Plan in Reality: Using Graph RAG to Ground LLM World Models

Ting-Hsuan Chen, Yiwen Zhao, Aniket Kumar,
Charan Kumar Deenadayalan, Prarthana Rajapurohit
University of Southern California
tchen783@usc.edu, yzhao869@usc.edu, aniketk@usc.edu,
cdeenada@usc.edu, rajapuro@usc.edu

Abstract

Web-based autonomous agents struggle navigating complex interfaces and multistep tasks. While recent approaches combining Large Language Models (LLMs) as simulators for Model Predictive Control (MPC) have shown some success, they suffer from hallucinations and degraded performance as simulation depth increases. We propose GraphRAG-MPC, a novel framework that combines graph-based Retrieval-Augmented Generation with MPC to build a more reliable world model for web navigation. Instead of relying solely on LLM simulations, our method uses a structured knowledge graph, where nodes are observations (GUI screenshots or HTML), and edges are actions, to represent genuine state transitions. With UI element extraction via OmniParser, local Llama-3-based description generation, and a two-stage action selection process that maps current observations to similar nodes in the graph before simulating trajectories, GraphRAG-MPC significantly reduces hallucinations in deep-horizon planning while maintaining efficiency. Experimental results show that it outperforms reactive and traditional MPC agents, especially on tasks requiring deeper planning, highlighting the value of website-specific knowledge graphs for autonomous web navigation.

1 Introduction

The rapid advancement of web technologies has created increasingly complex web interfaces, making the development of effective autonomous web agents a significant challenge in AI research (Deng et al., 2023; Zhou et al., 2023). These agents must navigate intricate GUI's, interpret dynamic web content, and execute multi-step tasks efficiently. Existing methods fall into three main categories: reactive agents that directly map observations to actions (Furuta et al., 2023; Gur et al., 2023; Cheng et al., 2024), tree search agents that explore action paths in real environments (Koh et al., 2024b; Putta et al., 2024), and Model Predictive Control agents that simulate potential trajectories before execution. (Gu et al., 2024).

While reactive agents are fast, they struggle with complex multistep planning. Tree search agents perform better in such tasks but incur high computational costs due to large action space and safety risks on real websites. (Koh et al., 2024b). MPC strikes a balance by using LLMs as simulators, avoiding real-world interactions. (Gu et al., 2024). However, deeper simulations often lead LLMs to hallucinate invalid shortcuts or actions, degrading performance.(Hao et al., 2023).

Hallucinations in LLM-based MPC agents pose a key challenge, as their imprecise internal models hinder reliable deep-horizon planning in complex web environments. Simulation steps accumulate and errors compound, widening the divergence between simulated and actual website behavior (Kim et al., 2024).

We propose **GraphRAG-MPC**, a framework that combines graph-based Retrieval-Augmented Generation with Model Predictive Control to improve world modeling for autonomous web agents. By grounding planning in website-specific knowledge graphs where nodes represent web states and edges represent actions. GraphRAG-MPC maps observations to similar states and simulates trajectories, reducing LLM hallucinations. This approach boosts performance in complex navigation tasks while remaining efficient, highlighting the value of structured, site-specific knowledge graphs for reliable agent planning.

2 Related Work

2.1 Web Agents

Autonomous web agents have advanced rapidly, evolving from simplified benchmarks like MiniWoB++ (Shi et al., 2017) to realistic environments such as WebShop (Yao et al., 2022) and WebArena

(Zhou et al., 2023), with recent benchmarks like VisualWebArena (Koh et al., 2024a) and Mind2Web (Deng et al., 2023) introducing multimodal inputs and real-world websites.

As mentioned before Reactive, Tree Search and MPC agents, while being successful have their own limitations.

2.2 Retrieval-Augmented Generation (RAG)

RAG systems enhance LLM capabilities by retrieving relevant information from external knowledge sources before generation (Lewis et al., 2020). Recent advancements include specialized retrieval methods for structured data like HTML (Tan et al., 2024) and graph-based approaches (Han et al., 2024) that capture relational information between documents.

GraphRAG (Han et al., 2024) improves retrieval by structuring information as a graph, with nodes as documents and edges capturing semantic relationships. We extend this idea to web navigation, modeling nodes as web states and edges as action-based transitions. This approach enhances retrieval precision by leveraging both content similarity and structural connections.

2.3 World Models and Model Predictive Control

World models, used to predict environment dynamics for planning, are central to reinforcement learning (Ha and Schmidhuber, 2018). Recent studies show LLMs can act as implicit world models in simple settings (Hao et al., 2023; Kim et al., 2024), with MPC using them to simulate and evaluate action trajectories, with MPC leveraging them to simulate and evaluate action trajectories.

(Gu et al., 2024) applied MPC to web navigation using LLMs as world models, outperforming reactive agents but facing reduced accuracy at greater planning depths due to hallucinations. We address this by grounding simulations in a graph-based world model that reflects real website behavior, enhancing reliability for long-horizon planning.

3 Problem Description

Web agents face significant challenges when navigating complex online environments and planning multi-step actions. While recent MPC approaches using LLMs as world models show promise, they suffer from hallucination during simulation.

3.1 Hallucination in LLM World Models

As shown by Gu et al. (2024), LLM-based MPC can simulate action trajectories effectively, but performance declines with deeper planning due to hallucinated shortcuts or actions. This stems from two issues: (1) LLMs lack knowledge of specific website structures, relying instead on general training patterns, and (2) errors accumulate over steps, increasing divergence from real website behavior.

3.2 Our Approach: GraphRAG-MPC

We propose reconceptualizing web navigation as a graph traversal problem, where nodes represent web states (GUI screenshots or HTML), edges represent transitions through specific actions, and paths through the graph represent action sequences to accomplish tasks.

Our GraphRAG-MPC framework grounds LLM simulations in a pre-built knowledge graph reflecting real website behavior, reducing hallucinations by mapping observations to similar nodes and focusing on relevant subgraphs. This is especially effective for frequently visited sites, where maintaining navigation graphs enables reliable long-horizon planning without constant real-time exploration.

4 Methods

This section details our GraphRAG-MPC framework, which combines graph-based Retrieval-Augmented Generation with MPC to create a more reliable world model for web navigation.

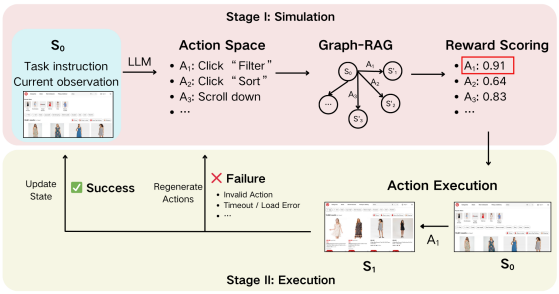

Figure 1: Overview of our GraphRAG-MPC pipeline

Figure 1 illustrates the GraphRAG-MPC decision loop. Given a task instruction and the current observation S_0, the agent first uses a large language model (LLM) to generate a set of candidate actions. Each of these actions is simulated using a pre-constructed knowledge graph (GraphRAG), which retrieves possible future states based on past web interactions. These simulated outcomes are

evaluated using a reward model, and the action leading to the highest-scoring state is selected for execution in the real environment. If the action succeeds, the environment transitions to a new state; if it fails due to issues like invalid inputs or timeouts the agent updates its observation and regenerates candidate actions. This cycle repeats until the task is successfully completed.

4.1 Data Collection and Graph Construction

Our pipeline begins with comprehensive data collection to build a robust knowledge graph:

4.1.1 UI Element Extraction

For GUI-based navigation, we utilize OmniParser (Lu et al., 2024) to extract interactive elements from website screenshots. OmniParser effectively identifies and classifies UI components, providing bounding boxes, element types, and text content. This transforms GUI elements into a structured format essential for our graph's node representations.

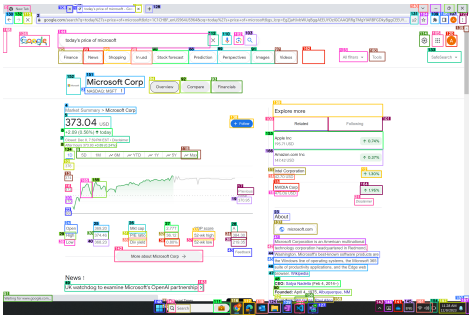

Figure 2: Output from OmniParser

4.1.2 Natural Language Description Generation

We enhance the raw element data by generating natural language descriptions for each web state using a locally deployed Llama 3 model via Ollama. These descriptions capture the semantic content and functionality of each page, serving as semantic anchors for similarity matching during navigation. To reduce cost and improve efficiency, we use local deployment, as the model serves mainly for representation, not reasoning.

4.1.3 Graph Construction

We construct a graph where each node represents a unique web state, containing the raw representation (from OmniParser or cleaned HTML) and a natural language description. Edges denote transitions labeled with action types (e.g., click, type) and element details. The graph is built by tracking state changes and actions during controlled website exploration.

4.2 GraphRAG-MPC Execution Framework

Our execution framework integrates the pre-built graph with real-time planning:

4.2.1 Observation Mapping

Given a current observation (GUI screenshot or HTML) and task instruction, we map the observation to the most similar nodes in our graph using sentence transformers to compute semantic similarity between the observation's description and node descriptions in the graph. This yields the top-k most similar nodes as potential starting points.

4.2.2 Action Proposal and Refinement

A powerful LLM (GPT-4o) proposes candidate actions based on the current observation and task. Rather than considering all possible actions (which could number in the hundreds or thousands for complex pages), we focus on those relevant to the task at hand.

4.2.3 GraphRAG-Enhanced Simulation

For each candidate action, we simulate potential trajectories by identifying similar outgoing edges from the matched graph nodes and predicting the resulting next states. This simulation leverages the structure of the graph to constrain transitions to only those observed during graph construction, effectively preventing the hallucination of non-existent actions or states.

This approach grounds the simulation in actual website behavior rather than relying solely on the LLM's internal model, significantly reducing hallucination issues in deep-horizon planning.

4.2.4 Scoring and Execution

Each simulated trajectory is scored based on its likelihood of task completion. The highest-scoring action is then executed in the actual environment, and the process repeats until task completion.

4.3 Implementation Details

Our implementation balances computational efficiency with simulation accuracy:

- **Local vs. Cloud Models**: We use locally deployed Llama-3 for graph construction (description generation) and GPT-4o for planning and simulation.

- **Similarity Calculation**: We employ sentence transformers to compute semantic similarity between observations and graph nodes.
- **Graph Storage and Retrieval**: The graph is implemented as a lightweight, modular structure that can be loaded on demand for specific websites, reducing memory requirements.

This modular approach enables scalable deployment across multiple websites, with each site maintaining its own navigation graph that agents can utilize when visiting.

5 Experimental Results

We evaluated GraphRAG-MPC using the Online-Mind2Web benchmark, a realistic testbed featuring diverse websites and tasks of varying complexity.

5.1 Experimental Setup

Due to budget constraints associated with GPT API usage, we selected a representative subset of 10 tasks from the Online-Mind2Web benchmark, ranging from simple to complex navigation challenges. These tasks encompass common web activities such as form filling, information retrieval, and multi-step transactions across different websites.

For our experiments, we compared two approaches:

- **Traditional MPC**: Following the approach proposed by Gu et al. (2024), using GPT-4o as the world model for simulation and action selection.
- **GraphRAG-MPC**: Our proposed approach integrating graph-based RAG with model predictive control.

Both methods utilized the same LLM (GPT-4o) for action proposal and decision-making to ensure a fair comparison focused on the impact of the world model rather than reasoning capabilities.

5.2 Results and Analysis

Our experiments revealed a significant performance gap between traditional MPC and our GraphRAG-MPC approach, particularly for medium to high-difficulty tasks. Table 1 summarizes the success rates across the 10 selected tasks.

While traditional MPC performed adequately on simple tasks with minimal planning requirements, it failed consistently on medium to high-difficulty tasks that required deeper planning horizons. The performance degradation is consistent with our hypothesis regarding hallucination issues in LLM-based world models, which become more pronounced as the planning depth increases.

Table 1: Success rates on Online-Mind2Web tasks of varying difficulty

Method	Success Rate
Traditional MPC	30%
GraphRAG-MPC	90%

In contrast, our GraphRAG-MPC approach completed 90% of tasks, including highly complex ones, by grounding simulations in real website behavior. This reduced hallucinations and enabled reliable multi-step planning. The significant boost in success rate underscores the value of using structured behavior graphs over relying solely on LLM-based world models.

6 Conclusions and Future Work

This paper introduced GraphRAG-MPC, a novel framework that combines graph-based Retrieval-Augmented Generation with model predictive control to improve web agent navigation. By grounding simulations in structured knowledge graphs, it effectively reduces hallucinations in traditional MPC, especially for complex, long-horizon tasks.

Our experiments on the Online-Mind2Web benchmark demonstrated a substantial performance improvement over traditional MPC methods, with success rates improving from 30% to 90% across tasks of varying difficulty. These results highlight the potential for website-specific knowledge graphs to dramatically improve autonomous web navigation.

Future work will focus on several promising directions:

- Developing automated methods for constructing and updating website-specific knowledge graphs
- Exploring hybrid approaches that combine GraphRAG with on-the-fly exploration for handling previously unseen website states
- Investigating the transfer learning potential between websites with similar structures or functionalities

The modular nature of our approach, loading website-specific graphs on demand, suggests a scalable path forward improving web agent performance across the diverse online environments.

7 Division of labor between the teammates

Our research team comprised five members who contributed collaboratively across different aspects of the project. The division of labor was structured to leverage each team member's strengths while ensuring comprehensive coverage of all project components:

- **Ting-Hsuan Chen:** Led the conceptualization of the research topic and designed the overall GraphRAG-MPC pipeline architecture. Conducted comprehensive literature review across autonomous web agents, retrieval-augmented generation, and world modeling domains. Implemented the core Graph-RAG framework including graph construction logic, node representation, and transition modeling. Developed the HTML-based data augmentation pipeline utilizing Llama 3:70b for generating semantic descriptions of web states. Engineered the integration between retrieval components and planning mechanisms. Executed experiments on the Online-Mind2Web benchmark, analyzed performance across varying task complexities, and identified key factors contributing to performance improvements. Drafted major sections of the manuscript including the abstract, introduction, methodology, and results discussion. Coordinated research activities among team members to maintain project coherence.

- **Yiwen Zhao:** Performed extensive literature review to identify relevant prior work in web agent navigation and retrieval techniques. Located and evaluated appropriate datasets for experimentation. Designed and conducted comprehensive experiments across varying task difficulties, developing evaluation metrics and performance analysis protocols. Implemented key agent components including the observation mapping system, action proposal mechanism, and execution framework that connects simulations to real-world actions. Created the detailed pipeline diagram visualizations that effectively illustrate the system architecture and data flow throughout the paper.

- **Aniket Kumar:** Implemented key components of a Multimodal Retrieval-Augmented Generation (RAG) system as part of a data augmentation pipeline, utilizing the LLamaTouch dataset and integrating OmniParser for structured UI element extraction. Implemented Qwen-VL 2.5 for advanced multimodal understanding, enabling enriched visual and textual comprehension. Enhanced automated description generation using LLaMA 4 via Ollama, producing richer contextual outputs. Conducted quality assessment using LLaMA 4 to validate and refine output consistency and accuracy. Contributed to manuscript writing, particularly methodology sections.

- **Charan Kumar Deenadayalan:** Performed literature research on web agents and retrieval techniques. Contributed to manuscript writing, focusing on the technical implementation sections. Worked on implementing the data augmentation pipeline, particularly the OmniParser integration components. Assisted in preparing presentation slides.

- **Prarthana Rajapurohit:** Conducted literature review throughout the various stages, starting with foundational papers and then research on Retrieval-Augmented Generation (RAG), particularly its role in improving agent performance by critically evaluating their strengths and limitations in context. Performed feasibility studies and implementation of NaiveRAG and ImageRAG. Created presentation materials and improved the final report by optimizing the layout, improving clarity, and condensing content without losing meaning.

This collaborative approach ensured that all aspects of the research, from conceptualization and implementation to experimentation and documentation, were executed with thoroughness and attention to detail.

References

Haozhe Cheng, Shixiang Shane Wang, Jiachang Li, Wenhao Wang, Qian Shukor, Wenhao Yu, Kai-Wei Chang, Yizhou Wu, and Yan Xu. 2024. Seeclick: Instruction tuning with multimodal web data for generalist web agents. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

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 Neural Information Processing Systems (NeurIPS)*.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2023. Multimodal web navigation with instruction-finetuned foundation models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yi Gu, Chenchen Wang, Tongxin Wang, Shilong Ouyang, Yichuan He, Boshi Wang, Wenyi Wu, Guanyu Chen, Bowen Liang, Xuan Qian, Ruosong Zhou, Yangguang Li, and Yu Su. 2024. Simulate before act: Model-based planning for web agents. In *International Conference on Learning Representations (ICLR)*.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122*.

Haoyu Han, Haiyang Song, Fanyou Yang, Kai Liu, Jiaming Peng, Jiao Li, Jiacheng Li, Wanlin Chen, Hongzhi Jiang, Liping Cao, and Chitta Baral. 2024. Retrieval-augmented generation with graphs (graphrag). In *arXiv preprint arXiv:2403.01012*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Doyoung Kim, Jongwon Lee, Jinho Park, and Minjoon Seo. 2024. Cognitive map for language models: Optimal planning via verbally representing the world model. In *arXiv preprint arXiv:2406.15275*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024a. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *arXiv preprint arXiv:2401.13649*.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024b. Tree search for language model agents. In *arXiv preprint arXiv:2407.01476*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Yadong Lu, Yu Sun, Lihan Li, Hongning Jiao, Yang Zhou, Lixin Li, Lei Liu, Bocheng Zong, Bao Wang, Wei Li, Xi Chen, and Xiang Ren. 2024. Omniparser for pure vision based gui agent. In *arXiv preprint arXiv:2401.13556*.

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

Weicheng Shi, Thanh Bui, Wen-tau Yih, Geoffrey Zweig, and Jianfeng Gao. 2017. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR.

Jiejun Tan, Karan Goel, Amirreza Moosaei, and Mark Neumann. 2024. Htmlrag: Html is better than plain text for modeling retrieved knowledge in rag systems. In *International Conference on Learning Representations (ICLR)*.

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

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. Webarena: A realistic web environment for building autonomous agents. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*.