HORIZON: EVIDENCE-GROUNDED TASK MEMORY, INTERFACE TRANSLATION, AND EXPERIENCE FOR LONG-HORIZON WEB AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon web tasks require agents to connect user requirements to the effects of actions across changing interfaces. An executed action can leave the requested outcome unresolved: a click may activate the wrong control, a value may remain unsaved, or a returned answer may lack supporting observations. We introduce **HORIZON**, a framework that makes these connections explicit through evidence-grounded task state. *Task Completion Memory* tracks requirements, object bindings, progress, and supporting observations. A *Document Object Model (DOM) Translation Layer* links structured interface views and semantic operations to their source elements. *State-conditioned experience* retrieves prior task records as advice and provides a formulation for contrasting successful and unsuccessful interactions while keeping historical advice separate from current-task evidence. On WebArena, HORIZON raises gpt-5.6-luna's overall success rate from 15.4% to 40.3%, a gain of 24.9 percentage points. With Qwen3.8-27B, it achieves 60.1% across all 812 tasks, including 66.7% on cross-site tasks, under a protocol with up to two attempts on four single-site domains. These results compare complete agent configurations. Selected trajectories illustrate source-linked targeting, requirement-directed readback, and archived-route adaptation.

1 INTRODUCTION

A web agent can follow a plausible plan and execute actions without errors while leaving the user's request unfinished. Consider a request to create a private project and return its link. The agent may attempt to select Private yet leave the project public after saving. In WebXSkill's failure analysis, 38% of failed tasks are attributed to reasoning errors after successful skill execution (Wang et al., 2026b). Successful local execution alone does not establish task completion.

Verification must connect task-level outcomes to local observations. The user specifies a desired result, while each action operates on part of a changing interface. Evidence for project creation, ownership, privacy, and the returned link must concern the same object. Navigating to another page can hide the relevant state, and later changes can invalidate earlier observations. The agent must retain unresolved requirements and identify the observations needed to establish them.

Recent work addresses this gap through trajectory verification and runtime requirement tracking. Universal Verifier (Rosset et al., 2026) assesses completed trajectories against task-specific criteria. WebRider (Li et al., 2026b) maintains intent contracts and evidence obligations during browsing, separating evidence assessment from action generation. HORIZON focuses on connecting requirement status to the identity and changing state of application objects. Its shared task state links each requirement to its bound object, attempted operations, and supporting observations, so that an attempted write can remain distinct from a verified saved outcome.

We introduce **HORIZON**, which organizes planning, execution, and experience reuse around this state (Figure 1). *Task Completion Memory* (TCM) records requirements, progress, and unresolved obligations, binding each requirement to the object or collection it concerns. The *Document Object Model* (DOM) Translation Layer connects operations to live page elements through source references—links to the elements in the current observation—and returns observations about the corresponding objects. The Planner proposes operations; browser observations and criterion assess-

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.

ments update requirement status. DOM interaction is available as needed. TCM preserves progress across steps and revisits requirements whose supporting evidence becomes stale.

The same record supports *state-conditioned experience*. An earlier task can reveal a useful route or a control that requires special handling. Reuse requires checking whether the current intent, object, and control state match the earlier context. HORIZON retrieves archived task records as advice; its contrastive formulation describes how differences in observed effects can produce candidate guidance. A successful forum selection can inform a candidate even when the episode later fails to submit the post. Keeping that local observation separate from the terminal task label gives experience a precise scope; admission and later behavioral reuse require their own evidence.

We evaluate HORIZON through WebArena task outcomes and recorded execution trajectories (Zhou et al., 2024). The reported overall success rates are 40.3% with gpt-5.6-luna and 60.1% with Qwen3.8-27B. On the five single-site groups, the gpt-5.6-luna comparison gives 41.5% for HORIZON versus 15.8% for the baseline. To explain the behavior behind these results, we follow individual requirements through the selected target, browser action, observed effect, and next decision. The resulting cases connect task-state feedback to verification, source-linked execution to preserving target identity, and retrieved experience to decisions that still require fresh observations. An element-targeted execution study completes 23 of 25 tasks selected for earlier widget failures; its two remaining failures show why successful interaction alone does not establish task completion.

Our contributions are:

- **Task Completion Memory.** A requirement-level representation of progress that supports different valid action sequences. TCM connects each requirement to bound objects, execution attempts, supporting observations, and evidence validity, keeping unresolved work explicit across planning and execution.
- **DOM Translation Layer.** A bidirectional interface contract preserving target identity from task-directed operations to readback and linking interaction to requirement updates.
- **State-conditioned experience capture and reuse.** A formulation that derives candidate guidance from compatible interaction contexts and contrasting outcomes, retains source evidence and validation status, and separates reusable advice from current-task verification.

2 Related Work

2.1 Completion State and Verification

WebArena (Zhou et al., 2024) and VisualWebArena (Koh et al., 2024) evaluate final responses and application state. Universal Verifier (Rosset et al., 2026) scores completed trajectories against task-derived rubrics using screenshot evidence. Grounded Checklist Partial Credit (Qin et al., 2026) constructs checklists from task instructions, human-written rules, and official verifier tests, then judges execution logs. CM2 (Zhang et al., 2026) uses per-turn checklists to compute reinforcement-learning rewards. Harness-1 (Jiang et al., 2026) maintains document links and verification records outside its search policy; EviGraph (Chen et al., 2026a) uses an evidence graph for working memory and process rewards. HORIZON maintains requirement status during execution using user instructions and public context; its evaluated policy receives no evaluator outputs.

WebRider (Li et al., 2026b) maintains intent contracts and evidence obligations during browsing. ContractSkill (Lu et al., 2026) gives procedural skills preconditions, postconditions, and repairable steps. TCM tracks requirements across alternative action sequences, including changes to application records. Its evidence needs enter subsequent planning: an unresolved privacy requirement can prompt a settings-page readback after project creation. Our trajectory analysis examines this feedback alongside the independent task outcome. Evaluated readiness remains diagnostic.

2.2 Interface Representation and Source-Linked Interaction

ReAct (Yao et al., 2022), Agent-E (Abuelsaad et al., 2024), and Agent S2 (Agashe et al., 2025b) structure reasoning, planning, and grounding for web actions. In AgentOccam's cumulative GPT-4-Turbo ablation (Yang et al., 2025), observation simplification raises WebArena success from 31.7%

to 37.1% after earlier action-space changes. BrowserGym (Chezelles et al., 2025) uses shared element identifiers across observations and actions. WebChallenger's PageMem (Hwang et al., 2026) retains selectors and values in hierarchical sections, while compound workflows handle intermediate page changes. OpAgent (Guo et al., 2026) separates action grounding from visual checks of effects. DOMSteer (Hao et al., 2026) adapts interfaces for human users. These systems show that interface representation and feedback matter.

Wuying-Browser-Agent (Chen et al., 2026b) combines schema-validated operations with structured feedback and updated observations. WALT (Prabhu et al., 2026) validates reusable website tools and delegates execution to an agent when scripts fail. ContractSkill (Lu et al., 2026) re-resolves selectors on the latest page snapshot after every action. HORIZON connects these execution concerns to persistent task state: source references link task-facing objects, grounded actions, and refreshed observations. Invalidated references require target reacquisition. DOM interaction is available on demand, and readback updates evidence for the corresponding requirement.

2.3 Experience Capture and Reuse

Reflexion (Shinn et al., 2023) retains verbal feedback to guide subsequent trials. Agent S (Agashe et al., 2025a) and A-MEM (Xu et al., 2025) support reuse through experience and linked memory. Agent Workflow Memory (Wang et al., 2024) reports 35.5% WebArena success against a 23.5% BrowserGym baseline. SkillWeaver (Zheng et al., 2025) nevertheless identifies four of nine car-search tasks where successful skills were followed by failure in the final steps. These findings motivate reusable procedures coupled to remaining task obligations. State-Grounded Dynamic Retrieval (Li et al., 2026a) validates sub-procedures by re-execution and retrieves them using the task goal and page state; AutoGuide (Fu et al., 2024) contrasts trajectories to derive conditional guidance. AutoManual (Chen et al., 2024) retains rules, examples, and validation logs. In the 154-task WebArena study of WebXSkill (Wang et al., 2026b) with GPT-5, removing skill verification lowers success from 69.5% to 55.2%, reinforcing the importance of checking reusable procedures.

MemoRepair (Zhao et al., 2026) withdraws dependent memory after source invalidation and republishes validated replacements. HORIZON likewise permits withdrawal of contradicted advice. Candidate guidance retains its task, object, and control contexts, outcome differences, provenance, and validation status. The formulation allows verified local progress to support a candidate even when the overall task fails. The evaluated reuse pathway retrieves archived task records; transfer from validated contrastive tips remains unmeasured. Retrieved advice guides later decisions, while fresh evidence establishes current-task requirements.

3 Method

3.1 Framework Overview

Given a user instruction u and allowed public context c, a web agent receives browser observations o_t, performs operations, and eventually returns an answer. Task completion requires the requested outcomes and compliance with any user-specified process constraints. HORIZON represents these requirements in a completion specification while leaving the agent to choose an execution path. Task Completion Memory (TCM), denoted M_t at step t, links requirements to relevant objects, values, and collections. It records supporting evidence and unresolved work throughout execution.

The execution loop separates four roles: requirements define what must hold, the Planner selects actions, current-task evidence informs requirement assessment, and historical experience suggests strategies. The Planner uses the TCM and current interface state to choose an operation or request information. The Executor performs local interactions, including DOM-backed operations, and feeds resulting observations into evidence updates. These updates maintain requirement status and return unresolved obligations to the Planner. The evidence contract requires valid, applicable support for satisfaction; invalidation or specification changes may reopen a requirement. The evaluated implementation can misinterpret or fail to admit evidence, so its assessments remain distinct from benchmark success (Section 4.4). Section 4.1 describes the evaluated configurations.

The loop uses trajectory evidence to choose between acting and verifying. An unresolved requirement can call for an application change; an attempted change without a confirmed effect calls for

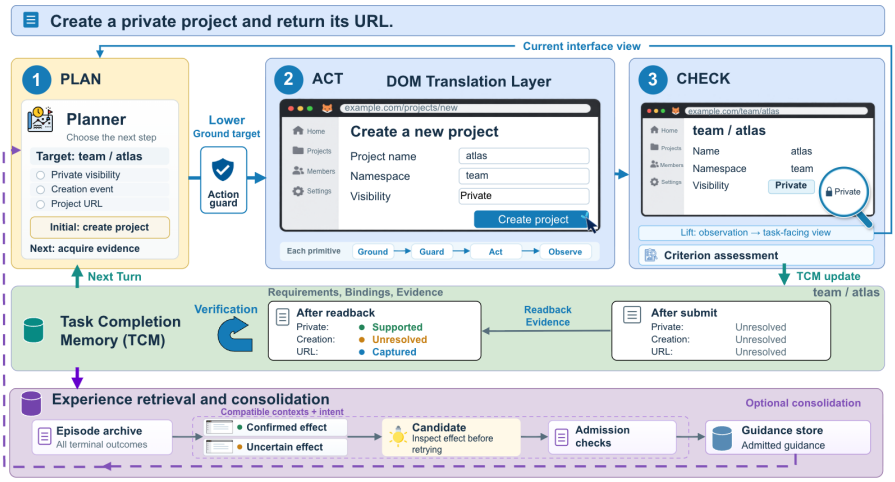

Figure 1: HORIZON's requirement–action–evidence loop, illustrated by private-project creation. TCM tracks unresolved requirements; the DOM layer grounds actions in interface elements; browser readback provides evidence of project identity, privacy, and URL. A nonblocking audit assesses readiness to answer. The lower path outlines experience capture, checking, and reuse; Section 4.4 distinguishes exercised archive retrieval from proposed contrastive consolidation.

readback; a wrong local effect calls for repair or target reacquisition. The next Planner request includes these outstanding needs together with the current interface. A request to create a private project can therefore progress from template selection and creation to inspection of saved visibility. Section 4.4 follows this sequence and compares assessment with observed state.

3.2 Task Completion Memory and Requirement-Level Assessment

TCM evaluates the evolving trajectory against the user request and keeps unfinished requirements available to planning. Its state is $M_t = (S_t, B_t, Z_t, E_t, O_t)$: the completion specification, object and collection bindings, requirement states, evidence with validity information, and outstanding obligations. A binding associates a requirement with a particular object or collection, such as the newly created project or a filtered review table. Compilation and coverage auditing link material instructions to criteria, constraints, conditional dependencies, reference slots, and output requirements, while recording unresolved interpretations.

Algorithm 1 shows how TCM guides the next decision. Initialization supplies the specification and first browser observation to TCM and constructs the current interface A_t. The Planner reads the obligations and this interface; an action guard checks authorization, target freshness, and retry safety, requiring confirmation before retrying an uncertain write. UPDATE applies activation changes and versioned specification revisions linked to the instruction, updates bindings, and invalidates stale support. It records each observation's source, scope, freshness, and derivation type, reassesses active requirements, and turns unresolved requirements into actionable obligations. The interface is refreshed from the resulting observation before the next decision. After a private-project form is submitted, a remaining visibility requirement directs the Planner to read back the saved object.

The assessment contract requires valid current-task evidence for the bound object or collection. Writes require saved-state readback. An exact count requires either a valid source-declared total for the requested scope or complete, deduplicated enumeration with the filters preserved. Partial enumeration without a valid total supports only a lower bound. Structured checks are preferred; otherwise, a restricted semantic or visual assessor evaluates one criterion and returns satisfaction, violation, or abstention. Insufficient evidence leaves the requirement unresolved. Activation, satisfaction, and evidence validity remain distinct: stale support calls for verification, not a violation judgment. Temporal constraints persist across steps; restoring a field cannot establish that it remained unchanged. Under this contract, Planner claims, action-success reports, and historical advice are not sufficient evidence of satisfaction.

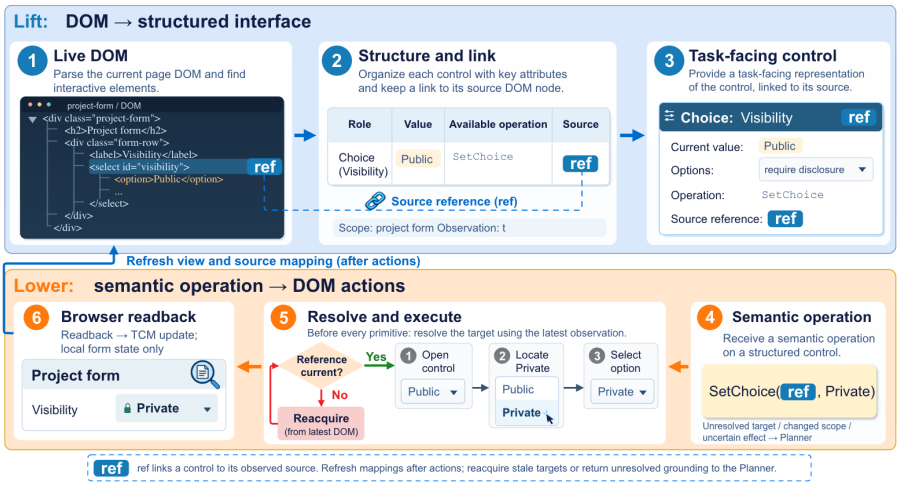

Figure 2: DOM translation for an illustrative visibility choice. Lifting (1–3) links the live control, structured attributes, and task-facing view through a source reference. Lowering (4–6) resolves SETCHOICE(ref, Private) against the current DOM, opens the control, selects the option, and returns readback to TCM. Actions refresh source mappings; stale targets require reacquisition. Readback confirms a form selection; saved privacy requires a created-object check.

When the Planner proposes an answer at step T, $\textsc{Ready}(M_T)$ checks criteria, bindings, scopes, and constraints; $\textsc{Ready}(M_T, \hat{y})$ also checks the proposed answer against independently acquired evidence. Algorithm 1 separates answer finalization from budget termination; runtime failures and runs without a readiness claim also receive distinct endings. Both readiness checks are nonblocking audits in the evaluated configuration; WebArena independently determines task success. Binding and interpretation errors can miss readback or produce incorrect assessments (Section 4.4).

Algorithm 1 Task Completion Memory: the requirement–action–evidence loop.

Require: instruction u, public context c, episode budget
$S \leftarrow \textsc{Compile}(u, c)$; $\textsc{CoverageAudit}(u, S)$
$o \leftarrow \textsc{Observe}()$; $A \leftarrow \textsc{Interface}(o)$; $M \leftarrow \textsc{Initialize}(S, o)$
while episode budget remains **do**
  $q \leftarrow \textsc{Planner}(\textsc{Obligations}(M), M, A)$
  if q proposes answer $\hat{y}$ **then**
    return $\hat{y}$, $\textsc{Ready}(M)$, $\textsc{Ready}(M, \hat{y})$ ▷ nonblocking audits
  end if
  if $\neg\textsc{ActionGuard}(q, M)$ **then**
    $M \leftarrow \textsc{AddObligation}(M, \textsc{MissingPrecondition}(q, M))$
    continue
  end if
  $(r, o) \leftarrow \textsc{Execute}(q)$ ▷ result and latest observation
  $e \leftarrow \textsc{Interpret}(r, o, M)$ ▷ execution and browser evidence
  $M \leftarrow \textsc{Update}(M, e)$
  $A \leftarrow \textsc{Interface}(o)$ ▷ refresh grounding for the next decision
end while
record budget termination without a readiness claim
Runtime failures receive a distinct termination record.

3.3 DOM Translation Layer: Connecting Intent, Action, and Readback

The DOM Translation Layer preserves control identity and scope across planning, execution, and readback. A requested visibility change must act on the intended control and obtain readback from the same form. Figure 2 shows the two directions: *lifting* exposes structured controls linked to the live DOM, and *lowering* translates operations on those controls into browser actions.

Lifting makes a control's role, value, available operations, and source reference explicit. A source reference identifies the originating element and observation version, allowing an operation or read-

back to be traced to that element. For observation o_t with source generation g_t, it constructs

$$\mathcal{L}(o_t, g_t) = (G_t, A_t, \Pi_t, C_t), \tag{1}$$

where G_t is the semantic *PageGraph*, A_t the role-specific `AgentInterface`, Π_t the mapping to source elements, and C_t the record of coverage and omissions. The source links retain form or row context when names or coordinates are ambiguous. A choice also indicates whether options are visible or require opening or search. Collection views preserve filters, scope, and supported record identities, and expose how to acquire remaining records.

Lowering turns a semantic operation $q = (v, r, \theta)$—verb, control reference, and arguments—into browser primitives. Before each primitive, the Executor grounds the target in the latest observation:

$$a_j = \mathcal{D}(q, o_j, \Pi_j), \qquad o_{j+1} = \mathrm{BrowserStep}(a_j), \qquad \mathrm{gen}(\Pi_j(r)) = \mathrm{gen}(o_j). \tag{2}$$

The generation check requires reacquisition when a page update invalidates a source mapping, even if the logical object remains unchanged. Each primitive passes the action guard. New choices, changed scope, unresolved targets, or uncertain effects return control to the Planner.

For SETCHOICE(ref, Private), the Executor opens the visibility control, locates Private, selects it, and refreshes the observation. Readback returns the selected value and its source context to TCM. This shared reference connects what was requested, which element was operated, and what changed. The resulting evidence concerns the form: establishing saved project privacy still requires submission and readback from the created object. Bindings identify the relevant project.

Unsupported controls fall back to native interaction or scoped vision within the adapter's capabilities. In the benchmark evaluations, low-level DOM interaction is available and invoked adaptively. Section 4.3 evaluates the additional source-linked DOM Translation Layer in a targeted ablation; the ablated configuration retains point-based DOM fallback, while the full configuration uses element-targeted execution. Appendix C gives the detailed configuration differences, and Appendix D specifies the interface fields and access levels.

3.4 STATE-CONDITIONED EXPERIENCE RETRIEVAL AND CONSOLIDATION

Experience reuses what earlier interaction revealed while preserving a fresh TCM for every task. The evaluated pathway retrieves archived task records. We additionally formulate contrastive candidate generation and admission below; their validated transfer is not established by the reported experiments. TCM snapshots from terminated episodes enter a per-website archive H_w, retaining their outcome and unresolved requirements. A separate store K holds site knowledge and is intended to admit validated guidance about controls, parameterized routes, and alternative operations. Retrieval is conditioned on the task, site, page and control state, bindings, and delegated intent. Applicability checks select role-relevant excerpts with source and version information. Historical entries guide exploration and execution; they cannot satisfy a current-task criterion.

Episode records and deterministic summaries supply the inputs for optional consolidation. In the proposed candidate pathway, lessons connect a triggering context and recommendation to source events and observed effects. A failed task can contribute a verified route or observed control effect while retaining its unsuccessful task outcome. Consolidation receipts record whether a candidate was produced, rejected, or left unsupported.

Contrastive capture compares trajectories with compatible pre-action contexts and intent. For successful τ^+ and failed or uncertain τ^-, let $c(\tau)$ encode site, task type, control state, bindings, and intent, and let $y(\tau)$ encode the observed effect and verified outcome:

$$C = \mathrm{Match}\big(c(\tau^+), c(\tau^-)\big), \qquad \Delta_y = \mathrm{Diff}\big(y(\tau^+), y(\tau^-)\big). \tag{3}$$

The shared context C defines applicability, while Δ_y records the outcome difference used to formulate candidate guidance. A tip $k = (C, a, \Delta_y, r, p)$ stores the triggering action a, recommended operation r, and provenance and validation status p. This representation preserves the evidence behind the recommendation for subsequent checking. For example, a forum-selection candidate would retain the unchanged placeholder after an earlier click, the later search-and-select operation, and readback showing the requested forum. Its scope is control selection; whether a post was subsequently submitted is a separate outcome. This distinction permits local successes in failed episodes to inform candidates without labeling the entire episode successful.

Table 1: Task success with the same gpt-5.6-luna backbone, without and with HORIZON. Rates are percentages; gains are percentage points from unrounded rates. Unsuccessful runs count as failures.

	Without HORIZON		With HORIZON		
Site	Pass / N	Success (%)	Pass / N	Success (%)	Δ (pp)
Shopping	33/187	17.6	60/187	32.1	+14.4
Shopping Admin	20/182	11.0	52/182	28.6	+17.6
GitLab	21/180	11.7	100/180	55.6	+43.9
Reddit	24/106	22.6	63/106	59.4	+36.8
Map	23/109	21.1	42/109	38.5	+17.4
Cross-site	4/48	8.3	10/48	20.8	+12.5
Single-site subtotal	121/764	15.8	317/764	41.5	+25.7
Overall	125/812	15.4	327/812	**40.3**	**+24.9**

The proposed admission contract checks scope, provenance, conflicts, sensitive content, and task-answer leakage, with stronger validation for broader recommendations. Replay consistency and independent behavioral support have separate statuses. Contradicted advice can be quarantined or withdrawn from the store, index, and rendered context. Retrieval logs distinguish finding an entry, exposing it to the Planner or Executor, consulting it, and acting on it. Section 4.4 distinguishes observed archive retrieval from proposed contrastive admission and transfer.

4 Experiments

We evaluate complete-system performance, then inspect how DOM grounding, TCM assessment, and archived experience affect recorded execution.

4.1 Evaluation Setup and Trajectory Analysis

We use WebArena's independent evaluator to check final responses and application state (Zhou et al., 2024). HORIZON uses the listed backbone for both Planner and Executor, with no visual-model calls. Its A3-Qwen3.5-9B evaluation uses one attempt on each of 812 tasks. Qwen3.8-27B (Qwen Team, 2026) allows 30 browser steps and two hours per attempt, with up to two attempts in Shopping, Admin, GitLab, and Reddit. The policy receives no evaluator feedback during an attempt; readiness assessment is nonblocking. Appendix A gives model settings and the aggregation protocol.

The baseline runs without HORIZON. The TCM reuse comparison retains current-task TCM and DOM execution in both arms, changing retrieval of prior records and site knowledge. The widget study compares two execution configurations without TCM reuse on 25 tasks selected for prior widget failures. For selected trajectories, we align requirements, targets, browser effects, internal assessments, and subsequent decisions. Cases explain mechanisms, without estimating their population frequency; Appendix B summarizes the selected trajectory evidence.

4.2 Overall Task Performance

With gpt-5.6-luna (OpenAI, 2026), HORIZON raises overall success from 15.4% to 40.3%, a gain of 24.9 percentage points (Table 1). Gains span every site group and are largest on GitLab and Reddit. Table 2 places these results alongside ColorBrowserAgent (Wang et al., 2026a), OpAgent (Guo et al., 2026), A3 (Lù & Reddy, 2026), and our WebChallenger reproduction (Hwang et al., 2026). HORIZON with Qwen3.8-27B reaches 60.1% overall and 66.7% cross-site, exceeding A3 and reproduced WebChallenger on the displayed single-site groups. Stronger published systems lead on several sites. These comparisons differ in backbones, training, coverage, and attempt budgets. In particular, HORIZON-A3's 20.7% over 812 tasks and published A3's 41.5% over 381 test tasks do not establish a matched harness effect.

Table 2: WebArena task success (%, ↑). All HORIZON rows include TCM. Bold marks the largest displayed rate within each block. †Cross-site results were not reported in the original paper.

Agent / backbone	Shop	Admin	GitLab	Reddit	Map	Cross-site	Total
Published results							
ColorBrowserAgent, GPT-5	**72.9**	**76.4**	65.7	**87.4**	55.9	**64.8**	71.2
OpAgent, Gemini 3 Pro + Qwen2.5VL-MFT†	59.2	71.3	**75.9**	86.0	**71.4**	–	**71.6**
Agent-as-Annotators, A3-Qwen3.5-9B†,[a]	33.3	44.9	49.5	56.2	25.0	–	41.5
Experiment results							
gpt-5.6-luna	17.6	11.0	11.7	22.6	21.1	8.3	15.4
Qwen3.8-27B	34.2	23.6	6.7	58.5	22.9	10.4	26.0
WebChallenger, GLM4-32B + Qwen2.5-VL-7B	45.5	26.6	44.7	55.7	45.0	29.2	41.4
HORIZON, A3-Qwen3.5-9B	24.6	19.8	19.4	22.6	19.3	12.5	20.7
HORIZON, gpt-5.6-luna	32.1	28.6	55.6	59.4	38.5	20.8	40.3
HORIZON, Qwen3.8-27B	**55.1**	**48.4**	**72.8**	**78.3**	**46.8**	**66.7**	**60.1**

[a]Published A3 uses BrowserGym's 381-task WebArena test split (Lù & Reddy, 2026). All other rows report results on full WebArena (812 tasks).

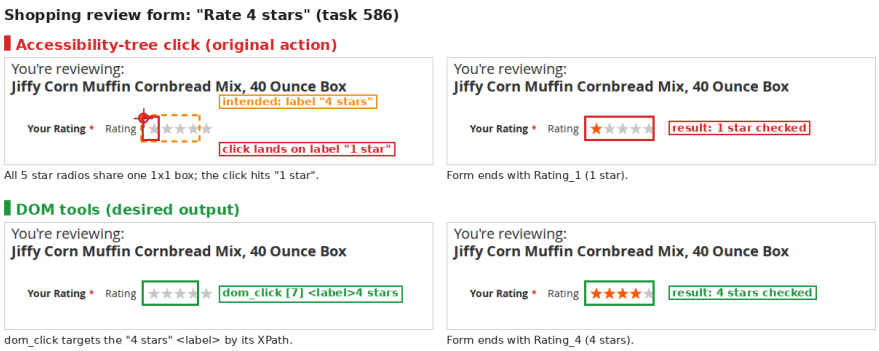

Figure 3: Preserving rating identity. Top: shared coordinates cause the requested four-star action to activate one star. Bottom: targeting the four-star DOM label produces the intended checked state. These are controlled live reproductions without submission; the scored task-586 trajectory separately confirms the selection and completed review.

4.3 DOM Execution Preserves Target Identity and Scope

To examine the DOM Translation Layer, we conduct a targeted ablation on 25 tasks that fail when HORIZON runs without the translation layer. This ablated configuration retains the adapter's point-based DOM fallback, while the full configuration enables element-targeted DOM execution with source-linked control identity. We rerun the same 25 tasks with the translation layer enabled, retaining current-task TCM and disabling experience reuse. The selected failures concentrate on ratings, template selection, and forum selection, where the intended control is identifiable but difficult to activate reliably through point-based interaction.

Choosing the correct option does not ensure that execution activates it. On Shopping, for example, the five rating radios share a 1×1-pixel box: reproducing the point-based four-star action activates one star, whereas targeting the four-star label selects the intended radio (Figure 3). In scored task 586, DOM readback confirms that four stars are checked and the other ratings are unchecked, after which the submitted review passes. Preserving source identity therefore allows execution to retain the requested choice even when multiple controls are geometrically indistinguishable.

The same issue appears at the level of row and container scope. Among GitLab's 30 `Use template` radios, task 755 targets the Plain HTML row and confirms Pages/Plain HTML, avoiding the earlier Pages/Gatsby selection. In task 600, DOM-backed search operates within Reddit's inde-

pendently scrolling forum picker, exposes `consoles`, and preserves that forum through selection, readback, and publication. Appendix C provides the configuration details and aggregate outcomes.

Across the 25 selected failures, enabling the DOM Translation Layer recovers all four rating tasks, all four template-selection tasks, and 15 of 17 forum-selection tasks. Nineteen successful reruns operate the target widget directly, while four forum tasks use a forum-specific URL exposed through the DOM-backed execution path. Table 3 shows representative reductions in repeated target attempts.

Table 3: Selected-task effort in the DOM Translation Layer ablation: without translation layer → element-targeted DOM execution.

Task	Clicks	Steps
Rating (586)	$15 \rightarrow 1$	$23 \rightarrow 12$
Template (755)	$8 \rightarrow 1$	$22 \rightarrow 10$
Forum (600)	$15 \rightarrow 1$	$30 \rightarrow 7$

The study is restricted to tasks selected after failure under the ablated configuration and therefore does not estimate a benchmark-wide DOM-layer effect. The intervention also changes element targeting, routing, click handling, and Executor context together. It provides targeted evidence that the translation layer's source-preserving execution can recover failures in which the intended action is known but point-based execution does not reliably realize it.

4.4 TCM Assessment and Archived Experience in Execution

TCM exposes both accepted evidence and remaining work. In task 345, the agent filters Shopping Admin reviews to April 1–30, 2023. The grid declares 351 records while displaying 20 rows. At finalization, TCM checks the answer against that scoped source total, marks the count and answer-format criteria satisfied, and clears the obligation list; the official evaluator passes. Acceptance uses the filtered total, not a count of visible rows, and occurs during finalization rather than immediately after clicking Search.

In task 755, creating the private Plain HTML project leaves persistent readback outstanding. The Planner opens the project and its settings, observes Private selected, and passes the official task. The requirement therefore guides verification beyond the form actions, although TCM's final criteria remain unresolved because the readback is not admitted as operation-linked evidence. Conversely, task 476 accepts project existence while its name is still in the creation form. These cases distinguish useful feedback from assessment accuracy; neither they nor the reuse comparison isolate current-task TCM's causal effect. Appendix B summarizes the corresponding trajectory evidence and counterexample.

In task 802, archive retrieval exposes the members-page pattern `/groups/<group>/-/group_members` after unsuccessful navigation attempts. The next plan substitutes the new group `coding_friends`; the agent reaches its members page, invites the two requested users, and reads back both accounts. The task passes in 16 logged steps. This sequence demonstrates route adaptation followed by current-task evidence. Appendix B summarizes the corresponding retrieval trajectory.

5 Conclusion

HORIZON connects user requirements to executable operations and observed effects through persistent task state. TCM exposes unresolved requirements and remaining verification work, source-linked DOM interaction preserves control identity during execution, and state-conditioned experience keeps prior-task guidance separate from current-task evidence. On WebArena, HORIZON reaches 40.3% with gpt-5.6-luna and 60.1% with Qwen3.8-27B. Recorded trajectories illustrate these mechanisms in use: the agent verifies saved privacy settings after project creation, resolves ambiguous controls through exact targeting, and adapts retrieved routes to new objects before checking their effects. The same analysis also reveals cases where locally successful interactions leave writes unfinished or requirements insufficiently verified; the examined retrieval cases provide qualitative evidence of reuse but do not establish its aggregate benefit. By making the requirement–action–evidence chain explicit, HORIZON exposes both what has been accomplished and what still prevents task completion.

AI USE STATEMENT

We used generative AI tools to polish the manuscript's language, clarity, and organization, and to assist with reviewing and summarizing recorded trajectory evidence. The authors are responsible for verifying the AI-assisted material and for the accuracy of the final text, analyses, and claims.

REPRODUCIBILITY STATEMENT

Evaluation protocols, attempt budgets, and model settings appear in Appendix A; trajectory analysis and widget-study configurations appear in Appendices B and C. Appendix E inventories the supporting records, including task ledgers and trajectory excerpts with source line references and file hashes. Figures 1 and 2 are illustrative; the widget figures are controlled reproductions.

REFERENCES

Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. Agent-e: From autonomous web navigation to foundational design principles in agentic systems, 2024. URL https://arxiv.org/abs/2407.13032.

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Wang. Agent s: An open agentic framework that uses computers like a human. In *International Conference on Learning Representations*, volume 2025, pp. 22924–22946, 2025a.

Saaket Agashe, Kyle Wong, Vincent Tu, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s2: A compositional generalist-specialist framework for computer use agents. *arXiv preprint arXiv:2504.00906*, 2025b.

Jiashun Chen, Yirong Mao, and Wenhui Que. Evigraph: Towards verifiable evidence construction for information-seeking agents. *arXiv preprint arXiv:2608.24667*, 2026a.

Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37:589–631, 2024.

Tianxiang Chen, Yan Cheng, Zhangye Han, Xiaowei Li, Chang Liu, Cheng Liu, Zhongqiang Ma, Long Peng, Xiaobing Tu, Yinggui Wang, et al. Wuying-browser-agent: Real-world centric fundamental long-horizon browser agents. *arXiv preprint arXiv:2608.17319*, 2026b.

Thibault Le Sellier De Chezelles, Maxime Gasse, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Quentin Cappart, Graham Neubig, Ruslan Salakhutdinov, Nicolas Chapados, and Alexandre Lacoste. The browsergym ecosystem for web agent research, 2025. URL https://arxiv.org/abs/2412.05467.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *Advances in Neural Information Processing Systems*, 37: 119919–119948, 2024.

Yuyu Guo, Wenjie Yang, Siyuan Yang, Ziyang Liu, Cheng Chen, Yuan Wei, Yun Hu, Yang Huang, Guoliang Hao, Dongsheng Yuan, et al. Opagent: Operator agent for web navigation. *arXiv preprint arXiv:2602.13559*, 2026.

Pan Hao, Rishi Selvakumaran, Jacob Sun, and Qianwen Wang. Beyond chat and clicks: Gui agents for in-situ assistance via live interface transformation. *arXiv preprint arXiv:2604.14668*, 2026.

Jayoo Hwang, Xiaowen Zhang, and Vedant Padwal. Webchallenger: A reliable and efficient generalist web agent. *arXiv preprint arXiv:2606.10423*, 2026.

Pengcheng Jiang, Zhiyi Shi, Kelly Hong, Xueqiang Xu, Jiashuo Sun, Jimeng Sun, Hammad Bashir, and Jiawei Han. Harness-1: Reinforcement learning for search agents with state-externalizing harnesses. *arXiv preprint arXiv:2606.02373*, 2026.

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

Jiaxi Li, Ke Deng, Yun Wang, Jingyuan Huang, Yucheng Shi, Qiaoyu Tan, Jin Lu, and Ninghao Liu. Online skill learning for web agents via state-grounded dynamic retrieval. *arXiv preprint arXiv:2606.04391*, 2026a.

Zhi Li, Tao Zhou, Yeqing Li, Eugene Ie, and Demetri Terzopoulos. Webrider: Persona-conditioned intent controllers for live-web assistance. *arXiv preprint arXiv:2608.06704*, 2026b.

Xing Han Lù and Siva Reddy. Structured distillation of web agent capabilities enables generalization. *arXiv preprint arXiv:2604.07776*, 2026.

Zijian Lu, Yiping Zuo, Yupeng Nie, Xin He, Weibei Fan, Lianyong Qi, and Shi Jin. Contractskill: Repairable contract-based skills for multimodal web agents. *arXiv preprint arXiv:2603.20340*, 2026.

OpenAI. GPT-5.6 System Card. OpenAI Deployment Safety Hub, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6. Published July 9, 2026; accessed September 25, 2026.

Viraj Prabhu, Yutong Dai, Matthew Fernandez, Krithika Ramakrishnan, Jing Gu, Yanqi Luo, silvio savarese, Caiming Xiong, Junnan Li, Zeyuan Chen, and Ran Xu. Walt: Web agents that learn tools. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 55589–55607, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5b175f9e93873e3a10a6ce43dbb82e05-Paper-Conference.pdf.

Suliu Qin, Lu Yin, and Xilu Wang. Grounded Checklist Partial Credit for Agent Skill Trajectories. arXiv:2608.27487, 2026. URL https://arxiv.org/abs/2608.27487.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Corby Rosset, Pratyusha Sharma, Andrew Zhao, Miguel Gonzalez-Fernandez, and Ahmed Awadallah. The art of building verifiers for computer use agents. *arXiv preprint arXiv:2604.06240*, 2026.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Jihong Wang, Jiamu Zhou, Weiming Zhang, Teng Wang, Weiwen Liu, Zhuosheng Zhang, Xingyu Lou, Weinan Zhang, Huarong Deng, and Jun Wang. ColorBrowserAgent: Complex long-horizon browser agent with adaptive knowledge evolution. In Yunyao Li, Georg Rehm, and Mei Tu (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 665–680, San Diego, California, USA, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-394-4. doi: 10.18653/v1/2026.acl-industry.46. URL https://aclanthology.org/2026.acl-industry.46/.

Zhaoyang Wang, Qianhui Wu, Xuchao Zhang, Chaoyun Zhang, Wenlin Yao, Fazle Elahi Faisal, Baolin Peng, Si Qin, Suman Nath, Qingwei Lin, et al. Webxskill: Skill learning for autonomous web agents. *arXiv preprint arXiv:2604.13318*, 2026b.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik A Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for llm-based web agents. In *International Conference on Learning Representations*, volume 2025, pp. 97533–97565, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Zhen Zhang, Kaiqiang Song, Xun Wang, Yebowen Hu, Weixiang Yan, Chenyang Zhao, Henry Peng Zou, Haoyun Deng, Sathish Reddy Indurthi, Shujian Liu, et al. Cm2: Reinforcement learning with checklist rewards for multi-turn and multi-step agentic tool use. *arXiv preprint arXiv:2602.12268*, 2026.

Yang Zhao, Chengxiao Dai, Mengying Kou, and Yue Xiu. Memorepair: Barrier-first cascade repair in agentic memory. *arXiv preprint arXiv:2605.07242*, 2026.

Boyuan Zheng, Michael Y Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

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

A Evaluation Details and Implementation Settings

A.1 Evaluation Protocol

WebArena's independent evaluator determines task success. The Qwen3.8-27B Planner uses temperature 1.0 and medium reasoning; the Executor uses temperature zero with thinking disabled. Each attempt permits 30 browser steps and two hours. Fuzzy-match and infeasibility evaluation use a separate gpt-5.6-luna judge with five-vote majority scoring. Unscored HORIZON runs are counted as failures.

The detailed Qwen records cover Shopping, Shopping Admin, GitLab, and Reddit; Map aggregates were supplied separately. The reported Qwen3.8-27B result allows up to two attempts per task on these four sites, with a second attempt only after an unsuccessful first attempt. Each task counts once if either attempt passes; Map is not retried. The separate widget study does not contribute successes to this aggregate. Evaluator outcomes select tasks for retry, but the policy receives no evaluator feedback during an attempt.

The second attempt retains the Qwen backbone, TCM, cross-task retrieval, and per-attempt budget, while using the element-targeted execution configuration. Executor context and action routing differ from the initial attempt, and the retry replicas correct the GitLab SSH hostname; the reported two-attempt aggregate reflects this joint protocol rather than an isolated execution change.

The gpt-5.6-luna baseline and HORIZON with gpt-5.6-luna use the same backbone and task groups, but available records do not establish fully matched prompts, budgets, provider routing, and evaluator settings; their difference therefore concerns the complete configurations. The reproduced WebChallenger results are likewise not a task-level matched comparison with HORIZON. Published A3 uses BrowserGym's 381-task WebArena test split, whereas the HORIZON-A3 row evaluates all 812 tasks.

A.2 TCM Reuse Comparison

Current-task TCM remains active in both reuse arms; the intervention changes cross-task retrieval of archived TCM records and site knowledge. Across the four sites with task-level pairing, 273 tasks pass in both arms, 50 pass only with reuse, 57 pass only without reuse, and 275 pass in neither arm. No site-level difference reaches the 0.05 level under a two-sided exact McNemar test. Map contributes 51/109 with reuse and 53/109 without reuse, without task-level pairing.

In the audited subset of initial reuse runs, archive retrieval functioned but learned tips were not activated. These outcomes support the distinction in Section 4.4 between observed archive reuse and validated contrastive-tip transfer; they do not establish an aggregate reuse gain.

A.3 Qwen3.8-27B Configuration

Table 4: Qwen3.8-27B settings for the recorded four-site evaluation.

Component	Model and decoding	Context and runtime setting
Planner	Qwen3.8-27B; $T = 1.0$; medium reasoning	32,768 input tokens; 16,384 output tokens; four tool rounds
Executor, initial	Qwen3.8-27B; $T = 0$; thinking disabled	16,384 input tokens; 16,000-character page view; one tool round
Executor, second	Same model and decoding	32,768 input tokens; 64,000-character page view; DOM tools; four tool rounds
Task memory	Current-task TCM	Cross-task retrieval varies in the initial comparison; enabled in the reported two-attempt aggregate
Attempt budget	Same backbone in both roles	30 browser steps and 7,200 seconds per attempt
Evaluation	Separate gpt-5.6-luna judge	Five-vote fuzzy-match / infeasibility scoring; readiness audit nonblocking
Browser	No visual-model calls	Accessibility tree and URL; DOM interaction available

B Recorded Trajectory Analysis

We inspect selected trajectories to explain the mechanisms discussed in Section 4.4. Cases are selected purposively for observable requirement–action–evidence transitions and do not estimate population frequencies. For each case, we distinguish the requested condition, browser observation, subsequent decision, internal assessment, and independent WebArena outcome.

Scoped evidence. In Shopping Admin task 345, the requested date filter produces a grid declaring 351 records while displaying only 20 rows. At finalization, TCM accepts the scoped source-declared total, verifies the proposed answer against it, and the official evaluator passes.

Requirement-directed verification. In task 755, the agent creates the requested Plain HTML project and then inspects the created object's settings because privacy remains unresolved. The page shows `Private` selected and the official evaluator passes, although the internal assessment does not fully admit the saved-state observation. Task 476 provides the complementary counterexample: project-existence evidence is accepted while the requested name is still in the creation form, before submission. Its eventual official pass does not validate that earlier judgment.

Interaction-count definitions. Table 3 compares the ablated and full execution configurations without cross-task TCM reuse. Click counts refer to attempts on the requested control; browser steps cover the full run, including navigation, form interaction, and submission. The differences are descriptive measurements for selected tasks, not an isolated estimate of targeting efficiency.

Archive retrieval. In task 802, retrieval exposes the reusable GitLab members-page pattern `/groups/<group>/-/group_members`. The Planner substitutes the current group name, reaches the corresponding page, invites both requested users, and obtains fresh browser evidence that they were added. The task passes. The retrieved record supplies navigation guidance rather than evidence for the current task. It supplies a concrete example of the archive pathway whose aggregate outcomes are reported in Appendix A.2.

C DOM Interaction and Widget Targeting

This appendix details the targeted ablation in Section 4.3 and the source-linked execution mechanism described in Section 3.3. The ablated configuration retains low-level DOM interaction: when point-based interaction leaves the observed page unchanged, the adapter can dispatch a DOM click at the target location. The full configuration enables element-targeted execution, exposing numbered DOM elements and resolving them to specific source elements through XPath targets. Both configurations therefore provide DOM access; the ablation examines recovery of selected widget failures when execution preserves element identity and control scope.

Ablation setup. We select 25 WebArena tasks that fail under the ablated Qwen3.8-27B configuration at Shopping star ratings, GitLab template selection, or the Reddit forum picker, and rerun those tasks under the full configuration. Both configurations retain current-task TCM, disable cross-task reuse, use the same evaluator, and allow 30 browser steps and two hours per task. Enabling element-targeted execution also changes Executor routing and context: direct Planner field assignment is delegated to the Executor, numbered DOM elements replace accessibility-tree targeting for element actions, and the Executor receives the larger context documented in Table 4. Click handling permits element-level JavaScript dispatch when pointer interaction fails. The ablation evaluates these changes jointly, so its results do not isolate the effect of element targeting alone.

Outcomes. The full configuration passes 23 of the 25 selected tasks: all four rating tasks, all four template-selection tasks, and 15 of 17 forum-selection tasks. Nineteen successful reruns operate the problematic widget directly; four forum tasks use an alternate URL route that bypasses the picker. The two remaining failures occur after successful widget interaction. These results demonstrate recovery on the selected failures; they do not estimate a benchmark-wide improvement or measure regressions on tasks that previously succeeded.

Observed mechanisms. The rating reproduction in Figure 3 illustrates how shared geometry can lose control identity: a point-based action activates the wrong rating, whereas targeting the requested label produces the intended checked state. Figures 4 and 5 examine two related problems: repeated control names across rows and options inside a separately scrolling picker.

Both appendix figures are controlled live reproductions using the browser adapter, without a model or form submission. The page is reloaded between interaction paths. In each figure, the upper row shows the ablated configuration's accessibility-tree targeting path and the lower row shows the full configuration's element-targeted execution; the left panels show the interaction and the right panels show its resulting state. These captures explain control behavior separately from the scored task trajectories.

Template selection: preserving row identity. Figure 4 shows why a control's accessible name is insufficient when it appears in multiple rows. GitLab exposes 30 template radios named `Use template`. In the reproduced task-750 interaction, the ablated configuration's adapter scrolls and then locates the control again by role and name, resolving to Pages/Gatsby instead of the requested Pages/Plain HTML row. The resulting form confirms the incorrect template. Element-targeted execution selects the Plain HTML row's own label through its XPath, and the resulting form confirms Pages/Plain HTML. The relevant information is the association between the control and its template row, which must survive scrolling and target lookup.

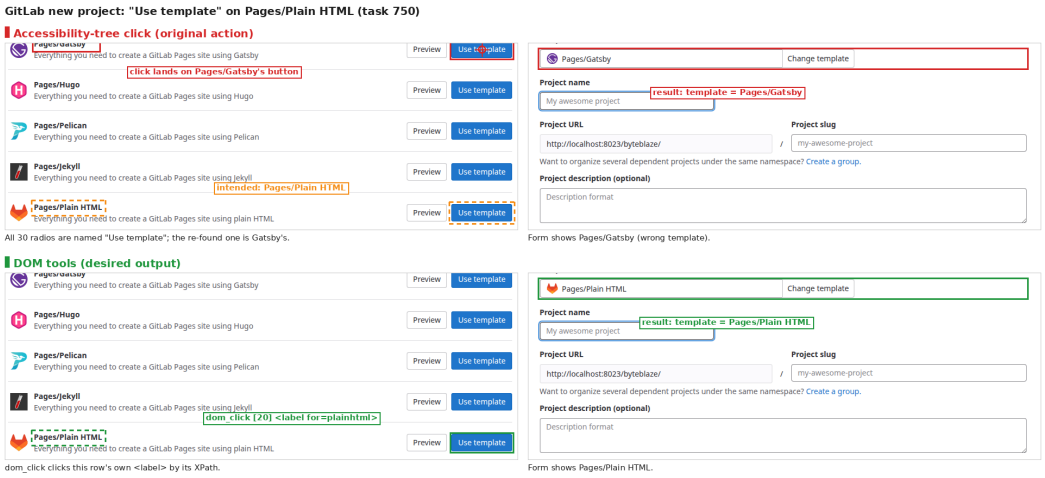

Figure 4: **Preserving template row identity.** Top left: the requested Pages/Plain HTML control is outlined in orange, but lookup by the repeated name `Use template` selects the Pages/Gatsby control, marked in red. Top right: the form displays Pages/Gatsby, confirming the incorrect selection. Bottom left: the DOM action targets the Plain HTML row's own label. Bottom right: the form displays Pages/Plain HTML, confirming the requested selection. This controlled reproduction of task 750 ends before project creation.

Forum selection: preserving interaction scope. Figure 5 shows a failure caused by scrolling the wrong container. The Reddit picker contains 96 options in an independently scrolling list approximately 200 pixels tall. The requested `relationship_advice` option lies below the visible portion of that list. Scrolling the outer page does not expose it, and the reproduced point-based click falls outside the viewport, leaving the picker unchanged. The element-targeted path searches within the picker, filters the list to the matching option, and selects that option. The displayed forum value then confirms the intended selection. This sequence preserves the interaction scope from search through selection and readback.



Figure 5: **Preserving forum-picker scope.** Top left: the requested option is outside the visible portion of the nested list; the point-based action attempts a click below the viewport. Top right: the picker still displays `Choose one`, confirming that no forum was selected. Bottom left: typing `relationship_advice` into the picker's search field exposes the matching option for element-targeted selection. Bottom right: the forum field displays `relationship_advice`. This controlled reproduction uses task 620's requested forum; its scored run instead succeeds through a URL shortcut. The lower panels illustrate the search-and-select procedure observed in other scored tasks, including task 600. No post is submitted during capture.

D RUNTIME AND INTERFACE DETAILS

The DOM Translation Layer represents page structure through a semantic *PageGraph* and exposes role-specific views through an `AgentInterface`. The PageGraph records regions, forms, tables, controls, and ownership. Exposed interface objects carry logical references, source generations, observed values, available operations, prerequisites, and possible effects; collection views additionally preserve scope and filters.

Hierarchical access provides a page overview (L0), task-focused regions (L1), local grounding and evidence (L2), and raw source access (L3). The Planner normally receives L0/L1, while the Executor receives the relevant L2 context with necessary ancestors and dialog or container scope. Source generations tie references to observation versions so stale mappings can be reacquired before execution.

E REPRODUCIBILITY RECORD

Supporting records include per-site metrics, evaluation indexes, second-attempt and widget-study ledgers, selected trajectory excerpts, run settings, and controlled widget captures. Appendix A documents the evaluation protocol, while Appendix C records the targeted execution intervention. Figures 1 and 2 are illustrative; Figures 3, 4, and 5 are controlled live reproductions rather than frames from scored trajectories.