STOP OR CONTINUE? COUNTERFACTUAL TIMING SUPERVISION FOR SEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn search agents must learn not only what to search for, but also when further interaction is worth its cost. Poor timing leads to two opposite errors: stopping prematurely with incomplete evidence, or continuing to search after sufficient evidence has already been acquired. Direct supervision of this timing decision requires a counterfactual comparison between the utility of continuing and that of stopping from the same decision context, whereas a standard rollout reveals only the outcome of the chosen action. We introduce **Action-Timing**, a cost-aware reinforcement learning framework with an explicit `STOP`/`CONTINUE` timing decision. Using main-trajectory branches together with matched same-action and counterfactual rollouts organized in a knowledge-state graph, Action-Timing estimates the *Marginal Utility of Continuation* (MUC), defined as the difference between the expected cost-adjusted downstream utilities of continuing and stopping at that decision point. MUC directly supervises whether further interaction is worthwhile, while continuation plans receive credit relative to the same stopping reference, separating learning when to continue from learning how to continue. Experiments on search-agent benchmarks show improved task performance over competing methods. Trajectory analyses further illustrate bidirectional timing correction, showing that Action-Timing can continue beyond premature stopping points to recover correct answers while terminating redundant searches earlier without sacrificing answer correctness.

1 INTRODUCTION

Large language models (LLMs) increasingly act as search agents that acquire external knowledge through multi-turn interaction. Retrieval-augmented generation (RAG) grounds responses in retrieved evidence (Lewis et al., 2020), while subsequent approaches interleave retrieval, reasoning, and external actions (Trivedi et al., 2023; Jiang et al., 2023; Yao et al., 2023). Reinforcement learning (RL) further enables agents to learn query formulation and search behavior from interaction and outcome feedback (Jin et al., 2025; Zheng et al., 2025b). These advances make information acquisition a learned, sequential process.

A central decision in this process is whether to continue searching or stop and answer. As illustrated in Figure 1, poor timing can cause an agent to answer prematurely with incomplete evidence or search unnecessarily despite having evidence that supports an answer. However, the presence of supporting evidence does not always justify immediate stopping, since additional verification may still be valuable. The value of continuing therefore depends on what the agent can achieve through subsequent search plans and whether that benefit justifies the additional interaction cost.

Learning this trade-off requires more than observing whether a completed trajectory succeeded. Recent approaches improve agentic RL through step-level advantages (Feng et al., 2025), process and information-gain rewards (Li et al., 2026; Hu et al., 2026), and structure-aware or tree-based learning signals (Zhu et al., 2026; Ji et al., 2026). These learning signals do not by themselves establish whether continuing is preferable to stopping from the same decision context. An incorrect answer after stopping does not reveal whether further search could have recovered a correct answer; a successful continuation does not establish whether its additional interactions were worthwhile. Timing supervision therefore calls for a comparison of the downstream utilities of both choices.

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.

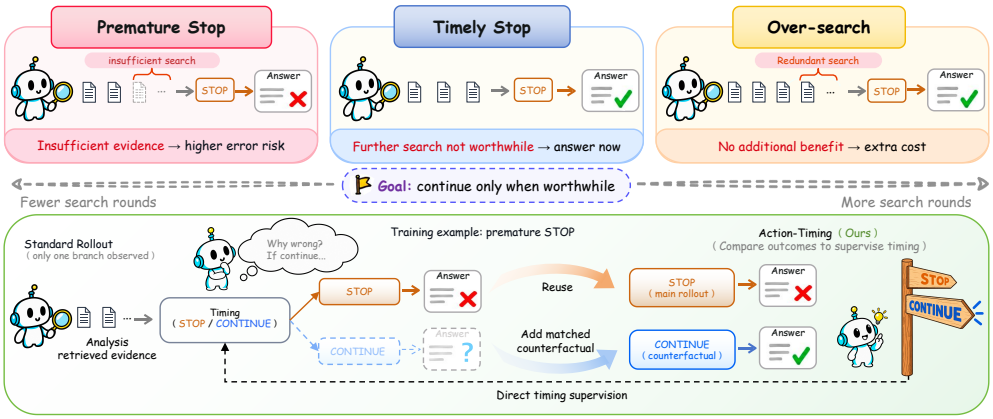


Figure 1: **The action-timing problem and counterfactual supervision. Top:** Effective timing balances premature stopping against redundant search, continuing only when further interaction justifies its cost. **Bottom:** The main rollout chooses `STOP`, leaving the `CONTINUE` outcome unobserved. Action-Timing samples the missing branch from the same decision context and compares cost-adjusted utilities to directly supervise timing.

Counterfactual intervention makes this comparison possible. In particular, DAS compares factual and counterfactual trajectories to identify decision-boundary errors, constructs preferences over alternative response sequences conditioned on the pre-decision history, and trains the agent through direct preference optimization (Zhang et al., 2026). Its preference objective jointly optimizes the timing choice and subsequent response content within each sequence, rather than assigning them separate utility-based learning signals. This raises a central question: *How can matched, cost-adjusted outcomes directly supervise when to continue and guide continuation plans using immediate stopping as a common reference?*

To address this question, we introduce **Action-Timing**, a cost-aware reinforcement learning framework with an explicit `STOP`/`CONTINUE` decision after evidence analysis. This decision separates timing from subsequent answering or search planning and provides a common intervention point while keeping the preceding analysis fixed. At each training iteration, we retain factual branches from on-policy main trajectories and supplement them with matched same-action and counterfactual rollouts generated by the same policy. A knowledge-state graph organizes these branches by evidence state while retaining their originating decision contexts. We jointly use these branch outcomes to estimate the *Marginal Utility of Continuation* (MUC): the difference in expected cost-adjusted downstream utility between continuing and stopping from the same decision context. Action-Timing uses these estimates for region-specific optimization that directly supervises timing with MUC-based targets and assigns continuation plans localized credit for their cost-adjusted downstream utility relative to immediate stopping from the corresponding decision context. Together, these objectives distinguish learning *whether* further interaction is worthwhile from learning *which* continuation plans are worth pursuing, while grounding both in a common stopping reference. Our contributions are threefold:

- We introduce **Action-Timing**, a cost-aware RL framework that exposes an explicit `STOP`/`CONTINUE` decision and estimates the Marginal Utility of Continuation (MUC) from main-trajectory branches and matched resampled rollouts.
- We develop a region-specific optimization scheme that uses immediate stopping as a common utility reference to directly supervise timing with MUC-based targets and assign localized credit to continuation plans.
- Experiments on search-agent benchmarks show that Action-Timing achieves the highest average EM among evaluated baselines and reduces over-searching and under-searching.

2 Related Work

2.1 Agentic Search and Adaptive Retrieval

Retrieval-augmented generation (RAG) grounds language models with external knowledge (Lewis et al., 2020), while subsequent studies extend static retrieval toward iterative reasoning and tool interaction through interleaved retrieval (Trivedi et al., 2023; Jiang et al., 2023), reasoning-and-acting (Yao et al., 2023), and adaptive retrieval conditioned on self-reflection, query complexity, or model uncertainty (Asai et al., 2024; Jeong et al., 2024; Yao et al., 2025). Recent agentic search systems further enable LLMs to autonomously formulate queries, inspect and integrate evidence, and perform multi-step reasoning through repeated search and web interaction (Li et al., 2025b;a; Wu et al., 2025a;b).

As search becomes a sequential action, its effectiveness and efficiency depend on controlling both insufficient information seeking and unnecessary interaction. Existing approaches address this trade-off through uncertainty-aware search (Wu et al., 2025c), knowledge-grounded process rewards for search and non-search decisions (Wu et al., 2026b), search penalties (Wu et al., 2026a), and knowledge-aware regulation of external knowledge use (Qiao et al., 2025; Xiong et al., 2026). Other studies analyze over-searching and its implications for search efficiency (Xie et al., 2026) or predict intermediate answer quality and its changes across retrieval–reasoning iterations to guide early stopping (Tian et al., 2026). These studies motivate adapting search behavior to the agent's information needs and interaction costs; our focus is on comparing the cost-adjusted downstream utilities of continuing and answering immediately from the same decision context.

2.2 Reinforcement Learning for Search Agents

Reinforcement learning enables agents to learn multi-turn search policies from interaction and outcome feedback, reducing reliance on handcrafted workflows or supervised trajectories (Jin et al., 2025; Zheng et al., 2025b;a; Dong et al., 2026). Sparse terminal supervision, however, provides limited guidance for intermediate decisions. Research on search agents and broader agentic RL addresses this challenge through anchor-state grouping and step-level advantages (Feng et al., 2025), round-level contribution estimation (Wang et al., 2026), structure-aware advantage normalization (Zhu et al., 2026), and tree-based estimation of step-level credit from outcome rewards (Ji et al., 2026). Complementary approaches improve intermediate reasoning and decision-making through self-correction (Li et al., 2026; An et al., 2026), process reward modeling (Xi et al., 2026), and semantic information-gain rewards (Hu et al., 2026). Curriculum-guided self-imitation and intrinsic reward scheduling also help balance exploration and exploitation while mitigating excessive tool interaction (Qin et al., 2026). Counterfactual supervision supports termination and initial search routing (Liu et al., 2025; Kim, 2026), while DAS aligns search decision boundaries through response-sequence preference optimization (Zhang et al., 2026). Our approach uses matched, cost-adjusted branch utilities to construct region-specific learning signals that directly supervise the explicit `STOP`/`CONTINUE` decision and assign localized credit to continuation plans relative to the same stopping reference.

3 Action-Timing

As illustrated in Figure 2, Action-Timing learns whether further search justifies its cost relative to answering immediately by exposing an explicit `STOP`/`CONTINUE` decision after evidence analysis. During training, main-trajectory branches and matched resampled rollouts are organized in a knowledge-state graph to estimate the *Marginal Utility of Continuation* (MUC) from cost-adjusted downstream utilities. Region-specific optimization uses MUC-based targets to supervise timing and assigns continuation plans credit relative to immediate stopping.

3.1 Cost-Aware Timing Decisions for Search Agents

For a question x, we represent the state immediately before the timing decision at step t as $s_t = (x, h_t, e_t, b_t)$, where h_t is the compact policy context, including the analysis preceding the decision, e_t records the accumulated retrieval evidence, and b_t is the remaining action budget. The evidence

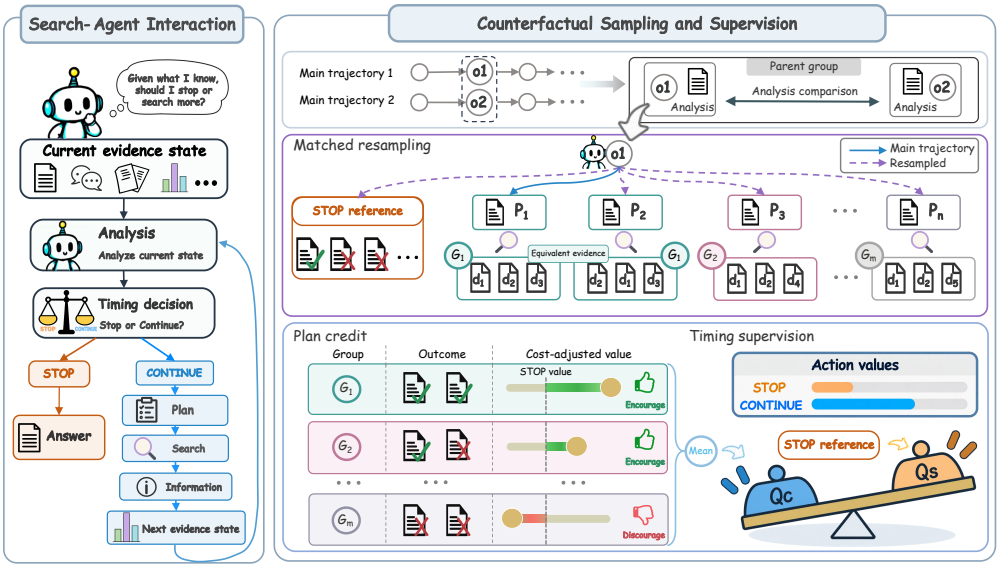


Figure 2: **Overview of Action-Timing. Left:** The agent explicitly chooses `STOP` or `CONTINUE` after evidence analysis. **Right:** Analyses are compared within parent knowledge groups, while continuation branches from main trajectories and matched resampling are grouped by equivalent successor evidence. Cost-adjusted group utilities determine planning credit relative to the stopping reference; their equal-weight average yields the continuation value, whose difference from the stopping value defines MUC for direct timing supervision.

record e_t supports knowledge-state grouping and need not be fully retained in the model context h_t. At this boundary, the agent chooses $a_t \in \{\texttt{STOP}, \texttt{CONTINUE}\}$. Choosing `STOP` ends information seeking and triggers answer generation. Choosing `CONTINUE` initiates another planning–retrieval cycle, after which the agent analyzes the updated evidence and makes the next timing decision. We expose this choice through an explicit timing boundary after evidence analysis:

$$\texttt{Analysis} \rightarrow \texttt{Timing} \rightarrow \begin{cases} \texttt{Answer}, & \texttt{STOP}, \\ \texttt{Plan} \rightarrow \texttt{Search} \rightarrow \texttt{Information}, & \texttt{CONTINUE}. \end{cases} \tag{1}$$

This boundary separates the timing choice from subsequent answering or search planning. It also allows either action to be evaluated from the same pre-decision context and remaining budget without regenerating the preceding analysis.

We evaluate a complete trajectory τ using $U(\tau) = \mathcal{U}(Y(\tau), C(\tau))$, where $Y(\tau) \in \{0, 1\}$ denotes answer correctness and $C(\tau) = \lambda_c T(\tau)$ measures the cost of the realized interaction rounds, including the terminal answering round. The utility $\mathcal{U}(y, C) = y(1 - C)$ discounts correct outcomes by their interaction cost while assigning incorrect outcomes zero utility; outcome evaluation and detailed cost accounting are specified in Appendix A.

3.2 Counterfactual Marginal Utility of Continuation

At a decision state s, we evaluate each choice by its post-decision utility $U_s^{\text{post}}(\tau) = \mathcal{U}(Y(\tau), C_s^{\text{post}}(\tau))$, where $C_s^{\text{post}}(\tau)$ counts the interaction cost from the selected boundary to termination. The timing action value averages post-decision utility within branch classes and then aggregates the class-level values under an action-specific evaluation distribution:

$$Q(s, a) = \mathbb{E}_{g \sim \nu_s^a} \left[\mathbb{E} \left[U_s^{\text{post}}(\tau) \mid s, a, g \right] \right], \tag{2}$$

Here, g indexes a branch class and ν_s^a specifies its evaluation weight under action a. For `STOP`, all immediate-answer branches form a single class. For `CONTINUE`, classes correspond to distinct observed successor knowledge groups and receive equal weight under ν_s^a, rather than their policy-induced probabilities. We define the *Marginal Utility of Continuation* (MUC) as

$$\text{MUC}(s) = Q(s, \texttt{CONTINUE}) - Q(s, \texttt{STOP}). \tag{3}$$

A positive MUC favors continuing, whereas a negative MUC favors stopping; its magnitude measures the utility gap between the two choices under this evaluation distribution. The continuation value accounts for both the effectiveness and the cost of subsequent search plans. A realized rollout provides only one branch outcome; estimating the contrast requires matched evidence for both timing actions.

3.3 Knowledge-State Graph and Matched Resampling

Knowledge-state structure. The graph $\mathcal{K} = (\mathcal{V}, \mathcal{E})$ groups states by question, state type, and accumulated retrieval evidence, with each physical visit retained as an occurrence o. A deterministic grouping map κ assigns its state s_o to a knowledge group $G = \kappa(s_o)$, while each occurrence retains its own decision context and rollout snapshot. Edges record `CONTINUE` transitions to successor knowledge groups and `STOP` transitions to terminal groups, with the grouping rules detailed in Appendix B. The occurrence-specific pool $\mathcal{B}_o^a$ contains admitted main-trajectory and resampled branches for action a. Each continuation branch is assigned to a child group g by applying κ to its direct successor after the first planning–retrieval cycle. The subset $\mathcal{B}_o^{\texttt{CONTINUE}}(g)$ collects branches reaching g, defining the observed child set $\mathcal{C}(o) = \{g \in \mathcal{V} \mid \mathcal{B}_o^{\texttt{CONTINUE}}(g) \neq \varnothing\}$.

Matched branch resampling. Matched resampling retains the original timing action for same-action branches and switches it for counterfactual branches, proceeding through stopping evaluation, continuation exploration, and value completion. Stopping evaluation forces `STOP` and samples answers, while continuation exploration forces `CONTINUE`, samples a new plan and search query, executes retrieval, and follows the rollout policy to termination. Exploration may revisit an existing child group or discover a new one, with each completed branch providing its terminal outcome and post-decision cost. Value completion supplies additional return samples for children with insufficient support by replaying a representative exploratory plan and query to recover the corresponding successor evidence, then sampling the remaining interaction. Occurrence selection, branch admission, representative-plan selection, and sampling quotas are detailed in Appendix C.

3.4 Group-Balanced Utility Estimation

We estimate action values from admitted main-trajectory and resampled branches using their post-decision utilities $u_i = U_{s_o}^{\text{post}}(\tau_i)$. Stopping branches form a single evaluation class g_{term}, with $\mathcal{B}_o^{\texttt{STOP}}(g_{\text{term}}) = \mathcal{B}_o^{\texttt{STOP}}$, yielding the evaluation distribution

$$\nu_{s_o}^a = \begin{cases} \delta_{g_{\text{term}}}, & a = \texttt{STOP}, \\ \text{Unif}\big(\mathcal{C}(o)\big), & a = \texttt{CONTINUE}, \end{cases} \tag{4}$$

where $\delta_{g_{\text{term}}}$ is a point mass and $\text{Unif}(\mathcal{C}(o))$ weights observed child groups equally. We first average branch utilities within each class and then aggregate the class means to estimate the action value:

$$\bar{u}_o^a(g) = \frac{1}{|\mathcal{B}_o^a(g)|} \sum_{\tau_i \in \mathcal{B}_o^a(g)} u_i, \qquad \widehat{Q}_{\text{post}}(s_o, a) = \mathbb{E}_{g \sim \nu_{s_o}^a}[\bar{u}_o^a(g)]\,. \tag{5}$$

Equal group weighting prevents a child from receiving greater aggregation weight solely because more repeated or value-completion samples have been collected for it. The difference between the estimated continuation and stopping values gives $\widehat{\text{MUC}}(s_o)$, while the stopping estimate also serves as the reference for planning credit.

3.5 Region-Specific Policy Optimization

Direct timing supervision. The timing target d_o is based on the empirical MUC $\widehat{\text{MUC}}(s_o)$ and may be refined by supported child-level comparisons against the same stopping reference, with target-selection rules and support criteria detailed in Appendix D. For the timing prefix h_o^{tim}, which ends at the opening timing tag, we score each action's content sequence $v_{a,1:L_a}$ and normalize over the two actions:

$$\ell_\theta(o, a) = \sum_{j=1}^{L_a} \log \pi_\theta\big(v_{a,j} \mid h_o^{\text{tim}}, v_{a,<j}\big)\,, \qquad p_\theta(a \mid o) = \frac{\exp(\ell_\theta(o,a))}{\sum_{a' \in \{\texttt{STOP},\texttt{CONTINUE}\}} \exp(\ell_\theta(o,a'))}. \tag{6}$$

The set $\mathcal{O}_T$ contains occurrences with accepted nonzero targets, whose preferred action $a_o^\star$ is CONTINUE for $d_o > 0$ and STOP for $d_o < 0$. With supervision weights $w_o = \min(|d_o|, w_{\max})$, the timing objective combines direct supervision with action-level KL regularization:

$$\mathcal{L}_{\text{timing}} = -\frac{1}{\max(|\mathcal{O}_T|, 1)} \sum_{o \in \mathcal{O}_T} w_o \log p_\theta(a_o^\star \mid o) + \gamma_{\text{AKL}} \mathcal{L}_{\text{AKL}}. \tag{7}$$

Here, $\mathcal{L}_{\text{AKL}}$ constrains drift from the rollout policy's two-action distribution, frozen for each batch, with its definition and eligibility criteria detailed in Appendix E.

Planning credit relative to stopping. When the stopping estimate is available, a locally supervised plan from occurrence o reaching child group $g \in \mathcal{C}(o)$ receives credit based on the difference between the estimated child utility and the utility of immediate stopping:

$$A_{o,g}^{\text{plan}} = \text{clip}\Big(\bar{u}_o^{\texttt{CONTINUE}}(g) - \widehat{Q}_{\text{post}}(s_o, \texttt{STOP}), -c_{\text{loc}}, c_{\text{loc}}\Big). \tag{8}$$

Plans from the same occurrence reaching the same child group share this advantage, pooling return evidence to reduce dependence on any single downstream realization. These child-level utility estimates also contribute to MUC estimation, connecting timing and planning through a common stopping reference. For resampled continuation branches, only newly sampled Plan/Search actions at the intervention boundary receive direct policy-gradient supervision, while subsequent rollout segments provide return evidence for utility estimation.

Joint policy optimization. We optimize the shared policy π_θ with a single weighted and normalized clipped policy-gradient objective $\mathcal{L}_{\text{PG}}$, aggregating eligible content tokens using their region-specific advantages and token weights. Main-trajectory analysis and answers receive question-level GRPO feedback from terminal utilities, while analysis additionally incorporates a weighted local correction obtained by centering $V_o^{\text{analysis}} = \max_a \widehat{Q}_{\text{post}}(s_o, a)$ within its knowledge group when local estimates are available. Plan/Search tokens use their local advantages and hierarchical token weights, while the structural auxiliary objective $\mathcal{L}_{\text{protocol}}$ applies cross-entropy supervision to canonical structural tokens. The total objective is

$$\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{PG}} + \lambda_T \mathcal{L}_{\text{timing}} + \lambda_P \mathcal{L}_{\text{protocol}} + \beta \mathcal{L}_{\text{KL}} - \eta \mathcal{H}. \tag{9}$$

Here, $\mathcal{L}_{\text{KL}}$ penalizes divergence from the fixed reference policy π_{ref}, and $\mathcal{H}$ is policy entropy. Further optimization details are provided in Appendix E.

4 Experiments

4.1 Experimental Setup

Datasets. We construct our training data from the training splits of NQ and HotpotQA. We evaluate Action-Timing on seven search-augmented question answering benchmarks, covering single-hop QA (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)) and multi-hop QA (HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023)).

Implementation Details. Our implementation builds on Search-R1 and initializes separate policies from Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, using E5-base-v2 to retrieve the top three passages per query from the Wikipedia corpus (wiki-18). All results from our own training experiments are averaged over 3 independent training runs with different random seeds. Hyperparameters and matched branch resampling settings are provided in Appendices F and C, respectively.

Metrics. We measure answer correctness using normalized Exact Match (EM) and adopt three metrics from DAS. Average Search Calls per Question (ASQ) measures retrieval usage; Over-Searching Rate (OSR) measures the proportion of cases where earlier forced answering succeeds; Under-Searching Rate (USR) measures the proportion with incorrect final answers not classified as over-searching. Calculation details are provided in Appendix G.

Table 1: **EM (%) comparison on QA benchmarks with Qwen2.5-3B/7B-Instruct backbones.** Bold and underlined scores mark the best and second-best results at each model scale, respectively. † marks results reproduced by us.

Methods	Single-Hop QA			Multi-Hop QA				
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-3B-Instruct								
Direct Inference	10.6	28.8	10.8	14.9	24.4	2.0	2.4	13.4
Search-o1	15.1	44.3	13.1	18.7	17.6	5.8	29.6	20.6
Search-R1	34.1	54.5	37.8	32.4	31.9	10.3	26.4	32.5
InfoReasoner	45.3	**63.4**	44.2	34.4	32.4	8.0	14.4	34.6
ReSeek	41.5	55.3	43.4	32.8	29.8	10.3	30.4	34.8
Tree-GRPO	<u>46.8</u>	59.7	43.6	<u>42.4</u>	**43.7**	<u>17.8</u>	**43.2**	<u>42.4</u>
GiGPO†	45.1	60.9	<u>44.9</u>	38.0	37.1	14.5	37.9	39.8
Action-Timing	**47.2**	<u>61.8</u>	**46.4**	**42.9**	<u>41.6</u>	**18.1**	<u>41.6</u>	**42.8**
Qwen2.5-7B-Instruct								
Direct Inference	13.4	40.8	14.0	18.3	25.0	3.1	12.0	18.1
Search-o1	23.8	47.2	26.2	22.1	21.8	5.4	32.0	25.5
Search-R1	39.3	61.0	39.7	37.0	41.4	14.6	36.8	38.5
InfoReasoner	44.7	61.4	41.6	41.4	30.2	12.0	42.4	39.1
ReSeek	46.9	64.0	**50.1**	38.9	38.2	18.5	39.2	42.3
Tree-GRPO	<u>48.1</u>	63.3	45.2	<u>44.6</u>	42.3	20.2	**44.0**	44.0
GiGPO†	46.8	<u>65.8</u>	48.1	42.3	42.9	18.2	43.2	43.9
DAS†	44.6	**67.4**	47.4	43.4	<u>45.2</u>	<u>20.6</u>	42.4	<u>44.4</u>
Action-Timing	**48.4**	**67.4**	<u>48.5</u>	**46.6**	**45.4**	**21.4**	<u>43.9</u>	**45.9**

4.2 Main Results

Table 1 shows that Action-Timing achieves the highest average EM with the Qwen2.5-7B-Instruct backbone, reaching 45.9% and exceeding DAS, the strongest baseline in average EM, by 1.5 percentage points and Search-R1 by 7.4 percentage points. On single-hop QA, it achieves the highest EM on NQ and ties with DAS for the best result on TriviaQA. On multi-hop QA, Action-Timing leads on HotpotQA, 2Wiki, and MuSiQue, exceeding the strongest baseline on each dataset by 2.0, 0.2, and 0.8 percentage points, respectively, while ranking second on Bamboogle, only 0.1 percentage point below Tree-GRPO. The Qwen2.5-3B-Instruct results provide further evidence of its effectiveness at a smaller model scale, with the highest average EM of 42.8%, surpassing Tree-GRPO and exceeding Search-R1 by 10.3 percentage points. At this scale, Action-Timing achieves the highest EM on NQ, PopQA, HotpotQA, and MuSiQue and ranks second on the remaining three benchmarks, demonstrating competitive performance across both single-hop and multi-hop QA.

4.3 Search Behavior Analysis

Figure 3 compares the search behavior of Action-Timing, DAS, Search-R1, and Qwen2.5-7B-Instruct. Under the DAS diagnostic protocol, Action-Timing achieves the lowest OSR and USR on all seven benchmarks, indicating improvements in both over-searching and under-searching. Compared with DAS, it reduces OSR from 30.4% to 23.0% on NQ and from 56.4% to 43.6% on TriviaQA, while USR decreases from 52.6% to 47.2% on HotpotQA and from 74.0% to 69.1% on MuSiQue. Action-Timing also achieves the lowest ASQ on six of the seven benchmarks, combining lower diagnostic rates with fewer search calls. MuSiQue presents a different pattern: Action-Timing uses more search calls than DAS and Qwen2.5-7B-Instruct while attaining lower OSR and USR than both. This observation highlights that search frequency alone does not determine the quality of search decisions. Together with the EM results in Table 1, these findings support improved answer correctness and search behavior. The combination of lower diagnostic rates and dataset-dependent search usage is consistent with our objective of learning when further interaction is worthwhile, rather than uniformly suppressing search. Appendix H provides qualitative examples of avoiding premature stopping and redundant search.

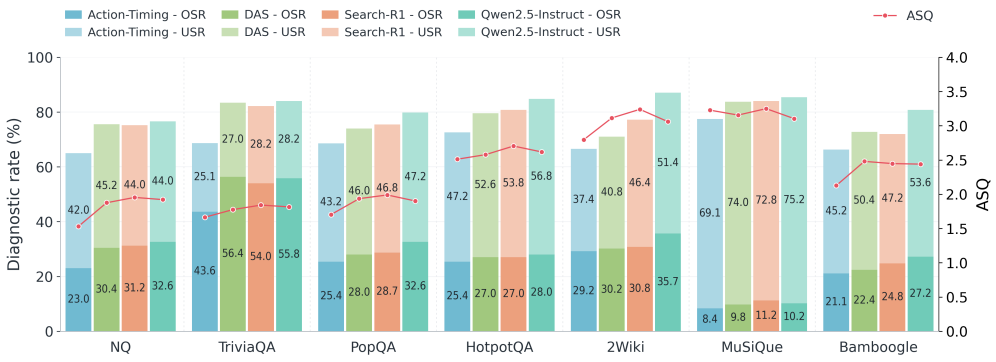


Figure 3: **Search behavior comparison across QA benchmarks.** Dark and light bar segments show over-searching rate (OSR) and under-searching rate (USR), respectively, following the diagnostic protocol of DAS. Red lines show average search calls per question (ASQ; right axis).



Figure 4: **Training dynamics and corrective timing supervision.** (a) Mean training reward and (b) average search calls on main trajectories, comparing Action-Timing with its timing-supervision ablation using Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. (c) Corrective target rates for the 7B model, measured separately among original STOP and CONTINUE occurrences with accepted timing targets. Arrows indicate the original action and its prescribed target.

4.4 Training Dynamics

We compare Action-Timing with a variant that retains the explicit `STOP`/`CONTINUE` protocol but uses GRPO to optimize timing actions. Action-Timing attains higher training rewards in later stages at both model scales (Figure 4(a)), while its average search calls decrease overall from their initial levels but remain higher than those of the GRPO timing variant toward the end of training (Figure 4(b)). The larger reduction in search calls and lower late-stage rewards of the GRPO timing variant suggest that cost penalties may overly suppress search when timing actions are optimized through trajectory-level returns. Figure 4(c) shows that both `STOP`-to-`CONTINUE` and `CONTINUE`-to-`STOP` targets remain present throughout training, indicating sustained supervision in both directions. They support learning when further interaction is worthwhile through supervision that addresses both stopping and continuation choices.

4.5 Ablation Studies

Timing supervision and planning credit. We ablate the full timing objective, comprising MUC-based supervision and action-level KL regularization, and STOP-referenced planning credit, replacing each disabled component with GRPO updates for its region while keeping the explicit timing protocol, graph construction, resampling, and action-cost settings unchanged. Table 2 reports

Table 2: **Component ablations on Qwen2.5-7B-Instruct.** Results are averaged over seven QA benchmarks. Ablated regions use GRPO updates.

Variant	Components		Metrics			
	Timing	Planning	EM (%) ↑	ASQ	OSR (%) ↓	USR (%) ↓
Action-Timing (Full)	✓	✓	**45.9**	2.22	**25.2**	**44.2**
w/o Timing Supervision	×	✓	<u>43.1</u>	2.01	27.7	<u>46.6</u>
w/o Planning Credit	✓	×	41.7	2.17	<u>26.3</u>	47.9
w/o Both	×	×	40.1	1.89	29.9	49.4

Table 3: **Utility aggregation and plan value sharing ablations.** Results use Qwen2.5-7B-Instruct and are averaged over seven QA benchmarks.

Aggregation	Plan value	EM (%) ↑	ASQ	OSR (%) ↓	USR (%) ↓
Group-balanced	Group-shared	**45.9**	2.22	**25.2**	**44.2**
Branch-averaged	Group-shared	<u>45.1</u>	2.28	25.9	<u>45.8</u>
Group-balanced	Individual return	44.7	2.19	<u>25.6</u>	46.1
Branch-averaged	Individual return	43.8	2.11	26.1	47.3

macro-averaged results across the seven QA benchmarks. Replacing the timing objective reduces EM from 45.9% to 43.1% and increases OSR and USR by 2.5 and 2.4 percentage points, respectively, despite lowering ASQ from 2.22 to 2.01. Replacing planning credit causes a larger EM decrease of 4.2 percentage points and raises USR from 44.2% to 47.9%, with little change in ASQ (2.22 to 2.17), consistent with planning credit improving continuation quality. Removing both components yields the lowest EM and highest OSR and USR; the full model improves EM by 5.8 percentage points over this variant, supporting their complementary roles in learning when to continue and how to plan subsequent search.

Utility aggregation and plan value sharing. Table 3 examines group-balanced continuation estimation and plan value sharing by replacing equal weighting of child-group means with direct branch averaging, and replacing shared plan values with individual branch returns. Value sharing is restricted to plans reaching the same child group from the same occurrence. All variants retain the full timing objective and STOP-referenced planning credit, with grouping, resampling, and other optimization settings unchanged. Group-balanced aggregation improves EM by 0.8 and 0.9 percentage points with shared values and individual returns, respectively, while value sharing improves EM by 1.2 and 1.3 points under group-balanced and branch-averaged aggregation. Both changes also reduce OSR and USR under either setting of the other factor. Combining them achieves the highest EM and lowest diagnostic rates, with a 2.1-point EM gain over branch averaging with individual returns. These results support assigning equal weight to observed continuation groups and pooling return evidence within each group for planning credit. Appendices I and J further examine the planning reference and interaction-cost penalty.

5 Conclusion

We introduced **Action-Timing**, a cost-aware reinforcement learning framework that learns whether further search is worthwhile relative to answering immediately. It combines main-trajectory branches with matched resampled rollouts to estimate the Marginal Utility of Continuation, using region-specific objectives to directly supervise explicit `STOP`/`CONTINUE` decisions and assign continuation plans localized credit relative to the same stopping reference. Action-Timing achieves the highest average EM among evaluated baselines and reduces over-searching and under-searching, supporting the value of jointly learning when to continue and how to make continuation worthwhile.

AI USE STATEMENT

The research ideas, method, experimental design, and mathematical claims and proofs were developed by the authors. We used generative AI tools to provide feedback on the methodological formulation and mathematical arguments, assist with method implementation and code editing, and support qualitative analysis and interpretation of experimental results. We also used these tools to assist with literature review and information search and to review and revise author-written manuscript text for clarity and readability. Figures were constructed manually, with selected cartoon elements, such as robot illustrations, generated using AI. Our experiments used publicly available datasets. We reviewed all AI-assisted outputs and feedback and take responsibility for the final content of this work, including text, claims, code, and visual elements produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 describes the Action-Timing framework, and Appendices A–E detail reward computation, knowledge-state grouping, matched resampling, timing-target construction, and policy optimization, including the supporting derivations and proofs. The datasets, retrieval setup, and training configuration are described in Section 4.1 and Appendix F, while Appendix G specifies the evaluation metrics and diagnostic protocol. The supplementary ZIP archive contains the core implementation of our method, along with training and inference workflows.

REFERENCES

Kang An, Ziliang Wang, Xuhui Zheng, Faqiang Qian, Weikun Zhang, Yuhang Wang, and Wu Yichao. Erase to improve: Erasable reinforcement learning for search-augmented llms. In *International Conference on Learning Representations*, volume 2026, pp. 98392–98419, 2026.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Senkang Hu, Yong Dai, Yuzhi Zhao, Yihang Tao, Yu Guo, Zhengru Fang, Sam Kwong, and Yuguang Fang. Optimizing agentic reasoning with retrieval via synthetic semantic information gain reward. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mUU7k4bN97.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 7036–7050, 2024.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 87362–87388, 2026.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Minho Kim. When should llms search? counterfactual supervision for search routing, 2026. URL https://arxiv.org/abs/2607.05752.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Shiyu Li, Yifan Wang, Peiming Li, Zheng Wei, and Yang Tang. Reseek: A self-correcting framework for search agents with instructive rewards. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SeO8pLS44H.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025a.

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. *Advances in Neural Information Processing Systems*, 38:120091–120131, 2025b.

Grace Liu, Yuxiao Qu, Jeff Schneider, Aarti Singh, and Aviral Kumar. Cart: Teaching llm agents to know when they know enough, 2025. URL https://arxiv.org/abs/2510.08517.

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 *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Shuofei Qiao, Zhisong Qiu, Baochang Ren, Xiaobin Wang, Xiangyuan Ru, Ningyu Zhang, Xiang Chen, Yong Jiang, Pengjun Xie, Fei Huang, et al. Agentic knowledgeable self-awareness. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12601–12625, 2025.

Yulei Qin, Xiaoyu Tan, Zhengbao He, Gang Li, Haojia Lin, Zongyi Li, Zihan Xu, Yuchen Shi, Siqi Cai, Renting Rui, et al. Learn the ropes, then trust the wins: self-imitation with progressive exploration for agentic reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 65092–65131, 2026.

Fangzheng Tian, Debasis Ganguly, and Craig Macdonald. Predicting partial answer quality and utility in agentic retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2609.16453.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Junzhe Wang, Zhiheng Xi, Yajie Yang, Hao Luo, Shihan Dou, Tao Gui, and Qi Zhang. Enhancing llm-based search agents via contribution weighted group relative policy optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31704–31718, 2026.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, et al. Webdancer: Towards autonomous information seeking agency. *Advances in Neural Information Processing Systems*, 38:120957–120985, 2025a.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10290–10305, 2025b.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun MA, and Ziwei Liu. MMSearch-r1: Incentivizing LMMs to search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2456–2487, San Diego, California, United States, July 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.114. URL https://aclanthology.org/2026.acl-long.114/.

Peilin Wu, Mian Zhang, Xinlu Zhang, Xinya Du, and Zhiyu Chen. Search wisely: Mitigating sub-optimal agentic searches by reducing uncertainty. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19723–19734, 2025c.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. Hiprag: hierarchical process rewards for efficient agentic retrieval augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 82317–82347, 2026b.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, et al. Agentprm: Process reward models for llm agents via step-wise promise and progress. In *Proceedings of the ACM Web Conference 2026*, pp. 4184–4195, 2026.

Roy Xie, Deepak Gopinath, David Qiu, Dong Lin, Haitian Sun, Saloni Potdar, and Bhuwan Dhingra. Over-searching in search-augmented large language models. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7714–7739, Rabat, Morocco, March 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.361. URL https://aclanthology.org/2026.eacl-long.361/.

Siheng Xiong, Oguzhan Gungordu, Blair Johnson, James Clayton Kerce, and Faramarz Fekri. Scaling search-augmented LLM reasoning via adaptive information control. In *The 1st Workshop on Scaling Post-training for LLMs*, 2026. URL https://openreview.net/forum?id=nGEWOtne5c.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

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*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Liu Weichuan, Lei Hou, and Juanzi Li. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27022–27043, 2025.

Wenlin Zhang, Kuicai Dong, Junyi Li, Yingyi Zhang, Xiaopeng Li, Pengyue Jia, Yi Wen, Derong Xu, Maolin Wang, Yichao Wang, et al. To search or not to search: Aligning the decision boundary of deep search agents via causal intervention. In *Proceedings of the ACM Web Conference 2026*, pp. 2049–2059, 2026.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21805–21830, 2025a.

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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, 2025b.

Mingkang Zhu, Xi Chen, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Stratified GRPO: Handling structural heterogeneity in reinforcement learning of LLM search agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZX2lOgKOTm.

A Reward and Action-Cost Definitions

We distinguish the reward assigned to a complete main trajectory from the utility assigned to a branch originating at a particular decision boundary, while using the same per-action cost coefficient in both cases.

A.1 Answer Extraction and Normalized Exact Match

For a question x, let $\mathcal{Y}_x$ denote the set of acceptable reference answers and $\hat{y}(\tau)$ the final answer extracted from trajectory τ. Extraction respects the provenance and structure of the interaction record: an eligible answer must appear in a model-generated, top-level `STOP–Answer` pair, excluding content from the input prompt, retrieved information, or answer-like blocks nested within analysis and planning. An `Answer` block following `CONTINUE` is not considered an eligible stopping answer; when multiple eligible candidates are present, the last extracted candidate is evaluated.

We determine answer correctness using normalized exact match (EM), which compares the extracted prediction against each acceptable reference after answer normalization. Let N denote the normalization function, which removes superficial differences through operations such as lowercasing, English article and punctuation removal, and whitespace normalization. For an otherwise evaluable trajectory, correctness is defined as

$$c(\tau) = \mathbf{1}[\exists y \in \mathcal{Y}_x : N(\hat{y}(\tau)) = N(y)]\,, \tag{10}$$

with $c(\tau) = 0$ when no eligible answer can be extracted. A prediction is therefore correct if its normalized form exactly matches at least one normalized reference, without relying on substring matching or semantic judgment.

Final-answer correctness and trajectory-wide protocol validity are recorded separately, so an earlier formatting error does not necessarily invalidate a subsequently extractable correct stopping answer for the main-trajectory reward.

A.2 Action Cost and Interaction Boundaries

Let $T(\tau)$ denote the number of charged action rounds in the realized interaction record. We use a constant per-round cost $\lambda_c = 0.05$, yielding

$$C(\tau) = \lambda_c T(\tau). \tag{11}$$

For a valid interaction, a `CONTINUE` planning–retrieval round incurs one unit of action cost, and the terminal `STOP`–answering round also incurs one unit. Analysis, planning, and query generation are components of these rounds rather than separately charged actions; similarly, retrieving multiple documents within one search call does not incur a separate action charge for each document.

For an ordinary valid trajectory containing k continuation rounds followed by one stopping round,

$$T(\tau) = k + 1, \qquad C(\tau) = \lambda_c(k + 1). \tag{12}$$

In particular, answering immediately has a nonzero cost of λ_c. The cost is an interaction-level proxy and does not directly measure generated tokens, retrieval latency, or wall-clock time, which are separate quantities when reporting computational expenditure.

The configured regular interaction limit is five rounds, with a final-answer mechanism enabled upon budget exhaustion. Because this mechanism may introduce an additional answering round, the configured limit does not imply that every realized trajectory contains at most five charged rounds. We therefore compute cost from the realized interaction record rather than identifying the configured limit with either the number of search calls or the total number of charged actions.

A.3 Correctness-Gated Main-Trajectory Reward

For the task-level reward, we instantiate the terminal outcome as $Y(\tau) = c(\tau)$ and use the utility function

$$\mathcal{U}(y, C) = y(1 - C). \tag{13}$$

Correctness c	Rounds T	Physical cost	Applied cost	Reward
1	1	0.05	0.05	0.95
1	3	0.15	0.15	0.85
1	5	0.25	0.25	0.75
0	1	0.05	0.00	0.00
0	5	0.25	0.00	0.00

Table 4: Constructed examples of correctness-gated trajectory rewards with $\lambda_c = 0.05$. Physical interaction cost is recorded for all evaluable trajectories, whereas the reward deducts that cost only for correct answers.

The reward of a complete main trajectory is therefore

$$\begin{aligned} R(\tau) = U(\tau) &= \mathcal{U}(c(\tau), C(\tau)) \\ &= c(\tau)\big(1 - \lambda_c T(\tau)\big) \\ &= \begin{cases} 1 - \lambda_c T(\tau), & c(\tau) = 1, \\ 0, & c(\tau) = 0. \end{cases} \end{aligned} \tag{14}$$

This construction applies the action-cost deduction only to correct answers. It distinguishes successful trajectories according to their interaction cost while assigning unsuccessful trajectories zero task reward, rather than rewarding shorter failures through a smaller unconditional penalty. Physical cost is nevertheless incurred and recorded for unsuccessful trajectories; the cost applied to the reward is

$$C_{\text{applied}}(\tau) = c(\tau)C(\tau), \qquad R(\tau) = c(\tau) - C_{\text{applied}}(\tau). \tag{15}$$

Table 4 illustrates the distinction between physical and applied cost. No additional format reward, retrieval-hit reward, or outcome-shaping term is included in this task reward; protocol supervision is introduced through separate optimization objectives, as described in Appendix E.

The scalar $R(\tau)$ provides the task-level input to question-group GRPO, whose normalization and regional assignment are specified in Appendix E. Because all incorrect trajectories receive zero task reward, a group containing only incorrect trajectories has no reward variation from which to derive a nonzero GRPO advantage; conversely, an all-correct group can retain reward variation when its trajectories incur different action costs.

A.4 Post-Decision Cost Accounting

Local action comparisons evaluate the suffix beginning at a selected decision state s, after the current analysis and before the timing choice. Let $T_s^{\text{post}}(\tau)$ denote the number of charged rounds from this boundary to termination, including the selected action and the terminal answering round. We define

$$C_s^{\text{post}}(\tau) = \lambda_c T_s^{\text{post}}(\tau), \qquad U_s^{\text{post}}(\tau) = \mathcal{U}\big(Y(\tau), C_s^{\text{post}}(\tau)\big)\,. \tag{16}$$

For an admitted branch with a measurable outcome, this becomes

$$U_s^{\text{post}}(\tau) = c(\tau)\big(1 - \lambda_c T_s^{\text{post}}(\tau)\big). \tag{17}$$

An immediate stopping branch incurs one answering-round cost, whereas a continuation branch also incurs the cost of its initial planning–retrieval round and all subsequent charged rounds. Costs accumulated before s are excluded because they have already been incurred at the point of comparison.

More precisely, if T_s^{pre} denotes the charged rounds preceding the decision boundary, then

$$T(\tau) = T_s^{\text{pre}} + T_s^{\text{post}}(\tau), \qquad U_s^{\text{post}}(\tau) = U(\tau) + c(\tau)\lambda_c T_s^{\text{pre}} \tag{18}$$

for an evaluable complete trajectory and its corresponding suffix.

The same accounting is applied to factual main-trajectory suffixes and matched resampled branches. For example, consider a valid trajectory with two continuation rounds followed by a correct stopping answer. Its complete-trajectory reward is 0.85, while its post-decision utilities at the first, second, and third decision boundaries are 0.85, 0.90, and 0.95, respectively, because the remaining charged

round counts are three, two, and one. These are different evaluations of the same terminal outcome from different decision boundaries, rather than additional rewards assigned to the trajectory.

All action comparisons used for MUC remain within the same originating decision context and remaining budget. Recomputing suffix cost does not reset the interaction budget or grant a resampled branch additional search opportunities.

B KNOWLEDGE-STATE GROUPING

The knowledge-state graph groups occurrences according to their retrieval evidence while preserving the decision contexts required for matched action evaluation. Grouping is performed separately for each question within a training rollout batch, jointly organizing main-trajectory and resampled branches. Occurrences from different questions are never merged, whereas occurrences associated with the same question may share a knowledge group when their state types and accumulated evidence representations agree.

This construction distinguishes shared knowledge-group identity from physical occurrence identity: the former organizes equivalent evidence representations, while the latter retains the context, remaining budget, and branch provenance needed for local estimation. Consequently, sharing a knowledge group does not authorize pooling returns across different originating occurrences or substituting one occurrence's policy context for another.

B.1 EVIDENCE REPRESENTATION AND FINGERPRINTING

Document-level evidence. For fully visible structured retrieval results, each document is represented by its available source identifiers and textual content. Source identifiers include fields such as document or chunk identifiers and URLs, while content is taken from the document's contents, content, or title–text representation, according to the available retrieval fields. Retrieval rank and score are excluded from the ordinary document representation, so changes in ranking alone do not create new evidence atoms.

Let $\phi(d)$ denote the selected representation of document d. Its fingerprint is computed as

$$f(d) = \mathrm{SHA256}(\mathrm{Serialize}(\mathrm{Normalize}(\phi(d)))), \tag{19}$$

where normalization applies Unicode NFKC normalization, whitespace collapsing, and Unicode case folding to textual fields, and serialization uses a deterministic field ordering.

Both document identity and content contribute to the fingerprint. Accordingly, changing the content under the same identifier, or changing an identifier while retaining the same content, can produce a distinct atom. The representation therefore provides deterministic deduplication of normalized retrieval records rather than semantic deduplication of their underlying claims.

Retrieval-event evidence. For retrieval event r returning document collection $\mathcal{D}_r$, the initial evidence representation is

$$F_r^{\mathrm{doc}} = \{f(d) : d \in \mathcal{D}_r\}. \tag{20}$$

Set construction removes repeated documents and makes the representation invariant to document-list order when the structured results remain fully visible. An empty retrieval result contributes no atoms, whereas a returned document with empty textual content can still contribute an atom through its document representation.

B.2 VISIBILITY-AWARE ACCUMULATION

The evidence associated with a retrieval event must reflect the information made available to the agent. Retrieved text is first bounded by the observation-length limit and can be further shortened when constructing the next compact policy context. When either restriction changes the visible information payload, the original document-level atoms for that event are replaced by a single fingerprint of the retained text, excluding the enclosing information tags.

Let z_r^{vis} denote the final visible payload and H_{text} the text fingerprint obtained using the normalization and hashing rules above. The event representation is

$$F_r = \begin{cases} F_r^{\mathrm{doc}}, & \text{if the document-level representation is retained,} \\ \{H_{\mathrm{text}}(z_r^{\mathrm{vis}})\}, & \text{if replaced by nonempty visible text,} \\ \varnothing, & \text{if no evidence is contributed.} \end{cases} \tag{21}$$

This replacement prevents an invisible document suffix from contributing to the successor's evidence identity through the fingerprint of the complete document.

Order invariance consequently applies to the retained document-level representation rather than to every retrieval event. A truncated payload can contain document numbers, titles, and order-dependent text fragments, so changing retrieval order can change its fingerprint. A complete document and a visible fragment derived from that document are also distinct evidence representations and are not subsequently merged by document identifier.

For occurrence o, let $\mathcal{H}(o)$ contain its recorded retrieval events. Its accumulated evidence is

$$E(o) = \bigcup_{r \in \mathcal{H}(o)} F_r, \tag{22}$$

represented deterministically by sorting the distinct fingerprints. Searches that were not executed contribute no evidence, and each executed event is incorporated once. Repeated acquisition of an existing atom does not increase its multiplicity.

Previously acquired atoms remain in $E(o)$ even when their corresponding text is no longer retained in a later compact prompt. Thus, $E(o)$ records accumulated evidence acquired during the interaction, whereas the policy context h_o contains the information currently presented to the model.

B.3 Decision and Terminal State Grouping

Let x_o denote the question associated with occurrence o. Initial states, states following retrieval, and nonterminal retry states are treated as decision states. Their grouping key is

$$K_{\mathrm{dec}}(o) = \big(x_o, \texttt{decision}, E(o)\big). \tag{23}$$

Two decision occurrences belong to the same knowledge group when their keys agree. Current analysis, planning text, query history, interaction depth, and remaining budget are retained with the occurrence or transition but are not separate components of this key.

Under a fixed question and evidence representation, occurrences can therefore share a group despite having different reasoning contexts or remaining budgets. These omitted variables can nevertheless affect grouping indirectly: for example, different context lengths can change how much retrieved information remains visible and thereby change $E(o)$. Likewise, the empty evidence set is not exclusive to initial states, since an empty retrieval result or a retry without new evidence can leave a later occurrence in the same group as an initial occurrence.

Terminal groups. Terminal states are distinguished from decision states and retain additional outcome information in their grouping key:

$$K_{\mathrm{term}}(o) = \big(x_o, \texttt{terminal}, E(o), r_o, H_{\mathrm{out}}(z_o), v_o^{\mathrm{term}}\big), \tag{24}$$

where r_o denotes the termination reason, z_o the terminal output, and v_o^{term} the recorded format-validity status, which can be true, false, or unknown. For an ordinary stopping answer, z_o is the extracted answer content; exceptional termination paths can instead use the failed output or a null output when no answer is produced. The output fingerprint H_{out} uses the same general text normalization as evidence fingerprinting.

Terminal format validity can reflect the full interaction sequence rather than only the final answer block. A successful answer produced at the budget boundary can still be an ordinary answer termination, whereas an unresolved budget exhaustion or a protocol failure can produce a different terminal state. System failures need not create terminal occurrences and are handled separately by the branch-admission rules.

B.4 Occurrences and Recoverable Decision Contexts

Every physical visit remains a separate occurrence, preserving its trajectory provenance, parent relation, interaction position, remaining budget, and available action and evidence records. Grouping these visits does not remove their individual identities or replace their contexts with a representative context.

For an occurrence with a recoverable timing boundary, we additionally retain the exact decision prefix and rollout snapshot used for matched resampling. The prefix contains the actual model input and the sampled analysis preceding the timing choice, so stopping and continuation branches can be evaluated without regenerating that analysis. This snapshot belongs to the originating occurrence rather than to the successor produced by its action.

A newly created post-retrieval occurrence may therefore have a knowledge-group identity before its next analysis and timing decision have been generated. Similarly, an invalid output may lack a recoverable timing boundary. Knowledge-group membership alone does not imply that an occurrence is eligible for resampling, and an occurrence with an unavailable or unaligned decision prefix is not made eligible by substituting the prefix of another group member. The precise recovery and admission conditions are specified in Appendix C.

B.5 Transitions, Convergence, and Self-Loops

For a valid `CONTINUE` action, the direct successor is established after executing retrieval and applying the relevant visibility restrictions, but before generating the next analysis. The first planning–retrieval cycle therefore determines its child knowledge group. A valid `STOP` action instead produces a terminal occurrence and a transition to its terminal group.

The graph retains physical parent–child relations together with their projection onto knowledge-group edges, preserving the source occurrence and action information of each transition. Different physical paths can consequently converge on the same knowledge group without being treated as the same sampled path. Similarly, multiple physical transitions can project onto one group edge while retaining separate branch identities and outcomes.

A decision-to-decision transition that leaves $E(o)$ unchanged can create a group-level self-loop, as can occur when retrieval returns no evidence or only previously recorded atoms.

Failure transitions are distinguished from valid retrieval transitions. An invalid main-trajectory action can produce a diagnostic retry state without executing a search, whereas a failed forced continuation can terminate and create a genuine terminal child.

B.6 Occurrence-Local Returns and Group-Level Comparisons

Timing and planning. Branch returns retain their originating occurrence, parent group, remaining budget, action, successor identity, and terminal outcome. Timing and Plan estimates are computed from the branches admitted for the corresponding physical origin, with budget consistency checked separately from knowledge-group identity.

Suppose o_1 and o_2 belong to the same knowledge group and both reach child group g. Their shared child identity does not merge the return collections

$$\mathcal{B}_{o_1}^{\texttt{CONTINUE}}(g) \quad \text{and} \quad \mathcal{B}_{o_2}^{\texttt{CONTINUE}}(g). \tag{25}$$

Each collection supports its own occurrence-conditioned child value, so a favorable return from o_1 is not transferred directly to o_2 merely because their successor evidence agrees. Matched value-completion samples likewise remain associated with the designated source occurrence and child.

Stopping class versus terminal groups. The single stopping class g_{term} in Section 3.4 is an evaluation category, not a single terminal node shared by all answers. Within an occurrence, admitted stopping returns are aggregated without further partitioning by answer text or terminal-group identity, although the graph preserves those distinctions. This separates the question of how terminal states are represented from the question of how immediate-answer utility is estimated.

Analysis comparisons. Analysis uses a different comparison scope: available values from eligible main-trajectory occurrences are compared within their knowledge group, and the current configuration allows this comparison across remaining budgets. Occurrences without usable local values retain the task-level fallback, while resampled analysis is not introduced as an additional comparison entry. The associated value definitions, advantage construction, and token assignment are specified in Appendix E.

C MATCHED BRANCH RESAMPLING

Resampling operates on selected physical occurrences from the main trajectories, preserving their pre-decision contexts and remaining budgets while forcing either `STOP` or `CONTINUE`. The procedure combines stopping evaluation, bounded continuation exploration, and support-based value completion.

Admitted main-trajectory branches and matched resampled branches jointly contribute to local action-value estimation, with each factual return included in the pool corresponding to its realized timing action and all admitted returns matched to the same originating decision context and remaining budget.

C.1 OCCURRENCE SELECTION

An occurrence is eligible for resampling only when it has a valid and canonical timing decision, its subsequent action is consistent with that decision, its exact decision snapshot is recoverable, and its remaining action budget is a positive integer. The snapshot must also preserve a valid source-attributed interaction prefix, and occurrences explicitly excluded from resampling are not admitted. Sharing a knowledge group with an eligible occurrence does not make another occurrence eligible.

The candidate occurrences are fixed before resampling begins, so states newly created by a resampled branch do not recursively become new resampling origins within the same training iteration. The current procedure considers all eligible occurrences in each selected knowledge group rather than substituting a single representative context.

Questions with all incorrect main trajectories. Ordinary resampling excludes questions whose main trajectories all have observed incorrect answers. This condition is based on answer correctness rather than zero cost-adjusted reward, and unknown outcomes are not treated as incorrect answers. To retain limited exploration on these questions, we select at most one eligible initial occurrence from each of up to 16 such questions.

When ordinary eligible occurrences are available, the number of selected difficult roots is bounded by both 16 and the number of ordinary occurrences, and an equal number of ordinary occurrences is randomly displaced. If no ordinary occurrences are available, up to 16 difficult roots can still be selected without displacement.

C.2 MATCHED DECISION CONTEXTS AND BRANCH EXECUTION

Each branch starts at the boundary after the current analysis and before the timing decision. We recover the actual model-input tokens together with the sampled analysis prefix, preserving the originating occurrence's evidence context and remaining budget. The selected timing action is then supplied as a forced condition: a stopping branch begins answer generation, whereas a continuation branch begins generation of a new Plan/Search action, executes retrieval, and samples the subsequent interaction until termination. Later timing decisions follow the rollout policy.

The analysis preceding the selected boundary is therefore fixed across the two actions, while their downstream outputs are newly sampled. Forced timing and structural prefixes are conditioning tokens rather than naturally sampled policy-gradient targets. For auxiliary continuation training, only the first eligible sampled Plan/Search content is considered for direct policy-gradient supervision; retaining a completed branch for evaluation does not make all of its subsequent outputs training targets.

Budget and cost accounting. The remaining budget is inherited from the originating occurrence rather than reset for the new branch. For a factual main-trajectory suffix, post-decision cost is obtained by subtracting the cost incurred before the boundary from the complete-trajectory cost; a newly generated branch records the costs of its own subsequent rounds, including its first forced action and terminal answering round.

Under the enabled final-answer mechanism, exhausting the remaining regular interaction budget can be followed by an additional forced answering round. This final round contributes to action cost even though it does not consume another regular budget unit. Replaying the first Plan/Search during value completion likewise retains the logical cost of that action, although its tokens are not sampled again.

C.3 Stopping Evaluation

The `STOP` value is estimated from a pool of four admitted stopping outcomes, including the factual main-trajectory `STOP` outcome when available, with matched stopping samples completing the pool. For the resulting stopping pool $\mathcal{S}_o$, we compute

$$\widehat{Q}_S(o) = \frac{1}{4} \sum_{b \in \mathcal{S}_o} U_b, \qquad |\mathcal{S}_o| = 4, \tag{26}$$

where U_b is the scored post-decision utility defined in Appendix A.

C.4 Bounded Continuation Exploration

For each selected occurrence, continuation exploration issues at most six additional forced-`CONTINUE` attempts. An admitted main-trajectory continuation can already contribute a child observation, support count, and return, but does not consume this additional exploration allowance. Stopping attempts and subsequent value-completion attempts have separate budgets.

Each exploration attempt samples a new initial Plan/Search action and, when executable, performs retrieval before continuing the rollout. The direct child is determined after this first retrieval and the visibility adjustments described in Appendix B, before the next analysis is generated. An accepted branch supplies its direct child identity together with its terminal outcome and post-decision cost.

Capacity-based collisions. Reaching an existing child is not sufficient to constitute a collision. A collision occurs when the first continuation reaches an existing reusable decision child for which the same originating occurrence already has at least three qualifying coverage returns. The threshold applies to the number of qualifying returns from this origin to this child, rather than the number of distinct children or the total number of graph occurrences sharing that child.

C.5 Support-Based Value Completion

After exploration, we allocate additional rollouts to children with insufficient return support. The support target is three returns per child, with at most six completion attempts per origin.

Support and representative paths. The allocation procedure counts recorded continuation returns with the matching origin, parent group and budget, a nonempty child identity, an evaluated outcome, and available post-decision cost. A child can receive completion attempts only when an accepted resampled discovery provides an eligible, protocol-valid Plan/Search action with an exactly replayable first round. The representative is the first qualifying branch under a deterministic branch ordering, rather than the branch with the largest observed return.

Frozen allocation. For each eligible child g, let $n_{o,g}$ be its pre-completion support count. Its initial deficit is

$$d_{o,g} = \max(0, 3 - n_{o,g}). \tag{27}$$

Up to six tasks are assigned by repeatedly selecting an eligible child with a remaining deficit, prioritizing its current planned support count, then the number of already assigned completion tasks, with a deterministic child ordering used to resolve ties.

Replay and return admission. A completion rollout restores the designated discovery's first-round prompt and replays its sampled Plan/Search tokens as conditioning, then executes retrieval again and samples the remaining interaction. Completion samples do not create additional Plan policy-gradient examples or increase discovery coverage, because their first actions were replayed rather than newly sampled; their admitted downstream outcomes instead augment the value evidence for the designated child.

D TIMING TARGET CONSTRUCTION

We first establish the relationship between the empirical utility gap and a weighted classification objective, then specify the reliability conditions, supported child-level refinements, and planning requirements that determine the final target.

D.1 LOCAL UTILITY AND THE BASIS FOR MUC SUPERVISION

For occurrence o, write $\widehat{Q}_C(o)$ and $\widehat{Q}_S(o)$ for the continuation and stopping estimates constructed in Appendix C, and define the base empirical MUC as

$$\Delta_o = \widehat{Q}_C(o) - \widehat{Q}_S(o). \tag{28}$$

Throughout the following analysis, these estimates are held fixed. Let $p_o = p_\theta(\texttt{CONTINUE} \mid o)$ denote the normalized two-action probability. The corresponding mixture of the estimated action values is

$$\widetilde{V}_o(\theta) = p_o\widehat{Q}_C(o) + (1-p_o)\widehat{Q}_S(o) = \widehat{Q}_S(o) + p_o\Delta_o. \tag{29}$$

Consequently, $\partial\widetilde{V}_o/\partial p_o = \Delta_o$: a positive empirical MUC favors increasing the continuation probability, a negative value favors decreasing it, and a zero value supplies no preference under the current estimates. This mixture is a local evaluation objective based on fixed branch estimates, rather than a claim about the realized return of a subsequently updated policy.

Proposition (Local decision-regret bound). Let $a_o^\star \in \arg\max_{a\in\{\texttt{STOP},\texttt{CONTINUE}\}} \widehat{Q}_a(o)$, and suppose $p_\theta(a_o^\star \mid o) > 0$. Define the local decision regret relative to the better estimated action by

$$\mathcal{R}_o(\theta) = \max_a \widehat{Q}_a(o) - \sum_a p_\theta(a \mid o)\widehat{Q}_a(o). \tag{30}$$

Then

$$\mathcal{R}_o(\theta) = |\Delta_o|\big(1 - p_\theta(a_o^\star \mid o)\big) \le -|\Delta_o| \log p_\theta(a_o^\star \mid o). \tag{31}$$

Proof. Let a_o^- denote the other action. Since the two action probabilities sum to one,

$$\begin{aligned}\mathcal{R}_o(\theta) &= \widehat{Q}_{a_o^\star}(o) - p_\theta(a_o^\star \mid o)\widehat{Q}_{a_o^\star}(o) - p_\theta(a_o^- \mid o)\widehat{Q}_{a_o^-}(o)\\ &= \big(1 - p_\theta(a_o^\star \mid o)\big)\big(\widehat{Q}_{a_o^\star}(o) - \widehat{Q}_{a_o^-}(o)\big)\\ &= |\Delta_o|\big(1 - p_\theta(a_o^\star \mid o)\big).\end{aligned} \tag{32}$$

Applying $1 - p \le -\log p$ for $p \in (0, 1]$ establishes the inequality. If $\Delta_o = 0$, both sides are zero regardless of which maximizing action is selected.

D.2 BASE TARGETS

Reliability-gated base target. Let $\widehat{r}_S(o)$ denote stopping accuracy in the complete stopping pool, using the scoring convention in Appendix A. With a stopping-accuracy threshold $\tau_S = 0.70$, the base target is

$$d_o^{\text{base}} = \begin{cases} \Delta_o, & \Delta_o > 0,\\ \Delta_o, & \Delta_o < 0 \ \wedge\ \widehat{r}_S(o) \ge \tau_S,\\ \perp, & \text{otherwise,}\end{cases} \tag{33}$$

where the first two cases require both action-value estimates to be available, and $\perp$ denotes the absence of a target rather than a zero-valued training label. The base improvement margin is zero,

so an accepted nonzero difference can provide supervision without exceeding an additional positive margin.

The stopping-accuracy condition prevents a negative average continuation gap from automatically reinforcing a stopping action that is itself insufficiently reliable. With four stopping returns, the threshold requires at least three successful stopping answers. A negative base MUC does not additionally require the stopping value to exceed every observed child value: the base comparison is against the group-balanced continuation estimate.

When $\Delta_o = 0$, the current value evidence does not support either action, and no base target is assigned. Unavailable estimates similarly produce no base target, whereas a negative gap rejected solely by the stopping-accuracy condition remains eligible for the supported continuation refinement described below.

D.3 Supported Child-Level Refinement

The base continuation value averages across observed evaluation classes, whereas the policy can benefit from learning a particular continuation whose child has sufficiently supported outcomes. We therefore construct a Teacher target from the same local return evidence, using "Teacher" to denote a data-derived target rather than an external model.

For a child group g associated with a valid planning transition from o, let $V_{o,g}$ denote its shared mean utility, $n_{o,g}$ its number of admitted returns, and $\widehat{r}_{o,g}$ its empirical answer-correctness rate over the admitted returns. These statistics include the admitted discovery and matched completion returns assigned to that child. Its improvement relative to stopping is

$$\delta_{o,g} = V_{o,g} - \widehat{Q}_S(o). \tag{34}$$

The initially supported candidate set is

$$\mathcal{H}_o = \{g \in \mathcal{C}(o) : n_{o,g} \geq n_{\min},\ \widehat{r}_{o,g} \geq \tau_C,\ \delta_{o,g} > m_C\}, \tag{35}$$

with $n_{\min} = 3$, $\tau_C = 0.80$, and $m_C = 0.05$. With exactly three supporting returns, the success-rate requirement is satisfied only when all three are successful; for larger pools, it is evaluated using the observed proportion.

Refinement priority. Teacher refinement is permitted in two cases. If the base MUC is positive, an eligible child may replace the base target only when its improvement exceeds the existing positive gap. If the base MUC is negative but its stopping target is withheld solely because stopping accuracy is below τ_S, a supported child may supply a positive continuation target. Accordingly, for occurrences with both action-value estimates available, define

$$\mathcal{H}_o^{\mathrm{adm}} = \begin{cases} \{g \in \mathcal{H}_o : \delta_{o,g} > \Delta_o\}, & \Delta_o > 0, \\ \mathcal{H}_o, & \Delta_o < 0 \ \wedge\ \widehat{r}_S(o) < \tau_S, \\ \varnothing, & \text{otherwise.} \end{cases} \tag{36}$$

Planning support and fallback. A Teacher target is retained only when the selected child has an associated positive-advantage Plan example that remains available for training after auxiliary-plan selection and weighting. Let

$$\mathcal{P}_o^+ = \{g : \text{a retained, trainable Plan from } o \text{ to } g \text{ has positive advantage}\}. \tag{37}$$

The final supported Teacher improvement is

$$d_o^{\mathrm{teach}} = \max_{g \in \mathcal{H}_o^{\mathrm{adm}} \cap \mathcal{P}_o^+} \delta_{o,g}, \tag{38}$$

when the intersection is nonempty. Operationally, a preferred child lacking retained planning support can be replaced by another child satisfying the same conditions; if none remains, the procedure returns to the base target, or leaves the occurrence without a target when the base target is absent.

This requirement couples a stronger continuation preference to an actual planning example directed toward the supporting child. It does not imply that the current policy has already learned to produce that Plan, but ensures that the timing refinement is accompanied by corresponding content supervision.

Example. Suppose the stopping pool has utilities $[0.95, 0.95, 0, 0]$, yielding $\widehat{Q}_S = 0.475$ and $\widehat{r}_S = 0.5$, while two child classes have mean utilities 0.85 and 0. The base continuation value is 0.425, giving $\Delta_o = -0.05$, but the stopping target is withheld because its accuracy is below 0.70. If the first child has three successful supporting returns and a retained positive-advantage Plan, it supplies a Teacher improvement of $0.85 - 0.475 = 0.375$, allowing a continuation target despite the negative base average. Supported child-level refinement supplies only 3–5% of accepted nonzero timing targets in our training runs; the remaining targets use the reliability-gated empirical MUC.

D.4 Final Targets, Weighting, and Update Direction

Combining the base-target and supported child-level refinement rules gives the final accepted target:

$$d_o = \begin{cases} d_o^{\text{teach}}, & \text{if a supported Teacher target remains available,} \\ d_o^{\text{base}}, & \text{if the base target is available,} \\ \perp, & \text{otherwise.} \end{cases} \tag{39}$$

The set $\mathcal{O}_T$ contains occurrences with accepted nonzero targets, and $N_T = |\mathcal{O}_T|$. For each such occurrence, define

$$y_o = \mathbf{1}[d_o > 0], \qquad w_o = \min(|d_o|, w_{\max}), \qquad w_{\max} = 0.25. \tag{40}$$

The weighted supervised component of the timing objective is

$$\mathcal{L}_{\text{timing}}^{\text{sup}} = -\frac{1}{\max(N_T, 1)} \sum_{o \in \mathcal{O}_T} w_o \left[y_o \log p_o + (1 - y_o) \log(1 - p_o) \right], \tag{41}$$

with zero contribution when $\mathcal{O}_T$ is empty.

Effect of weight truncation. For an unrefined base target $d_o = \Delta_o \neq 0$, the local regret bound yields

$$\frac{w_o}{|\Delta_o|} \mathcal{R}_o(\theta) \leq -w_o \log p_\theta(a_o^\star \mid o). \tag{42}$$

Thus, the weighted supervised term upper-bounds a correspondingly rescaled local regret; it does not necessarily upper-bound the original unscaled regret when $|\Delta_o| > w_{\max}$. Truncation preserves the preferred action while limiting the contribution of large estimated gaps.

Supervised update direction. The difference between the two action-content sequence scores is

$$z_o = \ell_\theta(o, \texttt{CONTINUE}) - \ell_\theta(o, \texttt{STOP}), \qquad p_o = \sigma(z_o). \tag{43}$$

For the unnormalized supervised term $\mathcal{L}_o^{\text{sup}} = -w_o[y_o \log p_o + (1 - y_o) \log(1 - p_o)]$, holding the estimated targets fixed gives

$$\frac{\partial \mathcal{L}_o^{\text{sup}}}{\partial z_o} = w_o(p_o - y_o). \tag{44}$$

This supervised component favors increasing the continuation score gap for a positive target and decreasing it for a negative target, with magnitude determined jointly by the target weight and the discrepancy between the current probability and the preferred action. The complete timing objective additionally includes the action-level KL regularization defined in Appendix E.

E Region-Specific Policy Optimization

Analysis combines question-level supervision with a knowledge-group-relative value signal, Plan/Search receives local credit relative to stopping, and Answer retains the question-level advantage, whereas Timing combines MUC-based supervision from Appendix D with action-level KL regularization. All advantages, action-value estimates, sampling-policy probabilities, and sample weights are held fixed during the actor updates associated with a rollout batch.

E.1 Question-Level Advantages

For each question x, we sample $G = 5$ main trajectories $\{\tau_{x,i}\}_{i=1}^{G}$ and evaluate their rewards $R_{x,i} = R(\tau_{x,i})$ using Appendix A. The question-level advantage is computed by centering rewards within the question and normalizing them by the sample standard deviation:

$$\begin{aligned} \overline{R}_x &= \frac{1}{G}\sum_{i=1}^{G} R_{x,i}, \\ s_x &= \sqrt{\frac{1}{G-1}\sum_{i=1}^{G}(R_{x,i}-\overline{R}_x)^2}, \\ A^{\mathrm{G}}_{x,i} &= \frac{R_{x,i}-\overline{R}_x}{s_x+\epsilon_{\mathrm{G}}}, \qquad \epsilon_{\mathrm{G}} = 10^{-6}. \end{aligned} \tag{45}$$

Only main trajectories participate in this question-level normalization; additional branches collected for local value estimation do not enlarge the group or change its reward statistics. For each main trajectory, $A^{\mathrm{G}}_{x,i}$ is assigned to its eligible Answer tokens and provides the trajectory-level component of its Analysis advantages. Plan/Search uses the local construction below, and Timing tokens are excluded from this trajectory-level policy-gradient objective.

E.2 Hybrid Advantages for Analysis

For a main-trajectory occurrence o in $\tau_{x,i}$ immediately following an Analysis segment, the local action-value estimates are abbreviated as $\widehat{Q}_C(o) = \widehat{Q}_{\mathrm{post}}(s_o, \texttt{CONTINUE})$ and $\widehat{Q}_S(o) = \widehat{Q}_{\mathrm{post}}(s_o, \texttt{STOP})$. When both estimates are available, the preceding Analysis segment is associated with the decision value

$$V_o^{\mathrm{analysis}} = \max\left\{\widehat{Q}_C(o), \widehat{Q}_S(o)\right\}. \tag{46}$$

The set $\mathcal{G}_A(k)$ contains main-trajectory occurrences in knowledge group k for which local Analysis comparison is enabled and both action-value estimates are available. For $o \in \mathcal{G}_A(k)$, where $k = \kappa(s_o)$, the group mean and local correction are

$$\begin{aligned} \bar{V}^{\mathrm{analysis}}(k) &= \frac{1}{|\mathcal{G}_A(k)|}\sum_{o'\in\mathcal{G}_A(k)} V_{o'}^{\mathrm{analysis}}, \\ a_A(o) &= \begin{cases} \mathrm{clip}\big(V_o^{\mathrm{analysis}} - \bar{V}^{\mathrm{analysis}}(k), -c_{\mathrm{loc}}, c_{\mathrm{loc}}\big), & |\mathcal{G}_A(k)| \geq 2, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{47}$$

with $c_{\mathrm{loc}} = 1$.

The final Analysis advantage combines the question-level signal with the clipped local correction, scaled by λ_A, and applies a second clipping operation to the combined value:

$$A_o^{\mathrm{analysis}} = \begin{cases} \mathrm{clip}\big(A^G_{x,i} + \lambda_A a_A(o), -c_A, c_A\big), & o \in \mathcal{G}_A(\kappa(s_o)), \\ A^G_{x,i}, & \text{otherwise}. \end{cases} \tag{48}$$

Here, c_A bounds the combined advantage, while occurrences without an enabled and available local correction retain the question-level advantage alone. The resulting advantage is assigned to the eligible tokens of the preceding Analysis segment.

E.3 STOP-Relative Advantages for Plan/Search

For an origin o, let $\mathcal{C}_{\mathrm{P}}(o)$ denote the successor groups with usable Plan/Search-associated values, and write $V_{o,g} = \bar{u}_o^{\texttt{CONTINUE}}(g)$ for the estimated post-decision utility of child g. A branch b originating at o and reaching child $g(b)$ receives an advantage relative to the utility of stopping at the same origin:

$$A^{\mathrm{Plan}}(b) = \mathrm{clip}\big(V_{o,g(b)} - B_{\mathrm{P}}(o), -c_{\mathrm{loc}}, c_{\mathrm{loc}}\big), \tag{49}$$

where the baseline is

$$B_{\mathrm{P}}(o) = \begin{cases} \widehat{Q}_S(o), & \text{if the STOP estimate is available,} \\ \dfrac{1}{|\mathcal{C}_{\mathrm{P}}(o)|} \displaystyle\sum_{g \in \mathcal{C}_{\mathrm{P}}(o)} V_{o,g}, & \text{otherwise, if } |\mathcal{C}_{\mathrm{P}}(o)| \geq 2, \\ \bot, & \text{otherwise.} \end{cases} \tag{50}$$

When the STOP estimate is available, this construction directly evaluates whether the evidence-acquisition branch improves upon immediate answering, including origins with only one observed usable child. When STOP evaluation is unavailable, the fallback instead provides a relative comparison among at least two children; a single child without a STOP baseline receives no Plan/Search policy-gradient target. Unlike Analysis, Plan/Search does not add the question-level advantage to its local advantage. These advantages are assigned to eligible main-trajectory Plan/Search segments and newly sampled auxiliary Plan/Search actions at the intervention boundary. Replayed actions used for value completion and subsequent auxiliary rollout segments contribute to return estimation without receiving direct content policy-gradient supervision.

At most $1{,}024$ auxiliary Plan/Search training sequences are retained per rollout batch, with selection distributed across origins and branches and priority given to positive Plan/Search samples supporting candidate Timing teachers. Value estimation continues to use all admissible sampled outcomes, including those whose planning sequences are not retained for actor training.

E.4 Token Weighting and Policy-Gradient Aggregation

A token-uniform objective can give greater weight to origins with more sampled branches, children with more repeated visits, or plans with longer text. To balance these factors within the retained Plan/Search training set, we distribute its total token mass equally across origins, then across children within each origin, branches within each child, and eligible tokens within each branch.

Let $\mathcal{O}_{\mathrm{P}}$ be the set of active origins after training-sample selection and masking, $\mathcal{C}^{+}_{\mathrm{P}}(o)$ the children with retained Plan/Search tokens, and $\mathcal{B}^{\mathrm{P}}_{o,g}$ the retained branches reaching child g. These sets include eligible main and auxiliary segments jointly. Define

$$O_{\mathrm{P}} = |\mathcal{O}_{\mathrm{P}}|, \qquad C_o = |\mathcal{C}^{+}_{\mathrm{P}}(o)|, \qquad B_{o,g} = |\mathcal{B}^{\mathrm{P}}_{o,g}|, \qquad T_{\mathrm{P}} = \sum_{o \in \mathcal{O}_{\mathrm{P}}} \sum_{g \in \mathcal{C}^{+}_{\mathrm{P}}(o)} \sum_{b \in \mathcal{B}^{\mathrm{P}}_{o,g}} L_b, \tag{51}$$

where $L_b > 0$ is the number of eligible Plan/Search tokens in branch b. For a nonempty training set, each such token receives weight

$$w^{\mathrm{P}}_{b,t} = \frac{T_{\mathrm{P}}}{O_{\mathrm{P}} C_o B_{o,g} L_b}, \qquad b \in \mathcal{B}^{\mathrm{P}}_{o,g}. \tag{52}$$

Proposition: mass preservation and hierarchical averaging. The weights in Eq. 52 preserve total Plan/Search token mass and induce an equally weighted average at each level of the origin–child–branch hierarchy. In particular,

$$\begin{aligned} \sum_{t=1}^{L_b} w^{\mathrm{P}}_{b,t} &= \frac{T_{\mathrm{P}}}{O_{\mathrm{P}} C_o B_{o,g}}, \\ \sum_{b \in \mathcal{B}^{\mathrm{P}}_{o,g}} \sum_{t=1}^{L_b} w^{\mathrm{P}}_{b,t} &= \frac{T_{\mathrm{P}}}{O_{\mathrm{P}} C_o}, \\ \sum_{g \in \mathcal{C}^{+}_{\mathrm{P}}(o)} \sum_{b \in \mathcal{B}^{\mathrm{P}}_{o,g}} \sum_{t=1}^{L_b} w^{\mathrm{P}}_{b,t} &= \frac{T_{\mathrm{P}}}{O_{\mathrm{P}}}, \end{aligned} \tag{53}$$

and summing over origins yields total mass T_{P}.

Proof. The weight is constant over the L_b eligible tokens of a branch, so summing over tokens cancels L_b in the denominator. Summing the resulting branch mass over the $B_{o,g}$ branches of a child then cancels $B_{o,g}$, and summing over the C_o children of an origin cancels C_o. Finally, the O_{P} origins each contribute $T_{\mathrm{P}}/O_{\mathrm{P}}$, establishing mass preservation. More generally, for arbitrary token losses $\ell_{b,t}$, direct substitution gives

$$\frac{1}{T_{\mathrm{P}}} \sum_{o,g,b,t} w^{\mathrm{P}}_{b,t} \ell_{b,t} = \frac{1}{O_{\mathrm{P}}} \sum_{o \in \mathcal{O}_{\mathrm{P}}} \frac{1}{C_o} \sum_{g \in \mathcal{C}^{+}_{\mathrm{P}}(o)} \frac{1}{B_{o,g}} \sum_{b \in \mathcal{B}^{\mathrm{P}}_{o,g}} \frac{1}{L_b} \sum_{t=1}^{L_b} \ell_{b,t}, \tag{54}$$

which establishes the hierarchical averaging identity. The weighting equalizes loss mass at each level without requiring equal branch counts or equal sequence lengths.

Unified policy-gradient objective. The combined collection $\mathcal{T}_{\mathrm{PG}}$ contains all eligible Analysis, Plan/Search, and Answer content tokens after selection and masking. Each token receives its regional advantage A_t and weight ω_t, with unit weights for Analysis and Answer and hierarchical weights $w^P_{b,t}$ for Plan/Search. Timing action content is excluded from this collection. Using the rollout policy π_{old}, frozen for the current batch, the token-level likelihood ratio is

$$\rho_t(\theta) = \frac{\pi_\theta(y_t \mid h_t)}{\pi_{\mathrm{old}}(y_t \mid h_t)}. \tag{55}$$

All eligible tokens contribute to a single clipped policy-gradient objective:

$$\mathcal{L}_{\mathrm{PG}} = -\frac{1}{Z_{\mathrm{PG}}} \sum_{t \in \mathcal{T}_{\mathrm{PG}}} \omega_t \min(\rho_t(\theta) A_t,\ \mathrm{clip}(\rho_t(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\, A_t)\,, \tag{56}$$

Here, ϵ_{clip} controls likelihood-ratio clipping, and $Z_{\mathrm{PG}} = \sum_{t \in \mathcal{T}_{\mathrm{PG}}} \omega_t$ normalizes the weighted loss over the combined token collection. An empty collection contributes zero loss.

E.5 Action-Level and Reference-Policy Regularization

We distinguish regularization of the two-action Timing distribution from regularization against the fixed reference language model. Let $\mathcal{O}_{\mathrm{K}}$ contain eligible main-trajectory decision occurrences for which both canonical action sequences can be scored, and let $N_{\mathrm{K}} = |\mathcal{O}_{\mathrm{K}}|$. Using the sequence-normalized action probabilities defined in the main text, we impose

$$\mathcal{L}_{\mathrm{AKL}} = \frac{1}{\max(N_{\mathrm{K}}, 1)} \sum_{o \in \mathcal{O}_{\mathrm{K}}} \sum_{a \in \{\texttt{STOP},\texttt{CONTINUE}\}} p_{\mathrm{old}}(a \mid o) \log \frac{p_{\mathrm{old}}(a \mid o)}{p_\theta(a \mid o)}. \tag{57}$$

The old action distribution is frozen once per rollout batch and remains unchanged across its optimizer updates. Membership in $\mathcal{O}_K$ does not require a nonzero Timing target, allowing regularization at eligible decisions without directional supervision. The complete timing objective combines this regularizer with the supervised component from Appendix D:

$$\mathcal{L}_{\mathrm{timing}} = \mathcal{L}^{\mathrm{sup}}_{\mathrm{timing}} + \gamma_{\mathrm{AKL}} \mathcal{L}_{\mathrm{AKL}}. \tag{58}$$

The two components retain their respective occurrence sets and normalizations, with γ_{AKL} controlling the strength of action-level regularization relative to direct supervision.

The reference model π_{ref} is fixed to the initial policy throughout training. For a token with reference-regularization weight m^{reg}_t, define

$$d_t = \log \pi_{\mathrm{ref}}(y_t \mid h_t) - \log \pi_\theta(y_t \mid h_t), \qquad \ell^{\mathrm{ref}}_t = \mathrm{clip}(\exp(d_t) - d_t - 1, -10, 10)\,. \tag{59}$$

The corresponding reference penalty is

$$\mathcal{L}_{\mathrm{KL}} = \frac{\sum_t m^{\mathrm{reg}}_t \ell^{\mathrm{ref}}_t}{\sum_t m^{\mathrm{reg}}_t}, \tag{60}$$

where an empty mask contributes zero. This sampled-token surrogate is distinct from the binary action KL in Eq. 57; it uses a different reference distribution, token support, and coefficient, and is applied as an actor loss without an additional deduction from the task reward.

Table 5: Key experimental settings shared by the 3B and 7B models.

Parameter	Value
Learning rate	1×10^{-6}
Training epochs	1
Learning-rate warmup ratio	0.05
Questions per training batch	128
Main trajectories per question	5
Training temperature	1.0
Training top-p	0.95
Maximum initial input length	8,192 tokens
Maximum model input length	24,576 tokens
Maximum generation length per interaction round	2,048 tokens
Maximum retrieval observation length	2,048 tokens
Maximum search calls per trajectory	5
Training hardware per run	8× NVIDIA A100

We also include conditional token entropy,

$$\mathcal{H} = \frac{\sum_t m_t^{\text{reg}} \left[-\sum_{v \in \mathcal{V}_{\text{tok}}} \pi_\theta(v \mid h_t) \log \pi_\theta(v \mid h_t) \right]}{\sum_t m_t^{\text{reg}}}, \tag{61}$$

where $\mathcal{V}_{\text{tok}}$ is the tokenizer vocabulary and an empty mask again contributes zero.

Regularization masks are defined separately from policy-gradient masks. Main sampled response tokens can retain reference regularization and entropy even when they have no content policy-gradient target, while auxiliary Plan/Search sequences are regularized only on their eligible auxiliary spans and synthetic structural examples only on their supervised structural spans.

E.6 Structural Auxiliary Supervision

Structural supervision encourages the model to express its decisions and content using the prescribed interaction format. We collect canonical structural-tag targets from eligible main trajectories together with a bounded set of short structural-completion targets, restricting supervision to identifiable structural tokens and excluding semantic content and EOS from the tag cross-entropy. If these two collections are $\mathcal{S}_v$ and $\mathcal{S}_r$, with sizes N_v and N_r, their pooled objective is

$$\mathcal{L}_{\text{tag}} = -\frac{1}{\max(N_v + N_r, 1)} \sum_{t \in \mathcal{S}_v \cup \mathcal{S}_r} \log \pi_\theta(y_t^* \mid h_t). \tag{62}$$

Pooling the token counts preserves a common cross-entropy scale: when the two collections are nonempty, this is equivalently $(N_v \mathcal{L}_v + N_r \mathcal{L}_r)/(N_v + N_r)$, where $\mathcal{L}_v$ and $\mathcal{L}_r$ are their separate token means. Thus, a small collection of structural-completion targets is not automatically assigned the same aggregate weight as a much larger collection of main-trajectory tags.

The structural auxiliary objective is therefore

$$\mathcal{L}_{\text{protocol}} = \mathcal{L}_{\text{tag}}, \qquad \lambda_P = 0.01. \tag{63}$$

F Implementation Details

We train the 3B and 7B policies for one epoch each using eight NVIDIA A100 GPUs, with the shared experimental settings summarized in Table 5. Each training batch contains 128 questions with five main trajectories sampled per question, excluding additional matched branches. Generation limits apply to individual interaction rounds, while the retrieval observation limit applies to the combined passages returned by a single query. Details of matched branch resampling, reward computation, and policy optimization are provided in Appendices C, A, and E, respectively.

G SEARCH BEHAVIOR METRICS

We evaluate answer accuracy using normalized Exact Match (EM), with the answer-matching procedure described in Appendix A, and characterize search behavior using Average Search Calls per Question (ASQ), Over-Searching Rate (OSR), and Under-Searching Rate (USR), following DAS (Zhang et al., 2026). Task accuracy is measured by EM, whereas the behavioral diagnostics use a token-F1 threshold of 0.9.

Average search calls. Let $\mathcal{D}$ denote the evaluated question set and T_i the number of retrieval calls executed in the main trajectory for question i. We compute

$$\mathrm{ASQ} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} T_i. \tag{64}$$

Each executed retrieval query counts as one call regardless of the number of passages returned, and questions answered without retrieval contribute zero. Training-time resampling and diagnostic generations are excluded from this count.

Forced-answer diagnostics. At each CONTINUE decision preceding an executed retrieval, we retain the available context and the current evidence analysis, replace continuation with STOP, and generate an answer without additional retrieval. Each state is evaluated once using greedy decoding with a maximum of 128 answer tokens, and the diagnostic answer is not inserted into the main trajectory. Diagnostic correctness is determined by the maximum token-F1 over reference answers after lowercasing, removing punctuation and English articles, and normalizing whitespace. The diagnostic records preserve the order of the evaluated decisions and include a terminal answering entry when such a decision is recorded.

Over-searching and under-searching. We follow the counting rules of the official DAS evaluation implementation. For question i, M_i is the number of entries in its ordered diagnostic record, and $\mathcal{D}_{\mathrm{diag}} = \{i \in \mathcal{D} : M_i > 0\}$ contains questions with nonempty records. Over-searching is evaluated on entries indexed by $\mathcal{J}_i = \{1, \ldots, M_i - 1\}$, excluding the last recorded entry. We write $\tilde{y}_{i,j}$ for the forced answer at entry j, $\hat{y}_i$ for the final answer from the main trajectory, and $F_i(y)$ for the maximum normalized token-F1 against the reference answers. The question-level indicators are

$$o_i = \mathbf{1}[\exists j \in \mathcal{J}_i : \; F_i(\tilde{y}_{i,j}) \geq 0.9], \tag{65}$$

$$u_i = (1 - o_i)\, \mathbf{1}[F_i(\hat{y}_i) < 0.9], \tag{66}$$

where $o_i = 0$ when $\mathcal{J}_i$ is empty. The aggregate rates are

$$\mathrm{OSR} = \frac{1}{|\mathcal{D}_{\mathrm{diag}}|} \sum_{i \in \mathcal{D}_{\mathrm{diag}}} o_i, \qquad \mathrm{USR} = \frac{1}{|\mathcal{D}_{\mathrm{diag}}|} \sum_{i \in \mathcal{D}_{\mathrm{diag}}} u_i. \tag{67}$$

Over-searching takes precedence regardless of final-answer correctness, making the two categories mutually exclusive. OSR identifies cases in which an earlier forced answer already meets the diagnostic correctness criterion, while USR captures final-answer failures not assigned to that category. A zero-search trajectory with a single recorded answering decision has $\mathcal{J}_i = \varnothing$ and $o_i = 0$, while its final-answer correctness determines u_i. Questions with empty diagnostic records are excluded from OSR and USR but remain included in ASQ.

H QUALITATIVE CASE STUDIES

We present two examples from 2WikiMultihopQA illustrating complementary timing behaviors: continuing to acquire missing evidence and stopping without an additional confirmatory search. Tables 6 and 7 summarize the observed trajectories, retaining key queries, evidence, decisions, and final answers.

Avoiding premature stopping. In Table 6, both policies initially retrieve evidence identifying *Crazy Heart* as an American film, while the country of origin of *Dragon Eyes* remains unsupported. The base policy assumes that the latter is a Hong Kong film and stops with an incorrect answer, whereas Action-Timing recognizes the missing information and searches specifically for it, obtaining evidence that both films are American and answering correctly after one additional search.

Table 6: **Avoiding premature stopping.** Action-Timing acquires the missing evidence before answering. Evidence and analysis are condensed from the recorded trajectories.

Question: Are Crazy Heart and Dragon Eyes both from the same country?		
	Base	**Action-Timing**
First query	country of origin for Crazy Heart	country of origin for Crazy Heart film
Evidence after the first search: Both policies retrieve evidence that *Crazy Heart* is a 2009 **American** drama film, but neither retrieves country-of-origin evidence for *Dragon Eyes*.		
Analysis	Assumes that *Dragon Eyes* is a Hong Kong film despite the absence of direct evidence.	Recognizes that the retrieved evidence does not establish the country of origin of *Dragon Eyes*.
Decision	`STOP`	`CONTINUE`
Second query	—	country of origin for Dragon Eyes film
New evidence	—	*Dragon Eyes* is a 2012 **American** martial arts film.
Final answer	No (**incorrect**)	Yes (**correct**)
Search calls	1	2

Avoiding redundant confirmation. In Table 7, both policies receive identical retrieved passages after their first search, identifying Tim McCann as the director of *Desolation Angels* and explicitly naming his alma mater. The base policy treats the graduation institution as uncertain and performs another search, whereas Action-Timing recognizes the explicit relationship and stops with the correct answer. The additional retrieval supplies further biographical details but confirms an institution already identified in the initial evidence.

Table 7: **Avoiding redundant confirmation.** Action-Timing answers correctly with one fewer search after recognizing the answer in the initial evidence. Evidence and analysis are condensed from the recorded trajectories.

Question: Where did the director of film Desolation Angels (1995 Film) graduate from?		
	Base	**Action-Timing**
First query	director of Desolation Angels 1995	director of Desolation Angels 1995 film
Evidence after the first search (identical passages): Tim McCann directed *Desolation Angels* in 1995 and is a professor of film at "his **alma mater**, the State University of New York at Purchase."		
Analysis	Describes the institution as "likely where he graduated from" and seeks confirmation.	Recognizes that the passage explicitly identifies the institution as his alma mater.
Decision	`CONTINUE`	`STOP`
Second query	Tim McCann alma mater	—
New evidence	Confirms that McCann graduated from the institution's film school in 1987.	—
Final answer	State University of New York at Purchase (**correct**)	State University of New York at Purchase (**correct**)
Search calls	2	1

I Effect of the Planning Reference

To examine the reference used for planning credit, we replace the immediate-stopping value with the mean utility across continuation groups while retaining the full timing objective. As shown in Table 8, the stopping reference improves average EM by 3.1 percentage points and reduces OSR and USR by 0.9 and 2.9 percentage points, respectively, while ASQ increases slightly from 2.13 to

Table 8: **Effect of the reference used for planning credit.** Both variants retain the full timing objective and differ only in the reference subtracted from continuation-group utilities. Results use Qwen2.5-7B-Instruct and are macro-averaged across the seven QA benchmarks.

Planning reference	EM (%) ↑	ASQ	OSR (%) ↓	USR (%) ↓
STOP value (ours)	**45.9**	2.22	**25.2**	**44.2**
CONTINUE-group mean	42.8	2.13	26.1	47.1

Table 9: **Effect of the interaction-cost penalty.** Results use Qwen2.5-7B-Instruct and are macro-averaged across seven QA benchmarks. The coefficient λ_c is varied consistently in main-trajectory rewards and post-decision utilities, with all other training settings unchanged.

Variant	λ_c	EM (%) ↑	ASQ	OSR (%) ↓	USR (%) ↓
Action-Timing (Full)	0.05	**45.9**	2.22	**25.2**	**44.2**
w/o cost penalty	0	45.5	2.89	29.3	44.9

2.22. These results support evaluating continuation plans relative to immediate stopping, showing that this reference improves performance even when direct timing supervision is already present.

J Effect of the Interaction-Cost Penalty

We remove the interaction-cost penalty by setting $\lambda_c = 0$ in both main-trajectory rewards and post-decision utilities, with all other training settings unchanged. As shown in Table 9, the default penalty reduces ASQ from 2.89 to 2.22, a reduction of approximately 23.2%, and lowers OSR from 29.3% to 25.2%, while maintaining comparable EM (45.9% versus 45.5%) and slightly reducing USR (44.2% versus 44.9%). These results support incorporating interaction costs into the training utilities to reduce search usage and over-searching while preserving answer accuracy.