Beyond Outcome Rewards: Process-Aware Optimization for Search Agents

Yiwei Dai[1], Hengyi Cai[2], Hui Wu[2], Yili Wang[1], Han Xu[2], Minlan Shao[1]
Yuchen Li[2], Shuaiqiang Wang[2], Xin Wang[1,†], Yi Chang[1,†], Dawei Yin[2]

[1]School of Artificial Intelligence, Jilin University
[2]Baidu Inc.
daiyw25@mails.jlu.edu.cn, {xinwang,yichang}@jlu.edu.cn

Abstract

Search agents based on large language models address complex information-seeking tasks by iteratively issuing queries, retrieving evidence, and integrating information. Effective training of such agents requires supervision beyond final-answer correctness, as outcome-level rewards cannot distinguish efficient evidence acquisition from redundant or misdirected search. Recent process-level methods introduce intermediate supervision, but typically apply uniform signals across trajectories. However, such uniformity overlooks the state-dependent nature of process quality, where the value of an intermediate decision depends on the current evidence state. This motivates a systematic analysis of how process quality varies across trajectories with different evidence states and final outcomes. To this end, we analyze search trajectories on BrowseComp-Plus and xBench, revealing two consistent patterns: successful trajectories often suffer from post-alignment inefficiency (redundant searches), whereas failed trajectories typically struggle with misdirected exploration and insufficient coverage. Motivated by these findings, we propose **P**rocess-**A**ware **G**rouped r**E**ward optimization (**PAGE**), a framework for search agents that provides a concrete instantiation of state-dependent process supervision. **PAGE** partitions trajectories according to final outcomes, assigns differentiated process rewards to successful and failed groups, and integrates them with outcome supervision at either the reward or advantage level. Experiments on three backbones across six multi-hop QA and web-search benchmarks show that **PAGE** consistently improves answer accuracy over outcome-only baselines.

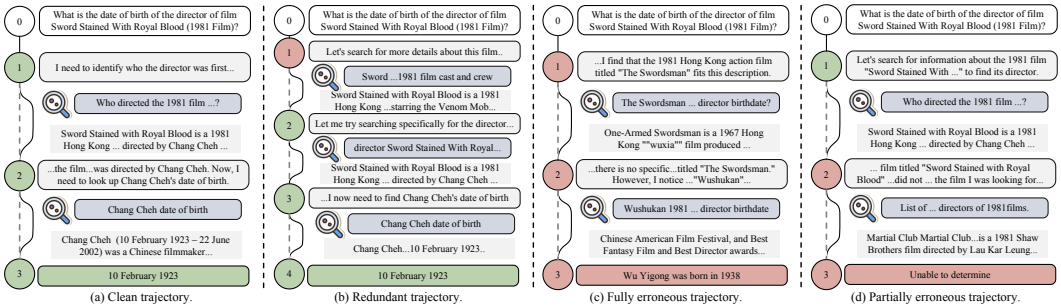
1 Introduction

Search agents based on large language models (LLMs) tackle information-seeking tasks by interleaving reasoning, query construction, and tool calls to an external retriever or browser [Singh et al., 2025, Chen et al., 2025c,b]. Each step in an agent trajectory involves a decision: *whether* to search, *what* query to issue, *when* to stop, and *how* to commit to an answer. However, these intermediate decisions are rarely directly supervised. Current training pipelines [Jin et al., 2025, Feng et al., 2025] therefore largely rely on outcome-level signals, such as filtering trajectories by final-answer correctness in supervised fine-tuning or assigning reinforcement learning rewards based on final-answer accuracy.

This outcome-only view assumes that final-answer correctness faithfully reflects the quality of the underlying search process. However, this assumption does not hold in multi-step search. As shown in Figure 1, trajectories with the same outcome can follow very different processes. A correct trajectory

[†]Corresponding authors.
[*]Work done during an internship at Baidu.

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



(a) Clean trajectory. (b) Redundant trajectory. (c) Fully erroneous trajectory. (d) Partially erroneous trajectory.

Figure 1: Outcome supervision fails to distinguish trajectories with varying qualities.

may find key evidence early and stop, or reach the answer only after redundant queries and drift. A failed trajectory may miss a small amount of evidence, or deviate from the correct direction early. As a result, outcome supervision assigns the same feedback to trajectories that differ in efficiency, stability, and error correction.

To address this limitation, recent works [Zheng et al., 2025, Wang et al., 2025, Yu et al., 2025] have introduced process-level supervision. However, most process signals are defined uniformly across trajectories, such as penalties on length or tool usage. This design overlooks a key property of search agents: *process quality depends on the current evidence state*. The same behavior may require different optimization depending on context. Continuing to search is redundant when sufficient evidence has been obtained, but necessary when evidence is still missing. Query reformulation may correct the trajectory or introduce further drift. Therefore, uniform process supervision can provide noisy or even conflicting signals.

To examine this issue, we conduct a process-level analysis on BrowseComp-Plus [Chen et al., 2025c] and xBench [Chen et al., 2025a]. We analyze two observable dimensions of search behavior: search actions and query evolution. These dimensions capture when the agent searches and how it specifies retrieval targets, while avoiding unverifiable assumptions about internal reasoning. Our analysis reveals two distinct patterns: successful trajectories often refine queries toward relevant evidence, but continue searching after sufficient evidence is obtained, leading to redundancy. In contrast, failed trajectories tend to suffer from insufficient coverage and unstable query evolution, where early errors are not corrected. These results suggest that deficiencies differ across trajectory types.

Motivated by this observation, we propose a Process-Aware Grouped rEward optimization framework (**PAGE**) for search agents, which provides a concrete instantiation of state-dependent process supervision by assigning differentiated process rewards to different trajectory groups. Specifically, **PAGE** partitions trajectories into successful and failed groups. For successful trajectories, it encourages early evidence alignment and penalizes redundant search. For failed trajectories, it encourages broader evidence acquisition and beneficial query reformulation. We instantiate these rewards in a shared semantic space using a frozen sentence encoder, with embedding similarity serving as the process-quality proxy. We further study two strategies for integrating process and outcome signals: reward-level merging and advantage-level merging. Experiments on multi-hop QA and web-search benchmarks validate the effectiveness of **PAGE** across different models and training settings.

Our contributions are summarized as follows:

- In Section 4, we identify the state-dependent nature of process quality in search agents and conduct a process-level analysis, revealing how successful and failed trajectories exhibit distinct behavioral patterns and dominant deficiencies.
- In Section 5, we propose **PAGE**, a process-aware grouped reward optimization framework that instantiates state-dependent process supervision. It assigns differentiated process rewards to successful and failed trajectories, then integrates them with outcome signals.
- In Section 6, we evaluate **PAGE** across multiple benchmarks, backbones and training settings, showing consistent improvements in answer accuracy.

2 Related Work

Search Agents. Search agents extend single-step RAG [Gao et al., 2023, Lewis et al., 2020] to multi-step interactions with external tools, iteratively generating queries, inspecting evidence, and deciding when to continue, reformulate or stop [Li et al., 2025b, 2026]. The central challenge is to optimize search decisions, with methods falling into three categories. Prompt-based search agents rely on manually designed interaction patterns. ReAct [Yao et al., 2022] uses a think-search-observe format, while Search-o1 [Li et al., 2025a] refines noisy results with interaction history. These methods are flexible but depend on model capability and prompts. SFT-based search agents [Yao et al., 2026, Ye et al., 2025, Lee et al., 2025, Liu et al., 2025c, Zhao et al., 2026] learn search behavior from annotated or synthesized trajectories. However, their generalization depends on the coverage and quality of trajectory data. RL-based search agents [Gao et al., 2025, Xi et al., 2025, Mei et al., 2025, Yue et al., 2026, Shi et al., 2025, Qiao et al., 2025, Jin et al., 2025, Wei et al., 2026] formulate search as a sequential decision problem and optimize multi-step interactions through reinforcement learning. Most of them use final-answer correctness as the main reward. While this enables policy improvement without dense annotations, it does not distinguish efficient evidence acquisition from redundant or misdirected search.

Reward Design for LLM Agents. Reward design provides optimization signals for training search agents. Existing methods fall into three categories. Outcome-level rewards [Jin et al., 2025, Feng et al., 2025] evaluate only the final answer. They are easy to compute but treat trajectories with different process quality as equivalent. Constraint-based rewards [Gao et al., 2025, Wang et al., 2025, Song et al., 2025, Dong et al., 2025, Liu et al., 2025b] enforce valid interaction behavior, such as tool usage, query format, and step limits. They improve stability but do not directly model information quality or search decisions. Process-level rewards evaluate intermediate behavior. Common approaches include reward-model-based methods [Deng et al., 2025, Xu et al., 2025], reference-based methods [Qian et al., 2026, Zheng et al., 2025], provenance-based credit assignment [Xu et al., 2026], and LLM-assisted evaluation [Yu et al., 2025, 2026, Wu et al., 2025b, Wen et al., 2026]. Recent work further explores differentiated process supervision before and after successful reasoning [Liu et al., 2026b]. While these methods provide fine-grained supervision, they primarily focus on reasoning progress, action-level credit assignment, or general process evaluation. In contrast, our analysis reveals distinct evidence-acquisition deficiencies in successful and failed search trajectories. This motivates our grouped reward design, which assigns outcome-conditioned process signals to optimize search efficiency, evidence coverage, and query reformulation.

3 Problem Formulation

3.1 Task Definition

We study search-agent question answering, where an LLM interacts with an external search environment over multiple steps. Let π denote the agent policy and $\mathcal{E}$ the search environment. Given a question $x \in \mathcal{X}$ and a step budget T_{max}, the interaction induces a trajectory

$$\tau \sim \mathcal{F}(x; \pi, \mathcal{E}, T_{max}), \tag{1}$$

from which the final answer is extracted as

$$\hat{y} = \mathrm{Ans}(\tau). \tag{2}$$

3.2 Interaction Protocol

To support unified analysis across implementations (e.g., MCP-based tool use or ReAct-style prompting), we represent each interaction trajectory as step-level transitions:

$$\tau = \{(s_t, a_t, o_t)\}_{t=1}^{T}, \tag{3}$$

where $T \leq T_{\max}$ is the actual trajectory length, s_t is the interaction state, a_t is the action, and o_t is the observation. The action space is defined as

$$a_t \in \{\texttt{think},\ \texttt{search}(q_t),\ \texttt{answer}\}. \tag{4}$$

At each step t, the agent samples $a_t \sim \pi(\cdot \mid s_t)$. If $a_t = \texttt{search}(q_t)$, the environment returns evidence $\mathcal{D}_t = \mathcal{E}(q_t)$ as the observation o_t, where $\mathcal{D}_t$ denotes retrieved documents or top-k web pages. Otherwise, $o_t = \varnothing$. The state is updated as $s_{t+1} = \Phi(s_t, a_t, o_t)$. The process terminates when the agent outputs `answer` or reaches $T_{\max}$.

4 Process-Level Analysis of Search Agents

To examine the state-dependent nature of process quality, we conduct process analysis of search-agent trajectories. We analyze observable interaction signals, including search actions, queries, retrieved documents, and final outputs, to compare successful and failed trajectories. This avoids unverifiable assumptions about reasoning. Based on these signals, we introduce a Process-Level Observability Framework (POF) to characterize group-specific behavioral patterns and optimization needs.

4.1 Experimental Settings

We conduct process-level analysis on two deep search benchmarks, BrowseComp-Plus [Chen et al., 2025c] and xBench [Chen et al., 2025a]. These benchmarks provide complementary settings: BrowseComp-Plus offers official trajectories with a local retriever for reproducible and fine-grained analysis, while xBench uses online search systems and reflects deployment-time interactions with less detailed process information. On BrowseComp-Plus, we use the official trajectories of GPT-5 [Singh et al., 2025] and o3 [OpenAI, 2025] under two retrieval settings, qwen3-embed-8b [Zhang et al., 2025] and BM25 [Wikipedia contributors, 2025]. On xBench, we execute GPT-5 [Singh et al., 2025], o3 [OpenAI, 2025], gemini-3-pro-preview [Google DeepMind, 2025b] and gemini-3-flash-preview [Google DeepMind, 2025a], each paired with its official online search system. Since closed-source models do not expose verifiable internal reasoning, we treat trajectories as unit of analysis and use only observable signals, including actions, queries, retrieved results and final outputs.

4.2 Behavior Definition

We organize process-level behaviors into three dimensions: Search Strategy, Query Reformulation and Evidence Synthesis. These dimensions correspond to three core questions in multi-step search, namely when to search, what to retrieve and how to use retrieved evidence. Search Strategy includes *Efficient Search*, *Overuse Search*, *Shallow Search*, *Redundant Exploration* and *Delayed Discovery*. Query Reformulation includes *Direct Query*, *Entity Adjustment*, *Constraint Refinement*, *Underspecified Query* and *Reformulation Drift*. Evidence Synthesis includes *Complete Synthesis*, *Incomplete Synthesis*, *Misaligned Synthesis* and *Unstructured Aggregation*. These labels capture dominant patterns in observable search behavior. Full definitions are provided in Appendix B.

4.3 Process-Level Observability Framework

We use POF to analyze search-agent behavior under limited observability. It follows a coarse-to-fine pipeline that derives rule-based anchors from observable signals and uses them to guide LLM-based semantic annotation for comparing successful and failed trajectories.

Stage 1: Rule-Based Process Anchors. From the full trajectory τ, we extract the search-step subsequence $\{q_i, \mathcal{D}_i\}_{i=1}^{K}$, where K is the number of search actions, q_i is the i-th search query, and $\mathcal{D}_i$ is the retrieved document set. We construct rule-based signals from these observable traces. First, we define the answer matching signal as $s_{\text{ans}} = \mathbb{I}[\hat{y} \approx y^*]$, where $\hat{y} = \text{Ans}(\tau)$ is the predicted answer, and y^* is the reference answer. We measure query redundancy by

$$r_{\text{rep}} = 1 - \frac{|\text{unique}(q_1, \ldots, q_K)|}{K}. \tag{5}$$

When gold supporting documents $\mathcal{D}^*$ are available, we compute coverage as

$$c = \frac{\left|\left(\bigcup_{i=1}^{K} \mathcal{D}_i\right) \cap \mathcal{D}^*\right|}{|\mathcal{D}^*|}. \tag{6}$$

We also record the first-hit step t_{first} and full-coverage step t_{full}, and define post-coverage search depth as $\Delta_{\text{over}} = K - t_{\text{full}}$. Based on these signals, we derive coarse anchors, such as low coverage and over-search. These anchors guide subsequent semantic annotation.

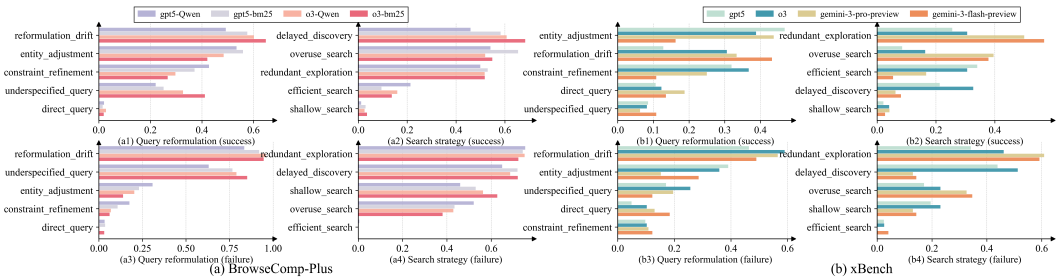


Figure 2: Distribution of process-level behavior patterns on BrowseComp-Plus and xBench.

Stage 2: LLM-Based Semantic Annotation. In the second stage, we use DeepSeek-V3.2 [Liu et al., 2025a] as a structured annotator with a fixed prompt and predefined schema. The annotator takes the question, predicted answer, search trajectory, retrieved evidence, visible reasoning text, and Stage 1 anchors as input. It assigns behavior labels along three dimensions: Search Strategy, Query Reformulation, and Evidence Synthesis. This step complements rule-based anchors with semantic annotations grounded in explicit trajectory evidence. We manually verify a random subset of annotations to assess labeling reliability.

4.4 Empirical Analysis

4.4.1 Successful Trajectories

Observation. At `Query Reformulation`, `entity_adjustment` and `constraint_refinement` are more frequent in successful trajectories. For `entity_adjustment`, the average proportion is 0.500 in the success group and 0.212 in the failure group, with a difference of -0.287 and a combined Fisher p-value of 3.700×10^{-63}. For `constraint_refinement`, the average proportion is 0.341 in the success group and 0.101 in the failure group, with a difference of -0.240 and a combined Fisher p-value of 2.556×10^{-57}. Both remain significant after Benjamini–Hochberg correction and are consistent across all four models. At the `Search Strategy`, `overuse_search` is more frequent in successful trajectories, with proportions 0.566 vs. 0.423, difference -0.143, and p-value 4.114×10^{-18}. This trend remains significant and shows $4/4$ directional consistency. Meanwhile, `efficient_search` is more prevalent in successful trajectories, while `overuse_search` remains common, indicating persistent redundancy.

Analysis. These results suggest that successful trajectories rarely rely on one-shot accurate retrieval. Instead, they often reach relevant evidence through gradual query correction. The higher prevalence of `entity_adjustment` and `constraint_refinement` indicates that models frequently revise entities or constraints before aligning with the true information need. At the same time, the prevalence of `overuse_search` shows that models often continue searching after useful evidence has been gathered. Thus, successful trajectories are often self-correcting, but still inefficient.

Pattern 1 (Late Alignment with Inefficient Search): Successful trajectories show late alignment via iterative refinement and inefficient post-retrieval search.

4.4.2 Failed Trajectories

Observation. At the `Query Reformulation` level, `underspecified_query` and `reformulation_drift` are more frequent in failed trajectories. For `underspecified_query`, the average proportion is 0.720 in the failure group and 0.302 in the success group, with a difference of $+0.418$. For `reformulation_drift`, the average proportion is 0.867 vs. 0.580, with a difference of $+0.288$. The combined Fisher p-values are 1.056×10^{-123} and 8.349×10^{-71}, respectively. Both remain significant after Benjamini–Hochberg correction and are consistent across all four models. At the `Search Strategy` level, `shallow_search` and `redundant_exploration` are more prominent in failed trajectories. For `shallow_search`, the proportions are 0.509 vs. 0.026, with a difference of $+0.483$. For `redundant_exploration`, the proportions are 0.707 vs. 0.516, with a difference of $+0.192$. The combined Fisher p-values are 4.164×10^{-248} and 2.545×10^{-25}. Both remain significant after correction and are consistent across all four models.

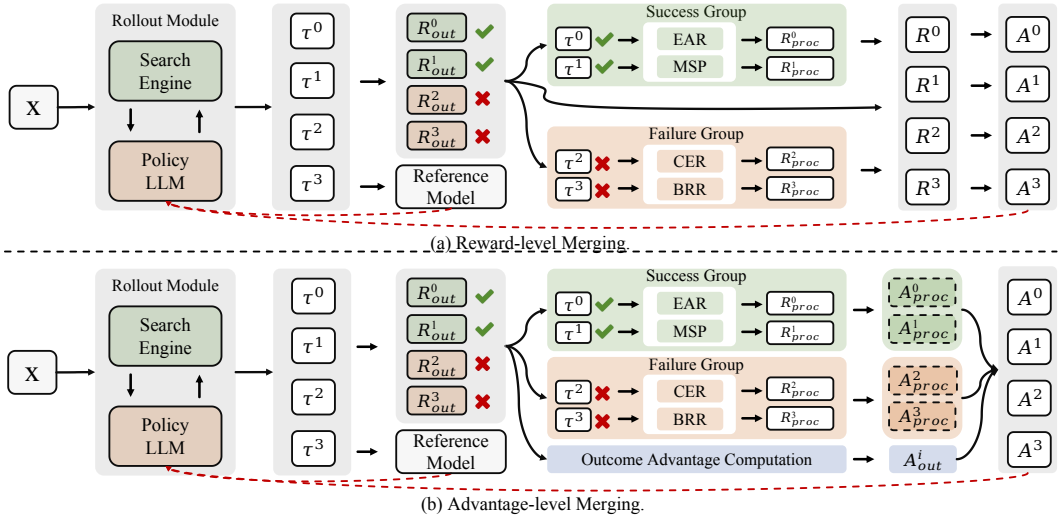


Figure 3: The framework of **PAGE**.

Analysis. These results show that failed trajectories reflect a structured failure pattern rather than merely insufficient search depth. The prevalence of `underspecified_query` suggests that models often fail to specify key entities, constraints, or qualifiers at the beginning of search. The prevalence of `reformulation_drift` further indicates that such early deviations are not corrected, but accumulate during later query reformulations. Meanwhile, `shallow_search` indicates premature stopping, while `redundant_exploration` reflects repeated low-yield exploration in misaligned directions. Thus, failed trajectories are mainly characterized by insufficient coverage and misdirected exploration.

Pattern 2 (Insufficient and Misdirected Exploration): Failed trajectories combine shallow search and unstable queries, leading to insufficient coverage and misdirected exploration.

Cross-benchmark support. We observe the same overall trend on xBench in Figure 2. Although xBench differs from BrowseComp-Plus in label distribution and task setup, successful trajectories still exhibit persistent query refinement and redundant search, while failed trajectories show higher proportions of shallow search, underspecified queries, and persistent drift. This suggests that the two dominant patterns are not specific to a single benchmark or retrieval configuration, but reflect common process characteristics of search-based agents across different task settings.

5 Method

In this section, we present Process-Aware Grouped rEward optimization (**PAGE**), as illustrated in Figure 3. **PAGE** is built on the observation that process quality is state-dependent: the value of a search decision depends on the evidence already acquired by the trajectory. Since the latent evidence state is only partially observable, **PAGE** uses final outcomes as a coarse proxy to separate trajectories into success and failure groups. It then assigns group-specific process rewards to encourage efficient evidence use in successful trajectories and coverage expansion in failed trajectories.

5.1 Embedding-based Process Modeling

In search-based question answering, process quality depends not only on whether each action is locally reasonable, but also on whether the trajectory progressively approaches the true information need. However, explicit process supervision for such progress is often unavailable or difficult to verify [Liu et al., 2026a]. We therefore approximate the latent evidence state with embedding-based process modeling. Specifically, we use a frozen pretrained language model $e(\cdot)$ as a semantic encoder to estimate trajectory-level evidence relevance. The frozen encoder provides a stable semantic space, avoids representation drift, and decouples process evaluation from policy optimization. Formally, given an input question x and the reference answer y, we define the reference representation as

$$z = e(x \oplus y), \tag{7}$$

where $\oplus$ denotes concatenation. We use $x \oplus y$ rather than x alone to measure whether retrieved evidence supports the correct answer, rather than being only superficially related to the question. The reference answer y is used only for reward construction during training and is never provided to the policy at inference time, avoiding answer leakage at test time.

For a trajectory with T search steps, let q_t denote the query at step t, and let D_t denote the retrieved result returned at that step. We define the query representation and retrieved-evidence representation: $u_t = e(q_t)$ and $v_t = e(D_t)$. We further define the prefix relevance up to step t as

$$r_t = \max_{i \le t} \cos(v_i, z), \tag{8}$$

which measures the highest semantic relevance to the reference information observed so far. Crucially, the prefix relevance r_t is computed exclusively using the external retrieved evidence rather than the model's generated text. Compared with step-wise similarity, r_t better captures accumulated evidence quality in multi-step search. This makes r_t a potential-like quantity for reward shaping: the increment $r_{t+1} - r_t$ reflects the marginal evidence gain induced by a new search step or query reformulation. Note that the external search environment is immutable and cannot be manipulated by the agent, and the model can only maximize this reward by generating high-quality queries to retrieve relevant documents, thus fundamentally preventing reward hacking via generating "hallucinated" answer tokens. In Appendix C.2, we report a positive association between r_t and golden document coverage.

5.2 Process Rewards

Using r_t as the evidence state, we define process rewards under two regimes. In the **high-evidence regime**, relevance is high, so additional search has low marginal utility and may become redundant. The reward should encourage efficient evidence use. In the **low-evidence regime**, relevance remains low, so further search or reformulation can still provide positive marginal utility. The reward should encourage coverage expansion and directional correction. Because the latent evidence regime is not directly observable, we use outcomes as a scalable proxy. While this is a coarse approximation (e.g., a model might fail reasoning despite having high-quality evidence), our analysis in Section 4 demonstrates that statistically, successful trajectories correspond to high-evidence cases, while failed trajectories correspond to low-evidence cases and misdirected queries.

Minimal-search Preference (MSP). For successful trajectories, we define

$$R_{\text{MSP}} = \frac{r_T}{T}. \tag{9}$$

This term instantiates the high-evidence regime, where additional search has low marginal utility. It encourages achieving high relevance with fewer steps, reducing redundancy in successful trajectories.

Early Alignment Reward (EAR). To measure whether the model aligns with the correct information direction at an early stage, we define

$$R_{\text{EAR}} = \cos(v_1, z). \tag{10}$$

A higher value indicates that the first retrieved result is already close to the reference representation. This reward encourages early alignment with the true information need, rather than relying on later trial-and-error correction.

Coverage-Encouraging Reward (CER). For failed trajectories, we encourage continued exploration when evidence coverage is still low:

$$R_{\text{CER}} = (1 - r_T) \cdot \frac{\ln(T+1)}{\ln(T_{\max}+1)}. \tag{11}$$

Here, r_T is the highest relevance reached by the end of the trajectory, and $T_{\max}$ is the maximum search budget. This term instantiates the low-evidence regime, where the marginal utility of additional evidence remains positive. It encourages exploration when evidence coverage is insufficient.

Beneficial Reformulation Reward (BRR). To reward query reformulations that genuinely improve the search direction, we define

$$R_{\mathrm{BRR}} = \tanh\left(\sum_{t=1}^{T-1}\left(1-\cos\left(u_{t+1},u_t\right)\right)\cdot\left(r_{t+1}-r_t\right)\right). \tag{12}$$

Here, $1-\cos(u_{t+1},u_t)$ measures query change, and $\max(0,r_{t+1}-r_t)$ the resulting relevance gain. Thus, this reward approximates positive marginal gain from reformulation, rather than rewarding arbitrary query changes. The $\tanh(\cdot)$ function bounds the reward scale for stable training.

5.3 Reward Integration

We combine the process rewards into group-specific signals: $R_{\mathrm{succ}} = R_{\mathrm{MSP}} + R_{\mathrm{EAR}}$ for successful trajectories and $R_{\mathrm{fail}} = R_{\mathrm{CER}} + R_{\mathrm{BRR}}$ for failed trajectories. We then integrate selected process reward R_{proc} with outcome reward R_{out} under GRPO [Shao et al., 2024] and Dr.GRPO [Liu et al., 2025d]. We consider two integration strategies for adding shaping signals to policy optimization.

Reward-level Merging. We combine the selected process reward with the outcome reward:

$$R = \begin{cases} R_{\mathrm{out}} + \lambda_{\mathrm{succ}} R_{\mathrm{succ}}, & \hat{y} \approx y, \\ R_{\mathrm{out}} + \lambda_{\mathrm{fail}} R_{\mathrm{fail}}, & \hat{y} \not\approx y \end{cases} \tag{13}$$

where the process branch is selected according to whether the predicted answer matches the ground truth. The weights λ_{succ} and λ_{fail} control the contributions of the success-and failure-side rewards.

Advantage-level Merging. We integrate process signals at the advantage level. Let $A^{(i)}_{\mathrm{out}}$ denote the outcome advantage of trajectory i. For each group $g \in \{\mathrm{succ}, \mathrm{fail}\}$, we compute the process advantage by mean-centering process rewards within the group: $A^{(i)}_{\mathrm{proc}} = R^{(i)}_{\mathrm{proc}} - \frac{1}{|g|}\sum_{j\in g} R^{(j)}_{\mathrm{proc}}$. The final advantage is defined as

$$A^{(i)} = \begin{cases} A^{(i)}_{\mathrm{out}} + \lambda_{\mathrm{succ}} A^{(i)}_{\mathrm{proc}}, & \hat{y}^{(i)} \approx y, \\ A^{(i)}_{\mathrm{out}} + \lambda_{\mathrm{fail}} A^{(i)}_{\mathrm{proc}}, & \hat{y}^{(i)} \not\approx y. \end{cases} \tag{14}$$

We do not apply standard-deviation normalization to A_{proc} to avoid unstable rescaling with small groups or low variance. Compared with reward-level merging, this strategy treats process rewards as within-group credit signals that rank trajectories of the same outcome by process quality.

6 Experiment

6.1 Experimental Settings

We evaluate on two task types: multi-hop QA and Web search. Multi-hop QA includes HotpotQA [Yang et al., 2018], 2WikiMultiHopQA [Ho et al., 2020], Musique [Trivedi et al., 2022] and Bamboogle [Press et al., 2023] with a local RAG retriever, while Web search includes GAIA [Mialon et al., 2024] and WebWalker [Wu et al., 2025a] with online Bing Search. Training is conducted only on the multi-hop QA training sets of HotpotQA, 2WikiMultiHopQA and Musique. We report accuracy on three Qwen2.5 backbones [Yang et al., 2024]: Qwen2.5-3B-Base, Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. Baselines are grouped into tool-free reasoning (Direct Inference, CoT [Kojima et al., 2022]), training-free search agents (ReAct [Yao et al., 2022], Search-o1 [Li et al., 2025a]), outcome-supervised search agents (Search-R1 [Jin et al., 2025], Search-R1†[1]) and process-aware search agents (StepSearch [Zheng et al., 2025]). Full details are in Appendix A.

6.2 Main Results

The experimental results are shown in Table 1. **PAGE** achieves the best average performance on multi-hop QA across all three backbones, with consistent gains over Search-R1 ($+0.087$, $+0.040$ and

[1] Search-R1 without standard-deviation normalization, following Dr.GRPO.

Table 1: Results on multi-hop QA and web search. Best in bold, second-best underlined.

Method	Multi-hop QA					Web Search		
	2WikiMQA	Bamboogle	HotpotQA	Musique	Average	GAIA	WebWalker	Average
Qwen2.5-3B-Base								
+ Direct Inference	0.165	0.168	0.165	0.045	0.136	0.019	0.010	0.015
+ CoT	0.155	0.200	0.160	0.035	0.138	0.019	0.020	0.020
+ ReAct	0.138	0.192	0.180	0.050	0.140	0.049	0.035	0.042
+ Search-o1	0.190	0.248	0.150	0.025	0.153	0.039	0.020	0.030
+ Search-R1	0.410	0.368	0.445	0.140	0.341	0.049	0.045	0.047
+ Search-R1†	0.185	0.264	0.305	0.100	0.214	0.000	0.025	0.013
+ StepSearch	0.430	0.384	0.430	0.190	0.359	0.029	0.045	0.037
+ PAGE	**0.490**	**0.456**	**0.515**	**0.250**	**0.428**	**0.058**	**0.050**	**0.054**
Qwen2.5-3B-Instruct								
+ Direct Inference	0.130	0.256	0.170	0.040	0.149	0.019	0.020	0.020
+ CoT	0.180	0.320	0.195	0.035	0.183	0.039	0.015	0.027
+ ReAct	0.320	0.336	0.360	0.100	0.279	0.039	0.075	0.057
+ Search-o1	0.345	0.320	0.355	0.105	0.281	0.058	0.020	0.039
+ Search-R1	0.435	0.384	0.525	**0.225**	0.392	**0.078**	0.125	0.102
+ Search-R1†	0.410	0.384	0.470	0.165	0.357	0.058	0.105	0.082
+ StepSearch	0.395	0.416	0.475	0.165	0.363	0.068	0.125	0.097
+ PAGE	**0.495**	**0.504**	**0.545**	0.185	**0.432**	**0.078**	**0.145**	**0.112**
Qwen2.5-7B-Instruct								
+ Direct Inference	0.230	0.336	0.235	0.095	0.224	0.029	0.010	0.020
+ CoT	0.215	0.360	0.230	0.065	0.218	0.049	0.020	0.035
+ ReAct	0.365	0.472	0.445	0.215	0.374	0.068	0.085	0.077
+ Search-o1	0.430	0.480	0.520	0.215	0.411	0.049	0.020	0.035
+ Search-R1	0.470	**0.552**	0.535	0.220	0.444	0.097	0.090	0.094
+ Search-R1†	0.470	0.472	**0.580**	0.215	0.434	0.097	**0.110**	0.104
+ StepSearch	0.445	0.472	0.510	**0.260**	0.422	0.078	0.075	0.077
+ PAGE	**0.545**	0.536	0.575	0.250	**0.477**	**0.107**	0.105	**0.106**

Table 2: Ablation: Qwen2.5-3B-Base.

Variant	2Wiki.	Bamb.	Hotpot.	Musiq.	Avg.
Outcome only (no process)	0.185	0.264	0.305	0.100	0.214
+ success branch only (MSP+EAR)	0.375	0.248	0.475	0.130	0.307
+ failure branch only (CER+BRR)	0.375	0.256	0.465	0.120	0.304
+ both branches (full method)	**0.490**	**0.456**	**0.515**	**0.250**	**0.428**

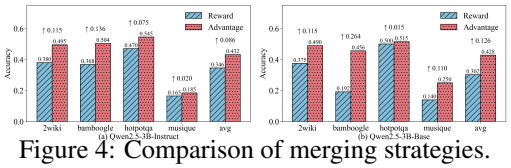


Figure 4: Comparison of merging strategies.

+0.033). This supports our hypothesis that outcome-level supervision alone is insufficient, and that state-dependent process supervision provides useful additional signals. **PAGE** also improves over process-aware baselines. This suggests that the benefit does not come merely from adding process rewards, but from assigning differentiated process signals to different trajectory groups. On Web Search tasks, **PAGE** shows smaller absolute but consistent gains (e.g., a relative improvement of 14.9% from 0.047 to 0.054 on the 3B Base model). This suggests that despite the extreme difficulty and high noise of online open-search settings (as evidenced by the overall low accuracy), PAGE's process supervision remains effective and transferable. The improvements are observed across all backbones. This suggests that **PAGE** is compatible with different model scales and training settings.

6.3 Ablation: Are Both Reward Branches Needed?

Table 2 evaluates whether both state-dependent reward branches are necessary. Each branch improves over the outcome-only baseline, but neither matches the full method. This suggests that successful and failed trajectories require different process signals: the success branch improves efficiency and early alignment, while the failure branch improves evidence coverage and query correction. Their combination provides complementary supervision and yields the strongest performance. We provide a per-reward ablation in Appendix D.1, showing unequal and complementary contributions.

6.4 Reward-Level vs. Advantage-Level Merging

Figure 4 compares the two merging strategies. Advantage-level merging outperforms reward-level merging across backbones and most datasets. This suggests that process signals work better as within-group corrections than as direct reward additions. Reward-level merging changes reward scale before advantage normalization, which may cause scale mismatch or weaken the outcome objective. In contrast, advantage-level merging preserves the outcome advantage and adds mean-centered process advantages within each group. This lets process signals serve as fine-grained credit assignment within the same outcome class, while preserving the preference of correct over incorrect. Generalization across training paradigms and tool analyses are provided in Appendix D.2 and Appendix D.3.

7 Conclusion

We study process supervision for search agents and highlight the state-dependent nature of process quality. Our analysis shows that successful and failed trajectories exhibit distinct patterns and require different optimization signals. We propose PAGE, a process-aware grouped reward optimization framework that instantiates state-dependent supervision through differentiated rewards. Experiments across multiple benchmarks show that **PAGE** improves answer accuracy over strong baselines.

Acknowledgments and Disclosure of Funding

This work was supported by a grant from New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2026ZD0127902), the National Natural Science Foundation of China (No.U2341229 and No.62406125), the Reform Commission Foundation of Jilin Province (No.2024C003), the New Cornerstone Science Foundation through the XPLORER PRIZE, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM903), the National Natural Science Foundation of China under grants (No.62372211), the China National Postdoctoral Program for Innovative Talents (No. BX2026297), and the China Postdoctoral Science Foundation (No. 2026M791630).

References

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, et al. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*, 2025a.

Xinran Chen, Yuchen Li, Hengyi Cai, Zhuoran Ma, Xuanang Chen, Haoyi Xiong, Shuaiqiang Wang, Ben He, Le Sun, and Dawei Yin. Multi-agent proactive information seeking with adaptive llm orchestration for non-factoid question answering. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4341–4352, 2025b.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, et al. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025c.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, et al. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward. *arXiv preprint arXiv:2508.12800*, 2025.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*, 2025.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous

RL. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025. URL https://openreview.net/forum?id=JSYCLJcniH.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Google DeepMind. Gemini 3 flash preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview, 2025a. Accessed: 2026-03.

Google DeepMind. Gemini 3 pro preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-pro-preview, 2025b. Accessed: 2026-03.

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

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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Zhicheng Lee, Shulin Cao, Jinxin Liu, Jiajie Zhang, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. Rearag: Knowledge-guided reasoning enhances factuality of large reasoning models with iterative retrieval augmented generation, 2025. URL https://arxiv.org/abs/2503.21729.

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.

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*, pages 5420–5438, 2025a.

Yuchen Li, Hengyi Cai, Rui Kong, Xinran Chen, Jiamin Chen, Jun Yang, Haojie Zhang, Jiayi Li, Jiayi Wu, Yiqun Chen, et al. Towards ai search paradigm. *arXiv preprint arXiv:2506.17188*, 2025b.

Yuchen Li, Jiamin Chen, Xinran Chen, Zhiyu Li, Haojie Zhang, Rui Kong, Jiayi Li, Xinyu Ma, Hengyi Cai, Lixin Su, et al. Retain to refine: Adaptive online question answering via query routing and long-short memory. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2312–2322, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025b.

Weiwen Liu, Xu Huang, Xingshan Zeng, xinlong hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong WANG, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Wang Xinzhi, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=8EB8k6DdCU.

Wenhan Liu, Xinyu Ma, Yutao Zhu, Yuchen Li, Daiting Shi, Dawei Yin, and Zhicheng Dou. Agentic-r: Learning to retrieve for agentic search. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15987–16005. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-acl.785. URL https://aclanthology.org/2026.findings-acl.785/.

Xiaojian Liu, Han Xu, Jianqiang Xia, Zhixuan Li, Ke Xu, Yiwei Dai, Xinran Chen, Changwo Wu, and Yuchen Li. Scope-rl: Optimizing reasoning paths before and after success. *arXiv preprint arXiv:2607.11506*, 2026b. doi: 10.48550/arXiv.2607.11506. URL https://arxiv.org/abs/2607.11506.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025d. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Lang Mei, Zhihan Yang, Xiaohan Yu, Huanyao Zhang, and Chong Chen. Ai-searchplanner: Modular agentic search via pareto-optimal multi-objective reinforcement learning, 2025. URL https://arxiv.org/abs/2508.20368.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

OpenAI. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, April 2025. OpenAI blog post, accessed 2026-03-18.

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

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=eOLdGbXT6t.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents, 2025. URL https://arxiv.org/abs/2509.13309.

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

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

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. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Otc: Optimal tool calls via reinforcement learning. *arXiv e-prints*, pages arXiv–2504, 2025.

Xiaolong Wei, Zerun Zhu, Simin Niu, Xingyu Zhang, Peiying Yu, Changxuan Xiao, Yuchen Li, Jicheng Yang, Zhejun Zhao, Chong Meng, Long Xia, and Daiting Shi. Unicreative: Unifying long-form logic and short-form sparkle via reference-free reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 23563–23583, 2026.

Tongyu Wen, Guanting Dong, and Zhicheng Dou. Smartsearch: Process reward-guided query refinement for search agents. *arXiv preprint arXiv:2601.04888*, 2026.

Wikipedia contributors. Okapi bm25 — Wikipedia, the free encyclopedia, 2025. URL https://en.wikipedia.org/w/index.php?title=Okapi_BM25&oldid=1330270429. [Online; accessed 18-March-2026].

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. WebWalker: Benchmarking LLMs in web traversal. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10290–10305, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.508. URL https://aclanthology.org/2025.acl-long.508/.

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. *arXiv preprint arXiv:2510.07794*, 2025b.

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

Ke Xu, Han Xu, Xinran Chen, Yuqian Wang, Zhixuan Li, Xiaojian Liu, Changwo Wu, Jianqiang Xia, and Yuchen Li. Stamp: Provenance-guided credit assignment for deep search agents. *arXiv preprint arXiv:2607.11172*, 2026. doi: 10.48550/arXiv.2607.11172. URL https://arxiv.org/abs/2607.11172.

Zhichao Xu, Zongyu Wu, Yun Zhou, Aosong Feng, Kang Zhou, Sangmin Woo, Kiran Ramnath, Yijun Tian, Xuan Qi, Weikang Qiu, et al. Beyond correctness: Rewarding faithful reasoning in retrieval-augmented generation. *arXiv preprint arXiv:2510.13272*, 2025.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

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*, 2022.

Zhiyuan Yao, Zishan Xu, Yifu Guo, Zhiguang Han, Cheng Yang, Shuo Zhang, Weinan Zhang, Xingshan Zeng, and Weiwen Liu. Ace-router: Generalizing history-aware routing from mcp tools to the agent web, 2026. URL https://arxiv.org/abs/2601.08276.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context management, 2025. URL https://arxiv.org/abs/2510.24699.

Xinmiao Yu, Liwen Zhang, Xiaocheng Feng, Yong Jiang, Bing Qin, Pengjun Xie, and Jingren Zhou. Webanchor: Anchoring agent planning to stabilize long-horizon web reasoning. *arXiv preprint arXiv:2601.03164*, 2026.

Yuanqing Yu, Zhefan Wang, Weizhi Ma, Shuai Wang, Chuhan Wu, Zhiqiang Guo, and Min Zhang. Steptool: Enhancing multi-step tool usage in llms via step-grained reinforcement learning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 3952–3962, 2025.

Zhenrui Yue, Kartikeya Upasani, Xianjun Yang, Suyu Ge, Shaoliang Nie, Yuning Mao, Zhe Liu, and Dong Wang. Dr. zero: Self-evolving search agents without training data. *arXiv preprint arXiv:2601.07055*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zhejun Zhao, Yuchen Li, Alley Liu, Yuehu Dong, Xiaolong Wei, Lixue Zheng, Pingsheng Liu, Dongdong Shen, Long Xia, Jiashu Zhao, and Dawei Yin. Tura: Tool-augmented unified retrieval agent for ai search. *arXiv preprint arXiv:2508.04604*, 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*, pages 21816–21841, 2025.

A Training Details

A.1 Baselines.

We compare **PAGE** with four categories of baselines, covering tool-free inference, training-free search agents, outcome-supervised search-agent training and process-aware search-agent training.

Tool-free reasoning. This category evaluates whether external search is necessary for the target tasks. **Direct Inference** directly prompts the backbone model to generate the final answer without intermediate tool use. **CoT** [Kojima et al., 2022] adds a "Let's think step by step" style instruction to elicit intermediate reasoning, but still does not access external evidence.

Training-free search agents. This category evaluates search-augmented inference without updating the model parameters. **ReAct** [Yao et al., 2022] follows an interleaved reasoning-and-search format, where the model alternates between reasoning, issuing search queries, observing retrieved information and producing the final answer. **Search-o1** [Li et al., 2025a] further improves search-augmented reasoning by refining noisy search results with the interaction history.

Outcome-supervised search agents. This category evaluates reinforcement learning methods that optimize search agents using only final-answer correctness. **Search-R1** [Jin et al., 2025] trains the search agent with GRPO under an outcome-level reward. We also include **Search-R1**†, trained with the same outcome-reward setting using GRPO without standard-deviation normalization, following Dr.GRPO [Liu et al., 2025d], to provide a stronger outcome-only comparison under our optimization backbone.

Process-aware search agents. This category evaluates methods that introduce intermediate supervision for search behavior. **StepSearch** [Zheng et al., 2025] uses step-wise process rewards to improve search-agent training. Compared with these baselines, **PAGE** assigns differentiated process rewards to successful and failed trajectories and evaluates both reward-level and advantage-level integration strategies. All experiments are conducted on a single node with 8 NVIDIA A800 GPUs.

A.2 Implementation Details.

We train each policy using the rollout-then-update loop of GRPO [Shao et al., 2024], and use Dr.GRPO for the corresponding variant [Liu et al., 2025d]. Other hyperparameters follow Search-R1 [Jin et al., 2025]. The training batch size is 256, the learning rate is 1×10^{-6}, and each model is trained for 300 steps. We save checkpoints every 50 steps. If reward divergence occurs, we either resume training from the latest stable checkpoint or select the most recent stable checkpoint for evaluation. During training, each trajectory is allowed up to 4 search rounds, and the retriever returns the top-3 results for each query.

For process reward computation, we use Qwen2.5-7B-Instruct [Yang et al., 2024] as the embedding model and apply mean pooling to obtain text representations. The success-side reward weight λ_{succ} is selected from $\{0.1, 0.2, 0.3, 0.4\}$ on a held-out validation set of 1,000 randomly sampled examples. The failure-side reward weight λ_{fail} is set to either λ_{succ} or $\lambda_{\mathrm{succ}}/2$. We set the maximum prompt length to 8192 and the maximum response length to 500 during training.

For evaluation, we use the same retriever and return the top-3 results for each search query. For benchmarks with more than 200 test examples, we evaluate on a randomly sampled subset of 200 examples to maintain a consistent evaluation budget. We use DeepSeek-V3.1 Liu et al. [2024] as the LLM judge for automatic answer evaluation. At inference time, each trajectory is up to 8 search rounds. Generation uses a temperature of 0.7 and a maximum generation length of 2048 tokens.

B Behavior Pattern Definitions

We provide full definitions of the behavior labels used in our process-level analysis. These labels describe dominant observable behavior patterns rather than mutually exclusive error categories. A single trajectory may exhibit multiple patterns at different stages, but our annotation focuses on the most salient pattern within each behavioral dimension. All definitions are based only on observable interaction signals, including search actions, query sequences, retrieved evidence, and final outputs.

B.1 Search Strategy.

Efficient Search describes trajectories that achieve sufficient evidence coverage in relatively few search steps and terminate accordingly. It reflects a compact and goal-directed search habit. *For example, in a multi-hop question asking for an actor, the agent first searches the TV series creator, then searches the candidate actor's education and first role, and stops once these retrieved facts jointly identify the actor.*

Overuse Search describes trajectories that continue searching after sufficient evidence has already been obtained. It reflects a tendency to extend search beyond what is necessary. *For example, the agent retrieves a source that directly states the required interview answer, but then continues searching for the event location, attendee list, and background context, none of which changes the final answer.*

Shallow Search describes trajectories that terminate before sufficient evidence coverage is achieved. It reflects a tendency toward under-exploration. *For example, in a question requiring both the identity of a match and its final score, the agent finds only the match page, fails to retrieve the score details, and then proceeds to answer without verifying the missing condition.*

Redundant Exploration describes trajectories with high overlap across search steps and limited marginal evidence gain. It reflects repetitive search behavior with low incremental utility. *For example, the agent repeatedly searches variants such as "missing teens found," "two missing teens found," and "missing teens case," and each search returns similar news reports while leaving the unresolved clue unchanged.*

Delayed Discovery describes trajectories in which relevant evidence is retrieved only after multiple search steps. It reflects a tendency to discover key information late in the search process. *For example, the agent first searches broad music-related phrases such as "band song release" and "music group history," but only after several failed steps introduces the correct band member or song title, which finally retrieves the key evidence.*

B.2 Query Reformulation.

Direct Query describes queries that closely follow the original question with minimal reformulation. It reflects a direct retrieval style. *For example, if the question asks which person satisfies a given description, the agent issues a query that nearly copies the full description instead of decomposing it into searchable entities or relations.*

Entity Adjustment describes queries that introduce or modify key entities in order to refine the retrieval target. It reflects a pattern of entity-level query adaptation. *For example, the agent moves from a vague description such as "company founder" to a more specific query using the company name, and then to a query using the candidate founder's name, while keeping the same information need.*

Constraint Refinement describes queries that add constraints to narrow the search scope. It reflects a tendency to progressively specify the retrieval intent. *For example, the agent refines a query about an actor by adding a graduation year, institution, location, or role name to filter out irrelevant candidates.*

Underspecified Query describes queries that omit important constraints or remain overly broad. It reflects a broad or weakly specified query formulation style. *For example, the agent searches only a generic phrase such as "missing teens case" or "archaeological report" without including the relevant year, location, entity name, or distinguishing relation.*

Reformulation Drift describes query sequences whose entities or semantic focus shift away from the original information need over time. It reflects a tendency for the retrieval target to drift during iterative search. *For example, the agent starts with queries about the intended publisher or work, but later switches to unrelated people, countries, or historical topics, so the search trajectory no longer targets the original question.*

Table 3: Full behavior-pattern statistics on BrowseComp-Plus.

Category	Label	Correct Mean	Correct Std	Incorrect Mean	Incorrect Std	Diff	Combined p-value (Fisher)	Same-direction Models	Combined p-value (FDR-BH)
query	underspecified_query	0.302	0.074	0.720	0.082	0.418	1.056×10^{-123}	4/4	5.280×10^{-123}
query	reformulation_drift	0.580	0.056	0.867	0.051	0.288	8.349×10^{-71}	4/4	2.783×10^{-70}
query	constraint_refinement	0.341	0.063	0.101	0.045	-0.240	2.556×10^{-57}	4/4	5.112×10^{-57}
query	entity_adjustment	0.500	0.053	0.212	0.059	-0.287	3.700×10^{-63}	4/4	9.251×10^{-63}
search_action	shallow_search	0.026	0.009	0.509	0.064	0.483	4.164×10^{-248}	4/4	4.164×10^{-247}
search_action	redundant_exploration	0.516	0.011	0.707	0.007	0.192	2.545×10^{-25}	4/4	3.635×10^{-25}
search_action	overuse_search	0.566	0.052	0.423	0.046	-0.143	4.114×10^{-18}	4/4	5.142×10^{-18}
search_action	efficient_search	0.151	0.043	0.003	0.004	-0.148	5.533×10^{-57}	4/4	9.222×10^{-57}

B.3 Evidence Synthesis.

Complete Synthesis describes outputs that cover major retrieved evidence and integrate it coherently into the final answer. It reflects a strong evidence utilization pattern. *For example, the final answer links the retrieved identity, relation, and constraint evidence into one explanation that shows why the selected answer satisfies the question.*

Incomplete Synthesis describes outputs that use only part of the available evidence. It reflects partial evidence utilization. *For example, the agent retrieves evidence for both the target entity and a required numerical or temporal condition, but the final answer only uses the entity evidence and omits the condition.*

Misaligned Synthesis describes outputs whose final claim is only partially supported by the retrieved evidence. It reflects a mismatch between retrieved information and final conclusion. *For example, the retrieved pages support one candidate or source, but the final answer attributes the evidence to a different candidate, paper, event, or author.*

Unstructured Aggregation describes outputs that list retrieved facts without clear integration or organization. It reflects a tendency to accumulate evidence without forming a coherent synthesis. *For example, the final output lists several retrieved names, dates, places, or candidate clues, but does not explain which one is selected or how the facts jointly support the answer.*

C Analysis of Search-Agent Trajectories

C.1 Process analysis on BrowseComp-Plus

Table 3 reports the full behavior-pattern statistics on BrowseComp-Plus. For each pattern, we report its mean prevalence in correct and incorrect trajectories, together with the standard deviation across four models. *Diff* is computed as Incorrect Mean − Correct Mean. Positive values indicate higher prevalence in incorrect trajectories, while negative values indicate higher prevalence in correct trajectories. The *Combined p-value (Fisher)* is obtained by first performing a one-sided Fisher's exact test within each model. The test direction follows the sign of *Diff*. We then combine the four model-specific p-values using Fisher's method. Smaller values indicate stronger evidence that the observed difference is unlikely to be due to chance. The *Combined p-value (FDR-BH)* is the Benjamini–Hochberg corrected version of the combined p-values over all tested patterns. Smaller corrected values indicate stronger evidence after accounting for multiple comparisons.

C.2 Step-wise Relevance and Coverage Analysis

To justify using embedding similarity as a proxy for evidence quality, we analyze its relationship with golden document coverage on BrowseComp-Plus. As shown in Figure 5, prefix relevance generally increases together with golden document coverage, suggesting that embedding similarity captures the accumulation of useful evidence along the search trajectory. Since golden supporting documents are not available during training, we use embedding-based prefix relevance as a scalable proxy signal. We also observe cases where golden document coverage has saturated, but the agent continues to issue additional search queries. This supports our observation that search agents may overuse search even after sufficient evidence has been retrieved.

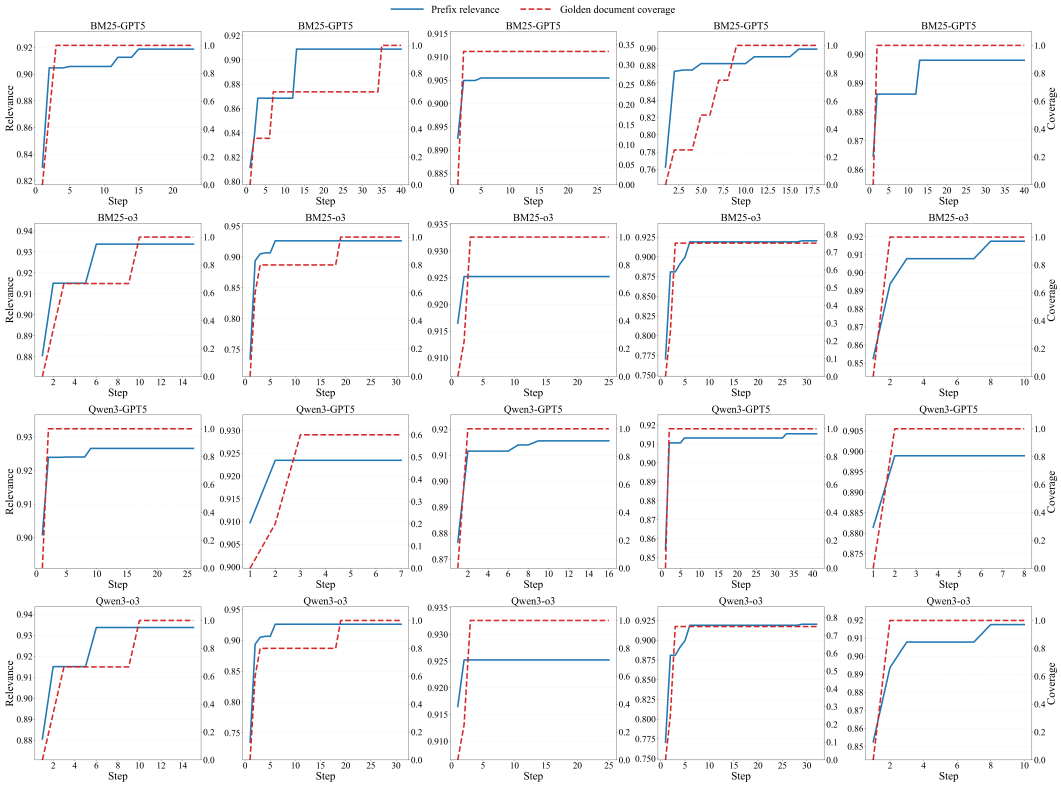


Figure 5: Step-wise evolution of prefix relevance (blue) and golden document coverage (red) on BrowseComp-Plus.

Table 4: Per Reward Ablation on Qwen2.5-3B-Base with Dr.GRPO.

Variant	2WikiMQA	Bamboogle	HotpotQA	Musique	Average
PAGE w/o MSP	0.435	0.352	0.410	0.130	0.332
PAGE w/o EAR	0.375	0.304	0.455	0.080	0.304
PAGE w/o CER	0.435	0.432	**0.545**	0.185	0.399
PAGE w/o BRR	0.355	0.312	0.470	0.130	0.317
PAGE	**0.490**	**0.456**	0.515	**0.250**	**0.428**

C.3 Reliability of LLM-Based Process Annotation

To assess the reliability of automatic process annotations, we randomly sample 50 trajectories for manual evaluation. Specifically, we use the BM25 retriever on BrowseComp-Plus to obtain search results and GPT-5 to generate final answers. We then compare the automatic annotations with those provided by two human annotators. The automatic annotations achieve an average agreement rate of 75.33% and an average Cohen's κ of 0.532. Since process behaviors must be inferred from multi-step search trajectories rather than directly observable ground-truth labels, some disagreement between automatic and human annotations is expected.

D Evaluation of the PAGE Framework

D.1 Per-Reward Ablation

Table 4 shows the contribution of each process reward. The full PAGE model achieves the best average performance, suggesting that the four rewards provide complementary supervision. Among them, removing EAR leads to the largest performance drop, indicating that early alignment is a key factor in search quality. When the initial retrieval direction is misaligned, later query reformulation and evidence accumulation can also be affected. Removing BRR also causes a clear degradation,

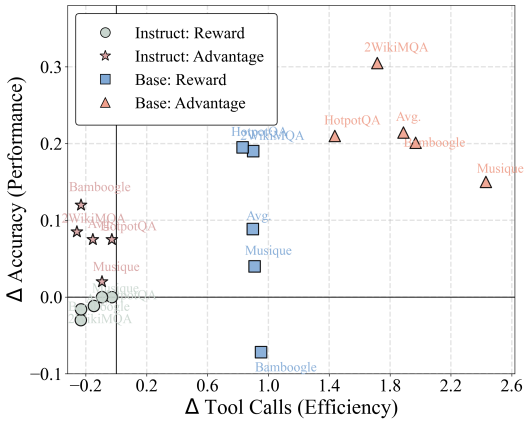


Figure 6: Per-question Δ in accuracy vs. Δ in tool calls (ours − Search-R1†).

showing that beneficial query reformulation is important for correcting the search direction, especially when the initial evidence is insufficient or misaligned. The effects of MSP and CER are relatively smaller but still meaningful. Removing MSP weakens the ability to discourage redundant search in successful trajectories. Removing CER reduces the exploration signal for failed trajectories, making it harder for the model to expand evidence coverage when evidence is insufficient.

D.2 Generalization Across Training Paradigms

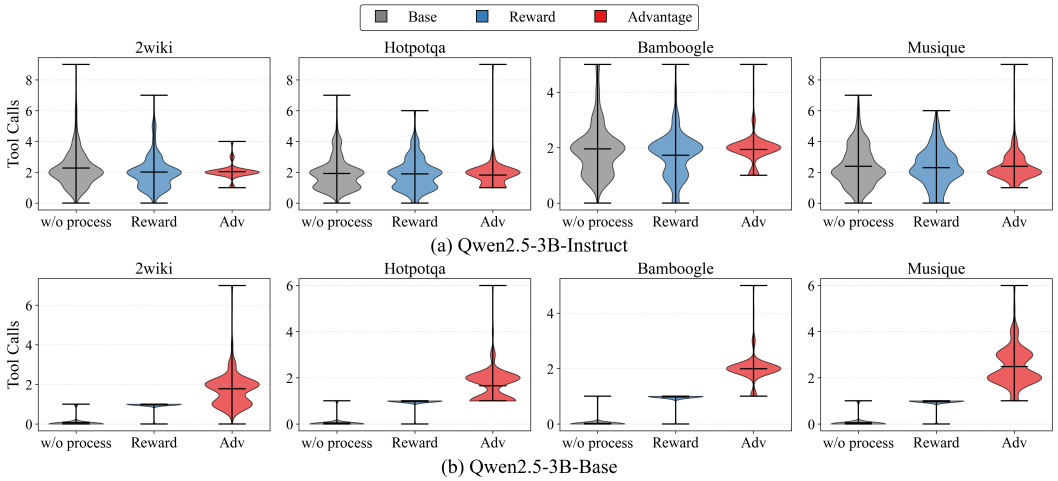
Table 5: Generalization across optimization methods. Accuracy on multi-hop QA and web-search benchmarks under GRPO and Dr.GRPO. **PAGE** consistently improves over the corresponding Search-R1 baselines across different backbones and training algorithms.

Method	Multi-hop QA					Web Search		
	2WikiMQA	Bamboogle	HotpotQA	Musique	Average	GAIA	WebWalker	Average
Qwen2.5-3B-Base								
+ Search-R1(GRPO)	0.410	0.368	0.445	0.140	0.341	0.049	0.045	0.047
+ Search-R1(DR.GRPO)	0.185	0.264	0.305	0.100	0.214	0.000	0.025	0.013
+ PAGE(GRPO)	0.440	**0.456**	0.505	0.240	0.410	**0.068**	**0.055**	**0.062**
+ PAGE(DR.GRPO)	**0.490**	**0.456**	**0.515**	**0.250**	**0.428**	0.058	0.050	0.054
Qwen2.5-3B-Instruct								
+ Search-R1(GRPO)	0.435	0.384	0.525	0.225	0.392	0.078	0.125	0.102
+ Search-R1(DR.GRPO)	0.410	0.384	0.470	0.165	0.357	0.058	0.105	0.082
+ PAGE(GRPO)	0.445	0.496	**0.545**	**0.215**	0.425	**0.087**	**0.145**	**0.116**
+ PAGE(DR.GRPO)	**0.495**	**0.504**	**0.545**	0.185	**0.432**	0.078	**0.145**	0.112
Qwen2.5-7B-Instruct								
+ Search-R1(GRPO)	0.470	**0.552**	0.535	0.220	0.444	0.097	0.090	0.094
+ Search-R1(DR.GRPO)	0.470	0.472	**0.580**	0.215	0.434	0.097	0.110	0.104
+ PAGE(GRPO)	0.480	0.512	0.560	0.240	0.448	0.087	**0.165**	**0.126**
+ PAGE(DR.GRPO)	**0.545**	0.536	0.575	**0.250**	**0.477**	**0.107**	0.105	0.106

The results in Table 5 show that **PAGE** generalizes across optimization methods. Under both GRPO and Dr.GRPO, **PAGE** improves over the corresponding Search-R1 baseline on most backbones. On multi-hop QA, the gains are especially consistent, suggesting that process rewards help the model learn more effective search behavior. The improvements also transfer to web-search benchmarks, indicating that the learned search policy can generalize beyond the local RAG setting. **PAGE** shows gains under both optimizers, which suggests that its benefit comes from the proposed process-reward design rather than from a specific optimization algorithm.

D.3 Tool-Use Efficiency

The tool-use analysis is shown in Figure 6 and Figure 7. Figure 6 compares the change in accuracy with the change in tool-call counts relative to Search-R1†. **PAGE** improves accuracy in most settings, but the corresponding change in tool calls is not uniform. Some settings achieve higher accuracy with



(a) Qwen2.5-3B-Instruct

(b) Qwen2.5-3B-Base

Figure 7: Distribution of tool-call counts per trajectory across benchmarks for the outcome-only baseline and **PAGE**.

fewer or comparable tool calls, while others improve accuracy with additional search. This suggests that **PAGE** does not rely on simply increasing search frequency, but changes how the model allocates search effort across questions.

Figure 7 further shows the distribution of tool-call counts. Reward-level merging tends to keep the number of tool calls close to the outcome-only baseline, especially on the Instruct model. Advantage-level merging produces a clearer increase in tool calls, particularly on the Base model. This indicates that advantage-level merging encourages stronger exploration, whereas reward-level merging provides a more conservative efficiency-performance trade-off.

D.4 Reliability of Final-Answer-Based Trajectory Grouping

To assess how often final-answer-based grouping conflicts with observed evidence quality, we examine two types of outcome–evidence mismatch: failed trajectories with high prefix relevance and successful trajectories with low prefix relevance. For each of Qwen2.5-3B-Base, Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct, we randomly sample 50 successful and 50 failed trajectories. We measure the proportion of failed trajectories whose final prefix-relevance score exceeds 0.85 and the proportion of successful trajectories whose score falls below 0.70. We additionally report results under stricter thresholds to characterize more extreme mismatch cases.

Table 6 reports the proportion of failed trajectories that nevertheless achieve high prefix relevance. Across the three models, 12%–20% of failed trajectories satisfy $r_T > 0.85$. This proportion decreases to 0%–4% under the stricter threshold of $r_T > 0.90$, and no sampled trajectory satisfies $r_T > 0.95$. Table 7 presents the complementary mismatch case. Only 4%–16% of successful trajectories satisfy $r_T < 0.70$. The proportions further decrease under stricter thresholds, with at most 6% satisfying $r_T < 0.65$ and at most 2% satisfying $r_T < 0.60$. The results confirm that final-answer correctness is not a perfect proxy for the latent evidence state. Some failed trajectories retrieve highly relevant evidence, while some successful trajectories exhibit relatively low prefix relevance. However, extreme mismatches become rare under stricter thresholds: only 1.3% of failed trajectories satisfy $r_T > 0.90$, and only 0.7% of successful trajectories satisfy $r_T < 0.60$. These results support the use of final-answer correctness as a coarse-grained and scalable signal for partitioning trajectories into the Success and Failure Groups in Section 5.2.

D.5 Recovery from Initial Query Drift

To determine whether PAGE improves recovery from incorrect initial searches rather than merely avoiding search errors, we conduct an inference-time intervention experiment without additional training. We compare PAGE with the *w/o Process Reward* variant using Qwen2.5-3B-Base. For each original trajectory, we use DeepSeek-V4-Flash with a fixed prompt to replace the first search query

Table 6: Proportion of failed trajectories with high final prefix-relevance scores.

Model	$r_T > 0.85$	$r_T > 0.90$	$r_T > 0.95$
Qwen2.5-3B-Base	18.0%	4.0%	0.0%
Qwen2.5-3B-Instruct	12.0%	0.0%	0.0%
Qwen2.5-7B-Instruct	20.0%	0.0%	0.0%
Average	16.7%	1.3%	0.0%

Table 7: Proportion of successful trajectories with low final prefix-relevance scores.

Model	$r_T < 0.70$	$r_T < 0.65$	$r_T < 0.60$
Qwen2.5-3B-Base	16.0%	6.0%	2.0%
Qwen2.5-3B-Instruct	6.0%	4.0%	0.0%
Qwen2.5-7B-Instruct	4.0%	0.0%	0.0%
Average	8.7%	3.3%	0.7%

with a broader and misdirected query, thereby simulating initial query drift. The evaluated model then continues the trajectory from this perturbed initial search. As shown in Table 8, the *w/o Process Reward* variant exhibits noticeable performance degradation after the intervention, decreasing from 0.385 to 0.345 on 2WikiMQA and from 0.336 to 0.304 on Bamboogle. Its average score decreases from 0.327 to 0.306. In contrast, PAGE remains stable across all four benchmarks, with its average score changing from 0.440 to 0.444. These results suggest that PAGE does not merely learn to avoid poor initial searches, but develops more robust search behavior that enables the agent to adjust subsequent searches and recover from early query drift.

D.6 Sensitivity to the Embedding Reference

To evaluate the sensitivity of PAGE to the embedding reference, we conduct an additional ablation on Qwen2.5-3B-Base by replacing the default question-and-answer reference with a question-only reference while keeping all other training settings unchanged. As shown in Table 9, the question-only variant improves the average performance over *w/o Process Reward* from 0.214 to 0.255 and achieves better results on three of the four benchmarks. However, it consistently underperforms the default PAGE configuration, which achieves an average score of 0.428. Qualitative inspection suggests that the question-only reference encourages the model to repeatedly reuse or reformulate the original question rather than generate targeted evidence-seeking queries. Although this behavior can be effective for relatively simple questions, it is less suitable for challenging multi-hop tasks that require retrieving specific supporting evidence. These results support the use of the question-and-answer reference, where the answer provides complementary semantic information for estimating evidence relevance.

D.7 Stability Across Repeated Runs

To evaluate the stability of our experimental results, we repeat the main experiments with Qwen2.5-3B-Base three times for both PAGE and Search-R1 (GRPO) under the same configuration. Table 10 reports the mean and standard deviation across the three runs on four multi-hop question-answering benchmarks. PAGE consistently outperforms Search-R1 (GRPO) across all four datasets, with standard deviations of 0.028, 0.036, 0.032, and 0.015 on 2WikiMQA, Bamboogle, HotpotQA, and MuSiQue, respectively. PAGE also exhibits lower variability than Search-R1 on 2WikiMQA and HotpotQA, comparable variability on Bamboogle, and slightly lower variability on MuSiQue. These results indicate that the performance improvements of PAGE are stable across repeated runs rather than being driven by a single run.

D.8 Process-Level Behavioral Analysis

To examine whether the performance gains of PAGE are accompanied by improved search behavior rather than only higher final-answer accuracy, we compare trajectories generated with and without the proposed process reward. Both models use the same Qwen2.5-3B-Base backbone and are evaluated on 200 HotpotQA samples. As shown in Table 11, PAGE increases the average number of searches

Table 8: Performance before and after the initial-query-drift intervention. No additional training is performed after the intervention.

Method	2WikiMQA	Bamboogle	HotpotQA	MuSiQue
w/o Process Reward	0.385	0.336	0.475	0.110
w/o Process Reward + Intervention	0.345	0.304	0.465	0.110
PAGE	0.520	0.496	0.525	0.220
PAGE + Intervention	0.515	0.496	0.540	0.225

Table 9: Performance with different embedding reference definitions. The question-only variant uses the question as the reference, whereas the default PAGE configuration uses both the question and answer.

Method	2WikiMQA	Bamboogle	HotpotQA	MuSiQue	Average
w/o Process Reward	0.185	0.264	0.305	0.100	0.214
PAGE (Question Only)	0.325	0.168	0.420	0.105	0.255
PAGE (Question + Answer)	0.490	0.456	0.515	0.250	0.428

Table 10: Mean performance and standard deviation using Qwen2.5-3B-Base.

Method	2WikiMQA	Bamboogle	HotpotQA	MuSiQue
Search-R1 (GRPO)	0.380 ± 0.033	0.333 ± 0.036	0.420 ± 0.075	0.118 ± 0.019
PAGE	0.518 ± 0.028	0.459 ± 0.036	0.538 ± 0.032	0.233 ± 0.015

from 1.07 to 1.61, indicating more active evidence gathering before producing an answer. Evidence Coverage improves from 39.0% to 50.0%, while the Any Gold Hit Rate increases from 66.5% to 72.5%, demonstrating more effective retrieval of supporting evidence. More notably, the Full Coverage Rate increases from 11.5% to 27.5%, suggesting that PAGE is substantially more likely to retrieve all required evidence. The First Hit Rate also increases from 66.5% to 72.5%, indicating a higher probability of retrieving relevant evidence during the search process. These results show that the gains of PAGE are associated with more active, effective, and complete evidence retrieval.

Table 11: Process-level search metrics on HotpotQA evaluation samples using Qwen2.5-3B-Base.

Method	Avg. Search Count	Evidence Coverage	Any Gold Hit Rate	Full Coverage Rate	First Hit Rate
w/o Process Reward	1.07	39.0%	66.5%	11.5%	66.5%
PAGE	1.61	50.0%	72.5%	27.5%	72.5%

E Limitations

Our experiments are conducted on models up to the 7B scale. Although the observed trends are consistent across different backbones, evaluation on larger models remains limited. In addition, our process rewards rely on embedding-based proxies and rule-based metrics, which may not fully capture all aspects of search process quality. Computing these rewards also introduces additional training-time overhead, although the inference procedure remains unchanged.

F Broader Impact

Our work aims to improve the reliability and efficiency of search agents for complex information-seeking tasks. By encouraging better evidence acquisition and synthesis, **PAGE** may support more accurate question answering, research assistance, and information access. We also encourage responsible use with source verification, especially in high-stakes applications.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main claims of the paper, including the state-dependent nature of process quality and the proposed **PAGE** framework.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations in Appendix E

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: The paper does not include formal theoretical results requiring assumptions or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides sufficient details to reproduce the experiments, including model architectures, training procedures, datasets and evaluation protocols. Additional implementation details are provided in the appendix A.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide the code in the supplementary material, together with instructions for reproducing the main experimental results.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Additional hyperparameters and implementation details are provided in the appendix A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report statistical significance in Analysis 4.4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide details about the compute resources used in Appendix A.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics. No ethical concerns arise from the methods or experiments presented.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss potential positive impacts in Appendix F

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release any models or datasets that pose a high risk for misuse, so safeguards for controlled release are not applicable.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use publicly available datasets and properly cite their original sources. Licensing terms are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not introduce or release any new datasets, models, or other assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or formal research with human subjects. Limited manual inspection is used only for qualitative verification.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, and therefore IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Large language models are a core component of the proposed method and are used as search agents for reasoning and tool interaction. Their usage is fully described in the methodology and appendix A.

    Guidelines:

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