From Local Skills to Long-Horizon Web Tasks: Progressive Skill Exploration for LLM Web Agents

Xueyang Feng[1†] **Xiaohe Bo**[1*] **Xu Chen**[1‡] **Quanyu Dai**[2‡] **Feng Wen**[2]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Huawei Technologies Ltd.
xueyangfeng@ruc.edu.cn xu.chen@ruc.edu.cn daiquanyu@huawei.com

Abstract

Large language models (LLMs) are driving autonomous agents toward real-world environments, with the Web emerging as one of the most representative interaction scenarios. However, training Web agents with naive exploration remains challenging: successful trajectories are rare, and the sparse terminal rewards make it difficult for agents to obtain useful feedback from failed trajectories. Inspired by skill acquisition theory, we propose **S**kill-**Co**mposed **P**rogressive **E**xploration (SCOPE), a short-to-long learning framework for long-horizon Web agents that decomposes complex task exploration into a gradual expansion process from local skills to complete task-solving capability. Specifically, SCOPE iteratively explores around local skills: *Skill Discovery* extracts reusable local skills and failure experiences from historical rollouts; *Skill Orchestration* composes discovered skills into high-level skill chains; and *Skill Expansion* leverages failure experiences aligned with the current skill chain to guide the agent toward exploring new sub-skills. Through this process, SCOPE progressively constructs task-completing skill chains across rollouts and internalizes them into the policy parameters through agentic on-policy distillation. Extensive experiments on WebArena, WebVoyager, and WebShop show that SCOPE consistently improves exploration dynamics and achieves 3.9%–18.6% relative gains over the strongest baselines in each setting.

1 Introduction

Large language models (LLMs) (Qwen et al., 2025; Grattafiori et al., 2024) have demonstrated remarkable capabilities in language understanding, commonsense reasoning, and complex planning (Xu et al., 2025; Feng et al., 2024), laying an important foundation for autonomous agents (Wang et al., 2024). In this context, Web agents that interact with real-world websites (Zhou et al., 2024; He et al., 2024; Yao et al., preprint) have become a highly promising research direction, as they hold the potential to bridge human natural language instructions and the operational workflows of traditional computer systems. Unlike conventional single-step text generation tasks, LLM-based Web agents must continuously perceive complex and dynamically changing Web environments and make long-horizon decisions throughout interaction, which poses substantially greater technical challenges.

To improve Web agents on long-horizon tasks, recent studies have begun to explore agentic reinforcement learning paradigms (Qi et al., 2025; Wei et al., 2025; Feng et al., 2025). The central idea is to enable agents to interact directly with online environments, collect trajectories, and optimize their policy parameters using task rewards. However, the naive exploration strategies used in existing agent training are often ineffective: (1) **Sparse Successes**: long-horizon interactions and dynamic

*Equal contribution.
†Work done during an internship at Huawei.
‡Co-corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

webpage states make it difficult for agents to obtain successful trajectories through naive end-to-end exploration; (2) **Unexploited Failures**: rewards in Web environments are typically sparse, so failed trajectories are often discarded entirely. As a result, when successful trajectories are difficult to obtain, failed attempts provide little guidance for later exploration, often leading to repetitive trial-and-error failures. This raises a key question: *how can we support effective and sustained exploration for long-horizon tasks when successful trajectories are difficult to obtain through naive exploration?*

Fitts and Posner's skill acquisition theory (Fitts and Posner, 1967) provides an instructive perspective for addressing these exploration challenges: complex skills are not acquired through a single end-to-end attempt, but are progressively composed from simpler skills. This perspective naturally fits long-horizon Web tasks, which typically exhibit an inherent compositional structure. Specifically, such tasks can often be decomposed into a sequence of subtasks that are locally verifiable through environmental feedback. For example, when a human first visits a shopping website, she may find it difficult to find orders with a specific status in a single attempt. However, even if the overall task fails, environmental feedback can still reveal useful sub-skills and informative failure experiences in the trajectory, such as navigating to the order page or entering irrelevant pages. In the next attempt, the former can serve as a basis for more complex skills like order sorting, while the latter can help the agent avoid repeating ineffective actions. In this way, learning a long-horizon task is decomposed into a progressive expansion process from local skills to complete task-solving capability.

Motivated by this, we propose **S**kill-**Co**mposed **P**rogressive **E**xploration (SCOPE), a short-to-long progressive learning framework for long-horizon Web agents. The core of SCOPE is an iterative exploration process organized around skills: (1) *Skill Discovery*, extracting local skills and failure experiences from historical rollout trajectories; (2) *Skill Orchestration*, recomposing existing skills into high-level skill chains to recover intermediate progress; and (3) *Skill Expansion*, leveraging failure experiences aligned with the current skill chain to guide the agent toward exploring new subskills. Through this iterative process, SCOPE can progressively obtain successful trajectories by accumulating task-completing skills across different rollouts. Finally, in *Skill Distillation*, we internalize the resulting task-completing skill chains into the policy parameters through agentic on-policy distillation. We evaluate SCOPE in three real and simulated Web environments: WebArena (Zhou et al., 2024), WebVoyager (He et al., 2024), and WebShop (Yao et al., preprint). Across these benchmarks, SCOPE improves the main task metric by 3.9%–18.6% over the strongest corresponding baselines.

Our main contributions are as follows:

- We propose a short-to-long perspective for long-horizon Web-agent learning, decomposing Web-task exploration into a progressive process from local skills to complete task-solving capability.
- We propose SCOPE, which enables agents to discover, organize, and expand local skills across rollouts, progressively transforming partial progress into successful trajectories. These trajectories are then internalized into the policy parameters through agentic on-policy distillation.
- We evaluate SCOPE on three real and simulated Web environments, where it consistently outperforms existing Web-agent training methods across different LLM backbones. Beyond the overall performance gains, we further analyze the source of these improvements and study how the effectiveness of two on-policy intervention strategies evolves with model capability.

2 Background

For a detailed survey of related work, please refer to Appendix B. In this section, we formalize the Web-task setting and the corresponding Web-agent training problem that motivates our method.

Web-Agent Task Formulation. We mainly focus on Web agents that can interact with real-world websites (Zhou et al., 2024; He et al., 2024). They access websites through a standard browser and perform browser actions based on text-based webpage observations, typically represented as HTML or AXTree. Formally, a Web task can be modeled as a long-horizon interactive decision-making problem. Given a user task G, the agent interacts with the Web environment over a horizon of T steps. At each step $t \in \{0, \ldots, T-1\}$, the agent receives a webpage observation o_t and forms a state s_t together with the interaction history. The agent's policy model π_θ then generates an action $a_t \sim \pi_\theta(\cdot \mid G, s_t)$. The interaction produces a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T)$, which receives a sparse terminal reward $R(\tau) \in \{0, 1\}$ indicating whether the task is completed.

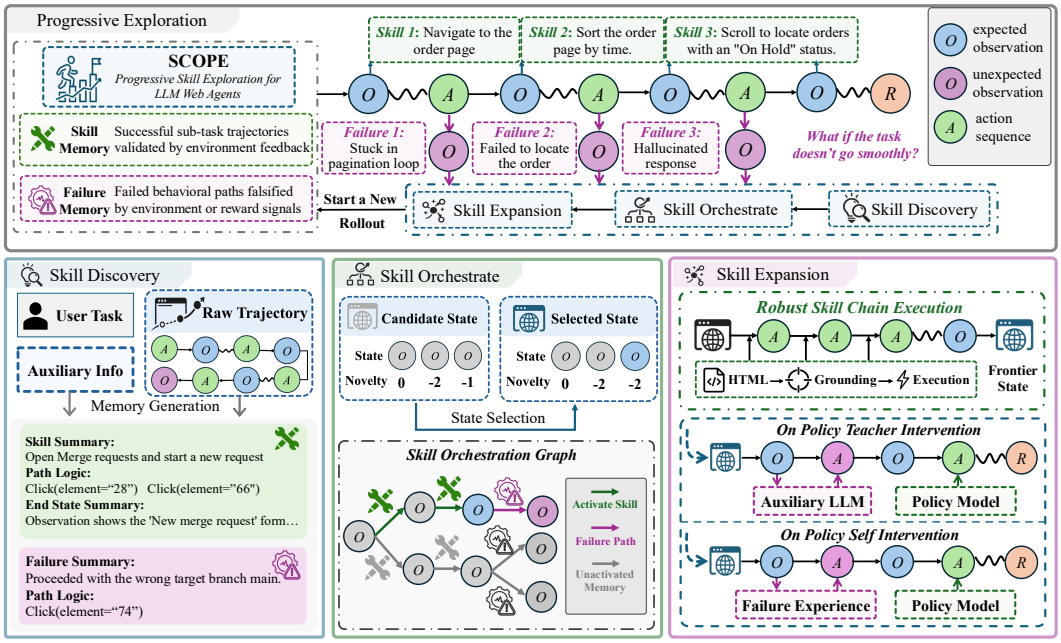


Figure 1: Overview of SCOPE's progressive exploration framework: **Skill Discovery** extracts local skills and failure experiences from historical trajectories; **Skill Orchestration** composes local skills into skill chains; and **Skill Expansion** uses chain-aligned failure experiences to guide exploration.

Web-Agent Training. To improve Web agents' ability to complete long-horizon tasks, recent work has begun training policy models by letting agents interact directly with the environment under the policy π_θ (Qi et al., 2025; He et al., 2025; Zhou et al., 2025; Yang et al., 2025). Existing methods typically adopt a naive end-to-end exploration strategy: they sample a user query G from a training task distribution $\mathcal{D}$, roll out the agent from the initial state to collect a complete trajectory, and rely on terminal rewards $R(\tau)$ to optimize the standard objective: $\max_\theta \ \mathbb{E}_{G\sim\mathcal{D},\,\tau\sim\pi_\theta(\cdot|G)}[R(\tau)]$.

However, in long-horizon Web tasks, such exploration strategies often struggle to directly obtain complete successful trajectories. Since rewards are provided only at the terminal step, if a trajectory fails to complete the final task, it is often discarded as a whole even if some of its segments contain meaningful local progress. This motivates our key observation that intermediate webpage observations o_t record environmental changes, such as state transitions, interface changes, and error messages, and therefore provide informative feedback beyond this sparse supervision. By leveraging such feedback, we can view failed trajectories as sources of local skills and failure experiences, and decompose long-horizon task solving into subtasks that can be progressively discovered, reused, and expanded.

3 Methodology

Inspired by Fitts and Posner's skill acquisition theory (Fitts and Posner, 1967), we propose SCOPE, a short-to-long learning framework for long-horizon Web agents. As illustrated in Figure 1, the core of SCOPE is a skill-centric progressive exploration process: *Skill Discovery* identifies reusable local skills and failure experiences from historical trajectories (Section 3.1); *Skill Orchestration* composes validated skills into skill chains (Section 3.2); and *Skill Expansion* uses chain-aligned failures to guide exploration beyond the current frontier (Section 3.3). Finally, the resulting task-completing skill chains are internalized into the policy through agentic on-policy distillation (Section 3.4).

3.1 Skill Discovery

Skill acquisition theory suggests that, in the early stage of learning, learners facing complex long-horizon tasks often rely on explicit task decomposition to gradually build an understanding of the task structure. Following this view, we identify local skills and failure experiences from rollout trajectories. Specifically, given a rollout trajectory τ, we use the task goal G, the task-completion

criterion c, and the intermediate environmental feedback contained in τ to parse it into semantically contiguous subtask units and assess whether each unit achieves its intended subgoal:

$$(\xi_i, y_i)_{i=1}^{n} = \textsc{Discover}(G, c, \tau), \quad \xi_i = (s^{(i)}_{\texttt{start}}, A^{(i)}, s^{(i)}_{\texttt{end}}; m_i),\ y_i \in \{0, 1\}.$$

Here, DISCOVER is implemented by an auxiliary LLM with a task-specific prompt, as are the symbolic operators introduced below. The criterion c typically includes goal-related constraints or target page states that help assess whether a subgoal has been achieved. In each segment, `start` and `end` refer to its boundaries in the original trajectory, and $s^{(i)}_{\texttt{start}}$ and $s^{(i)}_{\texttt{end}}$ denote the corresponding starting and ending states. $A^{(i)} = (a^{(i)}_{\texttt{start}}, \ldots, a^{(i)}_{\texttt{end}-1})$ denotes the action subsequence that transitions between them, m_i denotes the semantic subtask description, and y_i indicates whether the intended subgoal is achieved. When $y_i = 1$, the corresponding segment represents a local action sequence in the original trajectory that successfully completes a specific Web subtask; to enable reuse in subsequent rollouts, we maintain it as a skill. When $y_i = 0$, the corresponding segment represents a behavioral sequence that attempts but fails to complete a specific subtask, and we store it as failure experience. In implementation, we store non-redundant entries in two memory banks:

$$\mathcal{B}_{\text{skill}} \leftarrow \mathcal{B}_{\text{skill}} \cup \{\xi_i \mid y_i = 1\}, \quad \mathcal{B}_{\text{fail}} \leftarrow \mathcal{B}_{\text{fail}} \cup \{\xi_i \mid y_i = 0\}.$$

These two memory banks capture complementary intermediate feedback. The skill memory $\mathcal{B}_{\text{skill}}$ stores subtask-completion sequences verified by environmental feedback, each capturing a reusable transition between environment states. Notably, such progress does not necessarily lead to final task success. The failure memory $\mathcal{B}_{\text{fail}}$ records failure patterns triggered along particular progress paths. In the following sections, we study how to use these memories to support subsequent exploration.

3.2 Skill Orchestration

In the middle stage of skill acquisition, a learner composes previously discovered local rules into higher-level complex skills. Inspired by this view, we compose validated skills in the skill bank $\mathcal{B}_{\text{skill}}$ into reusable skill chains for subsequent attempts. Concretely, the skill bank can be viewed as a state graph connected by local skills, where each node s_k denotes a specific environment state, such as an order page or a sorting page, and each edge corresponds to a validated local skill. We collect candidate reachable nodes in this graph as the frontier state set $\mathcal{S}_{\text{frontier}} = \{s_k\}_{k=1}^{K}$, and perform group-level evaluation to select the frontier state $s^\star = \textsc{Select}(G, c, \mathcal{S}_{\text{frontier}})$ for the next exploration rollout. After selecting $s^\star$, we trace and compose a skill chain $p^\star$ of connected subskills from the initial state s_0 to $s^\star$. In the next rollout, this skill chain is invoked to restore the environment to the frontier state $s^\star$, which serves as the basis for further exploration.

In this process, we use two implementation details. *Exploration-aware frontier selection*: we maintain a visitation count for each frontier state and prompt the LLM to downweight frequently visited states, balancing progress reuse and exploration diversity (Ecoffet et al., 2021; Lu et al., 2025). *Robust skill execution*: Original agent trajectories often refer to target elements by indices or positions in AXTree/HTML. To make skills reusable across dynamic webpages, we extract element names from the original identifiers and ground them to the current AXTree/HTML context before execution.

3.3 Skill Expansion

Once skills have been organized into a chain, failure experiences aligned with it can guide exploration beyond the current skill graph. Specifically, we identify failure experiences in $\mathcal{B}_{\text{fail}}$ whose starting state matches the selected frontier state $s^\star$: $\mathcal{F}^\star = \{\, \xi_i \in \mathcal{B}_{\text{fail}} \mid s^{(i)}_{\texttt{start}} = s^\star \,\}$. The key question is then how to effectively inject failure experiences into subsequent exploration. To this end, we study two on-policy intervention strategies applied at the first frontier action:

- *On-policy Teacher Intervention* (*OPTI*). Inspired by on-policy distillation and imitation learning (Agarwal et al., 2024; Ross et al., 2011), we introduce a teacher model to directly generate an intervening action from $s^\star$ and $\mathcal{F}^\star$: $a^{\text{teach}} = \textsc{Intervene}(G, s^\star, \mathcal{F}^\star)$. This action replaces the policy model's original action at the frontier state, providing stronger external guidance.
- *On-policy Self Intervention* (*OPSI*). Inspired by on-policy self-distillation (Zhao et al., 2026), the policy conditions on retrieved failure experiences to generate the intervening action: $a^{\text{self}} \sim \pi_\theta(\cdot \mid G, s^\star, \textsc{Summarize}(G, p^\star, \mathcal{F}^\star))$. This variant lets the policy use failure experiences to correct its own behavior, thereby generating intervening actions closer to the policy model itself.

Importantly, these interventions remain on-policy at the level of agent-environment interaction steps: the frontier state $s^\star$ is reached by skill-chain-based state restoration, which composes locally verified skills produced by the policy model in historical rollouts. After the intervention, the policy model continues subsequent exploration autonomously from the updated state. We further analyze the effects and differences between these two intervention strategies in Section 4.4.

3.4 Skill Distillation

At the final stage of skill acquisition, learners internalize acquired capabilities into stable and efficient task execution. Accordingly, we adopt an agentic on-policy distillation strategy to internalize the high-quality behaviors induced by progressive exploration and memory-guided intervention into the model parameters. Let $q_{\mathrm{SCOPE}}(\tilde{\tau} \mid G)$ denote the teacher trajectory distribution produced by SCOPE's progressive exploration. Each $\tilde{\tau}$ is a task-completing teacher trajectory with $R(\tilde{\tau}) = 1$, composed from skill chains accumulated across multiple rollouts. We distill this teacher trajectory distribution constructed by progressive exploration into the policy model π_θ by minimizing

$$\mathcal{L}_{\mathrm{distill}}(\theta) = \mathbb{E}_{G\sim\mathcal{D},\, \tilde{\tau}\sim q_{\mathrm{SCOPE}}(\cdot|G)}[-\sum_{t=0}^{|\tilde{\tau}|-1} \log \pi_\theta(\tilde{a}_t \mid G, \tilde{s}_t)].$$

Here, $(\tilde{s}_t, \tilde{a}_t)$ are the state-action pairs actually executed in the teacher trajectory constructed by progressive exploration. The teacher trajectory mixes actions from two sources: non-intervention actions sampled from the policy itself, and frontier intervention actions generated with failure-experience guidance or auxiliary-model assistance. Taking *OPSI* as an example, frontier actions are produced from the failure-conditioned distribution $\pi_\theta(a \mid G, \tilde{s}_t, \mathcal{F}^\star)$, while distillation compresses them into the standard policy distribution $\pi_\theta(a \mid G, \tilde{s}_t)$. This process internalizes task-completing skill chains that rely on progressive exploration and failure correction into the student policy. Moreover, beyond the exploration perspective, the overall advantages of SCOPE can also be understood from the perspective of on-policy distillation. We provide a detailed discussion in Appendix C.

3.5 Overall Progressive Learning Process

Overall, SCOPE is a skill-organized framework that interleaves progressive exploration with skill distillation. From the current policy $\pi_\theta^{(r)}$, each training round performs an initial rollout and then runs iterative exploration, where *Skill Discovery*, *Skill Orchestration*, and *Skill Expansion* extend the frontier from local skills toward task-completing skill chains. *Skill Distillation* internalizes the resulting chains into the policy, producing $\pi_\theta^{(r+1)}$ for the next round:

$$\pi_\theta^{(r)} \to \tau_0^{(r)} \to \underbrace{\textsc{Discovery} \to \textsc{Orchestration} \to \textsc{Expansion}}_{\text{Iteration } j} \to \tau_j^{(r)} \to \cdots \to \tau_J^{(r)} \xrightarrow{\textsc{Distill.}} \pi_\theta^{(r+1)}.$$

Here, r indexes the outer training round, and $j = 1, \ldots, J$ indexes its progressive exploration iterations before skill distillation. The updated policy then enters the next round, allowing subsequent exploration to benefit from accumulated memories and capabilities internalized into the model parameters. We provide the complete algorithm in Appendix A to clarify SCOPE's training process.

We analyze SCOPE's progressive exploration mechanism from a variance perspective, showing that it can reduce the instability of rollouts required for long-horizon exploration from exponential to polynomial scale, thereby enabling stable and sustained exploration on long-horizon tasks.

Proposition 1 (Polynomial Variability of Progressive Exploration)**.** *Assume that making a locally correct frontier-expanding decision has probability at least $p \in (0, 1]$, and that, at any fixed depth, the number of webpage functional patterns relevant to the current subtask progression is bounded. Then the standard deviation of exploration rollouts T_{total} satisfies $\mathrm{Std}(T_{\mathrm{total}}) = O\Big(\frac{L^{(1+2\alpha)/2}}{p}\Big)$.*

Here L denotes the target task depth, and α is a task-dependent constant representing the key constraint factors that must be progressively satisfied to complete the Web task. By contrast, under naive direct exploration, completing a task of depth L requires making all L critical decisions correctly, so the success probability decays as p^L. This causes both the waiting time and its variability to grow exponentially with task depth. The full derivation is deferred to Appendix D.

4 Experiments

To comprehensively validate the effectiveness of our proposals, we conduct extensive experiments, which are outlined as follows:

- First, we use SCOPE to train two backbone models on both offline and online Web benchmarks, and compare them with a range of baselines to validate the benefit of progressive exploration for improving model capability and its broad applicability (Section 4.2).
- Second, we analyze SCOPE's progressive exploration dynamics across task depths, tracing how improved exploration translates into stronger long-horizon capability (Section 4.3).
- Third, we compare the two intervention strategies, OPTI and OPSI, and analyze how their exploration and internalization behaviors vary with model capability (Section 4.4).
- Finally, we further study SCOPE's scalability, self-managed memory, component necessity, and training efficiency (Section 4.5).

4.1 Experiments Setup

Benchmarks and Baselines. We evaluate SCOPE on three representative Web-agent benchmarks with diverse interaction settings: (1) **WEBARENA-LITE** (Liu et al., 2024) is an offline hosted Web environment with long-horizon tasks across Reddit, Shopping, CMS, and GitLab, involving complex information acquisition and page-state manipulation. We follow the experimental setup of WEBRL (Qi et al., 2025) and use its standard leaderboard results as the main comparison baseline; (2) **WEBVOYAGER** (He et al., 2024) evaluates agents through interactions with real online websites, such as GitHub and ArXiv, where the interaction process is more dynamic and non-stationary. We use the tasks provided by OPENWEBVOYAGER (He et al., 2025) as the training set and the original WEBVOYAGER tasks as the test set. Since online Web environments are generally difficult to accommodate purely online reinforcement learning methods (Qi et al., 2025), we mainly compare with SFT and RFT; (3) **WEBSHOP** (Yao et al., preprint) is a benchmark widely used for evaluating reinforcement-learning algorithms for Web agents. Since it supports repeatable sampling and has lower interaction cost, we further compare with online RL baselines such as GRPO (Shao et al., 2024). More details about the benchmarks and baselines are provided in Appendix F.

Training Details. We use Qwen2.5-3B (Qwen et al., 2025) and Llama3.1-8B (Grattafiori et al., 2024) as policy backbones throughout our experiments. Before SCOPE training, each policy undergoes a benchmark-appropriate warm-up stage; details are provided in Appendix F. For memory management, we consider both external-model and self-managed settings. In the external-model setting, the auxiliary functions DISCOVER, SELECT, SUMMARIZE, and INTERVENE are implemented with DeepSeek-V3.2 (DeepSeek-AI, 2025). In the self-managed setting, these functions are performed by a Llama3.1-8B model fine-tuned on pre-collected memory-management data. Prompts for all auxiliary-model calls are provided in Appendix F. For progressive exploration, we run 5 inner exploration iterations per outer training round on WEBARENA and WEBVOYAGER, and 4 iterations on WEBSHOP. Training is conducted with the LLaMA-Factory framework (Zheng et al., 2024). The policy learning rate is selected from $\{10^{-5}, 10^{-6}\}$, the batch size is 32, and the number of training epochs is 3. Unless otherwise specified, we report OPSI results after one outer training round; results with three outer training rounds are analyzed in Section 4.5.

Test Details. For environment interaction, real benchmarks are executed in browser environments. WEBVOYAGER provides AXTree-based observations, while WEBARENA provides HTML-based observations. We use vLLM (Kwon et al., 2023) for policy inference. To ensure reproducibility, we set the decoding temperature to 0 during evaluation. The maximum number of interaction steps is 10 for WEBSHOP, 15 for WEBVOYAGER, and 30 for WEBARENA. Because some website tasks are unreachable, we filter out a subset of websites in WEBARENA and WEBVOYAGER; details are provided in Appendix F. All prompts are kept consistent with the corresponding baselines.

4.2 Main Results

SCOPE achieves stronger overall performance than existing baselines. Table 1, Table 2, and Table 3 summarize the main comparisons. Overall, general models without task-specific training perform substantially worse on complex interactive Web tasks than fine-tuned models, highlighting

Table 1: Results of SCOPE on WEBARENA, an offline yet realistic Web benchmark. OPTI and OPSI denote on-policy Teacher Intervention and on-policy Self Intervention, respectively.

Model	Method	Total SR	Reddit	GitLab	Shopping	CMS
		Prompting Method				
Qwen2.5-3B	-	7.0	5.3	13.3	4.4	5.7
Llama3.1-8B	-	7.8	5.3	10.0	8.9	5.7
OAI-4o	-	13.2	10.5	10.0	11.1	20.0
		Finetuning Method				
Qwen2.5-3B	Behavior Cloning	20.2	42.1	16.7	11.1	22.9
	RFT (Yuan et al., 2023b)	36.5	47.4	40.0	17.8	48.6
	SCOPE (OPTI)	41.6 (+14.0%)	73.7	44.4	18.2	51.4
	SCOPE (OPSI)	40.2 (+10.1%)	52.6	51.7	22.7	45.7
Llama3.1-8B	Behavior Cloning	14.9	6.0	15.0	14.0	11.0
	AWR (Peng et al., 2019)	29.5	57.9	26.7	17.8	31.4
	DIGIRL (Bai et al., 2024)	31.0	57.9	26.7	17.8	37.1
	RFT (Yuan et al., 2023b)	40.9	68.4	46.7	26.7	37.1
	WEBRL (Qi et al., 2025)	45.8	63.2	46.7	31.1	54.3
	SCOPE (OPTI)	53.6 (+17.0%)	**84.2**	**61.4**	40.0	51.4
	SCOPE (OPSI)	**54.3** (+18.6%)	**84.2**	50.0	**42.2**	**57.1**

Table 2: Results of SCOPE on WEBVOYAGER, which evaluates agents on real online websites.

Model	Method	Total SR	Amazon	Apple	ArXiv	BBC	Cambridge	GitHub	HFace	Wolfram
					Prompting Method					
OAI-4o	-	42.2	51.3	41.9	27.9	32.6	41.9	36.6	46.5	56.5
DeepSeek-v3.2	-	42.0	46.2	40.6	34.6	40.6	29.0	45.0	20.7	65.2
					Finetuning Method					
Qwen2.5-3B	SFT	24.1	13.2	**31.2**	19.0	9.4	**14.0**	34.1	27.0	41.3
	RFT	26.1	18.4	25.0	21.4	**21.9**	0.0	**46.3**	21.6	**51.1**
	SCOPE	**28.4** (+8.8%)	**23.7**	**31.2**	**38.1**	9.4	7.1	34.1	**32.4**	45.7
Llama3.1-8B	SFT	33.4	18.4	25.0	38.1	9.4	**16.3**	58.5	**35.1**	52.2
	RFT	37.3	42.1	34.4	**52.4**	15.6	9.3	48.8	32.4	**56.5**
	SCOPE	**42.8** (+14.7%)	**52.6**	**37.5**	**52.4**	**28.1**	**16.3**	**65.9**	**35.1**	50.0

the importance of fine-tuning Web agents. Moreover, methods that optimize the policy through environment interaction, such as RFT and GRPO, generally outperform SFT, suggesting that direct interaction and reward-based improvement provide an effective path for enhancing Web-agent capabilities. Beyond these methods, SCOPE adopts a short-to-long progressive exploration strategy and uses agentic on-policy distillation to internalize effective skills discovered across rollouts. As a result, SCOPE consistently outperforms the strongest corresponding baselines across WEBARENA, WEBVOYAGER, and WEBSHOP, achieving relative gains of 3.9%–18.6%.

Table 3: SCOPE outperforms online RL methods in the simulated WEBSHOP environment.

Model	Method	Score	Succ. SR
Qwen2.5-3B	SFT	0.731	55.6
	RFT	0.780	60.0
	GRPO	0.898	80.8
	SCOPE	**0.933** (+3.9%)	**84.6** (+4.7%)
Llama3.1-8B	SFT	0.719	52.6
	RFT	0.833	68.0
	GRPO	0.889	76.2
	SCOPE	**0.938** (+5.5%)	**85.2** (+11.8%)

SCOPE shows broad applicability across Web environments and model scales. (1) **Environment perspective**: WEBARENA and WEBVOYAGER go beyond low-cost simulators and involve realistic or online Web environments, where interactions are often non-stationary, expensive, and difficult to parallelize. These properties make standard agentic RL methods, such as GRPO, difficult to apply effectively in such settings. By contrast, SCOPE consistently improves performance in both realistic online and controllable simulated environments, suggesting that its local-skill-driven exploration mechanism remains applicable in dynamic Web environments. (2) **Model perspective**: As shown in Table 1, Table 2, and Table 3, SCOPE improves both Qwen2.5-3B and Llama3.1-8B, showing that it is not limited to a particular model scale. Moreover, Llama3.1-8B obtains more pronounced gains than Qwen2.5-3B, suggesting that SCOPE's progressive exploration can scale with model capability.

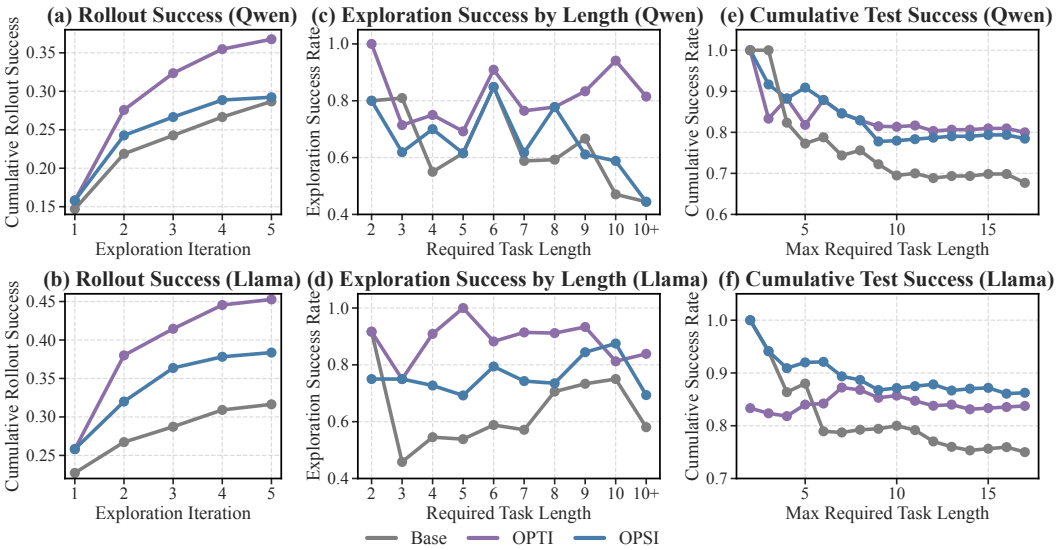


Figure 2: Long-horizon exploration dynamics and intervention behavior of SCOPE during training. Rows correspond to Qwen2.5-3B and Llama3.1-8B; columns show cumulative rollout success, exploration success by task length, and cumulative test success by maximum task length.

4.3 Tracing Exploration Dynamics to Long-Horizon Capability

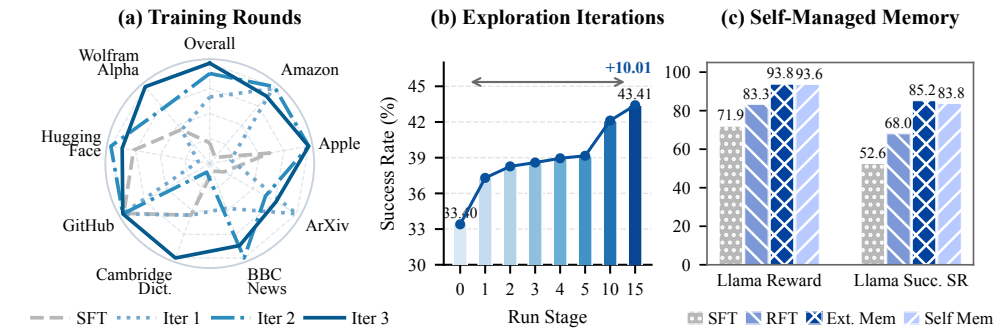
To further understand the source of SCOPE's performance gains, we conduct a systematic mechanistic analysis of its progressive exploration process, tracing how improved exploration dynamics translate into stronger long-horizon model capability. (1) **SCOPE exhibits more stable and sustained exploration ability**. As shown in Figure 2(a,b), for the same number of rollouts, SCOPE discovers more reward-positive samples than direct rollouts, indicating more effective exploration. (2) **SCOPE shows a clear exploration advantage on longer and harder tasks.** We group successful rollouts by the required number of steps for each task. We exclude tasks that cannot be solved by any model, and use the shortest successful trajectory length among all models as the estimated required length for each remaining task. As shown in Figure 2(c,d), SCOPE achieves higher rollout success rates on longer tasks, suggesting that its progressive exploration mechanism helps reach task states that are difficult to access through direct rollouts. (3) **The model trained with SCOPE obtains stronger performance gains on long-horizon tasks**. As shown in Figure 2(e,f), SCOPE-trained models maintain higher cumulative test success on tasks that require longer interaction trajectories, further demonstrating the clear advantage of SCOPE-trained models on long-horizon tasks.

4.4 Understanding the Effectiveness of OPTI and OPSI

Table 1 shows that both OPTI and OPSI improve WebArena performance, but their relative advantages depend on model capability. We explain this pattern from two aspects: (1) **Exploration.** As shown in Figure 2(c,d), OPTI achieves stronger rollout performance than OPSI, because auxiliary-model action intervention can more directly push the agent beyond its current exploration frontier. This is especially helpful for weaker policies such as Qwen2.5-3B. (2) **Internalization.** Although OPSI is less aggressive than OPTI during exploration, its corrective actions are generated by the policy model itself. For stronger policies such as Llama3.1-8B, these self-corrected behaviors are closer to the model's own policy distribution and are therefore easier to internalize through distillation. This observation is consistent with prior findings on student-friendly distillation (Yuan et al., 2023a).

4.5 Further Analysis of Scalability, Self-Managed Memory, Ablations, and Efficiency

SCOPE benefits from scaling both outer training rounds and inner exploration iterations. In this section, we further analyze two key scaling factors of SCOPE: (1) **the number of outer training rounds**. As shown in Figure 3(a), increasing the number of outer training rounds consistently improves performance on WEBVOYAGER, yielding a total gain of 4.25% from one round to three rounds. (2) **the number of inner progressive exploration iterations**. As shown in Figure 3(b),



(d) Ablation Study

Method	Reward	Succ. SR
SCOPE	**0.933**	**84.6**
w/o Skill Discovery	0.827 (-11.4%)	66.6 (-21.3%)
w/o Skill Orchestration	0.897 (-3.9%)	77.0 (-9.0%)
w/o Skill Expansion	0.816 (-12.5%)	67.2 (-20.6%)
RFT	0.780 (-16.4%)	60.0 (-29.1%)
SFT	0.731 (-21.7%)	55.6 (-34.3%)

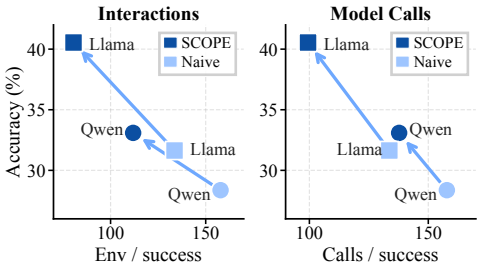


Figure 3: Further analysis of SCOPE. (a) SCOPE scales with outer training rounds. (b) SCOPE scales with progressive exploration iterations. (c) Self-managed memory achieves performance comparable to external memory. (d) Key components of SCOPE contribute to final performance. (e) SCOPE achieves a better training-efficiency trade-off in environment interactions and model calls.

increasing the number of inner exploration iterations also improves the overall success rate from 33.40% at the initial rollout stage to 43.41% after 15 stages. Overall, SCOPE shows favorable scaling behavior along both the policy-internalization and exploration-expansion dimensions: more outer rounds allow the policy to continually internalize task-completing skill chains discovered during exploration, while more inner iterations expand the reachable task frontier before each policy update.

SCOPE enables a self-evolving training loop. We investigate whether SCOPE can replace the external memory manager with a self-managed one by fine-tuning a memory-management model from the same backbone as the policy model on pre-collected training data. As shown in Figure 3(c), self-managed memory achieves a final improvement comparable to the external-model setting, suggesting that SCOPE can reduce reliance on external models and form a closed loop of self-evolution.

SCOPE remains effective without auxiliary information. SCOPE uses the auxiliary signal c in skill discovery and orchestration to align local-progress assessment with the standard task evaluator and provide limited evidence of progress toward task completion. A natural question is whether its performance gains depend on this additional information. To examine this, Table 4 compares SCOPE with and without c on WebArena using Llama3.1-8B. Removing c from all stages largely preserves the OPTI reference's success rate (53.6% with c versus 52.7% without it). These results indicate that the auxiliary information serves as a lightweight signal for the memory-management model and contributes only marginally to final performance.

All components of SCOPE are necessary for effective progressive learning. As shown in Figure 3(d), we evaluate three ablated variants: w/o Skill Discovery replaces semantic skill discovery with fixed-length trajectory segmentation every two steps; w/o Skill Orchestration replaces dynamic skill orchestration with a heuristic that always selects the deepest skill chain; and w/o Skill Expansion removes failure-experience-based frontier intervention, letting the policy continue self-exploration from the composed skill chain with a random temperature (tmp=1.0). Removing any component reduces performance; when all three are removed, the method degenerates to RFT and drops further, indicating that these modules play complementary roles in stable progressive learning.

SCOPE improves the cost–benefit trade-off. Training agents in real environments incurs two costs: model calls and environment interactions. We therefore evaluate training efficiency on WEBARENA by measuring the number of model calls and environment interactions required per successful task discovered during exploration. Compared with naive sampling, SCOPE introduces memory-

Table 4: Auxiliary-information ablation on WebArena with Llama3.1-8B.

Method	Total SR	Reddit	GitLab	Shopping	CMS
WEBRL	45.8	63.2	46.7	31.1	54.3
SCOPE w/ c	**53.6**	**84.2**	**61.4**	40.0	51.4
SCOPE w/o c	52.7	68.4	53.3	40.0	**60.0**

Table 5: Model calls during training data collection, decomposed by stage. RFT corresponds to naive sampling. The WebShop GRPO count is approximate.

Model	Method	Policy Rollout	Discovery	Orchestration	Expansion	Total
			WebArena			
Qwen2.5-3B	RFT	24,618	0	0	0	24,618
	SCOPE	20,359	2,107	1,569	1,034	25,069
Llama3.1-8B	RFT	23,250	0	0	0	23,250
	SCOPE	17,957	1,902	1,358	959	**22,176**
			WebShop			
Qwen2.5-3B	RFT	168,621	0	0	0	168,621
	GRPO	≈242,000	0	0	0	≈242,000
	SCOPE	92,354	7,767	3,067	5,541	**108,729**

management calls between rollout rounds. However, these calls help the agent reuse verified local skills and avoid repeated low-value exploration, reducing the cost of discovering successful tasks. As shown in Figure 3(e), SCOPE improves cost–benefit trade-off across model scales, discovering more reward-positive samples with fewer environment interactions and model calls.

Fine-grained model-call cost analysis. Table 5 reports total model calls at each stage. Overall, on WebArena, SCOPE has a model-call cost comparable to RFT/naive sampling, requiring 1.8% more calls with Qwen and 4.6% fewer with Llama. On WebShop, SCOPE uses 35.5% fewer model calls than RFT and 55.1% fewer than GRPO. These results show that SCOPE achieves better performance with comparable or fewer model calls.

5 Conclusion

We presented SCOPE, a progressive exploration framework for long-horizon Web agents. SCOPE decomposes exploration into a short-to-long skill-acquisition process that discovers skills and failure experiences, composes them into task-completing skill chains, and expands outward from them. The resulting task-completing skill chains are then internalized into the policy through agentic on-policy distillation. Experiments across realistic and simulated Web environments show that SCOPE consistently improves exploration dynamics and final task performance across model backbones.

Acknowledgments

This work is supported in part by Huawei Innovation Research Programs. We gratefully acknowledge the support from MindSpore, CANN, and Ascend AI Processor used for this research.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, et al. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Saaket Agashe, Jiuzhou Han, Shuyu Gan, et al. Agent s: An open agentic framework that uses computers like a human, 2024. URL https://arxiv.org/abs/2410.08164.

Hao Bai, Yifei Zhou, Mert Cemri, et al. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning, 2024. URL https://arxiv.org/abs/2406.11896.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, et al. First return, then explore. *Nature*, 590(7847): 580–586, Feb 2021. doi: 10.1038/s41586-020-03157-9. URL https://doi.org/10.1038/s41586-020-03157-9.

Lang Feng, Zhenghai Xue, Tingcong Liu, et al. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, et al. Large language model-based human-agent collaboration for complex task solving. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1336–1357, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.72. URL https://aclanthology.org/2024.findings-emnlp.72/.

Paul M. Fitts and Michael I. Posner. *Human Performance*. Brooks/Cole, 1967.

Apurva Gandhi and Graham Neubig. Go-browse: Training web agents with structured exploration, 2026. URL https://arxiv.org/abs/2506.03533.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hongliang He, Wenlin Yao, Kaixin Ma, et al. Webvoyager: Building an end-to-end web agent with large multimodal models, 2024. URL https://arxiv.org/abs/2401.13919.

Hongliang He, Wenlin Yao, Kaixin Ma, et al. OpenWebVoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, et al., editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27545–27564, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1336. URL https://aclanthology.org/2025.acl-long.1336/.

Tenghao Huang, Kinjal Basu, Ibrahim Abdelaziz, et al. R2D2: Remembering, replaying and dynamic decision making with a reflective agentic memory. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, et al., editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30318–30330, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1464. URL https://aclanthology.org/2025.acl-long.1464/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, et al. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Xiao Liu, Tianjie Zhang, Yu Gu, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. *arXiv preprint arXiv:2408.06327*, 2024.

Cong Lu, Shengran Hu, and Jeff Clune. Intelligent go-explore: Standing on the shoulders of giant foundation models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=apErWGzCAA.

Xue Bin Peng, Aviral Kumar, Grace Zhang, et al. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

Zehan Qi, Xiao Liu, Iat Long Iong, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning, 2025. URL https://arxiv.org/abs/2411.02337.

Qwen, An Yang, Baosong Yang, et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Paloma Sodhi, S. R. K. Branavan, Yoav Artzi, et al. Step: Stacked llm policies for web actions, 2024. URL https://arxiv.org/abs/2310.03720.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models, 2026. URL https://arxiv.org/abs/2604.00626.

Lei Wang, Chen Ma, Xueyang Feng, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL http://dx.doi.org/10.1007/s11704-024-40231-1.

Zhepei Wei, Wenlin Yao, Yao Liu, et al. WebAgent-r1: Training web agents via end-to-end multi-turn reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, et al., editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7909–7928, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.401. URL https://aclanthology.org/2025.emnlp-main.401/.

Fengli Xu, Qianyue Hao, Zefang Zong, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models, 2025. URL https://arxiv.org/abs/2501.09686.

Qianlan Yang, Xiangjun Wang, Danielle Perszyk, et al. Self-guided hierarchical exploration for generalist foundation model web agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9twwDW60Bw.

Shunyu Yao, Howard Chen, John Yang, et al. Webshop: Towards scalable real-world web interaction with grounded language agents. In *ArXiv*, preprint. URL https://arxiv.org/abs/2207.01206.

Mengyang Yuan, Bo Lang, and Fengnan Quan. Student-friendly knowledge distillation, 2023a. URL https://arxiv.org/abs/2305.10893.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, et al. Scaling relationship on learning mathematical reasoning with large language models, 2023b. URL https://arxiv.org/abs/2308.01825.

Siyan Zhao, Zhihui Xie, Mengchen Liu, et al. Self-distilled reasoner: On-policy self-distillation for large language models, 2026. URL https://arxiv.org/abs/2601.18734.

Yaowei Zheng, Richong Zhang, Junhao Zhang, et al. Llama-factory: Unified efficient fine-tuning of 100+ language models, 2024. URL https://arxiv.org/abs/2403.13372.

Shuyan Zhou, Frank F. Xu, Hao Zhu, et al. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

Yifei Zhou, Qianlan Yang, Kaixiang Lin, et al. Proposer-agent-evaluator (PAE): Autonomous skill discovery for foundation model internet agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 79490–79528. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25ah.html.

A Algorithm

Algorithm 1 SCOPE progressive skill exploration and distillation.

Require: task distribution $\mathcal{D}$, initial policy π_θ, inner exploration budget J
Notation: $\oplus$ denotes trajectory concatenation.
initialize skill memory $\mathcal{B}_{\text{skill}} \leftarrow \emptyset$ and failure memory $\mathcal{B}_{\text{fail}} \leftarrow \emptyset$
for each outer training round **do**
  sample a task goal $G \sim \mathcal{D}$ and obtain auxiliary task information c
  collect an initial rollout τ_0 from scratch using π_θ
  initialize teacher-trajectory buffer $\mathcal{H} \leftarrow \emptyset$
  for $j = 1, \ldots, J$ **do**
    $(\xi_i, y_i)_{i=1}^{n} \leftarrow \textsc{Discover}(G, c, \tau_{j-1})$
    $\mathcal{B}_{\text{skill}} \leftarrow \mathcal{B}_{\text{skill}} \cup \{\xi_i \mid y_i = 1\}$
    $\mathcal{B}_{\text{fail}} \leftarrow \mathcal{B}_{\text{fail}} \cup \{\xi_i \mid y_i = 0\}$
    build candidate frontier set $\mathcal{S}_{\text{frontier}}$ from $\mathcal{B}_{\text{skill}}$
    select $s^\star \leftarrow \textsc{Select}(G, c, \mathcal{S}_{\text{frontier}})$
    compose and execute a skill chain $p^\star$ from the initial state to $s^\star$
    retrieve matched failures $\mathcal{F}^\star \leftarrow \{\xi \in \mathcal{B}_{\text{fail}} \mid s_{\texttt{start}}(\xi) = s^\star\}$
    if using OPTI **then**
      generate frontier action $a^\star \leftarrow \textsc{Intervene}(G, s^\star, \mathcal{F}^\star)$
    else
      generate frontier action $a^\star \sim \pi_\theta(\cdot \mid G, s^\star, \textsc{Summarize}(G, p^\star, \mathcal{F}^\star))$
    end if
    execute $a^\star$ at $s^\star$ and observe the next state s^+
    continue rollout from s^+ using policy π_θ, obtaining suffix rollout τ_j^+
    construct $\tau_j \leftarrow p^\star \oplus a^\star \oplus \tau_j^+$
    if $R(\tau_j) = 1$ **then**
      add the task-completing trajectory τ_j to $\mathcal{H}$
    end if
  end for
  update π_θ by minimizing $\mathcal{L}_{\text{distill}}(\theta)$ on trajectories in $\mathcal{H}$
end for
return updated policy π_θ, skill memory $\mathcal{B}_{\text{skill}}$, failure memory $\mathcal{B}_{\text{fail}}$

B Related Work

LLM-Based Web Agents. Large language models have driven autonomous agents from text generation toward real interactive environments. As a representative interface through which humans interact with the digital world, the Web has become an important testbed for autonomous-agent research. Existing work has introduced a range of Web interaction benchmarks and agent frameworks, such as WEBARENA (Zhou et al., 2024), WEBVOYAGER (He et al., 2024), and WEBSHOP (Yao et al., preprint), to evaluate whether agents can complete user-specified tasks in real or simulated websites. In these environments, agents need to access target URLs through a browser and perform browser actions based on text-based webpage observations, such as HTML or AXTree. Early Web-agent methods mainly rely on prompting strong LLMs, using task decomposition, trajectory planning, webpage-observation compression, memory, and tool use to guide task execution (Sodhi et al., 2024; Huang et al., 2025; Agashe et al., 2024). In contrast, we focus on training lightweight open-source Web agents.

Web-Agent Training. Recent studies have begun to explore how to train Web agents, especially by allowing agents to interact directly with environments, collect behavioral data through trial and error, and optimize their policies using reward signals provided by environmental feedback. OpenWebVoyager (He et al., 2025) and PAE (Zhou et al., 2025) exploit the asymmetry that generating and verifying Web tasks can be easier than solving them, and continuously create new tasks to improve agent capabilities. WebRL (Qi et al., 2025) and SAGE (Yang et al., 2025) focus on the challenge of training-task difficulty by constructing dynamic curricula that generate tasks according

to the model's weaknesses on the training set. To address settings where the environment is difficult to characterize through predefined task generation, Go-Browser (Gandhi and Neubig, 2026) proposes a bottom-up approach that discovers and constructs tasks during exploration. WebAgent-R1 (Wei et al., 2025) and GiGPO (Feng et al., 2025) further optimize Web agents with online reinforcement learning from environment interactions. However, these methods still largely rely on naive rollouts: given a user query, the agent samples complete trajectories end to end from the initial state. In long-horizon Web tasks, such training processes often suffer from low exploration efficiency, scarce successful trajectories, and repeated failures. In contrast, SCOPE goes beyond this paradigm by decomposing long and difficult queries into reusable local skills during exploration, then progressively composing and expanding these skills to support effective and sustained exploration.

On-Policy Distillation. Distillation is a natural alternative to sparse-reward reinforcement learning, since teacher trajectories can provide supervision at each step. However, offline distillation often suffers from an off-policy issue: fixed teacher data can be mismatched with the interaction distribution of the current policy in dynamic Web environments, and step-level imitation becomes ineffective once the student policy fails to reach the states in the teacher trajectories. To mitigate this problem, prior work has proposed OPD (Song and Zheng, 2026) and OPSD (Zhao et al., 2026), whose core idea is to provide teacher supervision for each next token under states generated by the student LLM itself, often through techniques such as reverse KL divergence.

C Distillation View of SCOPE

The key advantages of SCOPE can also be understood from the perspective of on-policy distillation. Distillation is a natural alternative to sparse-reward reinforcement learning, since teacher trajectories can provide denser step-level supervision for policy learning. However, conventional offline distillation is limited in Web-agent settings: (1) it imitates all behaviors in complete teacher trajectories rather than focusing on the key decisions that determine task progress; (2) it suffers from off-policy distribution shift in dynamic Web environments, where the student policy may fail to reach the states covered by fixed teacher data; and (3) it often depends on the availability of a strong external teacher.

From this perspective, we propose agentic OPD. Unlike conventional OPD, which mainly focuses on token-level distribution matching, we emphasize on-policy learning at the level of agent-environment interaction steps: the states used for distillation are environment states that the student policy can actually reach. SCOPE addresses the limitations above through three designs:

- *Skill Discovery* decomposes trajectories into subtask units, thereby exposing critical local progress and failure decision points rather than imitating external expert guidance at every step.
- *Skill Orchestration* composes skill chains from behaviors produced by the policy model's own historical rollouts, helping preserve the on-policy nature of the distilled supervision.
- *Skill Expansion* constructs intervention signals from failure experiences, providing corrective actions for subsequent exploration. The teacher constructed from failure experiences can co-evolve with the policy model; furthermore, the teacher role can also be played by the policy model itself, enabling self-distillation and reducing reliance on a strong external teacher.

D Theoretical Details

D.1 Proof of the Effective-Frontier-State Lower Bound

Lemma 1. Let α denote the *intrinsic constraint dimension* of a Web task, i.e., the number of key task constraints that must be progressively satisfied in order to complete the task. Let δ_k denote the probability that, when the current memory frontier is at depth $k-1$, the agent selects an effective frontier state from the candidate frontier states that enables expansion from depth $k-1$ to depth k. If, at any fixed depth level, the number of webpage functional patterns relevant to the current subtask progression is bounded, then there exists a constant $C > 0$ such that

$$\delta_k \geq \frac{1}{Ck^{\alpha}}.$$

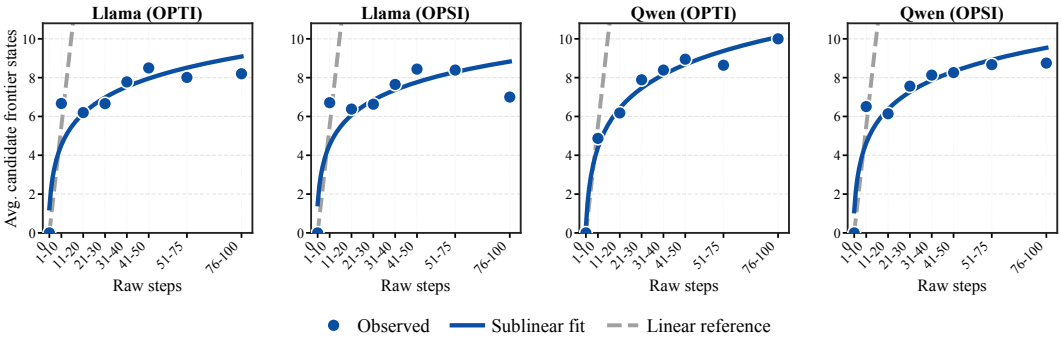


Figure 4: Growth of non-redundant candidate frontier states as raw exploration steps increase. Across model backbones and intervention variants, the number of candidate frontier states grows much more slowly than the linear reference $y = x$, supporting the conservative nature of the counting bound.

Proof. Task progression can be viewed as gradually satisfying α key constraint dimensions. By construction, successful entries that are functionally identical and progress-equivalent lead to the same canonical frontier state. Therefore, within any fixed functional pattern, redundant, cyclic, or equivalent rollouts do not increase the number of distinguishable candidate frontier states.

Moreover, at any fixed depth level, the number of webpage functional patterns relevant to the current subtask progression is bounded. Hence, whenever the frontier advances from depth $t-1$ to depth t, each constraint dimension can introduce at most a constant number of new distinguishable states. Accumulated up to frontier depth k, each constraint dimension therefore has at most $O(k)$ distinguishable states.

Since a candidate frontier state is jointly determined by these α constraint dimensions, the total number of candidate frontier states at depth k is at most

$$O(k^{\alpha}).$$

Thus, there exists a constant $C > 0$ such that

$$|\mathcal{S}^{(k)}_{\text{frontier}}| \leq Ck^{\alpha},$$

where $\mathcal{S}^{(k)}_{\text{frontier}}$ denotes the set of candidate frontier states used to expand the frontier from depth $k-1$ to depth k.

Since depth k is defined as a valid next expansion target of the current frontier, the set $\mathcal{S}^{(k)}_{\text{frontier}}$ contains at least one effective frontier state that enables expansion from depth $k-1$ to depth k. Therefore,

$$\delta_k \geq \frac{1}{|\mathcal{S}^{(k)}_{\text{frontier}}|} \geq \frac{1}{Ck^{\alpha}}.$$

□

Remark on the Bound. Although the analysis above establishes a polynomial bound, our empirical results suggest that the practical growth pattern is often substantially better than this theoretical upper bound. Specifically, we examine how the size of the candidate frontier set $|\mathcal{S}^{(k)}_{\text{frontier}}|$ changes with the number of rollout steps. As shown in Figure 4, across model backbones and intervention variants, the number of non-redundant candidate frontier states grows much more slowly than the linear reference $y = x$, indicating a highly sublinear growth pattern in practice. This happens because, although a Web task may allow up to dozens of interaction steps, the number of functionally distinct webpage modules that can affect the next subtask is usually limited. This leads to much slower growth of frontier candidates in practice.

D.2 Proof of Proposition 1

Proposition 1 (Restated). Assume that making a locally correct frontier-expanding decision has probability lower bounded by $p \in (0, 1]$. Under the polynomial lower bound on effective-frontier-

state retrieval established in Lemma 1, the total number of exploration rollouts T_{total} has polynomial variability in task depth; specifically,

$$\text{Std}(T_{\text{total}}) = O\left(\frac{L^{(1+2\alpha)/2}}{p}\right).$$

Proof. Let T_k denote the number of exploration rollouts required to extend the frontier from depth $k-1$ to depth k under self-composed progressive exploration, so that

$$T_{\text{total}} = \sum_{k=1}^{L} T_k.$$

At depth k, the probability of selecting an effective frontier state is δ_k, and conditioned on such a state, the probability of making a locally correct frontier-expanding decision is lower bounded by p. Hence the overall probability of extending the frontier at depth k is

$$q_k = p\delta_k,$$

and under this abstraction,

$$T_k \sim \text{Geom}(q_k).$$

For a geometric random variable with success probability q_k, we have

$$\mathbb{E}[T_k] = \frac{1}{q_k}, \qquad \text{Var}(T_k) = \frac{1-q_k}{q_k^2}.$$

Substituting $q_k = p\delta_k$ yields

$$\mathbb{E}[T_k] = \frac{1}{p\delta_k}, \qquad \text{Var}(T_k) = \frac{1-p\delta_k}{p^2\delta_k^2} = \Theta\left(\frac{1}{p^2\delta_k^2}\right).$$

Assuming $\{T_k\}_{k=1}^{L}$ are independent under the stylized abstraction, it follows that

$$\text{Var}(T_{\text{total}}) = \sum_{k=1}^{L} \text{Var}(T_k) = \Theta\left(\sum_{k=1}^{L} \frac{1}{p^2\delta_k^2}\right).$$

By Lemma 1, there exists a constant $c > 0$ such that

$$\delta_k \geq ck^{-\alpha}.$$

Hence

$$\frac{1}{\delta_k^2} \leq \frac{k^{2\alpha}}{c^2},$$

and therefore

$$\text{Var}(T_{\text{total}}) = O\left(\frac{1}{p^2c^2}\sum_{k=1}^{L} k^{2\alpha}\right).$$

Since

$$\sum_{k=1}^{L} k^{2\alpha} = \Theta(L^{2\alpha+1}),$$

we obtain

$$\text{Var}(T_{\text{total}}) = O\left(\frac{L^{2\alpha+1}}{p^2c^2}\right), \qquad \text{Std}(T_{\text{total}}) = O\left(\frac{L^{(2\alpha+1)/2}}{p}\right),$$

where the constant c from Lemma 1 is absorbed into the big-O notation.

Finally, under restart-from-scratch exploration, reaching depth k requires making correct decisions for all preceding k steps. Since each such decision succeeds with probability p, the probability of reaching depth k is p^k, so the number of rollouts required to reach depth k is geometric with mean $\Theta(p^{-k})$, which grows exponentially in k. In particular, completing a task of target depth L requires making all L critical decisions correctly, so the success probability becomes p^L. Consequently, both the expected number of exploration rollouts required and its variability grow exponentially with task depth under restart-from-scratch exploration. In contrast, under self-composed progressive exploration, the variability remains polynomial, establishing the claimed advantage. □

E Case Study

Figures 5 and 6 present a short-to-long exploration example from a shopping task in WebArena. In the first attempt, the agent discovers skills for navigating from the store front page to the order page and for flipping through the order list, while also recording a failure pattern in which it accidentally enters the contact/feedback page. In the second and third attempts, the agent reuses the validated order-page navigation skill, but further exposes a repeated failure pattern of flipping through order pages. In the fourth attempt, the agent continues reusing existing skills and explores a new view-changing skill that displays more orders per page; however, this attempt still fails because the target information is not found after the view change. In the final attempt, SCOPE composes a skill chain for entering the order page and changing the view, and successfully completes the task. This example illustrates the complementary roles of skill memory and failure memory: validated local skills preserve reusable progress, while failure records identify frontier states where subsequent exploration should be corrected.



Figure 5: Early attempts in the case study. Top: the first attempt discovers reusable navigation and pagination skills, while recording a failure in which the agent accidentally enters the contact/feedback page. Bottom: the second and third attempts reuse the order-page navigation skill but expose a repeated failure pattern of flipping through order pages.

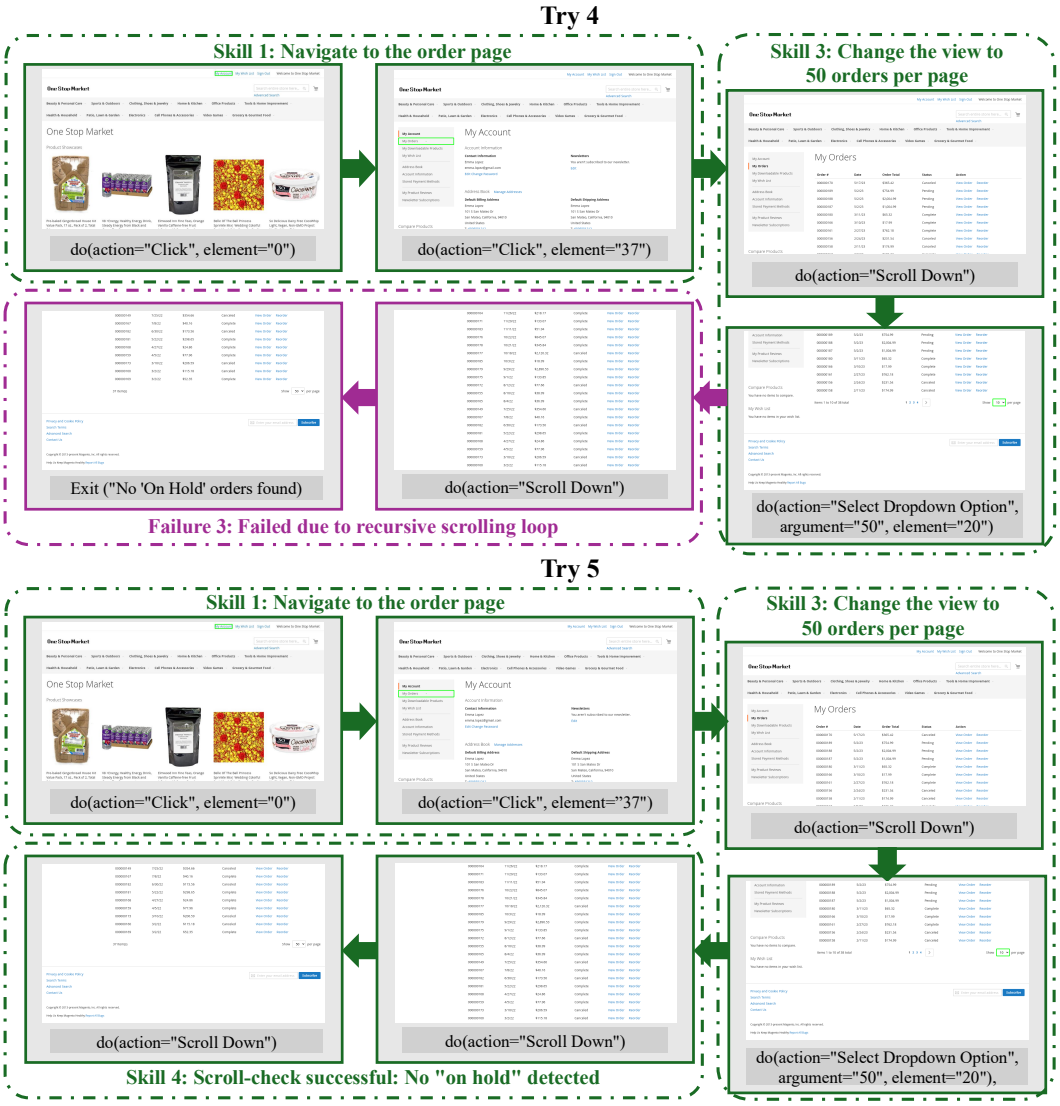


Figure 6: Later attempts in the case study. Top: in the fourth attempt, the agent changes the order list to show more entries per page, but still fails because the target information is not found after the view change. Bottom: in the final attempt, SCOPE composes the navigation and view-changing skills into a skill chain and successfully completes the task on top of this chain.

F Additional Experimental Details

Base Model Warm-Start Stage. Table 6 summarizes the supervised data sources used to initialize the policy models. Unless otherwise specified, we use the instruct versions of the corresponding backbone models as the base policy models. In the warm-up stage, we first apply supervised fine-tuning to the corresponding backbone model, and the data sources and scales used for fine-tuning are reported in Table 6. The main exception is the Llama3.1-8B setting on WEBARENA, where we follow prior work (Wei et al., 2025) and initialize from the WEBRL checkpoint rather than directly from the instruct model. This setup matches prior WebArena training practice and provides a stronger Web-specific starting point for subsequent progressive exploration.

Auxiliary Information. During training, SCOPE uses task-related auxiliary information c in DISCOVER and SELECT to identify local progress and judge which frontier state is closer to the task goal. This section details how c is constructed. c is used to assist answer evaluation, such as the target URL and the expected states of webpage elements, but its form varies across datasets: (1) in **WEBARENA**, c includes the target URL and the states of task-specific target elements; (2) in **WEBSHOP**, c consists of the constraint information of the target item; and (3) in **WEBVOYAGER**, where such structured auxiliary information is unavailable, we instead use the evaluation reason produced by the LLM-as-judge for each rollout as the auxiliary signal. Overall, c is an auxiliary signal extracted from answer-evaluation information in the original dataset and related to task completion.

Table 6: Data sources and statistics for supervised policy initialization across benchmarks.

Benchmark	SFT data source	Trajectories	Steps
WEBARENA	VisualAgentBench (Liu et al., 2024)	1122	9460
WEBVOYAGER	DeepSeek sampling (DeepSeek-AI, 2025)	2987	15763
WEBSHOP	DeepSeek sampling (DeepSeek-AI, 2025)	2400	13337

Train/Test Splits and Rollout Budgets. Table 7 reports the train/test splits, task budgets, and SCOPE iteration counts used in each benchmark. The total number of rollout attempts is matched between RFT and SCOPE.

Table 7: Train/test splits, task budgets, and SCOPE iteration counts used in each benchmark.

Benchmark	Reference setting	Train tasks	Test tasks	Max steps	Outer rounds	Inner iters.
WEBARENA	WEBRL (Qi et al., 2025)	551	134	30	1	5
WEBVOYAGER	OpenWebVoyager (He et al., 2025)	461	311	15	1/3	5
WEBSHOP	VERL-Agent (Feng et al., 2025)	6410	500	10	1	4

Auxiliary-Model Prompts. Figures 7, 8, and 9 show the prompt templates for DISCOVER, SELECT, and INTERVENE. DISCOVER decomposes a trajectory into CMUs and identifies reusable progress and failure-causing branches, SELECT chooses the reusable prefix as restart anchor for the next rollout, and INTERVENE generates corrective guidance and the next action at the selected frontier state.

G Baseline and Benchmark Details.

G.1 Baseline Details

Prompting. For prompting baselines, we follow the original setup of each dataset and directly prompt general-purpose models without fine-tuning. On WEBARENA and WEBVOYAGER, we also report representative strong prompting models such as OAI-4o and DeepSeek-V3.2 where applicable.

SFT. SFT denotes standard supervised fine-tuning on offline trajectory data for each benchmark; the detailed data sources are summarized in Table 6.

RFT. RFT denotes reject-sampling fine-tuning (Yuan et al., 2023b). Specifically, we first collect multiple rollout trajectories for each task, then filter for successful trajectories with $R(\tau) = 1$, and

Table 8: Benchmark details.

Benchmark	Websites	Observation and action space	Evaluation
WEBARENA	Reddit, Shopping, CMS, and GitLab.	HTML-based observations. Actions: `Click`, `Right Click`, `Type`, `Search`, `Hover`, `Scroll Up`, `Scroll Down`, `Press Enter`, `Switch Tab`, `Select Dropdown Option`, `Wait`, `exit(message)`, `go_backward()`, and `go_forward()`.	Rule matching + LLM-as-judge.
WEBVOYAGER	Amazon, Apple, ArXiv, BBC, Cambridge, GitHub, HFace, and Wolfram.	AXTree-based observations. Actions: `Click [id]`, `Type [id]; text`, `Scroll [id/WINDOW]; up/down`, `GoBack`, `Google`, `Wait`, and `ANSWER; text`.	LLM-as-judge.
WEBSHOP	A simulated shopping website.	Actions: `search[keywords]` and `click[value]`.	Rule matching.

finally fine-tune the policy on these trajectories. We evaluate RFT on all three benchmarks. Unless otherwise specified, RFT and SCOPE use the same total number of rollout attempts.

AWR AND DIGIRL. AWR (Peng et al., 2019) is an offline RL baseline that directly optimizes the policy model with an advantage-weighted regression loss. DigiRL (Bai et al., 2024) builds on AWR by improving the advantage estimator and introducing an automatic curriculum. For further details, we refer readers to the WEBRL paper (Qi et al., 2025). We directly take the results of these two baselines from the WEBRL paper.

WEBRL. WEBRL is a representative and competitive Web-agent training baseline on WEBARENA (Qi et al., 2025). Its core idea is to use an external model to generate additional tasks similar to failed ones after Web-agent rollouts, thereby forming a curriculum-style learning process. For policy optimization, WEBRL adopts an AWR-style off-policy RL learner.

GRPO. GRPO is a standard fully online RL method, and our implementation follows VERL-Agent (Feng et al., 2025). It samples a group of trajectories for each update and computes advantages within the group. However, because of its fully online nature, such methods are difficult to apply directly in non-stationary real Web environments with high interaction costs. We therefore include GRPO on WEBSHOP (Yao et al., preprint) as a representative online RL baseline.

G.2 Benchmark Details

Table 8 summarizes the benchmark websites, observation/action spaces, and evaluation protocols.

DISCOVER Prompt Template

You are a memory extractor for long-horizon Web or environment interaction tasks.

Your job is to analyze one trajectory and decompose it into multiple subtask completion segments (CMUs). Each CMU may consist of multiple atomic actions and should represent either reusable intermediate progress or an incorrect or failure-causing branch.

You will be given:
- `Task Goal`
- `Privileged Info`
- `Task Success`
- `Current Trajectory`

met_expectation Rules:
- `Set to true` when the segment achieves meaningful intermediate progress toward the task goal.
- `Set to false` when the segment follows an incorrect branch, gets stuck, reaches an incorrect conclusion, or fails to finish correctly.
- `If task_success=false`, the final CMU must be false and must include the failure-causing final action or failure region.

Privileged-Mismatch Diagnosis:
Use the privileged information to infer why the trajectory did not satisfy the task requirement. Do not use privileged information as an instruction for future actions; use it only to judge whether the observed trajectory satisfies the task.

Required Output JSON Schema:
Return JSON only. Do not include Markdown or any extra text.

```
{
"failure_summary": "<complete trajectory recap + global path logic + one
explicit failure reason>",
"cmus": [
{
"start_step_id": <integer>,
"end_step_id": <integer>,
"skill_summary": "<concrete subtask intent and operations>",
"status_summary": "<precise ending state after this segment>",
"path_logic": "From [Start Obs] to [Current Obs] via: [Action Sequence]",
"met_expectation": <true|false>
}
]
}
```

Hard Constraints:
- `cmus` must be time-ordered, non-overlapping, and cover the full trajectory.
- `The first CMU` is a special root no-op segment with `start_step_id=-1` and `end_step_id=-1`.
- `...`

Input Context:
- `Task Goal: {{goal}}`
- `Privileged Info: {{privileged_info}}`
- `Task Success: {{task_success}}`
- `Current Trajectory: {{trajectory_data}}`

Figure 7: Prompt template for DISCOVER.

SELECT Prompt Template

Your task is to select the best reusable prefix as the restart anchor for the next rollout.

A prefix candidate is a reusable positive trajectory prefix from prefix memory.

You are given:
- the final task goal (Global Goal),
- privileged information or oracle feedback,
- and a set of candidate prefixes with their associated intermediate states.

Each prefix candidate may include:
- a concise semantic description,
- the reached terminal state or boundary,
- select count.

Selection Rules:
- Prefer prefix candidates whose terminal state is highly relevant to the global goal.
- Prefer prefix candidates that provide meaningful intermediate progress.
- Avoid prefix candidates that have been over-explored.
- Avoid prefix candidates whose downstream branches are repeatedly disproven, unless they still expose unexplored branches.
- Do not select only by maximum depth.

Output Format:
Return JSON only. No extra text. Return exactly one JSON object with exactly two keys.

```
{
"rationale": <string>,
"selected_prefix_id": <string>
}
```

Input Context:
- Global Goal: {{goal}}
- Privileged Info: {{privileged_info}}
- Prefix Memory (JSON list of prefix candidates): {{cmu_trajectory_json}}

Figure 8: Prompt template for SELECT.

INTERVENE Prompt Template

You are a long-horizon environment interaction agent.

You are **NOT** starting from scratch. A reusable skill prefix has already been replayed, and you are now at the prefix frontier state.

Your task is to generate alternative action guidance and the next action based on prior failed trajectory reflections and the current step input.

Your output format must strictly follow:
- <dataset-specific action space>

Only output JSON in the following format:
```
{
"rational": "...",
"next_action_guidance": "...",
"next_action": "..."
}
```

Constraints:
1. Output exactly one JSON object. Do not output markdown, code fences, or any extra explanation text.
2. <dataset-specific constraints>

Task:
{intent}

Current step input (merged):
{current_step_input}

Matched failure memories for this prefix:
{failure_memories if failure_memories else "(empty)"}

Original failed behavior (from selected CMU actions):
{original_failed_behavior if original_failed_behavior else "(empty)"}

Figure 9: Prompt template for INTERVENE.

H Repeated WebVoyager Evaluation after Website Updates

During the rebuttal period, we evaluated each trained RFT and SCOPE model three times on six websites. Apple and arXiv underwent major website updates compared with the initial experiments, so we excluded them from this supplemental evaluation. The remaining websites are Amazon, BBC, Cambridge Dictionary, GitHub, Hugging Face, and Wolfram Alpha.

Avg@3 is the mean task success rate over the three evaluations. Pass@3 is the fraction of tasks solved at least once across those evaluations. We report the sample standard deviation of the three run-level overall success rates: RFT achieves 34.76% $\pm$ 0.55 percentage points and SCOPE achieves 38.70% $\pm$ 2.74 percentage points. SCOPE improves Avg@3 by 3.94 percentage points and Pass@3 by 8.53 percentage points.

Table 9: Supplemental three-run WebVoyager evaluation on six websites (success rates, %). Overall Avg@3 is reported as mean $\pm$ sample standard deviation in percentage points.

Metric	Method	Total SR	Amazon	BBC	Cambridge	GitHub	HFace	Wolfram
Avg@3	RFT	34.76 ± 0.55	39.74	18.28	9.40	43.90	38.71	55.04
	SCOPE	$\mathbf{38.70} \pm 2.74$	50.00	19.35	15.38	52.03	45.16	49.61
Pass@3	RFT	51.66	69.23	35.48	20.51	60.98	64.52	62.79
	SCOPE	**60.19**	73.08	41.94	35.90	73.17	74.19	65.12

I Limitations

Our current implementation mainly considers text-based observations of Web tasks. Incorporating visual inputs, such as screenshots, may further improve performance because cues such as layout, position, and color provide important information for Web interaction. In addition, in real dynamic websites, the validity of learned skills may fluctuate as page states, interfaces, or backend responses change over time. Although SCOPE uses robust skill execution to improve skill reusability, a small number of previously acquired skills may still become inapplicable under changed Web contexts.

J Broader Impacts

This work conducts experiments primarily in realistic Web environments, where both the exploration process and the resulting stronger Web agents may occasionally produce unintended or undesirable actions on live websites. Although our experiments are conducted for research purposes, these risks highlight the importance of effective monitoring and control mechanisms when improving Web agents in real environments. In future work, developing reliable safeguards, task-level permissions, and human oversight for real-world Web-agent training and deployment will be important.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions and scope of SCOPE. The theoretical discussion and experiments on WebArena, WebVoyager, and WebShop support the claimed improvements in long-horizon Web-agent training.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a separate Limitations section. It discusses the current focus on text-based Web observations and the possible invalidation of learned skills under changing dynamic Web contexts.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical assumptions and formal proofs are provided in Appendix D. The appendix includes the proof of the effective-prefix lower bound and the proof of Proposition 1.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The Method and Experiments sections describe the proposed framework, benchmarks, baselines, evaluation metrics, and main experimental protocols. We also set the decoding temperature to 0 during inference to ensure deterministic and reproducible evaluation. Additional theoretical and qualitative details are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will provide anonymized code and data in the Appendix, together with instructions for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The Experiments section specifies the evaluated benchmarks, model backbones, baselines, metrics, and held-out test setting used for the reported comparisons. Further implementation and analysis details are included in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Most main comparisons use a single training run and evaluation. Appendix H additionally reports three repeated WebVoyager evaluations, with sample standard deviations of run-level success rates.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Experiments section includes a cost analysis in terms of environment interactions and model calls per unique successful task. Figure 3(e) reports the resulting training-exploration efficiency frontier.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and believe the research conforms to it. The work evaluates Web agents on established benchmarks and does not involve human-subject experiments or the release of high-risk data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Broader Impacts section discusses risks that can arise when Web-agent exploration and stronger trained agents operate in realistic Web environments. It also notes the importance of monitoring, control mechanisms, task-level permissions, and human oversight for real-world Web-agent training and deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk pretrained models, scraped datasets, or other artifacts that require special responsible-release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing benchmarks, models, and related methods used in the paper are properly cited, including WebArena, WebVoyager, and WebShop. We follow the corresponding benchmark and model terms of use and do not redistribute third-party assets beyond the intended experimental setting.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset, benchmark, or model asset requiring separate asset documentation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing experiments or research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a core component of the research methodology. The Method and Experiments sections describe their use for Web-agent policies, progressive exploration, memory-guided intervention, and skill distillation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.