WebUncertainty: Dual-Level Uncertainty Driven Planning and Reasoning For Autonomous Web Agent

Lingfeng Zhang[1], Yongan Sun[1], Jinpeng Hu[1], Hui Ma[1], Ying Yang[1], Kuien Liu[1,2], Zenglin Shi[1]*, Meng Wang[1]
[1]Hefei University of Technology [2]Academy of Cyber, CETC Group

Abstract

Recent advancements in large language models (LLMs) have empowered autonomous web agents to execute natural language instructions directly on real-world webpages. However, existing agents often struggle with complex tasks involving dynamic interactions and long-horizon execution due to rigid planning strategies and hallucination-prone reasoning. To address these limitations, we propose WebUncertainty, a novel autonomous agent framework designed to tackle dual-level uncertainty in planning and reasoning. Specifically, we design a Task Uncertainty-Driven Adaptive Planning Mechanism that adaptively selects planning modes to navigate unknown environments. Furthermore, we introduce an Action Uncertainty-Driven Monte Carlo tree search (MCTS) Reasoning Mechanism. This mechanism incorporates the Confidence-induced Action Uncertainty (ConActU) strategy to quantify both aleatoric uncertainty (AU) and epistemic uncertainty (EU), thereby optimizing the search process and guiding robust decision-making. Experimental results on the WebArena and WebVoyager benchmarks demonstrate that WebUncertainty achieves superior performance compared to state-of-the-art baselines.

1 Introduction

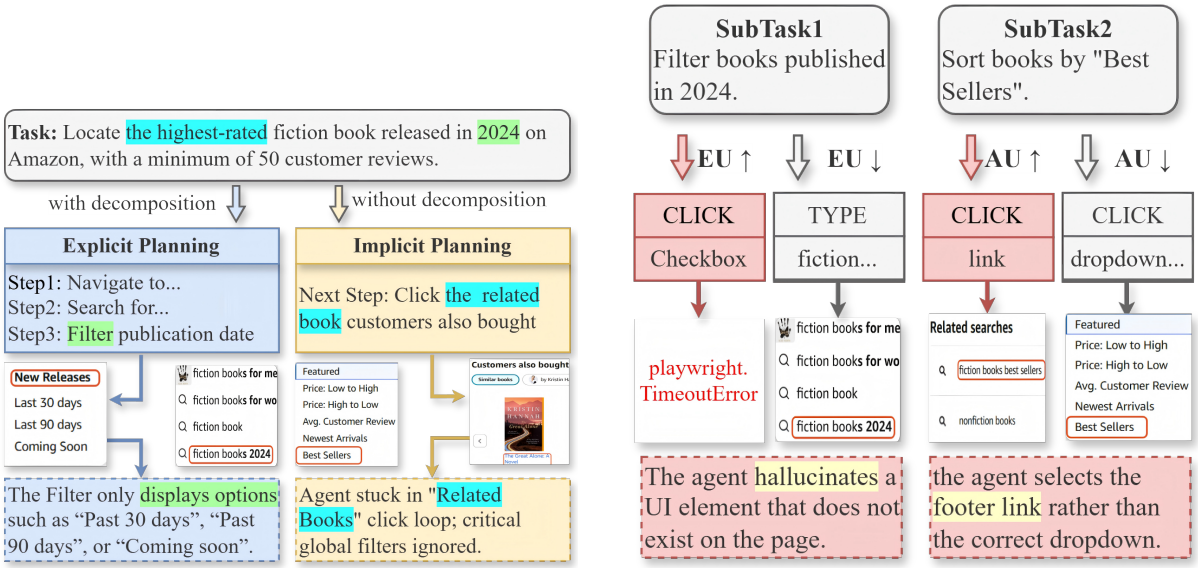
Web automation facilitates online services, including information aggregation, transaction processing, and enterprise workflows (Deng et al., 2023; Zheng et al., 2024). However, existing solutions based on hand-crafted scripts, programmatic APIs, and Robotic Process Automation (RPA) tools are brittle and task-specific, often failing under new tasks or minor interface changes(Liu et al., 2018; Pu et al., 2023). Recent advances in large language models (LLMs) with strong natural language understanding and reasoning capabilities (Deng et al., 2024; Du et al., 2026; Zhang et al., 2026a) enable more flexible web agents that execute instructions directly on real-world webpages (Hu et al., 2025; Nguyen et al., 2025; Ning et al., 2025). To enhance the reliability of these agents, recent studies have equipped them with planning mechanisms (Erdogan et al., 2025; Luo et al., 2025; Shahnovsky and Dror, 2026) to decompose user instructions into manageable subgoals, and reasoning mechanisms (Koh et al., 2024; Zhang et al., 2025; Wei et al., 2026) to guide the decision-making process. Despite these advancements, current agents still struggle with complex tasks requiring dynamic interaction and long-horizon execution (Wu et al., 2025; Yang et al., 2025b).

First, complex tasks involve dynamic web interactions. This dynamism makes it difficult for preplanned subgoals to adapt to unknown environments (He et al., 2024; Zhou et al., 2024). As shown in Figure 1a, an agent employing one-shot explicit planning intends to use the "Publication Year" filter to select 2024. However, it overlooks that the Amazon sidebar lacks this option, resulting in an execution failure. Conversely, iteratively generating subgoals via implicit planning introduces a different risk. It can distract the agent with the highest-rated book on the current page, causing it to neglect the global rating filter and fall into a local optimum. The agent can effectively resolve these issues by flexibly selecting its planning mode based on the webpage state and task progress. For instance, the agent can use implicit planning to correct cognitive biases during date filtering, and explicit planning to reduce local noise during rating sorting (Luo et al., 2025).

Second, complex tasks involve long-horizon execution, where reasoned actions are highly prone to errors due to LLM hallucinations and the snowball effect (Gan et al., 2025; Xia et al., 2025). As shown in Figure 1b, the agent may operate on nonexistent elements due to a lack of domain-specific knowledge, or select incorrect elements due to the proba-

*Corresponding author: zenglin.shi@hfut.edu.cn

Findings of the Association for Computational Linguistics: ACL 2026, pages 13072–13082



(a) The task uncertainty in planning.

(b) The task uncertainty in reasoning.

Figure 1: The dual-level uncertainty challenges in complex web tasks.

bilistic nature of LLMs. This issue primarily stems from an overreliance on LLM-generated actions without considering their uncertainty (Zhang et al., 2025; Zhao et al., 2025). Recent studies (Ma et al., 2025) have introduced logits-induced token uncertainty to decouple LLM uncertainty into aleatoric uncertainty (AU) and epistemic uncertainty (EU). However, these approaches focus primarily on discrete tokens, overlooking the semantic meaning of the generated actions.

In this work, we propose **WebUncertainty**, an autonomous web agent designed to address complex tasks requiring dynamic interactions and long-horizon execution by tackling the dual-level uncertainty arising from planning and reasoning. At the planning level, we design a Task Uncertainty-Driven Adaptive Planning Mechanism. Prior to each planning step, an analysis agent evaluates the task uncertainty based on the current state and task progress. Subsequently, a planning agent adaptively selects the appropriate planning mode based on this uncertainty to effectively handle unknown environments. At the reasoning level, we design an Action Uncertainty-Driven Monte Carlo tree search (MCTS) Reasoning Mechanism. During the MCTS expansion phase, a reasoning agent generates multiple candidate actions along with their confidence scores. We introduce the Confidence-induced Action Uncertainty (ConActU) strategy to quantify action uncertainty at both the AU and EU levels. Finally, we optimize the MCTS search process by combining this quantified uncertainty with feedback from an evaluation agent.

Our contributions are summarized as follows:

- We propose WebUncertainty, a novel autonomous web agent framework that addresses dual-level uncertainty in planning and reasoning, achieving robust performance in complex tasks involving dynamic interactions and long-horizon execution.
- We design a Task Uncertainty-Driven Adaptive Planning Mechanism, which adaptively switches planning modes based on dynamic environmental changes, enabling the system to effectively align sub-goals with unpredictable web environments.
- We introduce an Action Uncertainty-Driven MCTS Reasoning Mechanism, incorporating the ConActU strategy that quantifies both AU and EU to guide the search process, thereby mitigating hallucinations and ensuring reliable decision-making.

Experimental results on WebArena(Zhou et al., 2024) and WebVoyager(He et al., 2024) demonstrate that our WebUncertainty achieves superior performance, particularly for complex tasks, outperforming existing methods.[1]

[1]Code is available at: https://github.com/windbd/WebUncertainty

2 Related Work

Web Agents A web agent is an autonomous AI system that perceives web interfaces through Document Object Model (DOM) trees or screenshots, makes decisions, and executes actions to follow natural language instructions (Gur et al., 2024; Nguyen et al., 2025; Ning et al., 2025). Early approaches primarily relied on rule-based systems or imitation learning, which required extensive human demonstration and were brittle to interface changes (Liu et al., 2018; Pu et al., 2023). The emergence of LLMs has revolutionized this field (Deng et al., 2024). By leveraging their powerful natural language understanding and generation capabilities, modern agents generalize across diverse websites (Song et al., 2025; Lai et al., 2025; Gupta et al., 2026; Zhang et al., 2026b). However, deploying these agents in real-world scenarios remains challenging due to the dynamic nature of web environments and the complexity of long-horizon interactions (Huang et al., 2025; He et al., 2025).

Planning Mechanisms in Agents Planning serves as the strategic core of web agents, responsible for decomposing high-level instructions into executable subgoals (Zhang et al., 2024; Xi et al., 2025; Shahnovsky and Dror, 2026). Existing planning strategies are generally categorized into: 1) explicit planning, which involves formal task decomposition (Li et al., 2023; Niu et al., 2024; Zheng et al., 2024), and 2) implicit planning, where agents predict actions reactively without a formal decomposition phase (Koh et al., 2024; He et al., 2025; Zhang et al., 2025). One-shot explicit planning generates a complete sequence of actions upfront but lacks adaptability; for instance, pregenerated plans quickly become obsolete if the web environment shifts, such as when a pop-up appears. Iterative approaches address this via replanning at fixed steps, yet these methods typically employ rigid protocols without assessing the necessity of such adjustments. Crucially, current approaches fail to model Task Uncertainty (Ning et al., 2025). They do not dynamically adapt their planning mode between explicit and implicit planning based on the agent's familiarity with the environment, leading to either inefficiency in simple tasks or failure in complex, unknown domains (Zhou et al., 2024; He et al., 2024).

Reasoning Mechanisms in Agents Reasoning serves as the decision-making core of web agents, translating planned subgoals into atomic actions (Pahuja et al., 2025; Wei et al., 2026; Zhang et al., 2026a). Existing methods range from reactive reasoning (Abuelsaad et al., 2024; Yang et al., 2025a) to strategic reasoning that employs tree search to explore trajectories (Koh et al., 2024; Yu et al., 2025; Zhang et al., 2025). Crucially, most reasoning mechanisms overlook the risk of hallucinations, allowing execution errors, such as operating on nonexistent elements, to propagate through long-horizon tasks and lead to cascading failures (Gan et al., 2025; Zhao et al., 2025). While Ma et al. (2025) disentangled AU and EU using logits to identify hallucinations, their approach remains confined to discrete tokens and overlooks action semantics. WebUncertainty addresses this gap by incorporating the ConActU strategy into MCTS, explicitly quantifying these uncertainty dimensions at the action level to ensure semantically grounded decision-making.

3 Methodology

As illustrated in Figure 2, we propose WebUncertainty, a hierarchical framework that tackles dual-level uncertainty for web agents. The framework consists of two core components: (1) A Task Uncertainty-Driven Adaptive Planning Mechanism (Section 3.1). In this stage, an Analysis Agent evaluates task uncertainty based on the environment and task progress. A Planning Agent then adaptively switches planning modes to ensure that subgoals align with the evolving webpage state. (2) An Action Uncertainty-Driven MCTS Reasoning Mechanism (Section 3.2). A Reasoning Agent integrates the ConActU strategy to quantify both AU and EU. An Evaluation Agent then assesses action scores to mitigate hallucinations and guide robust execution.

Formally, we model the web navigation task as a Partially Observable Markov Decision Process (POMDP). Given a global instruction I and a webpage observation O_t, the agent operates hierarchically to generate an atomic action $a_t = (e, o, v)$ at each step. Here, e denotes the interactive element, o specifies the operation type (e.g., click or type), and v represents the optional value. The objective is to generate an optimal action sequence that maximizes the success probability of the instruction I.

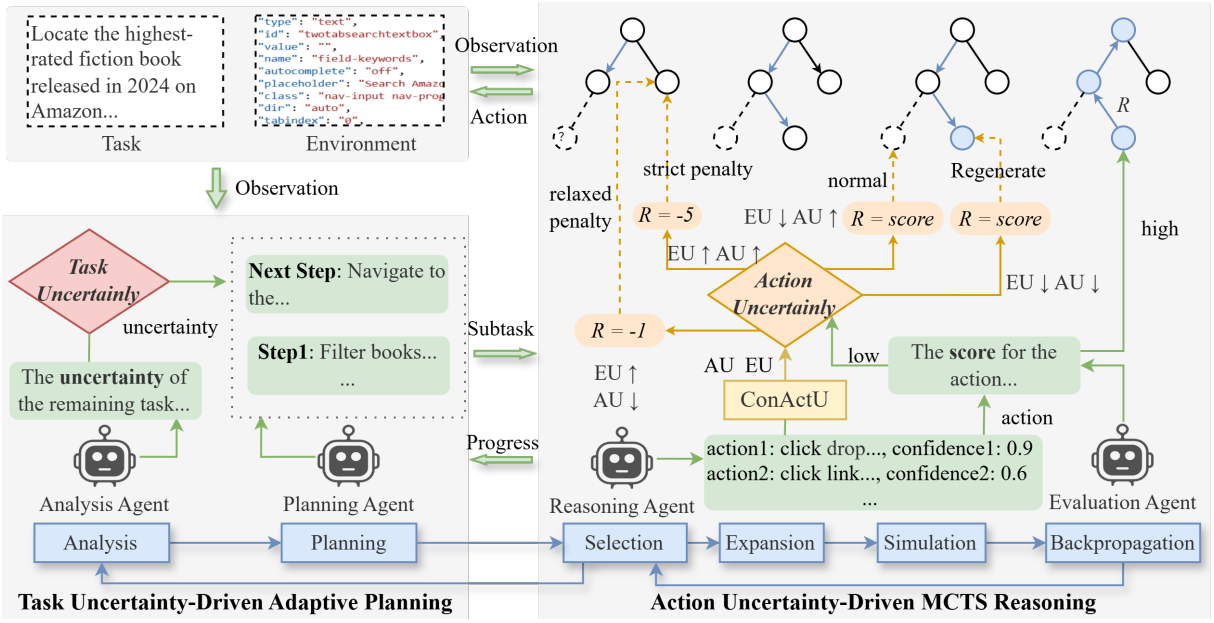


Figure 2: Overview of WebUncertainty. The framework decouples the process into Task Uncertainty-Driven Adaptive Planning (left) and Action Uncertainty-Driven MCTS Reasoning (right).

3.1 Task Uncertainty-Driven Adaptive Planning Mechanism

Static planning strategies often fail in complex web tasks. Explicit decomposition struggles with unknown environments, while implicit stepping risks falling into local optima. To address these issues, we propose the Task Uncertainty-Driven Adaptive Planning Mechanism. Before each planning step, an Analysis Agent evaluates task uncertainty based on the current webpage state and task progress. A Planning Agent then dynamically selects the optimal planning mode. It leverages implicit planning to adapt to unpredicted changes when uncertainty is high. Conversely, it employs explicit planning to maintain global coherence when uncertainty is low. This adaptive approach ensures that generated subgoals effectively align with the evolving web environment.

Task Uncertainty Analysis At each time step t, an Analysis Agent (π_{ana}) evaluates the task before plan generation. It processes the global instruction I, the current webpage observation O_t, and the execution history H_t. Its primary objective is to determine the remaining task objectives T_{rem} and quantify the associated task uncertainty $u_{\text{plan}} \in [0, 1]$, formulated as:

$$T_{\text{rem}}, u_{\text{plan}} = \pi_{\text{ana}}(I, O_t, H_t) \quad (1)$$

Here, T_{rem} represents the pending goals. The scalar u_{plan} links environmental unfamiliarity to execution complexity. A high u_{plan} indicates an unfamiliar webpage structure where achieving T_{rem} is highly complex. Conversely, a low u_{plan} suggests a familiar environment with minimal task complexity.

Adaptive Task Planning Guided by the task uncertainty u_{plan}, the Planning Agent (π_{plan}) selects a subgoal generation strategy based on a threshold δ. In low-uncertainty scenarios ($u_{\text{plan}} \leq \delta$), the agent activates the Explicit Planner ($\pi_{\text{plan}}^{\text{exp}}$) to perform one-shot decomposition. It then commits to the first subgoal in the generated sequence, formulated as $g_t = \text{First}(\pi_{\text{plan}}^{\text{exp}}(T_{\text{rem}}, O_t))$, to ensure long-horizon coherence.

Conversely, in high-uncertainty scenarios ($u_{\text{plan}} > \delta$), the agent shifts to the Implicit Planner ($\pi_{\text{plan}}^{\text{imp}}$) to adapt flexibly to unpredicted environmental dynamics. In this mode, the agent directly predicts the immediate subgoal as $g_t = \pi_{\text{plan}}^{\text{imp}}(T_{\text{rem}}, O_t)$. The resulting subgoal g_t then directs the subsequent reasoning phase. As execution proceeds, updated observations and task progress may reduce uncertainty, enabling a dynamic switch from implicit exploration back to explicit execution.

3.2 Action Uncertainty-Driven MCTS Reasoning Mechanism

After planning, the agent resolves the atomic subgoal g_t during the execution phase. We model this process as a tree search where nodes repre-

sent webpage states and edges denote concrete actions. To navigate vast action spaces and mitigate hallucinations, we propose the Action Uncertainty-Driven MCTS Reasoning Mechanism. This module employs the ConActU strategy to guide the four phases of MCTS:

Selection The agent traverses the tree from the root. At each step, it selects the child node that maximizes the predictor-corrected upper confidence bound (PUCT). We integrate the action confidence from the ConActU strategy as a prior to guide the search:

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left[Q(s,a) + U(s,a) \right] \tag{2}$$

$$U(s,a) = w_{\text{puct}} \cdot \frac{P_{\text{con}}(s,a) \sqrt{\sum_b N(s,b)}}{1 + N(s,a)} \tag{3}$$

Here, $Q(s,a)$ is the value estimate and $N(s,a)$ is the visit count. $P_{\text{con}}(s,a)$ represents the confidence score computed during expansion. This mechanism ensures that the search prioritizes actions with higher evidential support.

Expansion Upon reaching a leaf node, the reasoning agent generates K candidate actions and directly outputs their corresponding confidence scores $\mathbf{c} = [c_1, c_2, \ldots, c_K]$. To quantify uncertainty, we employ the ConActU strategy. First, we normalize the scores into a pseudo-probability distribution $p_i = c_i / \sum_{j=1}^{K} c_j$. We then compute the average confidence as a total evidence proxy $E = \frac{1}{K} \sum_{i=1}^{K} c_i$. To measure the competition among candidates, we calculate the normalized predictive entropy $H_{\text{norm}} = -\frac{1}{\log K} \sum_{i=1}^{K} p_i \log p_i$. Based on these metrics, we formulate EU and AU as follows:

$$\text{EU} = 1 - E \tag{4}$$

$$\text{AU} = H_{\text{norm}} \cdot E \tag{5}$$

In this formulation, EU captures the hallucination risk derived from a lack of overall confidence. Conversely, AU isolates the inherent ambiguity that occurs when the model possesses knowledge (high E) but faces competing valid options (high H_{norm}). Finally, all candidate actions are added to the search tree with their prior probability set to $P_{\text{con}} = p_i$.

Simulation Instead of random rollouts, an evaluation agent assesses the potential of the new state to yield a base feasibility score S_{base}. If the score indicates success ($S_{\text{base}} \geq \tau$), the action is accepted, and we directly assign the reward $R = S_{\text{base}}$. For low scores ($S_{\text{base}} < \tau$), we employ an uncertainty-aware modulation strategy to process the failure. The handling method and exploratory purpose for each condition are defined as follows:

1. High EU and High AU (Strict Penalty): The state is chaotic and unreliable. We assign a severe penalty ($R = -5$) to strictly prohibit the search from selecting this path in the future.

2. High EU and Low AU (Relaxed Penalty): The agent lacks domain knowledge, implying a hallucination. We assign a standard penalty ($R = -1$) to encourage the search to backtrack and explore the parent's sibling nodes.

3. Low EU and High AU (Normal): The agent possesses knowledge but faces stochastic ambiguity. We retain the base score ($R = S_{\text{base}}$) to encourage the search to select different candidate actions under the same node.

4. Low EU and Low AU (Regenerate): The agent is confident, but the execution yields a low score. This indicates a deterministic error. We assign a zero reward ($R = 0$) to trigger the agent to regenerate new actions based on the current node.

Backpropagation Finally, the modulated reward R is backpropagated to update the statistics of all ancestor nodes along the trajectory. We employ an iterative mean update rule to ensure value stability:

$$N(s,a) \leftarrow N(s,a) + 1 \tag{6}$$

$$Q(s,a) \leftarrow Q(s,a) + \frac{R - Q(s,a)}{N(s,a)} \tag{7}$$

This uncertainty-aware update ensures the MCTS converges to a robust policy that avoids epistemic ignorance while managing aleatoric ambiguity.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate WebUncertainty on two benchmarks designed for complex, long-horizon web tasks. WebArena (Zhou et al., 2024) serves as the primary simulation environment. It comprises 812 tasks derived from realistic platforms, such as GitLab and Reddit. Following Zhang et al. (2025), we adopt the text-only setting based on accessibility trees to focus on semantic reasoning. For live web evaluation, we utilize WebVoyager (He et al.,

2024). To ensure reproducibility and objectivity, we employ a curated subset of 129 tasks across 13 diverse environments, including Amazon and Google Maps. We strictly exclude unstable pages and open-ended questions to focus on deterministic outcomes.

Metrics We report **Success Rate** (SR) as the primary metric for functional correctness across all experiments.

Compared Baselines To evaluate WebUncertainty, we compare it against four state-of-the-art agents representing distinct paradigms. Browser Use[2] serves as a fundamental baseline for standard web automation. Agent-E (Abuelsaad et al., 2024) benchmarks our task uncertainty-driven planning against conventional hierarchical architectures. WebPilot (Zhang et al., 2025) utilizes MCTS, providing a direct comparison for our action uncertainty-driven strategy. Finally, AgentOccam (Yang et al., 2025a) evaluates the agent's robustness in observation-action alignment.

Implementation Details To ensure a fair comparison and assess generalizability, we conduct all experiments using two distinct LLM backbones: Qwen-Max-2025-01-25 and GPT-4-turbo-2024-04-09. We execute WebUncertainty and all baselines independently on each backbone. This setup disentangles architectural contributions from the underlying model capabilities. For both LLMs, we fix the temperature at 0.3. In the MCTS reasoning module, we set the maximum node expansion limit to 10 per subgoal and the exploration weight w_{puct} to 5. These settings balance exploration breadth with exploitation efficiency.

4.2 Results on WebArena

Table 1 presents the comparative analysis on the WebArena benchmark. WebUncertainty establishes a new state-of-the-art. It achieves an overall SR of 46.9% with GPT-4-Turbo. This performance surpasses the strong baseline AgentOccam (43.1%) and significantly outperforms the search-based competitor WebPilot (37.6%). These results empirically validate our dual-level uncertainty framework. It effectively mitigates the rigid planning and hallucination issues that hinder conventional agents in complex, long-horizon tasks.

[2] https://github.com/browser-use/browser-use

Adaptability in High-Uncertainty Domains Disaggregated analysis reveals that WebUncertainty excels in domains with high ambiguity and interaction complexity. The Reddit domain involves dense textual content and ambiguous user intents. Here, our agent achieves a 67.0% SR. It surpasses AgentOccam (61.3%) and nearly doubles WebPilot's performance (37.7%). This gain is attributed to the Action Uncertainty-Driven MCTS Reasoning Mechanism. By quantifying AU, the agent identifies ambiguous states with multiple plausible actions (High AU). It then prioritizes exploration over premature commitment to avoid local optima.

The GitLab domain requires precise execution of long-horizon workflows. In this domain, our method achieves a 40.0% SR, compared to WebPilot's 33.3%. This improvement validates the Task Uncertainty-Driven Adaptive Planning Mechanism. The agent dynamically switches between explicit decomposition for global coherence and implicit stepping for unexpected environmental states. This ensures robust navigation in technical environments.

Robustness Across Reasoning Backbones To assess architectural generalizability, we evaluate performance using Qwen-Max. As shown in the bottom section of Table 1, WebUncertainty maintains its lead with an overall SR of 40.1%. It outperforms AgentOccam (38.4%) and WebPilot (34.5%).

Notably, our framework powered by Qwen-Max outperforms the GPT-4-Turbo version of WebPilot (37.6%). This result underscores the efficacy of the ConActU strategy. By explicitly quantifying EU, our framework enables weaker models to detect their own knowledge boundaries. They can then prune hallucinated actions (High EU) before execution. This uncertainty-aware filtering effectively compensates for the lower intrinsic reasoning capability of the backbone model. It prevents the snowball effect of errors common in long-horizon tasks.

4.3 Results on WebVoyager

We extend our evaluation to WebVoyager to assess robustness in live, open-domain web environments. Unlike the controlled simulation of WebArena, WebVoyager involves real-world websites, such as Amazon and Google Maps. These sites feature dynamic content loading, complex DOM structures,

Agent	Backbone	SR (%)	Shop	Admin	GitLab	Map	Reddit	Multi
WebArena-rep	GPT-4-Turbo	16.5	16.6	15.9	10.0	22.9	21.7	16.7
Browser Use	GPT-4-Turbo	16.9	15.0	17.6	10.6	23.9	23.6	14.6
Agent-E	GPT-4-Turbo	13.9	13.4	10.4	11.1	19.3	20.8	12.5
WebPilot	GPT-4-Turbo	37.6	41.2	43.4	33.3	37.6	37.7	16.7
AgentOccam	GPT-4-Turbo	43.1	40.6	45.6	37.8	**46.8**	61.3	14.6
WebUncertainty	GPT-4-Turbo	**46.9**	**47.6**	**49.5**	**40.0**	45.9	**67.0**	**18.8**
Agent-E	Qwen-Max	14.2	12.3	9.9	12.8	17.4	23.6	14.6
WebPilot	Qwen-Max	34.5	40.1	33.0	29.4	**40.4**	36.8	**18.8**
AgentOccam	Qwen-Max	38.4	41.2	**42.9**	33.3	27.5	55.7	16.7
WebUncertainty	Qwen-Max	**40.1**	**42.8**	38.5	**37.8**	38.5	**57.5**	10.4

Table 1: Performance comparison on WebArena. The SR is reported over 812 tasks across domains: Shopping, Shopping Admin, GitLab, Map, Reddit, and Multisite. The best results for each backbone group are highlighted in **bold**.

Agent	Backbone	SR (%)
WebVoyager-rep	GPT-4-Turbo	50.4
Browser Use	GPT-4-Turbo	51.9
Agent-E	GPT-4-Turbo	59.7
WebPilot	GPT-4-Turbo	62.0
AgentOccam	GPT-4-Turbo	64.3
WebUncertainty	GPT-4-Turbo	**65.9**
WebVoyager-rep	Qwen-Max	46.5
Browser Use	Qwen-Max	48.8
Agent-E	Qwen-Max	54.3
WebPilot	Qwen-Max	55.8
AgentOccam	Qwen-Max	58.9
WebUncertainty	Qwen-Max	**63.6**

Table 2: SR comparison on the WebVoyager benchmark. The evaluation is conducted on a curated subset of 129 tasks involving real-world websites with deterministic outcomes. Best results for each backbone are highlighted in **bold**.

and potential network latency.

Robustness in Dynamic Real-World Settings As detailed in Table 2, WebUncertainty consistently achieves the highest SR across both backbone models. With GPT-4-Turbo, our method attains a 65.9% SR. It outperforms the strongest baseline AgentOccam (64.3%) and the search-based WebPilot (62.0%). AgentOccam enhances performance by optimizing observation grounding. However, it often struggles to recover from execution failures caused by unpredicted interface changes, such as pop-ups or layout shifts. Our framework addresses this limitation through the Action Uncertainty-Driven MCTS Reasoning Mechanism. The ConActU strategy distinguishes between epistemic hallucinations and aleatoric environmental noise. It penalizes high-risk paths and encourages the exploration of alternative actions during confident but unsuccessful executions (Low EU, High AU).

Efficiency on Weaker Backbones Results on the Qwen-Max backbone demonstrate the architectural efficiency of our approach. WebUncertainty achieves a 63.6% SR, outperforming AgentOccam (58.9%) and WebPilot (55.8%) by a substantial margin. Notably, our framework powered by the weaker Qwen-Max model outperforms the GPT-4-Turbo version of WebPilot (63.6% vs. 62.0%). This highlights a critical insight. In complex web navigation, raw LLM reasoning capability faces diminishing returns without effective uncertainty management. Our framework models Task Uncertainty to adaptively switch planning modes. It also uses Action Uncertainty to prune search trees. This dual-level strategy empowers weaker models to achieve performance levels comparable to, or exceeding, stronger models that rely on standard architectures.

4.4 Ablation Studies

To disentangle the contributions of individual components within our framework, we conduct ablation studies on both WebVoyager and WebArena benchmarks using the Qwen-Max backbone. We introduce three variants to strictly isolate the efficacy of the Task Uncertainty-Driven Adaptive

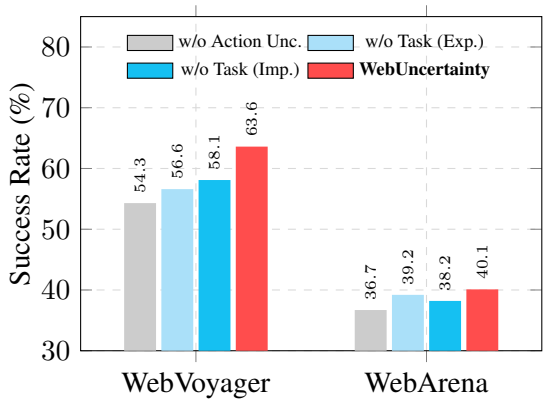


Figure 3: Ablation study using the Qwen-Max backbone.

Planning Mechanism and the Action Uncertainty-Driven MCTS Reasoning Mechanism. The comparative results are visualized in Figure 3.

Impact of Task Uncertainty-Driven Planning We analyze the necessity of the adaptive planning mechanism by freezing the agent into static explicit-only or implicit-only modes (blue bars in Figure 3). The results reveal a distinct domain-dependent preference. On WebArena, the explicit-only mode outperforms the implicit-only mode (39.2% vs. 38.2%). The implicit mode struggles to maintain the global thread in deep, structured workflows. Conversely, on WebVoyager, the implicit-only mode surpasses the explicit-only mode (58.1% vs. 56.6%). Rigid plans generated by the explicit mode often become obsolete due to high environmental volatility. Crucially, the full WebUncertainty framework consistently achieves the highest performance (63.6% and 40.1%). This confirms that task uncertainty effectively signals when to switch between explicit decomposition for stability and reactive stepping for flexibility.

Impact of Action Uncertainty-Driven Reasoning The w/o Action Unc. variant (gray bar) removes the ConActU strategy. This reverts the execution phase to standard MCTS and causes the most significant performance degradation. The SR drops by 9.3% on WebVoyager and 3.4% on WebArena. The critical flaw of the standard MCTS baseline lies in its inability to decouple error sources. Without EU quantification, the agent cannot identify hallucinations, often wasting search budget expanding nodes on nonexistent elements. Simultaneously, without AU awareness, it treats ambiguous states with multiple valid actions as failures. The agent prunes promising branches instead of triggering necessary exploration. The superior performance of WebUncertainty proves that distinguishing chaotic states from confident failures is essential for robust decision-making.

4.5 Performance-Cost Analysis

MCTS-based reasoning increases computational overhead. However, this is a deliberate trade-off to ensure robustness in complex web tasks. In these scenarios, the cost of a single execution error significantly outweighs the inference cost.

Importantly, our framework optimizes the MCTS process. It achieves higher performance with lower computational costs than existing search-based methods. As quantified in Table 3, we evaluate the average inference time per task on WebVoyager using the Qwen-Max backbone. WebUncertainty reduces the average inference time by over 56% compared to WebPilot (351.4s vs. 803.7s). It simultaneously improves the SR from 55.8% to 63.6%. Future deployments will explore a more systematic performance-cost analysis, including average token usage and total inference cost, to further demonstrate the framework's real-world practicality.

Agent	SR (%)	Avg. Time (s)
WebVoyager-rep	46.5	204.9
Browser Use	48.8	264.6
Agent-E	54.3	224.7
WebPilot	55.8	803.7
AgentOccam	58.9	306.5
WebUncertainty	**63.6**	**351.4**

Table 3: SR and Average Inference Time comparison on WebVoyager using the Qwen-Max backbone.

4.6 Sensitivity Analysis

To assess the robustness of our framework, we conduct a sensitivity analysis on the planning switch threshold δ and the evaluation threshold τ. The threshold δ balances long-horizon coherence from explicit planning with reactive flexibility from implicit planning.

As shown in Table 4, our framework demonstrates strong robustness. It consistently exceeds the strongest baseline (AgentOccam at 58.9%) across a wide range of values. The framework achieves the optimal SR at $\delta = 0.4$ and $\tau = 6$.

Threshold	Values & SR (%)					
δ	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
SR (%)	58.1	59.7	**63.6**	62.8	58.9	56.6
τ	**0**	**2**	**4**	**6**	**8**	**10**
SR (%)	52.7	57.4	62.0	**63.6**	61.2	0.0

Table 4: Sensitivity analysis of hyperparameters δ and τ on WebVoyager (Qwen-Max).

5 Conclusion

In this work, we presented WebUncertainty, an autonomous agent framework that tackles dynamic interactions and long-horizon execution by modeling dual-level uncertainty. Our Task Uncertainty-Driven Adaptive Planning Mechanism adaptively switches planning modes to ensure robust goal alignment. Furthermore, our Action Uncertainty-Driven MCTS Reasoning Mechanism leverages the ConActU strategy to prune hallucinations and guide decision-making. Extensive experiments on WebArena and WebVoyager demonstrate that WebUncertainty achieves state-of-the-art performance. These results validate the efficacy of integrating uncertainty awareness into the planning and reasoning of web agents.

Limitations

Despite its promising performance, WebUncertainty presents several limitations. First, MCTS and multiple candidate generation introduce computational overhead. Although our framework reduces inference time by 56% compared to WebPilot, this trade-off for robustness may still hinder deployment in real-time or low-cost applications.

Second, our text-only implementation relies on accessibility trees. The agent may therefore struggle with visually intensive websites where critical information is conveyed through spatial layouts or color coding rather than semantic text.

Finally, the framework depends on empirical hyperparameters (the thresholds δ and τ) and the intrinsic calibration of the backbone LLMs. While generally robust, rigid settings may cause suboptimal mode switching in highly volatile environments. Future work will explore adaptive tuning strategies to reduce this dependence.

Ethics Statement

This research involves autonomous agents interacting with live web environments. We ensured that all automated interactions were strictly for benign academic purposes, intentionally avoiding malicious actions, unauthorized data collection, or server disruption. Furthermore, as our framework relies on large language models, we acknowledge the inherent risks of propagated biases and hallucinated actions. We strongly advocate for human-in-the-loop oversight before deploying such autonomous agents in critical real-world applications to prevent unintended consequences.

Acknowledgments

This paper is funded by National Natural Science Foundation of China (No. 62472138).

References

Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. 2024. Agent-E: from autonomous web navigation to foundational design principles in agentic systems. *arXiv preprint*. ArXiv:2407.13032 [cs].

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. MIND2WEB: towards a generalist agent for the web. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 28091–28114, Red Hook, NY, USA. Curran Associates Inc.

Yang Deng, An Zhang, Yankai Lin, Xu Chen, Ji-Rong Wen, and Tat-Seng Chua. 2024. Large language model powered agents in the web. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pages 1242–1245, New York, NY, USA. Association for Computing Machinery.

Shangheng Du, Jiabao Zhao, Jinxin Shi, Zhentao Xie, Xin Jiang, Yanhong Bai, and Liang He. 2026. A survey on the optimization of large language model-based agents. *ACM Computing Surveys*, 58(9):1–37.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. PLAN-AND-ACT: improving planning of agents for long-horizon tasks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *ICML'25*, pages 15419–15462, Vancouver, Canada. JMLR.org.

Zeyu Gan, Yun Liao, and Yong Liu. 2025. Rethinking external slow-thinking: from snowball errors to probability of correct reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*,

pages 18170–18188. PMLR. ShortConferenceName: ICML.

Tanmay Gupta, Piper Wolters, Zixian Ma, Peter Sushko, Rock Yuren Pang, Diego Llanes, Yue Yang, Taira Anderson, Boyuan Zheng, Zhongzheng Ren, Harsh Trivedi, Taylor Blanton, Caleb Ouellette, Winson Han, Ali Farhadi, and Ranjay Krishna. 2026. MolmoWeb: open visual web agent and open data for the open web. *arXiv preprint*. ArXiv:2604.08516 [cs].

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. A real-world WebAgent with planning, long context understanding, and program synthesis. In *International Conference on Learning Representations*, volume 2024, pages 52690–52717.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Hongming Zhang, Tianqing Fang, Zhenzhong Lan, and Dong Yu. 2025. OpenWebVoyager: building multimodal web agents via iterative real-world exploration, feedback and optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27545–27564, Vienna, Austria. Association for Computational Linguistics.

Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, Yuhuai Li, Shengze Xu, Shenzhi Wang, Xinchen Xu, Shuofei Qiao, Zhaokai Wang, Kun Kuang, Tieyong Zeng, Liang Wang, and 10 others. 2025. OS agents: a survey on MLLM-based agents for computer, phone and browser use. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7436–7465, Vienna, Austria. Association for Computational Linguistics.

Tenghao Huang, Kinjal Basu, Ibrahim Abdelaziz, Pavan Kapanipathi, Jonathan May, and Muhao Chen. 2025. R2D2: remembering, replaying and dynamic decision making with a reflective agentic memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30318–30330, Vienna, Austria. Association for Computational Linguistics.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024. Tree search for language model agents. *arXiv preprint*. Version Number: 4.

Hanyu Lai, Xiao Liu, Hao Yu, Yifan Xu, Iat Long Iong, Shuntian Yao, Aohan Zeng, Zhengxiao Du, Yuxiao Dong, and Jie Tang. 2025. WebGLM: towards an efficient and reliable web-enhanced question-answering system. *ACM Transactions on Information Systems*, 43(5):1–43.

Tao Li, Gang Li, Zhiwei Deng, Bryan Wang, and Yang Li. 2023. A zero-shot language agent for computer control with structured reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11261–11274, Singapore. Association for Computational Linguistics.

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

Haohao Luo, Jiayi Kuang, Wei Liu, Ying Shen, Jian Luan, and Yang Deng. 2025. Browsing like human: a multimodal web agent with experiential fast-and-slow thinking. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14232–14251, Vienna, Austria. Association for Computational Linguistics.

Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. 2025. Estimating LLM uncertainty with evidence. *arXiv preprint*. ArXiv:2502.00290 [cs].

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, and 11 others. 2025. GUI agents: a survey. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22522–22538, Vienna, Austria. Association for Computational Linguistics.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S. Yu, and Qing Li. 2025. A survey of WebAgents: towards next-generation AI agents for web automation with large foundation models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 6140–6150, Toronto ON Canada. ACM.

Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. 2024. ScreenAgent: a vision language model-driven computer control agent. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 6433–6441. International Joint Conferences on Artificial Intelligence Organization.

Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Hassan Awadallah. 2025. Explorer: scaling exploration-driven web trajectory synthesis for multimodal web agents. In *Findings of the Association for Computational Linguistics: ACL 2025*,

pages 6300–6323, Vienna, Austria. Association for Computational Linguistics.

Kevin Pu, Jim Yang, Angel Yuan, Minyi Ma, Rui Dong, Xinyu Wang, Yan Chen, and Tovi Grossman. 2023. DiLogics: creating web automation programs with diverse logics. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–15, San Francisco CA USA. ACM.

Orit Shahnovsky and Rotem Dror. 2026. AI planning framework for LLM-based web agents. *arXiv preprint*. Version Number: 1.

Yueqi Song, Frank F. Xu, Shuyan Zhou, and Graham Neubig. 2025. Beyond browsing: API-based web agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11066–11085, Vienna, Austria. Association for Computational Linguistics.

Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, Zihao Li, Mengting Ai, Duo Zhou, Wenxuan Bao, Yunzhe Li, Gaotang Li, Cheng Qian, Yu Wang, Xiangru Tang, and 10 others. 2026. Agentic reasoning for large language models. *arXiv preprint*. ArXiv:2601.12538 [cs].

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. WebWalker: benchmarking LLMs in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10290–10305, Vienna, Austria. Association for Computational Linguistics.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, and 9 others. 2025. The rise and potential of large language model based agents: a survey. *Science China Information Sciences*, 68(2):121101.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. 2025. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21381–21396, Vienna, Austria. Association for Computational Linguistics.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2025a. AgentOccam: a simple yet strong baseline for LLM-based web agents. In *The Thirteenth International Conference on Learning Representations*.

Yingxuan Yang, Mulei Ma, Yuxuan Huang, Huacan Chai, Chenyu Gong, Haoran Geng, Yuanjian Zhou, Ying Wen, Meng Fang, Muhao Chen, Shangding Gu, Ming Jin, Costas Spanos, Yang Yang, Pieter Abbeel, Dawn Song, Weinan Zhang, and Jun Wang. 2025b. Agentic web: weaving the next web with AI agents. *arXiv preprint*. Version Number: 1.

Xiao Yu, Baolin Peng, Vineeth Vajipey, Hao Cheng, Michel Galley, Jianfeng Gao, and Zhou Yu. 2025. ExACT: teaching AI agents to explore with reflective-MCTS and exploratory learning. In *International Conference on Learning Representations*, volume 2025, pages 65157–65184.

Duzhen Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2026a. From system 1 to system 2: a survey of reasoning large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3335–3354.

Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. Ask-before-plan: proactive language agents for real-world planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10836–10863, Miami, Florida, USA. Association for Computational Linguistics.

Xuanwang Zhang, Yuteng Han, Jinnan Qi, Mulong Xie, Zhen Wu, and Xinyu Dai. 2026b. WebNavigator: global web navigation via interaction graph retrieval. *arXiv preprint*. Version Number: 1.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2025. WebPilot: a versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, volume 39 of *AAAI'25/IAAI'25/EAAI'25*, pages 23378–23386. AAAI Press.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, Haifeng Chen, and Xujiang Zhao. 2025. Uncertainty propagation on LLM agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6073, Vienna, Austria. Association for Computational Linguistics.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. GPT-4V(ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61349–61385. PMLR.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: a realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.