ADAPTARENA: EVALUATING TEST-TIME PERSONALIZATION OF WEB AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents have demonstrated strong performance on complex web navigation tasks, yet they remain brittle in real-world settings where user intentions are underspecified and preferences are heterogeneous. In practice, users rarely provide explicit profiles, requiring agents to infer latent preferences from implicit signals. Despite its importance for deployment, this problem setting is largely underexplored in existing benchmarks. To address this gap, we introduce ADAPTARENA, a benchmark for evaluating test-time personalization of web agents via implicit preference inference. ADAPTARENA consists of 480 tasks, featuring both single-preference and complex compositional (double-preference) scenarios. Each evaluation task must be solved by retrieving and leveraging the most relevant historical user trajectory that implicitly encodes the target preference. In addition, we introduce ADAPTIVEAGENT, a retrieval-based framework for standardized evaluation of implicit preference inference. Experiments reveal a substantial performance gap: while oracle agents with access to ground-truth preferences achieve an 82.92% success rate, state-of-the-art LLM agents using our framework reach only 15.62%. Furthermore, we find that correctly inferring user preferences is necessary but not sufficient for task success, as execution failures in downstream web interactions remain a significant bottleneck even when agents align with the target preference. These findings highlight implicit preference inference and robust action grounding as key challenges for deploying reliable, user-facing web agents.

1 INTRODUCTION

Large language model (LLM) agents have recently demonstrated strong performance across a wide range of interactive tasks, including web navigation (Deng et al., 2023; Zhou et al., 2023; He et al., 2024; Lù et al., 2024), computer-use (Xie et al., 2024; Wang et al., 2025b) and tool-calling (Yao et al., 2024; Wang et al., 2024a). In particular, web-based LLM agents have emerged as a compelling frontier for human-computer interaction. By operating directly within real-world browser environments, these agents enable end-to-end task automation and offer a practical pathway toward general-purpose autonomy.

Despite their impressive general capabilities, current web agents struggle to adapt when deployed in novel environments. Existing research has tried to address this challenge through self-improvement mechanisms, such as learning from past experiences or environmental exploration (Wang et al., 2024b; 2025c; Zheng et al., 2025; Nekoei et al., 2025). While these efforts have significantly improved the robustness of web agents against new websites, a critical dimension of adaptation remains largely unexplored: the behavioral diversity of users who interact with those websites.

This gap reveals a fundamental barrier to real-world applicability. In practice, web agents must support users with diverse preferences and backgrounds, often without access to explicit profiles. Addressing this challenge through full model retraining is both costly and impractical, as it requires repeated parametric updates (Quiñonero-Candela et al., 2009) whenever user distributions shift. Instead, web agents should adapt to user-specific behaviors under a fixed model (i.e., without gradient-based updates). This motivates test-time personalization (Zhang et al., 2025; Qu et al., 2025) – a form of test-time adaptation (Wang et al., 2020) where a deployed agent adjusts its behavior at inference time to latent user preferences without modifying model parameters. A natural paradigm for test-time personalization is to leverage a user's past interaction history, which provides implicit

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

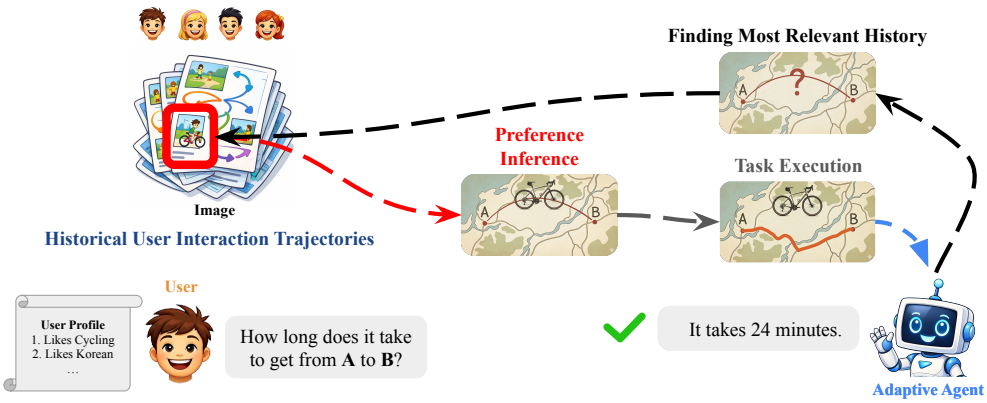


Figure 1: Benchmark overview. The agent resolves underspecified user tasks by inferring latent preferences from user history; in this example, the ADAPTIVEAGENT infers a preference for cycling and estimates the expected travel time from location A to B accordingly.

evidence of their preferences. This setting aligns with real-world usage, where explicit preference signals are rarely available but rich behavioral traces accumulate over time.

A few recent studies have tried to evaluate the personalization of web agents. PersonalWAB (Cai et al., 2025) relies on explicit, text-based user profiles, while CUPID (Kim et al., 2025) depends on direct verbal feedback to identify user needs. However, these benchmarks operate in unrealistic settings, as such curated metadata and explicit personas are rarely available in real-world browsing scenarios. Persona2Web (Kim et al., 2026) studies personalized web agents that resolve ambiguous web queries using browsing histories and user context, but primarily focuses on query disambiguation from static user histories rather than evaluating whether agents can retrieve and operationalize latent task-relevant preferences from historical interaction trajectories at test time.

Contributions. We introduce ADAPTARENA, a benchmark for evaluating test-time personalization of web agents through implicit preference inference from historical user interaction trajectories. ADAPTARENA comprises 480 tasks across three web environments, covering both single- and double-preference settings under underspecified user instructions. Unlike prior personalization benchmarks that provide explicit profiles or direct preference feedback, ADAPTARENA requires agents to recover task-relevant preferences from behavioral history. We further introduce ADAPTIVEAGENT, a simple and reproducible retrieval-based framework that provides a standardized way to retrieve relevant historical trajectories, infer latent preferences, and incorporate them into downstream web-agent execution. Across six backbone models, ADAPTIVEAGENT achieves at most 15.62% task success, compared with up to 82.92% with ground-truth preferences, revealing a substantial gap in implicit preference inference. Our analysis further shows a consistent drop from single to double preferences across models, highlighting the difficulty of compositional personalization. Finally, we find that correct preference inference does not reliably translate into successful execution, with downstream action failures remaining common even under correct preference alignment.

2 BENCHMARK

2.1 EVALUATION PARADIGM

In an ideal evaluation setting, a personalized agent would interact with a user over an extended period, gradually learning the user's preferences through natural interactions, and would subsequently be evaluated on new tasks that require integrating these preferences at test time. While realistic, such an interactive protocol is impractical for benchmarking: it is costly, difficult to scale, and introduces significant reproducibility challenges due to variability in interaction length and content.

To address these limitations, we adopt a two-phase evaluation paradigm that decouples preference acquisition from test-time evaluation. In the first phase, we construct *encoded user histories* by collecting interaction trajectories generated under explicit preference conditions. In the second phase,

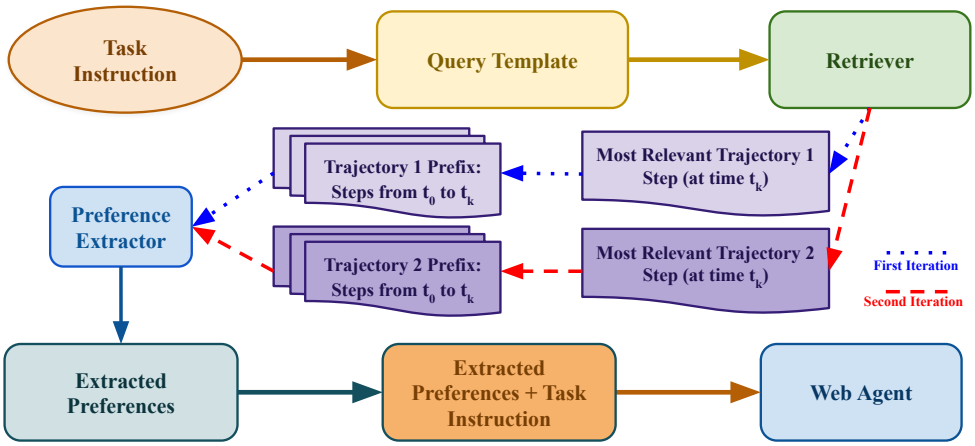


Figure 2: ADAPTIVEAGENT Structure. The task instruction is encoded as a retrieval query to identify the single most relevant trajectory step at time t_k. The corresponding trajectory prefix from t_0 to t_k is reconstructed, and the selected trajectory is then removed from the candidate pool. This process is repeated once to retrieve one additional relevant trajectory, yielding a total of two trajectories. Prefixes from the selected trajectories are aggregated into a unified sequence of screenshots, which is used to infer the user's implicit preference. The inferred preference is combined with the task instruction and provided to the web agent for execution.

agents are evaluated on new tasks in which all explicit preference information is removed. As a result, agents must infer the relevant preferences solely from the provided user history and act accordingly.

2.2 WEB ENVIRONMENTS

We construct tasks across three web environments drawn from WebArena (Zhou et al., 2023): an e-commerce store, a Reddit discussion forum, and OpenStreetMap. These environments are selected because they represent realistic settings in which personalization and user-specific preferences naturally arise. To enable agents to interact with our web environments, we use the BrowserGym (de Chezelles et al., 2025) platform, which provides accessibility trees, screenshots, and prompt interfaces. BrowserGym parses and executes actions produced by web agents, allowing for consistent and controlled interaction across tasks.

2.3 TASK ANNOTATION

2.3.1 TRAJECTORY COLLECTION

The primary objective of the trajectory collection phase is to construct user histories that encode clear preference signals. We define ten user personas; each persona is characterized by the same set of twelve preference categories, which are grounded across the three web environments.

Trajectory collection annotations consist of two components: a *Persona Trait* and a *General Instruction*. The Persona Trait encodes an explicit user preference (e.g., *prefers to drive*), while the General Instruction specifies the preference-agnostic task goal (e.g., *How long would it take to get from CMU to University of Pittsburgh?*). Together, these components form a user-specific instruction: *I prefer driving. How long would it take to get from CMU to University of Pittsburgh?* For each task, we design a user intent that encodes exactly one preference, with annotations manually reviewed to ensure solvability based on that preference alone.

Trajectory collection is performed by an agent acting as a user behavior generator, instantiated using Qwen-3-VL-32B-Thinking (Bai et al., 2025). In total, we construct 120 tasks for trajectory collection and obtain 120 corresponding user-specific trajectories, with twelve preference-aligned tasks and trajectories per persona across ten personas. Sample trajectories and user personas are provided in the Appendix B. We additionally analyze the quality and diversity of the collected trajectories. Across the 120 trajectories, the action distribution is dominated by clicks and fills,

reflecting the interaction repertoire required by the underlying environments. The full action-type and trajectory-length statistics are reported in Appendix B.5. To verify that the preference signal encoded in trajectories is recoverable without access to explicit annotations, we conduct an independent human study. Two annotators unaffiliated with the project are shown 60 shuffled raw trajectories with preference information removed and asked to identify the underlying preference. Humans recover the gold preference in 85.83% of cases on average, substantially above the 51.67% coverage obtained by our reconstructed-profile baseline.

2.3.2 DEPLOYMENT

This phase constitutes the primary evaluation stage of our benchmark, where agents are assessed on their ability to leverage user history at test time. To isolate this capability, we exclude any explicit preference information from the user intents. Agents must instead infer the appropriate preference from the provided user history and complete the task accordingly. Each task is designed such that successful completion depends on inferring a single, well-defined user preference. All deployment tasks undergo manual review to verify that (1) the task is underspecified without access to user history, and (2) it becomes unambiguous and solvable once the corresponding preference is inferred from the history. The collection is category-balanced by construction, with ten personas covering the same twelve preference categories. For deployment, we construct 480 tasks with equal numbers of single- and double-preference tasks. Task distributions are shown in B.6.

2.4 EVALUATION METRICS

2.4.1 TASK SUCCESS RATE

Following prior work (Zhou et al., 2023), we measure task performance using a *functional automatic evaluator* that returns a binary success label for each task. For every intent, a reference object is constructed to specify the success criteria. We employ two types of reference objects. The first is a reference answer, evaluated via string matching against the agent's output. The second is a *reference program*, used for tasks requiring information extraction. Each reference program specifies a reference URL, JavaScript-based locators for identifying target elements, and the expected content (a string or list), which is matched against the extracted result using exact or partial matching rules.

2.4.2 PREFERENCE ALIGNMENT SCORE

We evaluate an agent's ability to infer user preferences using a *Preference Alignment Score* computed by an LLM-based judge (Gu et al., 2024) implemented with Gemini-3-Flash. For each episode, the judge is provided with the complete interaction trajectory, along with a description of the target user preference. Based on the agent's observed navigation and interaction patterns, the judge determines whether the trajectory is consistent with the specified preference. To validate the reliability of this evaluation, we assess agreement between the LLM judge and independent human annotations using Cohen's κ (Cohen, 1960).

3 EXPERIMENT

3.1 BACKBONE MODELS

We evaluate six backbone models for our web agents: Gemini-3-Pro, Gemini-3-Flash (Google, 2025), GPT-5.4, GPT-5-mini (OpenAI, 2025b;a), Qwen-3.5-27B and Qwen-3-VL-32B-Thinking (Team, 2026; Bai et al., 2025). These models were selected for their strong reasoning capabilities and robust multimodal processing abilities, which are essential for web interactions.

3.2 ADAPTIVEAGENT

We introduce a simple ADAPTIVEAGENT framework that explicitly separates preference inference from task execution, without assuming access to any gold or pre-reconstructed user profile at inference time. Instead, the framework relies solely on past interaction trajectories as its source of supervision. At inference time, the framework performs user-specific retrieval over a database of past interaction

Table 1: Task success rates (%) of different models under varying levels of preference availability across 480 tasks. *Reconstructed Profile* benefits from partial access to ground-truth user preferences. *User-Centric* and *Oracle* assume full access to ground-truth user information and therefore serve as upper bounds, reflecting idealized conditions beyond realistic deployment scenarios.

Model	Baseline		Our Method	Upper Bound	
	No Profile	Reconstructed Profile	ADAPTIVEAGENT	User-Centric	Oracle
Gemini-3-Pro	7.71%	17.71%	15.62%	68.33%	82.92%
Gemini-3-Flash	9.17%	17.71%	15.00%	56.04%	72.29%
GPT-5.4	8.96%	14.17%	12.71%	51.04%	71.04%
GPT-5-mini	7.92%	14.17%	11.67%	41.87%	62.92%
Qwen-3.5-27B	8.96%	15.00%	11.04%	46.46%	66.87%
Qwen-3-VL-32B	7.29%	15.42%	10.42%	25.21%	40.83%

trajectories, where each trajectory is represented as an ordered sequence of screenshot-based steps $\{s_0, s_1, \ldots, s_T\}$. Given a task instruction, we first encode it into a shared embedding space and iteratively retrieve the most relevant trajectory using a CLIP-based similarity search with FAISS. From each retrieved trajectory, the framework then selects the step s_k (where $k \in \{0, \ldots, T\}$) most relevant to the current task. After retrieval, the selected trajectory is removed from the candidate pool. This process is repeated once to retrieve one additional relevant trajectory, yielding a total of two trajectories. For each retrieved trajectory, we construct a trajectory prefix spanning from the initial step s_0 to the selected step s_k. We then concatenate the prefixes from all selected trajectories into a single set of screenshots. The aggregated sequence of screenshots is provided to a preference extractor, implemented using GPT-5-nano, which infers the user's latent preferences. The inferred preference is expressed as a concise natural language summary and is then incorporated into the agent's prompt as an auxiliary context. We provide implementation details in Appendix C.8.

3.3 ALTERNATIVE AGENT DESIGNS

Baselines. The **No Profile** baseline measures how web agents perform when no user history is provided. The agent must rely solely on the task description and the observable environment. Additionally, the **Reconstructed Profile** is a strong baseline designed to assess whether user preferences can be inferred by jointly inspecting all trajectories from a single user, without requiring an explicit retrieval mechanism. During trajectory collection, the user behavior generator is instructed with an explicit user preference, which may implicitly influence its intermediate reasoning traces and action choices. To evaluate whether such information alone is sufficient for preference identification, we reconstruct a user profile by summarizing the agent's observed thoughts and actions and applying a strong LLM (GPT-5) to infer the underlying preference. Notably, this process effectively exposes the model to partial information about the ground-truth preference distribution. Empirically, the reconstructed profiles achieve an average coverage rate of *51.67%* with respect to the gold profiles across ten personas, highlighting that a substantial portion of user preferences can be recovered from trajectories alone. The resulting profile is then provided to the agent as a textual input during test-time. Sample reconstructed profiles are provided in the Appendix B.4.

User-Centric. This design evaluates agent performance when provided with the complete gold user profile, representing a realistic personalization setting in which multiple user preferences are available simultaneously. The agent must first identify which preference is relevant to the current task and then apply it appropriately during task execution. This setting therefore evaluates the full personalization process, including both preference identification and preference-conditioned execution, without requiring the agent to infer preferences that are absent from the provided profile.

Oracle. This design measures performance when the agent is given direct access to the single gold preference required for the task. Unlike the User-Centric setting, this removes the need for preference identification and isolates the agent's ability to correctly apply a known preference.

3.4 RESULTS

3.4.1 TASK SUCCESS RATE

No Profile and Reconstructed Profile baselines. Table 1 shows that in the absence of user history, all models perform poorly, with success rates ranging from 7.29% to 9.17%, highlighting the difficulty of resolving underspecified tasks without user-specific preference signals. Introducing reconstructed user profiles yields consistent improvements of approximately 5% to 10% across all models. For example, Gemini-3-Pro improves from 7.71% to 17.71%, and GPT-5-mini from 7.92% to 14.17%. It is important to note that reconstructed profiles are built with partial access to gold user preferences. As such, this setting provides a non-trivial amount of oracle information, and should be interpreted as a semi-privileged baseline rather than a fully realistic deployment scenario.

ADAPTIVEAGENT. In contrast, ADAPTIVEAGENT operates without access to any gold preference signals, instead relying solely on retrieval over past interaction trajectories at inference time. Despite this stricter and more realistic setting, our method remains competitive with the reconstructed profile baseline across all models. For instance, GPT-5.4 achieves 12.71% (vs. 14.17%) and Gemini-3-Pro achieves 15.62% (vs. 17.71%). Notably, ADAPTIVEAGENT consistently outperforms the no-profile baseline, demonstrating that even implicit preference signals recovered through retrieval provide meaningful gains. This result suggests that trajectory-based retrieval can recover a substantial portion of useful preference information without requiring explicit profile construction. The relatively small gap, despite the absence of privileged signals, highlights the effectiveness of our approach as a practical alternative for real-world deployment. Although the trajectory collector (Qwen-3-VL-32B-Thinking) shares a model family with one evaluated backbone, Qwen-3-VL-32B-Thinking obtains the lowest ADAPTIVEAGENT success rate (10.42%), suggesting that any model-family-specific stylistic advantage does not directly translate into higher benchmark performance.

User-Centric and Oracle Agents. The User-Centric agent, which has access to the complete gold user profile, substantially improves performance across all models, reaching 68.33% for Gemini-3-Pro. This demonstrates the upper bound achievable when full user information is available, though the agent must still identify which preferences are relevant to each task. The Oracle agent, which is given direct access to the single task-relevant preference, further increases performance to as high as 82.92%. The large gap between these upper bounds and all other settings underscores that the primary challenge lies not only in acquiring user preferences, but in accurately selecting and applying the task-relevant subset at inference time. Further analysis is in C.3.

3.4.2 PREFERENCE ALIGNMENT SCORE

In addition to task success, we evaluate each model using *Preference Alignment Score*. One of the most frequent error cases arises when the task requires understanding the user's product filtering preference, but the agent instead infers a product sorting preference (e.g., sorting by price or name), leading it to sort the products instead of filtering them according to the user's constraints.

Table 2: Preference alignment score across six models.

Model	ADAPTIVEAGENT
Gemini-3-Pro	27.71%
Gemini-3-Flash	26.67%
GPT-5.4	23.96%
GPT-5-mini	29.79%
Qwen-3.5-27B	22.50%
Qwen-3-VL-32B	24.58%

Across all backbone models, preference alignment scores (22.50%–29.79%) are consistently higher than the corresponding task success rates under ADAPTIVEAGENT (10.42%–15.62%), revealing a systematic gap between *preference understanding* and *preference-grounded task execution*. While models often correctly infer the user's intended preference, they frequently fail to operationalize it into successful interactions within the environment. This gap is particularly pronounced for GPT-5-mini, as it achieves the highest preference alignment score (29.79%) but only attains an 11.67% task success rate, indicating that correctly inferred preferences do not reliably translate into successful task completion. Similarly, Qwen-3-VL-32B-Thinking exhibits a preference alignment score of 24.58% but achieves only 10.42% success, suggesting substantial execution failures despite partial preference understanding.

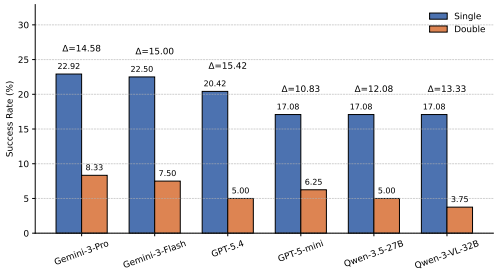


Figure 3: Task success rate of each model under single and double preferences.

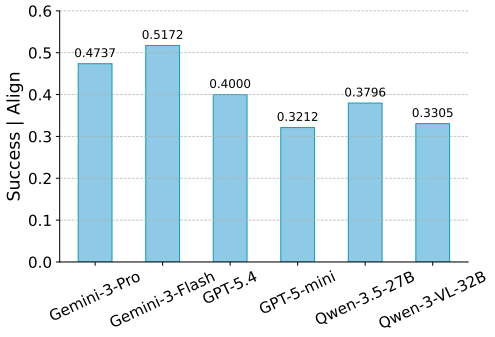


Figure 4: Conditional success rate across models.

To assess the reliability of the Preference Alignment Score, we conduct a human–LLM agreement study on a subset of episodes. The two human annotators exhibit near-perfect agreement (Cohen's $\kappa = 0.93$). The LLM judge also demonstrates strong agreement with human judgments (Cohen's $\kappa = 0.80$ and 0.87 against the two human annotators, respectively), supporting the use of LLM-based preference alignment as a reliable evaluation signal. Study details are provided in Appendix C.2.

4 Analysis

4.1 Single vs. Compositional Preferences

Figure 3 shows the performance of AdaptiveAgent across single-preference and double-preference tasks. Across all backbone models, we observe a consistent and substantial degradation when moving from single-preference to double-preference tasks. For instance, GPT-5.4 achieves 20.42% success on single-preference tasks but drops sharply to 5.00% on double-preference tasks, yielding a gap of 15.42 percentage points. Similar trends hold for other models: Gemini-3-Pro decreases from 22.92% to 8.33%, GPT-5-mini from 17.08% to 6.25%, and Qwen-3-VL-32B-Thinking from 17.08% to 3.75%. Performance remains challenging across domains: C.5 shows no domain–task-type cell exceeds 33% success, indicating that the difficulty is not confined to a particular environment. The single-to-double preference degradation is also consistent across domains and backbone models. While the absolute performance varies across backbones and domains, with Gemini-3-Pro generally achieving higher success on Map and Shopping tasks, the same qualitative trend persists across models.

This pattern reveals two key insights. First, compositional preference understanding remains a major bottleneck for current web agents. While models demonstrate limited ability to infer and apply a single latent preference, they struggle significantly when required to jointly reason over multiple constraints. Second, the magnitude of the gap, ranging from 10.83 to 15.42 percentage points across models, suggests that this limitation is systematic rather than model-specific.

A representative example of successful compositional alignment shows the agent correctly integrating both a driving preference and a location profile. It explicitly reasons that *the transport mode is set to Car (OSRM), which matches the user's preference*, while also grounding the task in *the user's location, Lucas County, Ohio*. Crucially, this reasoning is translated into action: when automatic detection fails, the agent manually inputs the location, demonstrating that both preferences actively guide decision-making rather than being passively satisfied. Ultimately, these findings underscore compositional personalization as a critical and underexplored challenge.

4.2 Retrieval Quality

The AdaptiveAgent pipeline acquires a preference in three stages: retrieving relevant trajectories, extracting a preference, and executing the task under that preference. Section 4.3 analyzes the latter two stages; here, we measure retrieval directly. Retrieval uses a frozen CLIP text query and FAISS inner-product search, making it deterministic and identical across all six backbones. We

define retrieval recall at the preference-category level. Each user encodes twelve preferences, one per category and trajectory. Retrieval succeeds when the top-K trajectories include the trajectory corresponding to the task's gold-preference category. The agent retrieves $K{=}2$ trajectories, so recall@2 is the operating point inherited by downstream stages. Recall@2 is 71.88% overall, 77.50% on single-preference tasks, and 66.25% on double-preference tasks; recall@1 is 59.17%. In the full pipeline, retrieval recovers the correct trajectory 71.88% of the time, but preference alignment is only 22.50–29.79% (Table 2), and task success is 10.42–15.62% (Table 1). Thus, most of the loss occurs after retrieval, in preference extraction and execution. This also explains why ADAPTIVEAGENT approaches the Reconstructed Profile baseline despite its stricter setting: downstream stages discard much of the signal that retrieval has already recovered.

Retrieval quality varies substantially across preference categories (Table 15). Location, Phone, Diet, Brand, and Reddit Forum achieve 100% recall@2, whereas Product Sorting and Wishlist vs. Cart achieve only 35.0% and 5.0%, respectively. The latter reflects a limitation of visual retrieval: sorting, filtering, and wishlist/cart actions can produce similar rendered states and differ mainly in the control action rather than the page appearance. Notably, Wishlist vs. Cart is retrieved only 5.0% of the time, yet achieves 63.3% alignment once the relevant trajectory is retrieved.

Separating retrieval from downstream stages explains the near-zero categories in Figure 11. Product Filtering is retrieved 75.0% of the time but aligns only 3.3%, indicating a downstream failure, whereas Wishlist vs. Cart is retrieved only 5.0% but aligns 63.3% once conditioned, indicating a retrieval failure. These contrasting cases show why retrieval must be measured separately: low task performance can arise because a preference is not retrieved or because it cannot be effectively used after retrieval, pointing to richer retrieval signals or stronger action grounding, respectively.

4.3 PREFERENCE ALIGNMENT VS. EXECUTION GAP

To isolate the source of the performance gap, we examine task success conditioned on correct preference alignment. Even when the inferred preference matches the ground truth, success rates remain limited: 47.37% for Gemini-3-Pro, 40.00% for GPT-5.4, and 37.96% for Qwen-3.5-27B, as shown in Figure 4. This reveals a consistent pattern: *correctly inferring user preferences is necessary but not sufficient for successful task completion.* Even when preference alignment is correct, execution failures remain common, indicating that downstream interaction control constitutes a distinct bottleneck beyond preference inference.

Table 3: Execution-level failure categories among unsuccessful episodes with correct preference alignment. AF: Action Failure, TF: Termination Failure, NL: Navigation Loop.

Model	AF	TF	NL	Other
Gemini-3-Pro	21.43%	44.29%	27.14%	7.14%
Gemini-3-Flash	35.29%	29.41%	30.88%	4.41%
GPT-5.4	37.68%	34.78%	21.74%	5.80%
GPT-5-mini	46.75%	35.06%	14.29%	3.90%
Qwen-3.5-27B	38.81%	46.27%	10.45%	4.48%
Qwen-3-VL-32B	40.51%	39.24%	12.66%	7.59%

To better understand these failures, we categorize unsuccessful aligned trajectories into four execution-level error types in Table 3: **Action Failure**, where the intended UI interaction fails; **Navigation Loop**, where the agent repeatedly revisits similar states without progress; **Termination Failure**, where the agent fails to stop appropriately; and **Other**, which covers execution failures that do not fit the above categories. The distribution of failure modes varies substantially across models. Termination Failure and Navigation Loop are prominent for several models, while Action Failure accounts for a substantial fraction of failures in others. For example, Termination Failure is the largest category for Gemini-3-Pro (44.29%) and Qwen-3.5-27B (46.27%), whereas Action Failure is most frequent for GPT-5-mini (46.75%), GPT-5.4 (37.68%), and Qwen-3-VL-32B (40.51%). Gemini-3-Flash exhibits a more balanced distribution across Action Failure (35.29%), Termination Failure (29.41%), and Navigation Loop (30.88%). These failures can occur even when the agent has correctly inferred the relevant user preference. For example, an agent may correctly infer that products should be sorted alphabetically, but fail to complete the task due to repeated navigation, unsuccessful UI interactions, or failure to terminate after obtaining the required information. In some cases, the agent explicitly recognizes the execution problem, e.g., *"we were not able to get the first product (the user's request) because the previous click attempts failed."* The most notable failure from *Other* category is environment instability, such as repeated page-load failures that prevent successful task execution. Overall, these findings suggest that downstream execution remains a major bottleneck even when the relevant user preference has been correctly inferred.

5 ABLATION

5.1 CROSS-USER TRAJECTORY ASSIGNMENT

To test whether performance gains depend on correctly aligned user context, we evaluate the agent under a *cross-user trajectory assignment* setting. Here, the agent is conditioned on trajectories $\{T_j\}$ generated from user U_j but evaluated on tasks X_i associated with a different user U_i. This preserves the trajectory distribution and task format while explicitly removing the correspondence between user intent and conditioning context.

Table 4: Success rate under cross-user trajectory assignment.

Model	ADAPTIVEAGENT
Gemini-3-Pro	7.50%
GPT-5-mini	7.71%
Qwen-3.5-27B	8.96%

To further stress-test this setting, we deliberately swap trajectories between users with the most contradictory preferences. As shown in Table 4, this misalignment consistently leads to a substantial drop in success rate across all models, reducing performance close to the no profile baseline in Table 1. If trajectory conditioning mainly captured task overlap or dataset artifacts, using mismatched trajectories should still provide some benefit. The degradation is observed across both Gemini-3-Pro and Qwen-3.5-27B, with success dropping from 15.62% to 7.50% and from 11.04% to 8.96%, respectively. These results preserve the trajectory distribution and task format while disrupting only the correspondence between user preferences and historical trajectories, providing evidence that the gains depend on correctly aligned user-specific signals rather than merely retrievable task overlap or generic trajectory artifacts.

6 RELATED WORKS

Benchmarks for Digital Agents Recent advances in Large Language Models have enabled agents to operate across diverse virtual environments, including web browsers (Yao et al., 2022a; Lù et al., 2024; Qi et al., 2025), mobile devices (Rawles et al., 2025; Ye et al., 2025), and operating systems (Wang et al., 2025b; Xie et al., 2025a; Lu et al., 2025), supported by specialized tools and evaluation harnesses (de Chezelles et al., 2025; Drouin et al., 2024). In parallel, a growing body of work has explored general-purpose LLM-based agents capable of planning and tool use, such as Yao et al. (2022b); Schick et al. (2023); Qin et al. (2023); Wang et al. (2023), as well as embodied and interactive agents operating in simulated environments (Shridhar et al., 2020; Mu et al., 2023). Despite this progress, existing benchmarks (Zhou et al., 2023; Deng et al., 2023; Xie et al., 2024; 2025b; Wang et al., 2025a) focus on task success rates, with limited attention to user preferences.

Agent Personalization Research on personalized agents has evolved from explicit user profiles toward retrieval-augmented generation (Salemi et al., 2024; Richardson et al., 2023) and memory-based approaches (Li et al., 2025; Su et al., 2026) that incorporate long-term user context. Another research (Li et al., 2025; Wu et al., 2025; Zhao et al., 2025; Kim et al., 2025; Zhang et al., 2025) works on user preference alignment through multi-turn dialogues from interactions. General User Models (Shaikh et al., 2025) present a general architecture that infers user knowledge and preferences from multimodal computer-use observations. In the web domain, benchmarks like PersonalWAB (Cai et al., 2025) and Persona2Web (Kim et al., 2026) have begun to evaluate personalization capabilities. A key open challenge remains enabling agents to infer and apply task-relevant implicit preferences from historical trajectories at test time.

7 CONCLUSION

We introduce ADAPTARENA, a benchmark for evaluating test-time personalization of web agents via implicit preference inference from historical user trajectories. To validate its difficulty, we propose ADAPTIVEAGENT, a simple framework that infers latent preferences from past interactions and integrates them into decision-making. Despite access to historical trajectories, ADAPTIVEAGENT achieves less than half the performance of an oracle with ground-truth preferences, identifying implicit preference inference as a key bottleneck in long-horizon web interactions. We further find that accurate preference inference alone is insufficient for successful execution, highlighting the need for robust downstream decision-making and action grounding.

AI USE STATEMENT

In this work, we used generative AI tools for polishing and improving the clarity and readability of the manuscript text, as well as for generating code used in our experiments. We have not used generative AI tools for developing research ideas, designing the methodology, analyzing or interpreting experimental results, or generating research claims. We reviewed all AI-assisted text for accuracy and consistency with the underlying work, and all AI-generated code was manually reviewed, tested, and verified for correctness. We take responsibility for the final content of this work, including all text, claims, code, and other research artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our proposed framework and experimental setup in the main paper, including trajectory collection, deployment task annotation and evaluation protocols. Additional implementation details, hyperparameters, and experimental results are provided in the appendix and supplementary materials. We will also provide the source code and necessary configuration files to facilitate reproduction of our experiments.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Hongru Cai, Yongqi Li, Wenjie Wang, Fengbin Zhu, Xiaoyu Shen, Wenjie Li, and Tat-Seng Chua. Large language models empowered personalized web agents. In *Proceedings of the ACM on Web Conference 2025*, pp. 198–215, 2025.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Thibault Le Sellier de Chezelles, Maxime Gasse, Alexandre Lacoste, Massimo Caccia, Alexandre Drouin, 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, Graham Neubig, Quentin Cappart, Russ Salakhutdinov, and Nicolas Chapados. The browsergym ecosystem for web agent research. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=5298fKGmv3. Expert Certification.

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.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/drouin24a.html.

Google. Introducing gemini 3: Our most intelligent ai model, November 2025. URL https://blog.google/products/gemini/gemini-3/. Accessed: 2026-01-07.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

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.

Serin Kim, Sangam Lee, and Dongha Lee. Persona2web: Benchmarking personalized web agents for contextual reasoning with user history. *arXiv preprint arXiv:2602.17003*, 2026.

Tae Soo Kim, Yoonjoo Lee, Yoonah Park, Jiho Kim, Young-Ho Kim, and Juho Kim. Cupid: Evaluating personalized and contextualized alignment of llms from interactions. *arXiv preprint arXiv:2508.01674*, 2025.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! llm-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5259–5276, 2025.

Dunjie Lu, Yiheng Xu, Junli Wang, Haoyuan Wu, Xinyuan Wang, Zekun Wang, Junlin Yang, Hongjin Su, Jixuan Chen, Junda Chen, Yuchen Mao, Jingren Zhou, Junyang Lin, Binyuan Hui, and Tao Yu. Videoagenttrek: Computer use pretraining from unlabeled videos, 2025. URL https://arxiv.org/abs/2510.19488.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*, 2024.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36:25081–25094, 2023.

Hadi Nekoei, Aman Jaiswal, Patrice Bechard, Oleh Shliazhko, Orlando Marquez Ayala, Mathieu Reymond, Massimo Caccia, Alexandre Drouin, Sarath Chandar, and Alexandre Lacoste. Just-in-time episodic feedback hinter: Leveraging offline knowledge to improve llm agents adaptation. *ArXiv*, abs/2510.04373, 2025. URL https://api.semanticscholar.org/CorpusID:281844145.

OpenAI. Introducing gpt-5 for developers (including gpt-5 mini), August 2025a. URL https://openai.com/index/introducing-gpt-5-for-developers/. Accessed: 2026-01-07.

OpenAI. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, 2025b. Accessed: 2026-05-04.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, Tianjie Zhang, Wei Xu, Jie Tang, and Yuxiao Dong. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning, 2025. URL https://arxiv.org/abs/2411.02337.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The twelfth international conference on learning representations*, 2023.

Zikun Qu, Min Zhang, Mingze Kong, Xiang Li, Zhiwei Shang, Zhiyong Wang, Yikun Ban, Shuang Qiu, Yao Shu, and Zhongxiang Dai. T-pop: Test-time personalization with online preference feedback. *arXiv preprint arXiv:2509.24696*, 2025.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence (eds.). *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2009. ISBN 978-0-262-17005-5.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William E Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. Integrating summarization and retrieval for enhanced personalization via large language models, 2023. URL https://arxiv.org/abs/2310.20081.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization, 2024. URL https://arxiv.org/abs/2304.11406.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S. Bernstein. Creating general user models from computer use, 2025. URL https://arxiv.org/abs/2505.10831.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

Miao Su, Yucan Guo, Zhongni Hou, Long Bai, Zixuan Li, Yufei Zhang, Guojun Yin, Wei Lin, Xiaolong Jin, Jiafeng Guo, et al. Beyond dialogue time: Temporal semantic memory for personalized llm agents. *arXiv preprint arXiv:2601.07468*, 2026.

Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Bowen Wang, Xinyuan Wang, Jiaqi Deng, Tianbao Xie, Ryan Li, Yanzhe Zhang, Gavin Li, Toh Jing Hua, Ion Stoica, Wei-Lin Chiang, Diyi Yang, Yu Su, Yi Zhang, Zhiguo Wang, Victor Zhong, and Tao Yu. Computer agent arena: Compare and test computer use agents on crowdsourced real-world tasks, 2025a. URL https://arena.xlang.ai/.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

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.

Jize Wang, Ma Zerun, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. Gta: a benchmark for general tool agents. *Advances in Neural Information Processing Systems*, 37: 75749–75790, 2024a.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Henry Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Boyuan Zheng, Peihang Li, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Jiarui Hu, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Dikang Du, Hao Hu, Huarong Chen, Zaida Zhou, Haotian Yao, Ziwei Chen, Qizheng Gu, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Flood Sung, Y. Charles, Zhilin Yang, and Tao Yu. Opencua: Open foundations for computer-use agents, 2025b. URL https://arxiv.org/abs/2508.09123.

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

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

Shujin Wu, Yi R Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. Aligning llms with individual preferences via interaction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7648–7662, 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis, 2025a. URL https://arxiv.org/abs/2505.13227.

Tianbao Xie, Mengqi Yuan, Danyang Zhang, Xinzhuang Xiong, Zhennan Shen, Zilong Zhou, Xinyuan Wang, Yanxu Chen, Jiaqi Deng, Junda Chen, Bowen Wang, Haoyuan Wu, Jixuan Chen, Junli Wang, Dunjie Lu, Hao Hu, and Tao Yu. Introducing osworld-verified. *xlang.ai*, July 2025b. URL https://xlang.ai/blog/osworld-verified.

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, 2022a.

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*, 2022b.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, et al. Mobile-agent-v3: Foundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*, 2025.

Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, et al. Personaagent: When large language model agents meet personalization at test time. *arXiv preprint arXiv:2506.06254*, 2025.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do llms recognize your preferences? evaluating personalized preference following in llms. *arXiv preprint arXiv:2502.09597*, 2025.

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, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023. URL https://webarena.dev.

A LIMITATION

ADAPTARENA has several limitations that define the scope of our conclusions.

First, our evaluation is conducted in three controlled web environments from WebArena through BrowserGym. While these environments provide reproducible and interactive settings, they do not capture the full diversity of real-world websites, including dynamically changing interfaces, longer-term user interactions, or domains with richer personalization signals. Second, historical trajectories are constructed under explicitly specified persona traits rather than collected from naturally occurring long-term user behavior. This controlled construction enables systematic evaluation and precise ground-truth preferences, but may not capture the ambiguity, inconsistency, and sparsity of preferences expressed in real user histories. Third, our benchmark primarily models preferences as relatively stable signals that can be inferred from past behavior. It therefore does not evaluate settings in which preferences evolve over time or depend strongly on context, task, or situational

constraints. Fourth, ADAPTIVEAGENT uses a specific screenshot-based retrieval design with CLIP embeddings and retrieves two trajectory prefixes. Finally, the trajectory collection model, Qwen-3-VL-32B-Thinking, belongs to the same model family as one evaluated backbone, introducing a potential source of stylistic bias in the collected histories. Although Qwen-3-VL-32B-Thinking achieves the lowest ADAPTIVEAGENT success rate among the evaluated backbones, ruling out such bias would require collecting trajectories with diverse models or, ideally, real users. Addressing these limitations with naturally occurring user histories, broader environments, dynamic preferences, and model-diverse trajectory collection is an important direction for future work.

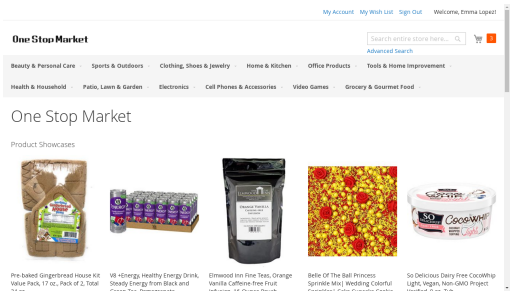
B BENCHMARK

B.1 SAMPLE TRAJECTORIES

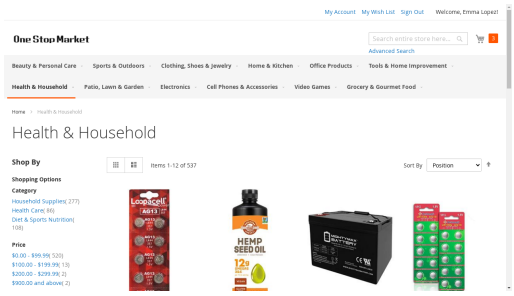
B.1.1 SUCCESS CASE

We provide a sample successful trajectory that captures the user's preference. The ground-truth user preference in this example is a preference for filtering Health & Household section in Health Care category.

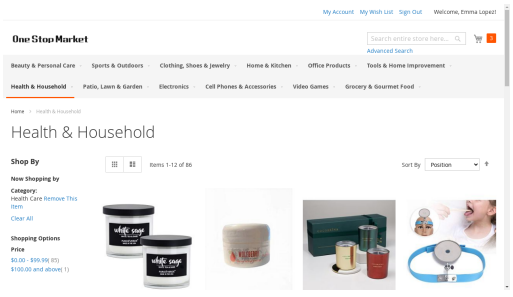
Task Instruction: I prefer buying Health & Households products from Health Care category. Can you tell me the price of the first product listed?



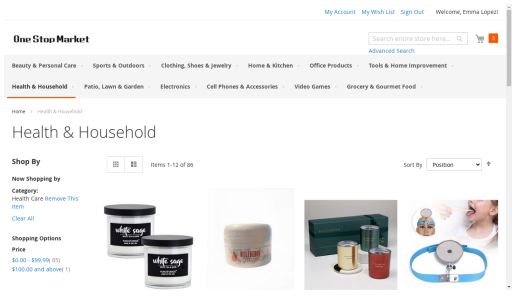
(a) Step 1



(b) Step 2



(c) Step 3



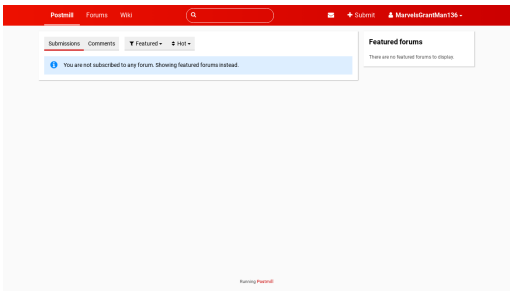
(d) Step 4

Figure 5: Sample successful trajectory shown as screenshot sequences.

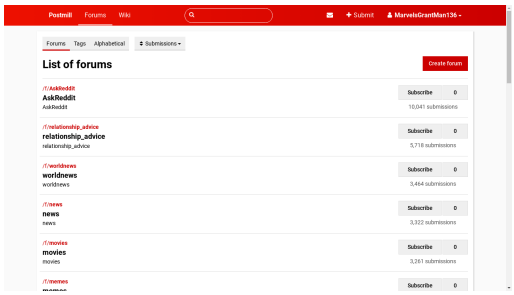
B.1.2 FAILED CASE

We also present a failed case in which the user preference is still correctly captured. The ground-truth user preference in this example is a preference for television Reddit forums.

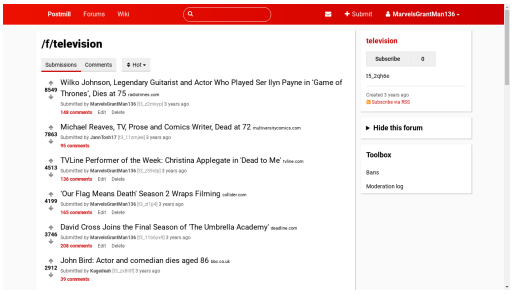
Task Instruction: I only use the television forum. Who wrote the second most active submission?



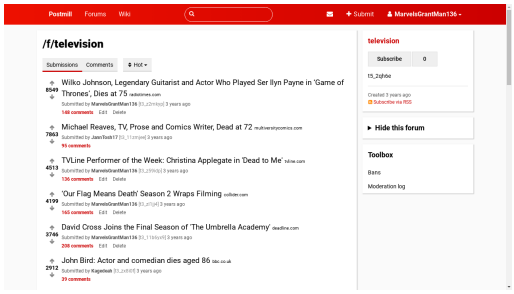
(a) Step 1



(b) Step 2



(c) Step 3



(d) Step 4

Figure 6: Sample failed trajectory shown as screenshot sequences.

B.2 Sample User Persona

Table 5: Sample profile.

Preference Type	Value
Product Sorting Preference	Ascending Product Name Order
Product Filtering Preference	Video Game filtering to XBox One Category
Wishlist vs. Cart Preference	Wishlist
Buying Preference	Lightweight product
Regional Product Preference	Korean
Brand Preference	Hebhac Herbs
Transportation Preference	Bicycle
Reddit Forum Preference	Movies
Reddit Forum Sorting Preference	Active
Diet Preference	Vegan
My Phone Model	Galaxy Z Flip 3
My Location	Holmes County, Mississippi

Table 6: Sample profile.

Preference Type	Value
Product Sorting Preference	Descending Product Name Order
Product Filtering Preference	Office products filtering to Office & School supplies category
Wishlist vs. Cart Preference	Wishlist
Buying Preference	Gaming product
Regional Product Preference	France
Brand Preference	LG
Transportation Preference	Walk
Reddit Forum Preference	Books
Reddit Forum Sorting Preference	Active
Diet Preference	non-GMO
My Phone Model	Galaxy Note 9
My Location	Lawrence County, Indiana

Table 7: Sample profile.

Preference Type	Value
Product Sorting Preference	Ascending Price Order
Product Filtering Preference	Health & Households filtering to Health Care category
Wishlist vs. Cart Preference	Cart
Buying Preference	Eco-friendly
Regional Product Preference	USA
Brand Preference	Estee Lauder
Transportation Preference	Bicycle
Reddit Forum Preference	Wallstreetbets
Reddit Forum Sorting Preference	New
Diet Preference	High-protein
My Phone Model	iPhone 13Pro Max
My Location	Franklin County, Vermont

B.3 Sample Trajectory collection and Deployment Tasks

Table 8: Sample single preference trajectory collection tasks from each domain.

Domain	Task Instruction	Gold User Profile
Shopping	I usually sort products by ascending price. Can you tell me the SKU code of the first product in Electronics category?	Sorting products in ascending price order
Shopping	I only purchase Sephora products. Can you add the powder brush to cart?	Prefers Sephora brand
Map	I live in Holmes County, Mississippi. How long does it take to get from here to lexington, mississippi by bicycle?	Lives in Holmes County, Mississippi
Map	I'd rather walk. Could you check how long it takes to get from British Museum to Tate Modern?	Prefers walking
Forum	I only browse the Movies forum. Can you tell me the title of the first submission listed in the forum?	Prefers Movies reddit forum
Forum	I only use the Books forum. How many years ago was the first submission on the list posted?	Prefers Books forum

Table 9: Sample single preference deployment tasks from each domain.

Domain	Task Instruction	Gold User Profile
Shopping	Could you save the Body Glide Original Anti-Chafe Balm for me?	Prefers to save products in shopping cart
Shopping	Could you search for waterproof phonecase for my phone, then tell me its manufacturer?	Uses iPhone12 Pro
Map	Can you search how long does it take to get from Cornell Tech to the Empire State Building?	Prefers driving
Map	How long does it take to get from here to Prairie View Elementary School, Oregon by walk?	Lives in Dane County, Wisconsin
Forum	How many upvotes does the most active submission have?	Prefers Space reddit forum
Forum	Can you tell me the number of comments for the first submission listed in the forum?	Prefers Funny forum

Table 10: Sample double preference deployment tasks from each domain.

Domain	Task Instruction	Gold User Profiles
Shopping	Can you tell me the manufacturer of the first video game product in the list?	1. Sorting products in ascending product name order 2. Filtering video game category into Xbox One subcategory
Shopping	Can you tell me the SKU code of the first electronics product in the list?	1. Sorting products in ascending price order 2. Filtering electronics category into video projectors subcategory
Map	How long does it take for me to go from here to Murray Forest Park, Bedford?	1. Prefers walking 2. Lives in Lawrence County, Indiana
Map	How long does it take for me to go from here to Willis Day Industrial Park, Wood County?	1. Prefers driving 2. Lives in Lucas County, Ohio
Forum	Can you tell me the title of the first forum submission in the list?	1. Prefers Wallstreetbets reddit forum 2. Prefers to sort reddit submissions in "New" Order
Forum	Can you tell me the name of the author of the third forum submission in the list?	1. Prefers mildlyinteresting forum 2. Sorting reddit submissions in "Active" Order

B.4 Sample Reconstructed Profile

Reconstructed profile for user01

I prefer vegan options and trusted brands (like Hebhac Herbs), and I often require products to be made in Korea. I usually research carefully by using search and name-ascending sorts, verifying SKUs, stock indicators, and origin details, then saving suitable items to my wishlist before buying. I tend to choose bicycling when looking up routes and rely on practical on-page cues (like breadcrumbs and Add to Cart buttons) to confirm context and availability.

Figure 7: Reconstructed profile for user01.

Reconstructed profile for user02

I prefer driving when getting directions and gravitate toward user-friendly products. I tend to be brand- and origin-conscious in shopping, buying only Sephora items in beauty and insisting on products made in Japan. I usually shop carefully with budget and details in mind—sorting by lowest price, checking stock status and SKUs, and selecting gluten-free options when relevant.

Figure 8: Reconstructed profile for user02.

Reconstructed profile for user03

I tend to search directly, sort and filter results, and focus on precise details like SKUs, counts, and timestamps before acting. I prefer trusted brands and clear quality markers (OEM accessories, non-GMO foods, and products made in France), and I often save items to a wishlist to compare them later. I usually shop for gaming gear and electronics alongside office supplies and gourmet items, and I often check walking times rather than driving when planning routes.

Figure 9: Reconstructed profile for user03.

B.5 User trajectory data statistics

We provide statistics for the user trajectory in table 11, 12, 13.

Table 11: User trajectory statistics.

Metric	Value
Trajectory Number	120
Total Number of Steps	1174
Average Total Steps per User	117.40
Average Number of Steps across Trajectories	9.78

Table 12: Action types across the 120 collected trajectories.

Action Type	Count	Fraction
Click	700	66.4%
Fill	137	13.0%
No-op	78	7.4%
Scroll	72	6.8%
Select option	36	3.4%
Focus	23	2.2%
Goto / Go back / Type	8	0.7%

Table 13: Trajectory length distribution.

Length	Count
3–5	43
6–9	46
10–15	13
16–29	5
31 (max)	13

B.6 Task Distribution

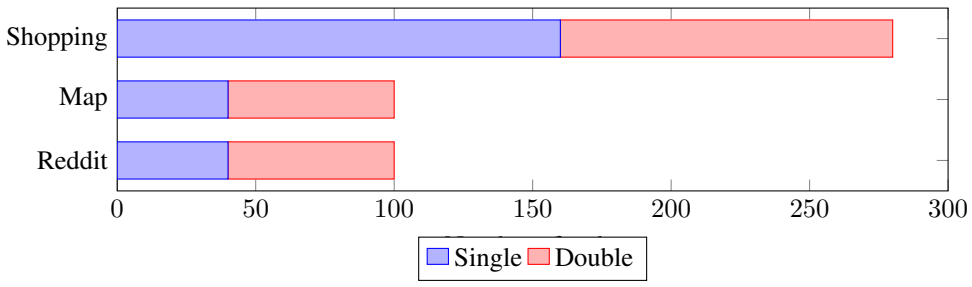


Figure 10: Distribution of the 480 deployment tasks across web domains and preference composition.

C EXPERIMENT DETAILS & ADDITIONAL RESULTS

C.1 COMPUTE RESOURCES

All experiments conducted in this study, including those for the deployment phase and the ADAPTIVEAGENT framework, were executed using NVIDIA A100 (80GB) GPUs. The substantial memory capacity of the A100 was essential for handling the large-scale multimodal inputs required for web navigation and trajectory processing. On average, a complete experimental run for a single ADAPTIVEAGENT configuration across the benchmark tasks required approximately 10 to 12 hours of execution time.

C.2 HUMAN–LLM AGREEMENT ANALYSIS

Human–LLM Agreement Analysis. To validate the reliability of the Preference Alignment Score, we conduct a human annotation study in which two annotators independently label whether an agent's interaction trajectory is behaviorally consistent with a specified user preference (binary labels: aligned or misaligned). Annotators are provided with the full interaction trajectory and preference description, and are blind to model identity and LLM judge outputs. We annotate 90 episodes sampled across three backbone models (Gemini-3-Pro, GPT-5.4, and Qwen-3.5-27B; 30 per model). Inter-annotator agreement is measured using Cohen's κ, yielding near-perfect human–human agreement ($\kappa = 0.93$). Agreement between the LLM-based judge and the two human annotators is also strong ($\kappa = 0.80$ and 0.87, respectively), supporting the reliability of LLM-based preference alignment as an evaluation signal.

No monetary compensation was provided because the annotators were the authors.

C.3 LIMITATIONS OF USER-CENTRIC AND ORACLE AGENTS

Both User-Centric and Oracle Agents reveal complementary limitations in personalized web interaction. User-Centric Agents primarily fail at *preference identification*: although they have access to the full ground-truth persona, they often struggle to determine which preference is relevant to the current task. Failures commonly arise from either over-applying multiple preferences or selecting irrelevant ones. For example, an agent may correctly recognize a preference for European cheese when searching for blue cheese, but additionally apply an unrelated product-sorting preference that alters the ranking and leads to an incorrect result. Similarly, when searching for a USB mouse, the agent may ignore the task-relevant gaming preference and instead prioritize a generic filtering heuristic. These cases show that richer personas alone do not guarantee better personalization; additional preferences can instead introduce more confounding signals. The core challenge is therefore not preference availability, but accurate *preference discrimination and selective application*.

In contrast, Oracle Agents remove preference ambiguity by explicitly providing the relevant preference, yet still exhibit substantial execution-level failures, closely mirroring the issues discussed in the *Preference vs. Execution Gap* analysis. Their errors stem primarily from weak UI grounding and brittle interaction strategies. Agents frequently rely on heuristic assumptions about interface structure (e.g., mapping "favorite" to Wish List or My Orders) without validating against the actual UI state, leading to repetitive navigation and looping behavior. They also tend to repeat failed interactions, assume nonexistent state changes, or continue re-checking pages even after the correct information has been found, reflecting poor adaptation and termination control.

C.4 PREFERENCE INFERENCE DIFFICULTY ACROSS CATEGORIES

Figure 11 shows preference inference performance across categories.

Categories such as Location and Phone achieve the highest alignment score across all models. Gemini-3-Pro reaches 90% on Location and 85% on Phone, while GPT-5.4 and Qwen-3.5-27B also perform strongly (roughly 60–70%). These preferences are grounded in clearly identifiable attributes, making them easier to extract and operationalize.

In contrast, Product Sorting, Product Filtering, and Reddit Forum Sorting are consistently the most challenging categories, with near-zero alignment score across all models. For example, Gemini-3-Pro

achieves only 5% on Product Sorting and 0% on Product Filtering, while the remaining models almost entirely fail. These tasks require interpreting procedural intent—such as distinguishing sorting from filtering operations—which agents frequently confuse. This indicates a fundamental limitation in capturing action-level semantics rather than static user attributes.

The difficulty gap becomes even more pronounced in the double-preference setting. While relatively simple combinations such as Transportation + Location retain moderate performance (33–40%), combinations involving structurally difficult preferences (e.g., Product Sorting + Product Filtering or Reddit Forum + Reddit Forum Sorting) collapse to near-zero across all models. This suggests that compositional preference inference is bottlenecked by the hardest constituent preference, highlighting the difficulty of jointly reasoning over heterogeneous user constraints.

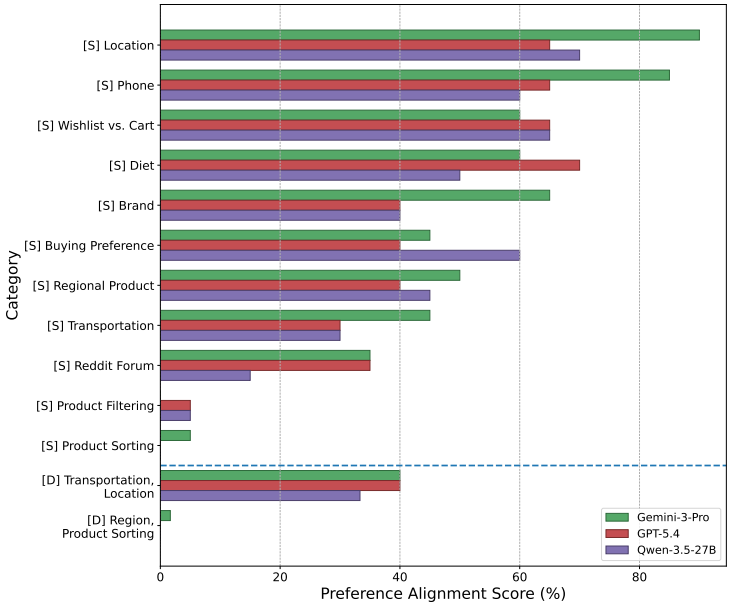


Figure 11: Preference inference performance across categories. Explicit attribute-based preferences (e.g., *Location*, *Phone*) achieve consistently high success rates, while operation-oriented preferences (e.g., *Product Sorting*, *Product Filtering*) remain near zero across all models. Categories with 0% alignment score across all models are omitted for clarity. [S]: Single, [D]: Double; separated by the dotted borderline

C.5 Disaggregated Results

Table 14: Task success rates (%) by domain and preference composition.

Domain	Model	Single	Double	Overall
Shopping	Gemini-3-Pro	25.0	7.5	17.5
	GPT-5.4	23.8	8.3	17.1
	Qwen-3.5-27B	20.0	7.5	14.6
Map	Gemini-3-Pro	32.5	18.3	24.0
	GPT-5.4	20.0	3.3	10.0
	Qwen-3.5-27B	17.5	5.0	10.0
Reddit	Gemini-3-Pro	5.0	0.0	2.0
	GPT-5.4	7.5	0.0	3.0
	Qwen-3.5-27B	5.0	0.0	2.0

C.6 Retrieval Quality

Table 15: Retrieval recall@2 versus preference alignment by category (single-preference tasks). Retrieval recall is deterministic and backbone-independent; alignment is the mean over the three backbones of Figure 11. The two signals diverge by category: Wishlist vs. Cart fails at retrieval (5.0%) yet aligns well once conditioned (63.3%), whereas Product Filtering is retrieved well (75.0%) yet almost never aligns (3.3%).

Category	Retrieval recall@2	Alignment (mean)
Location	100.0%	75.0%
Phone	100.0%	70.0%
Diet	100.0%	60.0%
Brand	100.0%	48.3%
Reddit Forum	100.0%	28.3%
Transportation	95.0%	35.0%
Reddit Sorting	90.0%	0.0%
Product Filtering	75.0%	3.3%
Regional	80.0%	45.0%
Buying	50.0%	48.3%
Product Sorting	35.0%	1.7%
Wishlist vs. Cart	5.0%	63.3%
Overall recall@2 (single)	77.5%	
Overall recall@2 (all 480)	71.9%	
Recall@1 (single)	59.2%	

C.7 HYPERPARAMETERS

We use the same hyperparameter settings from BrowserGym (de Chezelles et al., 2025), with max_prompt_tokens set to 40K and html_type set to pruned_html. For all models, we use the temperature of 1.0.

Table 16: Browsergym hyperparameters.

Hyperparameters	**Value**
use_ax_tree, use_focused_element, use_error_logs, use_history, use_action_history, extract_visible_tag, use_thinking, use_concrete_example, use_abstract_example, use_hints, be_cautious	True
use_html, use_past_error_logs, use_think_history, use_diff, use_screenshot, use_som, extract_clickable_tag, extract_coords, filter_visible_elements_only, use_plan, use_criticise, use_memory, enable_chat	False
Maximum Number of Steps	30

C.8 ADAPTIVEAGENT IMPLEMENTATION DETAILS

We provide additional implementation details of ADAPTIVEAGENT, an adaptive agent framework that separates preference inference from task execution via user-specific retrieval over past interaction trajectories. The framework maintains a database of trajectories for each user, where each trajectory is a sequence of timestamped screenshots $s_0, s_1, \ldots, s_T$. At initialization, all screenshots are embedded using a pretrained CLIP ViT-B/32 model, and the resulting feature vectors are ℓ_2-normalized and cached. For each user, we construct a FAISS inner-product index over these embeddings to enable efficient similarity search.

At inference time, a task instruction is encoded into the same embedding space using the CLIP text encoder. We then perform sequential retrieval: at each step, we identify the single most similar screenshot in the remaining pool, recover its associated trajectory, and record the corresponding step index k. The selected trajectory is subsequently removed from the pool to encourage diversity, and the process is repeated for a fixed number of iterations. For each selected trajectory, we reconstruct a

prefix consisting of all screenshots from s_0 to s_k, and aggregate prefixes from multiple trajectories into a unified set of images. These images are encoded in base64 format and provided as input to a multimodal preference extractor instantiated with GPT-5-nano. The model is prompted to infer the user's latent preferences strictly based on observable interaction patterns, producing a concise first-person natural language summary (e.g., "I prefer..."). The inferred preference is injected into the agent prompt as an auxiliary context block (`<image_summary>`), which is concatenated with the current task instruction, observation history, and action space specification. The downstream web agent then conditions on both the task and inferred preference to generate actions. All embedding computations and FAISS indices are cached on disk to avoid recomputation, enabling scalable and efficient deployment across multiple users.

C.9 PROMPTS

System prompt for Trajectory Collection

You are an user behavior generating agent solving a web task based on the content of the page and user instructions.

You MUST obey user preferences explicitly stated in the task instruction.

Also, you can interact with the page and explore.

Figure 12: System prompt for trajectory collection.

System prompt for ADAPTIVEAGENT

You are an expert assistant skilled in inferring user preferences from past interaction trajectories. You will receive a sequence of images representing a user's past interactions with interfaces.

These images may come from one or multiple interaction trajectories. Carefully analyze only the visible actions and decisions shown in the images to infer the user's preferences.

Do not speculate or assume anything that is not directly supported by the visual evidence.

If the interactions reflect a single consistent pattern, describe it clearly. If multiple distinct or complementary patterns are present, integrate them into a coherent preference description.

Provide a grounded, evidence-based summary written in the first person (e.g., 'I prefer...', 'I tend to...') as a single concise sentence.

Figure 13: System prompt for ADAPTIVEAGENT.

System prompt for LLM judge on preference alignment score

You are an expert evaluator of web-agent behavior.

Task:
Given a sequence of screenshots representing an agent's interaction trajectory, assess whether the agent's behavior is reasonably consistent with one or more specified user preferences.

Preference: {preference}

Evaluation Criteria:

- Each stated preference should be reflected in the agent's actions when relevant.
- Evidence may be direct or reasonably inferred from the screenshots.
- Minor deviations or unrelated actions should not outweigh overall alignment.
- Focus on the dominant behavior across the trajectory rather than isolated steps.
- If multiple preferences are provided, all of them should be supported by the trajectory.

Decision Rule:

- If a single preference is given, output YES if the agent generally follows it.
- If multiple preferences are given, output YES only if all preferences are generally followed.
- Output NO if any preference is largely ignored or contradicted.

Output Format:

Return exactly one token: YES or NO.

Figure 14: System prompt for LLM judge on preference alignment score.