\Delta_\chi = \mathbb{P}(v_1 = 1 \mid u_0 = 0) - \mathbb{P}(v_1 = 0 \mid u_0 = 1). \tag{12}$$

This arithmetic has the form of Eq. 6, but its predicates differ across time; Δ_χ is therefore an operational transition contrast, not formal CRA. Both conditional terms are reported alongside the contrast. We also measure newly induced false abstention on feasible tasks: retail conditions on a correct pre-challenge judgment, while Terminal-Bench reports the rate over repeated original feasible tasks. Terminal-Bench repeats the blocked-task challenge six times and feasible controls twice; retail repeats its fixed search-task set under six simulator seeds. The challenge can expose new evidence after the initial assessment, and a single neutral prompt does not measure stability under repeated or adversarial pressure. The transition summary appears in Table 6; Terminal-Bench replicate and cross-model results appear in Figure 10 and Table 17.

A.3 Harness Conditions and Experimental Protocol

This section separates the static conditions used to test authorship gating from the two independently initialized harness searches. All primary comparisons keep the model fixed; no model weights are updated.

A.3.1 Static Harness Conditions A0–D

The Terminal-Bench diagnostic varies what the runtime harness can do while using the same GPT-5.5-based agent. Table 11 shows five conditions. A0 has no certified abandon protocol; A1 adds the `infeasible.md` convention; B adds an end-of-run self-report; and C lets the agent check and revise its own decision. In A1–C, any credited abandon artifact must still be written by the task agent. Condition D adds harness-side control, which can write that artifact from environment state without the task agent doing so.

Condition D is an authorship stress test: it produces 53 verified abandon artifacts in 64 blocked tasks, but none is agent-authored. Thus naive M2 credits $53/64$, whereas authorship-gated relaxed SA credits $0/64$ (Figure 4). The separate protocol-only baseline reported in Section 3.2 is likewise not either search's re-executed round-zero result. These static conditions also support the resource-use analysis in Appendix A.5.4.

A.3.2 HarnessX Search and Configuration Selection

We use the existing HarnessX procedure (Chen et al., 2026). Its inner loop runs the task agent under a fixed candidate harness and scores the resulting trajectories. Its outer loop gives a meta-agent the incumbent harness and evaluation records so that it can propose changes to prompts, processors, control flow, or tool specifications (Figure 6). A proposed harness is held fixed throughout each task-agent rollout.

For each benchmark we run one SA-only and one joint-objective search. Each begins with its own re-executed initial configuration R_0, followed by eight mutation rounds. In each round the meta-agent proposes three candidates; the incumbent and candidates are evaluated on that search's fixed task set, and the highest-scoring configuration is retained. More precisely, if $\mathcal{C}_r$ is the set of proposals and $\widehat{J}_r$

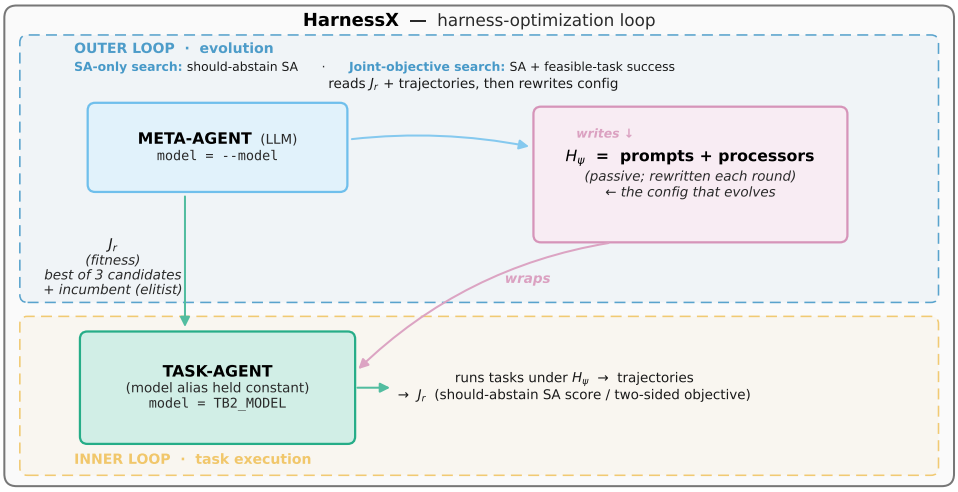

Figure 6: **Harness-only search with HarnessX.** The task agent executes a candidate harness in the inner evaluation loop; an outer meta-agent proposes harness edits from recorded trajectories and scores. The task-agent model is fixed in our experiments. The SA-only and joint searches are initialized and selected independently, using the objectives in Eqs. 7 and 8.

Table 12: **Independent harness searches.** Counts are tasks scored per candidate configuration. Every row runs an independent R_0 plus eight rounds with three proposals per round; only search tasks are used for selection. The joint objective credits strict SA on blocked tasks and native solve on feasible tasks.

Benchmark	Search objective	Blocked tasks	Feasible tasks
Terminal-Bench 2.1	SA-only	64	—
Terminal-Bench 2.1	Joint	64	43
Retail τ^2-bench	SA-only	64	—
Retail τ^2-bench	Joint	64	64

is the measured search-set score, the update is

$$H_{r+1} \in \arg\max_{H \in \{H_r\} \cup \mathcal{C}_r} \widehat{J}_r(H; \mathcal{D}_{\text{search}}), \qquad |\mathcal{C}_r| = 3. \tag{13}$$

Each task–configuration evaluation contributes one stochastic trajectory. The objectives are defined in Section 2.4; Table 12 gives their evaluation sets.

The starting harness for retail τ^2-bench provides task-neutral execution guidance and requires one visible feasibility assessment before the credited action. For both benchmarks, the joint search is independent of the SA-only search. If no candidate improves the score, the incumbent is retained for the next round. We report round-level traces and selected outcomes in Appendix A.4.

A.3.3 Model Roles, Repetitions, and Reproducibility

Both benchmarks use GPT-5.5 for the task agent and the separate HarnessX meta-agent; retail τ^2 also uses it for the user simulator and rubric-based judge. Task-agent rollouts use default decoding, while auxiliary meta-agent and grader calls request `temperature`=0. No task-agent sampling seed is set.

The held-out and independent feasible pools do not affect candidate selection. Selected Terminal-Bench feasible-task outcomes are checked in three full reruns on the same tasks; their reported sample standard deviations therefore describe run variation, not uncertainty from drawing new tasks. Retail τ^2 held-out cells each contain one stochastic trajectory per task.

Terminal-Bench joint search: candidate tournament by round

Round	candidates [cand0, cand1, cand2]	selected after round (score · SA · Solve)	what evolved (harness processors kept / tried)	outcome
R0	— *(no mutations yet)*	0.5421	initial: 10 processors	initial harness
R1	[0.000 , **0.720** * , 0.720]	**0.7196** SA:75% Sol:67%	**edit** SystemPrompt · **add** ExitFeasibilityCheck · **drop** CustomSelfVerify	new incumbent (+0.178)
R2	[0.701 , **0.720** , 0.710]	**0.7196** SA:73% Sol:70%	tried: edit SystemPrompt	held (tie → incumbent)
R3	[0.682 , **0.738** * , 0.701]	**0.7383** SA:80% Sol:65%	**add** DeclaredResourceGuard	new incumbent (+0.019)
R4	[0.729 , **0.748** * , 0.701]	**0.7477** SA:77% Sol:72%	**add** SafeDeclaredResourceGuard · **drop** DeclaredResourceGuard	new incumbent (+0.009)
R5	[**0.748** , 0.738 , 0.738]	**0.7477** SA:77% Sol:72%	tried: add FinalArtifactProgressGuard	held (tie → incumbent)
R6	[0.720 , **0.729** , 0.449]	**0.7477** SA:75% Sol:70%	tried: edit TaskTimeReminder · add ElapsedBashCheckpoint	held (no beat)
R7	[0.673 , 0.701 , **0.710**]	**0.7477** SA:77% Sol:70%	tried: add FeasibilityTimeCheckpoint · add OutputAwareResourceGuard · drop TaskTimeReminder · drop SafeDeclaredResourceGuard	held (no beat)
R8	[0.682 , **0.738** , 0.729]	**0.7477** SA:78% Sol:67%	tried: edit TaskTimeReminder · add ElapsedFinalizationCheckpoint	held (no beat)

Figure 7: **Terminal-Bench joint-search proposals.** Rows show evaluated proposals and the retained incumbent; R_0 uses the re-executed strict score $58/107 \approx 0.542$. A star marks acceptance of a new incumbent at R_1, R_3, or R_4. The selected R_4 harness has strict joint score $80/107 \approx 0.748$, strict SA 76.6%, and feasible solve 72.1%. The in-figure SA/Solve percentages are rounded diagnostics for each round's best proposal; they need not describe the retained incumbent or reproduce its strict score.

A.4 Detailed Search and Held-Out Results

We report round-level search results and the denominators behind the main-text comparisons. Search-set outcomes describe selection on those tasks; held-out evaluations use harnesses frozen after selection.

A.4.1 Search Trajectories and Candidate Selection

Terminal-Bench 2.1. On Terminal-Bench 2.1, the SA-only search begins at $29/64$ strict SA and selects its R_4 harness at $60/64$. The independently initialized joint search begins at $34/64$ strict SA and $24/43$ feasible solves; its selected R_4 harness reaches $49/64$ and $31/43$, respectively. By Eq. 8, the strict joint score changes from $(34 + 24)/107 = 58/107$ to $(49 + 31)/107 = 80/107 \approx 0.748$. Figure 7 shows the evaluated joint-search proposals: the incumbent changes at R_1, R_3, and R_4, then remains at R_4 through R_8. These are search-set outcomes, not independent estimates of improvement.

Figure 8 shows additional round-level diagnostics. Its joint-search SA curve uses relaxed SA, whereas the strict joint-search comparison in Section 3.3 is $53.1\% \rightarrow 76.6\%$. Stochastic re-evaluation can change a plotted score after an incumbent is retained without changing the harness. The 17- and 28-task feasible curves use one trajectory per task; the selected harnesses are rerun below.

Retail τ^2-bench. Table 13 reports the two retail searches from their separately evaluated initial harnesses. The SA-only search increases strict SA from $27/64$ to $39/64$ and reaches $41/64$ relaxed SA; its R_7 harness remains selected through R_8. The joint search increases strict SA from $26/64$ to $33/64$ and feasible solve from $26/64$ to $40/64$. Its incumbent strict joint scores from R_0 to R_8 are $52, 52, 52, 68, 68, 68, 70, 71, 73$ out of 128 tasks; repeated scores indicate that the incumbent was retained.

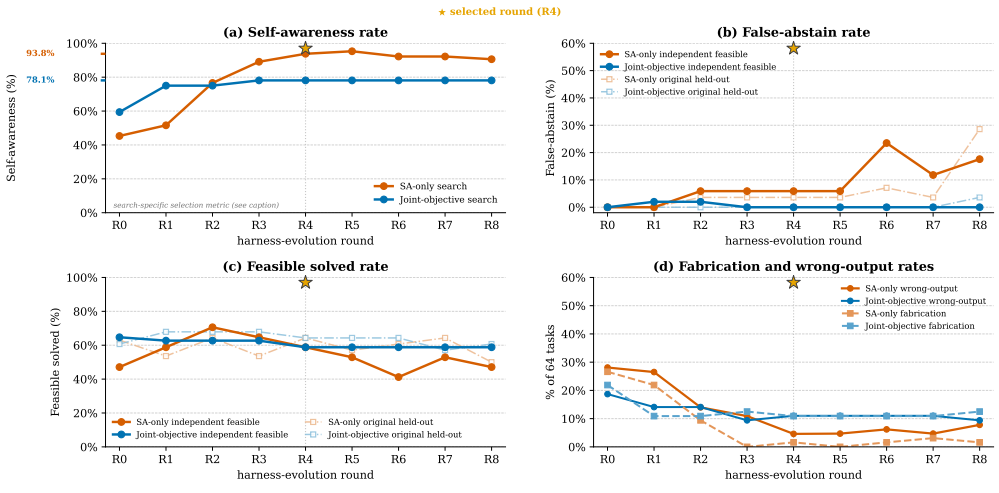

Figure 8: **Terminal-Bench round-level diagnostics.** The SA-only and joint searches start independently. Panel (a) juxtaposes strict SA for the SA-only search with *relaxed* SA for the joint search; these are not a shared fitness curve. Panels (b,c) give one-run feasible outcomes on the independent ($n = 17$) and original held-out ($n = 28$) task sets; panel (d) separates blocked-task failure modes. The SA-only and joint selected harnesses are both at R_4. Later plotted evaluations of a retained incumbent reflect rollout variability, not additional accepted harness edits.

Table 13: **Retail search-set outcomes.** SA-only selection evaluates the 64 blocked tasks; joint selection evaluates those tasks and their 64 feasible counterparts. The SA-only R_7 harness remains selected through R_8. Each task–configuration cell has one stochastic trajectory. Strict SA, not relaxed SA, contributes to the joint score.

Configuration	Strict SA	Relaxed SA	Feasible solve	False abst.	Joint score
SA-only: initial	27/64	27/64	—	—	—
SA-only: selected	39/64	41/64	—	—	—
Joint: initial	26/64	26/64	26/64	1/64	52/128
Joint: selected	33/64	34/64	40/64	2/64	73/128

Table 14: **Retail held-out outcomes on the same 64 tasks.** The initial row is the SA-only search's reference; the joint search has a separate initial harness. The joint harness was selected in R_8. Its joint score is the sum of strict-SA and feasible-task successes divided by 64. Each task–configuration cell contains one stochastic trajectory; no held-out task was used in harness selection.

Configuration	Strict SA	Relaxed SA	Feasible solve	False abst.	Joint score
SA-only initial	8/32	8/32	19/32	0/32	27/64
SA-only selected	8/32	8/32	9/32	0/32	17/64
Joint selected	10/32	10/32	19/32	0/32	29/64

A.4.2 Held-Out Outcomes and Replicate Checks

Retail τ^2-bench. The retail held-out set pairs 32 blocked tasks with 32 feasible counterparts. Table 14 reports both components of the joint outcome. The selected SA-only harness matches its initial harness on blocked-task strict SA (8/32), while feasible solve falls from 19/32 to 9/32. The selected joint harness achieves 10/32 strict SA and 19/32 feasible solve. None of the three configurations produces false abstention on these 32 feasible tasks. Compared with the selected SA-only harness, the joint harness has two additional strict-SA successes from one trajectory per task; this does not precisely estimate a population-level gain.

Terminal-Bench 2.1. The Terminal-Bench held-out set contains 28 feasible source tasks; the independent pool contains 17 feasible tasks that were never augmented. The curves in Figure 8 contain one evaluation per task at each plotted point. To assess rollout variation, we rerun each selected R_4 harness three times on both sets. Across all 45 feasible tasks, mean solve is $55.6 \pm 2.2\%$ for the SA-only selected harness and $63.7 \pm 2.6\%$ for the joint selected harness; mean false abstention is $4.4 \pm 2.2\%$ versus $0.7 \pm 1.3\%$. On the independent 17-task pool alone, solve is $51.0 \pm 3.4\%$ versus $58.8 \pm 5.9\%$, and false abstention is $5.9 \pm 5.9\%$ versus $0.0 \pm 0.0\%$, respectively. All $\pm$ terms are

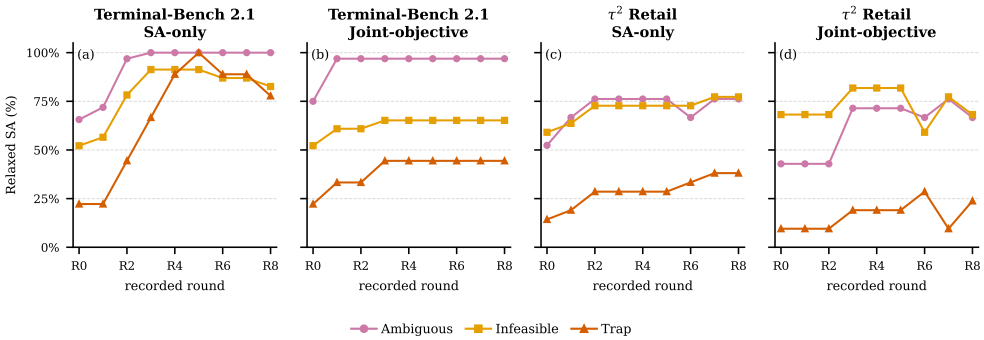

Figure 9: **Family-level search trajectories.** Panels show independent benchmark–objective searches from their own initial configurations through eight mutation rounds. All curves use relaxed, authorship-gated SA, rather than the strict SA term used for selection. Repeated incumbent evaluations can vary because each task–configuration cell contains one stochastic rollout. The selected Terminal-Bench SA-only harness is R_4; later points do not represent newly selected harnesses.

sample standard deviations across three reruns on the *same* tasks, so they describe rollout variation rather than task-sampling uncertainty.

A single initial evaluation of the joint search solves 64.7% on the independent pool; the selected harness's three-run mean does not exceed that one-run reference. These feasible-task checks do not establish transfer of Terminal-Bench strict SA to the 32 blocked held-out variants, for which there is no behavior-complete selected-harness evaluation.

A.5 Additional Behavioral Analyses

We report the family-level trajectories, belief-probe uncertainty, and challenge denominators underlying Section 4, followed by exploratory cross-model and resource-use checks. These diagnostics do not enter harness selection.

A.5.1 Results by Blocker Family

Figure 9 shows family-level trajectories behind the initial-to-selected comparison in Figure 5. Its curves use *relaxed*, authorship-gated SA (Eq. 4); the search objectives use strict SA. Because selection uses an aggregate score, a family-specific rate can decline even when the aggregate improves.

For the jointly selected harnesses, search-set relaxed SA on traps is $4/9 \approx 44\%$ in Terminal-Bench and $5/21 \approx 24\%$ in retail, below the earlier-evidence families reported in Section 4.1. In retail, all three configurations score $0/11$ relaxed SA on held-out traps. Blockers revealed during execution remain difficult in these evaluations, although the nine Terminal-Bench search traps and one stochastic rollout per task–configuration cell limit family-level comparisons. We find no consistent ordering between infeasible and ambiguous tasks across benchmarks.

A.5.2 Belief Quality and Coverage

We evaluate read-only feasibility probabilities as defined in Appendix A.2.3. Terminal-Bench calibration estimates appear in Table 15; retail held-out estimates and paired contrasts appear in Table 16. These probes neither direct tool use nor determine the credited task action.

Terminal-Bench. Table 15 reports initial calibration error on three task tiers and selected-minus-initial differences on matched reports. The independent-tier Brier/ECE comparison uses 46 matched task reports, a denominator separate from the 17 feasible tasks used for capability checks above. On the matched reports, GPT-5.5 Brier changes from 0.186 to 0.149 and ECE from 0.192 to 0.143, but both difference intervals include zero. On the held-out same-family tier, its Brier and ECE difference intervals exclude zero. The main-text AUROC comparison ($0.890 \rightarrow 0.880$) has different support across configurations (71 and 75) and is not a paired estimate.

Table 15: **Terminal-Bench feasibility-belief calibration.** The upper panel reports absolute initial-harness errors. The lower panel reports selected minus initial error on matched reports, with paired 10,000-resample bootstrap intervals. Negative differences mean improved calibration; the GPT-5.5 same-family intervals exclude zero, whereas all independent-tier intervals include zero. The independent-tier GPT-5.5 point estimates correspond to the matched 46-task Brier/ECE comparison in Table 5. Tiers and models are descriptive groups, not a controlled model comparison.

	GPT-5.5 initial		GPT-4o initial	
Task tier	Brier	ECE_{10}	Brier	ECE_{10}
Search distribution	0.138	0.146	0.346	0.356
Held-out, same family	0.167	0.169	0.319	0.331
Held-out, independent	0.186	0.192	0.270	0.276

Task tier	Model	ΔBrier [95% CI]	ΔECE_{10} [95% CI]
Same family	GPT-5.5	-0.048 $[-0.107, -0.003]$	-0.048 $[-0.103, -0.003]$
	GPT-4o	$+0.015$ $[-0.083, +0.112]$	$+0.026$ $[-0.066, +0.121]$
Independent	GPT-5.5	-0.037 $[-0.116, +0.040]$	-0.049 $[-0.120, +0.033]$
	GPT-4o	$+0.014$ $[-0.106, +0.133]$	$+0.010$ $[-0.104, +0.150]$

Retail τ^2-bench. All three harnesses provide valid probabilities on the same 32 blocked tasks and 32 feasible counterparts. Both selected harnesses improve AUROC, Brier, and ECE point estimates relative to the initial SA-only reference; their paired 95% intervals for these contrasts exclude zero (Table 16). The joint-minus-SA-only intervals all include zero. The selected systems nevertheless differ in held-out feasible solve rate ($19/32$ versus $9/32$; Table 14). The probe results do not establish why these solve rates differ; the probe and task trajectories are separate measurements.

Table 16: **Retail held-out feasibility beliefs.** Each configuration covers all 64 tasks. Intervals use 10,000 bootstrap resamples of the 32 blocked–feasible source pairs. Differences are first minus second; positive AUROC and negative Brier/ECE changes are better. The initial SA-only reports are a common held-out reference, whereas the joint search has its own initial configuration.

Harness / contrast	AUROC↑	Brier↓	ECE_{10} ↓
Held-out estimates [95% CI]			
SA-only initial	0.631 $[0.512, 0.751]$	0.318 $[0.243, 0.387]$	0.308 $[0.238, 0.386]$
SA-only selected	0.858 $[0.772, 0.936]$	0.227 $[0.146, 0.306]$	0.220 $[0.140, 0.305]$
Joint selected	0.825 $[0.732, 0.911]$	0.233 $[0.152, 0.315]$	0.242 $[0.160, 0.323]$
Paired differences [95% CI]			
SA-only − initial	$+0.227$ $[+0.100, +0.359]$	-0.091 $[-0.163, -0.019]$	-0.088 $[-0.170, -0.017]$
Joint − initial	$+0.193$ $[+0.100, +0.296]$	-0.085 $[-0.145, -0.030]$	-0.066 $[-0.135, -0.015]$
Joint − SA-only	-0.034 $[-0.130, +0.057]$	$+0.006$ $[-0.049, +0.056]$	$+0.022$ $[-0.038, +0.075]$

A.5.3 Terminal-Bench Challenge and Cross-Model Checks

Terminal-Bench uses operational abstention before the challenge and strict SA afterward. Table 17 therefore reports Δ_χ from Eq. 12, rather than formal CRA. For the selected GPT-5.5 joint harness, $86.9 \pm 5.1\%$ of pre-positive trajectories meet post-challenge strict SA; no pre-negative trajectory does, yielding $\Delta_\chi = -13.1 \pm 5.1$ points. The GPT-4o harness was selected at R_7 in a separate exploratory joint search. It has a $45.5 \pm 10.4\%$ negative-to-positive transition and a $19.0 \pm 5.5\%$ positive-to-negative transition, yielding $+26.5 \pm 15.7$ points. These rates compare separately selected model–harness systems, not matched model responses to the same harness.

A separate historical check applies a frozen GPT-5.5 SA-only harness to GPT-4o, changing its logged SA proxy from 37.5% to 81.3%. This proxy is not the strict-SA measure used for the GPT-5.5 results in Section 3.3. An independent GPT-4o joint search changes its logged two-sided score on search tasks from 33.6% to 53.3%. These results do not identify a model effect or a model–harness interaction, since the harnesses were not evaluated in a crossed design.

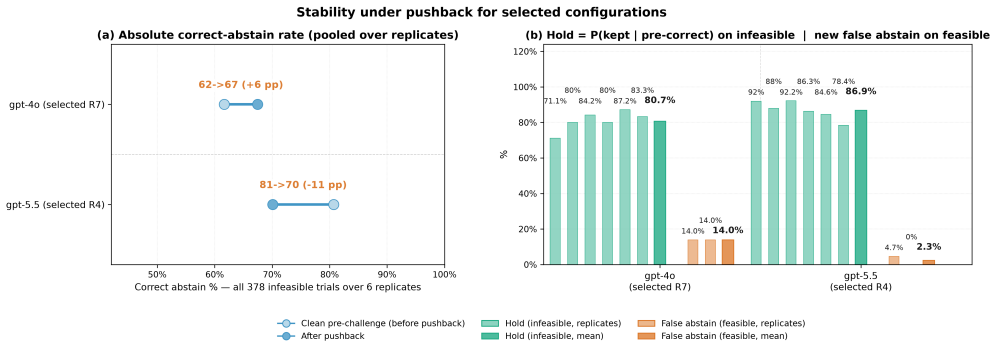

Figure 10: **Terminal-Bench response to one reconsideration request.** Panel (a) shows pre-challenge operational abstention and post-challenge strict SA on scorable blocked-task trajectories across six replicates; panel (b) shows conditional transitions on blocked tasks and new false abstentions on feasible search tasks. The denominators differ across panels, and their rates are not complements. Table 17 supplies the conditional transitions for the two selected model–harness systems.

Table 17: **Exploratory Terminal-Bench challenge transitions.** Blocked-task entries are run-level mean $\pm$ sample SD over six replicates, with 378 scorable blocked-task trajectories per model: 64 tasks in each of the first three runs and 62 in each of the last three after two persistent environment failures. Pre- and post-positive use different predicates; hence the signed contrast is Δ_χ, not CRA. The last row instead pools two runs over 43 original feasible search tasks. Neither the row groups nor the two model–harness systems form a controlled model comparison.

Terminal-Bench measure	GPT-5.5 R_4	GPT-4o R_7
Post-positive given pre-positive (%)	86.9 ± 5.1	81.0 ± 5.5
Pre-negative $\rightarrow$ post-positive (%)	0.0 ± 0.0	45.5 ± 10.4
Pre-positive $\rightarrow$ post-negative (%)	13.1 ± 5.1	19.0 ± 5.5
Signed Δ_χ (points)	-13.1 ± 5.1	$+26.5 \pm 15.7$
New false abstention, feasible (pooled)	$2/86$	$12/86$

A.5.4 Resource Use

Figure 11 reports output tokens and task-agent steps to first M0 recognition on Terminal-Bench's 64 blocked search tasks. Relative to a GPT-5.5+Codex reference with an abandon protocol (5,744 output tokens per task), the SA-only result at the final recorded round uses 1,879 tokens, and the selected joint R_4 harness uses 4,090. The final-round SA-only point need not equal the selected SA-only R_4 harness, so these values do not form a selected-to-selected cost comparison. Steps to recognition are similar in the plotted conditions; the output-token difference alone does not establish a mechanism or a cost–performance frontier.

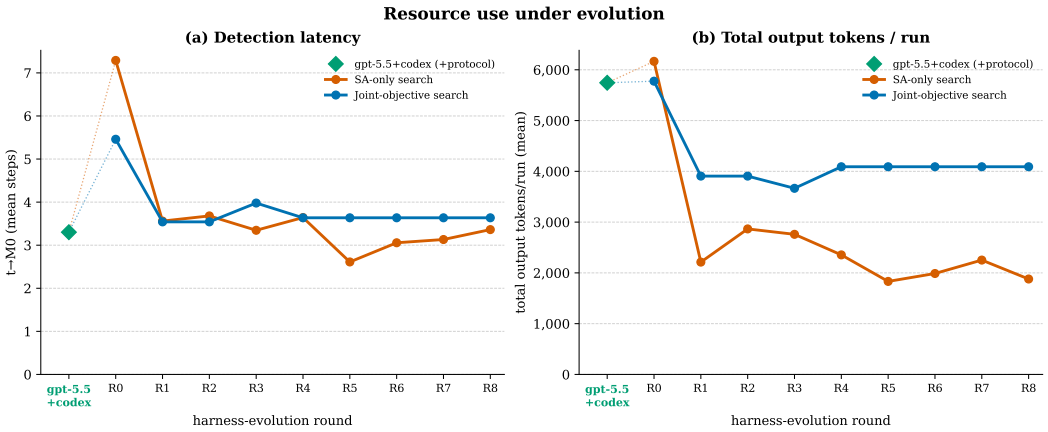

Figure 11: **Terminal-Bench resource-use diagnostics.** Per-round means on 64 blocked search tasks. The green diamond is the GPT-5.5+Codex protocol reference. Panel (a) records task-agent steps to first blocker recognition (M0); panel (b) records output tokens. The 1,879-token SA-only observation is from the final recorded round, while the 4,090-token joint observation corresponds to its selected R_4 incumbent. Later joint points re-evaluate a retained incumbent. Token use is descriptive and was not optimized directly.

A.6 Limitations

Our evaluation measures observable responses to task feasibility in two controlled agent settings. Pairing feasible tasks with augmented blocked tasks lets us assess warranted deferral and unnecessary refusal. However, these tasks do not estimate how often blockers appear in real-world settings or measure *subjective* awareness. The main experiments use GPT-5.5. Even though a separate GPT-4o joint search improves its logged two-sided score on search tasks, we did not test whether a selected harness transfers between models.

The largest harness gains are measured on the tasks used to select the harnesses. Although each round compares three candidate harnesses, each candidate is evaluated with one stochastic rollout per task, making the scores sensitive to rollout variation. The SA-only and joint searches also use different starting harnesses, task sets, and rollouts. We therefore cannot attribute differences in their final results to the selection objective alone.

We have less evidence about whether these gains carry over to *novel* blocked tasks. On retail τ^2's held-out blocked tasks, SA-only selection leaves strict SA unchanged at $8/32$. On Terminal-Bench, we do not have a full strict-SA evaluation of the selected harnesses on held-out blocked tasks. Thus, these results do not establish that the search-set gains transfer to new blockers.

In retail τ^2, a GPT-5.5 judge checks whether the agent recognized the blocker and gave the right reason. Because the judge and task agent use the same model, these semantic scores may share model-specific errors. Actions, authorship, and task completion are checked deterministically, so this concern is limited to the semantic judgments. Feasibility probabilities come from a separate read-only probe, so they need not reflect what the agent believed while performing the task. The stability test uses one neutral request to reconsider; it does not show how the agent would respond to repeated or adversarial challenges.