WHEN SHOULD AGENTS THINK? ADAPTIVE REASONING VIA CROSS-TURN ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based agents have demonstrated strong capabilities on complex tasks. They typically perform reasoning before each action throughout an interaction trajectory. However, reasoning may not be necessary at every turn, as reasoning produced earlier can continue to support subsequent actions. A key challenge is therefore to determine when existing reasoning remains sufficient and when a new reasoning step is needed, without relying on costly generation-based verification. We find that decreases in the likelihood of subsequent reference actions after removing additional reasoning closely track whether those actions remain recoverable given earlier reasoning, providing an effective and lightweight signal for estimating cross-turn action support. Based on this observation, we propose **R**easoning **A**daptation through **C**ross-Turn **E**stimation (**RACE**), a training approach for adaptive agent reasoning. RACE introduces a **L**ikeli**h**ood-**G**uided Progressive Reason**i**ng **C**over Detection **(LoGiC)** procedure that progressively identifies reasoning turns whose removal has limited impact on the current and subsequent reference actions. The resulting removal signals are incorporated into both supervised fine-tuning and agentic reinforcement learning, enabling the policy to learn when to reason and when to act directly. Extensive experiments on four representative agent benchmarks show that RACE substantially reduces reasoning cost while maintaining or improving task performance.

1 INTRODUCTION

Recent advances in Large Reasoning Models (LRMs) (OpenAI, 2024; DeepSeek-AI, 2025; Qwen Team, 2025) have significantly improved the capabilities of agents on complex tasks that require sustained planning and execution across multiple interaction turns (Wang et al., 2025; Li et al., 2026b; Jin et al., 2026; Li et al., 2026a; Tan et al., 2026). In these tasks, agents make sequential decisions by incorporating historical actions and environment feedback into their evolving context. Under this paradigm, many existing agents explicitly generate reasoning steps before action decisions (Yao et al., 2023; Li et al., 2025; Cheng et al., 2026b; Zheng et al., 2025), implicitly treating reasoning as necessary at every interaction turn and incurring substantial inference cost over long trajectories (Chen et al., 2025; Yang et al., 2026a; Chen et al., 2026a).

However, reasoning may not be confined to the interaction turn in which it is generated. Our analysis shows that earlier reasoning can support subsequent actions over varying ranges: some reasoning steps remain useful for only one subsequent turn, while others continue to support actions across multiple turns, as shown in Figure 1. Such persistence may be explained by the fact that previously generated plans, constraints, or intermediate judgments can remain useful beyond their original turns (Ning et al., 2026) and continue to support subsequent actions. We refer to this as cross-turn reasoning persistence. Under this perspective, we think agents do not need to generate reasoning at every turn when previous reasoning remains sufficient to support later actions. A new reasoning step is only required when changes in the interaction context make previous reasoning insufficient.

This perspective aligns with recent studies on *adaptive reasoning* (Zhang et al., 2025; Yang et al., 2026a; Lee et al., 2026a; Ning et al., 2026), which seek to reduce inference cost by determining whether reasoning is necessary at each interaction turn. To obtain turn-level removal signals, they typically compare actions generated with and without a reasoning step (Yang et al., 2026b). However, these approaches do not explicitly account for the varying cross-turn persistence of reasoning,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

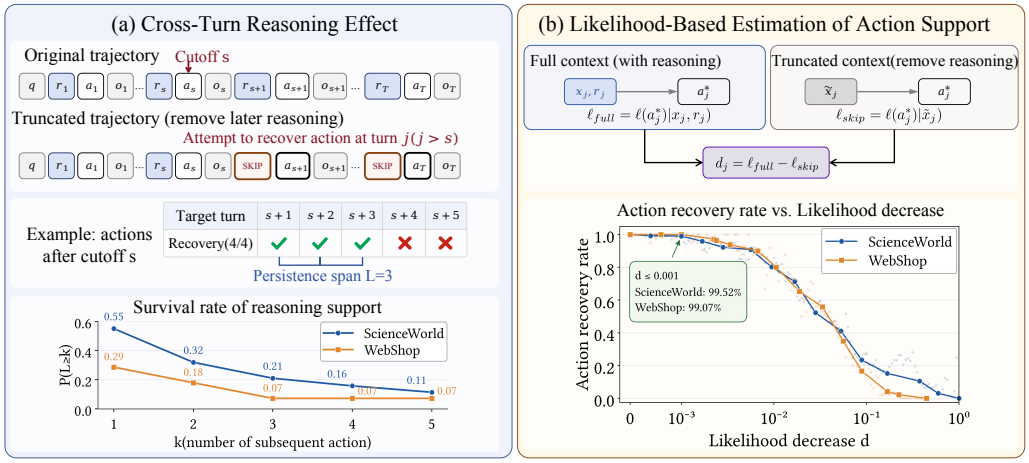


Figure 1: Analysis of cross-turn reasoning support. (a) Reference-action recovery at subsequent turns after removing later reasoning. (b) Decreases in the likelihood of subsequent reference actions closely correspond to the recovery rate, providing a lightweight proxy for action recoverability.

with reasoning necessity typically assessed independently at each turn. Moreover, obtaining the removal signals requires repeated generations for candidate reasoning turns, increasing the cost of supervised fine-tuning data synthesis as trajectories grow longer (Lee et al., 2026b). This challenge becomes more pronounced in on-policy reinforcement learning, where generation-based verification must be repeatedly performed at every interaction turn to obtain local turn-level signals for newly sampled trajectories as the policy evolves, introducing substantial computational overhead and limiting scalability. Therefore, the key to reducing inference cost is not independently determining whether reasoning steps can be removed, but estimating how far existing reasoning remains sufficient and when a new reasoning step becomes necessary.

In this paper, we investigate this problem and seek a lightweight alternative to turn-level verification of reasoning necessity. We find that decreases in the likelihood of subsequent reference actions after removing additional reasoning closely track whether those actions remain recoverable given earlier reasoning. This observation motivates using likelihood decrease of subsequent actions as a lightweight signal for evaluating reasoning removal from existing trajectories, without repeatedly generating candidate actions for verification. Therefore, we propose **RACE** (**R**easoning **A**daptation through **C**ross-Turn **E**stimation), a training approach that learns adaptive reasoning behavior by estimating the cross-turn impact of reasoning removal. The core of RACE is a **L**ikeli**h**ood-**G**uided Progressive Reasoning **C**over Detection **(LoGiC)** procedure that uses reference-action likelihood decrease as a lightweight signal for reasoning removal. In detail, LoGiC assesses each candidate reasoning by measuring the effect of its removal on the likelihood of the current and subsequent reference actions under the resulting context. A candidate is removed only when the decrease remains within a predefined threshold across all affected actions. The retained reasoning steps collectively form a *reasoning cover* that provides sufficient support for actions throughout the trajectory. Unlike generation-based turn-level verification, LoGiC requires only likelihood computation for existing reference actions, enabling efficient derivation of turn-level reasoning removal signals for training.

RACE further incorporates these signals into both supervised fine-tuning (SFT) and agentic reinforcement learning (RL). During the SFT stage, LoGiC transforms full-reasoning trajectories into adaptive reasoning trajectories, teaching the model when it can act directly without additional reasoning. Building on this initialization, RACE introduces cover-aware agentic RL, where LoGiC is applied to successful trajectories sampled from the current policy to derive their reasoning removal signals. The signals are converted into reasoning-closure supervision, providing a direct learning signal for when to terminate reasoning. In this way, the task reward optimizes task performance, while LoGiC-derived reasoning removal signals shape adaptive reasoning behavior.

We evaluate RACE on four representative agent benchmarks spanning scientific reasoning, web shopping, application use, and knowledge-intensive reasoning. Compared with standard SFT fol-

lowed by RL, RACE maintains comparable task performance while reducing reasoning tokens by 32.3%, 80.2%, 46.4%, and 62.0% on ScienceWorld, WebShop, AppWorld, and DeepSearch, validating its ability to learn effective adaptive reasoning behavior.

Our main contributions are summarized as follows:

• We characterize the cross-turn effect of agent reasoning, showing that earlier reasoning can continue to support subsequent actions over varying ranges, and that reference-action likelihood decrease provides a lightweight proxy for estimating this support.

• We propose **RACE**, a training approach for adaptive agent reasoning that leverages likelihood-guided cross-turn estimation to derive reasoning removal signals and incorporates them into both supervised fine-tuning and agentic reinforcement learning.

• Extensive experiments on four representative agent benchmarks show that RACE substantially reduces reasoning cost while maintaining or improving task performance.

2 PRELIMINARIES

Before presenting our method, we first establish the problem formulation, measure how far existing reasoning can support subsequent actions through repeated action generation, and further seek a lightweight signal that does not require repeated action generation.

2.1 PROBLEM FORMULATION

To formalize our study of agent reasoning, we consider an agent completing a task q over T decision turns. At each turn t, it takes action a_t and receives observation o_t, with optional reasoning r_t generated before the action. Let $x_t = (q, r_1, a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ denote the context before turn t. Rather than requiring reasoning before every action, we consider two execution modes under the policy π_θ. At turn t, the agent may generate reasoning and then take an action,

$$r_t \sim \pi_\theta(\cdot \mid x_t), \qquad a_t \sim \pi_\theta(\cdot \mid x_t, r_t), \tag{1}$$

or act directly without generating additional reasoning at the current turn,

$$a_t \sim \pi_\theta(\cdot \mid x_t). \tag{2}$$

Our objective is to optimize π_θ to selectively invoke reasoning across decision turns while still preserving task performance during agent execution.

2.2 MEASURING CROSS-TURN SUPPORT THROUGH GENERATION

To directly measure how far existing reasoning can support subsequent actions, we analyze successful trajectories generated by Qwen3.5-9B on ScienceWorld and WebShop. Given a successful full-reasoning trajectory $\tau = (q, (r_t, a_t^*, o_t)_{t=1}^T)$, where a_t^* is the reference action at turn t, we select turn s as the cutoff, retain all reasoning through turn s, and remove all **reasoning** steps after turn s while keeping the original actions and observations unchanged.

For each turn $j > s$, we denote the resulting truncated context as $\tilde{x}_j$, generate K actions from this context, and compute the reference-action recovery rate:

$$\mathrm{Rec}_j = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\left[\hat{a}_j^{(k)} \simeq a_j^*\right], \qquad \hat{a}_j^{(k)} \sim \pi_\theta(\cdot \mid \tilde{x}_j). \tag{3}$$

Here, $K = 4$, and $\simeq$ denotes canonical action equivalence. As shown in Figure 1(a), reasoning from earlier turns can continue to support later actions, allowing subsequent actions to be recovered without additional reasoning. However, the effective range of this support varies: some reasoning supports only the immediate action, while others remain useful across multiple subsequent turns.

While reference-action recovery rate provides a direct measure of such support, it requires full autoregressive decoding for every candidate context. This makes repeated verification costly for large-scale data synthesis and especially for on-policy reinforcement learning.

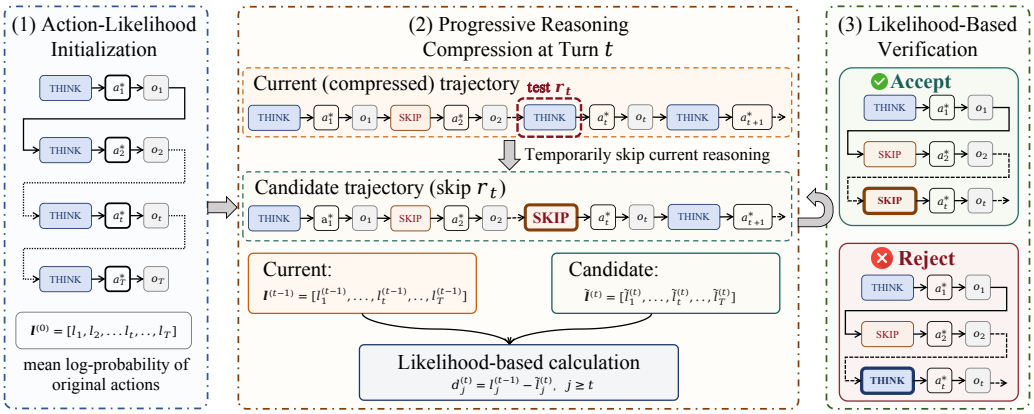


Figure 2: Overview of the Likelihood-Guided Progressive Reasoning Cover Detection (LoGiC). (1) Reference-action likelihoods are initialized on the full trajectory. Starting from the second turn, (2) LoGiC tests whether the reasoning can be removed, and (3) verifies the removal based on likelihood decrease, with each accepted removal updating the context used to evaluate subsequent turns.

2.3 LIKELIHOOD-BASED ESTIMATION OF ACTION SUPPORT

Generation-based recovery rate directly measures whether subsequent reference actions remain recoverable when only earlier reasoning is retained, while this behavior may also be reflected in their likelihoods. If a subsequent action remains recoverable given earlier reasoning, its reference-action likelihood should remain relatively stable; otherwise, it should decrease. We therefore examine the relationship between likelihood decrease and generation-based recovery rate.

For a reference action a_j^* with M_j tokens, we define its mean token log-likelihood under context h:

$$\ell(a_j^* \mid h) = \frac{1}{M_j} \sum_{m=1}^{M_j} \log \pi_\theta \left(a_{j,m}^* \mid h, a_{j,<m}^*\right). \tag{4}$$

For each truncated context $\tilde{x}_j$ and the corresponding complete context (x_j, r_j), we define the decrease in the reference action likelihood as

$$d_j = \ell(a_j^* \mid (x_j, r_j)) - \ell(a_j^* \mid \tilde{x}_j). \tag{5}$$

As shown in Figure 1(b), reference-action likelihood decrease closely corresponds to reference-action recovery rate. Small decreases generally correspond to reliable recovery, while larger decreases are associated with lower recovery rates. This suggests that likelihood decrease can serve as a lightweight proxy for subsequent action recoverability given earlier reasoning. Unlike action generation, it only requires batched teacher-forced scoring of reference actions.

3 METHODOLOGY

In this section, we introduce RACE, a training approach that enables agents to adaptively determine when to reason and when to act directly. An overview of RACE is illustrated in Figure 3.

- **Likelihood-Guided Progressive Reasoning Cover Detection** (§3.1): We use reference-action likelihood decrease to identify removable reasoning and construct a reasoning cover.
- **Agent Reasoning Synthesis for Supervised Fine-Tuning** (§3.2): Based on the identified reasoning cover, we synthesize adaptive reasoning trajectories for supervised fine-tuning.
- **Reasoning-Cover-Aware Agentic Reinforcement Learning** (§3.3): We further develop a cover-aware agentic RL approach that masks removable reasoning from policy-gradient optimization and provides reasoning-closure supervision on successful rollout trajectories.

3.1 LIKELIHOOD-GUIDED PROGRESSIVE REASONING COVER DETECTION

Motivated by the correspondence between likelihood decrease and action recovery observed in §2.3, we use reference-action likelihood decrease to evaluate the effect of reasoning removals and progressively construct a reasoning cover. As shown in Figure 2, the procedure consists of three steps:

(1) Action-Likelihood Initialization. Given an agent trajectory $\tau = (q, (r_t, a_t, o_t)_{t=1}^{T})$, we treat each action a_t as a fixed reference action, denoted by a_t^*, for likelihood evaluation under its original full-reasoning context. Following Equation 4, we initialize the action-likelihood vector as

$$\boldsymbol{\ell}^{(0)} = [\ell(a_1^* \mid x_1, r_1), \ldots, \ell(a_T^* \mid x_T, r_T)]\,. \tag{6}$$

This vector serves as the reference for subsequent reasoning-cover construction.

(2) Progressive Reasoning Compression. To assess the effect of removing a reasoning step under the current compressed context, we construct a candidate trajectory without that reasoning and recompute the likelihoods of the current and subsequent reference actions for comparison.

Specifically, we evaluate each reasoning step sequentially starting from the second turn and carry accepted removals forward to subsequent turns. Let $x^{(t)}$ denote the context before turn t after applying all reasoning removals accepted at preceding turns. In particular, $x^{(1)} = (q)$ and $x^{(2)} = \big(x^{(1)}, r_1, a_1^*, o_1\big)$. At turn $t \in \{2, \ldots, T\}$, we remove r_t to construct a candidate trajectory and compute the corresponding candidate action-likelihood vector as

$$\widetilde{\boldsymbol{\ell}}^{(t)} = \left[\widetilde{\ell}_1^{(t)}, \ldots, \widetilde{\ell}_T^{(t)}\right], \tag{7}$$

where

$$\widetilde{\ell}_j^{(t)} = \begin{cases} \ell_j^{(t-1)}, & j < t, \\ \ell(a_t^* \mid x^{(t)}), & j = t, \\ \ell(a_j^* \mid x^{(t)}, a_t^*, o_t, (r_u, a_u^*, o_u)_{u=t+1}^{j-1}, r_j), & j > t. \end{cases} \tag{8}$$

In the case of $j > t$ above, the candidate trajectory differs from the current trajectory only by the removal of r_t, while all subsequent reasoning, actions, and observations remain unchanged.

(3) Likelihood-Based Verification. To determine whether a candidate reasoning step can be safely removed under the current compressed context, we measure its effect on the likelihoods of the current and subsequent reference actions. Specifically, for each reference action a_j^* for $j \in \{t, \ldots, T\}$, we compute the corresponding likelihood decrease as

$$d_j^{(t)} = \ell_j^{(t-1)} - \widetilde{\ell}_j^{(t)}. \tag{9}$$

We accept the removal of r_t only if the decrease remains within the threshold ϵ consistently for all current and subsequent reference actions throughout the entire trajectory:

$$d_j^{(t)} \le \epsilon, \qquad \forall j \in \{t, \ldots, T\}. \tag{10}$$

The likelihood vector and context are then updated according to the verification result:

$$\left(\boldsymbol{\ell}^{(t)}, x^{(t+1)}\right) = \begin{cases} \left(\widetilde{\boldsymbol{\ell}}^{(t)}, \big(x^{(t)}, a_t^*, o_t\big)\right), & \text{if accepted,} \\ \left(\boldsymbol{\ell}^{(t-1)}, \big(x^{(t)}, r_t, a_t^*, o_t\big)\right), & \text{otherwise.} \end{cases} \tag{11}$$

After all turns have been examined, the retained reasoning steps constitute the reasoning cover of τ.

3.2 AGENT REASONING SYNTHESIS FOR SUPERVISED FINE-TUNING

To enable agents to adaptively determine when to reason and when to act directly, we use the reasoning covers identified by the Likelihood-Guided Progressive Reasoning Cover Detection (LoGiC) algorithm in §3.1 to synthesize adaptive reasoning trajectories for supervised fine-tuning.

LoGiC-Based Data Synthesis. Given an agent trajectory τ, we apply the LoGiC algorithm to identify reasoning that can be removed and obtain the corresponding adaptive agent reasoning trajectory $\tilde{\tau} = \mathrm{LoGiC}(\tau)$. The resulting trajectories constitute the SFT training data $\mathcal{D}_{\mathrm{SFT}}$.

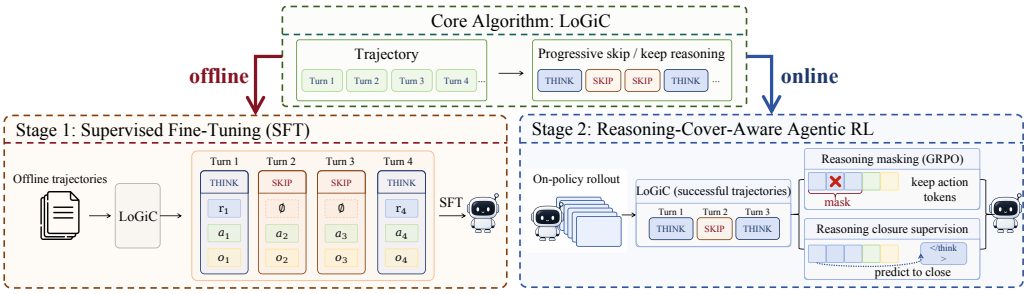


Figure 3: Overview of our training approach RACE. RACE uses the LoGiC algorithm to identify reasoning steps that can be safely removed, and the resulting signals are used in both supervised fine-tuning and agentic reinforcement learning to learn adaptive reasoning behavior.

Supervised Fine-Tuning. We fine-tune the agent on $\mathcal{D}_{\mathrm{SFT}}$ using supervised learning:

$$\mathcal{L}_{\mathrm{SFT}} = -\mathbb{E}_{\tilde{\tau}\sim\mathcal{D}_{\mathrm{SFT}}}\left[\sum_{n=1}^{|\tilde{\tau}|}\log \pi_\theta\left(\tilde{\tau}_n \mid \tilde{\tau}_{<n}\right)\right]. \tag{12}$$

The loss is applied only to model-generated reasoning and action tokens. This stage provides direct supervision for learning when to generate reasoning and when to act directly, serving as an initialization for subsequent agentic reinforcement learning.

3.3 REASONING-COVER-AWARE AGENTIC REINFORCEMENT LEARNING

Supervised fine-tuning provides an initial adaptive reasoning policy from fixed offline trajectories, which may not match the evolving policy. We therefore incorporate LoGiC into on-policy agentic reinforcement learning. For each rollout batch, LoGiC identifies removable reasoning steps in successful trajectories to mask unnecessary reasoning from the policy-gradient objective and supervise reasoning termination. Failed trajectories follow standard GRPO.

LoGiC-Based Reasoning Masking. We first prevent GRPO from reinforcing reasoning that LoGiC identifies as unnecessary. Specifically, for each successful on-policy trajectory, LoGiC derives a set of SKIP turns S_i. We mask the reasoning tokens at these turns from the GRPO objective while retaining their action tokens for task optimization. Let $\mathcal{P}_i$ denote all policy-token positions in τ_i, and $\mathcal{K}_i \subseteq \mathcal{P}_i$ denote those remaining after masking turns in S_i. The masked GRPO loss is

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{\sum_{i\in\mathcal{B}}|\mathcal{P}_i|}\sum_{i\in\mathcal{B}}\sum_{k\in\mathcal{K}_i}\min\left(r_{i,k}(\theta)\widehat{A}_i,\ \mathrm{clip}\left(r_{i,k}(\theta),1-\epsilon_{\mathrm{clip}},1+\epsilon_{\mathrm{clip}}\right)\widehat{A}_i\right). \tag{13}$$

Here, $r_{i,k}(\theta)$ is the importance ratio and $\widehat{A}_i$ the group-relative advantage. Masked reasoning tokens are excluded from the policy gradient while their original positions remain in the normalization.

LoGiC-Based Reasoning Closure. Masking prevents redundant reasoning from being reinforced, but does not directly encourage the policy to terminate reasoning at these turns. We therefore introduce reasoning-closure supervision on LoGiC-verified SKIP turns. Specifically, let e denote the reasoning termination token, $\mathcal{I}_{\mathrm{succ}}$ denote the successful trajectories, S_i denote their LoGiC-verified SKIP turns, and $x_i^{(t)}$ denote the corresponding compressed context. The reasoning-closure loss is

$$\mathcal{L}_{\mathrm{close}} = -\frac{1}{\sum_{i\in\mathcal{B}}T_i}\sum_{i\in\mathcal{I}_{\mathrm{succ}}}\sum_{t\in S_i}\log\pi_\theta\left(e\mid x_i^{(t)}\right). \tag{14}$$

where T_i is the number of interaction turns in τ_i. Normalizing by total rollout turns makes closure supervision scale with the prevalence of LoGiC-verified SKIP turns. The final actor objective combines task optimization with LoGiC-Based reasoning closure:

$$\mathcal{L}_{\mathrm{actor}} = \mathcal{L}_{\mathrm{GRPO}} + \lambda\mathcal{L}_{\mathrm{close}}. \tag{15}$$

where λ controls the strength of reasoning-closure supervision.

Table 1: Main results on four agent tasks, including ScienceWorld, WebShop, AppWorld, and DeepSearch. *Avg Tok.* denotes the average number of reasoning tokens generated per trajectory. For models with up to 32B parameters, the best results are in **bold** and the second are underlined.

Method	ScienceWorld		WebShop		AppWorld		DeepSearch	
	Success ↑	Avg Tok.↓	Success ↑	Avg Tok.↓	Success ↑	Avg Tok. ↓	Success ↑	Avg Tok.↓
ReAct-style Iterative Reasoning Agents								
DeepSeek-V4-Pro	32.11	1,662	40.20	1,288	87.69	3,016	40.00	1,243
GLM-5.3	30.29	1,854	46.60	1,715	77.61	17,142	24.00	3,287
GPT-5.5 (Medium)	69.88	793	40.20	799	91.11	1,002	46.25	557
Claude-Opus-4.6 (Medium)	59.37	-	53.40	-	78.80	-	30.00	-
Kimi-K3	45.52	1,633	45.80	1,471	80.00	2,751	44.25	1,679
Qwen3.8-27B	38.37	2,011	35.60	1,059	58.46	4,425	33.50	2,623
Qwen3.5-9B	18.52	2,210	15.60	2,713	46.15	3,134	29.25	1,281
Reasoning-Efficient Agents (Qwen3.5-9B)								
Think Tool	19.63	2,426	14.00	2,154	43.59	2,943	25.75	987
ADaPT	14.24	5,512	31.80	3,720	21.20	4,166	25.75	1,797
Sparse ReAct	19.85	2,126	16.40	2,426	45.47	3,187	26.00	2,228
DART	17.92	1,413	18.20	1,791	49.06	2,345	29.00	1,316
AdaptThink	64.49	**1,252**	44.60	748	**68.72**	2,381	33.50	1,481
RACE-SFT	43.05	1,966	45.40	1,232	63.08	3,330	28.25	1,856
RACE-SFT-RL	**73.28**	1,403	**53.40**	**535**	65.13	**1,900**	**35.00**	**822**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets and Evaluation. We evaluate RACE on four representative agent benchmarks: (1) **Scientific Reasoning**: ScienceWorld (Wang et al., 2022) evaluates science tasks in simulated environments requiring sequential reasoning and actions. (2) **Web Shopping**: WebShop (Yao et al., 2022) evaluates product search and selection based on user requirements. (3) **Application Use**: AppWorld (Trivedi et al., 2024) evaluates tasks through interactions with simulated apps and APIs. (4) **Knowledge-Intensive Reasoning**: DeepSearch (Jin et al., 2025) evaluates knowledge-intensive question answering through retrieval and reasoning. We report success rate and average reasoning tokens per trajectory. Dataset and implementation details are in Appendices B and D.

Baselines. We compare RACE with the following agent reasoning baselines: **(1) ReAct-style Iterative Reasoning Agents.** We evaluate ReAct (Yao et al., 2023)-style reasoning agents across a range of representative open-source and closed-source models, including DeepSeek-V4-Pro (DeepSeek-AI, 2026), GLM-5.3 (GLM, 2026), GPT-5.5 (OpenAI, 2026), Claude-Opus-4.6 (Anthropic, 2026), Kimi-K3 (Kimi Team, 2026), and Qwen3.8 series (Qwen Team, 2026b). **(2) Reasoning-Efficient Agents.** We further compare with Think Tool (Anthropic, 2025), ADaPT (Prasad et al., 2024), Sparse ReAct (Yao et al., 2023), DART (Lee et al., 2026a), and AdaptThink (Zhang et al., 2025). All reasoning-efficiency methods are evaluated with Qwen3.5-9B (Qwen Team, 2026a) as the backbone to ensure a fair comparison. Detailed descriptions of these baselines are provided in Appendix C.

4.2 MAIN RESULTS

Table 1 presents the main results across four representative agent benchmarks, evaluating both task performance and reasoning cost. We summarize the key findings below.

Competitive Performance against Strong LLM Agents. RACE achieves competitive performance against substantially larger and proprietary LLM agents, with success rates of 73.28, 53.40, 65.13, and 35.00 on ScienceWorld, WebShop, AppWorld, and DeepSearch, respectively. Interestingly, several LLM agents sometimes take actions without additional explicit reasoning, suggesting that

Table 2: Ablation studies of RACE. The best results are in **bold**.

Method	ScienceWorld		WebShop		AppWorld		DeepSearch	
	Success ↑	Avg Tok. ↓	Success ↑	Avg Tok. ↓	Success ↑	Avg Tok.↓	Success ↑	Avg Tok.↓
RACE-SFT-RL	73.28	1,403	**53.40**	535	65.13	**1,900**	35.00	**822**
w/o Reasoning Masking in RL	72.95	**609**	46.00	**124**	56.75	3,161	32.00	2,196
w/o LoGiC in RL (*RACE-SFT+GRPO*)	**75.54**	2,573	51.40	2,670	**67.18**	8,574	**35.75**	3,507
w/o LoGiC-RL (*RACE-SFT*)	43.05	1,966	45.40	1,232	63.08	3,330	28.25	1,856
w/o LoGiC (*SFT+GRPO*)	74.55	2,073	52.00	2,702	65.30	3,545	**35.75**	2,163

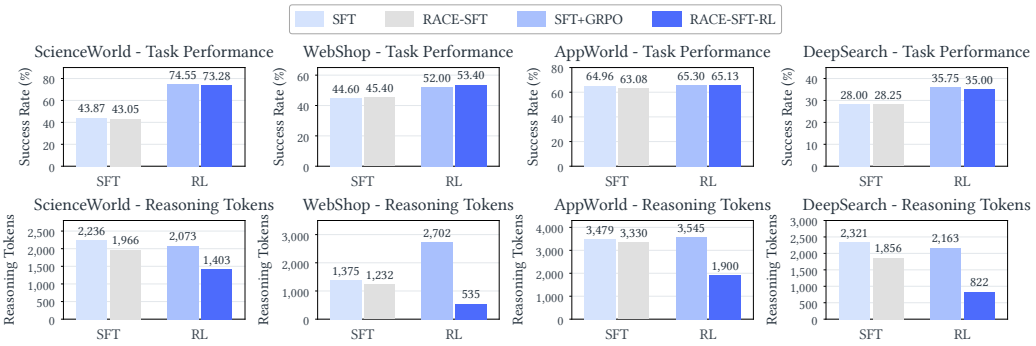


Figure 4: Comparison of task performance and reasoning efficiency with and without LoGiC.

such adaptive behavior can emerge in capable agents. RACE learns this behavior through training, skipping reasoning when appropriate while maintaining performance across diverse agent tasks.

Superior Performance and Reasoning Efficiency. RACE achieves a strong performance-efficiency trade-off across four agent benchmarks. Compared with Qwen3.5-9B, RACE-SFT-RL improves success by 54.76, 37.80, 18.98, and 5.75 points while reducing reasoning tokens by 36.5%, 80.3%, 39.4%, and 35.8% on ScienceWorld, WebShop, AppWorld, and DeepSearch, respectively. Among same-backbone baselines, RACE-SFT-RL performs best on ScienceWorld, WebShop, and DeepSearch, while using fewer tokens than AdaptThink on WebShop, AppWorld, and DeepSearch.

Further Gains from Cover-Aware Reinforcement Learning. Cover-aware RL further improves upon RACE-SFT. Compared with RACE-SFT, RACE-SFT-RL increases success by 30.23, 8.00, 2.05, and 6.75 points on ScienceWorld, WebShop, AppWorld, and DeepSearch, respectively, while reducing reasoning tokens by 28.6%, 56.6%, 42.9%, and 55.7%. These simultaneous gains show that LoGiC serves as an effective proxy for deriving turn-level reasoning removal signals during policy optimization without repeated generation-based verification.

4.3 Quantitative Analysis

Ablation Studies. We examine the effect of LoGiC-derived signals in SFT and RL. As shown in Figure 4, RACE-SFT stays within 1.88 points of standard SFT, while reducing reasoning tokens by 4.3% to 20.0%. The efficiency gains are larger during RL. Compared with standard SFT+GRPO, RACE-SFT-RL stays within 1.40 points while reducing reasoning tokens by 32.3%, 80.2%, 46.4%, and 62.0% on ScienceWorld, WebShop, AppWorld, and DeepSearch, respectively. Table 2 shows that removing LoGiC from RL sharply increases reasoning cost, from 1,403 to 2,573 tokens on ScienceWorld, 535 to 2,670 on WebShop, 1,900 to 8,574 on AppWorld, and 822 to 3,507 on DeepSearch. Removing reasoning masking also reduces success rate on WebShop, AppWorld, and DeepSearch, while its effect on reasoning cost varies across benchmarks.

Scaling Analysis on the Qwen3.5 Series. We evaluate RACE across Qwen3.5-2B, 4B, and 9B. Overall, performance improves with model scale, and RACE yields gains on ScienceWorld, WebShop, and DeepSearch. On AppWorld, RACE is less effective at smaller scales but reaches 65.13% with 9B, outperforming Base and RACE-SFT. Further scaling analysis is provided in Appendix A.1.

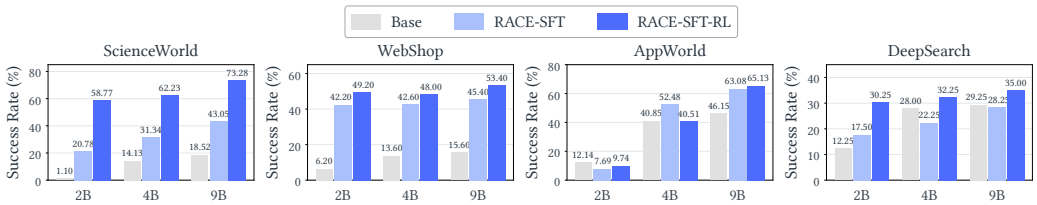


Figure 5: Scaling analysis across Qwen3.5-2B, 4B, and 9B on four agent benchmarks, comparing the base models without training, RACE-SFT, and RACE-SFT-RL.

Adaptive Reasoning Behavior Analysis. To understand how RACE reduces reasoning cost, we examine the skip ratio and average reasoning tokens over thinking turns. As shown in Table 3, RL increases the skip ratio across all four benchmarks. Meanwhile, reasoning tokens per thinking turn increase on ScienceWorld and AppWorld but decrease on WebShop and DeepSearch. This suggests that RACE does not uniformly shorten reasoning. Instead, it skips reasoning at more turns, while retained reasoning adapts to task characteristics. In state-intensive tasks such as ScienceWorld and AppWorld, reasoning concentrates on more demanding interaction turns, whereas WebShop and DeepSearch favor more concise reasoning. Case studies are provided in Appendix G.

Table 3: Analysis of adaptive reasoning behavior. *Skip* denotes the proportion of turns that directly execute actions without generating reasoning, while *Tok./Think* denotes the average reasoning tokens over reasoning turns.

Benchmark	**RACE-SFT**		**RACE-SFT-RL**	
	Skip (%)	Tok./Think	Skip (%)	Tok./Think
ScienceWorld	37.22	132	69.65	273
WebShop	7.15	144	55.72	101
AppWorld	7.74	273	52.15	302
DeepSearch	1.79	259	5.77	117

5 RELATED WORK

Reasoning-Efficient Methods. Prior studies have explored reducing unnecessary reasoning and improving reasoning efficiency in language models and agents (Wei et al., 2026). One line reduces agent reasoning cost through more efficient planning, execution, or direct efficiency optimization (Xu et al., 2023; Hu et al., 2024; Chen et al., 2026b; 2025). Another line studies adaptive reasoning more broadly, dynamically determining when or how much reasoning to perform (Lewis-Lim et al., 2025; Zhang et al., 2025; Fang et al., 2025; Lou et al., 2025; Wu et al., 2025; Lu et al., 2025; Su et al., 2026; Liang et al., 2025; Yang et al., 2026a; Ning et al., 2026; Yang et al., 2026b).

Agentic Reinforcement Learning. Reinforcement learning (RL) (Schulman et al., 2017; Shao et al., 2024; Rafailov et al., 2023) is widely used to optimize agents from task-level feedback. Recent work has explored effective policy optimization (Dong et al., 2025a), rollout exploration (Dong et al., 2025b; He et al., 2026a), and credit assignment (Feng et al., 2025; He et al., 2026b; Cheng et al., 2026a) for agent tasks. While these methods mainly focus on task performance, less attention has been paid to adaptive reasoning behavior during interaction.

6 CONCLUSION

In this work, we present Reasoning Adaptation through Cross-Turn Estimation (RACE), a training approach for adaptive reasoning in LLM-based agents. Our analysis shows that earlier reasoning can support subsequent actions over varying ranges, while reference-action likelihood decrease provides a lightweight signal for estimating this support. Based on these findings, RACE introduces a Likelihood-Guided Progressive Reasoning Cover Detection (LoGiC) procedure to identify removable reasoning steps and use the resulting signals in both supervised fine-tuning and agentic reinforcement learning. Across four representative agent benchmarks, RACE substantially reduces reasoning cost while maintaining or improving task performance, demonstrating the effectiveness of likelihood-guided cross-turn estimation for learning efficient adaptive reasoning behavior.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, including improving readability, refining technical descriptions, suggesting paper organization, and retrieving and discovering relevant work. We have reviewed all AI-assisted work. We manually reviewed all AI-assisted content and verified that the revised text accurately reflects the authors' intended contributions and technical details. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the complete workflow of the Likelihood-Guided Progressive Reasoning Cover Detection (LoGiC) in Appendix F. The datasets used in our experiments are described in Appendix B, and the evaluation protocol and training configurations, including hyperparameters for SFT and RL, are provided in Appendices D.1 and D.2, respectively. The complete task-specific prompt templates for all four benchmarks are included in Appendix E.

REFERENCES

Anthropic. The "think" tool: Enabling claude to stop and think in complex tool use situations. https://www.anthropic.com/engineering/claude-think-tool, 2025. Accessed: 2026-09-25.

Anthropic. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026. Accessed: 2026-09-07.

Qianben Chen, Tianrui Qin, King Zhu, Qiexiang Wang, Chengjun Yu, Shu Xu, Jiaqi Wu, Jiayu Zhang, Xinpeng Liu, Xin Gui, Jingyi Cao, Piaohong Wang, Dingfeng Shi, He Zhu, Tiannan Wang, Yuqing Wang, Maojia Song, Tianyu Zheng, Ge Zhang, Jian Yang, Jiaheng Liu, Minghao Liu, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Search more, think less: Rethinking long-horizon agentic search for efficiency and generalization. *CoRR*, abs/2602.22675, 2026a. doi: 10.48550/ARXIV.2602.22675. URL https://doi.org/10.48550/arXiv.2602.22675.

Sirui Chen, Mengshi Zhao, Lei Xu, Yuying Zhao, Beier Zhu, Hanwang Zhang, Shengjie Zhao, and Chaochao Lu. DEPO: dual-efficiency preference optimization for LLM agents. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 30279–30287. AAAI Press, 2026b. doi: 10.1609/AAAI.V40I36.40279. URL https://doi.org/10.1609/aaai.v40i36.40279.

Yifei Chen, Guanting Dong, and Zhicheng Dou. Toward effective tool-integrated reasoning via self-evolved preference learning. *CoRR*, abs/2509.23285, 2025. doi: 10.48550/ARXIV.2509.23285. URL https://doi.org/10.48550/arXiv.2509.23285.

Xin Cheng, Shuo He, Lang Feng, Haiyang Xu, Ming Yan, Lei Feng, and Bo An. Beyond trajectory-level attribution: Graph-based credit assignment for agentic reinforcement learning. *CoRR*, abs/2605.26684, 2026a. doi: 10.48550/ARXIV.2605.26684. URL https://doi.org/10.48550/arXiv.2605.26684.

Yiruo Cheng, Kelong Mao, Tianhao Li, Jiejun Tan, Ji-Rong Wen, and Zhicheng Dou. Chatshop-buddy: Towards reliable conversational shopping agents via reinforcement learning. *CoRR*, abs/2603.06065, 2026b. doi: 10.48550/ARXIV.2603.06065. URL https://doi.org/10.48550/arXiv.2603.06065.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. *CoRR*, abs/2606.19348, 2026. doi: 10.48550/ARXIV.2606.19348. URL https://doi.org/10.48550/arXiv.2606.19348.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization. *CoRR*, abs/2510.14545, 2025a. doi: 10.48550/ARXIV.2510.14545. URL https://doi.org/10.48550/arXiv.2510.14545.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. *CoRR*, abs/2507.19849, 2025b. doi: 10.48550/ARXIV.2507.19849. URL https://doi.org/10.48550/arXiv.2507.19849.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/de2ad3ed44ee4e675b3be42aa0b615d0-Abstract-Conference.html.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

GLM. GLM-5: from vibe coding to agentic engineering. *CoRR*, abs/2602.15763, 2026. doi: 10.48550/ARXIV.2602.15763. URL https://doi.org/10.48550/arXiv.2602.15763.

Bowei He, Yankai Chen, Xiaokun Zhang, and Xue Liu. Branching policy optimization: Sandbox-native language agent reinforcement learning. *CoRR*, abs/2607.14171, 2026a. doi: 10.48550/ARXIV.2607.14171. URL https://doi.org/10.48550/arXiv.2607.14171.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. *CoRR*, abs/2602.22817, 2026b. doi: 10.48550/ARXIV.2602.22817. URL https://doi.org/10.48550/arXiv.2602.22817.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Núria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.580. URL https://doi.org/10.18653/v1/2020.coling-main.580.

Mengkang Hu, Yao Mu, Xinmiao Yu, Mingyu Ding, Shiguang Wu, Wenqi Shao, Qiguang Chen, Bin Wang, Yu Qiao, and Ping Luo. Tree-planner: Efficient close-loop task planning with large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Glcsog6zOe.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Jiajie Jin, Yuyang Hu, Kai Qiu, Qi Dai, Chong Luo, Guanting Dong, Xiaoxi Li, Tong Zhao, Xiaolong Ma, Gongrui Zhang, Zhirong Wu, Bei Liu, Zhengyuan Yang, Linjie Li, Lijuan Wang, Hongjin Qian, Yutao Zhu, and Zhicheng Dou. Toward generalist autonomous research via hypothesis-tree refinement. *CoRR*, abs/2606.11926, 2026. doi: 10.48550/ARXIV.2606.11926. URL https://doi.org/10.48550/arXiv.2606.11926.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

Kimi Team. Kimi K3: open frontier intelligence. *CoRR*, abs/2607.24653, 2026. doi: 10.48550/ARXIV.2607.24653. URL https://doi.org/10.48550/arXiv.2607.24653.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/TACL_A_00276. URL https://doi.org/10.1162/tacl_a_00276.

Jungseob Lee, Seongtae Hong, Seungjun Lee, Jaehyung Seo, Junyoung Son, Sugyeong Eo, Chanjun Park, Hyeongju Park, Hyeonseok Moon, and Heuiseok Lim. DART: draft-agreement routing for training-free adaptive thinking budgets in hybrid reasoning models. *CoRR*, abs/2606.23181, 2026a. doi: 10.48550/ARXIV.2606.23181. URL https://doi.org/10.48550/arXiv.2606.23181.

Sangmook Lee, Dohyung Kim, Hyukhun Koh, Nakyeong Yang, and Kyomin Jung. Confidence-guided stepwise model routing for cost-efficient reasoning. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 31483–31491. AAAI Press, 2026b. doi: 10.1609/AAAI.V40I37.40413. URL https://doi.org/10.1609/aaai.v40i37.40413.

Samuel Lewis-Lim, Xingwei Tan, Zhixue Zhao, and Nikolaos Aletras. Can confidence estimates decide when chain-of-thought is necessary for llms? *CoRR*, abs/2510.21007, 2025. doi: 10.48550/ARXIV.2510.21007. URL https://doi.org/10.48550/arXiv.2510.21007.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/ae03bdef276132fae089692445725635-Abstract-Conference.html.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, and Zhicheng Dou. Deepagent: A general reasoning agent with scalable toolsets. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz (eds.), *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pp. 2219–2230. ACM, 2026a. doi: 10.1145/3774904.3792460. URL https://doi.org/10.1145/3774904.3792460.

Zhuofeng Li, Dongfu Jiang, Xueguang Ma, Haoxiang Zhang, Ping Nie, Yuyu Zhang, Kai Zou, Jianwen Xie, Yu Zhang, and Wenhu Chen. Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. *CoRR*, abs/2603.20278, 2026b. doi: 10.48550/ARXIV.2603.20278. URL https://doi.org/10.48550/arXiv.2603.20278.

Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. Thinkswitcher: When to think hard, when to think fast. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics:*

EMNLP 2025, Suzhou, China, November 4-9, 2025, pp. 5185–5201. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.278. URL https://doi.org/10.18653/v1/2025.findings-emnlp.278.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *CoRR*, abs/2505.11896, 2025. doi: 10.48550/ARXIV.2505.11896. URL https://doi.org/10.48550/arXiv.2505.11896.

Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, Han Wang, and Can Huang. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient LLM/MLLM reasoning. *CoRR*, abs/2505.15154, 2025. doi: 10.48550/ARXIV.2505.15154. URL https://doi.org/10.48550/arXiv.2505.15154.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.546. URL https://doi.org/10.18653/v1/2023.acl-long.546.

Yansong Ning, Jun Fang, Naiqiang Tan, and Hao Liu. Agent-omit: Training efficient LLM agents for adaptive thought and observation omission via agentic reinforcement learning. *CoRR*, abs/2602.04284, 2026. doi: 10.48550/ARXIV.2602.04284. URL https://doi.org/10.48550/arXiv.2602.04284.

OpenAI. Learning to reason with llms, September 2024. URL https://openai.com/index/learning-to-reason-with-llms/. Accessed: 2026-09-25.

OpenAI. Gpt-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, 2026. Accessed: 2026-09-07.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: As-needed decomposition and planning with language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, volume NAACL 2024 of *Findings of ACL*, pp. 4226–4252. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.264. URL https://doi.org/10.18653/v1/2024.findings-naacl.264.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pp. 5687–5711. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.378. URL https://doi.org/10.18653/v1/2023.findings-emnlp.378.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwen.ai/blog?id=qwq-32b. Accessed: 2026-09-25.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5. Accessed: 2026-09-25.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8. Accessed: 2026-09-25.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural*

Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. Cp-router: An uncertainty-aware router between LLM and LRM. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 33065–33073. AAAI Press, 2026. doi: 10.1609/AAAI.V40I39.40589. URL https://doi.org/10.1609/aaai.v40i39.40589.

Jiejun Tan, Zhicheng Dou, Yan Yu, Jiehan Cheng, Lifeng Liu, Jian Xie, and Jirong Wen. Hiersearch: A hierarchical enterprise deep search framework integrating local and web searches. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 19380–19388. AAAI Press, 2026. doi: 10.1609/AAAI.V40I23.39015. URL https://doi.org/10.1609/aaai.v40i23.39015.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL https://doi.org/10.1162/tacl_a_00475.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.850. URL https://doi.org/10.18653/v1/2024.acl-long.850.

Ruoyao Wang, Peter A. Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 11279–11298. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.775. URL https://doi.org/10.18653/v1/2022.emnlp-main.775.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning. *CoRR*, abs/2504.20073, 2025. doi: 10.48550/ARXIV.2504.20073. URL https://doi.org/10.48550/arXiv.2504.20073.

Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, Zihao Li, Mengting Ai, Duo Zhou, Wenxuan Bao, Yunzhe Li, Gaotang Li, Cheng Qian, Yu Wang, Xiangru Tang, Yin Xiao, Liri Fang, Hui Liu, Xianfeng Tang, Yuji Zhang, Chi Wang, Jiaxuan You, Heng Ji, Hanghang Tong, and Jingrui He. Agentic reasoning for large language models. *CoRR*, abs/2601.12538, 2026. doi: 10.48550/ARXIV.2601.12538. URL https://doi.org/10.48550/arXiv.2601.12538.

Siye Wu, Jian Xie, Yikai Zhang, Aili Chen, Kai Zhang, Yu Su, and Yanghua Xiao. ARM: adaptive reasoning model. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/48c218b01802a712ee8d7243d27fd9cf-Abstract-Conference.html.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. Agentgym-rl: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning. *CoRR*, abs/2509.08755, 2025. doi: 10.48550/ARXIV.2509.08755. URL https://doi.org/10.48550/arXiv.2509.08755.

Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. Rewoo: Decoupling reasoning from observations for efficient augmented language models. *CoRR*, abs/2305.18323, 2023. doi: 10.48550/ARXIV.2305.18323. URL https://doi.org/10.48550/arXiv.2305.18323.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. Ares: Adaptive reasoning effort selection for efficient LLM agents. *CoRR*, abs/2603.07915, 2026a. doi: 10.48550/ARXIV.2603.07915. URL https://doi.org/10.48550/arXiv.2603.07915.

Ruihan Yang, Fanghua Ye, Xiang We, Ruoqing Zhao, Kang Luo, Xinbo Xu, Bo Zhao, Ruotian Ma, Shanyi Wang, Zhaopeng Tu, Xiaolong Li, Deqing Yang, and Linus. Think fast and slow: Step-level cognitive depth adaptation for LLM agents. *CoRR*, abs/2602.12662, 2026b. doi: 10.48550/ARXIV.2602.12662. URL https://doi.org/10.48550/arXiv.2602.12662.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 3716–3730. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.184. URL https://doi.org/10.18653/v1/2025.emnlp-main.184.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng

(eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 414–431. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.22. URL https://doi.org/10.18653/v1/2025.emnlp-main.22.

APPENDIX

A SUPPLEMENTARY EXPERIMENTAL RESULTS **17**

A.1 Further Scaling Analysis of RACE 17

B Tasks and Datasets **18**

B.1 Scientific Reasoning Task 18

B.2 Web Shopping Task 18

B.3 Application Use Task 18

B.4 Deep Search Task 19

C Baseline Details **19**

C.1 ReAct-style Iterative Reasoning Agents 19

C.2 Reasoning-Efficient Agents 19

D Implementation Details **20**

D.1 Evaluation Protocol 20

D.2 Training Details 20

D.2.1 Supervised Fine-Tuning 20

D.2.2 Reinforcement Learning 20

E Prompt Template **21**

F The Algorithm Workflow of the Likelihood-Guided Progressive Reasoning Cover Detection **23**

G Case Study **23**

G.1 Case Study for Scientific Reasoning 23

G.2 Case Study for Web Shopping 23

G.3 Case Study for Application Use 24

G.4 Case Study for Deep Search 24

A SUPPLEMENTARY EXPERIMENTAL RESULTS

A.1 FURTHER SCALING ANALYSIS OF RACE

We further analyze RACE across Qwen3.5-2B, 4B, and 9B on four agent benchmarks. As shown in Figure 6, RACE generally achieves stronger task performance as model scale increases, although the gains vary across benchmarks. On AppWorld, although performance improves substantially with model scale, the gains of RACE-SFT-RL over RACE-SFT vary across model sizes. This may be related to the limited training data available for AppWorld, which makes RL training less stable, especially at smaller model scales. Across model scales, RACE-SFT-RL generally improves task performance over RACE-SFT while also substantially reducing reasoning tokens, with especially large reductions on WebShop. Overall, these results further suggest that the effectiveness of RACE largely persists across different model scales.

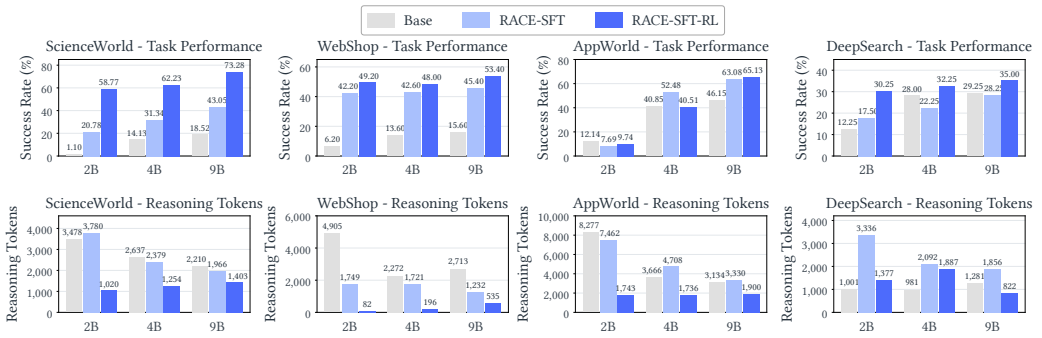


Figure 6: Scaling analysis with Qwen3.5-2B, 4B, and 9B across ScienceWorld, WebShop, AppWorld, and DeepSearch, showing task performance and reasoning efficiency at different model scales under no additional training, RACE-SFT, and RACE-SFT-RL.

B Tasks and Datasets

We evaluate our method on four representative agent tasks, covering scientific reasoning, web shopping, application use, and knowledge-intensive search.

B.1 Scientific Reasoning Task

For scientific reasoning, we use **ScienceWorld** (Wang et al., 2022), a benchmark designed to evaluate agents on elementary-level science tasks in a text-based simulated environment. It comprises 30 tasks across 10 science topics: changes of state, temperature measurement, electrical circuits, friction, object classification, chemical mixtures, plants and pollinators, life spans, life stages, and Mendelian genetics. Each task requires the agent to accomplish a specified scientific goal through a sequence of textual observations and actions. We use the official training split for training and evaluate on the full official test split, which contains 1,819 task instances.

B.2 Web Shopping Task

For web shopping, we use **WebShop** (Yao et al., 2022), which formulates shopping as a goal-directed decision-making task in the simulated e-commerce websites. WebShop contains 1,181,436 real-world products collected from Amazon and 12,087 crowd-sourced natural language instructions. Each instruction specifies requirements for a desired product, such as its attributes, options, and price, and the agent must search for relevant products, browse and compare candidates, select the required options, and complete the purchase. The instructions are officially divided into 10,587 training, 1,000 development, and 500 test instances. We use the official training split for training and evaluate on the full test split of 500 instances.

B.3 Application Use Task

For application use, we adopt **AppWorld** (Trivedi et al., 2024), where agents complete realistic day-to-day tasks by operating simulated applications through APIs. AppWorld provides 9 day-to-day applications with 457 APIs and includes 750 tasks instantiated from 250 task scenarios. These tasks involve multiple applications and require agents to reason over intermediate results and adapt subsequent operations accordingly. Task completion is evaluated using state-based unit tests that verify the resulting application states.

We use both the official Train and Dev splits from the benchmark release for training. While the AppWorld paper reports 105 training tasks and 60 development tasks, the benchmark release used in our experiments contains 90 available Train tasks and 57 available Dev tasks. Therefore, our training set consists of 147 tasks in total. For evaluation, we use the complete Test-N split with 168 tasks and the complete Test-C split with 417 tasks.

B.4 Deep Search Task

For deep search, we follow AgentGym-RL (Xi et al., 2025) and consider queries from seven question answering datasets following the setup of Search-R1 (Jin et al., 2025). These include **Natural Questions (NQ)** (Kwiatkowski et al., 2019), **TriviaQA** (Joshi et al., 2017), and **PopQA** (Mallen et al., 2023) for general question answering, as well as **HotpotQA** (Yang et al., 2018), **2WikiMultiHopQA** (Ho et al., 2020), **MuSiQue** (Trivedi et al., 2022), and **Bamboogle** (Press et al., 2023) for multi-hop question answering. Given a question, the agent can issue search queries to retrieve information and incorporate the retrieved evidence into its reasoning before producing the final answer.

We combine the training sets of NQ, HotpotQA, 2WikiMultiHopQA, and MuSiQue to construct the training data. Following AgentGym-RL, we evaluate on 400 examples randomly sampled from the development sets of seven datasets, covering diverse knowledge-intensive question answering tasks.

C Baseline Details

This section provides detailed descriptions of the baselines used in our experiments.

C.1 ReAct-style Iterative Reasoning Agents

- **DeepSeek-V4-Pro** (DeepSeek-AI, 2026) is a large-scale Mixture-of-Experts (MoE) language model released by the DeepSeek team. It consists of 1.6 trillion total parameters with 49 billion activated parameters and supports long-context processing. The model is designed for complex reasoning, coding, and agentic tasks, providing strong capabilities in handling challenging problems with extended contexts. We use DeepSeek-V4-Pro-0813 in our experiments.
- **GLM-5.3** (GLM, 2026) is an open-source large language model developed by Zhipu AI. It is designed for complex tasks including software engineering, reasoning, and agentic applications. The model supports long-context processing, function calling, and structured outputs, enabling it to handle multi-step workflows and interactive scenarios.
- **GPT-5.5** (OpenAI, 2026) is OpenAI's frontier model for complex knowledge work and real-world problem solving. It provides capabilities in coding, information analysis, and tool use, with support for long-context processing and multi-step task execution across diverse scenarios. We evaluate GPT-5.5 with the reasoning effort set to medium.
- **Claude-Opus-4.6** (Anthropic, 2026) is a frontier model developed by Anthropic. It is designed for complex tasks including software engineering, research, analysis, and agentic workflows. The model supports extended thinking and long-context reasoning, enabling multi-step problem solving. We use Claude-Opus-4.6 in our experiments with the thinking level set to medium.
- **Kimi-K3** (Kimi Team, 2026) is an open-weight native multimodal agentic model developed by Moonshot AI. It is a 2.8-trillion-parameter model built on Kimi Delta Attention (KDA) and Attention Residuals (AttnRes), with native vision capabilities and a 1-million-token context window. The model is designed for long-horizon coding, knowledge work, and reasoning tasks.
- **Qwen3.8 series** (Qwen Team, 2026b) is an open-source large language model series developed by the Qwen team of Alibaba. It provides capabilities in coding, professional work, research, and long-horizon agentic tasks. The series supports flexible thinking modes and agent execution, enabling the models to handle complex multi-step tasks. The series includes models of different scales to support diverse computational requirements and application scenarios. We use Qwen3.8-27B in our experiments.

C.2 Reasoning-Efficient Agents

- **Sparse ReAct** (Yao et al., 2023) is a prompting strategy based on the ReAct paradigm that reduces unnecessary reasoning generation during interaction. It allows the model to skip reasoning at some turns and directly execute actions when reasoning is not needed, while preserving the original ReAct interaction process between reasoning, actions, and observations.
- **ADaPT** (Prasad et al., 2024) is an approach for as-needed decomposition and planning in LLM-based agents. It adaptively performs task decomposition only when the executor cannot directly complete the current sub-task, avoiding unnecessary planning for simpler tasks. The approach

employs a planner-executor framework and recursively decomposes complex tasks when needed. By adjusting the planning process according to task complexity, ADaPT reduces redundant decomposition and planning during agent execution.

- **Think Tool** (Anthropic, 2025) provides the agent with an auxiliary no-op tool for internal deliberation. The agent can autonomously decide whether to invoke this tool before taking subsequent actions. We implement this mechanism through prompting without additional training.
- **DART** (Lee et al., 2026a) adaptively selects between thinking and non-thinking modes based on the agreement of multiple no-think drafts. Since DART was originally designed for standalone query-level reasoning, we adapt it to our multi-step agent scenarios by sampling multiple no-thinking tool-call drafts at each interaction turn and using their agreement as a signal for deciding whether to invoke reasoning. This adaptation applies DART's decision process at the turn level, where the agent determines whether to generate reasoning before executing each action.
- **AdaptThink** (Zhang et al., 2025) learns to allocate reasoning by allowing models to choose between thinking and non-thinking modes. We extend this idea to interactive agents by making the reasoning-mode choice at every decision turn, where the agent either generates reasoning before an action or directly executes the action with an empty think block. Following the original formulation, we train the agent with GRPO using a modified reward $r(\tau) = R(\tau) + \delta \cdot \frac{1}{|\tau|} \sum_{t=1}^{|\tau|} \mathbb{I}(\mathrm{NoThink}_t)$, where the second term measures the proportion of non-thinking turns. The bonus term is applied only to successful trajectories, and we set $\delta = 0.05$ while keeping the remaining GRPO settings unchanged.

D Implementation Details

D.1 Evaluation Protocol

We evaluate on ScienceWorld, WebShop, AppWorld, and DeepSearch. Across all benchmarks, we cap each episode at 30 interaction turns and allow up to 65,536 tokens for each model generation. We use a temperature of 1.0, top-p of 1.0, and a maximum context length of 262,144 tokens, with reasoning enabled and native tool calling. ScienceWorld, WebShop, and DeepSearch allow at most one tool call per interaction turn, whereas AppWorld supports multiple API calls within a turn using precomputed and audited task-specific API lists. We report task success rate and the average total number of reasoning tokens per trajectory, measuring task performance and reasoning cost.

For ScienceWorld, we evaluate on 1,819 test instances spanning 30 task types, where an instance is considered successful only when the final environment score reaches 100. For WebShop, we use the official test split of 500 crowd-sourced instructions and consider a task successful when the environment reward reaches 1. For AppWorld, we evaluate on both Test-Normal and Test-Challenge and report the overall strict task success over all 585 tasks. DeepSearch is evaluated on 400 questions using Exact Match (EM).

D.2 Training Details

D.2.1 Supervised Fine-Tuning

We construct the SFT data from successful trajectories annotated by DeepSeek-V4-Pro-0813. Specifically, we apply LoGiC with Qwen3.5-9B to these trajectories and set the likelihood-decrease threshold to $\epsilon = 0.001$ to identify reasoning steps that can be removed. The resulting adaptive reasoning trajectories are then used to fine-tune Qwen3.5-9B. For ScienceWorld, WebShop, and DeepSearch, we set the learning rate to 1×10^{-5} and the effective batch size to 32. Since AppWorld provides substantially fewer training trajectories, we use a smaller learning rate of 5×10^{-6} and an effective batch size of 8. The maximum training sequence length is 65,536 tokens.

D.2.2 Reinforcement Learning

We initialize RL from the corresponding RACE-SFT checkpoint for each benchmark. We use a batch size of 16 and sample 8 trajectories per instance, yielding 128 trajectories per rollout batch. The learning rate is set to 1×10^{-6}, with one policy update epoch per rollout batch and a clipping

ratio of 0.2. Rollouts are sampled with a temperature of 1.0 and top-p of 1.0, with a maximum of 30 interaction turns. LoGiC is applied only to successful on-policy trajectories during training, with the likelihood-decrease threshold set to $\epsilon = 0.001$. We set the closure-loss weight to $\lambda = 0.02$ for ScienceWorld, WebShop, and AppWorld, and $\lambda = 0.01$ for DeepSearch.

E PROMPT TEMPLATE

We use task-specific prompt templates for the four benchmarks, following their respective interaction protocols and tool interfaces. The full prompt templates are shown below.

Prompt Template for ScienceWorld

You are an agent for science world. Every round I will give you an observation, and you must call exactly one of the provided functions to act toward finishing the given task. Only use the provided functions; the objects you choose must exist in the current room. If the environment returns "No known action matches that input", your previous action was invalid and you should try another option.

Prompt Template for WebShop

You are web shopping. I will give you instructions about what to do. Every round I will give you an observation and a list of available actions. Use the available tools to choose one valid action for the current page. Use search only if the search bar is available. Use click only with an exact value from the page clickables. If the page remains unchanged, it might indicate that your action is invalid.

Prompt Template for DeepSearch

Answer the question using your own knowledge and information returned by the search tool.
Use search only when necessary, call it at most once per turn, and avoid repeating equivalent searches. Once you have enough information, stop searching.
Only give me the final answer and do not output any other words. The final answer should be the shortest possible answer span, without explanations, Markdown, citations, or extra details.

Prompt Template for AppWorld

I am your supervisor, and you are an AI Assistant whose job is to complete my day-to-day tasks fully autonomously.
My name is: `{full_name}`. My personal email is `{email}` and phone number is `{phone_number}`.
You will be given a task instruction and a list of functions in the standard format. The functions correspond to APIs from various apps you have access to. The function name has two parts, the app name and API name separated by "__", e.g., `spotify__login` is the login API for the Spotify app.
You will complete the task completely autonomously through multi-turn interaction with the execution environment. In each turn, you will make one or more function calls, and the environment will return its outputs. This will continue either until you call the `complete_task` API from the Supervisor app, or until a maximum of `{max_interactions}` turns are reached. Here are brief app-wise descriptions. `{app_descriptions}`
Key Instructions:
A. General instructions:

- Act fully on your own. You must make all decisions yourself and never ask me or anyone else to confirm or clarify. Your role is to solve the task, not to bounce questions back, or provide me directions to follow.
- You have full access – complete permission to operate across my connected accounts and services.
- Never invent or guess values. For example, if I ask you to play a song, do not assume the ID is 123. Instead, look it up properly through the right API.
- Never leave placeholders; don't output things like `your_username`. Always fill in the real value by retrieving it via APIs (e.g., Supervisor app for credentials).
- When I omit details, choose any valid value. For example, if I ask you to buy something but don't specify which payment card to use, you may pick any one of my available cards.
- Avoid collateral damage. Only perform what I explicitly ask for. Example: if I ask you to buy something, do not delete emails, return the order, or perform unrelated account operations.
- You only have `{max_interactions}` turns. Avoid unnecessary requests. You can batch unlimited function calls in a single turn – always group them to save steps.

B. App-specific instructions:

- All my personal information (biographical details, credentials, addresses, cards) is stored in the Supervisor app, accessible via its APIs.
- Any reference to my friends, family or any other person or relation refers to the people in my phone's contacts list.
- Always obtain current date or time, from the phone app's `get_current_date_and_time` API, never from your internal clock.
- All requests are concerning a single, default (no) time zone.
- For temporal requests, use proper time boundaries, e.g., when asked about periods like "yesterday", use complete ranges: `00:00:00` to `23:59:59`.
- References to "file system" mean the file system app, not the machine's OS. Do not use OS modules or functions.
- Paginated APIs: Always process all results, looping through the `page_index`. Don't stop at the first page.

C. Task-completion instructions:
You must call the `supervisor__complete_task` API after completing the task.

- The task is doable, but if you cannot find a way, you can call it with `status="fail"` to exit with failure.

When the answer is given:

- Keep answers minimal. Return only the entity, number, or direct value requested – not full sentences. E.g., for the song title of the current playing track, return just the title.
- Numbers must be numeric and not in words. E.g., for the number of songs in the queue, return "10", not "ten".

F THE ALGORITHM WORKFLOW OF THE LIKELIHOOD-GUIDED PROGRESSIVE REASONING COVER DETECTION

In this section, we present the detailed procedure of the Likelihood-Guided Progressive Reasoning Cover Detection (LoGiC) algorithm, as illustrated in Algorithm 1.

Algorithm 1 Likelihood-Guided Progressive Reasoning Cover Detection (LoGiC)

Input trajectory $\tau = (q, (r_t, a_t, o_t)_{t=1}^{T})$; reference actions $\{a_t^*\}_{t=1}^{T}$; likelihood function $\ell(\cdot)$; threshold ϵ

Initialize action-likelihood vector $\boldsymbol{\ell}^{(0)} = [\ell(a_1^*|x_1, r_1), ..., \ell(a_T^*|x_T, r_T)]$
Initialize compressed context $x^{(1)} \leftarrow q$, $x^{(2)} \leftarrow (x^{(1)}, r_1, a_1^*, o_1)$
Initialize reasoning cover $\mathcal{C} \leftarrow \{r_1\}$
Initialize compressed trajectory $\tilde{\tau} \leftarrow (q, r_1, a_1^*, o_1)$
for $t = 2, \ldots, T$ **do**
    Remove r_t from the trajectory while keeping previous accepted removals, and construct the candidate trajectory: $(x^{(t)}, a_t^*, o_t, r_{t+1}, a_{t+1}^*, o_{t+1}, ..., r_T, a_T^*, o_T)$
    Compute candidate likelihood vector under the candidate trajectory: $\widetilde{\boldsymbol{\ell}}^{(t)} = [\widetilde{\ell}_1^{(t)}, ..., \widetilde{\ell}_T^{(t)}]$
    Compute likelihood decrease: $d_j^{(t)} = \ell_j^{(t-1)} - \widetilde{\ell}_j^{(t)}, \quad j \in \{t, \ldots, T\}$
    if $d_j^{(t)} \leq \epsilon,\ \forall j \in \{t, \ldots, T\}$ **then**
        Accept the removal of r_t
        Update likelihood vector: $\boldsymbol{\ell}^{(t)} \leftarrow \widetilde{\boldsymbol{\ell}}^{(t)}$
        Update compressed context: $x^{(t+1)} \leftarrow (x^{(t)}, a_t^*, o_t)$
        Update compressed trajectory: $\tilde{\tau} \leftarrow (\tilde{\tau}, a_t^*, o_t)$
    else
        Retain r_t
        Update likelihood vector: $\boldsymbol{\ell}^{(t)} \leftarrow \boldsymbol{\ell}^{(t-1)}$
        Update compressed context: $x^{(t+1)} \leftarrow (x^{(t)}, r_t, a_t^*, o_t)$
        Update compressed trajectory: $\tilde{\tau} \leftarrow (\tilde{\tau}, r_t, a_t^*, o_t)$
        Add r_t into reasoning cover $\mathcal{C}$
    end if
end for
Output compressed trajectory $\tilde{\tau}$ and reasoning cover $\mathcal{C}$

G CASE STUDY

The examples in Tables 4, 5, 6, and 7 illustrate the adaptive reasoning behaviors learned through RACE across diverse agent tasks, including scientific reasoning, web shopping, application use, and deep search. Across these scenarios, the trained policy generates reasoning when needed while acting directly when the existing context already provides sufficient support for subsequent actions.

G.1 CASE STUDY FOR SCIENTIFIC REASONING

Table 4 presents a ScienceWorld trajectory that requires combining red and blue paint to obtain violet paint. The policy model trained with RACE initially uses reasoning to infer the required composition and plan the manipulation sequence. Its first action, however, attempts to pick up the liquid directly and is rejected by the environment, revealing that the planned operation is infeasible.

The returned feedback triggers a short revision of the action strategy: instead of handling the liquid directly, the model switches to pouring it into the glass cup. Once this correction is established and the first pour succeeds, the remainder of the task becomes straightforward. The policy directly pours the blue paint, mixes the contents, and focuses on the resulting violet paint. This trajectory highlights how RACE training concentrates reasoning on planning and recovery from execution failure, while allowing the policy to carry out the remaining supported actions without additional reasoning.

G.2 CASE STUDY FOR WEB SHOPPING

Table 5 presents a WebShop trajectory in which a policy model trained with RACE adaptively allocates reasoning during product search and verification. The initial reasoning consolidates the user's

constraints into a targeted query, leading to a candidate product that matches the requested item and price range. The policy then proceeds through the product page and description directly, since these navigation steps require little additional deliberation.

The need for reasoning emerges again only when the description fails to establish whether the product is individually wrapped. This missing attribute prompts the policy to inspect the feature page, where the remaining requirements are confirmed. With the product now fully verified, it returns to the product page and completes the purchase without further reasoning. The trajectory demonstrates how RACE training supports selective reasoning around uncertain product attributes while keeping routine browsing and selection steps concise.

G.3 Case Study for Application Use

Table 6 presents an AppWorld example that requires coordinated use of the file system, phone, and Gmail. The policy begins by decomposing the request into several subgoals: locating the birth certificate, finding the husband's contact information, and sending the document as an email attachment. With this workflow established, it carries out a sequence of routine operations, including account login and file retrieval, without generating reasoning at every turn.

Reasoning reappears when the task moves to a new subgoal that requires additional information or planning. After locating the target file, the policy recognizes that the husband's contact information is still needed and searches for the corresponding contact. After retrieving the email address, it reasons again to formulate the final email action and attach the correct birth certificate file. The remaining sending and task-completion steps are then executed directly without further reasoning. This trajectory highlights how RACE training leads the policy to concentrate reasoning around meaningful subgoal transitions, while avoiding unnecessary reasoning during routine operations whose next actions are already clear from the current context.

G.4 Case Study for Deep Search

Table 7 presents a DeepSearch example that requires identifying two film directors and comparing their birth dates. The policy first decomposes the question into the necessary information: the directors of *Reaching for the Sun* and *Trifling Women*, followed by their respective birth dates. Once this search structure is established, it retrieves the director of each film, skipping additional reasoning for the second lookup when the required action is already clear.

After both directors are identified, reasoning is invoked again to determine the remaining evidence needed for the comparison. The policy then retrieves their birth dates, directly issuing the second birth-date query after obtaining the first result. Finally, it combines the collected evidence and concludes that Rex Ingram, the director of *Trifling Women*, was born earlier than William A. Wellman. The trajectory shows that RACE training encourages reasoning at points that require query decomposition or evidence synthesis, while allowing straightforward retrieval steps to proceed directly.

Table 4: A scientific reasoning example on ScienceWorld illustrating the adaptive reasoning behavior enabled by RACE across different interaction turns. [...] indicates omitted content. Special symbols highlight reasoning (<think>), tool calls (<tool call>), and tool results (<tool response>).

Example from ScienceWorld Dataset

Task Instruction

Your task is to use chemistry to create violet paint. When you are done, focus on the violet paint.

Initial Observation

This room is called the art studio. In it, you see:
a substance called air
a large cupboard. The large cupboard door is closed.
a table. On the table is: a glass cup (containing nothing).
a wood cup (containing red paint)
a wood cup (containing blue paint)
a wood cup (containing yellow paint)
You also see:
A door to the hallway (that is closed)

Model Output by RACE (9B)

<think> We are in art studio. Need to create violet paint using chemistry. Violet is likely mixture of blue and red. We have red and blue paint. Need to combine to create violet. Maybe need proper containers. There is a glass cup and wood cups. Need to mix red and blue paints. The task says “use chemistry to create violet paint. When you are done, focus on the violet paint.” We need to use tool. Perhaps open? But first maybe move red paint and blue paint into glass cup, then mix. The red and blue are in wood cups. We can pickup? “a wood cup (containing red paint)”. The object is “red paint”? Maybe “pick up red paint”? The action “pickup” moves object to inventory. Then use on glass cup? Actually “mix” mixes contents of container. We can move blue to glass cup, red to glass cup, then mix. But can we pick up paint? Likely “pickup red paint” then “pickup blue paint”, then maybe “move red paint to glass cup”? Inventory has both. Then move each to glass cup? Then mix. ... We can call “pickup red paint” to get red paint in inventory. Then “pickup blue paint”. But “red paint” maybe “wood cup”? Science word: objects are visible. “a glass cup (containing nothing).” “a wood cup (containing red paint)”. We can call "pickup red paint"? If object not directly “red paint”? It should be fine. Let’s start by picking red paint. </think>
<tool call> pickup (obj="red paint") </tool call>
<tool response> You can’t pick up a liquid directly. Try pouring it from one container to another, or dunking empty containers into containers filled with the liquid. </tool response>
<think> Need pour from wood cup into glass cup. We can “pour red paint from wood cup to glass cup”? The tool “pour” likely “liq”: “red paint”, “container”: “glass cup”. But maybe need source? The L: “pour a liquid into a container.” So “pour liquid into glass.” They may require “pour red paint into glass cup”? We can use “pour” with liq=“red paint”, container=“glass cup”. But if red paint in wood cup, pours from wood cup? Science word maybe pours from wherever. The object available is “red paint”? Maybe it’s “a wood cup (containing red paint)”. The liquid is red paint. We can pour red paint into glass cup. Let’s try. </think>
<tool call> pour(liq="red paint", container="glass cup") </tool call>
<tool response> You pour the red paint into the glass cup. </tool response>
<think> </think>
<tool call> pour(liq="blue paint", container="glass cup") </tool call>
<tool response> You pour the blue paint into the glass cup. </tool response>
<think> </think>
<tool call> mix(container="glass cup") </tool call>
<tool response> Red and blue paint mix to produce violet paint. </tool response>
<think> </think>
<tool call> focus(obj="violet paint") </tool call>
<tool response> You focus on the violet paint. </tool response>

Table 5: A web shopping example on WebShop illustrating the adaptive reasoning behavior enabled by RACE across different interaction turns, including product search, browsing, and selection. [...] indicates omitted content. Special symbols highlight reasoning (<think>), tool calls (<tool call>), tool results (<tool response>), and clickable web elements (gray box).

Example from WebShop Dataset

Task Instruction

I like traditional, old and individually wrapped albert's chocolate ice cubes 60 count tray chocolate, and price lower than 60.00 dollars

Model Output by RACE (9B)

<think> The user wants to search for a product. Let me construct a search query based on the instruction. I'll search for "Albert's Chocolate Ice Cubes 60 count tray traditional old individually wrapped". </think>
<tool call> search (keywords="Albert's Chocolate Ice Cubes 60 count tray individually wrapped") </tool call>
<tool response>
Page 1 (Total results: 50) Next >
B00T7JEX44 Albert's Chocolate Ice Cubes 60 Count Tray $29.88
B00QOX3I6U Albert's Chocolate Ice Cubes 50 Count $19.91
B00A4BRAE4 Albert's Chocolate Ice Cubes 100 Count Tray $29.62 ...
</tool response>
<think> </think>
<tool call> click (item="B00T7JEX44") </tool call>
<tool response> Back to Search < Prev Albert's Chocolate Ice Cubes 60 Count Tray, Price: $29.88, Rating: N.A. Description Features Reviews Buy Now </tool response>
<think> </think>
<tool call> click (item="Description") </tool call>
<tool response> Back to Search < Prev Milk Chocolate Ice Cubes are the type of candy that chocolate lovers dream about. Because they are made from the highest quality ingredients, they are very perishable and have become quite hard to find. </tool response>
<think> </think>
<tool call> click (item="< prev") </tool call>
<tool response> Back to Search < Prev Albert's Chocolate Ice Cubes 60 Count Tray, Price: $29.88, Rating: N.A. Description Features Reviews Buy Now </tool response>
<think> The description page doesn't mention individually wrapped. Let me check features. </think>
<tool call> click (item="Features") </tool call>
<tool response> Back to Search < Prev Alberts Chocolate Ice Cubes - each container contains 60, 0.4 ounce, individually wrapped chocolate ice cubes. This is a traditional and old time favorite of many so order while you still can. These chocolate cubes are great for any candy dish or candy buffet and are sure to excite anyone looking for some old fashioned candy. </tool response>
<think> </think>
<tool call> click (item="< prev") </tool call>
<tool response> Back to Search < Prev Albert's Chocolate Ice Cubes 60 Count Tray, Price: $29.88, Rating: N.A. Description Features Reviews Buy Now </tool response>
<think> </think>
<tool call> click (item="Buy Now") </tool call>
<tool response> Thank you for shopping with us! </tool response>

Table 6: An application use example on AppWorld illustrating the adaptive reasoning behavior enabled by RACE across different interaction turns. [...] indicates omitted content. Special symbols highlight reasoning (<think>), tool calls (<tool call>), and tool results (<tool response>).

Example from AppWorld Dataset

Task Instruction

Email the birth certificate found in my file system to my husband.

User Profile

My name is: Kristin White. My personal email is kri-powe@gmail.com and phone number is 6017026518.

Model Output by RACE (9B)

<think> I need to email the birth certificate found in my file system to my husband. Let me start by understanding my accounts and finding the birth certificate file, then identify my husband's contact, and draft/send the email. First, I need credentials. Let me get supervisor profile and passwords, and maybe current phone/-time. Let me call supervisor_show_account_passwords and supervisor_show_profile. Also, maybe phone login requires phone number and password. I have personal email and phone number given: email kri-powe@gmail.com, phone 6017026518. Let me get credentials. Let me call supervisor APIs. </think>

<tool call>
supervisor.show_account_passwords()
supervisor.show_profile()
</tool call>

<tool response>
[
  {"account_name": "amazon", "password": "$AMAZON_PASSWORD"},
  {"account_name": "file_system", "password": "$FILE_SYSTEM_PASSWORD"},
  {"account_name": "gmail", "password": "$GMAIL_PASSWORD"},
  {"account_name": "phone", "password": "$PHONE_PASSWORD"},
  {"account_name": "simple_note", "password": "$SIMPLE_NOTE_PASSWORD"},
  {"account_name": "splitwise", "password": "$SPLITWISE_PASSWORD"},
  {"account_name": "spotify", "password": "$SPOTIFY_PASSWORD"},
  {"account_name": "todoist", "password": "$TODOIST_PASSWORD"},
  {"account_name": "venmo", "password": "$VENMO_PASSWORD"}
]
</tool response>

<tool response>
{
  "birthday": "1987-05-31",
  "email": "kri-powe@gmail.com",
  "first_name": "Kristin",
  "last_name": "White",
  "phone_number": "6017026518",
  "sex": "female"
}
</tool response>

<think> </think>

<tool call>
file_system.login (username="kri-powe@gmail.com", password=FILE_SYSTEM_PASSWORD)
phone.login (username="6017026518", password=PHONE_PASSWORD)
gmail.login (username="kri-powe@gmail.com", password=GMAIL_PASSWORD)
</tool call>

<tool response>
{ "access_token": "$FILE_SYSTEM_TOKEN", "token_type": "Bearer" }
</tool response>

<tool response>
{ "access_token": "$PHONE_TOKEN", "token_type": "Bearer" }
</tool response>

<tool response>
{ "access_token": "$GMAIL_TOKEN", "token_type": "Bearer" }
</tool response>

Example from AppWorld Dataset (continued)

<think> </think>
<tool call>
file_system.show_directory(directory_path="/", access_token=FILE_SYSTEM_TOKEN, substring="birth")
</tool call>
<tool response>
"/home/kristin/documents/personal/identity/birth_certificate.pdf"
</tool response>
<think> Good, found birth certificate pdf. Now need to identify husband contact. Let me search contacts with relationship “husband” or query “husband”. </think>
<tool call>
phone.search_contacts (relationship="husband", access_token=PHONE_TOKEN, page_limit=20)
</tool call>
<tool response>
[{
"birthday": "1981-10-14",
"contact_id": 540,
"created_at": "2021-10-18T16:24:13",
"email": "alexwhite@gmail.com",
"first_name": "Alex",
"home_address": "0031 Jeremy Cliff Suite 552\nSeattle\nWashington\nUnited States\n72731",
"last_name": "White",
"phone_number": "3707865891",
"relationships": ["partner", "husband"],
"work_address": "406 Lindsay Courts Suite 522\nSeattle\nWashington\nUnited States\n26281"
}]
</tool response>
<think> Husband email is alexwhite@gmail.com. Now need to draft/send email with birth certificate attachment. Let me create draft with recipient alexwhite@gmail.com, subject maybe “Birth Certificate”, body describing the document, and attach the file. The attachment file path is absolute: /home/kristin/documents/personal/identity/birth_certificate.pdf. Need file_system_access_token and gmail access_token. Let me create draft with attachment_file_paths. The gmail create_draft has optional scheduled_send_at etc, but we can just create draft and then send from draft. Let me create draft with attachment. </think>
<tool call>
gmail.create_draft(access_token=GMAIL_TOKEN, recipient_email_addresses=["alexwhite@gmail.com"], subject="Birth Certificate", body="Here is my birth certificate.", attachment_file_paths=["/home/kristin/documents/personal/identity/birth_certificate.pdf"], file_system_access_token=FILE_SYSTEM_TOKEN)
</tool call>
<tool response>
{ "draft_id": 518, "message": "Draft created." }
</tool response>
<think> </think>
<tool call>
gmail.send_email_from_draft(draft_id=518, access_token=GMAIL_TOKEN, file_system_access_token=FILE_SYSTEM_TOKEN)
</tool call>
<tool response>
{ "message": "Sent email from your draft.", "sent_email_id": 20867, "sent_email_thread_id": 47820 }
</tool response>
<think> </think>
<tool call>
supervisor.complete_task(status="success")
</tool call>
<tool response>
{ "message": "Marked the active task complete." }
</tool response>

Table 7: A deep search example on DeepSearch illustrating the adaptive reasoning behavior enabled by RACE across different interaction turns, including query decomposition, evidence collection, and answer synthesis. [...] indicates omitted content. Special symbols highlight reasoning (blue box), tool calls (green box), and tool results (yellow box), and the final answer (red box).

Example from DeepSearch Dataset

Task Instruction

Which film has the director who was born first, Reaching For The Sun or Trifling Women?

Model Output by RACE (9B)

<think> We need answer question. Need identify directors of “Reaching For The Sun” and “Trifling Women”, their birth dates, compare. Need search. Use search tool. </think>
<tool call> search(query="Reaching For The Sun film director born") </tool call>
<tool response>
Doc 1(Title: "Reaching for the Sun") Reaching for the Sun Reaching for the Sun is a 1941 American comedy film directed by William A. Wellman and written by W.L. River. The film stars Joel McCrea, Ellen Drew, Eddie Bracken, Albert Dekker, Billy Gilbert, George Chandler and Bodil Ann Rosing...
</tool response>
<think> </think>
<tool call> search(query=’"Trifling Women" film director’) </tool call>
<tool response>
Doc 1(Title: "Trifling Women") Trifling Women Trifling Women is a 1922 American silent romantic drama film directed by Rex Ingram. It is credited with boosting the careers of its leads, Barbara La Marr and Ramon Novarro. It has been described as Ingram’s most personal film...
</tool response>
<think> Need comparison of directors’ birthdates. William A. Wellman? Rex Ingram? Need know “Reaching For the Sun” director William A. Wellman birth date. Let’s search. Also need Rex Ingram birth date. Search. </think>
<tool call> search(query="William A. Wellman director birth date") </tool call>
<tool response>
Doc 1(Title: "William A. Wellman") William A. Wellman William Augustus Wellman (February 29, 1896 – December 9, 1975) was an American film director notable for his work in crime, adventure and action genre films, often focusing on aviation themes, a particular passion. He also directed several well-regarded satirical comedies... </tool response>
<think> </think>
<tool call> search(query="Rex Ingram director born birth date") </tool call>
<tool response>
Doc 1(Title: "Rex Ingram (director)") Rex Ingram (director) Rex Ingram (15 January 1892 – 21 July 1950) was an Irish film director, producer, writer and actor...
</tool response>
<think> Reaching For the Sun director William A. Wellman born Feb 29, 1896. Trifling Women director Rex Ingram born Jan 15, 1892. Question "Which film has the director who was born first" -> Trifling Women (1892 < 1896). Final concise. </think>
<answer
Trifling Women
</answer>