ForesightBench: Causal Evaluation of Prospective Information Acquisition under Irreversible Observability Loss

Anonymous authors
Paper under double-blind review

Abstract

Long-horizon agents take actions that determine both task progress and what remains observable. We study prospective information acquisition under irreversible observability loss: acquiring currently queryable environment state for later use before a task-required transition makes it unrecoverable. We formalize acquisition as constructing a durable sufficient statistic before the acquisition window closes. ForesightBench contains 90 executable, outcome-verified tasks across eight operational domains. Fifty Standard tasks provide lower- and mid-range difficulty anchors, while 40 Frontier tasks introduce longer dependencies, derived state, or multiple information windows. Each Frontier task has a matched Persistent Control designed to preserve the same query interface after the transition while holding other task conditions fixed. Under a common agent scaffold, six agent systems achieve 21.1–41.1% end-to-end success, with 30.0–62.0% on Standard tasks and 2.5–20.0% on Frontier tasks. Across 200 completed matched pairs from five systems, preserving access increases success by 47.5–97.5 percentage points; every system-level contrast is significant after Holm correction ($p_{\text{adj}} \leq 1.4 \times 10^{-4}$). These results show that the evaluated systems solve many more tasks when future-critical state remains queryable. Identifying which failures arise from missed acquisition windows additionally requires trajectory evidence.

1 Introduction

An agent can execute every local step correctly and still fail by acting too soon. The pattern occurs across the benchmark taxonomy, not only in infrastructure operations. A live routing table needed after a gateway reload maps to Network and edge; ephemeral source mappings needed after a database cutover map to Data and storage; expiring grants needed across a credential rotation map to Security and identity. Business and science workflows exhibit the same dependency when, for example, inflight settlements must survive a ledger migration, assay calibration must survive instrument retirement, or reservation state must support later order compensation. In each case, information is available early, becomes useful later, and disappears after a valid operation. Trajectory logging preserves observations, but cannot reconstruct state the agent never queried.

We call the required capability prospective information acquisition under irreversible observability loss. Before a task-required transition, the agent must identify state needed by a later stage and acquire a sufficient representation while access remains open. This form of information planning is related to active perception and epistemic action (Bajcsy, 1988; Kirsh & Maglio, 1994), with an additional long-horizon constraint: an action changes both the world and what can later be known about it. The decisive observation precedes its use and any visible sign that it was missed.

Modern benchmarks evaluate interactive agents across general environments, repository-level software engineering, user-facing tool workflows, terminal tasks, websites, and computer control (Liu et al., 2024; Jimenez et al., 2024; Yao et al., 2025; Merrill et al., 2026; Yang

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.

et al., 2023; Zhou et al., 2024; Xie et al., 2024). Their final scores, however, combine many possible failure causes. Information-seeking work studies queries or clarification for a current planning problem (Liu et al., 2022; Zhang et al., 2024); long-term memory benchmarks test whether information already in the interaction history remains usable (Wu et al., 2025; He et al., 2026). ATRBench most closely resembles our setting: a long-lived agent must elicit an unspoken preference while the user is online for use in a later session (Wu et al., 2026). ForesightBench instead exposes queryable environment state, and the agent closes its own observation window by performing a necessary operation within the workflow.

Task success alone cannot locate this failure: a failed workflow may reflect weak execution, a broken environment, incorrect use of acquired information, or a missed observation window. ForesightBench therefore pairs each Frontier Challenge, which removes access to future-critical state, with a Persistent Control that preserves access after the same transition. The pair shares the instruction, initialized workspace, seed state, and final-state verifier. Its outcome contrast measures sensitivity to access persistence; attributing a particular failure to missed acquisition additionally requires trajectory evidence.

The benchmark contains 90 executable Challenges across eight domains: 50 Standard tasks provide lower- and mid-range anchors, while 40 Frontier tasks add longer dependencies, multiple information windows, derived state, and conditional downstream use. Instructions state the operational objective without naming state to inspect, save, or checkpoint.

Across six systems evaluated with a common scaffold, Frontier success falls 16.0–47.0 percentage points below Standard success. Across 200 completed matched pairs from five systems, preserving post-transition access increases success by 47.5–97.5 percentage points. Challenge outcomes provide the primary system comparison; the paired intervention isolates sensitivity to access persistence.

Contributions.

- We formalize prospective information acquisition as control under a regime-specific observation kernel, with measurable query access, first sufficient-acquisition time, and complete timely sufficiency.
- We introduce ForesightBench, a suite of 90 outcome-verified, long-horizon agent tasks spanning eight domains, with 40 matched Persistent Controls that preserve post-transition observability.
- We report final-state success and matched observability effects, together with exploratory trajectory diagnostics and repeated-attempt results, without prescribing a solution path.

2 Related Work

Long-horizon interactive agent benchmarks. Language agents interleave reasoning with environment actions (Yao et al., 2023). Existing benchmarks span general interactive settings (Liu et al., 2024), embodied tasks (Shridhar et al., 2021), software engineering and terminals (Yang et al., 2023; Jimenez et al., 2024; Merrill et al., 2026), stateful user–tool interaction (Yao et al., 2025), and realistic web and computer environments (Zhou et al., 2024; Xie et al., 2024). These settings expose exploration, state tracking, and recovery, but their aggregate scores rarely isolate failure to observe state before the agent removes access. AgentBoard adds progress diagnostics (Ma et al., 2024); ForesightBench instead intervenes on a specific temporal dependency in observability.

Information acquisition and proactive agents. The value of information (Howard, 1966), active perception (Bajcsy, 1988), epistemic action (Kirsh & Maglio, 1994), and planning under partial observability (Kaelbling et al., 1998) provide classical accounts of why an agent may act to improve what it knows. Recent language-agent work makes such acquisition explicit. AFK trains agents to query an external knowledge source (Liu et al., 2022), while Ask-before-Plan elicits missing information for a current planning problem (Zhang et al., 2024). ProAct anticipates future requests and prepares evidence during idle time (Hu

et al., 2026); π-Bench and Pare-Bench evaluate proactive assistance in long-horizon, stateful settings with hidden intentions or active users (Zhang et al., 2026; Nathani et al., 2026). ATRBench is the closest work: it measures whether an agent asks an online user for an unspoken preference that will matter in a later offline session (Wu et al., 2026). ForesightBench changes both the information source and the closing event. The required state is already queryable in the environment, and the window closes when the agent performs a necessary operation in the same workflow. The decision is whether to inspect transient state before altering the system that contains it.

Memory over long interactions. LongMemEval evaluates retrieval, temporal reasoning, knowledge updates, and abstention over long chat histories (Wu et al., 2025). MemoryArena couples memory with action across interdependent multi-session tasks (He et al., 2026), and AgingBench diagnoses failures in memory writing, retrieval, and use as an agent's stored experience changes over time (Zhu et al., 2026). These benchmarks evaluate whether acquired evidence remains available and useful. ForesightBench focuses on acquiring environment state before access disappears.

Capability-specific diagnostic evaluation. Several recent benchmarks use controlled variants to make end-to-end failures more attributable. ATRBench separates asking, acquisition, and application with prompted, forced-ask, and oracle conditions (Wu et al., 2026); AgingBench uses temporal dependency graphs and counterfactual probes (Zhu et al., 2026); and AgentAbstain pairs executable tasks to test whether agents refrain from acting when required information is missing (Liu et al., 2026). ForesightBench instead isolates post-transition observability with an executable matched intervention, separating the system-level outcome contrast from trajectory-level attribution.

3 ForesightBench

Figure 1 summarizes the benchmark's causal task structure, matched intervention, and outcome-based evaluation. The tasks require agents to preserve sufficient information before a required transition closes the relevant query interface.

3.1 Formal Problem Setting

For item i and observability regime $d \in \{0, 1\}$, we model interaction as a finite-horizon controlled partially observed process

$$\mathcal{E}_i^{(d)} = (\mathcal{S}_i, \mathcal{A}_i, P_i, \Omega_i^{(d)}, \mu_i, u_i, V_i, H_i), \tag{1}$$

where $d = 0$ is the Challenge and $d = 1$ its Persistent Control. In the idealized matched design, the two regimes share the state space, action space, transition kernel, initial-state distribution, instruction, verifier, and horizon; $\Omega_i^{(d)}$ represents the intended post-transition access intervention. The completed structural audit verifies the allowlisted file-level differences. A source-diff semantic audit of all 40 pairs further confirms that the runtime changes preserve only the card-defined observation windows after closure; other differences are limited to control identifiers, documentation, a persistence flag, or non-agent-facing Oracle ordering. With interaction history $h_t = (u_i, o_0, a_0, \ldots, a_{t-1}, o_t)$, a system π induces

$$a_t \sim \pi(\cdot \mid h_t), \qquad s_{t+1} \sim P_i(\cdot \mid s_t, a_t), \qquad o_{t+1} \sim \Omega_i^{(d)}(\cdot \mid s_{t+1}, a_t), \tag{2}$$

and the potential terminal outcome is $Y_{ir}^{(d)}(\pi) = V_i(s_{ir,H_i}^{(d)}) \in \{0, 1\}$ for replicate r. The verifier scores only the terminal operational state; it imposes no command, trace, or verbal-rationale requirement.

Information windows. Each future-critical dependency is encoded by

$$\omega_{iw} = (X_{iw}, \mathcal{Q}_{iw}, b_{iw}, c_{iw}, R_{iw}), \tag{3}$$

where $X_{iw} = f_{iw}(s)$ is latent task state, $\mathcal{Q}_{iw}$ is the set of legitimate inspection procedures exposed by the task interface, $W_{iw} = [b_{iw}, c_{iw})$ is its acquisition window, and $R_{iw} = r_{iw}(X_{iw})$

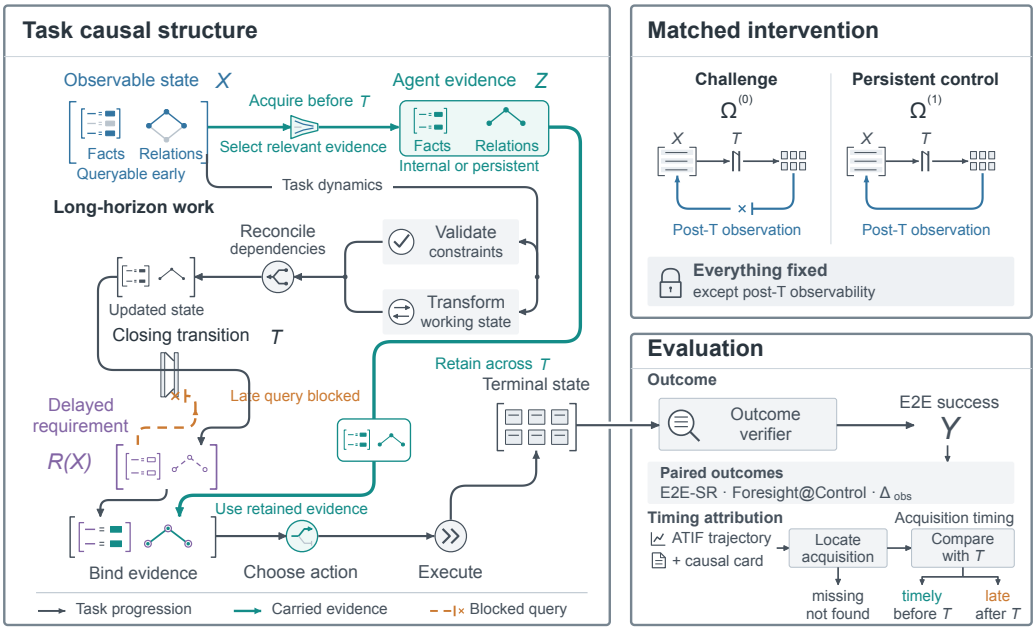


Figure 1: ForesightBench construction and measurement design. The task exposes future-critical state X through legitimate queries before a task-required closing transition T; success later depends on a durable sufficient statistic Z. The matched Persistent Control is designed to preserve post-transition observability while holding other task conditions fixed. Terminal reward is produced by an outcome-only verifier, while causal cards and ATIF trajectories support separate acquisition-timing analysis. Controls diagnose mechanism and do not enter the primary Challenge leaderboard.

is a downstream obligation. The reference workflow uses R_{iw} after closure, although an outcome-equivalent strategy may discharge the obligation earlier. The opening and closing indices are realized event times on a trajectory, with c_{iw} given by completion of the designated closing transition. To distinguish intrinsic access from information already stored by the agent, let $\mathfrak{H}^{-}_{iw,t}$ contain admissible histories with positive residual uncertainty about X_{iw}. We quantify idealized residual query access by

$$\mathcal{I}^{(d)}_{iw}(t) = \inf_{h \in \mathfrak{H}^{-}_{iw,t}} \sup_{a \in \mathcal{Q}_{iw}} \frac{I\left(X_{iw}; O^{(d)}_{t+1} \mid H_t = h, A_t = a\right)}{\mathsf{H}(X_{iw} \mid H_t = h)}. \tag{4}$$

Here an element of $\mathcal{Q}_{iw}$ is one benchmark-interface inspection that may return structured, multi-field state. The entropy is defined under the task-authoring distribution over hidden initial states; if no residual- uncertainty history remains, the access requirement is vacuously satisfied. Equation 4 is a construction idealization, not a statistic estimated from evaluation trajectories. Executable tests instead verify exact recoverability over the authored state cases. Construction targets $\mathcal{I}^{(0)}_{iw}(t) = 1$ for $t \in W_{iw}$ and $\mathcal{I}^{(0)}_{iw}(t) = 0$ for $t \geq c_{iw}$, while the Control targets $\mathcal{I}^{(1)}_{iw}(t) = 1$ after c_{iw}. Closure is caused by a task-required transition rather than an exogenous cutoff.

Operationalization. The causal card instantiates every formal object before evaluation. Its observation command and subcommand specify a canonical element of $\mathcal{Q}_{iw}$; its closure command and operation identify c_{iw}; and its required_fields, coverage_scope, and downstream_artifact specify what evidence is sufficient for the obligation R_{iw}. Construction tests exercise the authored hidden-state cases and require the canonical query to expose those fields before closure, to lose them after closure in the Challenge, and to retain them

in the Control. Thus Eq. 4 defines a benchmark-admission invariant; it is never substituted by a numerical mutual-information estimate in the reported results.

Let $Z_{irt}^{(d)} = \phi(h_{irt}^{(d)})$ denote the durable information or world-state commitment induced by the agent's trajectory, including observations, legitimate derived artifacts, lossless state-preserving operations, and obligations already discharged. The first time at which this state is sufficient for window w is

$$q_{irw}^{(d)} = \inf\{t \in \{b_{iw}, \dots, H_i\} : \mathsf{H}(R_{iw} \mid Z_{irt}^{(d)}) = 0\}, \qquad \inf \varnothing = +\infty, \tag{5}$$

and complete timely sufficiency is

$$A_{ir}^{(d)}(\pi) = \prod_{w \in \mathcal{W}_i} \mathbb{1}\left[q_{irw}^{(d)} < c_{iw}\right]. \tag{6}$$

The extractor separately records the first post-closure inspection attempt, $\ell_{irw}^{(d)} = \inf\{t \geq c_{iw} : a_t \in \mathcal{Q}_{iw}\}$. Thus a finite positive $q - c$ denotes sufficient acquisition after closure (possible in a Persistent Control), whereas a post-closure Challenge query is a late attempt ℓ and not evidence of successful acquisition. The criterion is defined by information sufficiency rather than a prescribed inspection command. Together with trajectory evidence, it supports distinguishing missed acquisition from incorrect use of acquired information and subsequent finalization failure. Empirically, the deterministic extractor realizes Eq. 5 by locating the first ATIF event whose direct observation or accepted equivalent strategy covers every card-required field and entity and remains usable for the downstream artifact. It then compares that event with the card-matched closure event and computes Eq. 6 across all windows. Ambiguous field coverage or event order is marked unresolved rather than imputed. Appendix Table 4 gives the complete field-to-measurement mapping.

3.2 Task Construction

Backward construction from a delayed dependency. Each task is designed backward from a final-world invariant. We specify a downstream obligation that depends on earlier state, expose a sufficient pre-transition information set through a normal interface, add a necessary transition that closes the interface, and delay the information's use with intervening work. The final instruction states the operational invariants without naming what to inspect, save, back up, or checkpoint. The agent must infer which state to acquire and when.

A task is admitted only if the transition is necessary, the delayed obligation depends on earlier state, the Oracle passes, and the no-operation (NOP) agent fails. For Frontier tasks, all file differences between the Challenge and Control must fall within the persistence allowlist (Appendix J).

A gateway task retains live tenant routing through a reload; a schema-registry task reconstructs a reference DAG after identifier reuse; a fulfillment task joins reservation state with a later, one-shot compensation capability. Each requires the agent to infer a future dependency and carry sufficient information across the transition that removes its source. We exclude difficulty caused by a hidden file, malformed interface, artificial shell trap, or prescribed command, and require enough trajectory evidence to audit the acquisition time in Eq. 5.

Appendix Figure 5 illustrates a two-window gateway release. The agent must acquire tenant policy before reload and selected-session affinity before activation, then use both in later routing and handoff decisions. The verifier evaluates the terminal service state and accepts any valid action sequence.

Difficulty and coverage. The scored suite contains 90 unique Challenge tasks across eight domains (Appendix Table 8). Standard-50 tasks contain a state-changing workflow and at least one information window, with a shorter dependency or more direct relation between state and obligation. Frontier-40 adds delayed subset selection, derived capabilities, transitive provenance, post-validation rollback, and multiple windows. The Frontier set contains 74 annotated windows: 34 tasks have two windows and six have one. It covers 30 dependency-pattern labels, and its maximum normalized pairwise instruction similarity is

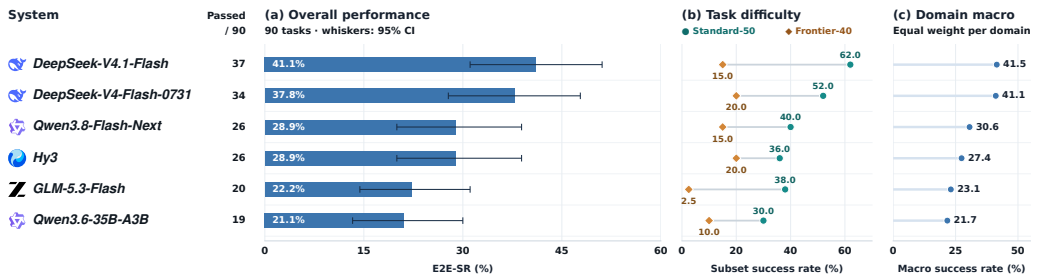


Figure 2: Challenge performance for the six complete agent systems. (a) Strict end-to-end success over all 90 Challenges; whiskers are task-bootstrap 95% confidence intervals, and the adjacent column gives pass counts. (b) Success on Standard-50 and Frontier-40. (c) Unweighted mean success across eight domains. Model-family marks identify vendors but do not encode performance. Exact values are reported in Appendix Table 1.

0.6673, below the frozen 0.80 portfolio ceiling. Split membership follows task structure and is independent of any model's pass rate.

3.3 Matched Persistent-Observability Intervention

Every Frontier task i defines two potential environments, $\mathcal{E}_i^{(d)}$, where d controls one access-persistence mechanism. The Challenge $\mathcal{E}_i^{(0)}$ makes X_{iw} unavailable after c_{iw}; the Persistent Control $\mathcal{E}_i^{(1)}$ keeps the same query interface readable. The pair shares the instruction, initialized task state, task-required workflow, and final-state verifier. The Control permits post-transition queries without supplying X_{iw}, identifying the relevant entity, or specifying when to inspect it. The agent must still discover, interpret, and use the required state.

For a fixed agent system, the item-level persistent-observability effect is the potential-outcome contrast

$$\tau_i(\pi) = \mathbb{E}_r\left[Y_{ir}^{(1)}(\pi) - Y_{ir}^{(0)}(\pi)\right]. \tag{7}$$

For the prespecified first replicate, write $Y_i^{(d)} := Y_{i1}^{(d)}$ and define the matched-pair contingency counts

$$n_{ab} = \sum_{i=1}^{N_F} \mathbb{1}\left[Y_i^{(0)} = a \ \wedge \ Y_i^{(1)} = b\right], \qquad (a,b) \in \{0,1\}^2, \tag{8}$$

where $N_F = 40$. Thus n_{01} counts failures rescued by persistent access, n_{10} counts negative realized contrasts requiring audit, and n_{11} counts tasks solved in both regimes. The paired estimators are

$$\widehat{\Delta}_{\text{obs}} = \frac{n_{01} - n_{10}}{N_F}, \qquad \widehat{F}_{\text{ctrl}} = \frac{n_{11}}{n_{01} + n_{11}}, \qquad \widehat{S}_{\text{obs}} = \frac{n_{01}}{n_{01} + n_{11}}. \tag{9}$$

$\widehat{F}_{\text{ctrl}}$ is Foresight@Control and $\widehat{S}_{\text{obs}} = 1 - \widehat{F}_{\text{ctrl}}$ is ObsSensitivity. The stronger prospective-acquisition label additionally requires process evidence of a missed window:

$$L_i = \mathbb{1}\left[Y_i^{(0)} = 0, \ Y_i^{(1)} = 1, \ A_{i1}^{(0)}(\pi) = 0\right]. \tag{10}$$

Controls are diagnostic and do not enter the primary Challenge score,

$$\widehat{J}_{\text{E2E}}(\pi) = \frac{1}{N}\sum_{i=1}^{N} Y_i^{(0)}(\pi), \qquad N = 90. \tag{11}$$

The reported outcome tables compute the contingency counts in Eq. 8, the three paired estimators in Eq. 9, and the primary score in Eq. 11 directly from binary verifier records. The process analysis computes timely sufficiency only for extractor-resolved trajectories

using the card-to-event mapping above. The remaining quantities define the environment, intervention, and benchmark-admission conditions; they are not presented as separately fitted empirical parameters.

The structural and source-diff audits verify all 40 pairs: instruction, verifier, and bootstrap hashes match; no file differs outside the persistence allowlist; and the reviewed runtime changes are restricted to preserving the card-defined observation windows after closure.

3.4 Outcome Verification and Validation

The verifier scores repair, required transition, exact reconciliation, preservation, and final health without constraining the procedure. Oracle passes all 90 frozen Challenges and NOP passes none, with no execution exceptions; materialized instances are checksum-matched to completed runs. The 40 causal cards cover all 74 Frontier windows and pass schema validation. The deterministic trajectory extractor is regression-tested and reports unresolved records explicitly; its timing analysis is descriptive and does not enter the primary outcome or matched-pair estimates (Appendices I and D).

4 Experimental Setup

4.1 Agent Systems and Evaluation Protocol

The unit of comparison is an agent system: a model, scaffold and version, Agent SDK revision, context policy, inference configuration, and execution budget. We report six systems with complete 90-task Challenge coverage under Claude Code 2.1.258, a common task-facing prompt and tool policy, a 220-turn limit, thinking enabled, and web tools disabled. Appendix K records provider- and model-specific context settings and execution limits. Figure 2 therefore compares agent systems under their recorded configurations.

Each reported system runs all 90 Challenges from the same frozen revision in fresh environments. The matched analysis uses every system row with all 40 Persistent Controls complete; partial pair rows are excluded rather than scored. The agent receives the task instruction and scaffold tools, but no causal cards, Oracle solutions, control labels, or verifier internals. A frozen registry records the model revision, endpoint class, scaffold and SDK versions, inference parameters, task checksum, and run ID. The leaderboard uses the first prespecified attempt for each task, with replacements allowed only for missing trials or recorded infrastructure exceptions. Each replacement is linked to its original record.

4.2 Outcome and Paired Metrics

The primary metric is strict end-to-end success on the 90 Challenges (E2E-SR; Section 3). We also report Standard-50, Frontier-40, and the unweighted mean of the eight domain success rates. The domain macro-average accounts for unequal domain counts; E2E-SR remains the primary score. Partial progress, trajectory quality, and Control performance are reported separately.

On the Frontier pairs, we report the contingency counts in Eq. 8, Control-SR, and the paired estimators in Eq. 9. Foresight@Control measures Challenge success among tasks passed under persistent access; ObsSensitivity is the complementary rescued-failure rate. We report both with the eligible denominator, the full matched-outcome matrix, and $\widehat{\Delta}_{\text{obs}}$. Low Control-SR limits the interpretation of either conditional estimate.

4.3 Process Measurement and Reliability

Frozen causal cards map ATIF trajectories to the process measures defined in Section 3: timely window coverage, complete timely sufficiency, information utilization, reactive-too-late rate, and normalized acquisition lag $(q - c)/\max(H, 1)$. Negative lag indicates acquisition before closure; positive lag indicates sufficient acquisition after closure and can occur only when access persists. Process rates use extractor-resolved trajectories only; unresolved

cases are reported as coverage gaps, not imputed. The deterministic extractor is regression-tested on field coverage, event order, and missing evidence, and never infers acquisition from terminal reward.

The main score uses one attempt per task. For GLM-5.3-Flash, we also collect three independent attempts per Frontier Challenge and Control. pass^k is the fraction passed in all first k attempts, whereas pass@k is the fraction passed at least once. Repeats are summarized within task and excluded from the primary score.

4.4 Uncertainty, Errors, and Reproducibility

Success rates use two-sided percentile task-bootstrap 95% intervals from 10,000 seeded resamples (Efron, 1979); they describe resampling of the fixed task set, not variation across independent agent runs. A family-clustered sensitivity analysis resamples task families with their constituent tasks; for the five complete paired rows, its endpoints equal those of the task bootstrap (Appendix Table 2). Challenge–Control differences use paired bootstrap intervals and exact McNemar tests (McNemar, 1947). Holm correction (Holm, 1979) covers the six prespecified system rows. Hy3 is evaluated only on Challenge and is non-estimable for the paired contrast, so its row receives the conservative value $p = 1$; the other five contrasts are estimable. Inference uses paired outcomes rather than overlap of marginal intervals. Repeats are summarized within task before resampling. Appendix Figure 6 reports per-item success, corrected item–total correlation, and leave-one-domain-out rank stability as descriptive diagnostics; none changes the primary score.

A verifier-returned zero is a clean failure, and a successful final-state verifier is authoritative. Agent, environment, or verifier exceptions without a valid reward count as failures; missing artifacts invalidate a leaderboard row. Tables and figures derive from versioned trial-, pair-, item-, window-, and verifier-check exports. The freeze contains 540 Challenge trials. For Qwen3.6-35B-A3B, a recovery run replaces 15 absent or infrastructure-exception Challenge trials and one absent Control trial; reward-zero outcomes are never replaced.

5 Results and Analysis

5.1 Challenge Performance

Oracle succeeds on all 90 frozen Challenges and NOP on none, without execution exceptions. Systems achieve 21.1–41.1% E2E-SR (Figure 2). The task-bootstrap intervals overlap, so point estimates alone do not establish reliable pairwise differences. Standard-to-Frontier success decreases by 16.0–47.0 percentage points across the six systems, which represent four model families. Appendix Figure 6 gives the item-level response structure.

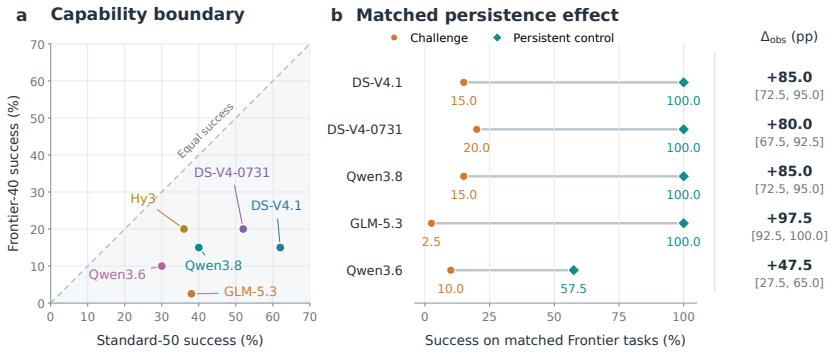


Figure 3: Challenge difficulty and matched-observability effects. (a) Standard versus Frontier success. (b) Forty-task matched results; labels give Control-minus-Challenge effects with task-bootstrap 95% CIs. Controls are diagnostic only.

5.2 Matched Persistent-Observability Diagnostic

The matched diagnostic contains 200 Challenge–Control pairs from five systems (Figure 3). Control success is 183/200 (91.5%) versus 25/200 (12.5%) on the matched Challenges. Pooled counts are $n_{11} = 23$, $n_{01} = 160$, $n_{10} = 2$, and $n_{00} = 15$; 23 of the 183 Control successes also pass the Challenge (Foresight@Control = 12.6%).

System-level effects range from 47.5 to 97.5 percentage points (Figure 3); every exact McNemar contrast remains significant after Holm correction ($p_{\text{adj}} \leq 1.4 \times 10^{-4}$). Control success ranges from 57.5% to 100%, so both-fail pairs are retained rather than assigned to lost observability. The matched results identify system-level sensitivity to access persistence; individual attribution still requires trajectory evidence.

5.3 Task-Level Difficulty and Effect Breadth

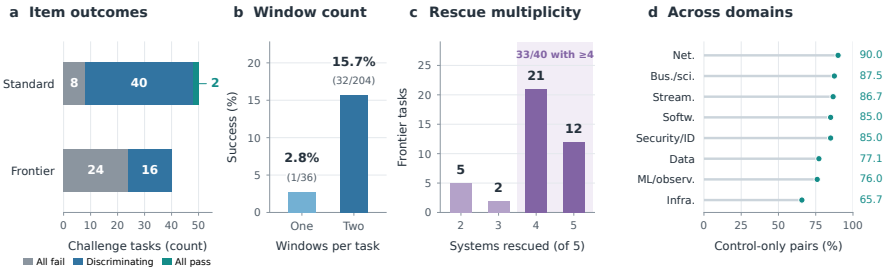


Figure 4: Task-level diagnostics. (a) Six-system item responses. (b) Success by window count. (c) Frontier tasks rescued in k of five systems. (d) Domain-level Control-only rates. Window and domain comparisons are descriptive.

All six systems fail 24/40 Frontier items and 8/50 Standard items; the other 16 Frontier items discriminate among systems (Figure 4a). One- and two-window Frontier tasks yield 1/36 and 32/204 successful system–task trials (Figure 4b). The one-window group is small and structurally different, so the contrast is descriptive.

Persistent access rescues 33/40 tasks for at least four systems and 12/40 for all five (Figure 4c). Domain-level Control-only rates range from 65.7% to 90.0%, with 2/200 reverse outcomes (Figure 4d). Domain comparisons are descriptive because counts are small and observations share tasks and systems.

6 Discussion and Conclusion

ForesightBench evaluates whether agents preserve future-critical information before task-required transitions remove access to it. The results expose two linked gaps: success falls sharply on the structurally harder Frontier tasks, and many failed Challenges become solvable when the same state remains observable. Matched outcomes identify sensitivity to access persistence rather than the causal process behind an individual failure; trajectory-based timing diagnostics provide the complementary process evidence. Source-level audits constrain the intervention to the card-defined observation windows. Together, these components support strict system comparison and an auditable analysis of when agents acquire information for later use.

AI Use Statement

Generative AI tools assisted code inspection and editing, experiment orchestration, analysis and visualization scripting, literature discovery, language editing, and early drafting. The authors selected the research question, designed the benchmark and interventions, verified sources, inspected generated code and tasks, and remain responsible for all claims, analyses,

and artifacts. Model output was used as empirical evidence only after verification against the executable benchmark or versioned analysis exports.

Ethics Statement

ForesightBench evaluates agents in isolated, synthetic environments and is not intended for production services. Its administration, migration, security, and recovery commands could be unsafe on real systems. Release materials specify sandbox-only use and exclude service credentials and internal endpoints. Benchmark scores do not certify safe infrastructure operation. The anonymous release inventory excludes credentials, private endpoints, and token-valued configuration fields; it publishes only sanitized configuration metadata.

Reproducibility Statement

The anonymized supplement contains the frozen 90-Challenge manifest, 40 Persistent Controls, checksums, environment definitions, causal cards, a sanitized six-system registry, long-form exports, and the scripts used for the reported tables and figures. Its integrity manifest indexes all 740 included terminal records and 730 available ATIF trajectories, records explicit trajectory missingness, preserves two truncated trajectory records as sanitized raw text, and verifies result/configuration hashes and retry lineage. Machine-readable audits record Oracle and no-operation results, pair integrity, the 40-pair source-diff review, and release-inventory checks. Credentials and endpoint addresses are excluded; proprietary services are represented by non-secret request and inference configuration.

References

Ruzena Bajcsy. Active perception. Proceedings of the IEEE, 76(8):966–1005, 1988. doi: 10.1109/5.5968.

DeepSeek-AI. DeepSeek-V4-Flash-0731 model card. Hugging Face model card, 2026a. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash-0731.

DeepSeek-AI. DeepSeek-V4.1-Flash model card. Hugging Face model card, 2026b. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash.

DeepSeek-AI et al. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026a. URL https://arxiv.org/abs/2606.19348.

DeepSeek-AI et al. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026b. URL https://arxiv.org/abs/2609.19969.

Bradley Efron. Bootstrap methods: Another look at the jackknife. The Annals of Statistics, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552.

GLM-5-Team. GLM-5.3-Flash model card. Hugging Face model card, 2026. URL https://huggingface.co/zai-org/GLM-5.3-Flash.

GLM-5-Team et al. GLM-5: from Vibe Coding to Agentic Engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments, 2026. URL https://doi.org/10.5281/zenodo.20953922.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. MemoryArena: Benchmarking agent memory in interdependent multi-session agentic tasks, 2026. URL https://arxiv.org/abs/2602.16313. ICML 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Ronald A. Howard. Information value theory. IEEE Transactions on Systems Science and Cybernetics, 2(1):22–26, 1966. doi: 10.1109/TSSC.1966.300074.

Haoyi Hu, Qirong Lyu, Xianghan Kong, Weiwen Liu, Jianghao Lin, Zixuan Guo, Yan Xu, Yasheng Wang, Weinan Zhang, and Yong Yu. Anticipate and learn: Unleashing idle-time compute in proactive agents, 2026. URL https://arxiv.org/abs/2605.25971.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In International Conference on Learning Representations, pp. 54107–54157, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

David Kirsh and Paul Maglio. On distinguishing epistemic from pragmatic action. Cognitive Science, 18(4):513–549, 1994. doi: 10.1207/s15516709cog1804_1.

Iou-Jen Liu, Xingdi Yuan, Marc-Alexandre Côté, Pierre-Yves Oudeyer, and Alexander Schwing. Asking for knowledge (AFK): Training RL agents to query external knowledge using language. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 14073–14093. PMLR, 2022. URL https://proceedings.mlr.press/v162/liu22t.html.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In International Conference on Learning Representations, pp. 52989–53046, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

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?, 2026. URL https://arxiv.org/abs/2607.10059.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In Advances in Neural Information Processing Systems, volume 37, pp. 74325–74362. Curran Associates, Inc., 2024. doi: 10.52202/079017-2365. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/877b40688e330a0e2a3fc24084208dfa-Abstract-Datasets_and_Benchmarks_Track.html.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

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. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

Deepak Nathani, Cheng Zhang, Chang Huan, Jiaming Shan, Yinfei Yang, Alkesh Patel, Zhe Gan, William Yang Wang, Michael Saxon, and Xin Eric Wang. Proactive agent research environment: Simulating active users to evaluate proactive assistants, 2026. URL https://arxiv.org/abs/2604.00842.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, Fan Zhou, Hao Luo, Haofeng Huang, Haoran Lian, Haoyan Huang, Hongqing Chen, Jianwei Zhang, Jing Xu, Junjie Wang, Langshi

Chen, Liangyu Wang, Linlang Jiang, Man Yuan, Minmin Sun, Peng Jin, Siqi Zhang, Siyu Wang, Xingzhang Ren, Yakai Wang, Yi Zhang, Yiming Dong, Yizhong Cao, Yubo Ma, Yunfei Mao, Bo Zheng, and Dayiheng Liu. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. Technical report, Alibaba Group, August 2026. URL https://arxiv.org/abs/2608.30320.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Official model release, April 2026a. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Qwen Team. Qwen3.6-35B-A3B model card. Hugging Face model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.6-35B-A3B.

Qwen Team. Qwen3.8-Flash-Next model card. Hugging Face model card, 2026c. URL https://huggingface.co/Qwen/Qwen3.8-Flash-Next.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Tencent Hy Team. Hy3 model card. Hugging Face model card, 2026. URL https://huggingface.co/tencent/Hy3.

Bin Wu, Guanyun Zou, Bingbing Wang, Huan Zhao, and Chuan Shi. Ask now, use later: Benchmarking the proactivity gap in long-lived LLM agents, 2026. URL https://arxiv.org/abs/2605.28108. Accepted to EMNLP 2026 Main Conference.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In The Thirteenth International Conference on Learning Representations, pp. 86809–86836, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In Advances in Neural Information Processing Systems 37, pp. 52040–52094, 2024. doi: 10.52202/079017-1650. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. InterCode: Standardizing and benchmarking interactive coding with execution feedback. In Advances in Neural Information Processing Systems 36, pp. 23826–23854, 2023. doi: 10.52202/075280-1035. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4b175d846fb008d540d233c188379ff9-Abstract.html.

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. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In International Conference on Learning Representations, pp. 9965–10017, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Haoran Zhang, Luxin Xu, Zhilin Wang, Runquan Gui, Shunkai Zhang, Haodi Lei, Zihao He, Bingsu He, Chicheng Qin, Tong Zhu, Xiaoye Qu, Yang Yang, Yu Cheng, and Yafu Li. π-Bench: Evaluating Proactive Personal Assistant Agents in Long-Horizon Workflows, 2026. URL https://arxiv.org/abs/2605.14678.

Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. Ask-before-Plan: Proactive language agents for real-world planning. In Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 10836–10863. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.636. URL https://aclanthology.org/2024.findings-emnlp.636/.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In The Twelfth International Conference on Learning Representations, pp. 15585–15606, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.

Jianing Zhu, Yeonju Ro, John Robertson, Kevin Wang, Junbo Li, Haris Vikalo, Aditya Akella, and Zhangyang Wang. Your agents are aging too: Agent lifespan engineering for deployed systems, 2026. URL https://arxiv.org/abs/2605.26302.

A Appendix Roadmap

The appendix begins with the complete numerical evidence (Appendix B), statistical and item-level diagnostics (Appendix C), trajectory auditing (Appendix D), and secondary system diagnostics (Appendix E). It then documents task construction and representative cases (Appendices F and G), the executable environment, verification, matched controls, and system configuration (Appendices H–K), followed by governance and release material (Appendices L–M).

Figure 5 illustrates these components in a two-window gateway task.

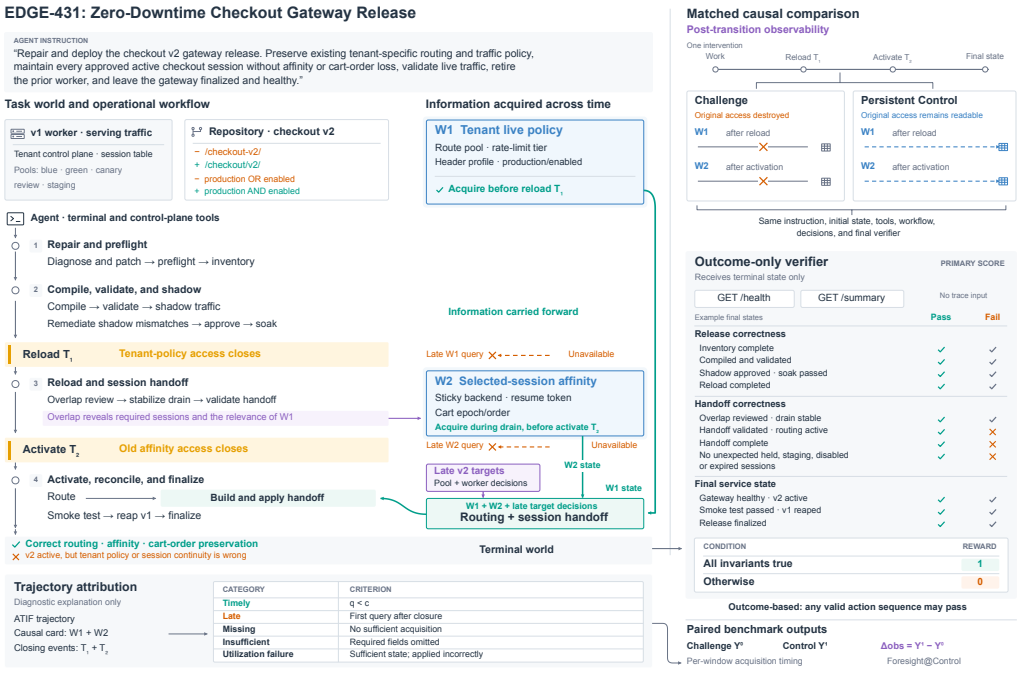


Figure 5: EDGE-431, a representative two-window Frontier task. Reload closes access to tenant policy (W_1), activation closes access to prior session affinity (W_2), and the delayed handoff requires both. Challenge and Persistent Control are designed to share the instruction, initialized world, tools, workflow constraints, and final-state verifier; the intended intervention is post-transition access to the same query interface. The primary reward depends exclusively on terminal invariants, whereas trajectory attribution classifies acquisition as timely, late, missing, insufficient, or misused.

B Complete Quantitative Results

This section reports exact Challenge scores and matched Challenge–Control outcomes. Appendix E reports resource use and partial verifier progress.

B.1 Exact Challenge scores

Table 1 provides the exact values underlying Figure 2. Std., Front., and D-Macro denote Standard-50, Frontier-40, and the unweighted mean over eight domains.

Table 1: Exact Challenge results under the common Claude Code scaffold. E2E-SR includes a task-bootstrap 95% confidence interval.

System	Pass	E2E-SR [95% CI]	Std.	Front.	D-M.
DeepSeek-V4.1	37	41.1 [31.1, 51.1]	62.0	15.0	41.5
DeepSeek-V4-0731	34	37.8 [27.8, 47.8]	52.0	20.0	41.1
Qwen3.8	26	28.9 [20.0, 38.9]	40.0	15.0	30.6
Hy3	26	28.9 [20.0, 38.9]	36.0	20.0	27.4
GLM-5.3	20	22.2 [14.4, 31.1]	38.0	2.5	23.1
Qwen3.6	19	21.1 [13.3, 30.0]	30.0	10.0	21.7

B.2 Matched persistent-observability results

Table 2 reports the matched outcome counts, conditional success rates, and observability effects.

Table 2: Complete matched Persistent Control diagnostic. The four n_{ab} columns use Challenge outcome a and Control outcome b. F@Ctrl is conditioned on Control success. p_{H} is the Holm-adjusted exact McNemar value under the six-row prespecified family; Hy3 has no complete matched contrast and contributes a raw value of one.

System	n_{11}	n_{01}	n_{10}	n_{00}	Ctrl.	F@Ctrl	Δ_{obs} [95% CI]	p_{H}
DeepSeek-V4.1-Flash	6	34	0	0	100.0	15.0	85.0 [72.5, 95.0]	5.8×10^{-10}
DeepSeek-V4-Flash-0731	8	32	0	0	100.0	20.0	80.0 [67.5, 92.5]	1.4×10^{-9}
Qwen3.8-Flash-Next	6	34	0	0	100.0	15.0	85.0 [72.5, 95.0]	5.8×10^{-10}
GLM-5.3-Flash	1	39	0	0	100.0	2.5	97.5 [92.5, 100.0]	2.2×10^{-11}
Qwen3.6-35B-A3B	2	21	2	15	57.5	8.7	47.5 [27.5, 65.0]	1.3×10^{-4}

C Metric and Statistical Specification

C.1 Outcome metrics and denominators

Table 3 defines the main metrics. The primary score uses all 90 Challenges; Controls and trajectory diagnostics are reported separately.

Additional metrics are complete timely sufficiency, reactive-too-late rate, normalized acquisition lag, error rates, pass^k, and pass@k. Missing acquisition has its own category. Unresolved causal records are excluded from process-rate denominators and counted in process-measurement coverage.

C.2 Uncertainty and paired tests

We use 10,000-replicate, two-sided percentile bootstrap intervals. The task-bootstrap 95% confidence intervals describe variability under resampling of the 90 Challenge tasks; they do not include stochastic-run variation because the primary comparison uses one prespecified attempt per task. The completed family sensitivity analysis resamples task families while retaining every task in each sampled family. Its endpoints equal the reported task-bootstrap endpoints for all five complete paired rows. Challenge–Control effect intervals resample the 40 matched task pairs. Exact McNemar tests compare discordant outcomes for each system.

Table 3: Core metrics. Every conditional metric is accompanied by its eligible denominator.

Metric	Definition	Missing/error policy
E2E-SR	Mean binary reward over all 90 Challenges	Attempted errors fail; absent tasks invalidate row
Standard / Frontier	Mean reward over 50 / 40 Challenge subsets	Same as E2E-SR
Domain-Macro	Unweighted mean of eight domain success rates	Missing domain invalidates metric
Control-SR	Mean reward over observed Persistent Controls	Non-estimable pairs excluded from the denominator and reported separately
Foresight@Control	Challenge success among pairs with Control success	Undefined with zero eligible pairs
ObsSensitivity	Challenge failure among pairs with Control success	Complement of Foresight@Control on eligible pairs
Timely coverage	Timely sufficient windows divided by required windows	Deterministically resolved records; unresolved records excluded and counted
Information utilization	Success conditional on complete timely sufficiency	Extractor-resolved and descriptive when denominator < 10

Holm correction uses the six prespecified system rows in the frozen matrix: Hy3 is evaluated only on Challenge and contributes a conservative raw value of one for the non-estimable paired contrast, while the five estimable contrasts are reported. System-level inference uses these paired outcomes rather than overlap between marginal intervals. The analysis export records the deterministic seeds and full (Y, C) matrix.

Repeated attempts are summarized within task before resampling. pass^k requires all first k attempts to pass; pass@k requires at least one. Extra attempts do not enter the one-attempt leaderboard. Seeds and attempt order are recorded in the analysis exports and run inventory.

C.3 Benchmark quality diagnostics

The item analysis identifies 2 all-pass, 32 all-fail, and 56 discriminative tasks. Corrected item–total correlation is estimable for the 56 non-degenerate items, with a mean of 0.197. The six-system ranking has Kendall's $\tau = 1.0$ against the full ranking in each of the eight leave-one-domain-out analyses. These diagnostics describe the current six-system response matrix and do not alter the benchmark score.

Figure 6 shows item-level outcomes by split and domain, including tasks failed by all systems and tasks with different outcomes across systems.

D Trajectory Measurement

D.1 Event extraction

The extractor maps ATIF tool calls and observations to causal-card events. For each window, it records the first acquisition attempt, first sufficient acquisition, closing transition, first late attempt, satisfaction strategy, normalized lag, and evidence pointers. Timeliness requires an executable representation sufficient for the downstream obligation to survive closure; model narration alone is not evidence.



Figure 6: Item-level Challenge responses for the six reported systems. Tasks are grouped by split and domain; each cell records one system–task outcome. The lower trace reports how many systems solve each item. Two tasks are solved by all six systems, 32 by none, and 56 distinguish at least one pair of systems.

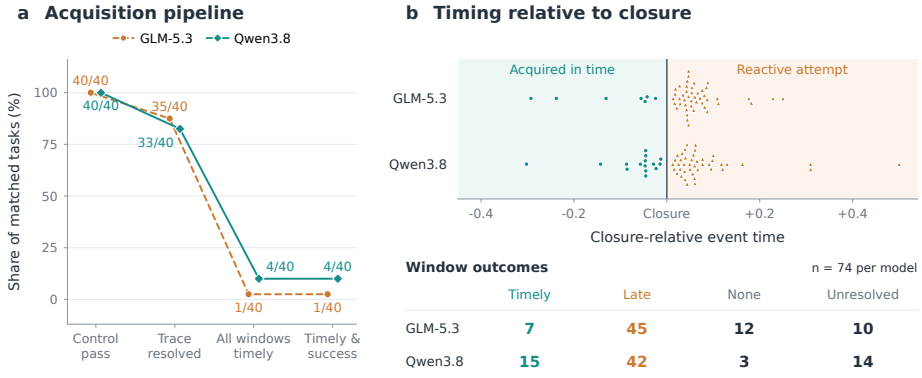


Figure 7: Exploratory acquisition timing for two systems with resolvable ATIF records. (a) Task-level pipeline over 40 matched Frontier pairs. (b) Acquisition events relative to closure; the lower strip summarizes all 74 windows per model. The timing measurements are descriptive and do not enter the primary outcome or matched-pair tests.

Table 4 makes explicit how the formal objects are implemented. In particular, the extractor does not estimate the mutual-information quantity in Eq. 4; that equation is enforced at benchmark construction by executable access tests. The trajectory analysis begins only after those construction gates have passed.

Table 4: Operational correspondence between the formal specification, causal-card fields, and trajectory measurements.

Formal object	Construction artifact	Evaluation operation
X_{iw}, R_{iw}	semantic_role, required_fields, coverage_scope, downstream_artifact	Require evidence for every field and covered entity needed by the downstream obligation
$\mathcal{Q}_{iw}$	observation.command and subcommand	Match a direct inspection or an encoded outcome-equivalent strategy
c_{iw}	closure.command and operation	Locate the closing event in ordered ATIF tool records
$\mathcal{I}_{iw}^{(d)}$	Executable pre-/post-closure access tests over authored states	Benchmark-admission gate; no trajectory-level numerical estimate
$q_{irw}^{(d)}, A_{ir}^{(d)}$	Card fields plus accepted-strategy rules	Record the first sufficient evidence event, compare it with closure, and aggregate across all required windows

Accepted strategies include direct acquisition, state-preserving operations, safe reordering, completion before closure, and other explicitly encoded equivalent paths. Records remain unresolved when output is truncated, field coverage is ambiguous, a wrapper obscures timing, or the deterministic rules cannot establish sufficiency.

D.2 Exploratory acquisition-timing analysis

The deterministic extractor resolves 35/40 GLM and 33/40 Qwen3.8 trajectories; complete timely sufficiency is observed in 1/35 and 4/33. For GLM, the 35 resolved trajectories contain one success and 34 failures; 30 of those failures include an identifiable first attempt after closure, while four contain no identifiable attempt. For Qwen3.8, the 33 resolved trajectories contain four successes and 29 failures; 28 of those failures include a first attempt after closure, while one contains no identifiable attempt. Two additional Qwen3.8 Challenge successes remain unresolved and therefore do not enter the 4/33 timely-sufficiency denominator. Utilization is 1/1 and 4/4, below the prespecified denominator of ten. Figure 7 summarizes these exploratory timing patterns; the small utilization denominators limit comparisons between acquisition and subsequent use. This analysis is limited to the GLM and Qwen3.8 records processed by the extractor. Across three GLM attempts, Challenge pass@3 is 7/40 but pass^3 is 0/40, whereas all 120 Control trials pass. The primary results retain the prespecified first attempt.

D.3 Failure attribution

Table 5: Primary trajectory-failure labels in priority order; final failure alone never determines the process label.

Label	Required evidence
Infrastructure error	Trial cannot validly test the agent because the execution substrate fails
Task/verifier error	Environment or verifier violates the frozen task contract
Prospective acquisition failure	Challenge fails, Control passes, and trajectory evidence shows that required information was not sufficiently preserved before closure
Information utilization failure	All windows are timely sufficient, but acquired information is interpreted or applied incorrectly
Post-acquisition finalization failure	Information is acquired and used correctly, but later execution or health checks fail
Ordinary execution failure	Trajectory evidence locates failure outside the information dependency; Control failure alone is insufficient for this label
Unresolved	Available evidence cannot support one of the preceding labels

E Secondary System Diagnostics

These analyses describe resource use and partial verifier progress alongside the primary success score. The accompanying exports contain:

1. complete per-system Challenge, Standard, Frontier, domain-macro, and family-macro results with confidence intervals and error counts;
2. all 40 Challenge–Control outcome matrices for every complete paired system row, eligible denominators, paired effects, and adjusted tests;
3. per-domain, per-family, per-pattern, one-window, and two-window breakdowns;
4. timely acquisition, utilization, reactive-too-late, acquisition-lag, and measurement-coverage analyses;
5. repeated-attempt pass^k/pass@k results for the completed GLM Frontier runs; and
6. provider-recorded cost, wall-clock, step, tool-call, token, and partial-verifier-progress diagnostics with explicit coverage.

Efficiency analyses are secondary. Provider-reported cumulative context tokens measure repeated turn-level billing input rather than unique text. Cost fields use the configured provider accounting and are missing for some interrupted or exception trials. We report coverage and compare matched Challenge–Control resource ratios within each system.

E.1 Execution footprint and accounting coverage

Figure 8 compares Challenge and Control resource use within each system–task pair. Median Challenge/Control ratios are 2.36–4.62 for recorded cost, 2.28–4.68 for cumulative reported tokens, 1.44–4.81 for agent time, and 1.60–2.18 for tool calls. Four systems expend more on at least 88% of cost-complete pairs and at least 85% of token-complete pairs. Qwen3.6 has broader dispersion and exceeds one on 60% of both measures. Persistent observability therefore changes both success and execution burden within the matched workflows. These resource ratios remain descriptive secondary outcomes; they do not enter the benchmark score or establish a mediation pathway from observability to reward. The companion step-count diagnostic gives median ratios of 1.88–2.39, with 61.5%–100% of complete pairs above one; the released analysis table records the full step-count quantiles.

Table 6 reports the denominator behind every resource diagnostic. Token coverage requires input, cached, and output-token fields; all observed cache-token values are zero, so the export provides no evidence for cross-system cache savings. Cost coverage ranges from 29/90 to 90/90 Challenge trials. It is complete for successful trials in every system except Hy3, where cost is recorded for 23/26 successes. Missing cost is concentrated in interrupted trials, limiting estimates of total benchmark cost. The released diagnostic table retains raw coverage and success-conditioned medians for audit.

Table 6: Coverage of secondary Challenge diagnostics. Entries are observed/eligible trials. Cost-pass conditions cost coverage on strict success; failure checks condition on failed trials. Steps/tools share the same coverage in every row.

System	Cost	Cost-pass	Tokens	Time	Steps/tools	Failure checks
DeepSeek-V4.1-Flash	66/90	37/37	90/90	90/90	90/90	53/53
DeepSeek-V4-Flash-0731	75/90	34/34	87/90	89/90	86/90	53/56
Qwen3.8-Flash-Next	89/90	26/26	90/90	90/90	90/90	64/64
Hy3	29/90	23/26	90/90	90/90	90/90	64/64
GLM-5.3-Flash	88/90	20/20	90/90	90/90	90/90	70/70
Qwen3.6-35B-A3B	90/90	19/19	90/90	90/90	84/90	71/71

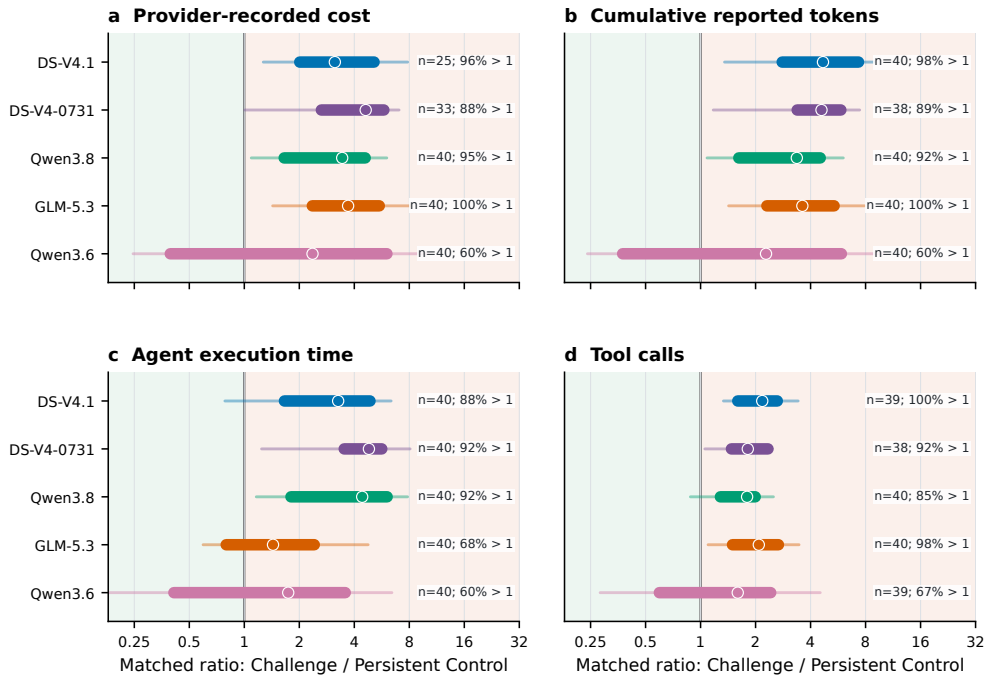


Figure 8: Matched execution burden on the five systems with complete Persistent Controls. Each dot is the median Challenge/Control ratio, the thick segment is the interquartile range, and the thin segment spans the 10th–90th percentiles. Ratios above one mean greater Challenge burden; annotations give complete-pair coverage and the share above one. Cost is provider-recorded and restricted to pairs with both values. Tokens are cumulative across turns rather than unique context.

E.2 Partial verifier progress

Across all six rows, 10/378 failed trials have recorded check completion at or above the pre-specified 90% near-miss threshold. Qwen3.6 has 57/71 (80.3%) measured failures satisfying fewer than half of the terminal checks, whereas the same low-completion share ranges from 4.7% to 25.0% for the other systems. Check-level coverage is reported in Table 6. These descriptive diagnostics distinguish near-complete workflows from failures that leave most terminal obligations unsatisfied; they do not alter strict success.

F Benchmark Construction and Dataset Card

F.1 Backward authoring protocol

Task authoring starts from a terminal condition that depends on earlier environment state. The author records the required entities and fields, their legitimate observation interface, the window-opening event, the closing transition, why later reconstruction fails, and the artifact or decision that consumes the state. Once this dependency is executable, the user instruction states the operational goal and preservation invariants without disclosing the acquisition strategy.

Each Frontier task has a machine-readable causal card containing its family, domain, dependency pattern, matched pair, and one or two information windows. Window records specify the semantic role, required fields, observation command, opening condition, closing operation, entity scope, and reference artifact. Cards remain outside the agent context and are bound by source hashes to the runtime and reference solution.

F.2 Admission and rejection criteria

Table 7: Task admission criteria. Every admitted Challenge must satisfy all required conditions.

Criterion	Operational test
Natural transition	Closing the window is a normal and necessary part of completing the stated objective.
Delayed necessity	At least one later verifier condition depends on state available before the transition.
Genuine information loss	The Challenge offers no legitimate post-transition query that reconstructs the required value.
Latent strategy	The instruction does not ask the agent to inspect, save, back up, checkpoint, or invoke a named acquisition command.
Outcome verification	Alternative valid strategies can pass; the verifier does not compare the trajectory with the Oracle.
Executability	A reference Oracle passes from a clean environment and produces the required state transition.
Non-vacuity	A no-operation agent fails cleanly, and every essential verifier check is exercised.
Auditable process	The observation and closure events leave enough ATIF evidence to measure timing without revealing hidden verifier state.

Table 7 defines the task-selection policy for measuring prospective acquisition within a legitimate agent workflow.

We reject candidates driven by broken tools, undisclosed paths, artificial command budgets, brittle string matching, irrelevant search, or reproduction of an author script. We also exclude tasks when persistent files reveal the target state after transition, the transition can be skipped, or the instruction discloses the desired ordering.

F.3 Composition and structural difficulty

The scored release contains 50 Standard and 40 Frontier Challenges. Matched Controls for all Frontier tasks yield 130 executable instances. The domain distribution appears in Table 8. Frontier covers 30 dependency patterns and 74 annotated windows: 34 tasks contain two windows and six contain one. Task structure determines split membership.

Standard tasks use a relatively direct dependency within a state-changing workflow. Frontier tasks add delayed subset selection, capability handoff, transitive provenance, identifier reuse, post-validation rollback, or a second window opened by the first transition. Domains, artifacts, and action vocabularies vary; the common structure is the temporal dependency.

Table 8: ForesightBench Challenge composition. Persistent Controls are paired with the 40 Frontier tasks and are not included in these counts.

Domain	Standard	Frontier	Total
Business and science	7	8	15
Data and storage	9	7	16
Infrastructure and orchestration	7	7	14
ML and observability	3	5	8
Network and edge	5	2	7
Security and identity	8	4	12
Software platform	7	4	11
Streaming and workflow	4	3	7
Total	50	40	90

The domain labels identify the application object of the delayed dependency, not merely the software substrate used to instantiate it. For example, live routing and gateway state

map to Network and edge; database cutovers and schema lineage to Data and storage; credentials and grants to Security and identity; and ledger settlement, claims adjudication, tax close, laboratory calibration, and order fulfillment to Business and science. Infrastructure and orchestration, ML and observability, Software platform, and Streaming and workflow analogously group tasks by their downstream operational contract.

The frozen portfolio audit reports maximum normalized pairwise instruction similarity of 0.6673, below the prespecified 0.80 ceiling. Semantic diversity is tracked separately through domain, family, dependency-pattern, information-object, and transition labels. The task catalog also records window count, estimated expert time, image digest, and checksum.

G Representative Tasks and Qualitative Cases

G.1 Task structures

Table 9 summarizes three information-window mechanisms without prescribing solution paths.

Table 9: Representative information-window mechanisms. x is the early state, T the closing transition, and R the delayed obligation.

Scenario	Early state x	Transition T	Later obligation R
Gateway repair	live tenant routing and policy	configuration reload	preserve routing while deploying a path fix
Schema migration	reference graph over current identifiers	source namespace deletion	rebuild compatible references under new identifiers
Order recovery	reservation ownership and compensation capability	reservation release	complete only dynamically affected sagas

The gateway case separates live state from persistent configuration; the schema case requires a relational structure; and the order case joins state from separate windows. The required information differs across these tasks; preserving a file is useful only if it retains the state needed downstream.

H Executable Environment and Agent Interface

H.1 Task package

Each Harbor task package (Harbor Framework Team, 2026) contains a specification, natural-language instruction, environment image, executable verifier under tests/, and reference solution used for validation. Frontier packages also contain the design metadata used to generate causal cards. The agent-facing workspace excludes the solution, card, control label, verifier source, and hidden reference state.

The specification records the container image, resource limits, healthcheck, and isolated agent and verifier execution settings. Images are built and prewarmed before evaluation and published by immutable digest. After the healthcheck succeeds, the agent uses the scaffold's shell and file tools. The verifier later inspects the final state through a separate evaluation entry point.

H.2 Lifecycle and isolation

Every attempt uses the same task bootstrap in a new sandbox. Trials share no mutable volume, shell history, process state, or agent-created artifact. Hidden values are instantiated before execution and fixed within each Challenge–Control pair. All compared systems share one network policy, and the future-critical state requires no network access. Cleanup follows collection of reward, verifier checks, trajectory, and infrastructure status.

H.3 Agent-visible and evaluator-only information

Table 10: Information boundary used in every evaluation.

Artifact	Agent-visible	Evaluator-only
User objective	Natural task instruction	Authoring notes and causal rationale
Environment	Workspace, tools, service responses, public documentation	Bootstrap truth and hidden reference state
Evaluation	Normal command feedback	Verifier source, expected final state, check-level labels
Diagnostics	Agent's own interaction history	Causal card, pair identity, parser labels, Oracle trajectory

The task service implements each information window. Its documented interface returns the live state before closure. After the required transition, the Challenge returns a domain-appropriate absence signal and removes legitimate reconstruction paths; the Control retains the query response. The user goal and action space remain fixed.

The agent can inspect the required state before the transition and must infer its relevance and acquisition timing. Table 10 separates agent-visible information from evaluator-only material that would reveal the acquisition strategy or reference solution.

I Outcome Verification and Quality Control

I.1 Verifier anatomy

The verifier requires repair of the initial defect, execution of the required transition, exact downstream reconciliation, preservation of designated entities, and a healthy final service or workflow. Check-level outcomes are grouped by defect, transition, preservation, reconciliation, and health, while the official reward remains binary.

The verifier neither compares trajectories with the Oracle nor requires a particular acquisition command or filename. Backups, sufficient statistics, native lossless migrations, safe reordering, and completion before closure are all valid when they satisfy the final contract. The timing extractor separately requires evidence that an accepted strategy preserved the state needed by the downstream obligation before closure.

I.2 Completed validation gates

Table 11: Completed release-validation checks.

Gate	Evidence	Status
Oracle solvability	Oracle passes all 90 frozen Challenges without execution exceptions	Passed
Verifier non-vacuity	NOP fails all 90 frozen Challenges without execution exceptions	Passed
Materialization	Task checksums match the completed Harbor evaluations	Passed
Pair structure	40/40 pairs preserve instruction, verifier, and bootstrap hashes	Passed
Runtime intervention	Every pair contains the expected persistence change and no unexpected changed file	Passed
Causal-card schema	All 40 Frontier cards and 74 windows satisfy the frozen schema	Passed
Pair semantics	Source-diff review of all allowlisted changes confirms that only card-defined post-closure access persists	Passed
Release inventory	Datasets, exports, sanitized registry, trial records, trajectories, scripts, hashes, and retry lineage are indexed	Passed

Oracle success demonstrates an executable solution; NOP failure confirms that leaving the environment unchanged does not pass. We retain check-level outputs, repeat both tests

after runtime or verifier changes, and bind results to task checksums. A changed task or verifier invalidates prior results unless semantic equivalence is established.

I.3 Verifier boundary checks

Oracle and NOP runs exercise both ends of the scoring boundary on the exact frozen tasks: Oracle passes 90/90, whereas leaving the environment unchanged passes 0/90, and neither run has an execution exception. Check-level verifier records separate repair, transition, preservation, reconciliation, and health. The task checksum audit binds these results to the materialized benchmark used for evaluation.

J Matched Persistent-Observability Controls

J.1 Intervention contract

For Frontier item i, Challenge and Control share the instruction, hidden values, workflow, commands, defects, and verifier. Their intended semantic difference is post-T access to the same observation interface. The Control allows a late query without revealing the relevant entity or its value.

The pair audit hashes the critical surface and allowlists packaging files that encode the intervention, including runtime and task metadata. In the frozen structural audit, all 40 pairs match on instruction, bootstrap, and verifier hashes, contain the persistence change, and have no unexpected changed file. A source-diff review of every allowlisted change further verifies that the runtime modifications only keep the causal-card observation windows readable after their closing transitions. Changes outside the runtime are limited to control identifiers and documentation, a persistence environment flag, or hidden Control Oracle ordering.

J.2 Why the Control supports attribution

The Control preserves access after closure while retaining the task requirements for discovery, interpretation, execution, and finalization. Control success shows that the system completed the workflow in that trial. A paired Challenge failure is consistent with sensitivity to access persistence, but assigning a prospective-acquisition failure label additionally requires evidence that no sufficient representation was obtained before closure.

A pair that passes in both conditions records success under both access regimes. A pair that fails in both provides no observed rescue and is excluded from Foresight@Control; its failure mechanism still requires trajectory evidence. Challenge success with Control failure can arise under stochastic execution and warrants review for drift, intervention effects, or run-to-run variation.

J.3 Threats to the matched design

Runtime changes can alter latency, error text, or service timing despite file equality. The source-diff audit therefore checks pre-transition interfaces, the required transition, action space, downstream work, verifier identity, and the scope of the post-transition accessibility change. No reviewed pair contains a material change outside post-closure persistence. The matched estimate should nevertheless be interpreted for these executable task implementations rather than as a substrate-independent causal law.

K Complete Experimental Configuration

K.1 Unit of comparison and system registry

All rows use Claude Code 2.1.258, 220 maximum turns, thinking enabled, disabled web tools, and the same task-facing protocol. Compact "stock" denotes no explicit compaction override; 131k@80 denotes a 131,072-token automatic compaction window triggered at 80%.

Each agent-system row records model and service revision, provider class, scaffold and SDK version, inference and retry settings, benchmark checksum, and run IDs. Any field change creates a new row; credentials and private addresses are excluded.

Table 12: Frozen registry for the six reported Challenge systems. Conc. denotes the maximum number of concurrent trials.

System	Reasoning	Compact	Conc.	Challenge
DeepSeek-V4.1-Flash	thinking	stock	16	90
DeepSeek-V4-Flash-0731	thinking	stock	16	90
Qwen3.8-Flash-Next	thinking+xhigh	stock	16	90
Hy3	thinking	131k@80	32	90
GLM-5.3-Flash	thinking	stock	16	90
Qwen3.6-35B-A3B	thinking	131k@80	16	90

Technical background is documented by the DeepSeek-V4.1-Flash and DeepSeek-V4 reports (DeepSeek-AI et al., 2026b;a), the Qwen3.8-Next report (Qiu et al., 2026), and the GLM-5 report (GLM-5-Team et al., 2026). Exact evaluated releases are identified by their official model cards: DeepSeek-V4.1-Flash and DeepSeek-V4-Flash-0731 (DeepSeek-AI, 2026b;a), Qwen3.8-Flash-Next (Qwen Team, 2026c), Hy3 (Tencent Hy Team, 2026), GLM-5.3-Flash (GLM-5-Team, 2026), and Qwen3.6-35B-A3B (Qwen Team, 2026b); the latter also has an official release announcement (Qwen Team, 2026a). Table 12 summarizes evaluation settings; the registry records served identifiers and service dates.

The comparison fixes the scaffold and tool policy while treating endpoint latency, context policy, and runtime behavior as system properties. The registry stores run IDs, endpoint classes, non-secret variables, task checksums, and sanitized configurations. Qwen3.6 replacements link to their original trials; reward-zero outcomes are unchanged.

K.2 Prompt and tool protocol

The instruction has no benchmark-specific preamble. Agents use the frozen scaffold's stock prompt and tools to inspect files, run commands, and access documented task interfaces. Web tools follow one policy and are unnecessary; no prompt hints at future-critical state or transition.

The artifact records the scaffold and tool protocol, redistributable prompt or release identifiers, behavior-affecting non-secret variables, configuration hashes, and sanitized run specifications. These are versioned with the registry; credentials and private endpoints are excluded.

K.3 Attempts, retries, and stopping

The first prespecified attempt is primary. Only predefined infrastructure failures, such as sandbox creation failure or documented service outage, may be replaced; agent errors, verifier failures, and malformed output may not. Provenance links each replacement to its original failure, and a complete row requires one terminal record per task.

L Limitations, Contamination, and Data Governance

Controlled executable environments support intervention and exact verification but omit some organizational context, uncertainty, and social consequences of production systems. Scores measure these tasks and interfaces, not general foresight or intelligence, and may reflect scaffold policy or systems familiarity.

Hidden runtime state limits lookup of task-specific values, but public release can expose task structure and strategies. We version task text, images, causal cards, and verifier hashes; retain a private mechanism-level holdout where permitted; distinguish pre- and post-release model dates; and do not treat lexical novelty alone as contamination protection.

The inventory records software and source-derived assets. Synthetic records contain no real credentials, customer data, or identifying information. The anonymous artifact excludes API keys, internal endpoints, infrastructure tokens, and unrelated logs; its release card documents image provenance, security-sensitive operations, and the research-use boundary.

M Artifact and Reproducibility Checklist

The artifact contains the frozen corpus and Controls, pair map, causal cards, validation logs, sanitized registry, long-form exports, and scripts. Its inventory indexes 740 terminal records and 730 ATIF trajectories; ten missing and two truncated trajectories are declared without imputation. It records hashes and Qwen3.6 recovery lineage (15 Challenge, one Control) under the stated policy. Reproduction validates manifests, image digests, Oracle/NOP runs, completeness, checksums, exports, manuscript generation, all 40 source diffs, the release inventory, and retry provenance (Table 11).