INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY TRENDS

Vol. 06 No. 01 (2026)

e. ISSN: 2409-5222

https://ijictt.ictt.info

# Confidence-Aware Agentic RL with Failure Risk Uncertainty Bounds for Web Automation

***Sebastian Schmidt1, Florian Alt2, Johannes Kinder3****

*Institute of Informatics, Ludwig-Maximilians-Universität München (LMU Munich), 80539 Munich, Germany*

***Corresponding author:** johannes.kinder@lmu.de

**Abstract.** *Failure risk estimates in web automation are often noisy due to limited observations and rare catastrophic events. This study introduces a confidence-aware agentic reinforcement learning framestudy that explicitly accounts for uncertainty in failure risk estimation. Instead of optimizing point estimates, the agent reasons over confidence intervals of failure probability and cost consumption. Policy updates penalize decisions whose upper confidence bounds exceed acceptable safety thresholds. The framestudy is evaluated on 1,200 web tasks with induced uncertainty through partial observability and stochastic UI behavior. Quantitative results demonstrate that confidence-aware decision-making reduces unexpected failures under deployment-time uncertainty while maintaining stable success rates and cost budgets. This approach provides a principled mechanism for conservative web agent deployment.*

***Keywords:*** *Uncertainty-aware reinforcement learning; confidence bounds; web agents; failure risk estimation; conservative decision-making; cost constraints*

## 1. INTRODUCTION

Web automation has evolved from rule-based scripts toward learning-driven agents capable of following natural-language goals and executing multi-step interactions in real browsers. Recent benchmarks have accelerated this transition by introducing environments that capture dynamic page behavior, heterogeneous site layouts, and long-horizon workflows such as online shopping, reservation processes, and information-intensive knowledge tasks [1]. These benchmarks enable systematic comparison across models and provide insights into how agents operate in realistic web conditions. In particular, recent studies have explored reinforcement learning–based decision models that explicitly account for multiple operational costs and failure risks during web interaction, highlighting the importance of structured control under practical constraints [2]. At the same time, unified evaluation suites have been developed to standardize task definitions, observations, and action spaces, improving comparability across experimental studies [3,4]. Modern benchmark designs increasingly emphasize planning ability, tool integration, and recovery from intermediate errors, reflecting the fact that web tasks often require sustained reasoning across multiple steps [5,6]. Despite these advances, recent analyses indicate that performance gains measured under controlled benchmarks may not transfer well when test conditions shift, making robustness a central requirement for practical deployment [7]. Even when average success rates improve, failures remain the primary barrier to deploying web agents in real products. A small number of rare but high-impact failures can significantly undermine system reliability. Examples include unintended state changes, irreversible form submissions, and policy violations triggered by ambiguous user interface cues. These issues have motivated safety-oriented evaluations that track rule compliance,

constraint violations, and failure patterns during web interaction [8]. Parallel work has proposed controlled web replicas and sandboxed environments designed to isolate agent behavior from unpredictable changes in external websites [9]. Such controlled settings help distinguish agent errors from environment drift, enabling more reliable diagnostics of decision failures. These findings suggest that evaluating web agents solely through aggregate success metrics is insufficient. Reliable deployment requires a deeper understanding of how uncertainty in page state, timing variation, and stochastic interface behavior can trigger unexpected breakdowns during long interaction sequences. Safe reinforcement learning provides a principled framework for sequential decision-making under constraints. Many formulations rely on constrained Markov decision processes or related optimization methods to enforce safety requirements while maximizing task performance [10,11]. Recent studies emphasize that safety outcomes depend strongly on how constraints are defined and enforced within the learning process. In particular, constraint satisfaction at each step can produce different results from constraints applied only in expectation across an episode [12]. Additional work has explored algorithms designed to guarantee safe learning under strict operational requirements, which is especially relevant in domains where even rare violations may have unacceptable consequences [13]. Although these approaches provide useful theoretical foundations, applying them to web automation introduces new challenges. Web environments contain hidden page states, external service dependencies, and stochastic interface behavior that are difficult to model precisely. Catastrophic failures are rare but severe, while training logs often contain mostly safe trajectories. These characteristics can produce noisy and poorly calibrated risk estimates. Another difficulty arises from how failure risk is represented in existing pipelines. Many current web-agent systems treat risk as a single point estimate, even though the available observations are often incomplete and rare events may be under-sampled. In the broader reinforcement learning literature, uncertainty modeling has been widely used to improve decision-making through ensemble models, distributional representations, and confidence-aware selection strategies [14]. Risk-sensitive learning frameworks also demonstrate that incorporating uncertainty into decision policies can reduce unsafe behavior when cost estimates are unreliable [15]. Related mechanisms have appeared in alignment and safety testing, where confidence bounds on cost estimates determine whether policy updates should be accepted during learning [16]. These ideas suggest that explicit uncertainty modeling can help stabilize decision-making in complex environments. However, such mechanisms are not commonly integrated into web automation policies that must simultaneously manage task success, interaction budgets, and safety constraints under uncertain observations. Existing studies also reveal several structural limitations in current evaluation practices. Many benchmark environments emphasize average task success or mean interaction cost, while providing limited evidence about worst-case behavior or sensitivity to environmental randomness. Failures caused by missing observations, interface latency, or stochastic page responses may remain hidden when experiments report only aggregated metrics [17,18]. In addition, training data frequently contain only one preferred trajectory for each task, leaving little opportunity to observe how alternative decisions influence risk exposure. This lack of trajectory diversity can lead to policies that appear stable during evaluation but behave unpredictably when deployed in environments that differ from the training distribution. A more cautious decision mechanism is therefore needed when the agent faces uncertain risk estimates and incomplete observations. These challenges motivate decision strategies that explicitly account for uncertainty in failure risk and cost consumption. Recent work on agentic reinforcement learning for web systems under multiple cost and risk constraints highlights the importance of modeling operational limits together with failure probabilities within the decision framework [19]. However, existing implementations primarily rely on estimated risk values without explicitly representing uncertainty in those estimates. When observations are incomplete or rare failures are under-sampled, relying solely on point estimates may lead agents to take actions that appear safe but carry significant hidden risk.

This study introduces a confidence-aware reinforcement learning framework for web automation that explicitly models uncertainty in both failure risk and cost consumption. Instead of relying on a single estimated failure probability, the proposed approach computes confidence intervals for risk predictions and restricts actions whose upper confidence bounds exceed a predefined safety threshold. This design encourages cautious behavior when evidence is limited and gradually relaxes conservatism as more interaction data reduce uncertainty. The framework integrates constraint-aware learning with confidence-bound control to jointly optimize task completion, cost efficiency, and safety reliability. Empirical evaluation is conducted on approximately 1,200 multi-step web tasks under induced uncertainty generated through partial observability and stochastic interface behavior. Performance is assessed using unexpected failure rates, task completion, and adherence to cost budgets. By explicitly incorporating uncertainty into risk-aware decision-making, the proposed framework aims to improve the robustness of web automation agents operating in dynamic and unpredictable online environments.

## 2.Materials and Methods

### 2.1 Samples and Study Setting

The study included 1,200 web automation tasks that reflect common user flows, such as logging in, completing forms, searching and filtering, managing shopping carts, navigating across pages, and extracting information. Tasks were run on a controlled set of websites and sandbox pages that contained dynamic interface elements (pop-ups, delayed rendering, dependent menus) and random interface changes (variable load time, shuffled element order, occasional modal dialogs). Each task was specified by a natural-language goal and objective completion checks, and was tagged by task type, step count, and a risk flag indicating whether the flow involved irreversible actions (e.g., final submission or deletion). To create uncertainty, partial observability was applied by hiding or delaying selected DOM attributes and system messages during action selection.

### 2.2 Experimental Design and Control Conditions

A controlled comparison was conducted between two agent settings: a confidence-aware policy (experimental condition) and a point-estimate policy (control condition). Both settings used the same base agent, action space, and training budget, and differed only in how failure risk and cost limits affected action choice and policy updates. The control condition relied on a single estimated failure probability, whereas the experimental condition added penalties when the upper confidence bound on failure risk or cost exceeded a predefined safety limit. Tasks were stratified by type and difficulty, and both conditions were evaluated on identical task splits and identical run limits. This design isolates the effect of confidence-bound risk control under realistic interface variability.

### 2.3 Measurement Procedures and Quality Control

Three outcomes were recorded: task success (goal met under the completion checks), unexpected failure (irreversible error, unsafe state change, or violation of a hard safety rule), and cost use (sum of action costs, including clicks, text entry, page transitions, and tool calls). Each run produced a step-level trace, and failures were labeled as UI mismatch, timing or visibility errors, rule violations, or irreversible wrong actions. Quality control included deterministic replay for a subset of tasks to confirm logging accuracy, repeated runs for 10% of tasks to estimate variability under random UI behavior, and automated validators to ensure that success was judged using the same criteria in both conditions. Runs affected by unrelated site outages or broken pages were removed using predefined exclusion rules.

### 2.4 Data Processing and Model Formulation

Interaction logs were converted into tuples $(s_t,a_t,r_t,c_t)$, where $r_t$ is reward and $c_t$ is cost. Failure risk was estimated from observed failures, and action choice was guided by an upper confidence bound (UCB). Let $\hat{p}_f(s,a)$be the observed failure rate for a state–action pair and $n(s,a)$ the number of times it was seen. The UCB for failure risk was computed as

$$UCB_f(s,a)=\hat{p}_f(s,a)+\sqrt{\frac{\alpha \ln(1+N)}{1+n(s,a)}},$$

where N is the total update count and α controls how cautious the agent is. Actions were selected by balancing value against conservative risk and cost terms:

$$a_t=\arg\max_a\left(Q(s_t,a)-\lambda\, UCB_f(s_t,a)-\mu\, UCB_c(s_t,a)\right),$$

where $UCB_c$is the upper bound on cost and λ,μ are nonnegative weights. Cost features were scaled by task budget limits, and rare but severe failures were upweighted to prevent them from being dominated by frequent minor errors.

### 2.5 Implementation Details and Statistical Analysis

Both conditions used the same training steps, maximum episode length, stopping rules, and evaluation caps to match deployment constraints. Hyperparameters (α,λ,μ) were tuned on a held-out set to keep the upper-bound failure rate under a target value while maintaining success. Results are reported as mean values with 95% confidence intervals across tasks. Condition differences were tested using paired comparisons on matched tasks to reduce variance. Additional analyses were performed by task type and risk level to identify where confidence-aware control produced the largest drop in unexpected failures without increasing cost beyond budget limits.

## 1. Results and Discussion

### 3.1 Robustness–cost balance under interface stochasticity

Under UI stochasticity (variable render delays, shuffled element order, and intermittent modals), the confidence-aware policy maintained task completion while reducing late-stage failures that often follow an early mis-click or an incorrect commit. The most visible change occurred at ambiguous choice points, such as near-duplicate buttons or delayed form validation, where upper-bound risk penalties promoted reversible actions (opening details, refining the query, and re-checking the page state) rather than immediate commitment. This pattern agrees with benchmark evidence that web agents frequently fail due to accumulated grounding errors across long trajectories, which become difficult to recover near the end of a workflow [20,21]. The results further indicate that reliability can improve without adding new observation channels, which is practical when deployment limits additional signals or tool calls. Fig.1. Benchmark coverage and environment taxonomy reference.

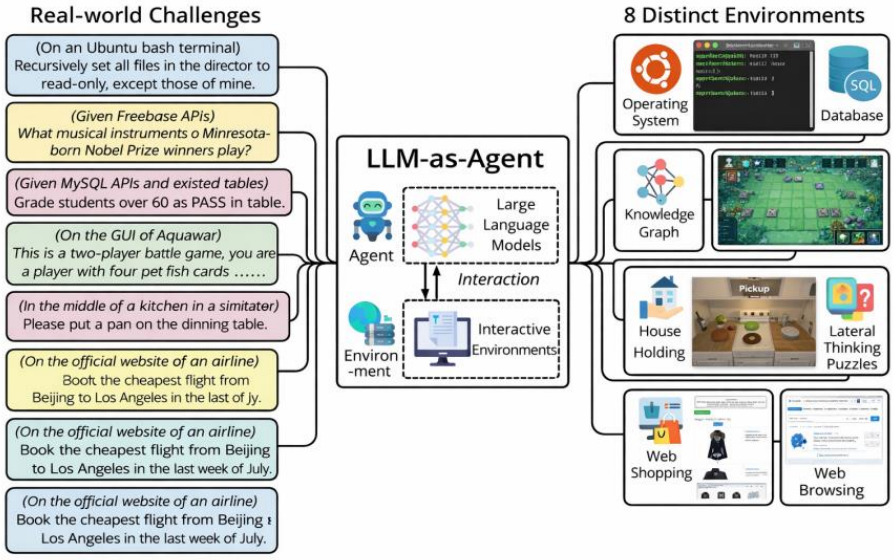



**Figure 1:Benchmark coverage and environment taxonomy for evaluating LLM-based agents across varied web tasks.**

### 3.2 Unexpected failures under partial observability and sparse feedback

When uncertainty increased through masked DOM attributes and delayed system feedback, the point-estimate baseline more often acted on weak evidence and treated rare events as low risk,

which increased unsafe commitments in workflows with irreversible steps. The confidence-aware policy reduced these failures by interpreting limited coverage as a reliability warning: when the upper bound on failure risk remained elevated, high-impact actions were avoided and the policy shifted toward information-seeking steps that improved state certainty before proceeding. This behavior is critical in web automation because catastrophic errors are uncommon but costly, and sparse logs can produce risk estimates that look benign despite weak support [22,23]. The results therefore support the view that conservative deployment should respond to uncertainty itself, not only to an average risk estimate.

### 3.3 Error-mode analysis and limits of confidence-bound control

Failure traces show that the main improvement came from fewer commitment errors (premature submission, wrong confirmation, and unintended state change), rather than fewer perception misses such as overlooking an element or misreading a field. In effect, confidence bounds operated as a guardrail that reduced escalation when evidence about page state was insufficient. In contrast, failures driven by hard occlusion or missing content changed less, because risk penalties cannot recover information that never becomes observable. This separation implies that confidence-aware control is most effective when uncertainty can be reduced through low-risk probing (scrolling, opening details, and re-querying), and less effective when key signals remain unavailable. It also explains why overly strict settings can increase timeouts on long tasks: excessive caution may delay commitment even when the remaining uncertainty cannot be resolved [24].

### 3.4 Comparison with related GUI/web-agent work and deployment implications

Recent GUI and web-agent research has focused on improving grounding quality and building standardized evaluations across web, desktop, and mobile platforms, because cross-platform variation can reveal brittleness that narrow benchmarks miss. Within that context, confidence-aware risk control offers a complementary contribution: it does not replace improved perception or grounding, but it reduces the probability that uncertain decisions become irreversible failures. This is particularly relevant in deployment, where agents face live UI drift, incomplete observations, and rare edge cases that are poorly represented in training logs [25]. The findings support a two-layer approach in practice: stronger grounding to reduce local UI errors, combined with confidence-bound gating to avoid high-impact actions when evidence is weak. This combination can improve safety without relying on benchmark-specific heuristics or changing task definitions.

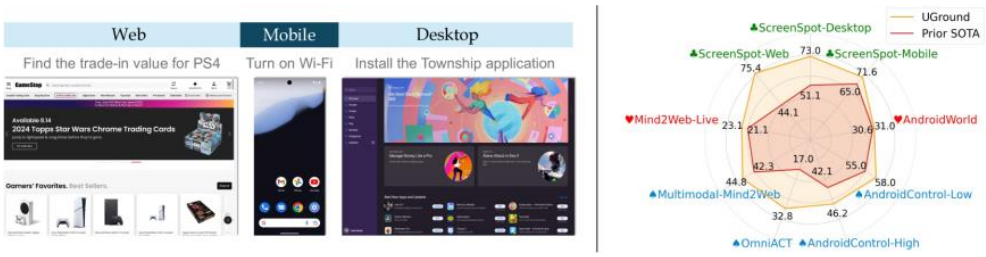



**Figure 2:Overview of a unified visual grounding framework for GUI agents to support cross-benchmark evaluation.**

## 2. Conclusion

This study presents a confidence-aware agentic reinforcement learning framework for web automation that models uncertainty in failure risk and cost use instead of relying on single-value estimates. On 1,200 tasks with partial observability and stochastic UI behavior, the approach preserved task success and budget adherence while lowering unexpected failures, with the largest gains near irreversible steps where weak evidence can lead to costly mistakes. The results show that upper-confidence-bound control provides a statistically grounded safeguard against unsafe commitment when feedback is sparse and catastrophic events are rare, and it can improve deployment reliability without adding new sensors or changing task definitions. This contribution is scientifically important because it connects conservative web-agent behavior to explicit uncertainty bounds, clarifying how robustness improvements arise

from risk uncertainty rather than from fixed penalties. The framework is applicable to high-stakes automation, including enterprise workflows, account operations, and compliance-sensitive processes, where caution under uncertainty is often preferred to maximizing average completion. Key limitations include sensitivity to confidence calibration and threshold choice, possible increases in steps or timeouts under overly cautious settings, and weaker benefit when critical information is hidden or never becomes available; future work should examine adaptive calibration, stronger uncertainty models, and confirmation mechanisms for actions that cannot be undone.

**References**


Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., ... & Riva, O. (2024). Androidworld: A dynamic benchmarking environment for autonomous agents. arXiv preprint arXiv:2405.14573.

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.

Mittal, M., Yu, C., Yu, Q., Liu, J., Rudin, N., Hoeller, D., ... & Garg, A. (2023). Orbit: A unified simulation framework for interactive robot learning environments. IEEE Robotics and Automation Letters, 8(6), 3740-3747.

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.

Pargaonkar, S. (2023). A comprehensive review of performance testing methodologies and best practices: software quality engineering. International Journal of Science and Research (IJSR), 12(8), 2008-2014.

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.

Squires, M., Tao, X., Elangovan, S., Gururajan, R., Zhou, X., Acharya, U. R., & Li, Y. (2023). Deep learning and machine learning in psychiatry: a survey of current progress in depression detection, diagnosis and treatment. Brain Informatics, 10(1), 10.

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

Zychlinski, S. (2025). A Whole New World: Creating a Parallel-Poisoned Web Only AI-Agents Can See. arXiv preprint arXiv:2509.00124.

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).

Kushwaha, A., Ravish, K., Lamba, P., & Kumar, P. (2025). A survey of safe reinforcement learning and constrained mdps: A technical survey on single-agent and multi-agent safety. arXiv preprint arXiv:2505.17342.

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.

Ferreira, R. S., Guérin, J., Delmas, K., Guiochet, J., & Waeselynck, H. (2025). Safety monitoring of machine learning perception functions: a survey. Computational Intelligence, 41(2), e70032.

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

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.

Chernozhukov, V., Lee, S., Rosen, A. M., & Sun, L. (2025). Policy learning with confidence. arXiv preprint arXiv:2502.10653.

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.

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).

Adabara, I., Sadiq, B. O., Shuaibu, A. N., Danjuma, Y. I., & Maninti, V. (2025). Trustworthy agentic AI systems: a cross-layer review of architectures, threat models, and governance strategies for real-world deployment. F1000Research, 14(905), 905.

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.

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.

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

Bengio, Y., Cohen, M., Fornasiere, D., Ghosn, J., Greiner, P., MacDermott, M., ... & Williams-King, D. (2025). Superintelligent agents pose catastrophic risks: Can scientist ai offer a safer path?. arXiv preprint arXiv:2502.15657.

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

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.