DOES THE WORLD WAIT WHILE AGENTS THINK?
A TEMPORAL FIDELITY AUDIT OF INTERACTIVE-AGENT BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive agents often spend seconds reasoning before they act, yet benchmarks disagree on what should happen in that interval: some freeze the task, others keep wall, episode, or simulated time running. Calling both settings "dynamic" hides whether state progresses, whether the evaluator charges the interval, and what elapsed time can change afterward. We introduce the **Temporal Contract Specification** (TCS) to record these three aspects in plain terms. Across a frozen set of eleven pinned interactive-agent suites, public descriptions fully determined the audited contract in 3 cases, 5 required implementation-level qualification, and 3 remained semantically underspecified. Holding action, state, legality, and success fixed, a MiniWoB++ *contract swap* shows that changing only timer accounting moves the official score by 0.185 at a two-second delay (95% CI $[0.165, 0.200]$). Controlled probes on RealtimeGym Freeway and a *project* AndroidWorld lifetime protocol further show that progression and hard expiry can change viability or success under matched actions. As inference latency grows, time spent reasoning is part of the evaluation contract and should be reported explicitly.

1 INTRODUCTION

Modern interactive agents can spend several seconds reasoning before they act. What that interval means depends on what the *benchmark* does while the agent is deciding. Imagine the same observation, the same intended action, and the same five seconds of thought. In one setting the task stays frozen until the action is dispatched. In another, a wall clock, an episode timer, or a simulated offset keeps running. By dispatch time, score may have decayed, evidence may have expired, or the action may no longer be safe, even though nothing about the model's choice changed.

Calling both environments "dynamic" hides this difference. Three questions matter separately (Figure 1): does *task state* change during think, does the *evaluator* count that interval, and can elapsed time change the *opportunity* (score, evidence, safety, or validity)? A frozen world does not imply an uncharged decision interval, and neither implies that the opportunity stays unchanged.

Prior work studies how agents should operate under temporal constraints or introduces particular real-time evaluation protocols (RealtimeGym / AgileThinker (Wen et al., 2026), RRARA (Zheng et al., 2025), AAPT (Dong et al., 2026), GameWorld-RT (Ouyang et al., 2026), VSAS-Bench (Vasu et al., 2026), Gaia2 / ARE (Froger et al., 2025), DynamicGUIBench (Liu et al., 2026)), alongside interactive suites that motivate the measurement setting (Rawles et al., 2025; Xie et al., 2024; Zhou et al., 2024b; Ma et al., 2026; Muennighoff et al., 2025; Snell et al., 2024). We ask what temporal contract an evaluation harness actually implements. Across established benchmarks, state progression, decision-time accounting, and opportunity consequences are separately specified, often cannot be inferred from high-level descriptions alone, and can be mixed independently across suites, backends, and modes.

We answer with the **Temporal Contract Specification** (TCS), which records those three aspects and makes them checkable via executable probes and pinned-code inspection (Section 3). We apply TCS to eleven suites frozen under a pre-specified lockfile spanning mobile GUI, games, async tools, web and desktop environments, embodied disaster simulation, and manipulation. In this frozen set, public descriptions fully determined the audited temporal contract in 3 cases, 5 required implementation-level qualification, and 3 remained semantically underspecified (Table 3).

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.

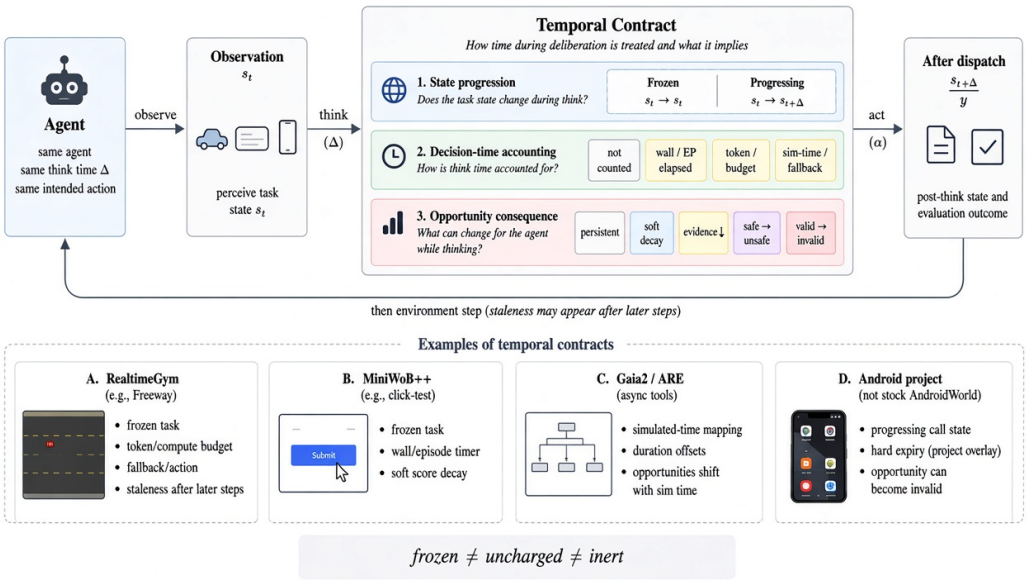

Figure 1: **Temporal Contract Specification** along observe→think→act: state progression, decision-time accounting, and opportunity consequences. Examples at bottom; slogan: frozen $\neq$ uncharged $\neq$ inert.

Fixed-action probes then ask whether contract differences can change measured outcomes. A MiniWoB++ *contract swap* is the cleanest identification case: same scripted click, same delay, same state and legality, but native wall-clock timer accounting versus a project contract that excludes injected delay from the episode timer. Official reward drops by 0.185 at two seconds (95% CI $[0.165, 0.200]$) with raw reward and success matched. On RealtimeGym Freeway, a controlled progression probe can make initially safe actions unsafe even though the audited think path freezes task state. In a *project* AndroidWorld protocol with finite call lifetime, extra dispatch delay interacts sharply with opportunity expiry.

Contributions.

1. A cross-suite temporal audit of eleven pinned interactive-agent benchmarks.
2. The Temporal Contract Specification with executable probes for reporting and comparison.
3. Fixed-action probes showing that timer accounting, controlled progression, and project hard-expiry overlays can change measured outcomes without changing the chosen action.

2 Related Work

Real-time and latency-aware agents. A growing line of work asks how agents should act when deliberation is slow relative to the environment. RealtimeGym and AgileThinker study agents under time pressure in evolving games (Wen et al., 2026). RRARA couples a rapid reflex policy with asynchronous LLM reflection and introduces a Time Conversion Mechanism that maps wall-clock deliberation into simulator frames on HAZARD (Zheng et al., 2025; Zhou et al., 2024a). AAPT addresses correct-but-late GUI failures by moving decode off the decision-time critical path (Dong et al., 2026). Timely Machine emphasizes wall-clock test-time scaling with tool latency (Ma et al., 2026), and streaming evaluation protocols argue that lockstep evaluation can overstate real-time competence (Vasu et al., 2026). Beyond explicit real-time controllers, reasoning-and-acting agents such as ReAct (Yao et al., 2023b), deliberate search methods such as Tree of Thoughts (Yao et al., 2023a), and cognitive-architecture syntheses such as CoALA (Sumers et al., 2024) make multi-step inference a first-class part of the control loop. Cooperative embodied agents (Zhang et al., 2024) and VLA systems likewise treat compute and latency as deployment constraints (Brohan et al.,

2023; Li et al., 2024), while ALFWorld aligns text agents with embodied interaction (Shridhar et al., 2021). These works primarily advance *agents*, *controllers*, or *single-suite protocols*. Our fixed-action probes instead isolate the harness contracts those systems run under.

Dynamic interactive-agent benchmarks. Interactive evaluation has expanded quickly across web, GUI, tool, and embodied domains. Web and GUI suites include WebArena / VisualWebArena (Zhou et al., 2024b; Koh et al., 2024), Mind2Web (Deng et al., 2023), WebShop (Yao et al., 2022), instruction-finetuned multimodal web navigation (Furuta et al., 2024), MiniWoB / MiniWoB++ (Shi et al., 2017; Liu et al., 2018), SeeClick (Cheng et al., 2024), AndroidWorld (Rawles et al., 2025), OSWorld (Xie et al., 2024), and DynamicGUIBench (Liu et al., 2026). Tool-use and coding evaluations include Gaia2 / ARE (Froger et al., 2025), AgentBench (Liu et al., 2024), SWE-bench (Jimenez et al., 2024), and compositional tool-calling settings such as Chameleon (Lu et al., 2023). Embodied suites such as HAZARD, SIMPLER, and LIBERO stress dynamic or manipulation settings (Zhou et al., 2024a; Li et al., 2024; Liu et al., 2023). OSWorld-Human studies where computer-use agents spend execution time (Abhyankar et al., 2026); GameWorld-RT and VSAS-Bench target non-pausing or streaming evaluation (Ouyang et al., 2026; Vasu et al., 2026). We treat these suites as *measurement subjects*, separating stock harness behavior from project overlays and non-native protocol variants. GameWorld-RT and VSAS-Bench are classified under TCS in Appendix A but were not part of the eleven pinned audit lockfile. Even among published interactive benchmarks, high-level labels such as "dynamic" or "real-time" rarely state whether task state progresses during think, how the evaluator charges that interval, or what elapsed time can change afterward.

Benchmark measurement and evaluation semantics. Reproducibility and evaluation-contract work stress that harness and protocol choices shape measured progress (Pineau et al., 2021; Henderson et al., 2018; Bowman & Dahl, 2021). Broader critiques of benchmark culture and holistic evaluation likewise argue that what is measured, and how, must be stated explicitly (Raji et al., 2021; Liang et al., 2023). Resource- and SLA-oriented agent benchmarks treat latency as a cost axis (Madiraju & Madiraju, 2026), and system-aware evaluation warns that ignoring infrastructure can distort comparisons (Dehghani et al., 2022). Classical value-of-computation views treat deliberation as costly under deadlines (Russell & Wefald, 1991); test-time scaling makes inference duration routinely nontrivial (Muennighoff et al., 2025; Snell et al., 2024). Taken together, these lines motivate reporting temporal semantics as part of the evaluation contract rather than leaving them implicit in harness code. We study how evaluation *harnesses* specify state progression, decision-time accounting, and opportunity consequences across established interactive-agent suites. Table 1 contrasts that measurement question with neighboring lines of work.

Table 1: Primary questions of neighboring lines of work versus this paper.

Neighbor	Primary question
RealtimeGym / AgileThinker	How should agents reason under real-time/dynamic constraints?
GameWorld-RT	How do paused vs real-time evaluation modes affect game agents?
VSAS-Bench	How should streaming visual assistants be evaluated in real time?
RRARA / TCM	How should inference latency map into simulator progression?
AAPT	How can GUI agents avoid being correct-but-late?
Gaia2 / ARE	How should simulated generation time enter async tool environments?
DynamicGUIBench	How should agents handle transient/dynamic GUI evidence?
This paper	What temporal contract does an evaluation harness instantiate, how can it be operationally measured, and how does it differ across established suites?

That table makes the contrast explicit: neighboring lines ask how to build or evaluate agents under particular temporal protocols, while we ask what contract a harness actually implements. We next formalize a shared three-question vocabulary (state progression, decision-time accounting, and opportunity consequences) as the Temporal Contract Specification.

3 TEMPORAL CONTRACT SPECIFICATION

A **Temporal Contract Specification** (TCS) is a structured, multi-hot reporting specification for what a benchmark does in the decision interval between observation and action dispatch (Figure 1). Assignments may combine options when behavior varies across task, execution mode, or substrate. Informal labels such as "dynamic" or "real-time" leave open whether task state moves, whether the evaluator charges the interval, and what elapsed time can change afterward.

Write s_t for the task state when deciding begins, Δ for decision duration, a for the selected action, and y for an evaluation outcome. A harness implements, often only implicitly,

$$s_{t+\Delta} = \Phi(s_t, \Delta), \qquad \tau = \Psi(\tau_t, \Delta, \kappa), \qquad y = \Gamma(s_{t+\Delta}, a, \tau). \tag{1}$$

Φ is state progression, Ψ is evaluator accounting (duration Δ, optional prior resources τ_t, and non-temporal metrics κ such as tokens), and Γ is the opportunity consequence. Shorthand $\tau = \Psi(\Delta)$ covers a single step; cumulative episode timers and token caps use the fuller signature. Table 2 lists typical options on each axis.

Table 2: The three TCS axes at a glance (multi-hot options).

Axis	Asks	Typical options
State progression Φ	Does *task* state change during think?	Frozen; step-triggered; async/wall; simulated advance
Decision-time accounting Ψ	Does the *evaluator* charge the interval?	Not counted; wall/EP timer; token/budget; sim-time; fallback
Opportunity consequence Γ	What can elapsed time change afterward?	Persistent; soft decay; evidence expiry; staleness; hard expiry

How to read the axes. Timers that only affect scoring are Ψ, not Φ, unless they mutate task state. Frozen Φ does *not* imply uncharged Ψ (MiniWoB++: frozen page + wall EP decay; RealtimeGym: frozen Freeway + budgets/fallbacks). Γ records what elapsed or accounted time is allowed to change afterward: some options are harness mechanisms (episode timers, expiry hooks), while others are task- or state-dependent effects once a transition exists (e.g., action staleness). Reporting both keeps suite comparisons honest. Opportunity effects separate cleanly: soft decay can lower score while success remains; staleness can leave an action legal yet unsafe; hard expiry can make success impossible. In step-frozen suites, latent staleness may appear only after later environment steps $s' \sim T(s_{t+\Delta}, a)$, where T is the post-dispatch transition (`env.step` / physics update), distinct from deliberation-time Φ.

Contract contrast. Holding start state, action, and delay fixed,

$$E_{\text{contract}}(\delta) = Y(s, a, \delta; \mathcal{C}_1) - Y(s, a, \delta; \mathcal{C}_0), \tag{2}$$

with $Y(s, a, \delta; \mathcal{C}) \equiv \Gamma_{\mathcal{C}}\big(\Phi_{\mathcal{C}}(s, \delta),\, a,\, \Psi_{\mathcal{C}}(\delta)\big)$, isolates changing the temporal contract. Section 5 uses this for the MiniWoB timer swap.

Measuring a contract. Executable probes make TCS checkable (Appendix A): Think-Freeze, Same-Action Staleness, Decision-Time Accounting, and Claim-to-Implementation probes. TCS reports the three aspects separately so unlike temporal effects stay comparable on their own terms. With Φ, Ψ, and Γ in hand, the next section asks what contracts eleven pinned suites actually implement, and how far public descriptions determine those answers. The audit therefore reads each suite as a measurement subject under a shared three-axis lens. Where public claims are incomplete, implementation-level inspection and feasible probes are used to recover the contract that the pinned harness actually runs. We also record evidence tiers and claim-to-implementation relations so readers can see which rows rest on high-confidence probes versus medium-confidence code-and-docs inspection.

4 What Does Time Mean Across Agent Benchmarks?

4.1 Benchmark selection

We study eleven suites spanning mobile GUI, games, async tools, web/desktop, embodied disaster, and VLA/manipulation. Inclusion criteria were fixed *before* seeing audit outcomes: established public benchmarks with diverse modalities and timing semantics, plus public papers and (where possible) public code. The set is diverse but **not** exhaustive.

4.2 Audit protocol

For each suite we compare public descriptions to pinned code, run feasible probes, and assign an evidence tier (HIGH or MEDIUM; Appendix A) and a claim-to-implementation relation (Appendix A). Per-suite traces are in Appendix A. Outcome-changing claims rely on HIGH-confidence rows plus the pre-specified Android *project* lifetime study.

4.3 What we found

Four patterns recur across the eleven frozen suites (Table 3).

Table 3: **Temporal contracts across eleven pinned suites** (stock harness unless noted). Appendix A lists formal multi-hot codes aligned with this table.

Benchmark	Domain	During decision	How time is counted	What time can change	Ev.	Desc. vs impl.
AndroidWorld[†]	mobile GUI	live device; step loop	not charged in harness[*]	budgets; persistent	Med.	Partial
RealtimeGym	games	**frozen task**; step on act	**budgets; fallback**	staleness; budgets	High	Underspecified
Gaia2/ARE	async tools	frozen; sim advance	**sim-time offset**	windows; expiry	High	Partial
DynamicGUIBench	desktop GUI	host-dependent	none in harness	evidence; family-dependent staleness / expiry	Med.	Partial
MiniWoB++	web UI	**frozen task**	**wall EP elapsed**	**soft decay**	High	Match
WebArena	web	step loop; live page[‡]	none	mostly persistent	Med.	Underspecified
VisualWebArena	visual web	step loop; live page[‡]	none	mostly persistent	Med.	Underspecified
OSWorld	desktop VM	step loop; VM may tick[‡]	none	mostly persistent	Med.	Partial
HAZARD[§]	embodied	frozen; step on act	none (stock)	budgets; damage	Med.	Partial
SIMPLER	VLA sim	frozen; step on act	none	horizon; persistent	High	Match
LIBERO	manip. sim	frozen; step on act	none	horizon; persistent	High	Match

[†] Project lifetime hard expiry is a project overlay; Sec. 5.3. [*] Stock harness does not charge deliberation; live emulator wall time may still continue. [‡] Browser or VM substrate may advance during think even when the agent loop is step-based; audited opportunity labels remain persistent / budget-limited absent suite-native soft decay. [§] RRARA frame mapping is a paper protocol variant on HAZARD.

Many step-based harnesses freeze task state during inference: the simulator does not advance until an action is dispatched. Frozen task state can still leave the decision interval charged: evaluators may still count wall time, apply token budgets, or trigger fallbacks. Temporal consequences also differ by suite (soft reward decay, evidence expiry, action staleness, or hard opportunity expiry) and call for separate reporting. MiniWoB++ illustrates the split cleanly: the probed page can stay frozen while a wall episode timer soft-decays official reward; RealtimeGym Freeway freezes during think yet reveals staleness only after later environment steps.

Public descriptions often under-specify these behaviors. In this frozen set of eleven suites, public descriptions fully determined the audited temporal contract in **3** cases (*Match*), **5** required implementation-level qualification (*Partial*), and **3** remained semantically underspecified (*Underspecified*): wording such as "dynamic"/"real-time" leaves the contract underdetermined until code is inspected (definitions in Appendix A). RealtimeGym is the boundary case: high-confidence probes show a frozen task during think with active budgets/fallbacks, while "evolving real-time" framing alone leaves progression during the decision interval open.

A second coder labeled prepared evidence packets and agreed on all eleven claim-to-implementation categories (Cohen κ=1.0). Exact field agreement was 100% for decision-time accounting, 54.5% for state progression, and 63.6% for opportunity labels; Jaccard overlap is higher, and disagreements concentrate on live or mixed substrates. Appendix A reports full rates and mechanical decision

rules. These audit findings set up the fixed-action probes in Section 5, which ask whether changing a contract while holding the action fixed can move official scores, viability, or success. The probes keep the intended action matched so any outcome change can be attributed to the temporal treatment rather than to a different decision. Together they cover accounting (MiniWoB++), controlled progression (Freeway), and hard expiry under a project overlay (Android), as complementary mechanism checks rather than a single pooled effect size.

5 When Timing Changes Measured Outcomes

The audit shows heterogeneous contracts. We next use fixed-action probes to show how those contracts can change measured outcomes when only the temporal treatment varies. Figure 2 summarizes three complementary signatures on their native units. Gaia2 provides simulated-time calibration of the accounting mechanism. MiniWoB uses a stock timer rule under a controlled accounting swap; Freeway and Android use *counterfactual* progression or project lifetime overlays that stress the contract under controlled conditions.

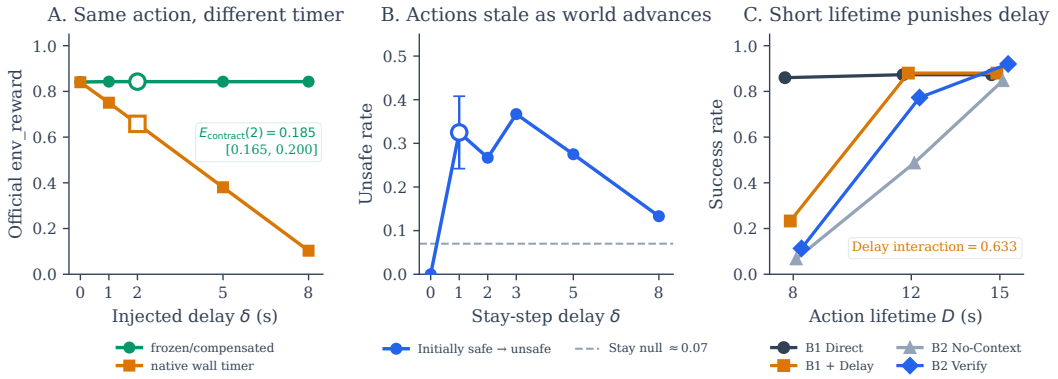

Figure 2: **Fixed-action temporal signatures.** **(A)** Same MiniWoB++ click and delay under native wall timer vs compensated timer accounting; official reward differs by 0.185 at 2 s while state and legality match. **(B)** After controlled Stay-step injections on Freeway, initially safe actions often become unsafe (32.5% after one step at the primary delay; Appendix A). **(C)** In a project AndroidWorld protocol, short call lifetime sharply penalizes delay-only dispatch (pooled success curves; delay interaction 0.633, CI $[0.540, 0.727]$).

5.1 Same action, different timer contract (MiniWoB++)

We isolate timer accounting on MiniWoB++ `click-test-2` (protocol in Appendix A). For matched seeds we replay the *same* scripted correct click after the *same* injected delay under native stock wall-clock timing versus a *project* compensated contract. In the compensated arm, after `sleep`(δ) we advance the episode timer origin by δ (`core.ept0 += delay_ms`, reschedule `EP_TIMER`, and advance Python `start_time`) so the injected sleep is excluded from official timer accounting while DOM, click target, legality, and success criteria stay identical. Everything matches across contracts except how elapsed wall time enters the official score `env_reward` = `raw` $\cdot \max(0, 1 - dt/EP_{\max})$ (default $EP_{\max}$=10 s; dt is episode wall elapsed): legal rate 1.0, success 0.925, raw reward 0.85, and state hashes stable over delay. At a two-second delay (n=40 paired seeds), official `env_reward` is lower under native accounting by $E_{\text{contract}}(2)$=0.185 (95% CI $[0.165, 0.200]$; Figure 2A). Here $E_{\text{contract}}(\delta)$ is Eq. (2) with Y the official `env_reward` under native vs compensated timer contracts. The full paired gap grows with delay (0.001, 0.092, 0.185, 0.462, 0.740 at $\{0, 1, 2, 5, 8\}$ s), while δ=0 stays near zero as a sanity check. The measured gap tracks the known soft-decay formula once injected delay enters dt; the experiment's role is identification (holding state, action, legality, success, and raw reward fixed), not discovering that the timer depends on elapsed time. A pre-specified breadth check on eight MiniWoB tasks finds the same soft timing pattern beyond `click-test-2`: five of five successfully scripted tasks reproduce the effect; three additional tasks could not be reliably scripted for infrastructure reasons (Appendix A).

Table 4: **Android project pooled success rates** (150 episodes/cell; project lifetime protocol). These cells underlie the delay and evidence contrasts below.

Policy	D=8	D=12	D=15
B1 Direct	0.860	0.873	0.873
B1+Delay	0.233	0.880	0.880
B2 No-Context	0.067	0.487	0.847
B2 Verify	0.113	0.773	0.920

5.2 Actions become stale as the world advances (Freeway)

Freeway under the audited pin freezes task state during `think()`, yet the environment can still make an action unsafe after controlled steps. We therefore inject Stay steps as a *counterfactual* progression probe: among N=120 states where action `U` is initially collision-safe, one Stay-step delay leaves **32.5%** of those actions unsafe (95% CI $[24.2\%, 40.8\%]$; Figure 2B). This is the stale/unsafe action rate $\mathrm{SAR}_U(\delta)$: the fraction of initially safe actions that become collision-unsafe after δ Stay steps under T. A Stay-only baseline stays near 7%; the cohort-matched contrast is 0.255 $[0.162, 0.348]$ (Appendix A). The full SAR curve is non-monotonic (0, 0.325, 0.267, 0.367, 0.275, 0.133 at $\delta \in \{0, 1, 2, 3, 5, 8\}$), matching periodic Freeway traffic. The harness can therefore apply decision budgets during think while staleness appears only after progression. Under the pinned RealtimeGym loop (`observe`→`think`→`act`→`env.step`), token budgets cap generation, seconds budgets can truncate streams, and empty parses fall back to a default action, all without advancing Freeway state during blocking think (Appendix A).

5.3 Short-lived opportunities punish delay (Android project)

In a *project* AndroidWorld protocol with scheduled call expiry, incoming calls expire after a scheduled `gsm cancel` at lifetime $D \in \{8, 12, 15\}$ s on `PhoneAnswerCall` (5 VLMs × 4 policies × 3 lifetimes × 30 repeats =1800 episodes). Policies span direct one-shot dispatch (**B1 Direct**), the same action with calibrated pre-dispatch delay (**B1+Delay**), an evidence pass then discard (**B2 No-Context**), and evidence-conditioned verification (**B2 Verify**). B1+Delay inserts $\mathrm{delay} = \max(0, T_{\mathrm{B2,target}} - T_{\mathrm{elapsed}})$, where T_{elapsed} is wall time already spent after the B1 action is chosen and $T_{\mathrm{B2,target}}$ is a pre-sampled two-pass B2 latency for the same model, so dispatch latency matches that target; calibration medians are approximately 1.3–3.9 s for B1 and 4.7–7.8 s for B2 across the five VLMs (Appendix A).

Let $\mathrm{SR}(P, D)$ denote success rate for policy P at call lifetime D. The primary pooled delay-only lifetime interaction is

$$E2 = \big[\mathrm{SR}(B1{+}\mathrm{Delay}, 15) - \mathrm{SR}(B1, 15)\big] - \big[\mathrm{SR}(B1{+}\mathrm{Delay}, 8) - \mathrm{SR}(B1, 8)\big], \qquad (3)$$

i.e., how much extra the B1+Delay vs B1 gap grows when the opportunity window lengthens from 8 s to 15 s. We estimate this interaction at 0.633 (95% CI $[0.540, 0.727]$; Holm p=0.00030). Companion contrasts (Appendix A) are $E1 = [\mathrm{SR}(B2, 15) - \mathrm{SR}(B1, 15)] - [\mathrm{SR}(B2, 8) - \mathrm{SR}(B1, 8)] = 0.793$ (evidence-conditioned vs direct across lifetimes) and $E3 = \mathrm{SR}(B2, 15) - \mathrm{SR}(B2\text{-NoContext}, 15) = 0.073$ (verification vs discarded evidence at D=15). Table 4 makes the short-lifetime collapse visible: at D=8, delay-only success is 0.233 while direct dispatch remains 0.860; at D=15 both are near 0.88 (Figure 2C). The pooled delay-only lifetime interaction averages across five VLMs, with model-specific values from 0.167 to 1.000 (Figure 3); per-model 4×3 success matrices are in Appendix A. Under this project protocol, the result is existence / sufficiency evidence that delay can interact with finite opportunity lifetime.

5.4 Simulated-time calibration (Gaia2)

Gaia2 exposes an explicit mapping from decision time to simulated time. Under official `fixed` mode, configured offsets advance simulated time exactly on the pinned commit. This confirms the accounting mechanism on the pinned commit. On the pinned local backend, `measured` mode advances zero simulated time without duration metadata, a backend-specific caveat detailed in Appendix A.

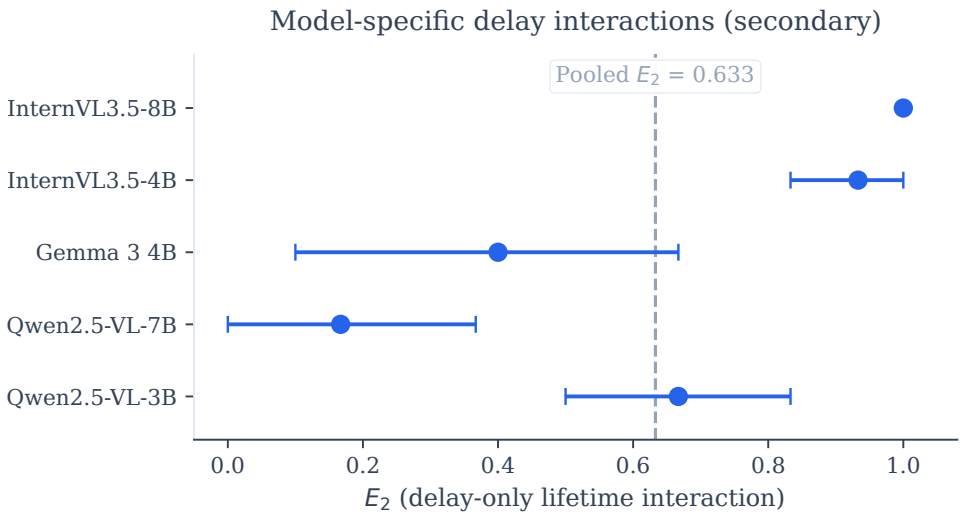

Figure 3: **Model-specific delay interactions** under the Android project protocol (95% CIs; dashed line is the pooled estimate 0.633). Per-model secondary estimates.

Under fixed actions, changing timer accounting, allowing progression, or shortening opportunity lifetime can change measured reward, viability, or success. These open-loop probes isolate harness mechanics so that adaptive or latency-aware controllers can later be compared under the same contracts (Section 7). The practical upshot is that temporal semantics are part of what a benchmark measures, which motivates the reporting recommendations next.

6 Discussion

Deliberative agents make multi-second inference routine, yet benchmarks still describe timing with vague words like "dynamic" or "real-time." Our results suggest a simpler message: *time spent reasoning is partly an evaluation-design choice*, and therefore part of what the benchmark measures.

When two suites both sound dynamic but one freezes the task while another counts wall time or simulated offsets, cross-suite comparisons mix different evaluation questions. Freezing during think can be appropriate when the goal is decision quality in isolation. The practical recommendation is to state the contract explicitly whenever inference latency is nontrivial. Wall-clock accounting can also be sensitive to hardware and network jitter; reporting backend and timing variance helps readers interpret charged intervals.

Freeway illustrates why the three aspects should be reported separately. Task state can remain frozen during think while budgets still apply and while controlled progression later makes actions unsafe. MiniWoB shows that wall-clock accounting alone can change official scores even when the task and action are unchanged. Latency-aware agent designs (reflex/async reflection, critical-path decode offloading, or anticipatory policy trees) may mitigate staleness or expiry under non-pausing contracts; our contribution is to make those contracts measurable so such comparisons are fair.

Reporting checklist. When releasing or comparing interactive-agent benchmarks, we recommend stating at least:

- **State progression:** Does task/environment state change during the decision interval? If so, is advance step-triggered, wall-async, or simulated?
- **Decision-time accounting:** Does the evaluator ignore the interval, count wall time, apply budgets/deadlines, map duration into simulated time, or trigger fallbacks?
- **Opportunity consequences:** Can elapsed or accounted time soft-decay reward, expire evidence, make actions stale, or close opportunities entirely?
- **Implementation / backend:** Are these semantics backend- or mode-dependent?

An expanded prompt list is in Appendix A. The checklist is intentionally short so it can sit beside a suite release; the limitations that follow bound how far the present audit and probes should be read.

7 LIMITATIONS

Our audit covers a diverse sample of interactive-agent benchmarks; some suites rest on medium-confidence evidence where full runtime probes were impractical. The outcome-changing studies use controlled, fixed-action interventions to isolate harness mechanics, including project overlays and explicit progression probes. Closed-loop or latency-aware agent architectures under those contracts, preferred-contract prescriptions, ranking reversals, and pooled scores across unlike temporal effects are outside the present scope. Some classifications remain judgment-sensitive, especially state progression on live substrates; secondary coding used prepared evidence packets. The reporting vocabulary may also need extension for fully concurrent or continuous-control settings (Appendix A).

8 CONCLUSION

Interactive benchmarks implement different answers to a simple question: what happens while the agent is deciding? Some freeze the task; some count wall or simulated time; some soft-decay reward, make actions stale, or close opportunities entirely. We introduce the Temporal Contract Specification to report these behaviors as separately specified aspects (state progression, decision-time accounting, and opportunity consequences) and audit eleven pinned interactive-agent suites under that lens. In this frozen audit set, public descriptions fully determined the audited temporal contract in 3 cases, 5 required implementation-level qualification, and 3 remained semantically underspecified. Fixed-action interventions then show that the contract is operational as well as descriptive: changing only timer accounting, applying a controlled progression probe, or imposing a project hard expiry overlay can change measured outcomes while the chosen action stays fixed. Time spent reasoning is part of the benchmark's evaluation contract, and that contract should be stated explicitly whenever inference latency is nontrivial.

REPRODUCIBILITY STATEMENT

Suite selection and commit pins were fixed before the cross-suite temporal audit outcomes. The fixed-action protocols and Android project design use pre-specified identifiers (full 40-character SHA-1 hashes in Appendix A; protocol hashes in Appendix A). Pinned commits, evidence tiers, probes, the full multi-hot matrix, branching details, Android latency calibration and per-model matrices, and secondary estimates are in the appendix. Anonymous code and probe scripts will be provided as supplementary material.

ETHICS STATEMENT

This work audits public interactive-agent benchmarks and does not involve human subjects or sensitive personal data. The Android project lifetime protocol uses an emulator-scheduled telephony cancel as a controlled hard-expiry case study on one task family. We follow the ICLR Code of Ethics.

AI USE STATEMENT

We used generative AI tools as writing assistants while preparing this manuscript: to help draft and rephrase prose, tighten wording, check consistency of terminology and cross-references, and suggest clearer ways to explain technical ideas. We also used them for literature discovery and retrieval, such as locating related papers and helping organize citation candidates. The authors directed this use, reviewed every AI-assisted suggestion, and verified all scientific claims, results, and interpretations. We take full responsibility for the final paper.

REFERENCES

Reyna Abhyankar, Qi Qi, and Yiying Zhang. OSWorld-Human: Benchmarking the efficiency of computer-use agents. In *Proceedings of Machine Learning and Systems (MLSys)*, 2026. arXiv:2506.16042.

Samuel R. Bowman and George E. Dahl. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021. arXiv:2104.02145; NAACL proceedings.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2023.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2024. arXiv:2401.10935; ACL Anthology.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2110.12894.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.06070; NeurIPS 2023 Spotlight.

Zihan Dong, Rui Qian, Qishi Zhan, Dongshen Peng, Kaixin Li, and Yu Li. Why are GUI agents correct but late? decode on the decision-time critical path, tested with pre-compiled policy trees. *arXiv preprint arXiv:2607.28399*, 2026. AAPT.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, et al. ARE: Scaling up agent environments and evaluations. *arXiv preprint arXiv:2509.17158*, 2025. Gaia2 / Meta Agents Research Environments.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.11854; accepted ICLR 2024.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018. arXiv:1709.06560.

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 (ICLR)*, 2024. arXiv:2310.06770; OpenReview.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2024. arXiv:2401.13649.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. SIMPLER / SimplerEnv.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research (TMLR)*, 2023. arXiv:2211.09110; HELM.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS Datasets and Benchmarks Track*, 2023. arXiv:2306.03310.

Enqi Liu, Liyuan Pan, Zhi Gao, Yan Yang, Chenrui Shi, Yang Liu, Jingrong Wu, and Qing Li. Benchmarking and improving GUI agents in high-dynamic environments. *arXiv preprint arXiv:2604.25380*, 2026. DynamicGUIBench.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1802.08802; MiniWoB++.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2308.03688; published in ICLR 2024.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2304.09842; accepted NeurIPS 2023.

Yichuan Ma, Linyang Li, Yongkang Chen, Peiji Li, Xiaozhe Li, Qipeng Guo, Dahua Lin, and Kai Chen. Timely machine: Awareness of time makes test-time scaling agentic. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2026. arXiv:2601.16486; ACL 2026.

Meher Bhaskar Madiraju and Meher Sai Preetam Madiraju. AgentSLABench: Evaluating and benchmarking agentic systems under resource constraints. *arXiv preprint arXiv:2608.00805*, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Mingyu Ouyang, Siyuan Hu, Kevin Qinghong Lin, Hwee Tou Ng, and Mike Zheng Shou. GameWorld: Towards standardized and verifiable evaluation of multimodal game agents. *arXiv preprint arXiv:2604.07429*, 2026. includes GameWorld-RT real-time variant.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research*, 22, 2021. arXiv:2003.12206.

Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. AI and the everything in the whole wide world benchmark. In *NeurIPS Datasets and Benchmarks Track*, 2021. arXiv:2111.15366; accepted NeurIPS 2021 D&B.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, et al. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2405.14573.

Stuart Russell and Eric Wefald. *Do the Right Thing: Studies in Limited Rationality*. MIT Press, 1991.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning (ICML)*, 2017. MiniWoB.

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 (ICLR)*, 2021. arXiv:2010.03768.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2309.02427; TMLR camera-ready.

Pavan Kumar Anasosalu Vasu, Cem Koc, Fartash Faghri, Chun-Liang Li, Bo Feng, Zhengfeng Lai, Meng Cao, Oncel Tuzel, et al. VSAS-Bench: Real-time evaluation of visual streaming assistant models. *arXiv preprint arXiv:2604.07634*, 2026.

Yule Wen, Yixin Ye, Yanzhe Zhang, Diyi Yang, and Hao Zhu. Real-time reasoning agents in evolving environments. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2511.04898; RealtimeGym / AgileThinker.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. arXiv:2404.07972.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2207.01206; NeurIPS camera-ready.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a. arXiv:2305.10601; NeurIPS 2023 camera-ready.

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)*, 2023b. arXiv:2210.03629; ICLR camera-ready.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2307.02485; ICLR 2024.

Yangqing Zheng, Shunqi Mao, Dingxin Zhang, and Weidong Cai. LLM-enhanced rapid-reflex async-reflect embodied agent for real-time decision-making in dynamically changing environments. *arXiv preprint arXiv:2506.07223*, 2025. RRARA agent; Time Conversion Mechanism (TCM) $F{=}T \cdot \mathrm{FPS}$ on HAZARD.

Qinhong Zhou, Sunli Chen, Yisong Wang, Haozhe Xu, Weihua Du, Hongxin Zhang, Yilun Du, Joshua B. Tenenbaum, and Chuang Gan. HAZARD challenge: Embodied decision making in dynamically changing environments. In *International Conference on Learning Representations (ICLR)*, 2024a. arXiv:2401.12975.

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 *International Conference on Learning Representations (ICLR)*, 2024b. arXiv:2307.13854.

A Supplementary Material

Temporal Contract Specification (formal)

A **Temporal Contract Specification** (TCS) is a multi-hot triple $(\mathrm{Q1}, \mathrm{Q2}, \mathrm{Q3})$ describing harness behavior during policy inference. It is a measurement/reporting object, not a standard.

Q1 State progression. Does **task/environment state** evolve during the decision interval (before dispatch)? Values: Frozen; Step-triggered; Async/wall evolving; Simulated-advance; Mixed/task-dependent. Evaluator timers alone do not imply Q1 unless they mutate task state.

Q2 Decision-time accounting. How does the **evaluator** account for time/resources during the decision interval? Values: None/uncharged; Wall-clock elapsed (episode timer); Token/resource budget; Wall-time deadline; Simulated-time offset; Timeout/fallback/default action; Backend-dependent. Q2 does not require detecting model computation specifically.

Q3 Opportunity consequence. What can elapsed or accounted time change? Values: Persistent action; Global episode budget; Soft score decay; Evidence expiry; State/action staleness; Hard action-opportunity expiry. Q1/Q2/Q3 are separately specified, not assumed orthogonal. Slogan: **world frozen $\neq$ decision interval uncharged $\neq$ opportunity inert.**

EVIDENCE-TIER DEFINITIONS

HIGH: pinned code inspected and at least one executable probe or unit calibration reproduced under the lockfile. **MEDIUM:** paper/docs plus code inspection, but full runtime P1 not executed (e.g., heavy VM images). **UNVERIFIED:** paper-only protocol variants without stock-repo implementation in the audited pin (e.g., RRARA TCM).

TAXONOMY DETAIL (T*/O* UNDER TCS)

Execution flags (multi-hot; refine Q1/Q2): T0 agent-driven static; T1 step-dynamic / think-frozen (P1); T2 evaluator-timed; T3 evidence-temporal (P3); T4 simulated-compute-coupled (P6); T5 wall-clock-coupled. Decision-time accounting subtypes (refine Q2): NONE / TOKEN-BUDGET / WALL-TIME DEADLINE / FALLBACK-DEFAULT / OTHER. Opportunity flags (refine Q3): O0 persistent action; O1 global episode budget; O2 soft value decay; O3 evidence expiry; O4 state/action staleness; O5 hard action-opportunity expiry. Neighbor rule: O3 concerns information availability; O4 concerns state-conditioned safety of an API-legal action; O5 concerns action possibility/validity. Flags are descriptive measurement labels, not an ordinal quality scale. Multi-hot cells may leave some subtypes unused for a given suite (e.g., T0-alone is rare in this set); unused codes are not claims that the subtype is impossible elsewhere. T2 marks evaluator timing machinery (episode timers, budgets) and may co-occur with uncharged step loops when timers exist but do not bind during think.

BENCHMARK LOCKFILE (PINNED COMMITS)

Suite	Full commit SHA-1
MiniWoB++	`33c3b4ddef8c6eb67c57a29663d844b1eda7e614`
WebArena	`dce04686a56253aefba7b18a4fa0937cf1dc987b`
VisualWebArena	`89f5af29305c3d1e9f97ce4421462060a70c9a03`
OSWorld	`fc31a9049664292fcb35d6e501ee1dc839f2cf6d`
HAZARD	`d35c71ebd24c08783552bab246df8ac82190e1e2`
SIMPLER	`06accaca93535902d408da4855f21cece12bceb7`
LIBERO	`8f1084e3132a39270c3a13ebe37270a43ece2a01`
AndroidWorld	`3e50888527ef9f29b9157ecd537e408008bb1c85`
RealtimeGym	`3d5b3ef37dafee25ca2e8d2e8d71f5fe842eda7b`
Gaia2/ARE	`87ebd38f31aafae0f11e14f55617903196236cfb`
DynamicGUIBench	`44adf69d36a9bd27fbf2173fa619eb0ee92da095`

Audit date: 2026-09-08. Farama MiniWoB++; WebArena / VisualWebArena / OSWorld / HAZARD / SimplerEnv / LIBERO / google-research android_world / SALT-NLP RealtimeGym / Meta ARE / HF DynamicGUIBench dataset.

FULL TEMPORAL MATRIX (13 ROWS)

Suite	Ev.	Active T	Active O	Δ@compute	Charge	Claim↔code
RealtimeGym	H	T1,T2	O1,O4	no	budget only	SEMANTIC_AMBIG.
Gaia2/ARE	H	T4,T2	O1,O3,O5	mode-dep	fixed/meas.	PARTIAL_MATCH
MiniWoB++	H	T2,T5	O1,O2	timer yes	wall EP	MATCH
AndroidWorld stock	M	T1,T2	O0,O1	emulator wall	no harness charge	PARTIAL_MATCH
AndroidWorld project	H	T5,T2	O5,O1	yes	wall cancel	N/A_PROJECT†
WebArena	M	T1,T2	O0,O1	mode-dep	no harness	SEMANTIC_AMBIG.
VisualWebArena	M	T1,T2	O0,O1	mode-dep	no harness	SEMANTIC_AMBIG.
OSWorld	M	T1,T2	O0,O1	mode-dep	unknown	PARTIAL_MATCH
DynamicGUIBench	M	T1,T3	O3, O4, O5	mode-dep	no	PARTIAL_MATCH
HAZARD stock	M	T1,T2	O1,O2	no	no	PARTIAL_MATCH
HAZARD RRARA TCM	M	T4,T5	O2,O4,O5	yes (paper)	TCM	UNVERIFIED code
SIMPLER	H	T1,T2	O0,O1	no	no	MATCH
LIBERO	H	T1,T2	O0,O1	no	no	MATCH

†Project overlay: formal MATCH/PARTIAL/SEMANTIC/UNVERIFIED categories apply to stock public descriptions; this row is a controlled protocol, not a stock claim↔code label. Aligned with Table 3: WebArena/VisualWebArena/OSWorld use O0,O1; DynamicGUIBench lists O3 with family-dependent O4/O5; AndroidWorld stock has no harness charge while the emulator substrate may continue.

PER-SUITE CLAIM→CODE→PROBE TRACES

Compact evidence chain used to assign TCS fields (full notes in the anonymous supplementary per-suite evidence records).

RealtimeGym (HIGH; Ambiguous). Claim: real-time reasoning in evolving environments. Code: `agile_eval.game_loop` observe→think(timeout)→act→env.step; budgets in agent base classes; `DEFAULT_ACTION` on empty parse. Probe: P1 frozen during think; P4 latent U-staleness; Q2 budgets active. TCS: Q1 frozen (step-triggered); Q2 token/s + fallback; Q3 latent staleness.

Gaia2/ARE (HIGH; Partial). Claim/docs: fixed/measured simulated generation time. Code: `SimulatedGenerationTimeConfig`; LiteLLM path may return `None` duration. Probe: P6 fixed ChargeError(20)=0; measured zero without metadata. TCS: Q1 pause+offset; Q2 sim-time (mode/backend-dependent); Q3 windows.

MiniWoB++ (HIGH; Match). Code: wall EP timer; time-proportional `env_reward`. Probe: P2/P6 wall delay soft-decays reward; legal/success stable. TCS: Q1 frozen task (probed); Q2 wall-clock elapsed (EP timer); Q3 soft score decay.

AndroidWorld stock (MEDIUM; Partial) / project (HIGH). Stock: step loop on live emulator; no stock scheduled RINGING cancel. Project: `gsm cancel` at $t_0{+}D$ (SHA ending `a689d8`); E2=0.633. TCS project: Q1 async/wall (telephony); Q2 none/uncharged (elapsed wall only); Q3 hard expiry.

WebArena / VisualWebArena (MEDIUM; Ambiguous). Step harness; live JS timers may continue; full P1 not executed. TCS: Q1 step (+ possible wall); Q2 rare; Q3 mostly persistent/soft.

OSWorld (MEDIUM; Partial). VM step loop; no full P1 in audit pin. TCS: Q1 VM may tick; Q2 step eval; Q3 mostly persistent.

DynamicGUIBench (MEDIUM; Partial). HF corpus stresses interstitial evidence; inherits host runtime. TCS: Q1 host-dependent; Q2 typically uncharged; Q3 evidence expiry.

HAZARD stock (MEDIUM; Partial) / RRARA TCM (UNVERIFIED code). Stock freezes without communicate/step; TCM ($F{=}T \cdot \mathrm{FPS}$) is paper protocol, not in pin. TCS stock: Q1 frozen; Q2 none; Q3 budgets/damage.

SIMPLER / LIBERO (HIGH; Match). Control loops freeze during `get_action`; advance on `env.step`. TCS: Q1 frozen in loop; Q2 not charged; Q3 persistent/horizon.

PROBE PROTOCOLS (P1–P7)

P1 Think-Freeze. Record state fingerprint $f(s)$; run dummy think/sleep without `env.step`; re-fingerprint. Pass-frozen if f and turn/time counters are unchanged.

P2 Controlled same-action delay. From (s_0, a), apply suite-native delay grid; evaluate success/reward/legal predicates.

P3 Evidence expiry. Test evidence presence vs control legality separately after progression.

P4 Action expiry/staleness. Estimate opportunity survival / stale-action rates among states with $V(a, s_0)$=1. **OSC** (Opportunity Survival Curve) is the fraction of initially viable actions that remain viable after delay δ; **SAR** (Stale/Unsafe Action Rate) is $1 - \mathrm{OSC}$ among the same denominator.

P5 Global budget. Confirm episode/step/time budgets bind independently of local opportunity predicates.

P6 Compute charging. Compare configured compute charge to observed env/sim advance; report **CCR** (Compute Charging Ratio: charged compute time or tokens divided by wall deliberation when defined) and ChargeError.

P7 Claim-vs-code. Map paper/docs temporal claims to pinned code paths; assign MATCH / PARTIAL_MATCH / SEMANTIC_AMBIGUITY / UNVERIFIED with evidence tier.

FIXED-ACTION PROTOCOL FREEZE

Protocol identifier: `study3-v1`. Protocol-files SHA256: `6d78e6759c56beade19c4f1a353c3b1c910da84b88e4f71450fd0e3ac65a78ca`. Branching holds (s_0, a) fixed; delay grids and viability predicates are suite-specific. Scientific exclusions require documented infrastructure failure; no post-hoc replacement of suites after seeing outcomes.

MINIWOB CONTRACT-SWAP PROTOCOL

Protocol `contract-swap-v1` on `click-test-2-v1` (commit `33c3b4ddef8c6eb67c57a29663d844b1eda7e614`). Contracts: **native** stock wall timer; **frozen/compensated** project timer-only intervention (`core.ept0 += delay_ms`; reschedule `EP_TIMER`; Python `start_time` advanced by delay). Delays $\{0, 1, 2, 5, 8\}$ s; n=40 paired seeds; scripted ONE-button click; no LLM. Primary: $E_{\mathrm{contract}}(2) = 0.185$ $[0.165, 0.200]$ (paired bootstrap B=2000). Legal=1.0, success=0.925, raw=0.85 match across contracts; DOM hashes stable over delay. The scripted click succeeds on $37/40$ paired seeds; the remaining failures are retained rather than excluded. Their exact cause is not used in the contract contrast because success is matched across both contracts. Full curve: $E(0)\approx 0.001$, $E(1)$=0.092, $E(2)$=0.185, $E(5)$=0.462, $E(8)$=0.740.

CLAIM-TO-IMPLEMENTATION RELATION

MATCH: public description materially determines the audited temporal semantics. **PARTIAL_MATCH:** directionally correct but omits an important qualifier (mode, backend, fallback, stock-vs-overlay). **SEMANTIC_AMBIGUITY:** public wording does not determine a unique Q1/Q2/Q3 assignment without implementation inspection (not an implementation contradiction). **UNVERIFIED:** insufficient evidence. Boundary example: RealtimeGym freezes task state during think yet applies decision-interval budgets/fallbacks (Q2); "evolving real-time" framing does not uniquely imply Q1 progression during inference, so the class is SEMANTIC_AMBIGUITY despite HIGH probe confidence. Frozen stock counts: MATCH 3, PARTIAL_MATCH 5, SEMANTIC_AMBIGUITY 3.

AUDIT RELIABILITY (SECOND CODER)

A second coder independently labeled *prepared evidence packets* for the 11 stock suites (not a from-scratch re-audit of all repositories). Inter-coder agreement on the claim-to-implementation *category* was 100% (Cohen κ=1.0). Under **revised** Q1/Q2/Q3 definitions, field agreement: Q1 exact 54.5% / mean Jaccard 0.74; Q2 exact 100% / Jaccard 1.0; Q3 exact 63.6% / mean Jaccard 0.91. Prior wording (Q2 "compute accounting"): Q1 exact 45.5% / Jaccard 0.71. Q1 disagreements remain mostly Mixed/task-dependent qualifier differences on step harnesses with live browser/VM/emulator substrates.

Q1 mechanical decision rules. Does **task/environment state** change after observe and before dispatch? Exclude from Q1 (assign to Q2): episode/evaluator timers that only affect scoring metadata; token or inference budgets; timeout/fallback bookkeeping; sim-time offsets that do not advance task state until applied. Exception: timers that *directly mutate* task state (e.g., scheduled telephony cancel) are Q1-relevant. Labels: **Frozen** (fingerprint unchanged); **Step-triggered** (advance only on `env.step`/dispatch); **Async/wall** (may change as wall time passes without agent step); **Simulated-advance** (configured offset advances sim/task clock); **Mixed** (different tasks/modes/substrate layers). Anchors: MiniWoB++ frozen (EP timer is Q2); RealtimeGym frozen+step-triggered; Gaia2 frozen+simulated-advance; AndroidWorld stock async/wall+step-triggered; Android project async/wall (telephony continues). The anonymous supplementary materials include coder labels and a disagreement ledger.

MINIWOB BREADTH CHECK

Protocol `miniwob-breadth-v1` (protocol-frozen before outcome inspection): eight tasks; native delays $\{0, 2\}$ s; n=20 seeds; scripted heuristics; no LLM. Five of five successfully scripted pre-specified tasks reproduced the soft timing effect (Δenv_reward$(0\to2)\approx$0.19–0.20: `click-test-2`, `click-button`, `click-dialog`, `click-link`, `click-tab`). Three additional pre-specified tasks could not be reliably scripted (`focus-text`, `click-option`, `click-checkboxes`). Scripting failures are infrastructure/method limits, not negative scientific outcomes.

FREEWAY BRANCHING STATISTICS

Baseline-U-safe N=120. At δ=1: remains safe 81, becomes unsafe 39; SAR_U=0.325, bootstrap 95% CI $[0.242, 0.408]$ (B=2000, starting-state level). Full SAR: $0, 0.325, 0.267, 0.367, 0.275, 0.133$ at $\delta \in \{0, 1, 2, 3, 5, 8\}$. Non-monotonicity is expected under periodic Freeway traffic (cars enter/leave collision lanes), not a probe artifact; Stay-null control remains near 0.07. Stay-null among Stay-viable starts (N=115): $\mathrm{SAR}_S(1)\approx$0.070. Cohort-matched contrast from the same frozen CSVs: $\mathrm{SAR}_U(1) - \mathrm{SAR}_S(1)$=0.255 $[0.162, 0.348]$ (independent start-level bootstrap, B=2000, seed 20260908). Paired-on-intersection sensitivity (N=115): 0.261 $[0.174, 0.348]$. P1: world frozen during `think()`; CCR=0 for stock think path under pin `3d5b3ef37dafee25ca2e8d2e8d71f5fe842eda7b`. Stay-step injection is a counterfactual progression probe, not native think-time world advance. Additionally, official token/seconds budgets and `DEFAULT_ACTION` fallbacks provide decision-interval accounting (Q2) without task-state advance (Q1) (Appendix A).

MINIWOB MATCHED-DELAY DETAILS

Task `miniwob/click-test-2-v1`; n=40 seeds; delays $\{0, 1, 2, 5, 8\}$ s. Official $\texttt{env_reward} = \texttt{raw} \cdot \max(0, 1 - dt/EP_{\max})$ with default $EP_{\max}$=10 s (cumulative episode elapsed time, not a memoryless single-step Ψ). $E_{\mathrm{contract}}(2)$=0.185 $[0.165, 0.200]$. Mean env_reward: $0.846, 0.752, 0.661, 0.383, 0.106$; legal=1.0; success=0.925; raw reward $\approx$0.85 (flat).

GAIA2 CALIBRATION

Pinned ARE commit `87ebd38f31aafae0f11e14f55617903196236cfb`. Fixed mode: observed advance equals configured $S \in \{0, 2, 20, 60\}$; ChargeError(20)=0. Pinned local LiteLLM

path returns no completion-duration metadata, so measured-mode generation offset is 0 unless duration metadata is supplied (sensitivity path only; not generalized).

ANDROID PROJECT METHODS

Project temporal protocol. Scheduled emulator `gsm cancel` at t_0+D, $D \in \{8, 12, 15\}$ s, on `PhoneAnswerCall`. **Not** stock AndroidWorld. Full protocol SHA256: `7ec09236651f31bf4ee068d19728d11107b328d813fe5bbc1fa6d83612a689d8`.

Policies (exact protocol definitions).

Policy	Definition
B1 (Direct)	One VLM call: screenshot → final action → dispatch; no evidence pass; minimal critical-path computation.
B1+Delay	Generate the same B1 action; retain it unchanged; insert calibrated delay before dispatch; no extra information (isolates critical-path time).
B2-NoContext	Evidence pass, then *discard* evidence; final action without transferred evidence (extra computation/time retained).
B2 Verify	Evidence pass; final decision conditioned on evidence (extra computation + transferred evidence).

Design. 5 VLM configurations (Qwen2.5-VL-3B/7B-Instruct; Gemma 3 4B; InternVL3.5-4B/8B) $\times$ 4 policies $\times$ 3 lifetimes $\times$ 30 repeats $=$1800 episodes. Within-model interleaved randomization; model $=$ fixed block.

Latency calibration and B1+Delay constants. Saved-image calibration (20 screenshots $\times$ 3 repeats $\times$ {B1,B2} per model; seed 20260903) yields median dispatch-proxy latencies (seconds):

Model	B1 median	B2 median
Qwen2.5-VL-3B	1.333	5.406
Qwen2.5-VL-7B	1.280	4.686
Gemma 3 4B	1.581	6.141
InternVL3.5-4B	3.909	6.636
InternVL3.5-8B	3.107	7.803

B1+Delay uses $\text{delay} = \max(0, T_{\text{B2,target}} - T_{\text{elapsed}})$ with $T_{\text{B2,target}}$ pre-sampled from the model-specific empirical B2 pool (schedule medians $\approx$ 5.41, 4.69, 6.18, 6.63, 7.80 s for the five models above). Simulated B1+Delay medians match B2 medians within < 0.06 s for all models.

Per-model success matrix (replaces redundant pooled appendix table). Locked cell SRs (n=30 per cell) underlying Figure 3:

Model	Policy	D=8	D=12	D=15
Qwen2.5-VL-3B	B1	1.000	1.000	1.000
	B1+Delay	0.333	1.000	1.000
	B2 No-Context	0.200	1.000	1.000
	B2 Verify	0.300	0.933	0.933
Qwen2.5-VL-7B	B1	0.967	0.933	0.967
	B1+Delay	0.833	0.967	1.000
	B2 No-Context	0.133	0.833	0.900
	B2 Verify	0.267	0.967	0.967
InternVL3.5-4B	B1	0.933	1.000	1.000
	B1+Delay	0.000	1.000	1.000
	B2 No-Context	0.000	0.000	1.000
	B2 Verify	0.000	0.967	0.900
Gemma 3 4B	B1	0.400	0.433	0.400
	B1+Delay	0.000	0.433	0.400
	B2 No-Context	0.000	0.467	0.333
	B2 Verify	0.000	0.667	0.800
InternVL3.5-8B	B1	1.000	1.000	1.000
	B1+Delay	0.000	1.000	1.000
	B2 No-Context	0.000	0.133	1.000
	B2 Verify	0.000	0.333	1.000

Pooled cells matching Table 4 average the five models equally (150 episodes/cell).

Estimands. Let $\mathrm{SR}(P, D)$ denote success rate for policy P at lifetime D. Primary estimands (paired over model×instance units; Holm over three contrasts):

$$E1 = \big[\mathrm{SR}(B2, 15) - \mathrm{SR}(B1, 15)\big] - \big[\mathrm{SR}(B2, 8) - \mathrm{SR}(B1, 8)\big], \tag{4}$$

$$E2 = \big[\mathrm{SR}(B1{+}\mathrm{Delay}, 15) - \mathrm{SR}(B1, 15)\big] - \big[\mathrm{SR}(B1{+}\mathrm{Delay}, 8) - \mathrm{SR}(B1, 8)\big],$$

$$E3 = \mathrm{SR}(B2, 15) - \mathrm{SR}(B2\text{-NoContext}, 15). \tag{5}$$

($E2$ as written above; same contrast as Eq. (3) in the main text.) Estimates: E1=0.793, E2=0.633, E3=0.073 (Holm over $\{E1, E2, E3\}$; paired bootstrap / sign-flip).

Model-specific delay interaction (secondary). Verified from locked analysis (95% CIs): Qwen2.5-VL-3B 0.667 $[0.500, 0.833]$; Qwen2.5-VL-7B 0.167 $[0.000, 0.367]$; Gemma 3 4B 0.400 $[0.100, 0.667]$; InternVL3.5-4B 0.933 $[0.833, 1.000]$; InternVL3.5-8B 1.000 $[1.000, 1.000]$ (see Figure 3 in the main text).

Timing caveat. B1+Delay is calibrated toward two-pass latency; residual mismatch possible; the delay interaction supports sufficiency of time, not perfect matching or mediation. Hardware and emulator jitter can affect wall-clock lifetime races under short D.

REALTIMEGYM FULL-HARNESS SEMANTICS

Pinned commit `3d5b3ef37dafee25ca2e8d2e8d71f5fe842eda7b`. Official loop (`agile_eval.game_loop`): observe → think(timeout) → act → env.step. **Q1:** Freeway state does not advance during blocking think (P1; CCR=0). **Q2:** token budgets cap generation; seconds budgets truncate streams and may sleep to fill unused wall budget; empty parses fall back to `DEFAULT_ACTION` (Freeway: "U"). Classification: task-state frozen during think + decision-interval budget/fallback (Q2).

RECOMMENDED TEMPORAL CONTRACT REPORTING CHECKLIST

Practical clarity aid (*not* a mandated standard), grouped by TCS field with linked probes:

State progression (Q1). (1) Does task/environment state advance during the decision interval? (P1) (2) Is progression step-triggered, async/wall, or simulated-advance? (P1)

Decision-time accounting (Q2). (3) Does the evaluator count elapsed wall time (episode timer), independent of compute detection? (P6) (4) Are token/resource budgets, sim-time offsets, or deadlines applied during the interval? (P5–P6) (5) What occurs on timeout / default action? (P6)

Opportunity consequence (Q3). (6) Can evidence change or expire? (P3) (7) Can selected actions become stale (API-legal but wrong/unsafe)? (P4; OSC/SAR) (8) Can selected actions become impossible / invalid? (P2/P4) (9) Are deadlines visible to the agent?

Implementation / backend. (10) Are timing semantics backend-dependent? (P7)

Wall-clock charged suites should also report hardware/backend timing variance when feasible.

RELATED-WORK POSITIONING

Neighbor systems primarily advance agents, controllers, or single-suite protocols. RealtimeGym studies reasoning under time pressure and exposes both think-frozen world dynamics and token/seconds budget coupling; RRARA supplies latency-to-simulation mapping (TCM) plus a reflex/async-reflect agent on HAZARD (protocol variant, not stock); Gaia2 exposes simulated generation-time modes; AAPT targets critical-path GUI latency; OSWorld-Human profiles computer-use efficiency/latency; GameWorld-RT contrasts paused vs real-time evaluation; VSAS-Bench evaluates streaming visual assistants under real-time constraints. This paper instead compares temporal contracts across pinned harnesses.

Neighbor	Focus	Relation to this paper / TCS sketch
RealtimeGym	agents + budgets	world-freeze $\neq$ budget coupling
RRARA	TCM on HAZARD	protocol variant, not stock
Gaia2/ARE	sim generation time	T4 exemplar; measured path scoped
AAPT	critical-path GUI decode	motivates contracts
OSWorld-Human	CUA efficiency/latency	adjacent efficiency audit
GameWorld(-RT)	paused vs RT eval	adjacent: RT $\approx$ progressing Φ during think
VSAS-Bench	streaming RT eval	adjacent streaming contract; not in pin set
DynamicGUIBench	interstitial evidence	O3 contrast

GameWorld-RT and VSAS-Bench are classified here as adjacent non-pausing/streaming contracts but were outside the eleven-suite lockfile audit.