WEBCOACH: SELF-EVOLVING WEB AGENTS WITH CROSS-SESSION MEMORY GUIDANCE

Genglin Liu *
University of California, Los Angeles

Shijie Geng
Amazon

Sha Li
Amazon

Hejie Cui
Amazon

Sarah Zhang
Amazon

Xin Liu
Amazon

Tianyi Liu
Amazon

ABSTRACT

Multimodal LLM-powered agents have recently demonstrated impressive capabilities in web navigation, enabling agents to complete complex browsing tasks across diverse domains. However, current agents struggle with repetitive errors and lack the ability to learn from past experiences across sessions, limiting their long-term robustness and sample efficiency. We introduce **WebCoach**, a model-agnostic self-evolving framework that equips web browsing agents with persistent cross-session memory, enabling improved long-term planning, reflection, and continual learning without retraining. WebCoach consists of three key components: (1) a *WebCondenser*, which standardizes raw navigation logs into concise summaries; (2) an *External Memory Store*, which organizes complete trajectories as episodic experiences; and (3) a *Coach*, which retrieves relevant experiences based on similarity and recency, and decides whether to inject task-specific advice into the agent via runtime hooks. This design empowers web agents to access long-term memory beyond their native context window, improving robustness in complex browsing tasks. Moreover, WebCoach achieves self-evolution by continuously curating episodic memory from new navigation trajectories, enabling agents to improve over time without retraining. Evaluations on the WebVoyager benchmark demonstrate that WebCoach consistently improves the performance of browser-use agents across three different LLM backbones. With a 38B model, it increases task success rates from 47% to 61% while reducing or maintaining the average number of steps. Notably, smaller base models with WebCoach achieve performance comparable to the same web agent using GPT-4o.

1 INTRODUCTION

Large language models (LLMs) have recently shown impressive skills in web and GUI navigation, enabling agents to fill forms, book flights, or compare shops across complex interfaces (Nakano et al., 2021; Wei et al., 2025; Wu et al., 2025b; Zhang et al., 2025e; Qin et al., 2025). This rapid progress spans both desktop-style pages and mobile apps, with techniques ranging from coordinate-free visual grounding to reinforcement fine-tuning and multimodal tutorial mining (Luo et al., 2025; Zhang et al., 2025a).

Despite these advances, web agents still waste many steps: they revisit the same links, stall at login gates, or trigger CAPTCHAs across sessions (Li et al., 2025; Lyu et al., 2025; Huang et al., 2025). Recent work introduces back-tracking or progress rewards to mitigate single-episode errors (Wu et al., 2025c; Zhang et al., 2025b), but agents rarely *remember* mistakes or successes beyond the current task. In short, most systems navigate without long-term memory – an ability humans rely on to internalize past experiences and anticipate pitfalls.

The absence of memory fundamentally limits sample efficiency and robustness. Even lightweight history compression or paging enhances web automation accuracy (Zhu et al., 2025; Kang et al.,

*Work done while interning at Amazon. Code available at `https://github.com/genglinliu/WebCoach`

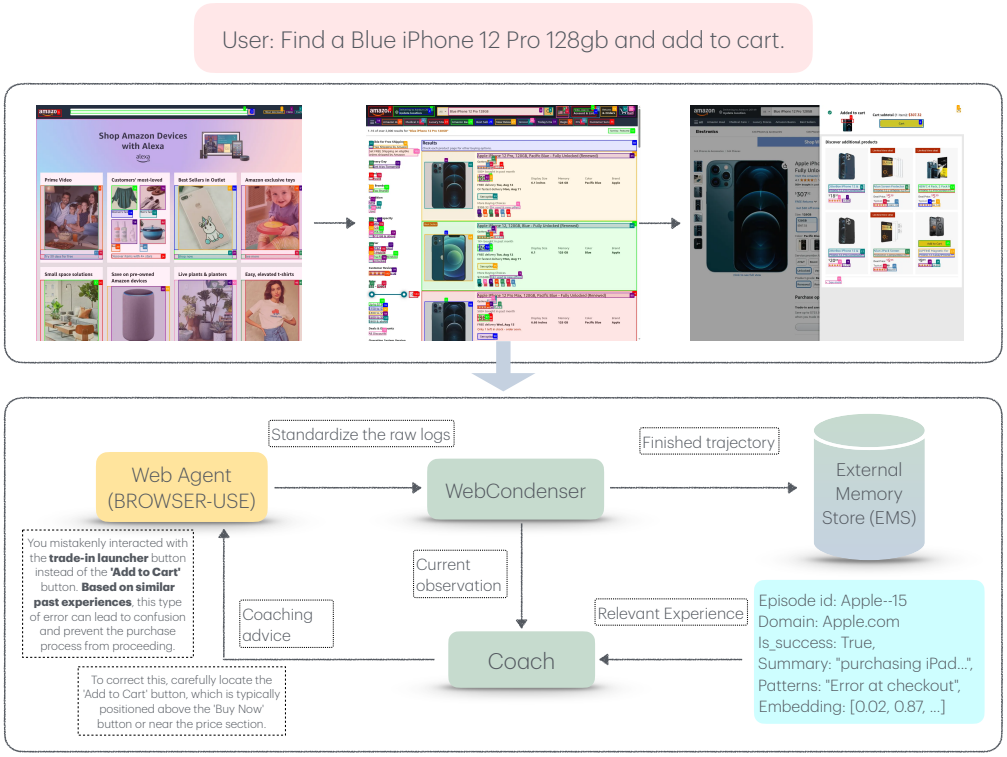

Figure 1: **Overview of the WebCoach framework.** WebCoach augments web-browsing agents with persistent, cross-session memory through an *External Memory Store (EMS)* and a retrieval-augmented coaching mechanism. The *Condenser* converts raw navigation histories into standardized summaries stored in *EMS*, from which the *Coach* retrieves relevant prior experiences to provide task-specific guidance to the main web agent. This design enables long-term planning, reflection, and continual improvement across browsing sessions.

2025), while episodic reflection and contextual replay are known to improve adaptation in other domains (Shinn et al., 2023; Liu et al., 2025b). However, these ideas have not been fully integrated into mainstream web-navigation pipelines.

We introduce **WebCoach**, a lightweight, model-agnostic framework that layers memory-aware guidance onto any existing web agent. WebCoach follows a simple premise: agents should learn from their own trajectories—successes, failures, and edge cases—*without* retraining the base policy. It realizes this through three modules: (1) a *WebCondenser* that converts raw interaction traces into compact semantic summaries; (2) an *External Memory Store* that indexes and retrieves relevant episodes; and (3) a *Coach* trainable LLM that decides when and how to intervene mid-episode. This architecture is framework-independent, as it wraps agents such as BROWSER-USE via simple trajectory hooks. Memories can be written online, shared across agents, and bootstrapped from curated traces, which avoids the lengthy reinforcement cycles demanded by prior improvements (Yin et al., 2025; Xie et al., 2025). As a result, even a small seed corpus yields useful advice from the first run.

On the WebVoyager benchmark (He et al., 2024), which spans 643 live browsing tasks across 15 web domains, WebCoach consistently improves long-horizon success rates and efficiency across diverse base models. When paired with the SKYWORK-38B agent, WebCoach raises the overall success rate from **47% to 61%** – a **14-point gain** – while maintaining or reducing the average number of steps per task. Other models, such as QWEN-VL-32B, exhibit similar gains, increasing their success rate from 49% to 57% and thereby achieving performance comparable to GPT-4o with open-source backbones. Notably, dynamic self-experience memory (where agents iteratively

expand their own memory store) outperforms externally seeded memories, highlighting that agents learn most effectively from their own trajectories rather than borrowed ones.

In summary, this work advocates for a new, memory-centered paradigm in web agent design – one where guidance is not hardcoded, but derived from experience. WebCoach is a step towards that goal, offering a practical and extensible architecture for building agents that truly remember and learn from their interactions.

2 WEBCOACH: A MODEL-AGNOSTIC FRAMEWORK FOR MEMORY-AUGMENTED WEB NAVIGATION

WebCoach is a lightweight, plug-and-play layer that augments any existing web-navigation agent (the *actor*) with memory-aware guidance. It operates through three decoupled components: **WebCondenser**, an **External Memory Store (EMS)**, and the **Coach**. These components communicate via function calls and therefore require *no* modifications to the actor's internal agentic workflow, enabling model-agnostic improvements on web browsing tasks.

2.1 WEBCONDENSER

The WebCondenser is the first stage in the WebCoach framework and serves as the bridge between raw agent trajectories and memory-aware reasoning. Its role is deliberately narrow but essential: to convert low-level environment traces into semantically meaningful summaries. After every environment step, the actor agent logs a JSON file describing its current trajectory. This includes its observations, actions, and intermediate rewards (not necessarily a numerical score but self-evaluations on the status of the task completion). The WebCondenser parses these raw traces and condenses them into a structured schema that includes a concise natural language summary, a dense embedding, and key metadata such as whether the task was successfully completed. More details of the WebCondenser's operations and output formats are provided in Appendix A.

Input. After every environment step, the actor writes a structured log describing the current partial trajectory $T_{1:t} = \left\{(o_i, a_i, r_i)\right\}_{i=1}^{t}$ (observation, action, reward). WebCondenser is agnostic to the exact schema, allowing compatibility with frameworks like BROWSER-USE, Nova-Act, or any other framework.

Processing. A small LLM ($\leq$ 8B) converts the raw trace into a fixed schema:

- `summary_text`: 3-5 sentences capturing the high-level outcome so far.
- `embedding`: 1536-d OpenAI embedding vector of the summary. This block could be replaced by other high quality embedding models as long as we keep the consistency across items in the EMS.
- `final_success`: `true`/`false`/`null` (task still running).
- `fail_modes` or `success_workflows`: Evidence with key steps for error analysis. If a complete trajectory successfully executed the user query and satisfied all the specified conditions, then the WebCondenser should highlight the workflows that led to this success; otherwise, it should summarize the errors or challenges that led to an unsuccessful attempt.

The model also detects whether the trace is *partial* or *complete*. This process ensures fast runtime performance and compatibility across different frameworks. Importantly, the WebCondenser is not designed to perform any reasoning or intervention itself. Instead, it acts as a lightweight, schema-normalizing filter that prepares trajectory data for downstream modules.

Routing. A crucial design decision is the distinction between partial and completed trajectories. While partial episodes (i.e., tasks still in progress) are streamed to the Coach for real-time decision-making, they are not stored. Only completed episodes that reach a natural stopping point are persisted to memory. This routing ensures the memory is populated with finalized examples that reflect either success or failure, preventing the accumulation of noise or transient decision states.

If the current step does not terminate the task, the record is streamed to the Coach for immediate use while *skipping* storage. Only once the task ends, the Condenser flags the trace as complete and persists it in the EMS; this prevents polluting memory with half-finished noise yet still gives the Coach a live context.

2.2 External Memory Store (EMS)

The External Memory Store (EMS) is the long-term memory backbone of the WebCoach framework. It accumulates all completed browsing episodes, storing a semantic embedding of each agent *experience* alongside the natural language summaries and metadata generated by the **WebCondenser**. This memory structure allows new browsing tasks to benefit from past experience, enabling efficient retrieval of relevant trajectories that reflect similar goals, domains, or interaction patterns.

Data schema. Each record stores $\langle \texttt{embedding}, \texttt{summary_text}, \texttt{meta} \rangle$, where `meta` information contains high-level identifications including `episode_id`, domain / URL root, user goal, model name, total steps, and timestamp.

Retrieval. The EMS is implemented as a vector database using FAISS with HNSW indexing Douze et al. (2025), which supports efficient top-K similarity search even as the memory grows to millions of episodes. Once candidate memories are retrieved, they are ranked by their similarity to the current context embedding, which is computed via a normalized dot product between the embedding vectors. HNSW builds a multi-layer navigable small-world graph that enables logarithmic-time approximate nearest-neighbor search with high recall (Malkov & Yashunin, 2018). In our implementation, we use FAISS with the HNSW-128.

Given the Condenser embedding $\mathbf{e}_t$ for the current partial trace, EMS returns the top-K past experiences using

$$\text{score}(\mathbf{e}_t, \mathbf{e}_i) \;=\; \frac{\mathbf{e}_t^\top \mathbf{e}_i}{\|\mathbf{e}_t\|_2 \, \|\mathbf{e}_i\|_2}\,.$$

Cold start. A key feature of the EMS is its generality: it is agnostic to the model, dataset, or domain an episode originated from. This universal design allows it to serve as a cross-actor, cross-task knowledge repository and enables bootstrapping. During a cold start phase, the EMS can be seeded with high-quality episodes from previously trained web agents, ensuring the Coach has relevant experience to draw upon from the very first online episode.

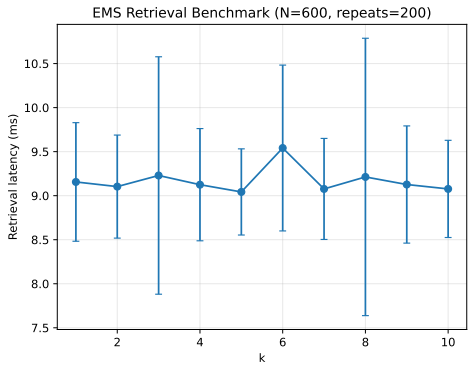

Figure 2: Retrieval speed at k for the EMS with 600 trajectories. Repeat each 200 times to measure the consistency. Most runs end up averaging between 9.0 and 9.5 ms for k ranging from 1 to 10.

2.3 Coach

The Coach is the reasoning engine of WebCoach. Unlike the WebCondenser and EMS, which deal with summarization and storage respectively, the Coach operates at runtime to enhance decision-making for the actor agent. At each step, the Coach receives a real-time summary of the actor's current partial trajectory along with a set of retrieved memories from the EMS. It then decides whether to intervene by providing the actor with additional guidance. It is implemented by a 8B LLM that decides whether to inject additional context into the actor's next prompt.

Inputs. The Coach leverages both the content and the outcome labels (success/failure) of past experiences to issue advice that is grounded in evidence. For example, it might say: "Avoid clicking 'Next'—previous agents got stuck in a loop here," or "Try clicking the first 'Used' filter on the left sidebar, it worked for others." The input will come from two branches simultaneously at each call:

1. The Condenser summary of the *current* partial trace ($\texttt{summary_text}_t$).
2. The top-K ($K = 5$) EMS summaries $\{\texttt{summary_text}_i\}_{i=1}^{K}$.

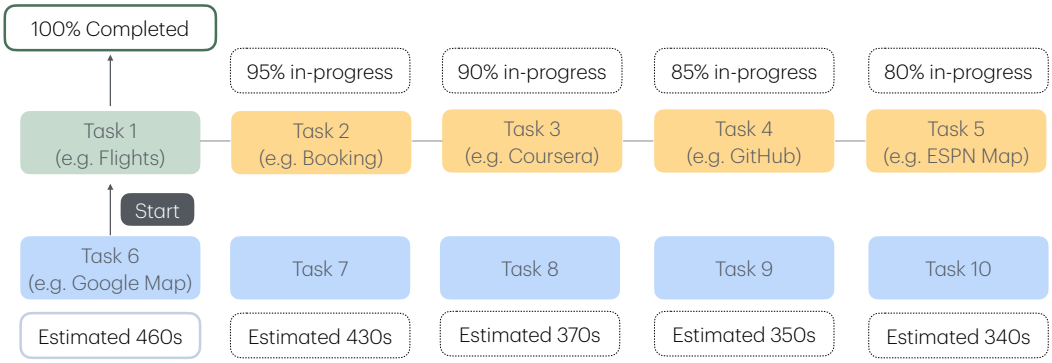

Figure 3: **Asynchronous Evaluation of WebVoyager.** WebCoach's asynchronous evaluation pipeline distributes the 15 subdomains in WebVoyager (e.g., *Amazon*, *Apple*, *ArXiv*) across parallel evaluation queues to maximize throughput and GPU utilization. Yellow boxes indicates in-progress tasks, green indicates completed tasks, and blue indicates tasks that are waiting in the queue. Our limited compute supports running 5 tasks in parallel, and once a task finishes earlier than the others in a batch, we immediately start another task from the wait list instead of waiting for the entire batch to finish. This asynchronous queueing strategy reduces total evaluation time by over 80%, enabling scalable benchmarking of web agents at large scale.

With 600 trajectories stored in the EMS, timings showed the cosine loop costs 10 ms per query regardless of k, so latency doesn't concern how we choose k's default value, as shown in Fig 2. We pick k=5 because presenting five closest past experiences gives the coach enough varied examples to detect patterns without overwhelming the LLM context window or drowning out the current state.

Decision rule. Intervention is selective. The Coach is explicitly trained or prompted to remain silent unless it detects high likelihood of failure (e.g., encountering CAPTCHAs, loops, dead ends) or recognizes a better workflow from memory. This ensures that interventions are timely and meaningful, rather than overwhelming the actor with irrelevant advice. The guidance itself is concise, usually a sentence or two. It is injected into the actor's prompt as a system message. This injection is entirely non-invasive: no gradients are backpropagated, and the actor's internal policy remains untouched.

Injection mechanism. The LLM is instructed to *intervene* when it predicts either a high failure probability (looping, CAPTCHA, HTTP 4xx), or a faster path exists in memory. Otherwise it returns `"intervene": false`. For `intervene=true`, the advice JSON is synchronously appended to the actor's message history as a system message before the next action selection. This requires no change to the actor's policy network.

The Coach offers a scalable way to improve agent reliability without any modifications to the actor's architecture. It enables continual improvement over time, as more episodes are accumulated and more patterns are learned—all without retraining the original navigation model. By separating summarization (WebCondenser), storage/retrieval (EMS), and decision-making (Coach), WebCoach remains *model- and framework-agnostic*. Each module is replaceable, enabling rapid experimentation with larger memories, improved retrieval, or custom intervention policies without disturbing the underlying navigation agent.

3 Experiments

3.1 Data

We conduct all evaluations in a real browser environment, using **WebVoyager** He et al. (2024). In contrast to prior work that relies on cached website snapshots (Deng et al., 2023; Lu et al., 2024) or sandboxed environments with a limited set of simulated sites (Yao et al., 2022; Zhou et al., 2023; Koh

et al., 2024), this benchmark involves live interaction with real-world webpages. This setup reduces the simulation-to-reality gap and better reflects the challenges faced by real users, such as dynamic content, login gates, and UI drift. WebVoyager complements this by enabling large-scale crawling and evaluation across a wide web surface, and remains one of the most widely used platforms for benchmarking web agents under online conditions (He et al., 2024).

By adopting an online benchmark, we ensure that our evaluation aligns with the deployment setting of web agents, where robustness to interface changes, session-level memory, and adaptation to unfamiliar page layouts are crucial. For evaluation, we utilize browser-use agent's evaluation capability by checking the resulting state of their last action against the initial user query, and determine the success of the task.

3.2 SETUP

All experiments are conducted in a real Chromium browser environment running inside Docker containers. A dedicated image is built for each run, and all dependencies are initialized at container startup to ensure consistent execution across checkpoints and models. Evaluations are performed on an NVIDIA A100 machine. To bound runtime and avoid degenerate navigation loops, we enforce a 30-second timeout per action step and a hard cap of 50 steps per task.

Asynchronous Evaluation with Dynamic Batching To increase throughput over WebVoyager's 15 subdomains, we adopt a two-level parallelization strategy. First, Docker containerization provides isolated browser instances. Second, a Python subprocess scheduler groups subdomains into batches and executes several in parallel, while tasks within each subdomain are processed sequentially to preserve browser determinism. Webvoyager has 15 distinct subdomains ("Amazon", "Apple.com", "Google Flights", etc) and each subdomain contains 30-50 tasks. In order to maximize evaluation speed and GPU utilization, we serve the open-source VLMs on vLLM and SGLang, and deploy the browser-use agent on chromium browser environment in groups of 5 subdomains at once. As illustrated in Fig 3, we maintain an evaluation queue of 5 subdomains, and whenever all tasks from one domain finished running, a new domain will enter the queue and be evaluated, until we process all tasks.

We treat each WebVoyager task as a job with an estimated runtime and schedule them on up to 5 parallel browser workers using an LPT-style (Longest Processing Time first) heuristic: longer tasks are launched earlier, and whenever a worker becomes free it pulls the longest remaining job from a global evaluation queue. This reduces tail latency and shortens the overall makespan compared to naive FIFO batching.

Take Qwen-VL-32b for example. Every task on average took 460s to complete, meaning that for all 643 tasks, a sequential evaluation would take 460*643/3600 = 82 hours to complete. With our parallelization queue strategy the actual running time for this evaluation turned out to be less than 14 hours, yielding a 83% reduction.

Base-Agent Configuration. The base web agent is a vision-language model (Qwen2.5-VL-7B, Qwen2.5-VL-32B, or Skywork-r1v3-38B) served on vLLM or SGLang. Optional chain-of-thought or verbose "thinking" modes are disabled unless explicitly required by an ablation, as they increase latency without observable gains in navigation performance. Apart from WebCoach's injected system messages, we use the default `browseruse` system prompt without modification.

Prompt Usage. While the functional roles of the `WebCondenser` and `Coach` are described in Section 2, here we note only the prompting behavior relevant to evaluation. The Condenser is used strictly for summarizing partial and complete trajectories under a fixed schema, and the Coach produces a lightweight JSON-formatted intervention decision. All prompts follow a fixed template to ensure deterministic behavior across runs.

Memory Retrieval and Leakage Control. Only completed trajectories are written to the External Memory Store (EMS). During evaluation, retrieval explicitly excludes any episode whose WebVoyager task ID matches that of the current task, preventing leakage of same-task experience. The EMS returns only experiences from distinct tasks or from prior runs of different subtasks. Embedding

similarity is computed using normalized dot products, and HNSW-128 indexing supports scalable nearest-neighbor search as the memory grows in the dynamic setting.

Reproducibility. Each run stores its configuration, Condenser outputs, Coach outputs, and (optionally) intermediate browser states. These logs enable reproducible replays, ablations, and cross-model comparisons.

4 RESULTS

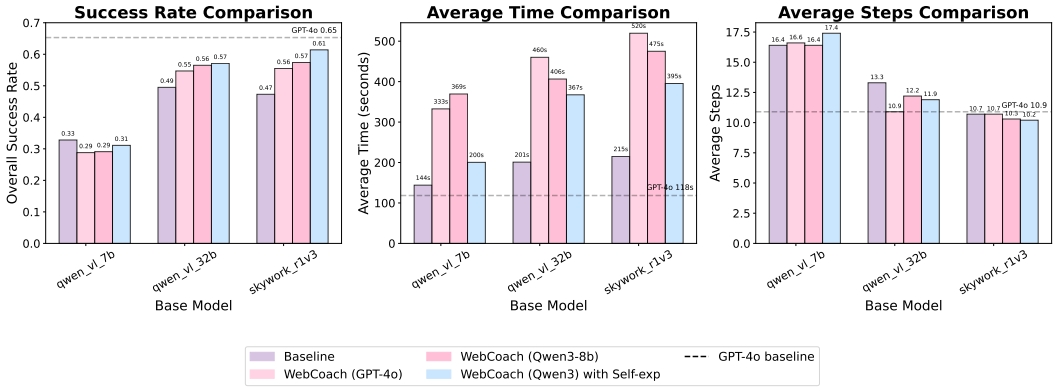

Figure 4: **Performance comparison across base models.** WebCoach consistently improves browser-use agents' reasoning and robustness across different backbones. The framework achieves higher success rates with equal or fewer average steps, while maintaining efficient completion times.

Memory	Coach Model	Base Model	Avg. Time (s)	Avg. Steps	Success Rate
None	None	GPT-4o	118.4	10.9	0.653
None	None	Qwen-VL-7B	144.0	16.4	0.328
None	None	Qwen-VL-32B	200.9	13.3	0.495
None	None	Skywork-38B	215.0	10.7	0.473
External (Frozen)	GPT-4o	Qwen-VL-7B	332.6	16.6	0.288
External (Frozen)	GPT-4o	Qwen-VL-32B	460.1	10.9	0.547
External (Frozen)	GPT-4o	Skywork-38B	519.6	10.7	0.555
External (Frozen)	Qwen3-8B	Qwen-VL-7B	369.4	16.4	0.291
External (Frozen)	Qwen3-8B	Qwen-VL-32B	406.4	12.2	0.565
External (Frozen)	Qwen3-8B	Skywork-38B	475.0	10.3	0.574
Self (Dynamic)	Qwen3-8B	Qwen-VL-7B	200.4	17.4	0.311
Self (Dynamic)	Qwen3-8B	Qwen-VL-32B	367.4	11.9	0.571
Self (Dynamic)	Qwen3-8B	Skywork-38B	395.4	10.2	**0.614**

Table 1: Success rate, average time, and average number of steps to completion across WebVoyager benchmark experiments. Columns reorganized to show memory type and coach model explicitly.

We conducted four sets of experiments on WebVoyager (He et al., 2024) across three base models: `Qwen2.5-VL-7B`, `Qwen2.5-VL-32B`, and `Skywork-r1v3-38B`, alongside a ceiling baseline with `GPT-4o`. Each run covered the full WebVoyager benchmark of 643 online tasks. The experiment configurations were:

1. Baseline: no coaching enabled.
2. Frozen EMS (GPT-4o coach): memory initialized with GPT-4o trajectories and GPT-4o acting as the coach.
3. Frozen EMS (Qwen3-8B coach): memory initialized with GPT-4o trajectories but coached by Qwen3-8B.

4. Dynamic EMS (Qwen3-8B coach): each main agent iteratively updates its own trajectories into the memory store, coached by Qwen3-8B.

Interestingly, GPT-4o did not consistently outperform Qwen3-8B as a coach, which led us to decide against DPO-based finetuning of Qwen3. We believe that Qwen3-8B is sufficiently capable to act as a condenser and coach in the zero-shot setting, since this task primarily depends on trajectory-level reasoning rather than complex instruction following.

Quantitative analysis. Figure 4 and Table 1 reveal a consistent pattern of improvement across both success rate and efficiency metrics. The largest jump occurs in the Skywork-38B model, which improves from 47.3% to 61.4% success rate when equipped with WebCoach, a 14.4-point gain, comparable to the GPT-4o ceiling baseline. The Qwen-VL-32B also improves by over 7 points (49.5% $\rightarrow$ 57.1%), confirming that experience-guided coaching scales effectively with model capacity. In all cases, the number of action steps per task remains flat or decreases, suggesting that the observed gains stem not from brute-force exploration but from better-informed decision paths.

While average completion time rises due to the additional inference overhead of the Coach and Condenser, the efficiency gain in navigation behavior outweighs the latency cost. For instance, dynamic EMS with self-expansion adds $\sim$150 seconds of average runtime relative to baseline, yet reduces redundant actions by 1–2 steps per episode. The overhead is also transient: as the EMS grows, HNSW's logarithmic search complexity ensures that retrieval remains highly efficient.

Effect of memory source and self-experience. We observe that self-generated experiences yield more transferable and actionable knowledge than foreign demonstrations. When agents iteratively grow their own EMS, they encode decision patterns consistent with their own inductive biases and representation space. For example, a Qwen-32B agent retrieves embeddings closer to its own prior successful action embeddings, resulting in smoother reasoning continuity across tasks. In contrast, GPT-4o-derived trajectories, while high-quality, occasionally inject stylistic mismatches. This explains why the dynamic EMS variant achieves higher success with lower step counts than frozen EMS, despite starting from an empty database.

Scaling effects and cognitive thresholds. The benefits of WebCoach are strongly correlated with model scale. Both Skywork-38B and Qwen-VL-32B show pronounced gains, while the 7B backbone does not benefit, with its success rate slightly decreasing (0.328-0.311). We hypothesize that there is a *cognitive threshold*, where smaller models lack the grounding and reasoning depth to exploit cross-episode memory effectively. Once models approach the reasoning frontier—able to nearly solve a task on their own—the coaching signal serves as a crucial disambiguator that helps avoid high-level decision errors (e.g., repeated scrolling, redundant navigation loops). Thus, memory guidance becomes more valuable at the boundary of partial competence rather than total ignorance.

Behavioral analysis. Qualitative inspection of logged traces further supports these trends. Agents equipped with WebCoach exhibit clearer high-level planning and fewer repetitive page visits. For instance, on the *Apple.com* and *Amazon* subtasks, coached agents learned to skip redundant login or trade-in prompts that previously caused deadlocks. In dynamic EMS mode, the coach often cited earlier self-generated traces when warning about such pitfalls—demonstrating retrieval-based reflection. In contrast, baseline agents frequently cycled between similar UI elements without learning from past errors. We have more visual recap of what our pipeline does at different steps of a web navigation task in Appendix A.

At the subdomain level, the largest improvements appear in semantically complex sites like *Apple*, *ArXiv*, and *BBC News*, where reasoning and element disambiguation are key. Simpler domains such as *Booking.com* and *Google Flights* show smaller or negligible gains, suggesting that memory is most beneficial for multi-step information extraction tasks rather than atomic button-clicking ones.

Summary. Overall, the results highlight that (1) WebCoach enhances task robustness through evidence-grounded reflection, (2) self-evolving EMS accelerates adaptation without external demonstrations, and (3) scaling the base agent amplifies the value of memory-guided reasoning. Together, these findings demonstrate that persistent, cross-session memory can close much of the performance gap between open-source and proprietary LLM agents on complex, dynamic web environments.

5 RELATED WORK

Reasoning-Centric Web and GUI Agents Research on interactive agents has advanced along three converging directions. First, GUI-centric studies combine reward shaping, curriculum design, and self-reflection to turn small or open-source VLMs into capable mobile or desktop controllers (Lu et al., 2025; Lian et al., 2025; Shi et al., 2025; Bai et al., 2024; Liu et al., 2025c). These studies often leverage automated supervision through goal synthesis, self-critique, and cross-platform action libraries to reduce annotation cost and boost generalization (Yang et al., 2025; Wu et al., 2025a; Xu et al., 2024; Wu et al., 2024). Second, work on web navigation integrates multi-turn RL, structured exploration, and hierarchical planning, improving performance on benchmarks like WebArena and WebShop while underscoring the importance of strong visual grounding (Wei et al., 2025; Zhang et al., 2025d; Gandhi & Neubig, 2025; Li et al., 2025; Yang et al., 2024; Putta et al., 2024; Zheng et al., 2024; Nakano et al., 2021; Gur et al., 2023; Lee et al., 2025). Finally, platform-level systems pursue generality by unifying planning, tool use, and self-evolution to automate complex desktop or Windows workflows with minimal task-specific engineering (Agashe et al., 2024; Zhang et al., 2024; Qiu et al., 2025).

Agentic Memory and Context Management Work on memory has evolved from simple history compression to scalable, structured substrates. While even simple summarization of past states improves automation accuracy, more advanced multimodal agents employ dual episodic-semantic stores that reduce storage while increasing precision (Zhu et al., 2025; Zhang et al., 2025c; Wang & Chen, 2025). Other approaches draw on operating-system analogies to organize information into short-, mid-, and long-term layers or action logs, enhancing performance across language, code, and GUI tasks (Kang et al., 2025; Tang et al., 2025; Gao et al., 2025b). Techniques like latent-space retrieval and Zettelkasten-style linking further extend reasoning across hundreds of thousands of tokens (Wang et al., 2025a; Yu et al., 2025; Xu et al., 2025). Workflow-oriented stores mine reusable action sequences to improve web and multi-agent performance (Wang et al., 2024; Rahman et al., 2025; Liu et al., 2025a), while production systems achieve constant-memory operation with lower latency and strong downstream gains (Zhou et al., 2025; Chhikara et al., 2025). These directions together indicate a shift toward adaptive, long-horizon memory for multimodal and collaborative agents.

Self-Evolving Agents Self-evolving agents refine themselves through experience rather than retraining. Foundational techniques include replay buffers, natural-language reflection, and automatic curricula, which improve performance in both web and embodied environments (Liu et al., 2025b; Shinn et al., 2023; Wang et al., 2023; Ouyang et al., 2025). Subsequent efforts discover reusable skills or APIs via program synthesis for cross-task transfer (Wang et al., 2025c; Zheng et al., 2025), while self-generated trajectories and scalable curricula close the gap between open-source and proprietary models (Qi et al., 2024; Patel et al., 2024). Recent extensions integrate world-model imagination and transition abstraction for long-horizon or mobile settings (Fang et al., 2025; Chae et al., 2024; Wang et al., 2025b). A comprehensive survey summarizes these evolving strategies and highlights open challenges for continual self-improvement (Gao et al., 2025a).

6 CONCLUSION

We presented WebCoach, a model-agnostic, self-evolving framework that augments web browsing agents with persistent, cross-session memory. By integrating a lightweight WebCondenser, a scalable External Memory Store (EMS), and a retrieval-based Coach, WebCoach enables existing agents to reflect on prior trajectories, generalize from experience, and recover from repeated errors without retraining. Through large-scale online evaluations on WebVoyager, WebCoach demonstrated consistent gains across three open-source LLM backbones, while maintaining comparable or lower average step counts. These results underscore the value of experience-driven memory in improving robustness and efficiency in web navigation. Future work will (1) integrate the external memory directly into the agent's internal policy to eliminate multi-LLM dependency, and (2) explore reinforcement-based optimization over long-term reward signals. More broadly, our findings highlight that memory-centric design is an essential step towards building continually improving, self-reflective web agents.

REFERENCES

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s: An open agentic framework that uses computers like a human. *arXiv preprint arXiv:2410.08164*, 2024.

Hao Bai, Yifei Zhou, Jiayi Pan, Mert Cemri, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495, 2024.

Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. *arXiv preprint arXiv:2410.13232*, 2024.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. Webevolver: Enhancing web agent self-improvement with coevolving world model. *arXiv preprint arXiv:2504.21024*, 2025.

Apurva Gandhi and Graham Neubig. Go-browse: Training web agents with structured exploration. *arXiv preprint arXiv:2506.03533*, 2025.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025a.

Xinzge Gao, Chuanrui Hu, Bin Chen, and Teng Li. Chain-of-memory: Enhancing gui agents for cross-application navigation. *arXiv preprint arXiv:2506.18158*, 2025b.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

Jing Huang, Zhixiong Zeng, Wenkang Han, Yufeng Zhong, Liming Zheng, Shuai Fu, Jingyuan Chen, and Lin Ma. Scaletrack: Scaling and back-tracking automated gui agents. *arXiv preprint arXiv:2505.00416*, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. *arXiv preprint arXiv:2506.06326*, 2025.

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. *arXiv preprint arXiv:2401.13649*, 2024.

Dongjun Lee, Juyong Lee, Kyuyoung Kim, Jihoon Tack, Jinwoo Shin, Yee Whye Teh, and Kimin Lee. Learning to contextualize web pages for enhanced decision making by llm agents. *arXiv preprint arXiv:2503.10689*, 2025.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

Shuquan Lian, Yuhang Wu, Jia Ma, Zihan Song, Bingqi Chen, Xiawu Zheng, and Hui Li. Ui-agile: Advancing gui agents with effective reinforcement learning and precise inference-time grounding. *arXiv preprint arXiv:2507.22025*, 2025.

Genglin Liu, Vivian Le, Salman Rahman, Elisa Kreiss, Marzyeh Ghassemi, and Saadia Gabriel. Mosaic: Modeling social ai for content dissemination and regulation in multi-agent simulations. *arXiv preprint arXiv:2504.07830*, 2025a.

Yitao Liu, Chenglei Si, Karthik Narasimhan, and Shunyu Yao. Contextual experience replay for self-improvement of language agents. *arXiv preprint arXiv:2506.06698*, 2025b.

Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*, 2025c.

Xing Han Lu, Zdenek Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue. In *ICML*, 2024. URL `https://openreview.net/forum?id=mUSPhG4uDW`.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025.

Yougang Lyu, Xiaoyu Zhang, Lingyong Yan, Maarten de Rijke, Zhaochun Ren, and Xiuying Chen. Deepshop: A benchmark for deep research shopping agents. *arXiv preprint arXiv:2506.02839*, 2025.

Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025.

Ajay Patel, Markus Hofmarcher, Claudiu Leoveanu-Condrei, Marius-Constantin Dinu, Chris Callison-Burch, and Sepp Hochreiter. Large language models can self-improve at web agent tasks. *arXiv preprint arXiv:2405.20309*, 2024.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*, 2024.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

Salman Rahman, Liwei Jiang, James Shiffer, Genglin Liu, Sheriff Issaka, Md Rizwan Parvez, Hamid Palangi, Kai-Wei Chang, Yejin Choi, and Saadia Gabriel. X-teaming: Multi-turn jailbreaks and defenses with adaptive multi-agents. *arXiv preprint arXiv:2504.13203*, 2025.

Yucheng Shi, Wenhao Yu, Zaitang Li, Yonglin Wang, Hongming Zhang, Ninghao Liu, Haitao Mi, and Dong Yu. Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment. *arXiv preprint arXiv:2507.05720*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending memoryllm with scalable long-term memory. *arXiv preprint arXiv:2502.00592*, 2025a.

Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. Mobile-agent-e: Self-evolving mobile assistant for complex tasks. *arXiv preprint arXiv:2501.11733*, 2025b.

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

Zora Zhiruo Wang, Apurva Gandhi, Graham Neubig, and Daniel Fried. Inducing programmatic skills for agentic tasks. *arXiv preprint arXiv:2504.06821*, 2025c.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, et al. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *arXiv preprint arXiv:2505.16421*, 2025.

Penghao Wu, Shengnan Ma, Bo Wang, Jiaheng Yu, Lewei Lu, and Ziwei Liu. Gui-reflection: Empowering multimodal gui models with self-reflection behavior. *arXiv preprint arXiv:2506.08012*, 2025a.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, et al. Gui-actor: Coordinate-free visual grounding for gui agents. *arXiv preprint arXiv:2506.03143*, 2025b.

Qinzhuo Wu, Pengzhi Gao, Wei Liu, and Jian Luan. Backtrackagent: Enhancing gui agent with error detection and backtracking mechanism. *arXiv preprint arXiv:2505.20660*, 2025c.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.

Bin Xie, Rui Shao, Gongwei Chen, Kaiwen Zhou, Yinchuan Li, Jie Liu, Min Zhang, and Liqiang Nie. Gui-explorer: Autonomous exploration and mining of transition-aware knowledge for gui agent. *arXiv preprint arXiv:2505.16827*, 2025.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*, 2024.

Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, et al. Zerogui: Automating online gui learning at zero human cost. *arXiv preprint arXiv:2505.23762*, 2025.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

Xiaoran Yin, Xu Luo, Hao Wu, Lianli Gao, and Jingkuan Song. Unlocking smarter device control: Foresighted planning with a world model-driven code execution approach. *arXiv preprint arXiv:2505.16422*, 2025.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Building generalized gui agents by learning from multimodal web tutorials. *arXiv preprint arXiv:2504.12679*, 2025a.

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, et al. Ufo: A ui-focused agent for windows os interaction. *arXiv preprint arXiv:2402.07939*, 2024.

Danyang Zhang, Situo Zhang, Ziyue Yang, Zichen Zhu, Zihan Zhao, Ruisheng Cao, Lu Chen, and Kai Yu. Progrm: Build better gui agents with progress rewards. *arXiv preprint arXiv:2505.18121*, 2025b.

Hongxin Zhang, Zheyuan Zhang, Zeyuan Wang, Zunzhe Zhang, Lixing Fang, Qinhong Zhou, and Chuang Gan. Ella: Embodied social agents with lifelong memory. *arXiv preprint arXiv:2506.24019*, 2025c.

Yimeng Zhang, Tian Wang, Jiri Gesi, Ziyi Wang, Yuxuan Lu, Jiacheng Lin, Sinong Zhan, Vianne Gao, Ruochen Jiao, Junze Liu, et al. Shop-r1: Rewarding llms to simulate human behavior in online shopping via reinforcement learning. *arXiv preprint arXiv:2507.17842*, 2025d.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, et al. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*, 2025e.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

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. *arXiv preprint arXiv:2307.13854*, 2023.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025.

Xiyue Zhu, Peng Tang, Haofu Liao, and Srikar Appalaraju. Turbocharging web automation: The impact of compressed history states. *arXiv preprint arXiv:2507.21369*, 2025.

A Appendix A: Step-by-Step Visual Illustrion of WebCoach's Operation

Task Description: On Apple's website, check if the HomePod mini in store is available in multiple colors and list them. Website: https://www.apple.com/

Coaching Coaching output is a json file including whether or not to intervene, and if so, the exact actionable advice message. For debugging purposes, we also present the retrieved experiences that include meta information, a summary, and current patterns with the specific web elements that the agent interacted with. At this step the coach detected some challenges and decides to issue an advice:
[Coaching advice output]
Based on past experiences, it's important to avoid getting stuck in loops or repeatedly failing to apply specific filters. Instead of navigating through the site without a clear path, look for a 'Choose your color' section or dropdown menu on the HomePod mini product page. If you can't find it, consider using the search function with specific terms like 'HomePod mini color options' or 'HomePod mini available colors' to directly locate the information. If you encounter any navigation issues or loops, immediately return to the product page and check for color options there.
[Retrieved Relevant Experience 1]
Summary: The user is attempting to find information about the MacBook Air color options on Apple's website. They started by scrolling the homepage, clicked on a MacBook Pro link, and then navigated through several unrelated pages like AirPods and iPhone. After using the search bar to find 'MacBook Air', they reached the product page but have been scrolling and searching without success in locating the color options section. The user has attempted various methods, including using the search function and looking for a 'Customize' button, but has not yet found the relevant information.
[Current Patterns]

1. Repeated Navigation to MacBook Air Page: The user repeatedly navigates to the MacBook Air product page, likely due to a misstep or a need to verify the correct page.
2. Persistent Scrolling on MacBook Air Page: The user continues to scroll down the MacBook Air product page in an effort to find the color options section, indicating that they haven't found the information yet.
3. Repeated Search Bar Interactions: The user frequently interacts with the search bar, possibly due to confusion about the page structure or uncertainty about the location of the color options.

Similarity score: 0.8895191550254822
[2 more relevant retrieved experiences omitted]
[Main agent returned results:]
is_done: true,
success: true,
long-term memory: Task completed: True - The available colors for the HomePod mini are: Blue, Yellow, Midnight, White, and Orange.
[Condenser output (to be stored in the EMS)]
Summary: The user successfully completed the task of checking the availability of the HomePod mini in multiple colors. They navigated from the Apple homepage to the TV and Home section, then to the HomePod mini product page, and finally identified the color options. Al-

though there were some steps where the user scrolled and searched for color options, they eventually found the color selection on the product page.
Success workflows:

1. name: Navigation to Product Page, description: The user navigated from the Apple homepage to the TV and Home section and then to the HomePod mini product page, which is a key step in accessing the product information.
2. name: Color Option Identification, description: The user successfully identified the available colors (Blue, Yellow, Midnight, White, and Orange) on the product page, which is the main goal of the task.
3. name: Scrolling and Searching for Color Options, description: The user attempted to locate the color options by scrolling through the product page, which eventually led them to the color selection section.

embedding: (...omitted...)

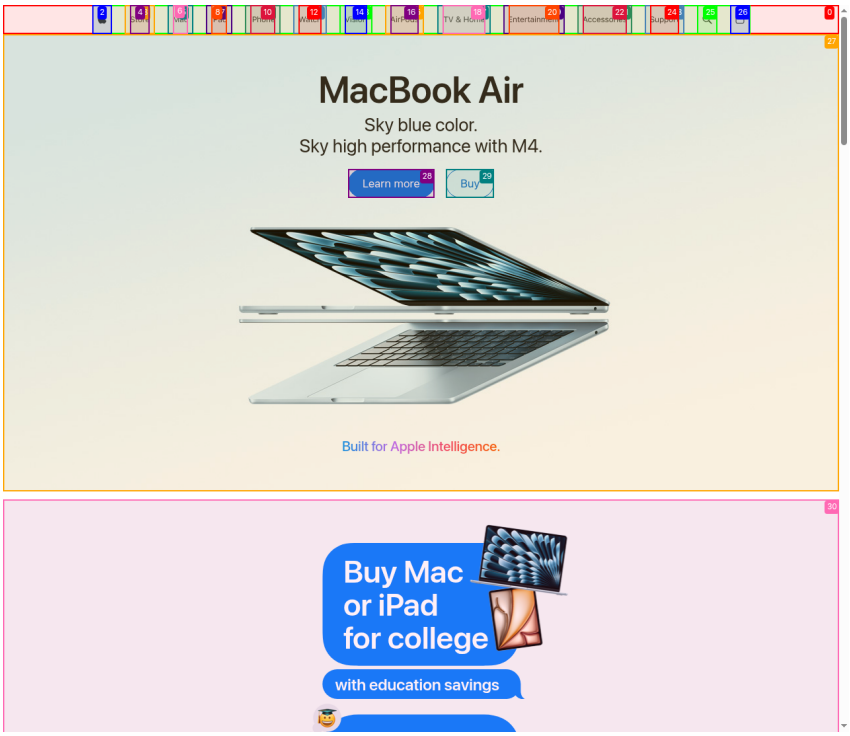

Figure 5: Step 1 Screenshot

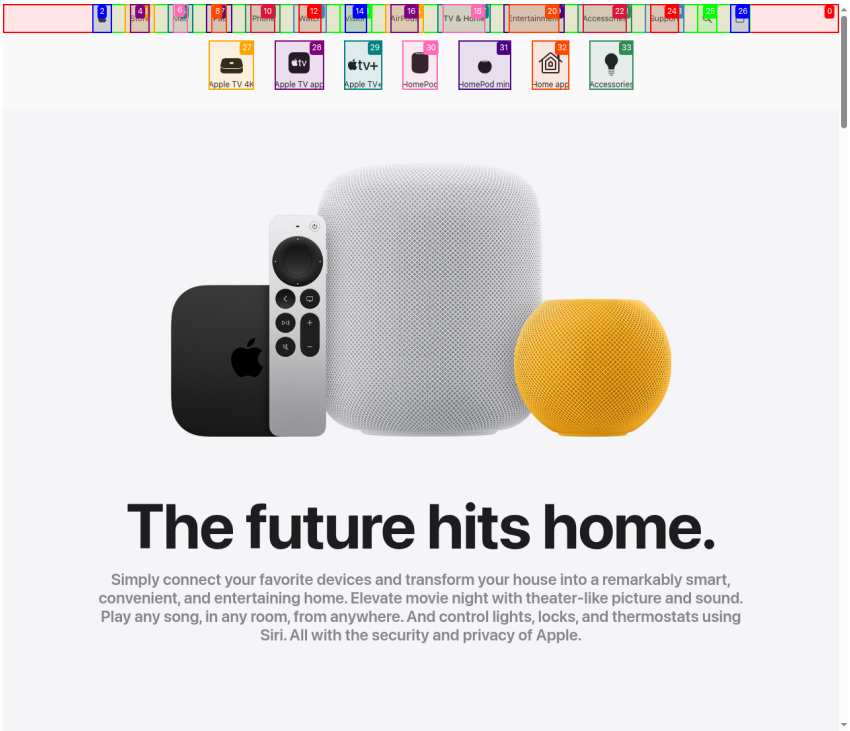

Figure 6: Step 2 Screenshot

Figure 7: Step 3 Screenshot

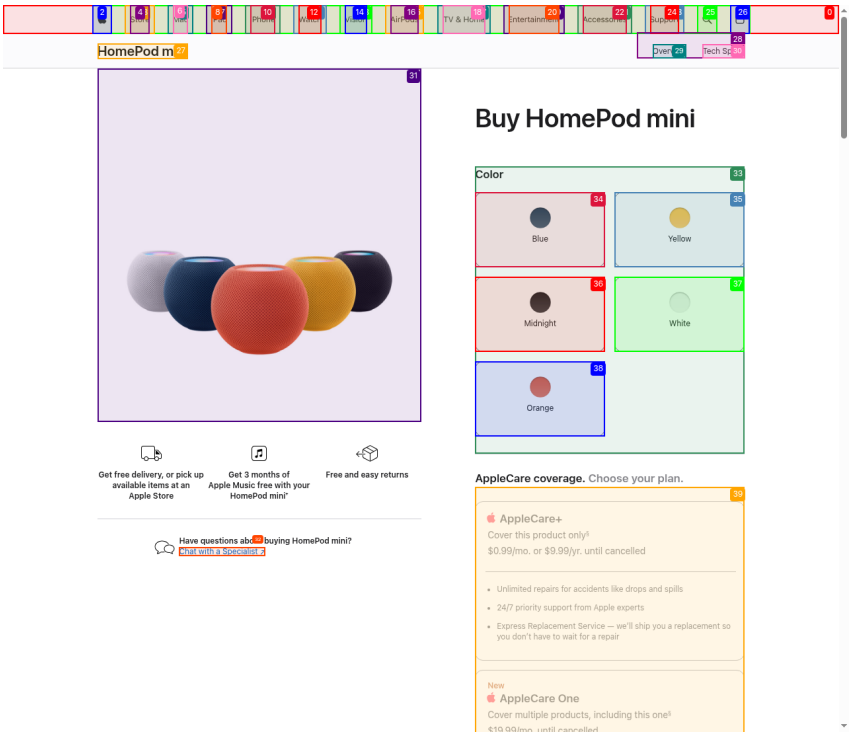

Figure 8: Step 4 Screenshot

B Appendix B: Full Per-Subtask Results

Subtask	GPT-4o				Qwen-VL-32B				Qwen-VL-7B				Skywork-r1v3			
	SR	S/T	Time	Steps	SR	S/T	Time	Steps	SR	S/T	Time	Steps	SR	S/T	Time	Steps
Allrecipes	0.889	40/45	67	7.4	0.644	29/45	254	12.3	0.356	16/45	142	16.1	0.556	25/45	204	11.0
Amazon	0.951	39/41	92	6.5	0.707	29/41	210	11.1	0.659	27/41	193	17.1	0.463	19/41	195	9.1
Apple	0.605	26/43	119	11.7	0.558	24/43	264	12.9	0.209	9/43	187	15.6	0.488	21/43	209	10.5
ArXiv	0.837	36/43	77	8.0	0.628	27/43	153	10.8	0.233	10/43	166	15.0	0.698	30/43	150	7.2
BBC News	0.714	30/42	87	10.3	0.571	24/42	159	12.0	0.190	8/42	141	14.3	0.690	29/42	181	9.4
Booking	0.136	6/44	303	18.4	0.068	3/44	398	18.6	0.182	8/44	162	17.5	0.045	2/44	297	12.7
Cambridge Dict.	0.953	41/43	48	5.3	0.837	36/43	92	7.6	0.628	27/43	115	13.9	0.721	31/43	119	6.4
Coursera	0.595	25/42	109	12.9	0.714	30/42	177	13.3	0.452	19/42	149	16.3	0.571	24/42	254	13.5
ESPN	0.659	29/44	114	11.7	0.432	19/44	205	14.2	0.182	8/44	178	17.9	0.455	20/44	201	9.5
GitHub	0.854	35/41	82	8.5	0.537	22/41	157	13.8	0.317	13/41	118	16.9	0.610	25/41	274	12.9
Google Flights	0.071	3/42	200	19.3	0.095	4/42	249	18.6	0.310	13/42	169	18.9	0.071	3/42	370	16.5
Google Map	0.878	36/41	81	8.2	0.561	23/41	141	11.6	0.463	19/41	112	15.8	0.610	25/41	205	10.4
Google Search	0.233	10/43	192	15.6	0.070	3/43	249	17.5	0.279	12/43	114	19.0	0.279	12/43	181	9.9
Huggingface	0.674	29/43	123	11.3	0.512	22/43	169	13.9	0.279	12/43	107	15.0	0.581	25/43	214	11.1
Wolfram Alpha	0.761	35/46	79	8.3	0.522	24/46	130	11.9	0.435	20/46	108	17.0	0.283	13/46	178	9.9
Overall	**0.653**	**420/643**	**118.4**	**10.9**	**0.495**	**319/643**	**200.9**	**13.3**	**0.344**	**221/643**	**144.0**	**16.4**	**0.473**	**304/643**	**215.0**	**10.7**

Table 2: Baseline: Without WebCoach, per-subtask comparison across base models on the **WebVoyager** benchmark. Each model's subcolumns report Success Rate (SR), number of successful vs. total tasks (S/T), average completion time (s), and average action steps. GPT-4o serves as the ceiling reference.

Subtask	Qwen-VL-32B				Qwen-VL-7B				Skywork-r1v3			
	SR	S/T	Time	Steps	SR	S/T	Time	Steps	SR	S/T	Time	Steps
Allrecipes	0.644	29/45	337	12.2	0.311	14/45	292	16.5	0.600	27/45	561	12.5
Amazon	0.805	33/41	370	11.8	0.585	24/41	370	17.4	0.829	34/41	556	11.1
Apple	0.395	17/43	353	13.0	0.233	10/43	347	14.8	0.395	17/43	667	13.7
ArXiv	0.698	30/43	282	9.9	0.209	9/43	312	14.4	0.721	31/43	322	7.1
BBC News	0.595	25/42	311	12.8	0.143	6/42	346	16.7	0.667	28/42	479	10.4
Booking	0.045	2/44	910	15.5	0.182	8/44	378	17.7	0.091	4/44	858	14.8
Cambridge Dict.	0.907	39/43	261	6.1	0.605	26/43	261	13.9	0.953	41/43	273	6.1
Coursera	0.690	29/42	637	13.2	0.357	15/42	348	17.7	0.738	31/42	535	11.3
ESPN	0.591	26/44	467	8.8	0.114	5/44	382	17.6	0.614	27/44	460	9.1
GitHub	0.829	34/41	520	10.4	0.195	8/41	294	17.0	0.829	34/41	526	10.6
Google Flights	0.071	3/42	926	17.5	0.119	5/42	367	18.0	0.095	4/42	882	17.8
Google Map	0.732	30/41	461	9.7	0.463	19/41	309	16.1	0.707	29/41	426	9.5
Google Search	0.233	10/43	314	7.0	0.140	6/43	336	18.3	0.302	13/43	413	9.0
Huggingface	0.674	29/43	425	8.9	0.163	7/43	317	15.3	0.605	26/43	460	10.1
Wolfram Alpha	0.348	16/46	341	7.5	0.500	23/46	330	17.0	0.239	11/46	383	8.3
Overall	**0.547**	**352/643**	**460.1**	**10.9**	**0.288**	**185/643**	**332.6**	**16.6**	**0.555**	**357/643**	**519.6**	**10.7**

Table 3: Per-subtask results **with WebCoach** (GPT-4o experiences, frozen EMS, GPT-4o as coach). Each subtask corresponds to a specific website domain. Metrics include success rate (SR), successful vs. total tasks (S/T), average completion time (s), and average action steps.

Subtask	Qwen-VL-32B				Qwen-VL-7B				Skywork-r1v3			
	SR	S/T	Time	Steps	SR	S/T	Time	Steps	SR	S/T	Time	Steps
Allrecipes	0.644	29/45	305	11.7	0.267	12/45	328	16.3	0.711	32/45	406	10.0
Amazon	0.805	33/41	276	9.9	0.488	20/41	349	15.0	0.732	30/41	477	9.7
Apple	0.488	21/43	402	14.5	0.233	10/43	354	15.8	0.605	26/43	460	10.5
ArXiv	0.721	31/43	261	10.0	0.233	10/43	290	14.3	0.791	34/43	268	5.9
BBC News	0.571	24/42	259	12.5	0.190	8/42	320	15.5	0.619	26/42	411	9.5
Booking	0.045	2/44	869	17.2	0.182	8/44	485	18.1	0.068	3/44	870	15.6
Cambridge Dict.	0.930	40/43	253	6.0	0.558	24/43	372	13.5	0.837	36/43	241	6.1
Coursera	0.690	29/42	553	12.1	0.333	14/42	510	18.0	0.690	29/42	589	12.6
ESPN	0.614	27/44	507	10.5	0.159	7/44	437	16.7	0.545	24/44	456	9.9
GitHub	0.805	33/41	447	9.3	0.220	9/41	445	18.4	0.878	36/41	412	9.2
Google Flights	0.119	5/42	538	18.4	0.214	9/42	377	18.2	0.095	4/42	892	18.0
Google Map	0.732	30/41	279	10.8	0.390	16/41	318	16.0	0.756	31/41	412	9.3
Google Search	0.070	3/43	504	17.3	0.116	5/43	342	18.5	0.279	12/43	401	9.1
Huggingface	0.535	23/43	359	12.6	0.233	10/43	320	16.1	0.651	28/43	435	10.0
Wolfram Alpha	0.717	33/46	277	9.8	0.543	25/46	299	15.9	0.391	18/46	400	8.7
Overall	**0.565**	**363/643**	**406.4**	**12.2**	**0.291**	**187/643**	**369.4**	**16.4**	**0.574**	**369/643**	**475.0**	**10.3**

Table 4: Per-subtask results **with WebCoach** (GPT-4o experiences, frozen EMS, Qwen3-8B as coach). Each model's subcolumns report success rate (SR), successful vs. total tasks (S/T), average completion time (s), and average action steps.

Subtask	Qwen-VL-32B				Qwen-VL-7B				Skywork-38B			
	SR	S/T	Time	Steps	SR	S/T	Time	Steps	SR	S/T	Time	Steps
Allrecipes	0.644	29/45	276	11.4	0.289	13/45	178	17.3	0.733	33/45	338	9.9
Amazon	0.805	33/41	250	9.7	0.488	20/41	189	15.9	0.756	31/41	397	9.6
Apple	0.512	22/43	363	14.2	0.256	11/43	192	16.7	0.651	28/43	383	10.4
ArXiv	0.721	31/43	236	9.8	0.256	11/43	157	15.2	0.814	35/43	223	5.9
BBC News	0.571	24/42	234	12.2	0.214	9/42	174	16.4	0.643	27/42	342	9.4
Booking	0.068	3/44	786	16.8	0.205	9/44	263	19.2	0.159	7/44	724	15.5
Cambridge Dict.	0.930	40/43	229	5.9	0.558	24/43	202	14.3	0.860	37/43	201	6.1
Coursera	0.690	29/42	500	11.8	0.357	15/42	277	19.1	0.714	30/42	490	12.5
ESPN	0.614	27/44	458	10.3	0.182	8/44	237	17.7	0.591	26/44	380	9.8
GitHub	0.805	33/41	404	9.1	0.244	10/41	241	19.5	0.878	36/41	343	9.1
Google Flights	0.143	6/42	486	18.0	0.238	10/42	205	19.3	0.190	8/42	742	17.9
Google Map	0.732	30/41	252	10.6	0.415	17/41	173	17.0	0.780	32/41	343	9.2
Google Search	0.093	4/43	456	16.9	0.140	6/43	186	19.6	0.349	15/43	334	9.0
Huggingface	0.535	23/43	325	12.3	0.256	11/43	174	17.1	0.674	29/43	362	9.9
Wolfram Alpha	0.717	33/46	250	9.6	0.565	26/46	161	16.9	0.457	21/46	333	8.6
Overall	**0.571**	**367/643**	**367.4**	**11.9**	**0.311**	**200/643**	**200.4**	**17.4**	**0.614**	**395/643**	**395.4**	**10.2**

Table 5: Per-subtask results **with WebCoach** (self experience, dynamic EMS updates, Qwen3-8B as coach). Each model's subcolumns report success rate (SR), successful vs. total tasks (S/T), average completion time (s), and average action steps.