

000 DEPART: HIERARCHICAL MULTI-AGENT SYSTEM 001 FOR MULTI-TURN INTERACTION 002

003 **Anonymous authors**

004 Paper under double-blind review

005 ABSTRACT

006 Large Language Models (LLMs) excel at short-horizon tasks but struggle in complex,
007 long-horizon scenarios involving multi-turn interactions, multi-step reasoning, and selec-
008 tive multi-modal perception. Two core challenges in these settings are effective long-term
009 planning and mitigating cross-modal distraction. Our empirical analysis shows that single
010 LLM agent exhibits steep performance drops as interaction steps increase, underscoring the
011 limitations of monolithic approaches. To overcome these challenges, we propose **DEPART**,
012 a hierarchical multi-agent framework that decomposes planning, action execution, and
013 visual understanding into specialized agents. Through its **D**ivide, **E**valuate, **P**lan, **A**ct,
014 **R**eflect, and **T**rack cycle, **DEPART** supports dynamic task decomposition, feedback-driven
015 adaptation, and selective vision grounding to reduce cost and improve robustness. Building
016 on this architecture, we introduce Hierarchical Interactive Multi-turn Policy Optimization
017 (HIMPO), a two-round post-training strategy that alternately optimizes planner and ex-
018 ecutor with dense role-specific and sparse task-level rewards to encourage specialization
019 and coordinated long-horizon reasoning. Across WebArena-Lite, VisualWebArena, and
020 [AlfWorld](#) benchmarks, **DEPART** with HIMPO consistently outperforms strong single-agent
021 and post-trained baselines.

022 1 INTRODUCTION

023 Large Language Models (LLMs) have shown remarkable capabilities across diverse domains. However, their
024 strengths are most evident in single-turn, non-interactive tasks, such as math problem solving (Shao et al.,
025 2024; Yu et al., 2025) and code generation (Wei et al., 2025a). Despite this progress, current LLMs struggle
026 with complex, dynamic scenarios that demand multi-step decision-making across multi-turn interactions with
027 diverse solution spaces (Yao et al., 2023). Conventional approaches to long-horizon tasks have primarily
028 relied on single-agent architectures that directly map user queries to low-level actions (Gur et al., 2024; Qi
029 et al., 2025; Wei et al., 2025b; Yang et al., 2024), overlooking the benefit of multi-step planning. While recent
030 methods have introduced intermediate planning steps to improve performance (Rawat et al., 2025), assigning
031 both strategic planning and precise execution to a single model introduces a bottleneck (Yang et al., 2025b).

032 To motivate the separation of planning and execution, we conduct a preliminary study using Claude 3.7 (Anthropic, 2025) in a single-agent setting on the shopping category of WebArena-Lite benchmark (Zhou et al.,
033 2023a; Liu et al., 2025a). Using human-annotated oracle action counts as a proxy for task complexity
034 (detailed in Appendix C.2), we observe that performance degrades as the number of required interaction steps
035 increases in Figure 1. This trend suggests that long-horizon task failures may stem from the difficulty of
036 jointly managing high-level planning and fine-grained execution in a single model (Erdogan et al., 2025).

037 In addition to separating planning and execution, we enforce a modular separation of vision and action, which
038 enhances both efficiency and robustness by empowering the planner to dynamically determine when visual
039 understanding is required. This design is motivated by two core observations. First, vision-language models
040 are computationally expensive, and many steps in long-horizon tasks do not require visual input. Unnecessary

047 inclusion of visual context increases inference costs and may degrade performance due to cross-modal
 048 distraction (Shen et al., 2025), where irrelevant modality signals interfere with reasoning, as supported by our
 049 experiments in Appendix in Table 6. **This challenge is not addressed by the text-only data generation pipelines**
 050 **in mulit-agent systems in prior works (Erdogan et al., 2025; Chen et al., 2025a)**. Second, training a unified
 051 multi-modal agent to handle both vision and action introduces fragility. Alignment fine-tuning can impair
 052 generalization (Zhai et al., 2024; Springer et al., 2025), and such agents require substantially more training
 053 data (Huang et al., 2025; Wan et al., 2025a). Moreover, reasoning gains achieved through reinforcement
 054 learning (RL) in LLMs do not seem to effectively transfer to large vision-language models (Wu et al., 2025c).

055
 056 Concretely, we propose **DEPART**, a hierarchical multi-agent framework for long-horizon, multi-turn inter-
 057 action, illustrated in Figure 2 and detailed in Section 3. DEPART decomposes complex task into modular
 058 sub-problems with three specialized agents: a planner that generates sequential high-level plan steps; an
 059 action executor that performs grounded actions in the environment; and a vision executor that interprets visual
 060 context and shares relevant information. These agents operate within a structured communication loop that
 061 supports replanning and retry mechanisms (Bensal et al., 2025; Erdogan et al., 2025), enabling dynamic
 062 adaptation and error recovery. **In contrast to prior works (Li et al., 2023; Hong et al., 2024; Wu et al., 2025d),**
 063 **our planner assigns only one plan step at a time, allowing the system to focus on the current interaction and**
 064 **dynamically adapt to environmental feedback before proceeding.** **DEPART** stands for **D**ivide, **E**valuate,
 065 **P**lan, **A**ct, **R**eflect, and **T**rack, capturing the core stages of intelligent behavior, ensuring progress evaluation,
 066 reflection, and long-term coherence.

067
 068 While DEPART’s modular structure improves system-level
 069 coordination, LLMs pre-trained on general corpora often underperform in interactive, sequential environments. To address
 070 this, we introduce a post-training optimization framework, Hi-
 071 erarchical Interactive Multi-turn Policy Optimization (HIMPO),
 072 inspired by multi-agent post-training (Park et al., 2025; Wan
 073 et al., 2025b; Leong and Wu, 2025). HIMPO alternately trains
 074 the planner and executor across two rounds using the same op-
 075 timization framework, critically differing in the reward design.
 076 The first round applies dense, role-specific feedback to foster
 077 strategy exploration (Yang et al., 2025b), support the decom-
 078 position of high-level goals into executable actions (Erdogan et al.,
 079 2025), and compensate for the lack of decision-relevant data
 080 in the pre-training of open-source LLMs (Yao et al., 2023; Wei
 081 et al., 2025b); the second round uses sparse, task-level rewards
 082 to align behavior with overall task success and mitigate reward
 083 hacking (Kumar et al., 2025). This curriculum encourages both
 084 specialization and coordinated long-horizon reasoning. Details
 085 are provided in Section 4.

086 We evaluate DEPART on WebArena-Lite (Zhou et al., 2023a;
 087 Liu et al., 2025a) and VisualWebArena (Koh et al., 2024a), two
 088 challenging web-based benchmarks, **as well as AlfWorld in**
 089 **embodied environment** for long-horizon interaction. Our ex-
 090 periments show that DEPART consistently outperforms strong
 091 single-agent baselines across various LLM backbones. Building on the DEPART architecture, we further
 092 boost performance using a smaller open-source model (Qwen3-4B) through two-round HIMPO., which
 093 enhances planner and executor via role-specific and task-level optimization.

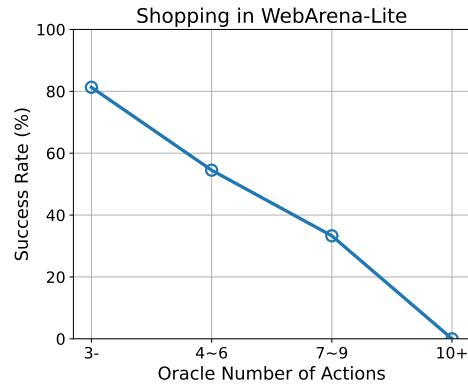


Figure 1: Success rate of single-agent (Claude 3.7) on shopping tasks in WebArena-Lite. X-axis represents the number of actions (i.e., interaction steps) required to complete each task. 3- denotes tasks solvable in 3 steps or fewer, while 10+ denotes tasks requiring more than 10 steps.

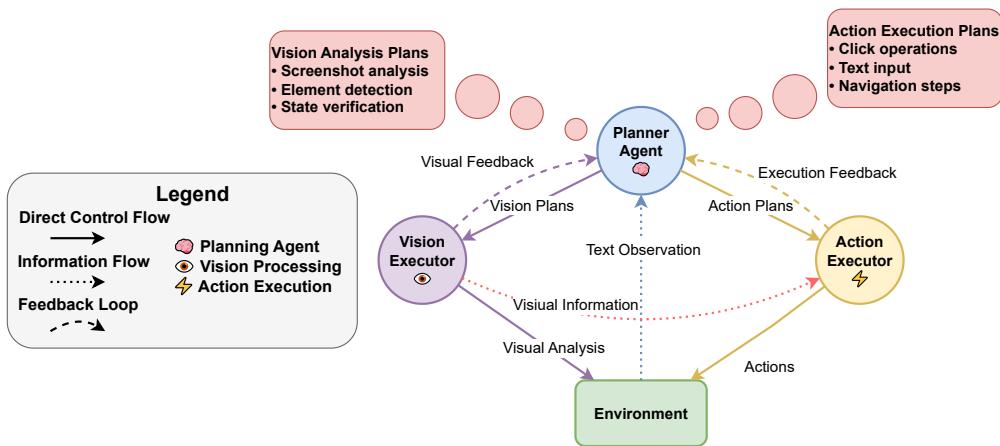


Figure 2: **Overview of DEPART**: The framework (1) divides tasks into distinct modalities and stages to reduce complexity; (2) evaluates the environment and task progress via observations and feedback; (3) plans high-level strategies based on evolving context; (4) acts through specialized executors for grounded interaction and vision understanding; (5) reflects on action outcomes to guide replanning; and (6) tracks global task status and historical context to maintain coherence across multiple turns.

2 RELATED WORK

Our work lies at the intersection of web-based agent systems and RL post-training for LLMs. We organize related work into two areas: (1) benchmarks and agent architectures for interactive web-based decision-making, and (2) RL methods for enhancing LLMs in long-horizon, multi-turn settings. Due to space constraints, we summarize key developments here and include full discussions in Appendix B. Recent advances in web agents have been driven by increasingly realistic and interactive benchmarks, progressing from synthetic environments like WoB (Shi et al., 2017) and MiniWoB++ (Liu et al., 2018) to high-fidelity platforms such as WebArena (Zhou et al., 2023a). These benchmarks have enabled the emergence of various agent architectures, including domain-specific models with lightweight policy heads (Furuta et al., 2024; Deng et al., 2024), prompt-based LLM agents leveraging modular tool-use strategies (Song et al., 2024a; Koh et al., 2024b), and RL-enhanced agents with post-training (Wei et al., 2025b; Qi et al., 2025). RL has shown promise in aligning LLMs with downstream objectives in single-turn scenarios (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b; Ouyang et al., 2022; Zhao et al., 2025b). Recent methods extend to multi-turn interactive tasks (Zhou et al., 2025; 2024b; Wei et al., 2025b; Qi et al., 2025) while most prior RL-based approaches rely on sparse, final-state rewards or make simplifying assumptions about the environment (Wei et al., 2025b; Qi et al., 2025), limiting their adaptability in complex, long-horizon tasks. In contrast, our method introduces fine-grained, agent-specific reward functions tailored to the planner and executor roles within a collaborative framework. We further propose a unified turn-level policy gradient objective that improves training stability while enabling expressive, role-specific behavior in multi-agent, sequential decision-making.

3 DEPART: HIERARCHICAL MULTI-AGENT SYSTEM

3.1 PROBLEM FORMULATION

A Markov Decision Process (MDP) is a standard framework for modeling sequential decision-making under full observability. A Partially Observable Markov Decision Process (POMDP) extends the MDP to settings

where the system follows MDP dynamics, but the agent only has partial access to the underlying state. We formulate a web browsing task as a POMDP to capture the inherent uncertainty and limited observability present in complex and dynamic web environments (Wei et al., 2025b). A POMDP is defined by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. The state space \mathcal{S} encompasses the entire internet content, browser context, and user-related metadata. \mathcal{O} denotes the observation space, which may consist of structured text \mathcal{O}^{text} and image elements \mathcal{O}^{img} . As the full state \mathcal{S} is prohibitively large and often not fully observable in practice, decision-making is based only on partial observations $\mathcal{O} = (\mathcal{O}^{text}, \mathcal{O}^{img})$. The action space \mathcal{A} includes low-level browser interaction primitives such as clicking, typing, and scrolling. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the transition probability ($\Delta(\mathcal{S})$ is the distribution over \mathcal{S}), capturing the environment dynamics given the current state and action. We denote $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as the reward function. Specifically, the agent receives a binary task reward $r_T \in \{0, 1\}$ at the final step T of each task, indicating if task completion was successful (e.g., reaching a target page or submitting a correct form). Episodes terminate upon answer submission or after a fixed step limit. To facilitate learning, we further introduce dense, role-specific rewards in early training (e.g., execution alignment and plan confidence), along with penalties for format violations and step limits. In a second training stage, we remove these intermediate signals and optimize solely for task-level success. Full reward details are provided in Appendix D.4.

3.2 FROM SINGLE AGENT TO MULTI-AGENT

We begin by considering a single agent operating within the previously defined POMDP. The agent’s policy π_θ is implemented on the top of an autoregressive language model, which serves as the decision-making backbone. To improve performance given a user query \mathbf{x} , the agent generates an explicit high-level multi-step plan \mathbf{m} , followed by a response \mathbf{y} that encodes executable actions (Rawat et al., 2025). The plan \mathbf{m} consists of reasoning and the corresponding plan.

During interaction with the web environment, the executed action at time step t , denoted as a_t , is abstracted from the response \mathbf{y}_t using a deterministic mapping function f , i.e., $a_t = f(\mathbf{y}_t)$. For instance, if the response includes the span `<act>do('Scroll Down')</act>`, the resulting action is $a_t = \text{do('Scroll Down')}$. The generation process for a single interaction step can be modeled probabilistically as:

$$\pi_\theta(\mathbf{y}, \mathbf{m}|\mathbf{x}, o) = \pi_\theta(\mathbf{y}|\mathbf{m}, \mathbf{x}, o) \cdot \pi_\theta(\mathbf{m}|\mathbf{x}, o) \quad (1)$$

where $\pi_\theta(\mathbf{y}, \mathbf{m}|\mathbf{x}, o)$ denotes the probability of producing a plan \mathbf{m} and a response \mathbf{y} given input query \mathbf{x} and $o \in \mathcal{O}$. Assuming that \mathbf{y} is conditionally independent of \mathbf{x} given \mathbf{m} , the factorization simplifies to:

$$\pi_\theta(\mathbf{y}, \mathbf{m}|\mathbf{x}, o) = \pi_\theta(\mathbf{y}|\mathbf{m}, o) \cdot \pi_\theta(\mathbf{m}|\mathbf{x}, o) \quad (2)$$

This assumption reflects the intuition that once a coherent plan \mathbf{m} is established, the query \mathbf{x} provides no further information required for generating the response \mathbf{y} . However, this monolithic formulation demands that a single language model manages both planning and low-level execution simultaneously, limiting scalability and modularity.

To address these challenges, we build upon prior work (Erdogan et al., 2025; Wan et al., 2025b; Chen et al., 2025a) and propose a hierarchical multi-agent framework with three specialized agents with their individual policy: a planner π_p , a vision executor π_v , and an action executor π_a . Inspired by hierarchical planning in robotics (Zhang et al., 2025; Hsu et al., 2024a) and structured decision-making in MDPs (Pignatelli et al., 2024), the planner decomposes the user query \mathbf{x} into a sequence of semantically meaningful plan steps (i.e., sub-goals), such as ‘log in’, ‘navigate to product page’, or ‘apply filters’, which serve as intermediate waypoints toward task completion, and then orchestrates the execution process. Each plan step functions as a localized objective, enhancing interpretability and reliability, while enabling executors to operate over shorter, context-specific horizons. By focusing on the current sub-goal rather than the entire task, each executor improves sample efficiency and behavioral robustness.

188 The planner produces two high-level plans (\mathbf{m}^v , \mathbf{m}^a): \mathbf{m}^v for the vision executor and \mathbf{m}^a for the action
 189 executor. The vision executor handles perception by processing visual content from the web environment,
 190 while the action executor performs fine-grained browser actions. Formally, the decision process at a single
 191 time step is modeled as:
 192

$$194 \quad \mathbf{y} \sim \pi_a(\mathbf{y}|\mathbf{v}, \mathbf{m}^a, o^{text}) \cdot \pi_v(\mathbf{v}|\mathbf{m}^v, o^{img}) \cdot \pi_p((\mathbf{m}^a, \mathbf{m}^v)|\mathbf{x}, o^{text}) \quad (3)$$

196 where the planner $\pi_p(\mathbf{m}^a, \mathbf{m}^v|\mathbf{x}, o^{text})$ generates two distinct plans $\mathbf{m} = (\mathbf{m}^a, \mathbf{m}^v)$ for action and vision
 197 executors given query \mathbf{x} and o^{text} . The vision executor then produces vision-derived information \mathbf{v} , and the
 198 action executor generates the response \mathbf{y} .

199 3.3 MULTI-TURN INTERACTIONS

200 To handle long-horizon interactions, we extend this model over T time steps:

$$202 \quad \mathbf{y}_T \sim \prod_{t=1}^T \pi_a(\mathbf{y}_t|\mathbf{v}_t, \mathbf{m}_t^a, o_t^{text}) \cdot \pi_v(\mathbf{v}_t|\mathbf{m}_t^v, o_t^{img}) \cdot \pi_p((\mathbf{m}_t^a, \mathbf{m}_t^v)|\mathbf{x}, o_t^{text}, \{\mathbf{v}, \mathbf{m}^a, \mathbf{m}^v, \mathbf{y}\}_{<t}) \quad (4)$$

206 Execution unfolds in a sequential loop. Given a user query and the current text-based observation $o_t^{text} \in$
 207 \mathcal{O}^{text} , the planner generates two parallel but distinct high-level plans: one for the vision executor and one
 208 for the action executor. These plans consist of ordered directives tailored to the perception and interaction
 209 subroutines needed to complete the task. Importantly, the planner may omit visual planning steps entirely if it
 210 determines that visual input is unnecessary, thereby avoiding the overhead of vision processing when it does
 211 not contribute meaningfully to task completion.

212 During execution, the planner dispatches individual plan steps to the corresponding executor and tracks their
 213 progress internally. When visual perception is required, the vision executor processes the web page image
 214 $o_t^{img} \in \mathcal{O}^{img}$ and extracts relevant information, which is communicated to both the planner and the action
 215 executor. The action executor then integrates this visual feedback (when available), textual observations
 216 $o_t^{text} \in \mathcal{O}^{text}$, and planner directives to perform fine-grained interactions within the web environment.

217 The planner continuously monitors execution feedback and updated textual content to decide whether to: (i)
 218 proceed to the next plan step, (ii) retry the current step, or (iii) generate a new plan [instead of purely update
 219 the plan at each step \(Erdogan et al., 2025; Chen et al., 2025a\)](#). Throughout the process, the planner maintains
 220 an internal representation of executor progress and task state, enabling adaptive coordination and real-time
 221 plan revision. An overview of the full framework is illustrated in Figure 2.

222 4 HIERARCHICAL INTERACTIVE MULTI-TURN POLICY OPTIMIZATION

224 To reduce data requirements, computational overhead, and instability associated with multi-modal post-
 225 training (Huang et al., 2025; Wan et al., 2025a; Zhai et al., 2024; Springer et al., 2025), we focus training
 226 on the high-level planner and low-level action executors. This leverages the observation that static image
 227 interpretation by the vision executor is generally less dependent on long-horizon context and can operate
 228 effectively without additional post-training.

229 We formulate this multi-agent system as a two-player Markov game (Shapley, 1953) between the planner
 230 and the action executor, treating the vision executor as part of the environment and excluding it from the
 231 training loop. Extending the standard POMDP framework, we define a Partially Observable Markov Game
 232 (POMG) by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{P}, \mathcal{R}^p, \mathcal{R}^a)$, where $\mathcal{S}, \mathcal{O}, \mathcal{A}$ retain their usual definitions in a POMDP.
 233 Specifically, \mathcal{A} is the action space of the action executor and contains low-level browser interaction primitives.
 234 \mathcal{M} is the space of plan messages from the planner, where $\mathbf{m}_t \in \mathcal{M}$ denotes the plan assigned to the action

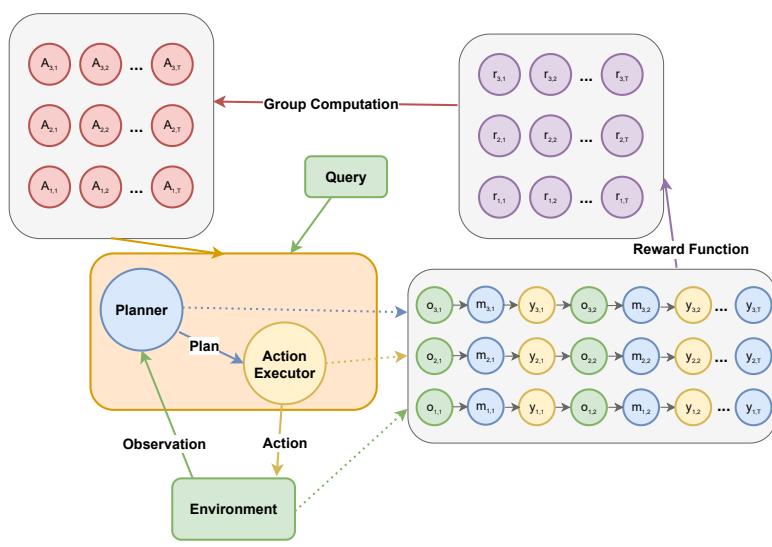


Figure 3: Hierarchical Interactive Multi-turn Policy Optimization (HIMPO). Parallel group rollouts generate trajectories for planner and action executor, with collective advantage computation for policy optimization. The vision executor is excluded from training.

executor at time step t ¹ $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{M} \rightarrow \Delta(\mathcal{S})$ is the transition probability. $\mathcal{R}^p, \mathcal{R}^a : \mathcal{S} \times \mathcal{A} \times \mathcal{M} \rightarrow \mathbb{R}$ are the reward functions for the planner and action executor, respectively, where the step-wise reward for the planner is built upon intrinsic reward (Kang et al., 2025; Zhao et al., 2025a) while the executor receives reward if the executor’s actions align with the planner’s expectations. This setup yields a general-sum game rather than a purely cooperative or zero-sum structure.

We adopt a parameter-sharing strategy in which the planner and the action executor share the same model weights θ but are distinguished by role-specific system prompts S_p and S_a (Wan et al., 2025b). This design avoids costly GPU model swaps during rollouts, supports larger batch sizes, and enables efficient joint optimization within a unified training pipeline. Building on Group Relative Policy Optimization (GRPO), we introduce Hierarchical Interactive Multi-turn Policy Optimization (HIMPO), a novel training framework that alternatively optimizes the planner and the action executor. In HIMPO, each agent has a distinct learning objective corresponding to its role $\in \{\text{plan, action}\}$:

$$\mathcal{J}_{\text{HIMPO}}^{\text{role}}(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{e}) \sim \mathcal{D}, \{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\mathcal{P} | (\mathbf{x}, \mathbf{e}))} \left[\frac{1}{\sum_{i=1}^G |\tau_i^{\text{role}}|} \sum_{i=1}^G \sum_{t=1}^{|\tau_i^{\text{role}}|} \left(\min(l_{i,t}^{\text{role}}(\theta) A_{i,t}, \text{clip}(l_{i,t}^{\text{role}}(\theta), 1 - \epsilon, 1 + \epsilon) A_{i,t}) - \beta D_{\text{KL}}(\pi_{\theta} || \pi_{\text{ref}}) \right) \right] \quad (5)$$

¹We omit the superscript of \mathbf{m}_i^a because we exclude vision executor during training.

282

283

$$l_{i,t}^{plan}(\theta) = \exp \left(\frac{1}{|\mathbf{m}_{i,t}|} \sum_{j=1}^{|\mathbf{m}_{i,t}|} \log \frac{\pi_{\theta}(\mathbf{m}_{i,t,j} | \mathbf{x}, \{\mathbf{m}_{i,\cdot}, \mathbf{y}_{i,\cdot}\}_{<t}, \mathbf{m}_{i,t,<j}, o_t^{text})}{\pi_{\theta_{old}}(\mathbf{m}_{i,t,j} | \mathbf{x}, \{\mathbf{m}_{i,\cdot}, \mathbf{y}_{i,\cdot}\}_{<t}, \mathbf{m}_{i,t,<j}, o_t^{text})} \right) \quad (6)$$

287

288

$$l_{i,t}^{action}(\theta) = \exp \left(\frac{1}{|\mathbf{y}_{i,t}|} \sum_{j=1}^{|\mathbf{y}_{i,t}|} \log \frac{\pi_{\theta}(\mathbf{y}_{i,t,j} | \mathbf{m}_{i,t}, \mathbf{y}_{i,t,<j}, o_t^{text})}{\pi_{\theta_{old}}(\mathbf{y}_{i,t,j} | \mathbf{m}_{i,t}, \mathbf{y}_{i,t,<j}, o_t^{text})} \right) \quad (7)$$

291

292

293

$$\hat{A}_{i,t} = \frac{r_{i,t} - \text{mean}(\{r_{i,t}\}_{i=1}^G)}{\text{std}(\{r_{i,t}\}_{i=1}^G)} \quad (8)$$

Our unified objective in equation 5 can be viewed as a multi-turn variant of dynamic sampling policy optimization (DAPo) (Yu et al., 2025) with turn-level policy gradient loss for long reasoning in 2-agent setting, illustrated in Figure 3. The key distinction between two agents are their trajectories τ_i^{role} and turn-level ratio $l_{i,t}^{role}$. For each task query \mathbf{x} , we sample a group of trajectories $\{\tau_1, \tau_2, \dots, \tau_G\}$, where τ_i^{plan} indicates a sequence of plan steps from the planner and τ_i^{action} denotes the sequence of low-level browser interaction primitives. The turn-level ratio for both agents are listed in equation 6 and equation 7. We extend the advantage computation in a group-relative manner to consider multi-turn steps t as in equation 8.

5 EXPERIMENTS

We evaluate our approach in two stages: (1) measuring the impact of the DEPART architecture on long-horizon web tasks, and (2) analyzing the effectiveness of the HIMPO post-training algorithm via ablation studies. Experiments are conducted on complex web-browsing tasks (Appendix C.1) where agents use 12 high-level actions via Playwright (Koh et al., 2024a) and receive observations from the accessibility tree, a semantically rich DOM representation. Visual elements are linked to image assets when grounding is required. We report results on two web-agent benchmarks: WebArena-Lite (Zhou et al., 2023a; Liu et al., 2025a) (165 text-dominant tasks) and VisualWebArena (Koh et al., 2024a; Zhang et al., 2024) (114 vision-intensive tasks) with full experiment details are in Appendix E.1. **We additionally evaluate our method on the AlfWorld benchmark** (Shridhar et al., 2021), a text-based household task simulator that requires multi-step reasoning to follow high-level instructions and provides sparse, binary success signals. Following standard protocols, we use the **seen** split for in-distribution evaluation and the **unseen** split to assess out-of-distribution generalization.

5.1 DEPART: IMPACT OF MODULAR AGENT DESIGN

We compare three configurations of our architecture: (i) a single-agent model handling both planning and execution; (ii) a 2-agent setup separating planner and executor; and (iii) the full 3-agent DEPART system, which includes a vision executor dynamically invoked by the planner. Baselines include general-purpose LLMs, reasoning-specialized models, and RL-fine-tuned agents. **In web browsing tasks**, to ensure fair comparison with prior work (Wei et al., 2025b; Qi et al., 2025), we first evaluate models without visual input on WebArena-Lite. As shown in Table 1, prompting strong foundation models (e.g., OpenAI-o3, OpenAI-o4-mini, Claude 3.7) yields over 35% success. RL post-trained agents like WebRL and WebAgent-R1 further improve to over 40% (Table 2). Notably, introducing modularity leads to consistent gains. For example, Claude 3.7 in the 2-agent configuration reaches 46.1% average success in Table 1, surpassing both WebRL and WebAgent-R1 in Table 2. These results highlight the benefit of structured collaboration between specialized agents. **In AlfWorld in Table 3, we empirically confirm that modularity increases success rate in both Qwen3-4B and Claude 3.7.**

329 Table 1: Task success rate in WebArena-Lite (Zhou et al., 2023a) for **prompting-only methods without**
 330 **visual input**. The highest value in each column is in bold. Avg SR denotes the average success rate.
 331

Category	Model	Reddit	GitLab	CMS	Map	Shopping	Avg SR
1 Agent	Qwen2.5-3B	5.3	13.3	5.7	0.0	4.4	6.1
	Llama3.1-8B	5.3	10.0	5.7	15.4	8.9	8.5
	Qwen2.5-32B	10.5	20.0	20.0	19.2	17.8	16.9
	GPT-4o	10.5	10.0	20.0	20.0	11.1	13.9
	GPT-4o-Turbo	10.5	16.7	14.3	36.7	13.3	17.6
	QwQ-32B	15.8	33.3	25.7	15.4	20.0	22.4
	OpenAI-o3	36.8	46.7	45.7	38.5	33.3	39.4
	OpenAI-o4-mini	47.4	43.3	45.7	26.9	28.9	36.9
2 Agents	Qwen3-4B	5.3	0.0	11.4	8.3	26.7	12.1
	(ours)	Claude 3.7	42.1	16.7	40.0	16.7	35.8
2 Agents	Qwen3-4B	5.3	3.3	14.3	8.3	31.1	14.5
	(ours)	Claude 3.7	47.4	56.7	42.9	16.7	60.0

345 Table 2: Task success rate in WebArena-Lite (Zhou et al., 2023a) for **RL post-training methods without**
 346 **visual input**. The highest value in each column is in bold. Avg SR denotes the average success rate.
 347

Category	Model	Reddit	GitLab	CMS	Map	Shopping	Avg SR
1 Agent	Llama3.1-8B (WebRL) (Qi et al., 2025)	63.2	46.7	54.3	36.7	31.1	42.4
	Llama3.1-8B (WebAgent-R1) (Wei et al., 2025b)	47.4	56.7	57.1	23.1	44.4	44.8
	Qwen2.5-3B (WebAgent-R1) (Wei et al., 2025b)	26.3	53.3	48.6	26.9	24.4	33.9
	Qwen3-4B (MT-GRPO)	26.3	43.3	54.3	16.7	42.2	37.6
2 Agents	Qwen3-4B (HIMPO) (ours)	52.6	56.7	62.9	27.8	57.8	51.5

355 5.2 HIMPO: IMPROVING OF POST-TRAINING AND ABLATION STUDY

356 Beyond architectural improvements, we evaluate our proposed post-training algorithm, HIMPO. Following
 357 prior work (Wei et al., 2025b; Qi et al., 2025), we train on 647 WebArena tasks not included in WebArena-Lite,
 358 reserving the 165 human-verified WebArena-Lite tasks for evaluation. Table 2 reports results from prior
 359 methods and our own experiments. WebRL and WebAgent-R1 are state-of-the-art RL post-training methods
 360 on this benchmark, motivating us to explore whether our HIMPO can improve the performance of smaller
 361 models compared with those proprietary LLMs. Specifically, we use Qwen3-4B and include a single-agent
 362 baseline trained with multi-turn GRPO (denoted MT-GRPO), a direct extension of GRPO to multi-turn
 363 settings (Wei et al., 2025b). We show that applying HIMPO in a 2-agent configuration with separate planner
 364 and executor with two-round HIMPO post-training (bottom row in Table 2) consistently outperform all
 365 single-agent post-training, especially that it improves the success rate against MT-GRPO with the same model
 366 size (Qwen3-4B) in every category by a wide margin. [We follow the same training pipeline for AlfWorld,](#)
 367 [showing that post-training Qwen3-4B with HIMPO outperforms single-agent post-training methods.](#)

368 5.3 ABLATION STUDIES

369 **RL post-training** To assess the contribution of individual components in HIMPO, we conduct ablations
 370 targeting four key design choices: (i) dense role-specific rewards, (ii) turn-level ratio estimation, (iii) dynamic
 371 sampling, and (iv) modular training. Methodological details are provided in Appendix E.2. We evaluate
 372 eight training variants, with M1–M4 trained in a single-agent setting, and report their success rates on
 373 WebArena-Lite across epochs (Figure 4).² Notably, all methods outperform the pre-trained base model (12.1%
 374

375 ²WebArena-Lite tasks are not used for training.

376 Table 3: Task success rate in AlfWorld (Shridhar et al., 2021) for **prompting-only methods**.
377

Category	Model	Seen	Unseen
1 Agent	OpenAI-o3-mini	58.9	62.7
	Deepseek-R1	61.4	53.7
	Qwen2.5-7B-Instruct	72.1	76.1
	Llama-3.1-8B-Instruct	32.9	40.3
	GPT-4	64.8	64.4
	Qwen3-4B	36.4	37.3
	Claude 3.7	78.6	82.1
2 Agents (ours)	Qwen3-4B	42.1	41.0
	Claude 3.7	82.1	85.1

388 Table 4: Task success rate in AlfWorld (Shridhar et al., 2021) for **post-training methods**.
389

Category	Model	Seen	Unseen
1 Agent	Llama3.1-8B-Instruct (ETO) (Song et al., 2024b)	78.6	71.6
	Llama3.1-8B-Instruct (MPO) (Xiong et al., 2025)	82.9	78.4
	Llama3.1-8B-Instruct (BPO) (Wang et al., 2025a)	87.9	89.6
	Qwen2.5-7B-Instruct (ETO) (Song et al., 2024b)	82.1	76.1
	Qwen2.5-7B-Instruct (MPO) (Xiong et al., 2025)	81.4	88.1
	Qwen2.5-7B-Instruct (BPO) (Wang et al., 2025a)	90.0	91.0
	Qwen3-4B (MT-GRPO)	82.1	88.1
2 Agents (ours)	Qwen3-4B (HIMPO) (ours)	92.9	91.8

401 in Table 1) without supervised fine-tuning (SFT), indicating that tool-augmented agents can benefit from
402 RL alone. Building on M1, M2 introduces dense, role-specific rewards (details in Appendix D.4) and M3
403 apply turn-level ratio clipping (Wan et al., 2025b), by treating an entire agent turn as a single action. Both
404 contribute to further performance gains by better aligning reward scaling with the underlying MDP structure
405 compared with token-level ratios (Shao et al., 2024; Yu et al., 2025).

406 To isolate the effect of modular training, M5 trains only the planner, and M6 trains both planner and executor.
407 The progression M3 → M5 → M6 shows consistent gains, confirming that combining planning and execution
408 in a single model can create a performance bottleneck (Yang et al., 2025b). Although M6 (single-round
409 HIMPO) already outperforms all single-agent baselines, early training is slower, likely due to multi-agent
410 optimization complexity and susceptibility to reward hacking from output format violations (Figure 4). To
411 better understand this issue, a preliminary study with dynamic sampling (Yu et al., 2025) (M4) over-samples
412 and filters trajectories with incorrect output formats, accelerating early-stage convergence but plateauing
413 sooner, highlighting a trade-off between early stability and long-term generalization. This insight motivates
414 our two-round HIMPO strategy: M7 uses the first 6 epochs as a warm start with role-specific rewards, format
415 penalties, and dynamic sampling, and M8 continues training using only task-level rewards. M8 consistently
416 improves over M6, demonstrating the benefit of curriculum-style refinement (Qi et al., 2025) in reward design.

417 **Cross-modal Distraction** We also study the effect of adding visual input. In WebArena-Lite, injecting
418 vision often introduces irrelevant context or reasoning overhead, confirming the cross-modal distraction
419 phenomenon (Shen et al., 2025) (green vs. red cells in Appendix in Table 6; detailed cases in Appendix G).
420 Conversely, on VisualWebArena, all models benefit from visual input when tasks require perception (Table 7 in
421 Appendix E.3). Motivated by this contrast, our final configuration adopts the full 3-agent DEPART framework,
422 in which the planner selectively invokes visual processing only when needed. Because Qwen3-4B lacks

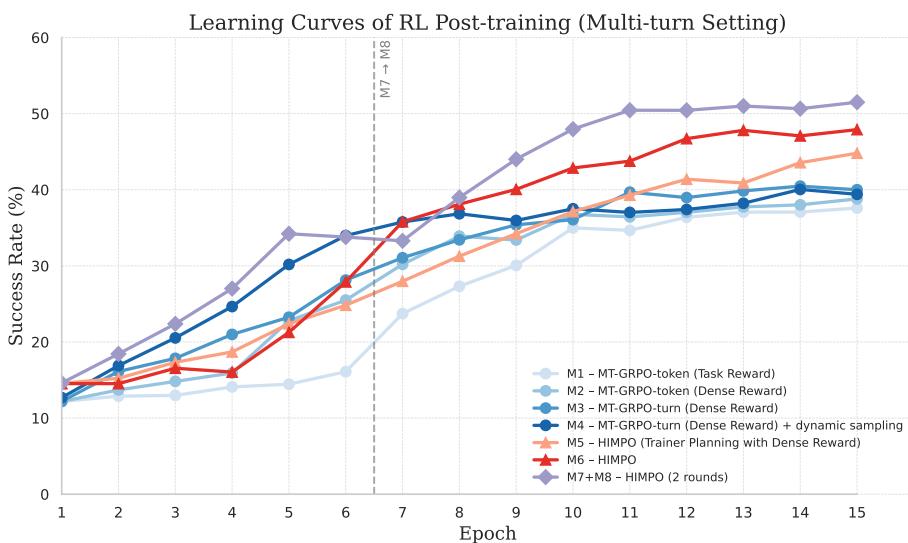


Figure 4: Evaluation on WebArena-Lite across training epochs. Methods M1–M4 correspond to single-agent training approaches, whereas M5–M8 are multi-agent training approaches. The highest success rate is achieved by initializing with M7 (warm start) and continuing training with M8; the dashed line marks the point of this transition.

native vision capability, we employ Claude 3.7 as its vision executor within the three-agent setup, combined with two-round HIMPO training (M7 + M8). This approach yields a 52.1% success rate on WebArena-Lite, surpassing all baselines and demonstrating that selectively grounding vision improves performance. Moreover, applying the three-agent configuration with HIMPO on Qwen3-4B raises its success rate on VisualWebArena tasks from 9.6% to 36.0% (Table 7), further highlighting the benefit of targeted visual grounding.

6 CONCLUSION

This work addresses key limitations of LLM-based agents in solving complex, long-horizon tasks. We propose DEPART, a modular multi-agent framework that separates planning, vision, and execution into specialized components, enabling structured coordination through dynamic communication, replanning, and retries. To complement this architecture, we introduce HIMPO, a two-stage post-training algorithm that improves learning efficiency through role-specific and task-level optimization. Together, DEPART and HIMPO significantly improve performance on realistic web-based and embodied benchmarks. We view structured multi-agent systems as a necessary foundation for advancing post-training and long-horizon decision-making in LLM-based agents.

7 ETHICS STATEMENT

This research adheres to the ICLR Code of Ethics. All datasets used are publicly available and open-source, with licenses permitting research use. Human annotation was performed solely to estimate oracle action counts for evaluation purposes, based on known task outcomes in existing benchmarks. No private or personally identifiable information was involved, and no new data from human subjects was collected. The study does not raise privacy, security, or fairness concerns, and we disclose no conflicts of interest or external sponsorships.

470 8 REPRODUCIBILITY STATEMENT
471472 The main training equations are presented in Section 4, and the algorithmic designs are detailed in Appendix
473 D. All datasets used in our experiments are publicly available and described in Section 5 and Appendix E.1.
474 Implementation details, computational resources, and system prompts are also provided in Appendix F. Taken
475 together, these resources enable independent researchers to verify and reproduce our findings.476 REFERENCES
477478 T. Abuelsaad, D. Akkil, P. Dey, A. Jagmohan, A. Vempaty, and R. Kokku. Agent-e: From autonomous web
479 navigation to foundational design principles in agentic systems. *arXiv preprint arXiv:2407.13032*, 2024.
480 Anthropic. Claude 3.7 sonnet and claude code. [https://www.anthropic.com/news/](https://www.anthropic.com/news/claude-3-7-sonnet)
481 claude-3-7-sonnet, 2025.
482 S. Bensal, U. Jamil, C. Bryant, M. Russak, K. Kamble, D. Mozolevskyi, M. Ali, and W. AlShikh. Reflect,
483 retry, reward: Self-improving llms via reinforcement learning. *arXiv preprint arXiv:2505.24726*, 2025.
484 S. Casper, X. Davies, C. Shi, et al. Open problems and fundamental limitations of reinforcement learning
485 from human feedback. *Transactions on Machine Learning Research*, 2024.
486 J. Chen, H. Li, J. Yang, Y. Liu, and Q. Ai. Enhancing llm-based agents via global planning and hierarchical
487 execution. *arXiv preprint arXiv:2504.16563*, 2025a.
488 M. Chen, T. Li, H. Sun, et al. Research: Learning to reason with search for llms via reinforcement learning.
489 *arXiv preprint arXiv:2503.19470*, 2025b.
490 P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from
491 human preferences. In *Advances in Neural Information Processing Systems*, Proceedings of Machine
492 Learning Research. PMLR, 2017.
493 T. Chu, Y. Zhai, J. Yang, S. Tong, S. Xie, D. Schuurmans, Q. V. Le, S. Levine, and Y. Ma. Sft memorizes,
494 rl generalizes: A comparative study of foundation model post-training. In *Proceedings of the 34th*
495 *International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2025.
496 P. Das, B. Nortmann, L. J. Ratliff, V. Gupta, and T. Mylvaganam. Learning in stochastic stackelberg games.
497 In *2024 American Control Conference (ACC)*, pages 3557–3562, 2024.
498 X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, and Y. S. Huan Sun. Mind2web: Towards a
499 generalist agent for the web. In *Advances in Neural Information Processing Systems*, Proceedings of
500 Machine Learning Research. PMLR, 2023.
501 Y. Deng, X. Zhang, W. Zhang, Y. Yuan, S.-K. Ng, and T.-S. Chua. On the multi-turn instruction following for
502 conversational web agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational*
503 *Linguistics*, pages 8795–8812. Association for Computational Linguistics, 2024.
504 Y. Erdogan, N. Lee, S. Kim, S. Moon, H. Furuta, G. Anumanchipalli, K. Keutzer, and A. Gholami. Plan-
505 and-act: Improving planning of agents for long-horizon tasks. In *International Conference on Machine*
506 *Learning*, 2025.
507 Z. Fei, L. Ji, S. Wang, J. Shi, J. Gong, and X. Qiu. Unleashing embodied task planning ability in llms via
508 reinforcement learning. *arXiv preprint arXiv:2506.23127*, 2025.
509 J. Feng, S. Huang, X. Qu, G. Zhang, Y. Qin, B. Zhong, C. Jiang, J. Chi, and W. Zhong. Retool: Reinforcement
510 learning for strategic tool use in llms, 2025a.
511

517 L. Feng, Z. Xue, T. Liu, and B. An. Group-in-group policy optimization for llm agent training. In *Advances*
 518 *in Neural Information Processing Systems*, Proceedings of Machine Learning Research. PMLR, 2025b.
 519

520 X. Feng, B. Han, Z. Zhou, J. Fan, J. Yao, K. H. Li, D. Yu, and M. Ng. DyPO: Dynamic policy optimization
 521 for multi-turn interactive reasoning. In *ICML 2025 Workshop on Programmatic Representations for Agent*
 522 *Learning*, 2025c.

523 H. Furuta, K.-H. Lee, O. Nachum, Y. Matsuo, A. Faust, S. S. Gu, and I. Gur. Multimodal web navigation with
 524 instruction-finetuned foundation models. *International Conference on Learning Representation*, 2024.
 525

526 X. Geng, P. Xia, Z. Zhang, et al. Webwatcher: Breaking new frontiers of vision-language deep research agent.
 527 *arXiv preprint arXiv:2508.05748*, 2025.

528 I. Gur, H. Furuta, A. Huang, M. Safdari, Y. Matsuo, D. Eck, and A. Faust. A real-world webagent with
 529 planning, long context understanding, and program synthesis. *International Conference on Learning*
 530 *Representation*, 2024.

531 H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, and D. Yu. WebVoyager: Building an end-to-end
 532 web agent with large multimodal models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings*
 533 *of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,
 534 pages 6864–6890, Bangkok, Thailand, 2024. Association for Computational Linguistics.

535 S. Hong, M. Zhuge, J. Chen, and et al. Metagpt: Meta programming for a multi-agent collaborative framework.
 536 *International Conference on Learning Representations*, 2024.

537 H.-L. Hsu, A. K. Bozkurt, J. Dong, Q. Gao, V. Tarokh, and M. Pajic. Steering decision transformers via
 538 temporal difference learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and*
 539 *Systems (IROS)*, pages 7477–7483, 2024a.

540 H.-L. Hsu, H. Meng, S. Luo, J. Dong, V. Tarokh, and M. Pajic. Reforma: Robust reinforcement learning
 541 via adaptive adversary for drones flying under disturbances. In *2024 IEEE International Conference on*
 542 *Robotics and Automation (ICRA)*, pages 5169–5175, 2024b.

543 C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang. Thinkact: Vision-language-action
 544 reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

545 I. L. Iong, X. Liu, Y. Chen, H. Lai, S. Yao, P. Shen, H. Yu, Y. Dong, and J. Tang. Openwebagent: An open
 546 toolkit to enable web agents on large language models. In *ACL 2024 System Demonstration Track*, 2024.

547 B. Jin, H. Zeng, Z. Yue, J. Yoon, S. O. Arik, D. Wang, H. Zamani, and J. Han. Search-r1: Training LLMs
 548 to reason and leverage search engines with reinforcement learning. In *Second Conference on Language*
 549 *Modeling*, 2025.

550 Z. Kang, X. Zhao, and D. Song. Scalable best-of-n selection for large language models via self-certainty.
 551 *arXiv preprint arXiv:2502.18581*, 2025.

552 J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and
 553 D. Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *ACL*, 2024a.

554 J. Y. Koh, S. McAleer, D. Fried, and R. Salakhutdinov. Tree search for language model agents. *arXiv preprint*
 555 *arXiv:2407.01476*, 2024b.

556 K. Kumar, T. Ashraf, O. Thawakar, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, P. H. S. Torr, F. S.
 557 Khan, and S. Khan. Llm post-training: A deep dive into reasoning large language models, 2025.

564 W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient
 565 memory management for large language model serving with pagedattention. In *Proceedings of the ACM*
 566 *SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

567

568 H. Y. Leong and Y. Wu. pilotrl: Training language model agents via global planning-guided progressive
 569 reinforcement learning. *arXiv preprint arXiv:2507.23261*, 2025.

570

571 G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbulin, and B. Ghanem. Camel: Communicative agents for
 572 “mind” exploration of large language model society. In *Thirty-seventh Conference on Neural Information*
 573 *Processing Systems*, 2023.

574

575 K. Li, Z. Zhang, H. Yin, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint*
 576 *arXiv:2507.20673*, 2025.

577

578 E. Z. Liu, K. Guu, P. Pasupat, T. Shi, and P. Liang. Reinforcement learning on web interfaces using
 579 workflow-guided exploration. *International Conference on Learning Representations*, 2018.

580

581 X. Liu, T. Zhang, Y. Gu, I. L. Iong, Y. Xu, X. Song, S. Zhang, H. Lai, X. Liu, H. Zhao, et al. Visualagentbench:
 582 Towards large multimodal models as visual foundation agents. *International Conference on Learning*
 583 *Representation*, 2025a.

584

585 Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding r1-zero-like training: A
 586 critical perspective. In *Conference on Language Modeling (COLM)*, 2025b.

587

588 R. Nakano, J. Hilton, S. Balaji, et al. Webgpt: Browser-assisted question-answering with human feedback.
 589 *arXiv preprint arXiv:22112.09332*, 2022.

590

591 T. Ni, M. Ma, B. Eysenbach, and P.-L. Bacon. When do transformers shine in RL? decoupling memory from
 592 credit assignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

593

594 L. Ouyang, J. Wu, et al. Training language models to follow instructions with human feedback. In *Advances*
 595 *in Neural Information Processing Systems*, Proceedings of Machine Learning Research. PMLR, 2022.

596

597 C. Park, S. Han, X. Guo, A. E. Ozdaglar, K. Zhang, and J.-K. Kim. MAPoRL: Multi-agent post-co-training
 598 for collaborative large language models with reinforcement learning. In *Proceedings of the 63rd Annual*
 599 *Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics,
 600 2025.

601

602 E. Pignatelli, J. Ferret, M. Geist, T. Mesnard, H. van Hasselt, and L. Toni. A survey of temporal credit
 603 assignment in deep reinforcement learning. *Transactions on Machine Learning Research*, 2024. ISSN
 604 2835-8856. Survey Certification.

605

606 Z. Qi, X. Liu, I. L. Iong, H. Lai, X. Sun, W. Zhao, Y. Yang, X. Yang, J. Sun, S. Yao, T. Zhang, W. Xu, J. Tang,
 607 and Y. Dong. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning.
 608 *International Conference on Learning Representation*, 2025.

609

610 R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization:
 611 Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*,
 612 Proceedings of Machine Learning Research. PMLR, 2023.

613

614 R. Ramamurthy, P. Ammanabrolu, K. Brantley, J. Hessel, R. Sifa, C. Bauckhage, H. Hajishirzi, and Y. Choi. Is
 615 reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks
 616 for natural language policy optimization. *International Conference on Learning Representation*, 2023.

611 M. Rawat, A. Gupta, R. Goomer, A. D. Bari, N. Gupta, and R. Pieraccini. Pre-act: Multi-step planning and
 612 reasoning improves acting in llm agents. *arXiv preprint arXiv:2505.09970*, 2025.

613

614 A. Setlur, C. Nagpal, A. Fisch, X. Geng, J. Eisenstein, R. Agarwal, A. Agarwal, J. Berant, and A. Kumar.
 615 Rewarding progress: Scaling automated process verifiers for llm reasoning. *International Conference on
 616 Learning Representations*, 2025.

617 Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo.
 618 Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint
 619 arXiv:2402.03300*, 2024.

620 L. S. Shapley. Stochastic games. In *Proceedings of the national academy of sciences*. National Academy of
 621 Sciences, 1953.

622

623 Y.-H. Shen, C.-Y. Wu, Y.-R. Yang, Y.-L. Tai, and Y.-T. Chen. Mitigating cross-modal distraction and ensuring
 624 geometric feasibility via affordance-guided, self-consistent mllms for food preparation task planning. *arXiv
 625 preprint arXiv:503.13055*, 2025.

626 I. Shenfeld, J. Pari, and P. Agrawal. RL's razor: Why online reinforcement learning forgets less. *arXiv preprint
 627 arXiv:2509.04259*, 2025.

628

629 T. Shi, A. Karpathy, L. Fan, J. Hernandez, and P. Liang. World of bits: An open-domain platform for
 630 web-based agents. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference
 631 on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3135–3144. PMLR,
 632 2017.

633 M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht. Alfworld: Aligning text and
 634 embodied environments for interactive learning. *International Conference on Learning Representation*,
 635 2021.

636 Y. Song, F. Xu, S. Zhou, and G. Neubig. Beyond browsing: Api-based web agents. *arXiv preprint
 637 arXiv:2410.16464*, 2024a.

638

639 Y. Song, D. Yin, X. Yue, J. Huang, S. Li, and B. Y. Lin. Trial and error: Exploration-based trajectory
 640 optimization for llm agents. In *ACL*, 2024b.

641 J. M. Springer, S. Goyal, K. Wen, T. Kumar, X. Yue, S. Malladi, G. Neubig, and A. Raghunathan. Overtrained
 642 language models are harder to fine-tune. In *International Conference on Machine Learning*, 2025.

643

644 Y. Tan, Z. Dou, W. Wang, M. Wang, W. Chen, and J.-R. Wen. Htmrlrag: Html is better than plain text for
 645 modeling retrieved knowledge in rag systems. In *Proceedings of the ACM on Web Conference*, 2025.

646 Z. Wan, Z. Dou, C. Liu, Y. Zhang, D. Cui, Q. Zhao, H. Shen, J. Xiong, Y. Xin, Y. Jiang, C. Tao, Y. He,
 647 M. Zhang, and S. Yan. Srpo: Enhancing multimodal llm reasoning via reflection-aware reinforcement
 648 learning. *arXiv preprint arXiv:2506.01713*, 2025a.

649 Z. Wan, Y. Li, X. Wen, Y. Song, H. Wang, L. Yang, M. Schmidt, J. Wang, W. Zhang, S. Hu, and Y. Wen. Rema:
 650 Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*,
 651 2025b.

652

653 Y. Wang, P. Ji, K. Li, B. Bi, T. Feng, and G. Sartoretti. Beyond policy optimization: A data curation flywheel
 654 for sparse-reward long-horizon planning. *arXiv preprint arXiv:2508.03018*, 2025a.

655 Z. Wang, K. Wang, Q. Wang, P. Zhang, L. Li, Z. Yang, X. Jin, K. Yu, M. N. Nguyen, L. Liu, E. Gottlieb,
 656 Y. Lu, K. Cho, J. Wu, L. Fei-Fei, L. Wang, Y. Choi, and M. Li. Ragen: Understanding self-evolution in llm
 657 agents via multi-turn reinforcement learning, 2025b.

658 Y. Wei, O. Duchenne, J. Copet, Q. Carboneaux, L. Zhang, D. Fried, G. Synnaeve, R. Singh, and S. I. Wang.
 659 Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. In *Advances in*
 660 *Neural Information Processing Systems*, Proceedings of Machine Learning Research. PMLR, 2025a.

661 Z. Wei, W. Yao, Y. Liu, W. Zhang, Q. Lu, L. Qiu, C. Yu, P. Xu, C. Zhang, B. Yin, H. Yun, and L. Li.
 662 Webagent-rl: Training web agents via end-to-end multi-turn reinforcement learning. *Empirical Methods in*
 663 *Natural Language Processing*, 2025b.

664 J. Wu, B. Li, R. Fang, W. Yin, L. Zhang, Z. Tao, D. Zhang, Z. Xi, G. Fu, Y. Jiang, P. Xie, F. Huang, and
 665 J. Zhou. Webdancer: Towards autonomous information seeking agency. In *Advances in Neural Information*
 666 *Processing Systems*, Proceedings of Machine Learning Research. PMLR, 2025a.

667 J. Wu, W. Yin, Y. Jiang, Z. Wang, Z. Xi, R. Fang, L. Zhang, Y. He, D. Zhou, P. Xie, and F. Huang. Webwalker:
 668 Benchmarking llms in web traversal. *ACL*, 2025b.

669 M. Wu, M. Li, J. Yang, J. Jiang, K. Yan, Z. Li, M. Zhang, and K. Nahrstedt. Aha moment revisited: Are vlmss
 670 truly capable of self verification in inference-time scaling? *arXiv preprint arXiv:2506.17417*, 2025c.

671 Q. Wu, G. Bansal, J. Zhang, and et al. Autogen: Enabling next-gen llm applications via multi-agent
 672 conversation. *arXiv preprint arXiv:2308.08155*, 2025d.

673 W. Xiong, Y. Song, Q. Dong, B. Zhao, F. Song, X. Wang, and S. Li. Mpo: Boosting llm agents with meta
 674 plan optimization. In *EMNLP Findings*, 2025.

675 A. Yang, A. Li, B. Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

676 K. Yang, Y. Liu, S. Chaudhary, R. Fakoor, P. Chaudhari, G. Karypis, and H. Rangwala. Agentoccam: A
 677 simple yet strong baseline for llm-based web agents. *International Conference on Learning Representation*,
 678 2025b.

679 Z. Yang, P. Li, M. Yan, J. Zhang, F. Huang, and Y. Liu. React meets actre: Autonomous annotation of agent
 680 trajectories for contrastive self-training. In *CoLM*, 2024.

681 S. Yao, H. Chen, J. Yang, and K. Narasimhan. Webshop: Towards scalable real-world web interaction
 682 with grounded language agents. In *Advances in Neural Information Processing Systems*, Proceedings of
 683 Machine Learning Research. PMLR, 2022.

684 S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and
 685 acting in language models. *International Conference on Learning Representation*, 2023.

686 Q. Yu, Z. Zhang, R. Zhu, and others. Dapo: An open-source llm reinforcement learning system at scale.
 687 *arXiv preprint arXiv:2503.14476*, 2025.

688 S. Zeng, Q. Wei, W. Brown, O. Frunza, Y. Nevmyvaka, and M. Hong. Reinforcing multi-turn reasoning in
 689 llm agents via turn-level credit assignment. *arXiv preprint arXiv:2505.11821*, 2025.

690 Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma. Investigating the catastrophic forgetting in
 691 multimodal large language models. In *Conference on Parsimony and Learning*, 2024.

692 W. Zhang, T. Hu, Y. Qiao, H. Zhang, Y. Qin, Y. Li, J. Liu, T. Kong, L. Liu, and X. Ma. Chain-of-action:
 693 Trajectory autoregressive modeling for robotic manipulation. *arXiv preprint arXiv:2506.09990*, 2025.

694 Z. Zhang, S. Tian, L. Chen, and Z. Liu. Mmina: Benchmarking multihop multimodal internet agents, 2024.

695 X. Zhao, Z. Kang, A. Feng, S. Levine, and D. Song. Learning to reason without external rewards. *arXiv*
 696 *preprint arXiv:2505.19590*, 2025a.

705 Y. Zhao, Y. Liu, J. Liu, J. Chen, X. Wu, Y. Hao, T. Lv, S. Huang, L. Cui, Q. Ye, F. Wan, and F. Wei.
706 Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025b.

707

708 C. Zheng, P. Ke, Z. Zhang, and M. Huang. Click: Controllable text generation with sequence likelihood
709 contrastive learning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association
710 for Computational Linguistics: ACL 2023*, pages 1022–1040, Toronto, Canada, July 2023. Association for
711 Computational Linguistics.

712

713 C. Zheng, S. Liu, M. Li, X.-H. Chen, B. Yu, C. Gao, K. Dang, Y. Liu, R. Men, A. Yang, J. Zhou, and J. Lin.
714 Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

715

716 S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, Y. Bisk, D. Fried, U. Alon, et al. Webarena:
717 A realistic web environment for building autonomous agents. *International Conference on Learning
Representation*, 2023a.

718

719 W. Zhou, Y. E. Jiang, L. Li, et al. Agents: An open-source framework for autonomous language agents. *arXiv
preprint arXiv:2309.07870*, 2023b.

720

721 W. Zhou, Y. Ou, S. Ding, et al. Symbolic learning enables self-evolving agents. *arXiv preprint
arXiv:2406.18532*, 2024a.

722

723 Y. Zhou, A. Zanette, J. Pan, S. Levine, and A. Kumar. Archer: Training language model agents via hierarchical
724 multi-turn rl. In *International Conference on Machine Learning*, 2024b.

725

726 Y. Zhou, S. Jiang, Y. Tian, J. Weston, S. Levine, S. Sukhbaatar, and X. Li. Sweet-rl: Training multi-turn llm
727 agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025.

728

729 D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving.
730 Fine-tuning language models from human preferences. In *Advances in Neural Information Processing
Systems*, Proceedings of Machine Learning Research. PMLR, 2022.

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752 **A LLM USAGE DISCLOSURE**

753
 754 We used Large Language Models (LLMs), specifically OpenAI’s ChatGPT, to aid in writing and editing. The
 755 LLM assisted with grammar, phrasing, and improving clarity, but did not contribute to research ideation,
 756 experimental design, analysis, or result interpretation. All content was authored and verified by the human
 757 authors.

758 **B FULL RELATED WORK**

759 **B.1 WEB BENCHMARKS AND AGENTS**

760 The development of web-browsing agents has progressed hand-in-hand with the creation of benchmarks
 761 designed to evaluate their reasoning and interaction capabilities. Early efforts, such as WoB (Shi et al., 2017)
 762 and MiniWoB++ (Liu et al., 2018), introduced synthetic environments composed of simple website widgets.
 763 These benchmarks supported basic mouse and keyboard interactions but lacked the complexity and variability
 764 required to reflect real-world web tasks.

765 To more closely approximate practical scenarios, WebShop (Yao et al., 2022) simulated a large-scale e-
 766 commerce platform containing rich product data and goal-directed user instructions. While offering more
 767 realistic content than earlier benchmarks, its scope was confined to a single domain with limited interactivity,
 768 constraining its utility for evaluating general-purpose web agents. As LLMs have become increasingly capable
 769 of processing structured web data such as HTML and DOM trees (Tan et al., 2025), newer benchmarks have
 770 sought broader coverage and greater realism. For example, Mind2Web (Deng et al., 2023) extended the task
 771 set to diverse websites using human-recorded demonstrations. Nevertheless, its static nature and limited
 772 interactivity restrict its ability to evaluate adaptive behavior.

773
 774 More recent benchmarks have emphasized interactivity reproducibility, and higher fidelity. WebWalkerQA (Wu et al., 2025b) evaluates text-based reasoning abilities in a question–answer format while constraining
 775 actions to simple ‘click’ operations, thereby focusing on navigation and information-seeking capabilities.
 776 BrowseComp-VL (Geng et al., 2025) further increases task complexity by requiring agents to perform
 777 multi-step information retrieval involving both visual and textual inputs. WebArena (Zhou et al., 2023a)
 778 introduced a suite of simulated websites with real functionality, enabling realistic and controlled experimen-
 779 tation and quickly becoming a central benchmark for testing LLM-based web agents in dynamic settings.
 780 VisualWebArena (Koh et al., 2024a) extended WebArena by integrating visual understanding with language
 781 processing.

782
 783 Building on these benchmarks, a diverse array of web agents has emerged (Nakano et al., 2022; Wei et al.,
 784 2025b; Wu et al., 2025b; Qi et al., 2025; Zhou et al., 2023b; 2024a). Broadly, these agents can be grouped
 785 into three categories. First, domain-specific agents typically rely on smaller LLMs trained or fine-tuned to
 786 select relevant HTML elements or execute low-level actions (Furuta et al., 2024; Deng et al., 2024). Second,
 787 prompt-based agents leverage large foundation models orchestrated through prompting strategies or modular
 788 tool-use workflows to tackle complex navigation tasks (Song et al., 2024a; Koh et al., 2024b; He et al., 2024).

789
 790 Our work aligns with a third line of research: post-training methods for improving web agents. Prior
 791 approaches in this direction typically reduced web scenarios to question–answer pairs (Wu et al., 2025a;
 792 Li et al., 2025; Geng et al., 2025) or relied on sparse, trajectory-level rewards provided only upon task
 793 completion (Qi et al., 2025; Wei et al., 2025b). Such reward structures posed significant challenges for credit
 794 assignment and efficient exploration, particularly in long-horizon settings. For instance, WebRL (Qi et al.,
 795 2025) trained a reward model that judges success solely based on the final state of the webpage, assuming that
 796 reaching the correct page equates to task completion. In practice, however, agents may navigate to the correct
 797 page yet fail to extract or generate the required output. WebAgent-R1 (Wei et al., 2025b) performed clipping at
 798 the token level and aggregates the resulting loss directly, introducing bias and leads to training instability over
 long sequences. In addition to these algorithmic limitations, many RL-based approaches assumed simplified

environments, overlooking the challenges inherent in multi-turn interactions and multimodal observation handling in real-world web settings (Qi et al., 2025; Wei et al., 2025b; Wu et al., 2025a).

Our work addresses these gaps by incorporating fine-grained, agent-specific reward functions tailored to sub-goal completion and action relevance within a multi-agent setting. We further extend agent capabilities to handle multi-turn interactions and visual inputs, enabling robust behavior in visually complex and dynamic web environments (Abuelsaad et al., 2024; Iong et al., 2024). Moreover, we propose a turn-level training objective designed for stable optimization in hierarchical multi-agent settings, mitigating the instability and credit-assignment issues observed in prior methods. Other efforts in the multi-agent setting for web tasks (Erdogan et al., 2025) have proposed synthetic data generation pipelines for fine-tuning. While orthogonal to our focus, these methods could be conceptually integrated with our approach in future work.

B.2 REINFORCEMENT LEARNING FOR LLMs

Reinforcement learning has proven effective for aligning LLMs with downstream objectives, but most existing approaches have focused on single-turn settings (Shao et al., 2024; Casper et al., 2024; Ouyang et al., 2022; Ziegler et al., 2022; Christiano et al., 2017). Algorithms such as PPO (Ouyang et al., 2022; Ramamurthy et al., 2023), GRPO (Shao et al., 2024), and DPO (Rafailov et al., 2023) became standard in this regime. In contrast, multi-turn long-horizon applications, where an agent must reason, plan and act through sequential interactions, pose optimization difficulties that remain insufficiently addressed in domains such as web navigation (Zhou et al., 2024b; Wei et al., 2025b), embodied planning (Fei et al., 2025), and multi-turn mathematical reasoning (Wan et al., 2025b; Zheng et al., 2025).

Many studies (Chen et al., 2025b; Jin et al., 2025; Feng et al., 2025a) framed multi-turn tasks as bandit problems, relying solely on outcome-level rewards such as answer or format correctness. This formulation was inadequate for long-horizon reasoning because it treated the entire trajectory as a single decision step and ignored turn-level signals indicating whether intermediate steps were helpful or harmful. A natural remedy was to learn a process-level reward model or critic model, but this typically required expensive on-policy data collection and did not generalize well with limited fine-tuning data (Setlur et al., 2025; Zhou et al., 2024b; 2025).

Another line of work attempted to directly adapt successful single-turn algorithms to multi-turn objectives (Wei et al., 2025b; Qi et al., 2025; Wang et al., 2025b). RAGEN (Wang et al., 2025b), for instance, concatenated all states, intermediate reasoning, and actions into a unified episode-level response, which created scalability challenges in long-horizon tasks. WebAgent-R1 and WebRL also lacked explicit turn-level credit assignment (Wei et al., 2025b; Qi et al., 2025). To address this, GiGPO (Feng et al., 2025b) introduced a two-level structure for estimating relative advantage with an additional anchor-state grouping mechanism, but this approach struggled in highly complex environments where identical states were hard to detect due to noise or subtle differences. Zeng et al. (2025) incorporated a turn-level advantage estimation strategy to enable more precise credit assignment in multi-turn agent interactions, but it was evaluated only in two-turn tool-use settings. IPO (Fei et al., 2025) modeled turn-level optimization by treating different tokens in GRPO as distinct decision steps. Although this better reflected multi-turn structure, the cumulative product of token-level ratios shrunk rapidly as sequence length grew, hitting the clipping threshold early and introducing exponentially worsening bias.

Our proposed algorithm, HIMPO, leverages a turn-level ratio combined with token-level averaging strategies (Wan et al., 2025b; Zheng et al., 2025) and a two-round post-training scheme. In the first round, we introduce fine-grained reward function to encourage exploration, and in the second round, we apply standard task rewards to promote improved generalization.

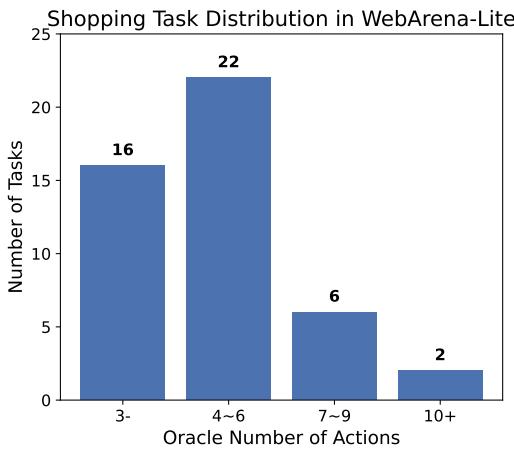


Figure 5: Distribution of shopping tasks in WebArena-Lite, categorized by the oracle number of required actions. The dataset includes 46 tasks spanning a range of interaction complexities.

C PROPERTIES OF WEB BROWSING TASKS

C.1 CHALLENGE OF MULTI-TURN WEB BROWSING TASKS

In Figure 1, we empirically show that web browsing tasks become increasingly challenging as the number of interactions and action executions grows. This difficulty can also be understood through the lens of MDPs, particularly the concept of MDP depth (Pignatelli et al., 2024). Following (Ni et al., 2023), we define the depth of an MDP as the number of temporal steps that intervene between a **key action** and its observable outcome. In long-horizon web tasks, agents often encounter bottleneck decisions, i.e., critical actions that are required for eventual success but whose effects manifest only after a long delay. For example, in an e-commerce setting, clicking the ‘log in’ button and entering credentials is a **key action**: it may not yield any immediate reward but is necessary for accessing the order history page or editing account information later in the task. Despite the existence of many viable action paths, all successful trajectories must pass through such bottlenecks. When these decisions are temporally distant from their consequences, the MDP becomes deep, making it harder for the agent to maintain causal coherence and long-term focus through the interaction. We address this challenge by deriving sub-goal decomposition in Section 3.2 and extending it to multi-turn interactions in Section 3.3. This decomposition exposes bottleneck decisions as explicit intermediate objectives, reducing MDP depth and improving execution reliability by anchoring the agent’s reasoning to structurally important waypoints.

C.2 ANNOTATORS & ANNOTATION PROTOCOL

To support the observation that task difficulty increases with the number of required actions for an LLM agent (as illustrated in Figure 1), we perform a detailed analysis of the shopping category in WebArena-Lite (Zhou et al., 2023a; Liu et al., 2025a), which includes 46 diverse tasks. These tasks fall into three broad categories: webpage navigation, question answering, and content modification. We define the oracle action count as the minimal number of environment interactions needed to successfully complete a task. These oracle counts, grouped into discrete complexity ranges, are shown in Figure 5.

Among the three task types, webpage navigation is the most challenging to define optimally. This is due to the presence of multiple valid search strategies: basic keyword search, advanced search with filters, or navigating through category hierarchies. Since success in these tasks is judged by reaching the correct final

893 **Algorithm 1** Hierarchical Interactive Multi-turn Policy Optimization (HIMPO)

894 **Input:** π_θ , Dataset \mathcal{D} , ϵ , learning rate for planner and action executor α_p, α_a , batch size B .

895 **Output:** Parameter for the agent policy θ .

896 1: Initialize planner $\pi_p = \pi_\theta(\cdot|S_p)$, action executor $\pi_a = \pi_\theta(\cdot|S_a)$ with system prompt S_p and S_a

897 2: **for** each episode **do**

898 3: **for** $i = 1 : B$ **do**

899 4: $(\mathbf{x}, \mathbf{e}) \sim \mathcal{D}$

900 5: collect group of trajectories in equation 4

901 6: **end for**

902 7: estimate advantage in equation 8

903 8: calculate $\mathcal{J}_{\text{HIMPO}}^{\text{plan}}(\theta)$ in equation 5

904 9: $\theta \leftarrow \theta + \alpha_p \nabla_\theta \mathcal{J}_{\text{HIMPO}}^{\text{plan}}(\theta)$

905 10: **for** $i = 1 : B$ **do**

906 11: $(\mathbf{x}, \mathbf{e}) \sim \mathcal{D}$

907 12: collect group of trajectories in equation 4

908 13: **end for**

909 14: estimate advantage in equation 8

910 15: calculate $\mathcal{J}_{\text{HIMPO}}^{\text{action}}(\theta)$ in equation 5

911 16: $\theta \leftarrow \theta + \alpha_a \nabla_\theta \mathcal{J}_{\text{HIMPO}}^{\text{action}}(\theta)$

912 17: **end for**

913 URL, identifying the optimal path requires knowledge of that target in advance. Annotators leveraged the
 914 known final URL from (Zhou et al., 2023a; Liu et al., 2025a) to retrospectively trace the shortest path back to
 915 the start, yielding a reliable estimate of the oracle trajectory.

916
 917 Despite the fact that LLM agents are evaluated solely on task success rather than trajectory optimality, we
 918 observe that the choice of search strategy implicitly affects success rate. In 7 shopping tasks with oracle
 919 action counts in the 4–6 range, the optimal strategy is advanced search or category navigation. However, LLM
 920 agents often defaults to basic keyword search, which introduces ambiguity. In cases where advanced search is
 921 optimal, keyword search typically leads to longer horizons, requiring inspection of multiple irrelevant pages.
 922 In other cases, the keyword search fails altogether if the query does not match any product title, whereas
 923 structured category navigation would succeed. For these 7 tasks, the success rate drop from 54.5% (across all
 924 tasks in the 4–6 oracle range) to 28.6% when isolating just those requiring more deliberate search strategies.

925
 926 In contrast, question answering and content modification tasks are more straightforward for annotators. After
 927 logging in (if required), most steps involved clicking buttons on the same page or conducting linear navigation,
 928 such as browsing through paginated content (e.g., reviewing order history). These task types generally pose
 929 fewer ambiguities and show less performance degradation.

930
 931 **D ALGORITHM DESIGN**

932
 933 Algorithm 1 summarizes the practical implementation of HIMPO, which optimizing both planner and
 934 action executor agents using the alternating procedure until convergence. In each phase, a query \mathbf{x} and the
 935 corresponding environment \mathbf{e} are sampled from dataset \mathcal{D} . After collecting enough trajectories, we compute
 936 advantage values via equation 8 for updating parameters θ with different learning rate, inspired by two-time
 937 scale analysis (Das et al., 2024). Note that both agents share the same parameter θ in practice, so the entire
 938 training process can be viewed as a multi-objective optimization, where planner and action executor have
 939 their own objectives with the same parameters θ .

940 D.1 MITIGATING SFT WITH RELIABLE TOOLS
941

942 Supervised fine-tuning (SFT) is commonly used as a warm start for RL post-training, helping models
943 memorize task-specific rules. However, SFT has well-known limitations, including poor generalization to
944 out-of-distribution (OOD) scenarios (Chu et al., 2025) and susceptibility to catastrophic forgetting (Shenfeld
945 et al., 2025). We observe that in structured tasks that can be decomposed hierarchically, the reliance on SFT
946 can be partially mitigated through the use of reliable external tools. For instance, tools like Playwright for web
947 browsing (Zhou et al., 2023a) or PID controllers for drone navigation (Hsu et al., 2024b) can handle low-level
948 execution reliably, reducing the need for the model to memorize procedural rules via SFT. Moreover, SFT
949 demands large-scale, high-quality training data, which can be difficult to obtain and may not align with the
950 deployment distribution. By leveraging stable tool use, we reduce this dependency and enable more robust
951 and generalizable learning.

952 D.2 TWO-ROUND CURRICULUM: DROPPING AND REINTRODUCING KL DIVERGENCE
953

954 We adopt a two-round curriculum using HIMPO. In the first round, we remove the KL divergence term
955 typically used in RL post-training. KL divergence acts as a regularizer by penalizing deviation from a
956 reference policy, which is usually a SFT model (Ouyang et al., 2022; Shao et al., 2024). However, since
957 we do not perform SFT prior to first-round RL, the only available reference would be the raw base model.
958 Constraining the policy to stay close to this untuned model provides little practical benefit and may hinder
959 effective learning. Removing the KL term not only simplifies training by eliminating the need to tune its
960 coefficient but also reduces memory and computational overhead. More importantly, it enables more flexible
961 exploration (Wan et al., 2025b), which is crucial in the early stages of learning. From a methodological
962 perspective, general-purpose LLMs are not inherently optimized for dynamic or multi-turn reasoning (Feng
963 et al., 2025c); thus, pattern drift during RL is both expected and beneficial. Removing the KL constraint
964 allows the policy to adapt more freely to such interactions. In the second round of training, we reintroduce
965 the KL divergence term to encourage the policy to remain closer to the improved reference policy obtained
966 from the first round, thereby stabilizing further optimization.

967 D.3 TURN-LEVEL RATIO AND POLICY GRADIENT LOSS
968

969 To unify our objectives under multi-turn interaction settings, we define a turn-level ratio based on sequence
970 likelihood (Zheng et al., 2023; 2025), as shown in Equations equation 6 and equation 7. Unlike token-level
971 formulations, the turn-level ratio aligns more closely with the underlying MDP structure by treating all tokens
972 within a turn as a single action. This enables us to apply clipping at the turn level (Wan et al., 2025b), which
973 improves stability and consistency in policy optimization.

974 Building on this formulation, we further compute the policy gradient loss at the turn level. Specifically, the
975 loss is averaged over the total number of turns across all sampled trajectories, representing a multi-turn variant
976 of DAPO (Yu et al., 2025). This design helps address two key limitations commonly observed in previous
977 multi-turn methods (Fei et al., 2025; Wei et al., 2025b; Wan et al., 2025b).

978 First, in long trajectories consisting of multiple high-quality steps, prior approaches often dilute the learning
979 signal across many individual actions, weakening the model’s ability to capture reasoning-relevant patterns.
980 Second, excessively long turns, which often contains low-quality content such as repetition or irrelevant
981 actions, tend to be under-penalized. Our turn-level formulation mitigates both issues by emphasizing coherent
982 turn-level decision-making and discouraging verbose, low-quality outputs.

983 D.4 REWARD DESIGN
984

985 Our Hierarchical Multi-Agent Intrinsic Policy Optimization (HIMPO) is trained in two stages, each de-
986 signed to balance learning stability and generalization for multi-turn tasks that benefit from dividing labor
987 between strategic planning and concrete action execution. In both rounds, agents optimize task success via
988 reinforcement learning, but we structure rewards differently to facilitate learning.

987 **Two-Round Training Procedure**

988

- 989 • First Round (Dense + Format-Aware Rewards): We employ dense, role-specific rewards for both
990 planner and executor, supplemented with penalties for incorrect output $r_t^P = -0.5$ formatting and
991 exceeding episode limit $r_t^P = -0.1$. This setup promotes stable credit assignment and reduces noise
992 during early training. We also optionally use dynamic sampling (Yu et al., 2025) to over-sample
993 and filter out outputs with format errors, ensuring that policy updates are based on well-formed and
994 executable actions.
- 995 • Second Round (Sparse Final Reward Only): Once the agents have developed stable behavior, we
996 remove all intermediate and format-related signals and train using only the final binary task reward
997 ($r_T \in \{0, 1\}$), defined in Section 3.1. This second round encourages agents to generalize beyond
998 hand-crafted signals and optimize purely for end-task success.

999 The following subsections detail our fine-grained reward design for the first round, which enables efficient
1000 learning within the hierarchy.

1001 **D.4.1 REWARD FOR THE LOW-LEVEL EXECUTOR**

1002 Separating the planner from the action executor in a hierarchical framework naturally creates sub-goals for
1003 the executor as the high-level query is decomposed into a sequence of planning steps. This setup allows us to
1004 provide a step-wise reward measuring how closely the executor’s actions align with the planner’s expectations.
1005 Specifically, we define the step-wise reward for the action executor as $r_t^s = 1/T_{max}$, where T_{max} is the
1006 predefined time step limit.

1007 **D.4.2 REWARD FOR THE HIGH-LEVEL PLANNER**

1008 Designing a dense reward for the planner is more challenging because its plans do not immediately yield
1009 observable environmental outcomes. We therefore introduce an intrinsic reward (Kang et al., 2025; Zhao
1010 et al., 2025a) based on the planner’s own confidence in its generated plans. The intuition is to incentivize the
1011 planner to produce decisive and unambiguous plans, which are more likely to be coherent and executable by
1012 the low-level action executor.

1013 For plan step \mathbf{m}_t ³, the intrinsic reward is defined as the average confidence score across its N tokens:

$$1014 r_t^s(\mathbf{m}_t | o_t^{text}) = \frac{1}{N} \sum_{i=1}^N \text{KL}(p_{\pi_\theta}(\cdot | o_t^{text}, \mathbf{m}_{t,< i}) || U_v) \quad (9)$$

1015 , where $p_{\pi_\theta}(\cdot | o_t^{text}, \mathbf{m}_{t,< i})$ is the planner’s next-token distribution conditioned on the current observation and
1016 previously generated tokens, and U_v is the uniform distribution over the model’s entire vocabulary.

1017 In the first round of HIMPO, we combine dense, role-specific rewards with the final task reward $r_T \in \{0, 1\}$
1018 to improve exploration and credit assignment. This phase acts as a curriculum, incorporating format-aware
1019 supervision and optional dynamic sampling to ensure stable early learning. In the second round, we remove
1020 these auxiliary signals and train using only the task reward for both agents, encouraging generalization and
1021 end-to-end task success.

1022 **E EXPERIMENT DETAILS**1023 **E.1 EXPERIMENT SETUPS**

1024 **Environments and Datasets** We conduct our experiments in the WebArena environment (Zhou et al.,
1025 2023a), a realistic and self-hostable platform for web agents. It provides automatic success evaluation via
1026 rule-based rubrics, such as detecting confirmation messages or checking for expected content on a web page.

1027 ³We omit the superscript of \mathbf{m}_t^a because we exclude vision executor during training.

1034 Tasks span multiple domains, including social forums (Reddit), collaborative coding (GitLab), e-commerce
 1035 content management systems (CMS), open street maps (Map), and online shopping (Shopping).
 1036

1037 Agent interactions are defined using a condensed action space of 12 high-level actions (Koh et al., 2024a),
 1038 implemented through the Playwright library. This abstraction captures core web navigation and interaction
 1039 behaviors. The full action set is shown in Table 5.

1040
 1041 Table 5: Action space leveraging Playwright library

Action Type	Description
click [elem]	Click on element elem
hover [elem]	Hover on element elem
type [elem] [text]	Type text on element elem
press [key_comb]	Press a key combination
new_tab	Open a new tab
tab_focus [index]	Focus on the i-th tab
tab_close	Close current tab
goto [url]	Open URL
go_back	Click the back button
go_forward	Click the forward button
scroll [up down]	Scroll up or down the page
stop [answer]	End the task with an output

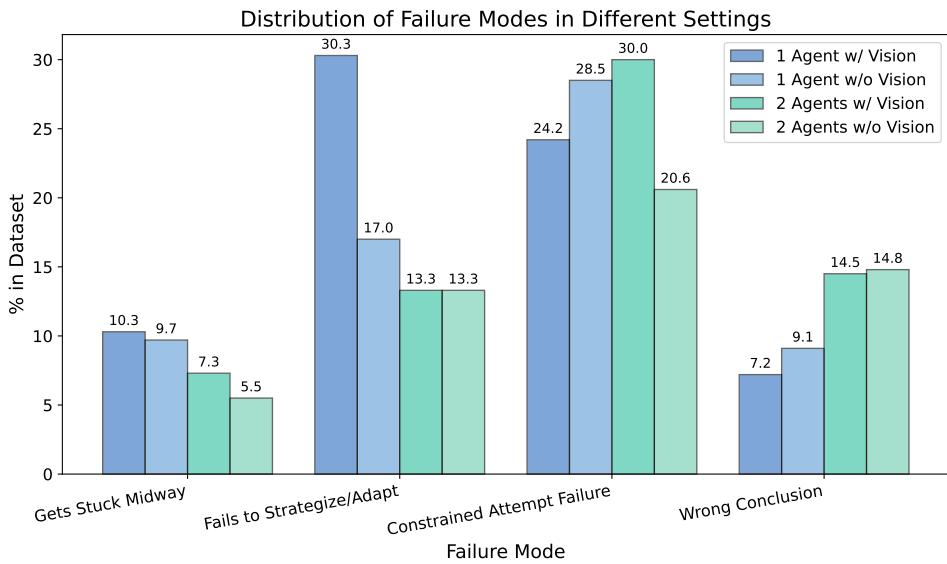
1055
 1056
 1057
 1058 To support multi-modal perception, the agent’s observation space \mathcal{O} is derived from the accessibility tree (Zhou
 1059 et al., 2023a), a structured and compact subset of the DOM tree. Each node includes its element ID, semantic
 1060 role, textual content, and relevant properties (e.g., focusability). For visual elements, the corresponding
 1061 images are downloaded and tagged with the associated element ID, enabling cross-modal grounding for
 1062 vision-capable agents.

1063 We evaluate on the following benchmarks:

1064

- WebArena-Lite (Zhou et al., 2023a; Liu et al., 2025a): 165 tasks across the WebArena domains.
 1065 Each task includes a high-level natural language instruction, with oracle solutions averaging 10 steps.
 1066 Visual understanding is not required to solve these tasks.
- VisualWebArena (Zhang et al., 2024): Designed to evaluate the capabilities of vision-language
 1067 agents in web environments. It includes tasks that require interpreting visual elements such as
 1068 identifying an object’s shape or color, as well as recognizing higher-level visual features. We focus
 1069 on Reddit and Shopping tasks, which also appear in the WebArena-Lite benchmark. Specifically,
 1070 Reddit domain contains 38 tasks and Shopping domain includes 76 tasks.

1071 **Baselines for comparison** We evaluate DEPART using Qwen3-4B (Yang et al., 2025a) and Claude
 1072 3.7 (Anthropic, 2025). Each model is tested under three agent configurations to assess performance differences:
 1073 a single-agent setup that unifies planning and execution; a two-agent setup that separates planning from
 1074 execution; and a three-agent setup that further distinguishes between the planner, a vision-based executor,
 1075 and an action executor. Because Qwen3-4B is text-only models, we incorporate Claude 3.7 as the vision
 1076 executor in the three-agent configuration, while retaining Qwen3-4B for the planning and action execution
 1077 components.



1100 Figure 6: Analysis of error type distribution for Claude 3.7 under single-agent and hierarchical multi-agent
1101 configurations, comparing performance with (w/) and without (w/o) vision input.

1102
1103
1104 In the WebArena-Lite benchmark, we further compare DEPART against a range of competitive baselines,
1105 including both open-source and proprietary models. These consist of general-purpose large language models
1106 (e.g., Qwen2.5, Llama3.1, GPT-4) and reasoning-specialized models (e.g., QwQ, OpenAI), as reported
1107 by (Wei et al., 2025b). We also evaluate against RL post-training agents specifically trained for HTML-based
1108 decision-making tasks, such as WebRL (Qi et al., 2025) and WebAgent-R1 (Wei et al., 2025b).

1109 **Evaluation metrics** We evaluate our system using complementary scoring strategies, adopting three
1110 evaluation criteria: (1) Exact Match, where an agent’s response must exactly match the expected token
1111 sequence; (2) Must Include, which checks for the presence of essential task-specific keywords, marking
1112 a failure if any are missing; and (3) Fuzzy Match, which leverages a language model to assess semantic
1113 similarity between the agent’s response and a reference answer via inference-based prompts. This combination
1114 of complementary metrics allows for both strict and flexible judgment.

1115 E.2 WEBARENA-LITE

1116 **Distribution analysis of error types** As shown in the previous results, multi-agent systems consistently
1117 outperform the single-agent setting in the WebArena-Lite benchmark. Additionally, we empirically observe
1118 instances of cross-modal distraction (Shen et al., 2025), particularly when vision input is present. To further
1119 understand agent performance, we analyze the distribution of four primary error types: *Get Stuck Midway*,
1120 *Fails to Strategize/Adapt*, *Constrained Attempt Failure*, and *Wrong Conclusion* (see Figure 6).

1121 The Get Stuck Midway error typically occurs when the agent enters a loop, repeating the same sequence
1122 of actions without making progress. This issue is often linked to limited planning capacity. The Fails to
1123 Strategize/Adapt error reflects a failure to revise plans in response to failure signals. For instance, retrying the
1124 same approach even when it has consistently failed. Both of these error types can be mitigated by separating
1125 planning and execution across different agents, which allows for greater specialization. Additionally, removing
1126 irrelevant vision input reduces cognitive load, allowing the agent to better focus and fully leverage its
1127 capabilities. These design choices contribute to the observed reduction in these failure modes.

1128 The remaining two error types—Constrained Attempt Failure and Wrong Conclusion—are less directly
 1129 addressed by our multi-agent architecture. Constrained Attempt Failure occurs when the agent fails to
 1130 complete a reasonable attempt within a predefined time step limit, often due to system constraints or
 1131 neglecting necessary setup steps (e.g., forgetting to log in). Wrong Conclusion refers to cases where the agent
 1132 either navigates to an incorrect page or fails to generate a fully correct response despite reaching the right
 1133 destination.

1134 Although the two-agent system without vision does not achieve the lowest error rate in all categories, it
 1135 achieves the lowest overall error rate, and performs best across three of the four failure types. The only
 1136 exception is Wrong Conclusion, which often reflects the final stage of failure—when the agent has nearly
 1137 completed the task but falls short on comprehension or reasoning. This suggests that our proposed architecture
 1138 significantly improves robustness in earlier stages of the task (e.g., planning, adaptation, navigation), with the
 1139 remaining challenges concentrated in the final decision-making step.

1140 In summary, while our framework does not directly target Constrained Attempt Failure or Wrong Conclusion,
 1141 the overall performance gains indicate that some tasks previously failing due to planning or adaptation issues
 1142 are now able to proceed further, though they may still fail due to system constraints or subtle reasoning errors.

1144 **Ablation Studies of RL Post-training**

- 1146 • **M1 – MT-GRPO-token** : token-level ratio similar to WebAgent-R1 (Wei et al., 2025b) with only
 1147 binary task reward. The key differences between M1 and WebAgent-R1 is that M1 does not have
 1148 supervised fine-tuning and the backbone models are not the same.
- 1149 • **M2 – MT-GRPO-token (Dense Reward)**: as M1 but augmented with our designed dense reward in
 1150 Appendix D.4. Since we only have a single-agent here, the role-specific reward function for M2 is its
 1151 intrinsic reward as the average confidence score, similar to equation 9 but with y_t as output instead
 1152 of m_t
- 1153 • **M3 – MT-GRPO-turn (Dense Reward)**: as M2 but replacing token-level ratio with turn-level ratio
 1154 clipping.
- 1155 • **M4 – MT-GRPO-turn (Dense Reward) + dynamic sampling**: as M3 with dynamic sampling of
 1156 rollouts (Yu et al., 2025), which over-sampling and filtering the prompts with outputs in incorrect
 1157 format.
- 1158 • **M5 – HIMPO (Trainer Planning with Dense Reward)**: train only the planner agent with dense
 1159 reward and penalties.
- 1160 • **M6 – HIMPO**: train both planner and executor jointly with dense reward and penalties.
- 1161 • **M7 – Joint Dynamic Sampling**: as M6 with dynamic sampling enabled.
- 1162 • **M7+M8 – HIMPO (2 rounds)**: continue training the model from M7 using only task reward (plus
 1163 format/episode penalties) for both agents.

1168 E.3 VISUALWEBARENA

1169 In Section 5, we present experimental results on WebArena-Lite (Zhou et al., 2023a; Liu et al., 2025a),
 1170 highlighting the potential impact of cross-modal distraction (Shen et al., 2025). As shown in Table 7,
 1171 incorporating visual input consistently improves performance across different agent configurations. Notably,
 1172 the 2-agent configuration using Claude 3.7 with vision achieves the highest success rate. This suggests that
 1173 the 3-agent setup may introduce slight coordination challenges or information loss that directly impacts task
 1174 success, making it a valuable direction for future investigation.

1175 **F IMPLEMENTATION DETAILS**
1176

1177 Our work consists of two main components: the multi-agent framework DEPART and the multi-agent RL
 1178 post-training algorithm HIMPO. The full system prompts for the planner, action executor, and vision executor
 1179 are provided below. For evaluation, DEPART is primarily tested with the proprietary model Claude 3.7, while
 1180 HIMPO is implemented on Qwen3-4B as the backbone model. We use Playwright as an external tool to
 1181 interact with the web environment, and RL post-training is performed exclusively on the original 647 training
 1182 tasks from WebArena (Zhou et al., 2023a) without any supervised fine-tuning (SFT). All experiments are
 1183 run on five Amazon EC2 g6e.48xlarge servers, each with 192 vCPUs, 768 GB of memory, and up to eight
 1184 NVIDIA L4 GPUs (48 GB each), which provides ample compute for large-scale multi-agent rollouts and RL
 1185 optimization.

1186
 1187 During RL post-training, we use a constant learning rate 5e-7 for planner and 1e-6 for action executor with a
 1188 batch size of 16 in RL post-training. The KL divergence regularization coefficient β and the clip ratio ϵ are
 1189 set to 0.001 and 0.2, respectively. The maximum number of new tokens is fixed at 2048. For efficient LLM
 1190 rollouts, we employ vLLM (Kwon et al., 2023) with a tensor-parallel size of 1 and a GPU memory utilization
 1191 ratio of 0.6. Rollout sampling is performed with a temperature of 0.7.

1192
1193 **LLM Prompt for Strategic Planning Agent**

1194
 1195 You are a high-level planning agent responsible for creating strategic plans to accomplish web-based
 1196 tasks with **HYBRID AGENT COORDINATION**. Your role is to analyze objectives, create step-by-
 1197 step plans, and manage the execution process for the low-level vision agent and action executor agent,
 1198 where the action executor agent accomplishes tasks through specific Playwright actions.
 1199 You must conduct reasoning inside `<think>` and `</think>` tags first every time you get new
 1200 information.

1201 You must maintain and update your plan inside `<plan>` and `</plan>` tags.

1202 After reasoning, perform actions using `<act>action_description</act>` tags.

1203 **CRITICAL PLAN PERSISTENCE RULES:**

- 1204 • When starting a NEW task with a NEW intent, create individual COMPLETE PLAN for
 1205 vision agent (optional) and execution agent, and store it.
- 1206 • Once a complete plan exists, MAINTAIN it across all steps — do not recreate unless
 1207 absolutely necessary.
- 1208 • For each step, decide between: NEXT_STEP (from existing plan), RETRY_CURRENT (same
 1209 step), or REPLAN_ENTIRELY (new plan).
- 1210 • Only REPLAN_ENTIRELY when there are fundamental issues that make the current plan
 1211 impossible.

1212 **Your primary responsibilities:**

- 1213 1. Analyze the given objective and create a complete step-by-step plan.
- 1214 2. Assign plan steps to BOTH vision and execution agents with clear goals and validation
 1215 criteria.
- 1216 3. Track which step you are currently assigning (step tracking is crucial).
- 1217 4. Based on BOTH vision and executor feedback, decide to: proceed to next step, replan
 1218 entirely, or retry current step.
- 1219 5. Coordinate between vision agent (for visual analysis) and execution agent (for actions).
- 1220 6. Provide clear goals and validation criteria for each step to both agents.
- 1221 7. **PRESERVE** the original complete plan across execution steps.

1222
1223

Execution Agent Prompt

1224
1225
1226
1227

You are an execution agent responsible for carrying out specific Playwright actions based on step-by-step plans from a planning agent and visual context from a vision agent in a hybrid architecture. Your role is to execute one plan step at a time using both textual and visual information, and then provide feedback back to the high-level planner agent.

1228
1229

After understanding the plan step and vision context, perform actions using `<act>action_description</act>` tags.

1230
1231
1232
1233
1234
1235
1236
1237**Your primary responsibilities:**

1. Follow the current step assignment from the planning agent.
2. Focus on achieving the step's specific goal and meeting validation criteria.
3. Utilize visual analysis and context provided by the vision agent.
4. Execute precise Playwright actions based on current webpage state and visual information.
5. Provide structured feedback about step completion, success, or issues to the planner.
6. Coordinate with vision agent insights to make informed action decisions.
7. Report detailed results so the planner can coordinate next actions for both agents.

1238

1239
1240
1241

Vision Analysis Agent Prompt

1242
1243
1244
1245
1246
1247
1248
1249

You are a vision analysis agent responsible for analyzing webpage screenshots and providing detailed visual descriptions to support planning and execution agents in a hybrid architecture. Your role is to understand visual elements and provide information to planner and action executor if required by the planner.

After understanding the plan step from the high-level planner, provide your visual information using `<act>visual_description</act>` tags to both planner and action executor agent.

1250
1251
1252
1253
1254
1255
1256**Your primary responsibilities:**

1. Follow the current step assignment from the planning agent.
2. Focus on achieving the step's specific goal and meeting validation criteria.
3. Analyze webpage screenshots to understand layout, elements, visual information, and downloaded images.
4. Give feedback to the planner about visual confirmation of completed actions.
5. Support the executor agent with detailed visual context for action execution.

1257

Table 6: Task success rate in WebArena-Lite (Zhou et al., 2023a), where the entry with the format: $X \rightarrow Y$, X denotes the success rate without visual input and Y represents the success rate with visual input (increasing in **green** and decreasing in **red**). Avg SR denotes the average success rate over the whole VisualWebArena benchmark.

1262
1263
1264
1265
1266
1267
1268

Category	Model	Reddit	GitLab	CMS	Map	Shopping	Avg SR
1 Agent	Claude 3.7	42.1 → 31.6	16.7 → 33.3	40.0 → 0.0	16.7 → 13.8	57.8 → 55.6	35.8 → 27.9
2 Agents	Claude 3.7	47.4 → 42.1	56.7 → 36.7	42.9 → 31.4	16.7 → 19.4	60.0 → 44.4	46.1 → 34.5
3 Agents	Qwen3-4B	10.5	3.3	14.3	13.9	33.3	17.0
	Qwen3-4B (HIMPO)	63.2	50.0	60.0	30.6	60.0	52.1
(ours)	Claude 3.7	42.1	53.3	40.0	33.3	57.8	46.1

1269 Table 7: Task success rate in VisualWebArena (Koh et al., 2024a), where the entry with the format: $X \rightarrow Y$,
 1270 X denotes the success rate without visual input and Y represents the success rate with visual input (increasing
 1271 in **green** and decreasing in **red**). Avg SR denotes the average success rate over the whole VisualWebArena
 1272 benchmark.

Category	Model	Reddit	Shopping	Avg SR
1 Agent	Claude 3.7	15.8 → 26.3	10.5 → 30.2	12.3 → 28.9
2 Agents	Claude 3.7	18.4 → 36.8	17.1 → 39.5	18.4 → 38.6
3 Agents	Qwen3-4B	5.3	11.8	9.6
(ours)	Qwen3-4B (HIMPO)	39.5	34.2	36.0
	Claude 3.7	34.2	35.5	35.1

G CASE STUDIES AND FINDINGS

In Section 1, we propose a hierarchical framework, improving planning by separation of planning and execution. On the top of it, we separate vision and action execution due to concerns around inference performance, computational cost, training stability, and data efficiency. In this section, we evaluate Claude 3.7 (Anthropic, 2025) on WebArena-Lite (Liu et al., 2025a), and present case studies that illustrate key findings regarding our proposed framework.

G.1 BALANCE BETTER HIGH-LEVEL PLANNING OBJECTIVES AND LOW-LEVEL EXECUTION WITH MULTI-AGENT

In this case study, we examine Task 1, where the agent is instructed to change the delivery address of the most recent order to "77 Massachusetts Ave, Cambridge, MA." However, in reality this modification is not allowed, as the order is likely already en route. The agent is therefore expected to recognize this constraint and eventually conclude that the address cannot be changed. We compare two settings: a single-agent setup where planning and execution are tightly coupled, and a multi-agent setup where these functions are handled by separate agents. In both configurations, the agent initially navigates to "My Account" and views the order history. However, in the single-agent setting, the agent repeatedly follows similar plan steps aligned with the task's objective, despite failing to know how to modify the address. Even after several replanning attempts, the agent maintains the incorrect assumption that the change is possible. In contrast, the multi-agent setting explicitly separates the planner and executor. The planner assigns one high-level action per round, and the executor performs that action and returns feedback. This structure encourages explicit reflection and adjustment. Despite ambiguous feedback from the executor suggesting that modification might still be possible, the planner ultimately infers that the address cannot be changed. This result demonstrates that decoupling planning and execution allows the planner to focus on high-level reasoning while the executor handles low-level operations, improving the agent's ability to align its decisions with real-world constraints.

1311
 1312 Task 1: Image-Non-Relevant Task
 1313

1314 Change the delivery address for my most recent order to 77 Massachusetts Ave, Cambridge, MA.
 1315

1316
 1317 **Response with Single Agent for both Planning and Execution**

1318 **Think:** I need to change a delivery address for a recent order. Based on the webpage, I can see that
 1319 the user (Emma Lopez) is already logged in. To change a delivery address for a recent order, I would
 1320 typically need to: 1. Go to the user's account section. 2. Find the order history or recent orders. 3.
 1321 Select the most recent order. 4. Change the delivery address. I can see a "My Account" link at the top
 1322 of the page, which is where I should start to access order information.
 1323 **Plan:**

1. Click on "My Account" link to access account information
2. Look for recent orders or order history section
3. Find the most recent order
4. Look for an option to change delivery address
5. Input the new address: 77 Massachusetts Ave, Cambridge, MA

1324 **Act:** `page.get_by_role("link", name="My Account").click()`

1325 **Think:** Looking at the current page, I'm on Emma Lopez's account page. The objective is to
 1326 change the delivery address for the most recent order to 77 Massachusetts Ave, Cambridge, MA. To
 1327 accomplish this, I need to: 1. Find the most recent order. 2. Access the order details. 3. Look for an
 1328 option to change the delivery address. Looking at the page, I can see a "Recent Orders" section with
 1329 several orders listed. The most recent order appears to be order #000000189 from 5/2/23 with a status
 1330 of "Pending". Since it's the most recent and still pending (not yet delivered), this is likely the order
 1331 we need to modify. Each order has a "View Order" link, which I should click to see the order details
 1332 and find options to change the delivery address.
 1333 **Plan:**

1. Click on the "View Order" link for the most recent order (#000000189)
2. Look for options to change the delivery address on the order details page
3. If found, enter the new address: 77 Massachusetts Ave, Cambridge, MA

1334 **Act:** `page.get_by_role("link", name="View Order", exact=True).click()`

1335 **Think:** I need to change the delivery address for the most recent order to 77 Massachusetts Ave,
 1336 Cambridge, MA. Looking at the current page, I can see this is the "My Account" page for Emma
 1337 Lopez. I need to locate the most recent order and find a way to change its delivery address. From the
 1338 "Recent Orders" section, I can see several orders with their dates: 1. Order #000000170 from 5/17/23
 1339 (Canceled) 2. Order #000000189 from 5/2/23 (Pending) 3. Order #000000188 from 5/2/23 (Pending)
 1340 4. Order #000000187 from 5/2/23 (Pending) 5. Order #000000180 from 3/11/23 (Complete). The
 1341 most recent order appears to be #000000189 from 5/2/23 which is still in "Pending" status, which
 1342 means it might still be possible to change the delivery address. I need to first click on "View Order"
 1343 for this order to see its details and find where I can change the delivery address.
 1344 **Plan:**

1. Locate the most recent order (#000000189 from 5/2/23)
2. Click on "View Order" for this order
3. On the order details page, look for an option to change the delivery address
4. Enter the new address: 77 Massachusetts Ave, Cambridge, MA

1345 **Act:** `page.get_by_role("link", name="View Order").click()`

1346 **Think:** I need to change the delivery address for the most recent order to 77 Massachusetts Ave,
 1347 Cambridge, MA. Looking at the current webpage, I'm on the "My Account" page. I can see there's
 1348 a "Recent Orders" section with several orders listed. The most recent order appears to be order
 1349 #000000189 from 5/2/23 with a status of "Pending". To change the delivery address for this order,
 1350 I need to: 1. Access the specific order details. 2. Find an option to change the delivery address. 3.
 1351 Enter the new address: 77 Massachusetts Ave, Cambridge, MA. I see that each order has a "View
 1352 Order" link, which I should click for the most recent order (#000000189)
 1353 **Plan:**

1. Click on "View Order" for the most recent order (#000000189)
2. Look for an option to change the delivery address
3. Enter the new address information 29
4. Save the changes

1354 **Act:** `page.get_by_role("link", name="View Order").click()`

1363

Response with Multi-Agent for Separating Planning and Execution

1365
1366
1367

Think from Planner: I need to click on the "My Account" link to access the account settings. This is the first step toward changing a delivery address. Looking at the observation, I can see the "My Account" link is available at the top of the page with element ID [258].

1368

1. I will click on the "My Account" link to navigate to the account settings page where I can access order information

1371

Act from Executor: page.get_by_role("link", name="My Account").click()

Feedback to Planner:

1. Step Status: COMPLETED SUCCESSFULLY
2. Goal Achievement: Successfully clicked on the "My Account" link
3. Validation Status: Action executed as requested
4. Results: Clicked on the "My Account" link to navigate to the account settings page. This should lead to a page where we can access order information and manage delivery addresses

1377
1378
1379
1380
1381
1382

should lead to a page where we can access order information and manage delivery addresses. **Think from Planner:** Looking at the page, I need to find a way to change the delivery address for the most recent order. From the current view, I can see that I'm on the "My Account" page where there's a "Recent Orders" section with several orders listed. I need to access the most recent order and then look for options to change the delivery address. Based on the order dates visible in the table, the most recent order appears to be one of the orders from 5/2/23 (orders #189, #188, or #187). I should first click on the "View Order" link for the most recent order (#189) to access its details.

1382

Plan Assigned to Executor:

1. Click on "View Order" for order #000000189 (the most recent order from 5/2/23). Once on the order details page, look for options to change the delivery address

1385

Act from Executor: `page.get_by_role("link", name="View Order", exact=True).click()`

Feedback to Planner:

1. Step Status: IN PROGRESS
2. Goal Achievement: Working toward finding a way to change the delivery address
3. Validation Status: Not yet validated
4. Results/Issues: Clicked on the "View Order" link for the most recent order (#189) dated 5/2/23. This should take us to the order details page where we can look for options to change the delivery address. The order is currently in "Pending" status which suggests it may still be possible to modify the delivery address

1393
1394
1395
1396
1397

Think from Planner: I cannot change the delivery address for this order because it has already been ordered and placed in the website. You would need to place a new order with the desired delivery address (77 Massachusetts Ave, Cambridge, MA). The page shows Emma Lopez's order details with the current shipping address as 101 S San Mateo Dr, San Mateo, California, 94010.

1397

Plan Assigned to Executor:

1. Since the order is already placed, there's no way to change the delivery address. I need to communicate that the order cannot be modified

1400

Act from Executor: page.stop("I cannot change the delivery address for this order because it has already been placed")

1402
1403
1404
1405
1406
1407
1408
1409

1410
1411
1412G.2 UNNECESSARY IMAGES HINDER PERFORMANCE IN TASKS THAT DO NOT REQUIRE VISUAL
UNDERSTANDING1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424

In Task 2, the agent is asked to calculate the total spending at One Stop Market for November 2022, which involves identifying all relevant orders and summing their totals while accounting for any applicable discounts. Crucially, this task does not require visual understanding, as all necessary information is available in the structured page content. To evaluate the impact of unnecessary visual input, we compare Claude 3.7's performance with and without image downloads enabled in the accessibility tree (as described in Section 5). The agent is expected to locate three relevant orders on page 2 of the "My Orders" section (see Figure 7), one of which—dated 11/26/22—is eligible for a 20% discount. When visual input is enabled, the agent entirely misses this discounted order and produces an incorrect total. In contrast, with images disabled, the same agent successfully identifies all relevant orders and applies the discount correctly. This case demonstrates that introducing unnecessary visual information in tasks that are inherently non-visual can distract the model and lead to degraded performance.

1425
1426
1427
1428
1429

Task 2: Image-Non-Relevant Task

How much did I spend on shopping at One Stop Market on November 2022? They gave me a 20% discount on the total amount for orders exceeding \$200 in cash.

1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453

Order #	Date	Order Total	Status	Action
000000148	1/29/23	\$440.64	Complete	View Order Reorder
000000153	1/1/23	\$132.24	Complete	View Order Reorder
000000154	12/18/22	\$97.15	Complete	View Order Reorder
000000184	12/14/22	\$20.49	Complete	View Order Reorder
000000162	12/12/22	\$53.29	Complete	View Order Reorder
000000174	12/4/22	\$32.47	Complete	View Order Reorder
000000164	11/26/22	\$218.17	Complete	View Order Reorder
000000171	11/20/22	\$133.07	Complete	View Order Reorder
000000183	11/11/22	\$51.94	Complete	View Order Reorder
000000176	10/22/22	\$845.07	Complete	View Order Reorder

Items 11 to 20 of 37 total 1 2 3 4 > Show 10 per page

1454
1455
1456

Figure 7: Webpage shown the information for answering task 2: How much did I spend on shopping at One Stop Market on November 2022? They gave me a 20% discount on the total amount for orders exceeding \$200 in cash.

1457
1458

Response with Irrelevant Image

1459
1460
1461
1462
1463
1464
1465

Think: Looking at the current webpage, I can see this is the One Stop Market homepage showing various products. The objective is to find out how much the user spent on shopping at One Stop Market in November 2022, including information about a 20% discount on orders exceeding \$200 in cash. This information is not visible on the current page. I need to check the user's account or order history to find this information. I notice there's a "My Account" link at the top of the page, which is likely where I can find order history information

Plan:

1. Click on the "My Account" link to access the account page
2. From there, I will look for order history or past purchases
3. Find orders from November 2022
4. Calculate the total spent and check if any 20% discount was applied

Act: `page.get_by_role("link", name="My Account").click()`

Think: I need to find information about how much the user spent on shopping at One Stop Market in November 2022, with a 20% discount on orders exceeding \$200. Looking at the current page, I can see we're on the "My Account" page of One Stop Market. This page shows account information and recent orders. I need to check if there's any order history from November 2022 visible on this page. I can see a "Recent Orders" section with a table showing order numbers, dates, order totals, and statuses. However, the dates shown are:- 5/17/23- 5/2/23 (multiple orders)- 3/11/23. None of these orders are from November 2022. I need to look for a way to view older orders that might include November 2022. I see there's a "View All" link next to "Recent Orders" and also a "My Orders" link in the sidebar. Clicking either of these should take me to the complete order history where I might find November 2022 orders.

Plan:

1. Since I need to find orders from November 2022 and the current page only shows more recent orders from 2023, I need to view the complete order history.
2. I'll click on the "View All" link next to "Recent Orders" to see the complete order history.
3. Once there, I'll look for any orders from November 2022 and check their totals to see if any exceeded \$200 and received a 20% discount.

Act: `page.get_by_role("link", name="View All", exact=True).click()`

Think: I need to find information about purchases made in November 2022 to determine how much was spent at One Stop Market during that time. Looking at the current page, I can see this is the "My Orders" page showing order history with dates, order numbers, and amounts. I need to check if any of these orders are from November 2022. The current page shows orders from: - 5/17/23 - 5/2/23 (multiple orders) - 3/11/23 - 3/10/23 - 2/27/23 - 2/24/23 - 2/11/23 - 2/9/23 None of these are from November 2022. I need to look at more orders by navigating to other pages since the current page shows "Items 1 to 10 of 37 total" and there are pagination links available. I should navigate to the next page to see if there are orders from November 2022.

Plan:

1. The current page shows orders from 2023, but none from November 2022.
2. I need to check more orders by navigating to the next page.
3. Click on the "Next Page" link to see more orders.

Act: `page.get_by_role("link", name="Page Next").click()`

Think: I need to find out how much was spent at One Stop Market in November 2022. Looking at the current page, I can see this is the "My Orders" page showing order history. I need to identify all orders from November 2022 and sum their totals. Looking at the order table on the current page (page 2), I can see these November 2022 orders: 1. Order #000000171 from 11/20/22: \$133.07 2. Order #000000183 from 11/11/22: \$51.94 So for November 2022, based on the orders visible on this page, the total is \$133.07 + \$51.94 = \$185.01. However, I should check if there are any other November 2022 orders on other pages. I'm currently on page 2, and there are 4 pages total. I should look at the other pages to ensure I have all November 2022 orders before calculating the final amount

Plan:

1. Identify all November 2022 orders on the current page (already found two: \$133.07 + \$51.94)
2. Check page 1 for additional November 2022 orders by clicking "Page 1"
3. After reviewing all pages, calculate the total amount spent in November 2022.
4. Determine if any orders exceeded \$200 (for the 20% discount mentioned).

Act: `page.get_by_role("link", name="Page 1").click()`

1504

Response with Irrelevant Image (continue)

1506
1507
1508
1509
1510
1511
1512
1513
1514

Think: I need to find out how much the user spent at One Stop Market in November 2022. Looking at the current webpage, I can see wefe on the "My Orders" page which shows order history. I need to check if there are any orders from November 2022 visible on the current page. Looking at the order dates displayed: - 5/17/23 - 5/2/23 - 5/2/23 - 5/2/23 - 3/11/23 - 3/10/23 - 2/27/23 - 2/24/23 - 2/11/23 - 2/9/23 I don't see any orders from November 2022 on the current page. The earliest orders shown are from February 2023, and we need to find orders from November 2022. I need to navigate to earlier pages in the order history. I notice there are pagination links at the bottom: Page 2, Page 3, Page 4, and "Page Next". Since we need to go back to November 2022, and the current page shows orders from February 2023 and later, I should navigate to earlier pages. Let's try going to Page 2 first.

1514

1. The current page shows orders from February 2023 to May 2023
2. I need to find orders from November 2022
3. I should navigate to Page 2 of the order history to see earlier order.
4. If Page 2 doesn't show November 2022 orders, I'll continue to Page 3 and Page 4 as needed

1519

Think: I need to find out how much was spent at One Stop Market in Nov 2011.

1521
1522
1523
1524
1525
1526
1527

THUR: I need to find out how much was spent at One Stop Market in November 2022. Looking at the current page, I can see we're on page 2 of the order history. I can see orders listed with dates and amounts. Looking at the orders from November 2022 specifically: - Order #000000171 from 11/20/22: \$133.07 - Order #000000183 from 11/11/22: \$51.94 So the total spent in November 2022 based on what I can see is $133.07 + 51.94 = \$185.01$. However, I should check if there are any more November 2022 orders on other pages to make sure I have the complete information. I'm currently on page 2, and there are pages 3 and 4 available. Let me check those pages for any additional November 2022 orders.

1527

— 100 —

1. I've identified two November 2022 orders on the current page. - 11/20/22. \$155.07 - 11/11/22. \$51.94
2. Need to check if there are more November 2022 orders on page 3
3. Will click on "Page 3" to see if there are additional November orders.

1532

Act: page.get_by_role("link", name="Page 3").click()

1533
1534
1535
1536
1537
1538
1539
1540

Think: I need to find out how much the user spent on shopping at One Stop Market in November 2022. I'm currently on the "My Orders" page showing orders from page 3, which displays orders from July to October 2022 based on the dates shown. Looking at the order table, I can see these orders: - 10/21/22: \$345.84 (Complete) - 10/18/22: \$2,126.32 (Canceled) - 10/3/22: \$18.99 (Complete) - 9/29/22: \$2,890.53 (Complete) - And other earlier orders I don't see any November 2022 orders on this page. I need to check if there are November 2022 orders on other pages. Since I'm currently on page 3, I should check page 1 and 2 to see if there are more recent orders (which would include November 2022). First, let me check page 1 which would have the most recent orders.

1540

1. Navigate to page 1 of the order history to find the most recent orders
2. Look for orders from November 2022
3. Calculate the total spent in November 2022
4. Determine if any orders exceeded \$200 to qualify for the 20% discount
5. Calculate the final amount after any applicable discounts

1545
1546
1547
1548
1549
1550

Act: `page.get_by_role("link", name="Page 1").click()`

1546
1547
1548
1549
1550

Think: I need to find out how much the user spent on shopping at One Stop Market in November 2022. I'm currently on the "My Orders" page which shows order history. Looking at the current page, I can see a table of orders with dates, but I need to find orders from November 2022 specifically. The current page shows orders from: - 5/17/23 (May 2023) - 5/2/23 (May 2023) - 3/11/23 (March 2023) - 3/10/23 (March 2023) - 2/27/23 (February 2023) - 2/24/23 (February 2023) - 2/11/23 (February 2023) - 2/9/23 (February 2023) None of these are from November 2022. I need to navigate to other pages to find orders from November 2022. The current page shows "Items 1 to 10 of 37 total" and there are pagination links to pages 2, 3, and 4. I should check the other pages to find orders from November 2022. Let's start by checking page 2.

1551
1552

Response with Irrelevant Image (continue)

1553

Plan:

1554

1. Since the current page (page 1) doesn't show any orders from November 2022, I need to navigate to other pages.
2. Click on "Page 2" to check if there are any November 2022 orders there.
3. If not found on page 2, continue checking pages 3 and 4 until I find November 2022 orders.
4. Once I find orders from November 2022, calculate the total amount spent.

1555

Act: `page.get_by_role("link", name="Page 2").click()`

1556

Think: I need to find out how much the user spent on shopping at One Stop Market in November 2022. Looking at the current webpage, I can see we're on the "My Orders" page showing order history. I need to examine the orders from November 2022 and calculate the total amount. Looking at the table of orders, I can see several orders with dates. The ones from November 2022 are: - Order #000000171 from 11/20/22: \$133.07 - Order #000000183 from 11/11/22: \$51.94 So the total amount spent in November 2022 is $\$133.07 + \$51.94 = \$185.01$ The objective also mentions a 20% discount on orders exceeding \$200, but since the total is \$185.01, which is less than \$200, no discount would have been applied.

1557

Plan:

1558

1. Calculate the total spent in November 2022 from the visible orders
2. Return the answer using `page.stop()`

1559

Act: `page.stop("In November 2022, you spent a total of $185.01 at One Stop Market ($133.07 on November 20 and $51.94 on November 11). Since your total spending for the month was less than $200, you wouldn't have received the 20% discount that applies to orders exceeding $200.")`

1560

1561

1562

1563

1564

1565

1566

1567

1568

1569

1570

1571

1572

1573

1574

1575

1576

1577

1578

1579

1580

1581

1582

1583

1584

1585

1586

1587

1588

1589

1590

1591

1592

1593

1594

1595

1596

1597

1598

Response without Image

1599

1600

Think: I need to find out how much the user spent on shopping at One Stop Market in November 2022, with a 20% discount on orders exceeding \$200. Looking at the current page, I can see this is the "My Account" page showing recent orders. I need to examine if any of these orders are from November 2022. The recent orders table shows: - Order #000000190 from 8/2/25 - Order #000000170 from 5/17/23 - Order #000000189 from 5/2/23 - Order #000000188 from 5/2/23 - Order #000000187 from 5/2/23 I don't see any orders from November 2022 in this list. I need to check if there's more order history available. I see there's a "View All" link next to "Recent Orders" which would likely show the complete order history. I should click this link to see if there are any orders from November 2022.

1601

1602

1603

1604

1605

1606

1607

1608

1609

Plan:

1. Click on "View All" next to the Recent Orders section to see the complete order history
2. Look for orders from November 2022
3. Calculate the total spending for November 2022 orders
4. Apply the 20% discount rule if applicable
5. Provide the answer

1610

1611

1612

1613

Act: `page.get_by_role("link", name="View All", exact=True).click()`

Think: I need to find out how much was spent on shopping at One Stop Market in November 2022, with a 20% discount on orders exceeding \$200 in cash. Looking at the current page, I can see a table of orders with dates, order numbers, and totals. I need to: 1. Check if there are any orders from November 2022 in the current view. 2. If not, I may need to navigate to other pages to find November 2022 orders. Looking at the dates in the current table: - 8/2/25 - 5/17/23 - 5/2/23 - 5/2/23 - 5/2/23 - 3/11/23 - 3/10/23 - 2/27/23 - 2/24/23 - 2/11/23 I don't see any orders from November 2022 in the current view. There are 38 total items according to the pagination info, and we're only seeing the first 10. I need to navigate to other pages to find orders from November 2022.

1621

1622

1623

1624

1625

1626

1627

1628

1629

1630

1631

1632

1633

1634

1635

1636

1637

1638

1639

1640

1641

1642

1643

1644

1645
1646

Response without Image (continue)

1647

Plan:

1648

1. I need to check other pages of order history to find orders from November 2022.
2. The current page shows orders from 2023-2025, so I should check page 2 next.

1649

Act: `page.get_by_role("link", name="Page 2").click()`

1650

Think: I need to calculate how much was spent on shopping at One Stop Market in November 2022. Looking at the current page, I can see a table of orders with dates, order numbers, and order totals. I need to find all orders from November 2022 and add up their totals. From the current observation, I can see these November 2022 orders: - Order #000000171 dated 11/20/22: \$133.07 - Order #000000183 dated 11/11/22: \$51.94 So the total spending for November 2022 from this page is \$133.07 + \$51.94 = \$185.01. I should check if there are more November 2022 orders on other pages. I am currently on page 2, so I should check page 1 as well to see if there are any other November orders.

1651

Plan:

1652

1. I have identified two November 2022 orders on the current page (page 2): - 11/20/22: \$133.07 - 11/11/22: \$51.94
2. I need to check page 1 to see if there are any additional November 2022 order
3. After collecting all November 2022 orders, I'll calculate the total and check if any orders exceeded \$200 to qualify for the 20% discount

1653

Act: `page.get_by_role("link", name="Page 1").click()`

1654

Think: I need to find out how much the user spent on shopping at One Stop Market in November 2022, with a 20% discount on orders exceeding \$200 in cash. First, I need to analyze the current page. I see this is the "My Orders" page showing order history. The current page shows orders from: - 8/2/23 - 5/17/23 - 5/2/23 (multiple orders) - 3/11/23 - 3/10/23 - 2/27/23 - 2/24/23 - 2/11/23 I don't see any orders from November 2022 on this page. I need to check other pages of the order history. I see pagination at the bottom showing "Page" with links to pages 2, 3, and 4. Since I already tried clicking "Page 1" in the previous action (which was unsuccessful since we're already on page 1), I should try navigating to the next pages to find orders from November 2022

1655

Plan:

1656

1. Navigate through the order history pages to find orders from November 2022
2. Start by clicking "Page 2" to see the next set of orders
3. If not found, continue to other pages until I find November 2022 orders
4. Once found, calculate the total spending and apply the 20% discount rule

1657

Act: `page.get_by_role("link", name="Page 2").click()`

1658

Think: I need to determine how much was spent at One Stop Market in November 2022. Looking at the current page, I can see a table of orders with dates and order totals. Let me identify all orders from November 2022 and sum them up. From the orders table, I can see these November 2022 orders: 1. Order #000000164 from 11/26/22 with amount \$218.17. 2. Order #000000171 from 11/20/22 with amount \$133.07. 3. Order #000000183 from 11/11/22 with amount \$51.94. So for November 2022, the total spending was: \$218.17 + \$133.07 + \$51.94 = \$403.18. Now regarding the 20% discount on orders exceeding \$200 in cash: - Only order #000000164 (\$218.17) exceeds \$200 - If this order was paid in cash, it would qualify for the 20% discount - The discount would be \$218.17 * 0.2 = \$43.63 - The final amount after discount would be \$218.17 - \$43.63 = \$174.54 - So the total November spending with discount would be \$174.54 + \$133.07 + \$51.94 = \$359.55

1659

1. Calculate the total spending for November 2022 from the visible orders
2. Provide the answer through `page.stop()`

1660

Act: `page.stop("In November 2022, the total spending at One Stop Market was $403.18 (sum of three orders: $218.17 on 11/26/22, $133.07 on 11/20/22, and $51.94 on 11/11/22). If the $218.17 order was paid in cash, it would qualify for the 20% discount, reducing the November total to $359.55")`