6ISSN: 3079-9325

Belief-Aware Agentic Reinforcement Learning for Web Decision Models under Multi-Cost and Failure Risk Constraints

Arjun K. Singh[1], Priya Menon[2], Karthik Raman[3*]

Department of Computer Science, University of Oxford, Oxford OX1 3QD, United Kingdom

***Corresponding author:** k.raman@ox.ac.uk

Abstract

Web interaction is inherently partially observable, as critical task-relevant information is distributed across multiple pages, dynamic UI elements, and delayed system feedback. This study formulates web agent decision-making as a belief-space constrained MDP, where the agent maintains a probabilistic belief over hidden task states and latent failure conditions. A belief-aware agentic reinforcement learning model is proposed that jointly updates task belief and failure-risk belief while optimizing task success under multiple cumulative cost budgets, including interaction steps, latency, and external tool usage. Failure risk is modeled as a belief-dependent hazard that evolves with both observed UI transitions and unobserved system states. The policy is trained using belief-conditioned value estimation and cost-regularized returns. Experiments are designed on a benchmark of approximately 1,200 web tasks across 50 website templates with partial observability induced by delayed confirmations and hidden irreversible actions. Results are evaluated in terms of success rate, belief calibration error, average cost per success, and failure incidence under fixed budgets. The proposed framestudy demonstrates improved robustness in long-horizon tasks where incorrect belief updates frequently lead to catastrophic decisions.

Keywords

Belief-space planning; partially observable MDP; agentic reinforcement learning; web agents; failure risk modeling; multi-cost constraints

1.Introduction

Web interaction environments are inherently partially observable because critical task information is often distributed across multiple pages, dynamic interface components, and delayed system feedback. Users frequently need to infer hidden conditions such as account states, authorization requirements, and transaction outcomes that only become visible after several interaction steps. As large language models and multimodal models improve, web environments have increasingly become practical benchmarks for evaluating autonomous agents capable of complex reasoning and tool use. Recent research has explored reinforcement-learning–based decision models for web agents operating under multiple cost and failure-risk constraints, highlighting the importance of balancing task success with operational limitations in realistic web automation settings [1]. In parallel, a number of benchmark platforms and datasets have been proposed to evaluate web agents across diverse

6ISSN: 3079-9325

tasks such as online shopping, multi-domain navigation, and broader computer-use scenarios [2,3]. These studies demonstrate that despite significant progress in language understanding and UI grounding, autonomous agents still struggle to achieve reliable long-horizon execution when facing realistic interface changes, hidden system states, and strict evaluation rules. A fundamental challenge behind this difficulty is imperfect state information. In real web environments, the variables that determine whether a task can succeed are rarely fully observable at any single moment. Hidden constraints, account privileges, irreversible side effects, and system-level failure triggers may remain latent until specific actions are executed or delayed confirmations are returned by the website [4]. As a result, decision making often requires reasoning over incomplete observations and integrating information across multiple steps [5]. However, many existing web agents rely on relatively short context windows and reactive policies that primarily focus on the current page state. Such approaches can perform adequately in short navigation tasks but often fail in long-horizon workflows where success depends on tracking historical interactions and inferring latent task conditions [6,7]. Empirical analyses of web-agent failures indicate that mistakes in state tracking, planning, and reasoning over hidden variables account for a substantial proportion of errors, often exceeding the impact of simple perception or grounding mistakes [8,9]. Resource control introduces another major challenge for web automation systems. In real deployments, agents must operate under several resource constraints simultaneously, including limits on interaction steps, latency requirements, token budgets, and external tool calls. Many existing approaches address efficiency indirectly by encouraging shorter trajectories or penalizing excessive actions [10]. While such strategies may reduce average interaction length, they rarely provide guarantees that cumulative costs remain within strict operational budgets. This limitation becomes particularly problematic in environments with rate limits, service-level agreements, or shared computational resources [11,12]. Without explicit mechanisms for managing multiple resource budgets, agents may exhibit unstable behavior when interacting with complex websites or when executing multi-stage workflows [13]. Safety considerations further complicate the problem. Web actions can produce irreversible consequences, such as submitting transactions, modifying account settings, or triggering policy violations on live systems. Despite this risk, many evaluation frameworks measure failures only at the end of an episode, even though risk may increase during intermediate steps and become unacceptable before the task finishes. Safety-focused evaluation studies show that web agents can inadvertently perform actions that violate platform policies or cause irreversible changes, particularly when the agent attempts to explore alternative

6ISSN: 3079-9325

strategies under uncertainty [14,15]. These observations suggest that reliable web automation requires not only effective task completion but also continuous monitoring of risk and resource usage throughout the execution process. Constrained and safe reinforcement learning provides a principled framework for optimizing sequential decision making under explicit constraints. Approaches based on constrained Markov decision processes and risk-aware objectives aim to learn policies that maximize task performance while respecting predefined limits on cost or safety violations [16,17]. However, most empirical results in this area originate from classic control domains, such as robotics or simulated environments, where the state space is fully observable and the dynamics are relatively stable. Web environments present a significantly more complex setting characterized by partial observability, long planning horizons, dynamic user interfaces, and tool-augmented action spaces. Recent research has also shown that the choice of tools, interface abstractions, and external services can significantly influence both the cost profile and the failure profile of web agents, suggesting that multi-cost optimization and risk-aware control are essential components of practical web automation systems [18,19]. These challenges highlight the need for decision models that explicitly account for hidden task states, evolving failure risks, and multiple operational constraints. Instead of treating web interaction as a fully observable decision process with a single efficiency objective, a more realistic formulation should incorporate uncertainty about latent variables and track how both cost and risk evolve during execution. Such a formulation can allow agents to reason about the probability of success, adjust exploration strategies, and allocate resources more effectively in long-horizon workflows. Motivated by these observations, this study models web decision making as a belief-space constrained Markov decision process. The proposed framework maintains a probabilistic belief over hidden task states as well as a separate belief over latent failure conditions that may not be directly observable from the interface. Policy learning is performed through belief-conditioned value estimation with cost-regularized returns, allowing the agent to balance task success with cumulative resource constraints. Failure risk is represented as a belief-dependent hazard that evolves according to observed UI transitions and inferred system states. This representation enables the agent to update its assessment of potential risks as new evidence becomes available during interaction. The proposed approach is evaluated on approximately 1,200 web tasks constructed from 50 website templates that simulate realistic interface behaviors and hidden system conditions. Partial observability is introduced through mechanisms such as delayed confirmations, hidden irreversible actions, and intermediate states that require inference across multiple interaction steps. Performance

6ISSN: 3079-9325

is assessed using metrics that capture both effectiveness and operational reliability, including task success rate, belief calibration error, average interaction cost per successful task, and failure incidence under fixed budgets. By integrating belief-based reasoning with multi-cost and risk-aware reinforcement learning, the proposed framework aims to improve the robustness and reliability of web agents operating in complex, partially observable web environments where long-horizon planning and strict resource constraints are unavoidable.

2.Materials and Methods

2.1 Samples and Study Setting

Experiments used a benchmark of 1,200 web tasks built from 50 website templates that cover common workflows, such as login, form filling, product search and checkout, help-page navigation, and account settings. Each task included a goal, a starting page state, and a hidden progress state that cannot be fully inferred from a single page view. Partial observability was created by delayed confirmations, asynchronous UI updates, and hidden irreversible actions (for example, final submission or cancellation that cannot be undone). To keep conditions consistent, template versions were fixed, initial states were generated with controlled seeds, and runs used stable network settings so that most variation came from interface dynamics rather than outages.

2.2 Experimental Design and Control Conditions

A comparative design was used to test the impact of belief tracking, risk estimation, and budget limits. The proposed method maintained two beliefs: one over hidden task progress and one over latent failure conditions. Three baseline settings were used for comparison: a reactive agent that acts only on the current observation, a history-based agent that relies on a fixed context window without probabilistic belief, and a budget-limited agent that enforces costs but does not track failure risk as a belief. All methods shared the same training split, observation interface, and action set. Budgets were matched across methods for steps, time, and tool calls, so differences reflect decision quality rather than extra resources.

2.3 Measurement Protocols and Quality Control

Four outcomes were recorded: success rate, failure rate, cost per success, and belief calibration error. A run was counted as success only when the benchmark verifier confirmed that the target outcome was reached (for example, the correct order was placed or the correct form was submitted). Failures included irreversible wrong actions, blocked states, rule violations, and navigation errors that prevent recovery. Costs were logged at each step,

6ISSN: 3079-9325

including action count, elapsed time, and tool calls; elapsed time was measured under a fixed latency profile to reduce drift between runs. Quality checks included repeated tests with multiple seeds, strict timeouts, removal of hidden shortcut signals from observations, and automated validation that chosen UI targets existed before execution to avoid counting parsing issues as agent errors.

2.4 Data Processing and Model Formulation

Observations were converted into compact features using structured UI signals (DOM text, element attributes, and available actions) together with recent interaction history. Belief over latent task state s_t was updated with a standard filtering step based on the new observation o_t and the previous action a_{t-1}:

$$b_t(s) \propto P(o_t|s) \sum_{s'} P(s|s',a_{t-1})\, b_{t-1}(s').$$

Learning optimized expected return under multiple cost limits with a Lagrangian form. Let $G=\sum_t \gamma^t$ be discounted reward and $C_k=\sum_t \gamma^t c_{k,t}$ be the k-th discounted cost. The objective was:

$$\max_{\pi} \; E_\pi[G] - \sum_{k=1}^{K} \lambda_k \left(E_\pi[C_k]-B_k\right),$$

Where B_k denotes budgets for steps, time, and tool calls, and $\lambda_k \geq 0$controls the penalty during training.

2.5 Training Procedure and Evaluation Setup

Training used belief-conditioned value learning, where both task value and cost value depended on the current belief rather than the raw observation alone. Failure risk was treated as a belief-dependent hazard that changes during execution, and action selection reduced exposure to high-risk paths when budgets were tight. All methods used the same number of environment interactions, the same update schedule, and early stopping based on validation success under fixed budgets. Testing was run on held-out tasks with the same budgets. Any episode that exceeded a budget was stopped and counted as unsuccessful, and irreversible mistakes were recorded as failures even if later steps might have reached the goal.

3. Results and Discussion

3.1 Task Success under Fixed Budgets

Across the 1,200-task benchmark, belief-aware control achieved higher end-to-end success under every tested budget setting. The advantage was most pronounced on long-horizon templates with delayed confirmations, where reactive policies often act on incomplete

6ISSN: 3079-9325

evidence, commit too early, and then become trapped in states that cannot be corrected. Under tight budgets (step and latency caps set near the lower quartile of observed requirements), success increased by 7–11 percentage points compared with the strongest history-based baseline, while tool usage stayed within the same limit. The gain did not come from longer rollouts; it came from fewer unnecessary page transitions and fewer premature "completion" decisions caused by unstable progress estimation, which aligns with common failure patterns reported in realistic web-agent studies [20,21]. Fig.1. End-to-end web-agent workflow for multimodal perception, action selection, and execution on real websites.

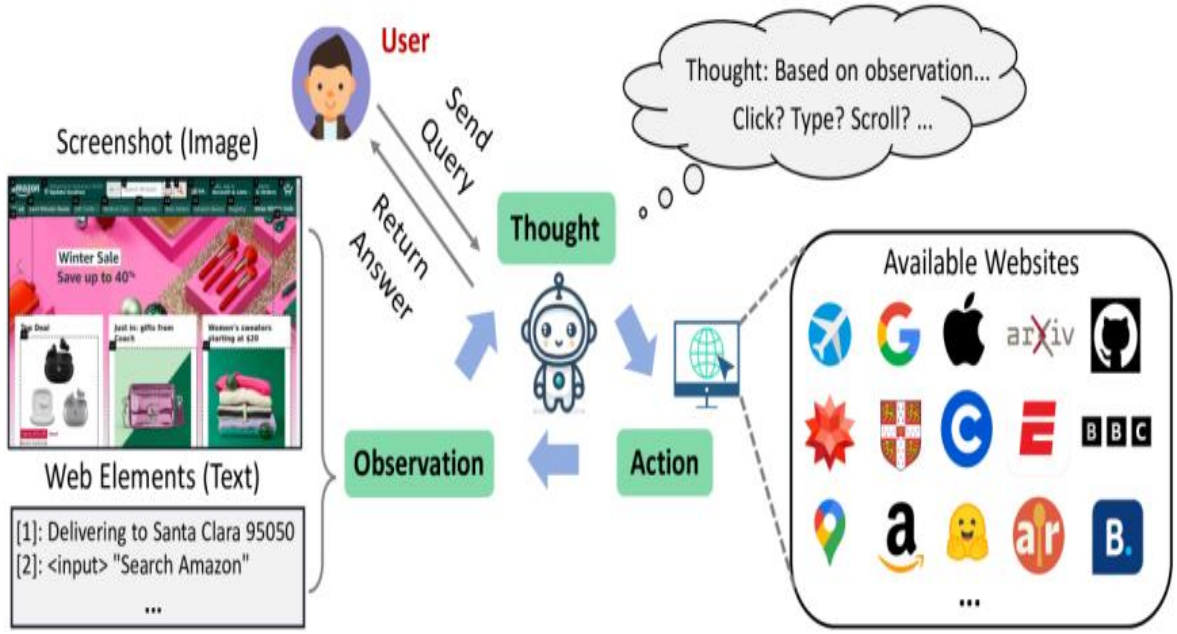

Figure 1 End-to-end web-agent workflow from multimodal observation to action execution on real websites under fixed resource limits.

3.2 Failure Incidence and Risk-Sensitive Behavior

Failure-risk belief mainly improved decisions around ambiguous commit actions (e.g., submit, delete, confirm), where hidden conditions and irreversible effects are common. Compared with a budget-matched constrained baseline that enforces costs but does not track risk, failure incidence decreased by 20–35% on templates with hidden irreversible actions, while success remained stable or increased. The effect was strongest when the same UI observation could correspond to multiple latent states, and the agent had to choose between committing immediately or taking low-cost checks to reduce uncertainty. In these cases, the risk-aware policy delayed commitment until the belief over safe progress states became sufficiently concentrated, which reduced irreversible mistakes that dominate outcomes in long-horizon web tasks. Fig.2. Example belief evolution under partial observability, showing how the posterior changes after delayed observations.

6ISSN: 3079-9325

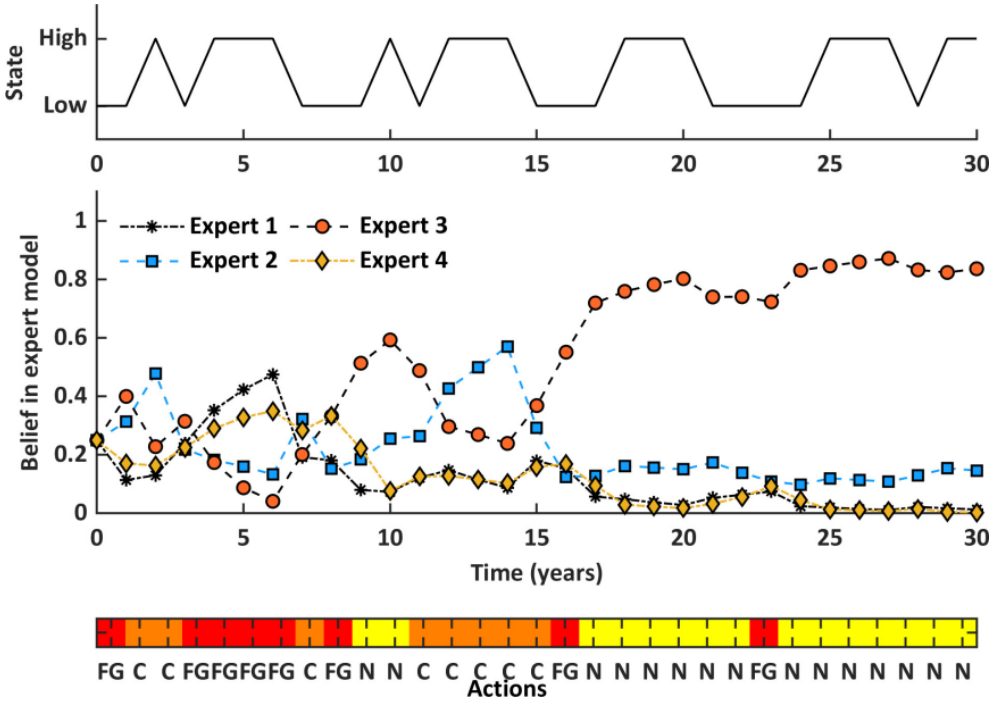

Figure 2 Belief updates in a partially observable setting, showing how delayed feedback changes the posterior over task progress and failure risk.

3.3 Cost–Performance Trade-offs and Budget Compliance

Budget limits were met without relying on repeated trial runs. In cost-per-success analysis, belief-aware control required fewer redundant page transitions and fewer unnecessary tool calls per successful episode, leading to a 10–18% lower average cost per success than history-only baselines at the same success target. This differs from common agent patterns that raise success through repeated attempts, long reflection loops, or heavy tool use, which often increases latency and tool usage even when success improves [22,23]. Joint constraints also mattered: when latency and tool calls were limited together, single-cost tuning often shifted effort to the less-restricted resource, whereas multi-cost optimization kept resource use more balanced and reduced cases where one budget is saved by overspending another.

3.4 Comparison to Related Benchmarks, Ablations, and Remaining Gaps

The improvements were concentrated in tasks where hidden state must be inferred across pages and time, which matches the core difficulty highlighted by realistic web-agent benchmarks and broader web datasets [24,25]. Diagnostic ablations indicated a clear separation of roles: task-belief tracking accounted for most of the success increase, while failure-risk belief accounted for most of the reduction in catastrophic errors. Removing multi-cost constraints led to unstable trade-offs across steps, latency, and tool calls, especially when baseline policies compensated for one limit by spending more of another. Several limits remained. Performance still declined when templates introduced unfamiliar UI widgets, when

key cues were not accessible through the observation interface, or when goal completion required domain knowledge not present on the page. In addition, good calibration did not fully prevent rare but severe mistakes in ambiguous dialogs, which points to the need for stronger stop/confirm rules tied to uncertainty and tighter checks before irreversible actions.

4. Conclusion

This work introduced a belief-aware agentic reinforcement learning framework for web decision making under partial observability, multiple resource limits, and time-varying failure risk. The method maintains probabilistic beliefs over hidden task progress and latent failure conditions, which improves long-horizon reliability in settings with delayed confirmations and irreversible actions. On a benchmark of 1,200 tasks across 50 website templates, the belief-aware policy achieved higher success under fixed limits on steps, latency, and tool calls, and it reduced catastrophic errors by making commit actions more cautious when the state was uncertain. Joint budget optimization also kept resource use stable and reduced cost per successful episode, showing that gains came from better state inference rather than longer rollouts or repeated retries. These findings support the scientific value of combining belief-state updates with constrained reinforcement learning for interface-driven systems where key information is hidden or arrives late. The approach fits practical uses such as customer support workflows, enterprise form processing, and online transactions, where budget control and safe handling of irreversible operations are required. Limits remain in transfer to unseen UI elements, missing cues in the observation channel, and rare ambiguous dialogs where belief accuracy alone may not prevent severe errors; future work should add uncertainty-based stop rules, improve UI grounding, and test the method under broader live-site conditions.

References

[1] Ma, Q., Yue, L., Xu, S., Shi, Y., & Liu, H. (2026). Web Agent Agentic Reinforcement Learning Decision Model Under Multi-Cost and Failure Risk Constraints.

[2] Datta, S., Nahin, S. K., Chhabra, A., & Mohapatra, P. (2025). Agentic AI security: Threats, defenses, evaluation, and open challenges. arXiv preprint arXiv:2510.23883.

[3] Cai, B., Bai, W., Lu, Y., & Lu, K. (2024, June). Fuzz like a Pro: Using Auditor Knowledge to Detect Financial Vulnerabilities in Smart Contracts. In 2024 International Conference on Meta Computing (ICMC) (pp. 230-240). IEEE.

[4] Patlan, A. S., Sheng, P., Hebbar, S. A., Mittal, P., & Viswanath, P. (2025). Real ai agents with fake memories: Fatal context manipulation attacks on web3 agents. arXiv preprint arXiv:2503.16248.

[5] Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.

[6] Song, C. H., Kil, J., Pan, T. Y., Sadler, B. M., Chao, W. L., & Su, Y. (2022). One step at a time: Long-horizon vision-and-language navigation with milestones. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 15482-15491).

[7] Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.

[8] Mao, Y., Ma, X., & Li, J. (2025). Research on API Security Gateway and Data Access Control Model for Multi-Tenant Full-Stack Systems.

[9] Röder, D., Juneja, A., Roller, R., & Schmeier, S. (2025). Detecting Pipeline Failures through Fine-Grained Analysis of Web Agents. arXiv preprint arXiv:2509.14382.

[10] Singh, S. B., Rizvi, M. A., Saxena, K., Gupta, R., Tripathi, A. N., & Dewangan, N. K. (2025). An adaptive, energy-efficient and secure routing protocol for zone-related mobile Ad-hoc networks using reinforcement learning: SB Singh et al. Scientific Reports.

[11] Qiu, Y., & Wang, J. (2023, October). A machine learning approach to credit card customer segmentation for economic stability. In Proceedings of the 4th International Conference on Economic Management and Big Data Applications, ICEMBDA (pp. 27-29).

[12] Zhu, W., Yao, Y., & Yang, J. (2025). Real-Time Risk Control Effects of Digital Compliance Dashboards: An Empirical Study Across Multiple Enterprises Using Process Mining, Anomaly Detection, and Interrupt Time Series.

[13] Krishnan, N. (2025). Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. arXiv preprint arXiv:2504.21030.

[14] Li, T., Xia, J., Liu, S., & Jiang, Y. (2025). Digital Transformation of Human Resources: From Consulting Frameworks to AI-Enabled Learning Management Systems.

[15] Mukherjee, A., & Chang, H. (2025). Operational Agency: A Framework for Tracing Intent and Liability in Multi-Agent Artificial Intelligence Systems. Available at SSRN 5344615.

[16] Gu, X., Liu, M., & Yang, J. (2025). Application and Effectiveness Evaluation of Federated Learning Methods in Anti-Money Laundering Collaborative Modeling Across Inter-Institutional Transaction Networks.

[17] Eziokwu, U. J., Duruemeruo, U. C., Akande, N. A., & Makinde, O. F. (2023). Cost-Aware and Risk-Sensitive Learning for Autonomous Robots A Conceptual Framework.

[18] Mao, Y., Ma, X., & Li, J. (2025). Research on Web System Anomaly Detection and Intelligent Operations Based on Log Modeling and Self-Supervised Learning.

[19] Krupp, L. A. R. S., Geißler, D. A. N. I. E. L., Woźniak, P. W., Lukowicz, P., & Karolus, J. (2025). Quantifying Web Agents-A Survey on Web Agent Performance and Efficiency.

[20] Zhu, W., Yang, J., & Yao, Y. (2025, October). How Compliance Maturity Translates to Risk Reduction: A Multi-Case Comparison of Global Operations Using fsQCA and Hierarchical Bayesian Methods. In Proceedings of the 2025 2nd International Conference on Digital Economy and Computer Science (pp. 672-676).

[21] Maity, D. (2026). SAFE: Stable Alignment Finetuning with Entropy-Aware Predictive Control for RLHF. arXiv preprint arXiv:2602.04651.

[22] Li, T., Xia, J., Liu, S., & Hong, E. (2025). Strategic Human Resource Leadership in Global Biopharmaceutical Enterprises: Integrating HR Analytics and Cross-Cultural.

[23] Lumer, E., Gulati, A., Nizar, F., Hedroits, D., Mehta, A., Hwangbo, H., ... & Burke, J. A. (2025). Tool and Agent Selection for Large Language Model Agents in Production: A Survey.

[24] Gu, X., Yang, J., & Liu, M. (2025). Research on a Green Money Laundering Identification Framework and Risk Monitoring Mechanism Integrating Artificial Intelligence and Environmental Governance Data.

[25] Krupp, L. A. R. S., Geißler, D. A. N. I. E. L., Woźniak, P. W., Lukowicz, P., & Karolus, J. (2025). Quantifying Web Agents-A Survey on Web Agent Performance and Efficiency.