HiPlan: Hierarchical Planning for Web Agents with Adaptive Global-Local Guidance

Ziyue Li, Yuan Chang, Yuanbo Kong, Gaihong Yu†, Xiaoqiu Le†

National Science Library, Chinese Academy of Sciences, Beijing, China

Department of Information Resources Management, School of Economics and Management, University of Chinese Academy of Sciences, Beijing, China

Abstract

Large language model (LLM)-based agents have shown remarkable potential in automating web-based tasks, but struggle significantly with complex, long-horizon web navigation and interaction scenarios characterized by vast action spaces, dynamic changes, and multi-step task dependencies. To address these challenges, we introduce HiPlan, a hierarchical planning framework that provides adaptive global-local guidance to boost web agents' decision-making. HiPlan first decomposes high-level objectives into milestone action guides for maintaining navigation direction and progress, while simultaneously offering step-wise hints for fine-grained and real-time interaction decisions that effectively bridge gaps and correct navigation errors. To enhance generalization and robustness across diverse tasks, we propose a dual-level experience reuse mechanism that leverages expert demonstrations to guide both the decomposition of the task and the generation of hints aligned with each milestone objective. This mechanism enables the agent to learn from successful historical interactions while adapting to current task contexts. Extensive experiments on challenging web agent benchmarks demonstrate that HiPlan substantially outperforms strong baselines, with ablation studies validating the critical and complementary benefits of all components for web-based agent planning.

Keywords: Autonomous Web Agents; Hierarchical Planning; Retrieval-Augmented Planning; Long-Horizon Decision Making; LLM-Based Agents

1. Introduction

The rapid expansion of Web-based services and online platforms has fundamentally transformed how users interact with digital information and accomplish daily tasks on the internet. From information seeking [1] and social communication [2] to online shopping [3] and travel booking [4], the Web has become an indispensable infrastructure for modern life. However,

†Corresponding author: Gaihong Yu (Email: yugh@mail.las.ac.cn; ORCID:0000-0003-1301-2871)

†Corresponding author: Xiaoqiu Le (Email: lexq@mail.las.ac.cn; ORCID:0000-0002-7114-5544)

Email addresses: liziyue@mail.las.ac.cn (Ziyue Li), changyuan@mail.las.ac.cn (Yuan Chang), kongyuanbo@mail.las.ac.cn (Yuanbo Kong), yugh@mail.las.ac.cn (Gaihong Yu†), lexq@mail.las.ac.cn (Xiaoqiu Le†)

many of these online activities remain repetitive, tedious, and time-consuming, such as navigating complex e-commerce interfaces to compare products and read user reviews, or extracting and synthesizing information across multiple heterogeneous web pages, which have caused a significant cognitive burden on users. Automating such routine yet complex Web tasks with intelligent agents (known as Web agents) can dramatically enhance productivity and user experience [5, 6].

Recent advances in large language models (LLMs) have exhibited remarkable human-like language understanding, reasoning, and instruction-following capabilities [7, 8], offering a promising foundation for building such autonomous Web agents [6]. By grounding LLMs in web interfaces through APIs or browser simulations, these agents can perform end-to-end Web tasks [9]. Despite this potential, enabling LLM-based agents to reliably and efficiently carry out complex, long-horizon navigation tasks in real-world Web environments remains a great challenge, as they inherently involve intricate interactions and multi-step dependencies across dozens of actions, requiring coherent progress toward high-level objectives within dynamic and unpredictable Web content.

Prior efforts on long-horizon Web agents usually employ ReAct-style approaches [10], where agents reason and take action iteratively based on immediate observations at each step [11, 12]. However, such methods often cause agents to drift from ultimate goals, get trapped in infinite loops, or pursue suboptimal trajectories due to limited foresight. Further works alleviate these issues by first breaking down the user task into a structured plan and solving them in sequence [13, 14]. Although this high-level planning provides better directional guidance, such methods often lack flexibility in handling dynamic errors and may require frequent replanning [15, 16], resulting in excessively long action sequences that dilute the focus of agents and increase computational costs.

In this work, we propose **HiPlan**, a hierarchical planning framework that endows LLM-based Web agents with adaptive global-local guidance to address these limitations. At the global level, HiPlan generates a milestone action guide by decomposing the user's high-level objective into a sequence of critical subgoals, thereby providing a roadmap. At the local level, HiPlan dynamically produces step-wise hints that offer fine-grained, context-aware suggestions for each action as real-time navigation, enabling the agent to timely correct errors and bridge gaps between the current state and the next milestone. This hierarchical planning structure helps the agent maintain goal-oriented while being responsive to the dynamic nature of the Web environments. As illustrated in Figure 1, compared to the reactive step-wise strategy that drifts from ultimate goals due to limited foresight and the task-decomposition planning that lacks flexibility in handling dynamic errors, HiPlan combines global milestone guidance with local step-wise hints to timely correct errors (e.g., selecting the wrong product attribute) and achieve reliable, efficient web task execution.

To further enhance reliability and generalization in real-world Web scenarios, we introduce a **dual-level experience reuse mechanism** that leverages expert demonstrations from past successful Web interactions. During agent execution, HiPlan retrieves experience at both hierarchical levels: (1) task-level retrieval finds similar Web tasks and their milestone sequences to guide task decomposition, and (2) milestone-level retrieval fetches trajectory fragments from analogous milestones to inform step-wise hint generation. This approach allows strategy transfer for both high-level planning and low-level execution, effectively grounding agent decisions in expert knowledge while adapting to the specific context of the current task. We conduct extensive experiments on Mind2Web and WebShop benchmarks, and the results demonstrate that HiPlan substantially outperforms strong baselines in terms of success rate and completion efficiency. Ablation studies further confirm that both the hierarchical guidance structure and the dual-level

experience reuse mechanism contribute significantly to the overall performance.

Our main contributions are summarized as follows:

- We propose HiPlan, a hierarchical planning framework that integrates global milestone action guides with local step-wise hints to enable adaptive and robust decision-making for LLM-based Web agents in long-horizon tasks.
- We design a dual-level experience reuse mechanism that leverages expert demonstrations to enhance both high-level task decomposition and real-time hint generation, improving generalization and responsiveness in dynamic web environments.
- Through extensive experiments on complex Web agent benchmarks, we show that HiPlan achieves substantial gains in task success rate compared to strong baselines while improving efficiency.

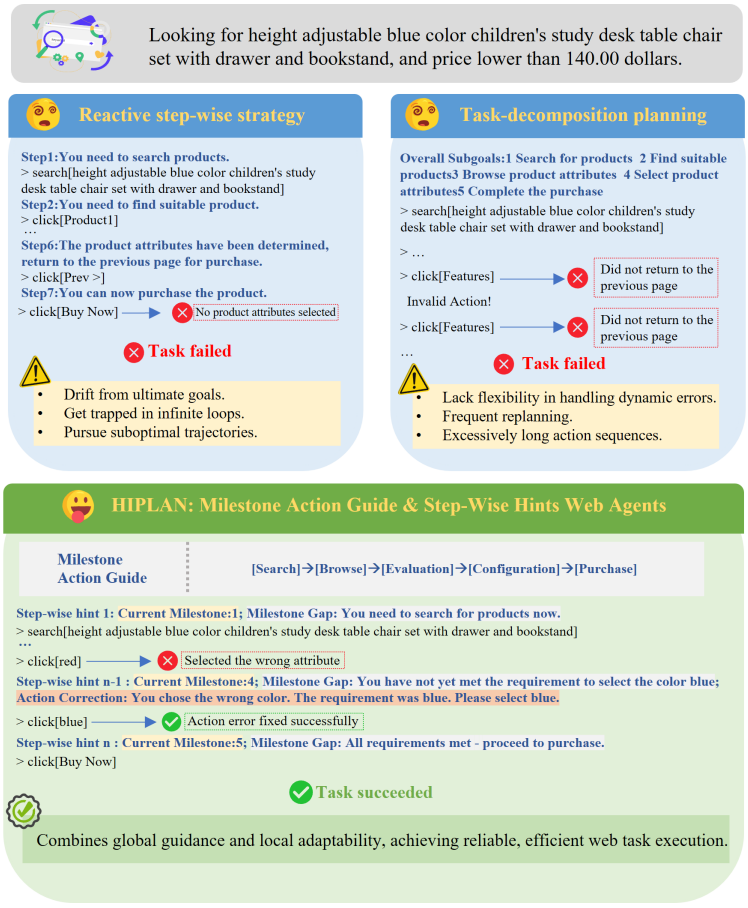

Figure 1: **Top-Left**: Reactive, step-by-step execution; **Top-Right**: One-time task decomposition planning; **Bottom**: HiPlan's hierarchical approach combining milestone guidance and step-wise hints for adaptive and robust planning for web tasks.

2. Related Work

2.1. LLM-Based Web Agents

LLMs have demonstrated remarkable capabilities across diverse domains such as complex reasoning [17, 18], problem-solving [19, 20], and text generation [21, 22]. The emergence of LLMs has advanced the development of autonomous agents capable of interacting with web environments [6]. These LLM-based Web agents are designed to automate a variety of online tasks such as information seeking [23] and online shopping [24]. Recent research has explored techniques such as supervised fine-tuning (SFT) [25], reinforcement learning (RL) [26], and in-context learning (ICL) [27] to enhance the reasoning and decision-making of LLM agents in Web tasks. In this paper, we mainly focus on tuning-free ICL methods, given their flexibility, cost efficiency and strong generalization across diverse tasks.

A prominent line of research adopts reactive frameworks, where agents iteratively predict next actions based on immediate observations. Zheng et al. [27] introduce ReAct method that interleaves chain-of-thought reasoning with action execution, enabling agents to maintain explicit reasoning traces throughout task completion. Building upon this, various studies enhance the decision-making of agents by transforming or summarizing observations into a task-aware context [11, 12]. These reactive methods excel at real-time adaptation, but often lead agents to lose sight of the overall task structure, resulting in inefficient exploration or suboptimal behaviors, particularly in long-horizon scenarios. To mitigate these issues, several works employ planning modules which focus on decomposing complex tasks into structured subgoals, forming comprehensive plans to guide task execution [13, 14, 28, 29, 30, 31]. However, they often exhibit limited flexibility when encountering unexpected execution errors or adapting to dynamic environmental changes. Other methods allow dynamic planning, which can refine the plan based on recent observations [32, 16, 33, 15], but frequent replanning can lead to lengthy action sequences, requiring more efficient context management.

Another line of work focuses on memory-augmented systems [34, 35, 36] that leverage historical experiences for improved decision-making, and reflection-based frameworks [37] that enable agents to learn from failures through self-critique and iterative improvement.

Despite these advances, the gap between high-level objectives and low-level execution remains a fundamental challenge. Our work seeks to bridge this gap through a hierarchical planning approach that synergistically combines milestone action guides with step-wise hints, enabling coarse-to-fine guidance for enhanced efficiency and robustness in long-horizon tasks.

2.2. Retrieval-Augmented Planning

Retrieval-augmented planning (RAP) methods enhance LLM-based agents by retrieving past experience—trajectories, plans, or instruction graphs—to ground planning in real execution data.

One approach retrieves relevant exemplars or context fragments conditioned on task similarity, enabling improved planning in both text-only and multimodal environments [34, 38, 39, 40]. Other works organize retrieval around abstract [41] or structured representations [42] to improve transferability.

Despite their strength, existing RAP approaches often rely heavily on task-level retrieval, which can introduce noise and limit flexibility. In this work, we incorporate milestone-level retrieval, enabling fine-grained experience transfer while retaining adaptability to execution context.

3. Preliminaries

We study the problem of long-horizon task completion in Web environments, where an autonomous agent must execute a coherent sequence of interactions with Web interfaces to fulfill a user's natural language instruction. The environment is formalized as a partially observable Markov decision process (POMDP), represented as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, T)$, where $\mathcal{S}$ denotes the set of latent environment states, $\mathcal{A}$ the discrete action space, $\mathcal{O}$ the observation space, and $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ the transition function.

At the beginning of each episode, the agent is provided with a user instruction $\tau \in \mathcal{T}$, expressed in natural language. Over a sequence of steps $t = 1, 2, \ldots, T$, the agent receives observations $o_t \in \mathcal{O}$ and selects actions $a_t \in \mathcal{A}$, aiming to reach a successful terminal state that satisfies τ.

4. Method

We propose HiPlan, a hierarchical planning framework that equips LLM-based agents with adaptive global-local guidance for tackling long-horizon tasks.

4.1. Overview

Fig. 2 illustrates the overall architecture and workflow of HiPlan. Our method decomposes the complex user instruction input into a sequence of critical milestones forming a high-level action guide, while progressively generating fine-grained step-wise hints to refine each action based on real-time observations.

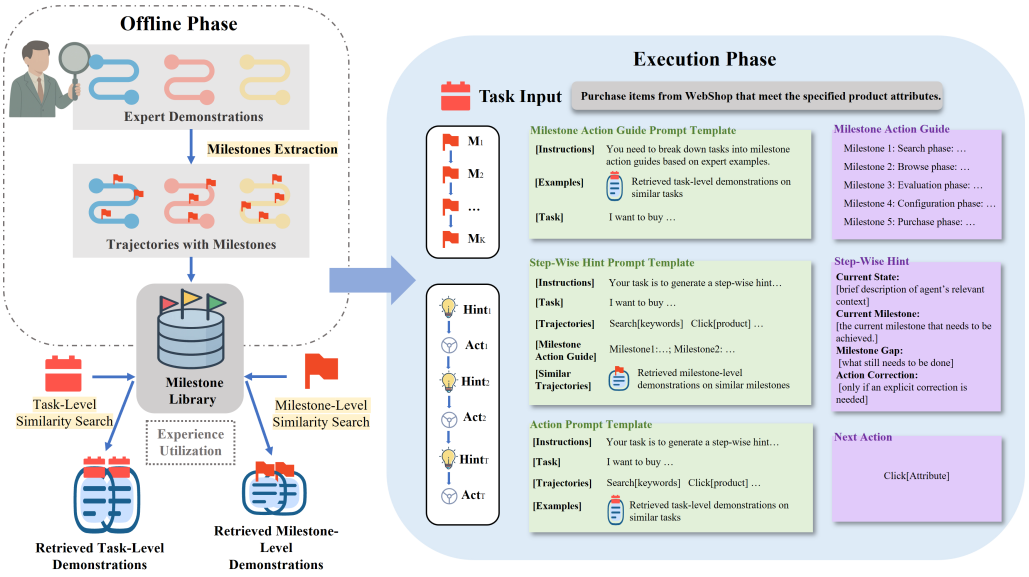

Figure 2: The HiPlan framework. In the offline phase (top left), a milestone library is constructed from expert demonstrations. During online execution (right), the retrieval flow is divided into two distinct stages: first, Task-Level Retrieval occurs exactly once at the start of the task to construct the fixed, instruction-specific global Milestone Action Guide; subsequently, Milestone-Level Retrieval happens dynamically at each individual execution step to inform the generation of local Step-Wise Hints.

To facilitate experience reuse, we assume access to a set of successful task demonstrations $\mathcal{D} = \{(\tau^{(i)}, \xi^{(i)})\}_{i=1}^{N}$, where each trajectory $\xi^{(i)} = \left[(o_1^{(i)}, a_1^{(i)}), \ldots, (o_{T^{(i)}}^{(i)}, a_{T^{(i)}}^{(i)})\right]$ corresponds to a task $\tau^{(i)}$. From these demonstrations, HiPlan constructs a milestone library and retrieves structured experience during execution.

We introduce two complementary forms of guidance for hierarchical planning:

- **Global Guidance:** A milestone action guide $\mathcal{G}_\tau = [m_1, m_2, \ldots, m_K]$, where each m_k is a natural language subgoal that represents a critical stage in completing the task. This sequence provides a high-level plan structure which serves as the coarse-grained directional guidance.
- **Local Guidance:** A step-wise hint h_t, generated at each timestep by retrieving trajectory fragments that correspond to the current milestone. This hint offers fine-grained behavioral suggestions conditioned on the current observation and subgoal.

The agent's policy $\pi : (\tau, o_t, \mathcal{G}_\tau, h_t) \rightarrow a_t$ integrates both levels of guidance to produce context-aware, goal-directed actions. The combination of a global roadmap and localized hint and retrieval enables robust and efficient decision-making in complex long-horizon tasks.

To explicitly articulate the "adaptive" nature of our global-local guidance, we emphasize that adaptivity operates at both levels of the hierarchical framework. At the global level, the macro-level **Milestone Action Guide** is adaptive to the specific semantics of the unseen user instruction. It is generated prior to execution by retrieving relevant task-level experiences, providing a tailored, instruction-specific roadmap that remains fixed during the current episode. At the local level, the micro-level **Step-Wise Hints** provide dynamic online adaptivity. They are continuously updated at each timestep during execution, dynamically adjusting to real-time environmental observations, interaction history, and the current milestone gap. This dual-adaptivity ensures that the agent is not only guided by a customized overarching plan but also equipped with highly responsive execution strategies to handle unpredictable environmental feedback.

4.2. Offline Phase: Milestone Library Construction

HiPlan constructs a milestone library $\mathcal{M}_L$ from the set of successful demonstrations $\mathcal{D} = \{(\tau^{(i)}, \xi^{(i)})\}_{i=1}^{N}$. For each trajectory $\xi^{(\rangle)}$, we segment it into $K^{(i)}$ contiguous fragments $\{\zeta_k^{(i)}\}_{k=1}^{K^{(i)}}$, each corresponding to a semantically meaningful subgoal. An LLM is prompted to generate a natural language description $m_k^{(i)}$ for each segment, forming a milestone sequence $\mathcal{G}_{\tau^{(i)}} = [m_1^{(i)}, \ldots, m_{K^{(i)}}^{(i)}]$.

It is worth noting that these milestones are not restricted to a fixed schema or task-specific templates. Instead, they are formulated through free-form natural language generation, allowing them to flexibly capture the diverse vocabulary and interface semantics across different environments. To ensure the reliability of this automated segmentation, a representative subset of the generated milestones was manually reviewed. This quality assurance step confirmed that the extracted fragments accurately represent semantically complete and functionally distinct subgoals, establishing a high-quality foundation for the subsequent retrieval process.

To support efficient retrieval, each instruction and milestone is embedded into a dense vector space. The milestone library stores tuples of the form:

$$(\mathbf{v}_{\text{task}}^{(i)}, \mathbf{v}_{\text{milestone}}^{(i,k)}, \tau^{(i)}, m_k^{(i)}, \zeta_k^{(i)}), \tag{1}$$

where $\mathbf{v}_{\text{task}}^{(i)}$ and $\mathbf{v}_{\text{milestone}}^{(i,k)}$ are vector representations for the task $\tau^{(i)}$ and the milestone $m_k^{(i)}$, respectively, and $\zeta_k^{(i)}$ denotes the trajectory segment corresponding to milestone $m_k^{(i)}$. Similarity is computed via dot product over normalized embeddings.

The final milestone library is defined as:

$$\mathcal{M}_L = \bigcup_{i=1}^{N} \bigcup_{k=1}^{K^{(i)}} \left\{ (\mathbf{v}_{\text{task}}^{(i)}, \mathbf{v}_{\text{milestone}}^{(i,k)}, m_k^{(i)}, \zeta_k^{(i)}) \right\}. \tag{2}$$

The milestone library abstracts higher-level structured experiences while preserving task-specific low-level details. Compared to raw trajectory or task-level retrieval, this mid-granularity representation provides a balanced trade-off between generalizability and specificity, making it ideal for hierarchical planning under varied conditions.

4.3. Execution Phase: Hierarchical Planning and Execution

In the execution phase, HiPlan performs hierarchical planning by dynamically integrating global milestone guidance and fine-grained step-wise hints. Leveraging structured experiences from the milestone library, our framework provides an adaptive planning mechanism that maintains global consistency and facilitates local flexibility, as illustrated in Fig. 2. We assume this dual-level approach can help mitigate common issues in long-horizon task execution, such as deviation from global objectives and inability to adapt to real-time uncertainties.

Global Guidance: Milestone Action Guide. To establish strategic direction for long-horizon task completion, HiPlan provides global guidance through a dynamically generated milestone action guide. Given a test-time task instruction τ, HiPlan retrieves similar task entries from the milestone library using the embedding of τ as the query key:

$$\{(\tau^{(j)}, \xi^{(j)}, \mathcal{G}_{\tau^{(j)}})\}_{j=1}^{M} = \text{Retrieve}(\mathbf{v}_\tau) \tag{3}$$

These entries include task instructions, trajectories, and corresponding milestone sequences, which serve as references to generate a tailored guide:

$$\mathcal{G}_\tau = \text{LLM}(\tau, \{(\tau^{(j)}, \xi^{(j)}, \mathcal{G}_{\tau^{(j)}})\}_{j=1}^{M}), \tag{4}$$

where $\mathcal{G}_\tau = [m_1, m_2, \ldots, m_K]$ and each m_k is a critical subgoal adapted to the current task context.

This process aims to transfer insights from historical experiences, enabling the agent to benefit from past successful planning strategies while maintaining flexibility to handle new task dynamics.

Local Guidance: Step-Wise Hints. Complementing the global milestone structure, HiPlan provides detailed local guidance through context-awareness step-wise hints, dynamically generated at each step. At timestep t, the current milestone $m_{\psi(t)}$ is identified, and its embedding $\mathbf{v}_{m_{\psi(t)}}$ serves as the query key to retrieve similar milestones and their corresponding trajectory segments from the milestone library:

$$\{(m_l^*, \zeta_l^*)\}_{l=1}^{P} = \text{Retrieve}(\mathbf{v}_{m_{\psi(t)}}), \tag{5}$$

where m_l^* and ζ_l^* denote retrieved milestones and the trajectory segments that complete them, respectively. These retrieved elements are then used as references, combined with past action-observation pairs $\{(o_s, a_s)\}_{s=1}^{t}$, to generate the step-wise hint h_t:

$$h_t = \text{LLM}(m_k, \{(o_s, a_s)\}_{s=1}^{t}, \{(m_l^*, \zeta_l^*)\}_{l=1}^{P}). \tag{6}$$

Each hint explicitly highlights the current state context, the gap to the milestone, and, when necessary, corrections to the agent's intended actions, thus providing immediate feedback to rectify errors or inefficiencies:

$$h_t = \{\text{State Context}, \text{Milestone Gap}, \text{Action Correction}\}. \tag{7}$$

The step-wise hints dynamically track the agent's progression, recognizing when milestone subgoals are achieved and seamlessly guiding the agent toward subsequent objectives.

Regarding the formulation of the Milestone Gap and the determination of milestone completion, it is important to clarify that the Milestone Gap serves as an explicit, intermediate reasoning step performed by the LLM. The LLM infers this gap dynamically by comparing the text-based milestone description against the current textual observation and the recent interaction history. This process encourages the model to ground its internal reasoning in the immediate environmental context. Consequently, the transition between milestones is implicitly determined by this very reasoning process: when the LLM concludes during the hint generation that the Milestone Gap has been successfully closed, it signals the completion of the current subgoal, thereby triggering the framework to seamlessly advance to the next milestone in the hierarchical plan.

Dual-level Guidance Enhanced Policy. At each timestep t, the agent leverages the milestone action guide to maintain global task coherence, while simultaneously utilizing step-wise hints to adaptively transition between milestones based on real-time observations.

Formally, the integrated policy π combines these two complementary sources of guidance as:

$$a_t = \pi(\tau, \{(o_s, a_s)\}_{s=1}^{t}, m_k, h_t) \tag{8}$$

In this formulation, HiPlan performs adaptive hierarchical planning by closely integrating the global milestone action guide with dynamic local step-wise hints.

The pseudocode of HiPlan is presented in Alg. 1. For more implementation details, please refer to Appendix A and Appendix B.

5. Experiments

In this section, We conduct extensive experiments to evaluate HiPlan, aiming to answer the following research questions:

- **RQ1:** How effectively does HiPlan complete tasks and satisfy user requirements compared to baseline methods?
- **RQ2:** What are the individual contributions of HiPlan's core components to its overall performance?
- **RQ3:** Can HiPlan generalize to long-horizon tasks beyond Web environments, demonstrating broader applicability?
- **RQ4:** How efficiently does HiPlan complete web navigation tasks in terms of execution steps?
- **RQ5:** What are the primary failure modes of HiPlan, and what insights do they provide for understanding its current limitations?

Algorithm 1 HiPlan: Hierarchical Planning with Adaptive Global-Local Guidance

Require: Task instruction τ, Demonstration set $\mathcal{D} = \{(\tau^{(i)}, \xi^{(i)})\}_{i=1}^{N}$, Maximum number of steps T

// Offline Phase: Milestone Library Construction
for $(\tau^{(i)}, \xi^{(i)}) \in \mathcal{D}$ **do**
  for each fragment $\zeta_k^{(i)}$ in $\xi^{(i)}$ **do**
    $m_k^{(i)} \leftarrow \text{LLM}(\zeta_k^{(i)})$
    $(\mathbf{v}_{\text{task}}^{(i)}, \mathbf{v}_{\text{mile}}^{(i,k)}) \leftarrow \text{Embed}(\tau^{(i)}, m_k^{(i)})$
    Store $(\mathbf{v}_{\text{task}}^{(i)}, \mathbf{v}_{\text{mile}}^{(i,k)}, m_k^{(i)}, \zeta_k^{(i)})$ in $\mathcal{M}_L$
  end for
end for
// Execution Phase: Hierarchical Planning
$\mathbf{v}_\tau \leftarrow \text{Embed}(\tau)$
Retrieve $\{(\tau^{(j)}, \xi^{(j)}, \mathcal{G}_{\tau^{(j)}})\}_{j=1}^{M}$ from $\mathcal{M}_L$ using $\mathbf{v}_\tau$
$\mathcal{G}_\tau \leftarrow \text{LLM}(\tau, \{(\tau^{(j)}, \xi^{(j)}, \mathcal{G}_{\tau^{(j)}})\})$
$k \leftarrow 1$ {Milestone index}
for $t = 1$ to T **do**
  $o_t \leftarrow$ observe environment
  Retrieve $\{(m_l^*, \zeta_l^*)\}_{l=1}^{P}$ from $\mathcal{M}_L$ using $\mathbf{v}_{m_k}$
  $h_t \leftarrow \text{LLM}(m_k, \{(o_s, a_s)\}_{s=1}^{t}, \{(m_l^*, \zeta_l^*)\})$
  $a_t \leftarrow \pi(\tau, \{(o_s, a_s)\}_{s=1}^{t}, m_k, h_t)$
  Execute a_t
  if milestone m_k completed **then**
    $k \leftarrow k + 1$
  end if
  if task completed **then**
    break
  end if
end for
return $[a_1, a_2, \ldots, a_T]$

5.1. Experimental Setup

5.1.1. Datasets

We evaluate HiPlan on two challenging benchmarks for long-horizon web agent decision-making:

Mind2Web [25] is a generalist web agent benchmark covering 137 real-world websites across 31 domains. Unlike simplified simulation environments, Mind2Web challenges agents with authentic web complexity. We evaluate on 300 test tasks—100 each from its three splits: *Cross-Task* (unseen tasks on seen websites), *Cross-Website* (unseen websites within seen domains), and *Cross-Domain* (unseen domains entirely). Following the benchmark's evaluation protocol, we report three key metrics: **Element Accuracy** (fraction of correctly interacted UI elements), **Step Success Rate** (proportion of successfully completed individual steps), and **Success Rate** (end-to-end task completion).

WebShop [24] is a large-scale interactive environment that simulates an online shopping

website with over 1.18 million products. It requires agents to ground natural language instructions into sequential search and click actions to locate and purchase a target product. We evaluate on 200 test instructions, reporting both **Success Rate** (exact product match) and the average **Reward**, which reflects the degree of attribute alignment with the request.

5.1.2. Implementation Details

To construct the milestone library, we collect successful expert trajectories from both WebShop and Mind2Web. Each trajectory is segmented into key milestones using GPT-4o (2024-08-06). Task instructions and extracted milestone descriptions are encoded using the Sentence-Transformers model `all-mpnet-base-v2` and indexed via inner-product similarity for efficient retrieval.

All components of HiPlan—including milestone action guide generation, step-wise hint construction, and action prediction—are driven by the same LLM. We evaluate the framework using two open-source models: `LLaMA-3.3-70B-Instruct` (denoted as LLaMA) and Mixtral-8x22B-Instruct-v0.1 (denoted as Mixtral). These two represent distinct model architectures: Mixtral is a sparse mixture-of-experts (SMoE) model, while LLaMA is a dense LLM. We use this setting to assess the effectiveness of HiPlan across different LLM architecture.

To avoid any ambiguity regarding data leakage, we enforce a strict data separation protocol. The expert demonstrations used to construct the milestone library are drawn exclusively from the training data of the respective benchmarks. They are strictly disjoint from the evaluation tasks. This ensures that the agent encounters entirely unseen tasks and web environments during the testing phase, providing a rigorous evaluation of its generalization capabilities.

All outputs from LLMs are generated deterministically with temperature set to 0.0 to ensure consistent and reproducible results.

5.1.3. Baselines

We compare HiPlan against several representative and strong baselines to evaluate its effectiveness:

ReAct-based methods [10] represent the class of *reason-and-act* paradigm widely adopted for web navigation and interaction tasks, where reasoning and acting are interleaved through chain-of-thought prompting and action generation at each step based on immediate observations.

Reflexion-based methods [37] represent the class of *self-reflection and correction* paradigm that leverages linguistic feedback for iterative improvement across episodes. Through episodic memory maintenance and failure analysis, the agent refines its behavior via natural language reasoning without parameter updates.

TRAD [40] is a *retrieval-augmented* method that improves in-context decision-making by selecting action-level demonstrations based on state abstraction similarity, enabling the agent to draw directly on prior experience.

All baselines are evaluated under the same configuration as HiPlan, including environments and LLMs, to ensure a fair comparison.

5.2. Performance Comparison with Baselines (RQ1)

We summarize our experimental results in Tables 1 and 2, showing that HiPlan consistently outperforms all baseline methods across both Mind2Web and WebShop benchmarks. We discuss these results in detail below.

5.2.1. Mind2Web

As shown in Table 1, HiPlan demonstrates superior performance across all three splits of Mind2Web, achieving the highest Success Rate in most settings, with absolute gains up to 15

Models	Methods	Cross Task			Cross Website			Cross Domain		
		Ele. Acc	Step SR	SR	Ele. Acc	Step SR	SR	Ele. Acc	Step SR	SR
LLaMA3.3-70b	ReAct	66.06	55.20	18.00	75.31	62.35	24.00	73.82	67.67	10.00
	ReAct+Reflexion	68.75	59.33	18.00	78.94	66.22	25.00	73.82	67.67	10.00
	TRAD	70.41	**69.00**	21.00	76.42	67.48	10.00	80.78	70.75	8.00
	HiPlan	**74.66**	65.84	**24.00**	**79.95**	**73.11**	**30.00**	**81.89**	**77.72**	**22.00**
Mixtral-8x22b	ReAct	61.76	48.64	12.00	66.99	49.39	12.00	70.47	55.71	10.00
	ReAct+Reflexion	63.96	48.64	12.00	71.19	52.76	15.00	70.47	55.71	10.00
	TRAD	68.20	71.08	20.00	**79.24**	55.17	**18.00**	84.47	**74.93**	**12.00**
	HiPlan	**86.75**	**73.83**	**26.00**	78.23	**59.39**	**18.00**	**87.89**	74.43	**12.00**

Table 1: Element accuracy (Ele. Acc), step success rate (Step SR) and success rate (SR) of HiPlan and baselines on the Mind2Web benchmark.

Models	Method	Reward	Success Rate
LLaMA3.3-70b	ReAct	0.09	0.08
	ReAct+Reflexion	0.23	0.12
	TRAD	0.27	0.14
	HiPlan	**0.58**	**0.40**
Mixtral-8x22b	ReAct	0.26	0.19
	ReAct+Reflexion	0.40	0.24
	TRAD	0.10	0.04
	HiPlan	**0.50**	**0.36**

Table 2: Average reward and success rate of HiPlan and baselines on the WebShop benchmark.

points for LLaMA and up to 6 points for Mixtral compared to the strongest baselines. HiPlan consistently outperforms ReAct-based methods, which often fail to maintain coherent navigation strategies across complex multi-page workflows. Reflexion shows minimal improvements over ReAct, suggesting that episode-level reflection struggles to capture the fine-grained interaction patterns and correct nuanced errors encountered in complex web navigation. Interestingly, while TRAD exhibits strong Step Success Rates, it suffers from inconsistent final task completion, particularly in Cross-Website and Cross-Domain settings where it underperforms significantly on overall Success Rate. This highlights a critical limitation: only providing action-level demonstrations can lead agents to execute locally reasonable but globally misaligned action sequences. Under the challenging *Cross-Domain* split, where agents must adapt to entirely unseen website domains, HiPlan achieves remarkable Success Rates of 22.00% (LLaMA) and 12.00% (Mixtral). This underscores HiPlan's strong generalization capabilities in unfamiliar web environments.

5.2.2. WebShop

Results shown in Table 2 reveal a similar trend, with HiPlan achieving the highest Success Rates of 36% (Mixtral) and 40% (LLaMA), outperforming baselines by up to 32 absolute points. All baseline methods perform poorly on this environment. Interestingly, TRAD shows inconsistent performance, even underperforming ReAct and Reflexion with the Mixtral model. This

suggests that in WebShop environment, directly providing low-level action demonstrations as in-context examples can often introduce significant noise, severely misguiding the agent and preventing target product purchase within limited steps. This also indicates the sensitivity of TRAD to different task characteristics. HiPlan achieves consistent and substantial improvements in average Reward, indicating that even when HiPlan fails to find the exact target product, it can identifies alternative products that better satisfy the specified constraints, guided by the synergy between high-level and step-wise signals.

Overall, these experimental results highlight the effectiveness of HiPlan's hierarchical guidance mechanism. By leveraging structured milestone-level plans and providing dynamic global-local integration, HiPlan significantly enhances the performance of LLM-based Web agents in complex, long-horizon tasks. Its robust performance across different environments and model architectures underscores the generality and adaptability of our approach.

5.3. Ablation Studies (RQ2)

To assess the contribution of the core mechanisms in HiPlan, we conduct ablation studies across two Web environments by evaluating three variants:

- **HiPlan-Direct:** Removes both the milestone action guide and step-wise hints, relying solely on direct action generation from task instructions, task-level similar demonstrations and observations.
- **HiPlan-Milestone:** Retains only the high-level milestone action guide while removing the step-wise hints.
- **HiPlan-w/o milestone-level demonstrations:** Includes both milestone action guides and step-wise hints, but constructs step-wise hints without milestone-level similar trajectory fragments as references from the milestone library.

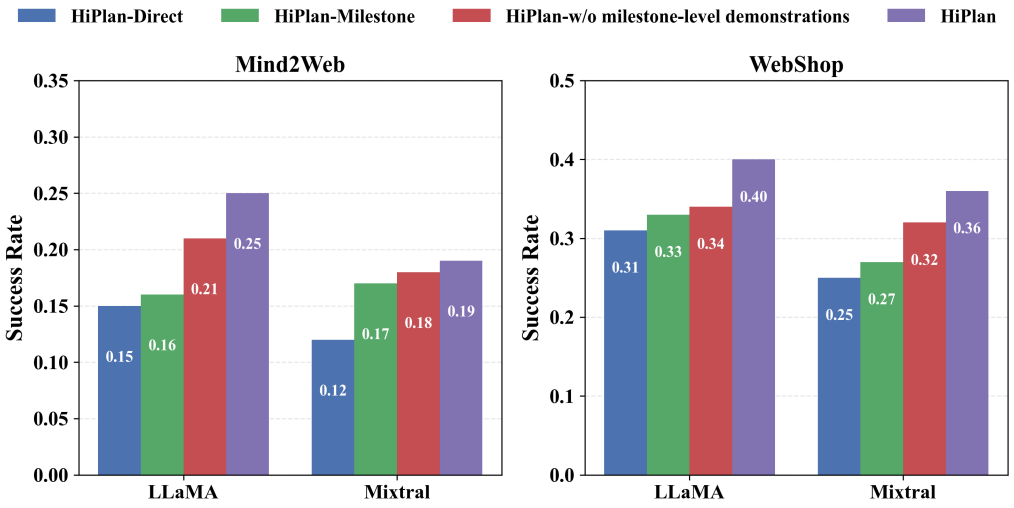

Figure 3: Ablation study results across Mind2Web and WebShop with Mixtral and LLaMA. HiPlan consistently outperforms all ablated variants, validating the importance of hierarchical guidance and the reuse of milestone-level demonstrations.

Models	Method	Put	Examine	Clean	Heat	Cool	PutTwo	All
Mixtral-8x22b	ReAct	0.46	0.89	0.55	0.83	0.62	0.18	0.59
	ReAct+Reflexion	0.77	0.89	0.61	0.83	0.62	0.29	0.64
	TRAD	0.75	0.89	**0.84**	0.70	**0.95**	0.53	0.78
	HiPlan	**0.92**	**0.94**	**0.84**	**0.87**	0.71	**0.59**	**0.82**
LLaMA3.3-70b	ReAct	0.50	0.83	0.39	0.65	0.48	0.18	0.50
	ReAct+Reflexion	0.58	0.83	0.48	0.65	0.57	0.24	0.56
	TRAD	0.96	0.89	0.90	0.65	0.67	0.59	0.79
	HiPlan	**1.00**	**1.00**	**0.97**	**0.91**	**0.90**	**0.82**	**0.94**

Table 3: Task success rates of HiPlan and baselines on the ALFWorld benchmark across six task types.

The results in Fig. 3 demonstrate that all variants underperform HiPlan across environments and LLMs, which highlight several key findings:

Hierarchical Guidance is Crucial. While HiPlan-Milestone provides modest improvements over the direct setting, the combination with step-wise hints yields substantial gains (7-10% points in Mind2Web, 9-11% points in WebShop). These results validate the synergy effect of HiPlan's dual-level guidance.

Milestone Demonstrations Enhance Planning Quality. The variant HiPlan-w/o milestone-level demonstrations shows consistent performance drops. This validates the effect of leveraging milestone-level similar experiences from the milestone library rather than providing step-wise hints from scratch.

Robust Generalizability. The performance gains from HiPlan are consistent across both Mind2Web and WebShop, despite their distinct characteristics—generalist web navigation versus online shopping. Both LLMs exhibit consistent cumulative effects from the components. This suggests that the benefits of hierarchical guidance and experiences reuse generalize across different scenarios and LLMs.

5.4. Generalization Beyond Web Tasks (RQ3)

While this work focuses on enhancing planning for Web agents, we also study whether the core design of HiPlan generalizes to other complex, long-horizon domains beyond the Web. To this end, we conduct additional experiments on ALFWorld [43], a benchmark that challenges an agent's ability to perform complex, multi-step tasks in an embodied household environment. ALFWorld comprises six distinct task types: Pick & Place, Pick Two & Place, Examine in Light, Clean & Place, Heat & Place, and Cool & Place, testing an agent's capacity for understanding object states and interactions, and executing long action sequences. Our evaluation is conducted on the split of 134 out-of-distribution tasks.

Setup. We adopt the same experimental settings as in the main experiments. A separate milestone library is constructed from expert ALFWorld demonstrations, following the same procedure.

Results. Table 3 presents the results on ALFWorld. HiPlan achieves the highest success rates across all task categories, demonstrating substantial improvements over baselines with absolute gains ranging from 4 to 23 points for Mixtral and 15 to 44 points for LLaMA. Notably, HiPlan shows larger improvements on challenging tasks like *PutTwo*, which involves multiple target objects and requires longer action sequences. The consistent improvements demonstrate the

broader applicability of HiPlan's hierarchical planning and dual-level experience reuse for long-horizon decision-making in diverse environments.

5.5. Execution Efficiency Analysis (RQ4)

In addition to task success rates, a critical measure of an agent's planning capability is its efficiency, i.e., the ability to complete tasks in a minimal number of steps. Inefficient planning often leads to redundant actions, error-prone exploration, and failure to complete tasks within a limited step-horizon. We evaluate step efficiency on WebShop and ALFWorld, as these benchmarks allow for variable-length episodes where step count directly reflects planning efficiency. Mind2Web employs fixed-length episodes in its evaluation protocol, and we exclude it in this analysis.

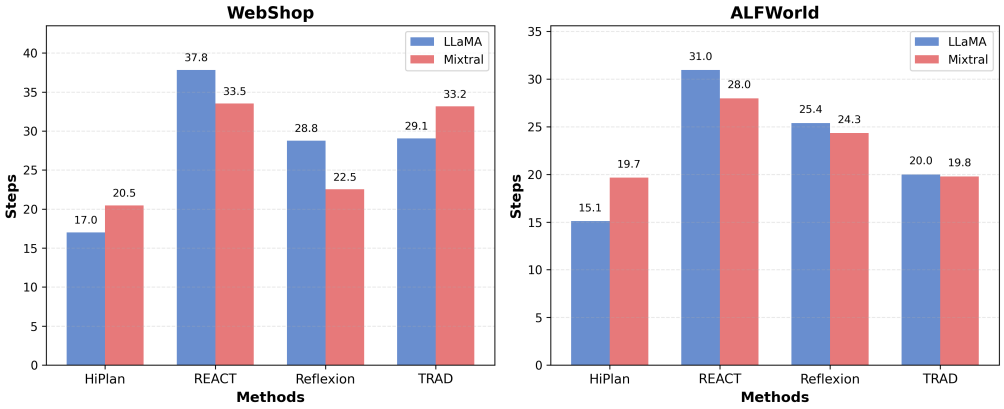

Figure 4: Average steps to task completion on WebShop and ALFWorld. HiPlan consistently achieves goals in fewer steps than baseline methods across both benchmarks, highlighting its superior planning efficiency.

Fig. 4 illustrates the average number of steps taken by different methods across the Webshop and ALFWorld benchmarks. HiPlan consistently requires significantly fewer steps to complete tasks compared to all baselines. On average, HiPlan achieves a remarkable reduction of 37% in steps on WebShop and 28% on ALFWorld

WebShop. A similar trend is observed on WebShop. HiPlan reduces the planning steps by up to 55% compared to ReAct and by over 40% relative to Reflexion and TRAD.

ALFWorld. HiPlan outperforms all baselines by a significant margin, reducing the average number of steps by up to 51% compared to ReAct and by up to 41% compared to Reflexion. Even against the strong retrieval-based TRAD baseline, HiPlan achieves a 25% reduction in required steps. The improvements hold consistently for both LLaMA and Mixtral backbones.

Analysis. We attribute the superior step efficiency of HiPlan to its hierarchical guidance mechanism: global milestone action guides maintain strategic direction, while adaptive step-wise hints correct deviations and eliminate redundant exploration. This dual-level guidance enables agents to advance towards goals in a more focused and error-averse manner, leading to not only higher success rates but also markedly shorter trajectories. In contrast, ReAct often exhibits meandering behavior due to its reactive nature, and Reflexion is prone to repeated similar mistakes as it relies on post-hoc, episode-level feedback after a failure. While TRAD shows competitive

performance on task success rate, it still requires more steps to complete tasks than HiPlan, suggesting that its low-level trajectory demonstrations may guide agents along suboptimal planning paths.

While the dual-level retrieval and intermediate LLM reasoning in HiPlan inevitably increase the token consumption and inference latency per individual step compared to reactive baselines, this added per-step complexity is a justifiable trade-off. HiPlan effectively prevents the agent from pursuing lengthy suboptimal trajectories or falling into infinite loops, thereby significantly reducing the total number of execution steps. As a result, the overall time-to-completion and total API costs for an entire task remain highly competitive, while achieving a substantial improvement in the overall success rate and task effectiveness.

The substantial drop in completion steps across heterogeneous tasks and LLMs confirms the generalizability and effectiveness of HiPlan's design for long-horizon, complex decision-making tasks.

5.6. Error Mode Analysis (RQ5)

To answer RQ5 and better understand the limitations of our framework, we conduct a qualitative error mode analysis on the failed trajectories. We summarize two primary failure modes (with detailed step-by-step case studies provided in Appendix C):

(1) Specification Drift in Ultra-Long Horizons: While the milestone guide provides strong directional boundaries, in tasks requiring extremely long execution steps, the agent occasionally loses track of fine-grained constraints specified in the initial instruction (e.g., forgetting a specific color preference while navigating through multiple product pages).

(2) Underutilization of Subtle Environmental Clues: HiPlan occasionally fails to ground its step-wise hints in the most subtle interactive elements of the current html observation. This leads to repetitive actions or incorrect element selection when multiple similar UI components are present.

Recognizing these failure modes highlights critical directions for future work, particularly in enhancing long-context state tracking and developing more robust fine-grained textual grounding mechanisms.

6. Conclusion

In this paper, we introduced HiPlan, a hierarchical planning framework that enhances LLM-based Web agents' ability to tackle complex, long-horizon tasks by synergistically combining global milestone action guides with adaptive step-wise hints. This integration addresses critical challenges in maintaining global coherence while adapting actions to dynamic local contexts, leveraging dual-level experience reuse to balance generalization and specificity effectively. Our extensive experiments demonstrate that HiPlan significantly improves success rates and robustness across diverse benchmarks and backbone LLMs, outperforming strong baselines. Furthermore, it is worth noting that our dual-level experience reuse mechanism operates entirely in a tuning-free manner. This significant advantage eliminates the need for computationally expensive parameter updates while maintaining strong generalization capabilities across diverse web tasks.

While HiPlan marks a substantial advancement, several avenues remain for future exploration. One promising direction is extending the framework to a broader range of tasks and domains beyond Web to assess its generalizability and scalability. Additionally, we plan to investigate methods for summarizing and abstracting experience gained through step-wise hints, enabling effective cross-task knowledge transfer and improving adaptation in novel scenarios.

We believe that HiPlan represents a meaningful step toward empowering LLM-based Web agents with robust hierarchical reasoning and adaptive planning capabilities. By bridging global guidance and fine-grained adaptability, it lays the groundwork for more scalable, flexible, and intelligent autonomous Web agent systems capable of operating effectively in complex and dynamic real-world Web environments.

References

[1] M. Kobayashi and K. Takeda, "Information retrieval on the web," *ACM Comput. Surv.*, vol. 32, no. 2, p. 144–173, 2000. [Online]. Available: https://doi.org/10.1145/358923.358934

[2] M. Carbone, K. Honda, and N. Yoshida, "Structured communication-centered programming for web services," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 34, no. 2, pp. 1–78, 2012.

[3] M. Khalifa and M. Limayem, "Drivers of internet shopping," *Communications of the ACM*, vol. 46, no. 12, pp. 233–239, 2003.

[4] R. S. Toh, C. F. DeKay, and P. Raven, "Travel planning: searching for and booking hotels on the internet," *Cornell Hospitality Quarterly*, vol. 52, no. 4, pp. 388–398, 2011.

[5] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried *et al.*, "Webarena: A realistic web environment for building autonomous agents," *arXiv preprint arXiv:2307.13854*, 2023.

[6] L. Ning, Z. Liang, Z. Jiang, H. Qu, Y. Ding, W. Fan, X.-y. Wei, S. Lin, H. Liu, P. S. Yu *et al.*, "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*, 2025, pp. 6140–6150.

[7] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[8] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[9] H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, and D. Yu, "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)*, 2024, pp. 6864–6890. [Online]. Available: https://aclanthology.org/2024.acl-long.371/

[10] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=WE_vluYUL-X

[11] R. Lo, A. Sridhar, F. Xu, H. Zhu, and S. Zhou, "Hierarchical prompting assists large language model on web navigation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 10 217–10 244. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.685/

[12] D. Lee, J. Lee, K. Kim, J. Tack, J. Shin, Y. W. Teh, and K. Lee, "Learning to contextualize web pages for enhanced decision making by llm agents," in *International Conference on Representation Learning*, vol. 2025, 2025, pp. 81 039–81 072.

[13] T. Li, G. Li, Z. Deng, B. Wang, and Y. Li, "A zero-shot language agent for computer control with structured reflection," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 11 261–11 274. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.753/

[14] I. Gur, H. Furuta, A. Huang, M. Safdari, Y. Matsuo, D. Eck, and A. Faust, "A real-world webagent with planning, long context understanding, and program synthesis," in *International Conference on Representation Learning*, vol. 2024, 2024, pp. 52 690–52 717.

[15] Y. Zhang, Z. Ma, Y. Ma, Z. Han, Y. Wu, and V. Tresp, "Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 22, pp. 23 378–23 386, Apr. 2025. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/34505

[16] L. E. Erdogan, H. Furuta, S. Kim, N. Lee, S. Moon, G. Anumanchipalli, K. Keutzer, and A. Gholami, "Plan-and-act: Improving planning of agents for long-horizon tasks," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=ybA4EcMmUZ

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, brian ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, 2022. [Online]. Available: https://openreview.net/forum?id=_VjQlMeSB_J

[18] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. R. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=5Xc1ecxO1h

[19] Z. Li, Y. Chang, and X. Le, "Simulating expert discussions with multi-agent for enhanced scientific problem solving," in *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, 2024, pp. 243–256. [Online]. Available: https://aclanthology.org/2024.sdp-1.23/

[20] Y. Xu, X. Liu, X. Liu, Z. Hou, Y. Li, X. Zhang, Z. Wang, A. Zeng, Z. Du, Z. Wenyi, J. Tang, and Y. Dong, "ChatGLM-math: Improving math problem-solving in large language models with a self-critique pipeline," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 9733–9760. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.569/

[21] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling large language models to generate text with citations," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 6465–6488. [Online]. Available: https://aclanthology.org/2023.emnlp-main.398/

[22] Y. Chang, Z. Li, H. Zhang, Y. Kong, Y. Wu, Z. Guo, and N. Wong, "Treereview: A dynamic tree of questions framework for deep and efficient llm-based scientific peer review," 2025. [Online]. Available: https://arxiv.org/abs/2506.07642

[23] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Liu, Z. Dou, and J.-R. Wen, "Large language models for information retrieval: A survey," *ACM Trans. Inf. Syst.*, 2025, just Accepted. [Online]. Available: https://doi.org/10.1145/3748304

[24] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20 744–20 757, 2022.

[25] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2web: Towards a generalist agent for the web," *Advances in Neural Information Processing Systems*, vol. 36, pp. 28 091–28 114, 2023.

[26] J. Liu, J. Hao, C. Zhang, and Z. Hu, "Wepo: Web element preference optimization for llm-based web navigation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 25, 2025, pp. 26 614–26 622.

[27] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su, "Gpt-4v(ision) is a generalist web agent, if grounded," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[28] S. Sun, Y. Liu, S. Wang, D. Iter, C. Zhu, and M. Iyyer, "PEARL: Prompting large language models to plan and execute actions over long documents," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 469–486. [Online]. Available: https://aclanthology.org/2024.eacl-long.29/

[29] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 2609–2634. [Online]. Available: https://aclanthology.org/2023.acl-long.147/

[30] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=_nGgzQjzaRy

[31] C. Shao, X. Hu, Y. Lin, and F. Xu, "Division-of-thoughts: Harnessing hybrid language model synergy for efficient on-device agents," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 1822–1833.

[32] A. Prasad, A. Koller, M. Hartmann, P. Clark, A. Sabharwal, M. Bansal, and T. Khot, "ADaPT: As-needed decomposition and planning with language models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 4226–4252. [Online]. Available: https://aclanthology.org/2024.findings-naacl.264/

[33] K. Yang, Y. Liu, S. Chaudhary, R. Fakoor, P. A. Chaudhari, G. Karypis, and H. Rangwala, "Agentoccam: A simple yet strong baseline for llm-based web agents," in *International Conference on Representation Learning*, vol. 2025, 2025, pp. 97 533–97 565.

[34] L. Zheng, R. Wang, X. Wang, and B. An, "Synapse: Trajectory-as-exemplar prompting with memory for computer control," in *International Conference on Representation Learning*, vol. 2024, 2024, pp. 19 036–19 066.

[35] A. Zhao, D. Huang, Q. Xu, M. Lin, Y.-J. Liu, and G. Huang, "Expel: Llm agents are experiential learners," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, pp. 19 632–19 642.

[36] M. Hu, T. Chen, Q. Chen, Y. Mu, W. Shao, and P. Luo, "Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model," 2024. [Online]. Available: https://arxiv.org/abs/2408.09559

[37] N. Shinn, F. Cassano, A. Gopinath, K. R. Narasimhan, and S. Yao, "Reflexion: language agents with verbal reinforcement learning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=vAElhFcKW6

[38] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022, pp. 100–114. [Online]. Available: https://aclanthology.org/2022.deelio-1.10/

[39] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proceedings of the 61st Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 10 014–10 037. [Online]. Available: https://aclanthology.org/2023.acl-long.557/

[40] R. Zhou, Y. Yang, M. Wen, Y. Wen, W. Wang, C. Xi, G. Xu, Y. Yu, and W. Zhang, "Trad: Enhancing llm agents with step-wise thought retrieval and aligned decision," in *SIGIR*, 2024, pp. 3–13. [Online]. Available: https://doi.org/10.1145/3626772.3657788

[41] M. Kim, V. Bursztyn, E. Koh, S. Guo, and S.-w. Hwang, "RaDA: Retrieval-augmented web agent planning with LLMs," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 13 511–13 525. [Online]. Available: https://aclanthology.org/2024.findings-acl.802/

[42] Z. Wang, S. X. Teo, J. J. Chew, and W. Shi, "Instructrag: Leveraging retrieval-augmented generation on instruction graphs for llm-based task planning," 2025. [Online]. Available: https://arxiv.org/abs/2504.13032

[43] M. Shridhar, X. Yuan, M.-A. Cote, Y. Bisk, A. Trischler, and M. Hausknecht, "{ALFW}orld: Aligning text and embodied environments for interactive learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=0IOX0YcCdTn

Appendix A. Milestone Library Construction Details

Metric	ALFWorld	WebShop	Mind2Web	Total
Source demonstrations	500	77	100	677
Avg. milestones per trajectory	5.89	5.00	5.92	5.77
Avg. actions per milestone	1.78	1.11	1.33	1.70
Total entries	2,945	385	592	3,922

Table A.4: Statistics of the constructed milestone library across ALFWorld, WebShop and Mind2Web benchmarks.

The construction of milestone library relies on an automated pipeline that extracts meaningful subgoals from expert demonstration trajectories. We employ GPT-4o to segment each trajectory into semantically coherent milestones using the prompt shown in Appendix D.

Appendix A.1. Milestones Extraction and Trajectory Segmentation

The extraction process operates on successful task demonstrations $\mathcal{D} = \{(\tau^{(i)}, \xi^{(i)})\}_{i=1}^{N}$, where each trajectory $\xi^{(i)} = \{(o_1^{(i)}, a_1^{(i)}), \ldots,$
$(o_{T^{(i)}}^{(i)}, a_{T^{(i)}}^{(i)})\}$ corresponds to task $\tau^{(i)}$. For each trajectory, the LLM identifies key milestones and maps trajectory segments to corresponding milestones.

For each identified milestone $k \in \{1, \ldots, K^{(i)}\}$, the LLM provides:

- A description $m_k^{(i)}$ capturing the subgoal
- Action indices $\mathcal{I}_k^{(i)} = \{t_{start}^{(k)}, \ldots, t_{end}^{(k)}\}$ corresponding to the trajectory segment

The corresponding trajectory segment is then extracted as:

$$\zeta_k^{(i)} = \{(o_t^{(i)}, a_t^{(i)}) : t \in \mathcal{I}_k^{(i)}\} \tag{A.1}$$

This segmentation ensures that each $\zeta_k^{(i)}$ represents a cohesive sub-trajectory that accomplishes a specific intermediate objective toward the overall task completion.

Appendix A.2. Indexing and Storage

The constructed milestone library maintains a hierarchical organization with two-level indexing. At the task level, entries are indexed by task embedding vectors $v_{\text{task}}^{(i)}$ for efficient task-level retrieval. At the milestone level, individual milestones are indexed by their semantic embeddings $v_{\text{milestone}}^{(i,k)}$, enabling fine-grained milestone-level retrieval. Each entry stores tuples of the

form $(v^{(i)}_{\text{task}}, v^{(i,k)}_{\text{milestone}}, m^{(i)}_k, \xi^{(i)}_k)$. All embeddings are computed using the SentenceTransformers model `all-mpnet-base-v2`.

The detailed statistics of the constructed milestone library for all benchmarks are summarized in Table A.4.

Appendix B. Experience Retrieval Implementation

This section provides detailed implementation of the experience retrieval mechanisms used in HiPlan for generating global milestone action guides and step-wise hints.

Appendix B.1. Task-Level Retrieval for Global Guidance

For generating the milestone action guide G_τ, HiPlan retrieves $M = 2$ most similar tasks from the milestone library. The retrieval process follows these steps:

Task Embedding and Similarity Computation: Given a test-time task instruction τ, we first encode it into a dense vector using SentenceTransformers (`all-mpnet-base-v2`):

$$v_\tau = \text{Embed}(\tau) \tag{B.1}$$

The similarity between the query task and each stored task is computed using normalized dot product:

$$\text{sim}(v_\tau, v_{\text{task}^{(i)}}) = \frac{v_\tau \cdot v_{\text{task}^{(i)}}}{\|v_\tau\|\|v_{\text{task}^{(i)}}\|} \tag{B.2}$$

Ranking and Selection: Tasks are ranked by similarity in descending order, and the top-$M = 2$ most similar tasks are selected:

$$\{(\tau^{(j)}, \xi^{(j)}, G_{\tau^{(j)}})\}_{j=1}^{M} = \text{TopK}(\text{sim}(v_\tau, v_{\text{task}^{(i)}}), M) \tag{B.3}$$

Length-based Re-ranking: The retrieved demonstrations are further re-ranked by trajectory length to prioritize shorter, more generalizable examples that provide clearer structural guidance.

Appendix B.2. Milestone-Level Retrieval for Local Guidance

For generating step-wise hints h_t, HiPlan retrieves top-$P = 2$ most similar milestones from the milestone library. The process follows similar embedding and similarity computation steps as global guidance retrieval, but operates at the milestone-level rather than task-level.

Trajectory-level Deduplication: Since similar milestones may exist within the same trajectory and could be recalled together, we enforce that no two selected milestones come from the same expert trajectory to ensure the diversity of demonstrations. This prevents over-reliance on specific trajectories and enhances generalization.

Trajectory Segment Recovery: For each retrieved milestone, we access the corresponding pre-stored trajectory segment that completes the milestone. For each trajectory segment, we include one additional action step to provide forward-looking context.

We set $M = 2$ for global guidance retrieval and $P = 2$ for step-wise hint retrieval based on empirical observations. These values align with established practices in in-context learning, where 2-3 examples typically yield optimal performance without introducing excessive noise or computational overhead.

Appendix C. Detailed Case Studies of Failure Modes

As summarized in Section 5.6, we identified two primary failure modes of HiPlan. In this appendix, we provide the detailed qualitative case studies and complete execution trajectories to further illustrate these failure modes.

Despite the overall strong performance of HiPlan across diverse long-horizon tasks, we observe certain failure cases that highlight challenges in reasoning under partial observability and language ambiguity. In this section, we analyze the common error modes of HiPlan on the WebShop benchmark.

Specification Drift in Product Selection.. Some agents fail to strictly adhere to all constraints specified in the instruction, such as size, material, or color. In one example, although the user explicitly requested a product with "6.76 fl oz" volume, the agent skipped the option selection step and directly purchased the default variant. This behavior stems from partial satisfaction of constraints (e.g., product match but wrong size) and suggests limitations in multi-attribute reasoning under ambiguous product descriptions.

Underutilization of Available Information.. In certain tasks, agents overlook available high-quality candidates presented early in the session. For example, a correct item appeared in the first search results but was ignored in favor of less suitable alternatives. In some cases, agents also skipped detailed product descriptions and features, leading to premature purchase decisions that violated key constraints like "lace closure" or "water resistance." These failures indicate occasional overemphasis on surface cues (e.g., color or title keywords) over deeper semantic validation.

These cases help reveal not only the complexity of the task environments but also opportunities for further enhancing hierarchical planning and adaptive reasoning.

Appendix D. Prompts used in HiPlan

Milestone extraction and trajectory segmentation for WebShop

You are given a web shopping task and its successful action trajectory. Your job is to classify each action in the trajectory into one of the following standardized shopping milestones:

1. "Search phase: Formulating and executing search query" - Actions related to creating and executing search queries.
2. "Browse phase: Examining search results and selecting items" - Actions related to browsing search results and clicking on specific items.
3. "Evaluation phase: Assessing product details" - Actions related to examining product features, descriptions, reviews, or other details.
4. "Configuration phase: Selecting product options" - Actions related to selecting specific options like color, size, quantity, etc.
5. "Purchase phase: Completing the transaction" - Actions related to buying the product.

Instructions:
- Each action must be assigned to exactly one of these standardized milestones.
- Focus on the action's purpose rather than the specific content.
- For each milestone, provide a list of action indices from the trajectory that belong to that milestone.
- If no actions correspond to a particular milestone, include the milestone with an empty actions list.
- The output should be a JSON array, where each object contains:
    - "milestone": one of the five standardized milestone descriptions.
    - "actions": a list of action indices from the trajectory that fulfill this milestone.

Example 1:
Input:
- Task: i am looking for long sleeve men t-shirt.and please also choose the black one, and price lower than 40.00 dollars
- Trajectory: [TRAJECTORY EXAMPLE1]
Output: [MILESTONES EXAMPLE1]

Example 2:
Input:
- Task: i am looking for height adjustable blue color children's study desk table chair set with drawer and bookstand, and price

lower than 140.00 dollars
- Trajectory: [TRAJECTORY EXAMPLE2]
Output: [MILESTONES EXAMPLE2]

Input:
Task: {TASK}
Trajectory: {TRAJECTORY}

Now, please categorize the actions into the five standardized milestones:

Step-wise hint generation for Webshop

You are an assistant guiding an agent that performs online shopping tasks in a simulated webshop environment. Your task is to generate a step-wise hint that helps the agent complete the current milestone.

WebShop tasks follow 5 milestones:
1. Search phase: Formulating and executing search query
2. Browse phase: Examining search results and selecting items
3. Evaluation phase: Assessing product details
4. Configuration phase: Selecting product options
5. Purchase phase: Completing the transaction

IMPORTANT CONSTRAINTS:
- The agent CANNOT click [Next] to view more search results. They must find a suitable product on the first page.
- Pay VERY CLOSE ATTENTION to the current observation - only suggest actions that are actually available in the current state.
- CAREFULLY ANALYZE THE ENTIRE TRAJECTORY HISTORY to avoid suggesting actions that have already been taken.

Analyze the agent's current state, the milestone-based guide, and similar trajectories, and generate the step-wise hint including only the following fields:
- Current State: VERY PRECISELY identify the exact page type the agent is currently on.
- Current Milestone: Specify which of the milestones the agent should currently be addressing. WebShop milestones can be addressed non-sequentially if needed.
- Milestone Gap: First analyze what still needs to be done to complete the current milestone (the gap). Then, based on this analysis, recommend ONLY the NEXT single action in exact format.
- Action Correction (optional): Include this field ONLY if the MOST RECENT action resulted in "Invalid action!" error. Point out exactly why the action was invalid and specify the correct action to take. OMIT THIS FIELD ENTIRELY if the most recent action was valid.

Format your output as follows (omit Action Correction if not needed):
Current State: Describe exact page type
Current Milestone: Milestone X – description
Milestone Gap: Analysis of what's needed to complete milestone + ONLY the next immediate action in exact format
Action Correction: Only if the MOST RECENT action resulted in "Invalid action!"

Here are several examples: [FOUR FIXED EXAMPLES FOR STEP-WISE HINTS GENERATION]

Your Input: Current Task: {TASK}
Current Trajectory: {TRAJECTORIES}
Similar Trajectories: {MILESTONE-LEVEL DEMONSTRATIONS}

Now, please generate the hint for the next action:

Next action prediction for Webshop

Task: Complete online shopping tasks by searching for products, navigating product pages, selecting appropriate options, and making purchases.

Action Space:
1. search[query]: Search for products using specific keywords related to the shopping requirement.
2. click[button/option]: Click on buttons, product links, or options.

IMPORTANT CONSTRAINTS:
- You CANNOT click [Next] to view more search results. You must find an appropriate product on the first page.
- Focus on products that most closely match the requirements in the task.

You can refer to the following milestone-based action guide proposed for this task: [MILESTONE ACTION GUIDE]
Here are examples of similar shopping tasks: [TASK-LEVEL DEMONSTRATIONS]
Your current shopping task and actions taken so far: [TRAJECTORIES]
You can follow the hint for the next action: [STEP-WISE HINT]

Now, take the next action for your shopping task. Provide only the action in the format specified above:
Action:

Milestone extraction and trajectory segmentation for Mind2Web

You are given a web navigation task and its successful action trajectory. Your job is to:
1 Identify the key milestones (sub-goals or logical steps) necessary to complete the task.
2 Divide the action trajectory into segments, where each segment corresponds to a milestone.
3 For each milestone, list the indices of the actions (from the trajectory) that belong to that milestone.

Instructions:
- Only consider the actions (ignore the observations) when mapping actions to milestones.
- Each milestone should represent a clear, meaningful sub-goal within the overall task.
- A milestone can contain one or multiple consecutive actions that work together to achieve the same sub-goal.
- Actions within a milestone should be logically related and work towards the same objective.
- The output should be a JSON array, where each object contains:
    - "milestone": a concise description of the sub-goal.
    - "actions": a list of action indices (can be a single index or multiple consecutive indices).

Example 1:
Input:
- Task: Find the cheapest Keen branded new pair of women's hiking shoes in the color brown, number 9, available in Columbus, Ohio store, average rating must be above 4.
- Trajectory: [TRAJECTORY EXAMPLE1]
Output: [MILESTONES EXAMPLE1]

Example 2:
Input:
- Task: Find a train with the shortest trip from New York, NY to Washington DC on June 1 in first class and add the ticket to the cart, preferably in the morning.
- Trajectory: [TRAJECTORY EXAMPLE2]
Output: [MILESTONES EXAMPLE2]

Input:
Task: {TASK}
Trajectory: {TRAJECTORY}

Now, please generate the milestone list and map the actions to each milestone in the same format:

Milestone action guide generation for Mind2Web

You are a professional planner specializing in breaking down complex web navigation tasks into clear, milestone-driven action guides based on expert examples and initial page observations.

Your instructions:
- **CRITICAL**: First analyze the Initial Observation to understand the current page state, available elements, and navigation structure.
- Based on the page state, identify the high-level strategic milestones needed to accomplish the task.
- Each milestone should represent a major functional goal (e.g., "Access search functionality", "Navigate to product category", "Complete user authentication") rather than specific UI interactions.
- Study the provided example(s) and match their strategic thinking and milestone structure.
- Focus on WHAT needs to be achieved at each stage, not HOW to achieve it (avoid specific click/type instructions).
- Organize milestones in logical sequence based on the current page state and task requirements.

Example: [EXAMPLES]
Task: [TASK]

Based on the Initial Observation, analyze the current page state and outline high-level strategic milestones for accomplishing the task (no unnecessary explanations).
Milestone action guide:

Step-wise hint generation for Mind2Web

You are an assistant guiding an agent that performs web navigation tasks in Mind2Web dataset. Your task is to generate a step-wise hint that helps the agent for each action step.

Here is the action space:

1. ‘CLICK [id]‘: Click on an HTML element with its id.
2. ‘TYPE [id] [value]‘: Type a string into the element with the id.
3. ‘SELECT [id] [value]‘: Select a value for an HTML element by its id.

Your hint must be based on the following:
- Current Task: The overall task objective.
- Current Trajectory: The sequence of actions and observations up to the current step.
- Milestone Action Guide: A list of milestones required to complete the task.
- Similar Trajectories: Retrieved segments of successful past action/observation trajectories matching the current task and state, including those aligned with the most similar milestone.
Do not use or reference specific element IDs from these similar trajectories when generating hints for the current task.

Critical Analysis Steps:
1. **State Analysis**: Carefully examine the latest observation to understand the current page state and available elements.
2. **Action Sequence Analysis**: Consider the previous action and its expected result to understand the current context.
3. **Milestone Progress Assessment**: Determine which milestone is currently active based on the page state, not just the action sequence.
4. **Element Type Recognition**: Identify the types of interactive elements in the current observation:
   - ‘¡input¿‘ elements: Use TYPE action
   - ‘¡button¿‘ elements: Use CLICK action
   - ‘¡select¿‘ elements: Use SELECT action
   - Other clickable elements: Use CLICK action
5. **Action Type Inference**: Based on element types and milestone requirements, determine the appropriate action type. Pay special attention to dropdown implementations - some use ‘¡select¿‘ (SELECT action) while others use ‘¡ul¿¡li¿‘ (CLICK action).

Guidelines:
- Be concise and specific while maintaining accuracy.
- Never guess or infer element locations from similar trajectories; only use what is observed in the current trajectory.
- Always relate feedback to the current milestone; once a milestone is complete, advance to the next.
- Focus on helping the agent locate the correct element on the current page.
- Provide clear, actionable guidance for the next step.
- When suggesting elements, verify they are actually present in the current observation.
- For TYPE actions, consider common input patterns (e.g., city names, dates, search terms) and provide format guidance.
- **IMPORTANT**: Always analyze the HTML structure to determine the correct action type. Look at the element tags and attributes, not just the context or similar examples.

Format your output as follows:
Current State: Brief description of agent’s relevant context based on the latest observation Current Milestone: The current goal that needs to be achieved. Milestone X – description Milestone Gap: What still needs to be done, grounded only in observed information and milestones
Next Action: Specific action type based on HTML element analysis, target description with EXACT element ID from the current observation, and input format guidance if applicable. Analyze the HTML structure first to determine the correct action type - look at the element tags (‘¡input¿‘, ‘¡button¿‘, ‘¡select¿‘, ‘¡li¿‘, etc.) to choose between CLICK, TYPE, or SELECT.

Here are several examples: [THREE FIXED EXAMPLES FOR STEP-WISE HINTS GENERATION]

Your Input: Current Task: {TASK}
Current Trajectory: {TRAJECTORIES}
Similar Trajectories: {MILESTONE-LEVEL DEMONSTRATIONS}

Now, please generate the hint for the next action (no unnecessary explanations). Output:

Next action prediction for Mind2Web

You are a large language model trained to navigate the web. You will be given a task, an observation, and your previous actions. Each time you should output the next action and wait for the next observation. Here is the action space:
1. ‘CLICK [id]‘: Click on an HTML element with its id. 2. ‘TYPE [id] [value]‘: Type a string into the element with the id. 3. ‘SELECT [id] [value]‘: Select a value for an HTML element by its id.

ACTION TYPE GUIDELINES: - Use ‘CLICK‘ for: buttons, links, list items (‘¡li¿‘), clickable divs, and most interactive elements - Use ‘TYPE‘ for: input fields (‘¡input type=”text”¿‘, ‘¡input type=”email”¿‘, etc.) - Use ‘SELECT‘ for: dropdown select elements (‘¡select¿‘ with ‘¡option¿‘ children)

CRITICAL: You must use the exact element ID from the current observation. Never use placeholder text like ”[time frame input field id]” or ”[element id]”. Always look for the specific numeric ID in the HTML elements.

You can refer to the following milestone-based action guide proposed for this task to take action: [MILESTONE ACTION GUIDE]

Here are some expert demonstrations, follow these demonstrations and make your decision: [TASK-LEVEL DEMONSTRATIONS]

Your task and trajectories are as follows:: [TRAJECTORIES]

You can follow the hint below to generate your action: [STEP-WISE HINT]

IMPORTANT: Even if the hint contains placeholder text like ”[element id]”, you must find the actual numeric ID from the current observation. Scan the HTML carefully for interactive elements and use their exact IDs. In your output, the action should be quoted by a pair of ’‘.

Now, take the next action for your task (no unnecessary explanations):

Action: