# Search-P1: Path-Centric Reward Shaping for Stable and Efficient Agentic RAG Training

**Tianle Xia***, **Ming Xu***, **Lingxiang Hu**, **Yiding Sun**, **Wenwei Li**, **Linfang Shang**
**Liqun Liu**†, **Peng Shu**, **Huan Yu**, **Jie Jiang**
Tencent
{tianlexia,flemingxu,lingxianghu,emanuelsun,wenweiwwli,faelynshang}@tencent.com
{liqunliu,archershu,huanyu,zeus}@tencent.com
*Equal contribution. †Corresponding author.

## Abstract

Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by incorporating external knowledge, yet traditional single-round retrieval struggles with complex multi-step reasoning. Agentic RAG addresses this by enabling LLMs to dynamically decide when and what to retrieve, but current RL-based training methods suffer from sparse outcome rewards that discard intermediate signals and low sample efficiency where failed samples contribute nothing. We propose SEARCH-P1, a framework that introduces **path-centric reward shaping** for agentic RAG training, comprising two key components: (1) **Path-Centric Reward**, which evaluates the structural quality of reasoning trajectories through order-agnostic step coverage and soft scoring that extracts learning signals even from failed samples, and (2) **Dual-Track Path Scoring** with offline-generated reference planners that assesses paths from both self-consistency and reference-alignment perspectives. Experiments on multiple QA benchmarks demonstrate that SEARCH-P1 achieves significant improvements over Search-R1 and other strong baselines, with an average accuracy gain of 7.7 points.

## 1 Introduction

Large Language Models (LLMs) have demonstrated strong reasoning capabilities (Xia et al., 2025; Hu et al., 2025), but their static knowledge often leads to hallucinations on knowledge-intensive queries. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) addresses this by incorporating external knowledge, yet single-round retrieval is insufficient for complex multi-step reasoning—a common need in industrial applications such as advertising guidance, where answering a question often requires synthesizing information across multiple knowledge domains.

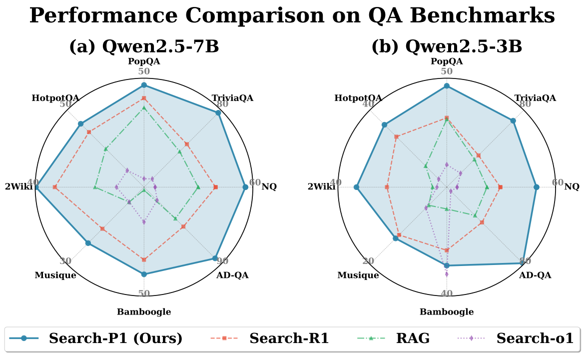


Figure 1: Performance comparison of SEARCH-P1 against baselines on QA benchmarks. Our method achieves the highest average accuracy across all datasets on both (a) Qwen2.5-7B and (b) Qwen2.5-3B models.

Agentic RAG extends traditional RAG by enabling LLMs to dynamically invoke search and iteratively refine answers. Recent methods like Search-R1 apply RL with outcome-based rewards, but this approach has three limitations: (1) **sparse rewards** that ignore intermediate reasoning quality, (2) **low sample efficiency** where partially correct trajectories receive zero reward, and (3) **slow convergence** due to weak training signals when most samples share similar binary rewards.

We propose SEARCH-P1, a framework introducing **path-centric reward shaping** for agentic RAG training that addresses all three limitations. Instead of evaluating only final answers, our reward design comprises: (1) **dual-track path scoring** that provides dense intermediate signals by evaluating reasoning trajectories from both self-consistency and reference-alignment perspectives, directly alleviating reward sparsity; and (2) **soft outcome scoring** that assigns partial credit to incorrect trajectories, converting zero-reward samples into useful training signals to improve sample efficiency. Together, the denser reward landscape accelerates convergence by providing more informative gradients throughout training. Experiments on public QA bench-



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 517–534
July 2-7, 2026 

marks and an internal advertising dataset (AD-QA) show SEARCH-P1 outperforms existing methods with an average accuracy gain of 7.7 points, while also transferring effectively to enterprise knowledge base systems. Our contributions:

- We propose dual-track path scoring that evaluates trajectories from self-consistency and reference-alignment perspectives with order-agnostic matching.
- We design a path-centric reward shaping framework that extracts learning signals even from failed trajectories via path-level reward.
- Extensive experiments on public benchmarks and an industrial dataset demonstrate consistent improvements across models and settings.

## 2 Related Work

**Prompt-Based Agentic RAG.** Initial efforts leverage prompts to guide LLMs through multi-step retrieval (Singh et al., 2025; Li et al., 2025a). These approaches interleave reasoning with retrieval actions (Yao et al., 2023; Trivedi et al., 2023) or enhance reasoning through sophisticated retrieval strategies (Li et al., 2025b; Wang et al., 2025; Guan et al., 2025). However, prompt-based methods depend heavily on the base model's instruction-following ability.

**RL-Based Agentic RAG.** Recent work applies reinforcement learning to train adaptive search agents (Zhang et al., 2025a; Jin et al., 2025). Follow-up methods incorporate auxiliary signals to stabilize training (Song et al., 2025a; Chen et al., 2025; Huang et al., 2025) or improve search efficiency (Sha et al., 2025; Song et al., 2025b; Wu et al., 2025b).

**Process Rewards for RAG.** Another line of work introduces process rewards for RAG (Sun et al., 2025; Wu et al., 2025a; Zhang et al., 2025c). PRM-based schemes, however, typically require step-level annotations and an auxiliary reward model loaded alongside the policy during RL, inflating both annotation cost and GPU memory. Concurrent work such as LeTS (Zhang et al., 2025b) further invokes an LLM-as-a-judge at *every* intermediate step, which is prohibitively expensive at industrial scale; ProRAG (Wang et al., 2026) and TreePS-RAG (Zhang et al., 2026) likewise rely on per-step process supervision. In contrast, SEARCH-P1 operates at the *trajectory* level: a single offline pass produces reference planners, and the only online cost during training is one LLM evaluator call per rollout. This makes path-centric reward shaping both annotation-free at the step level and compatible with standard GRPO/PPO pipelines, while still providing dense intermediate signals for exploration.

## 3 Methodology

We first formalize the problem setting (§3.1), then describe the path-centric reward framework including dual-track scoring and soft outcome scoring (§3.2). Figure 2 provides an overview.

### 3.1 Problem Formulation

We consider an agentic RAG system where a language model $\pi_\theta$ generates a reasoning trajectory $\mathcal{T}$ in response to a question $q$. In standard agentic RAG frameworks, the trajectory consists of interleaved reasoning and action steps:

$$\mathcal{T} = (r_1, a_1, o_1, \ldots, r_n, a_n, o_n, r_{\text{final}}, \hat{a}) \quad (1)$$

where $r_i$ denotes reasoning, $a_i$ denotes a search action, $o_i$ is the observation (search results), and $\hat{a}$ is the final answer.

We make the implicit planning in $r_1$ explicit by restructuring the trajectory as:

$$\mathcal{T} = (p, r_1, a_1, o_1, \ldots, r_n, a_n, o_n, r_{\text{final}}, \hat{a}) \quad (2)$$

where $p$ is an explicit planner that outlines the reasoning strategy. This serves two purposes: (1) providing a self-declared plan against which execution can be evaluated, and (2) making the intended reasoning structure observable for path-centric evaluation.

Standard GRPO assigns binary rewards based on answer correctness:

$$R_{\text{outcome}} = \mathbb{1}[\text{match}(\hat{a}, a^*)] \quad (3)$$

where $a^*$ is the ground-truth answer. This formulation ignores the quality of the reasoning path and suffers from the limitations discussed in §1.

### 3.2 Path-Centric Reward

We propose a path-centric reward that evaluates trajectory quality rather than solely relying on final answer correctness, addressing the three limitations of outcome-based methods. The complete reward function is:

$$R_{\text{total}} = \lambda_p \cdot R_{\text{path}} + \lambda_a \cdot R_{\text{outcome}} + \lambda_f \cdot R_{\text{format}} \quad (4)$$

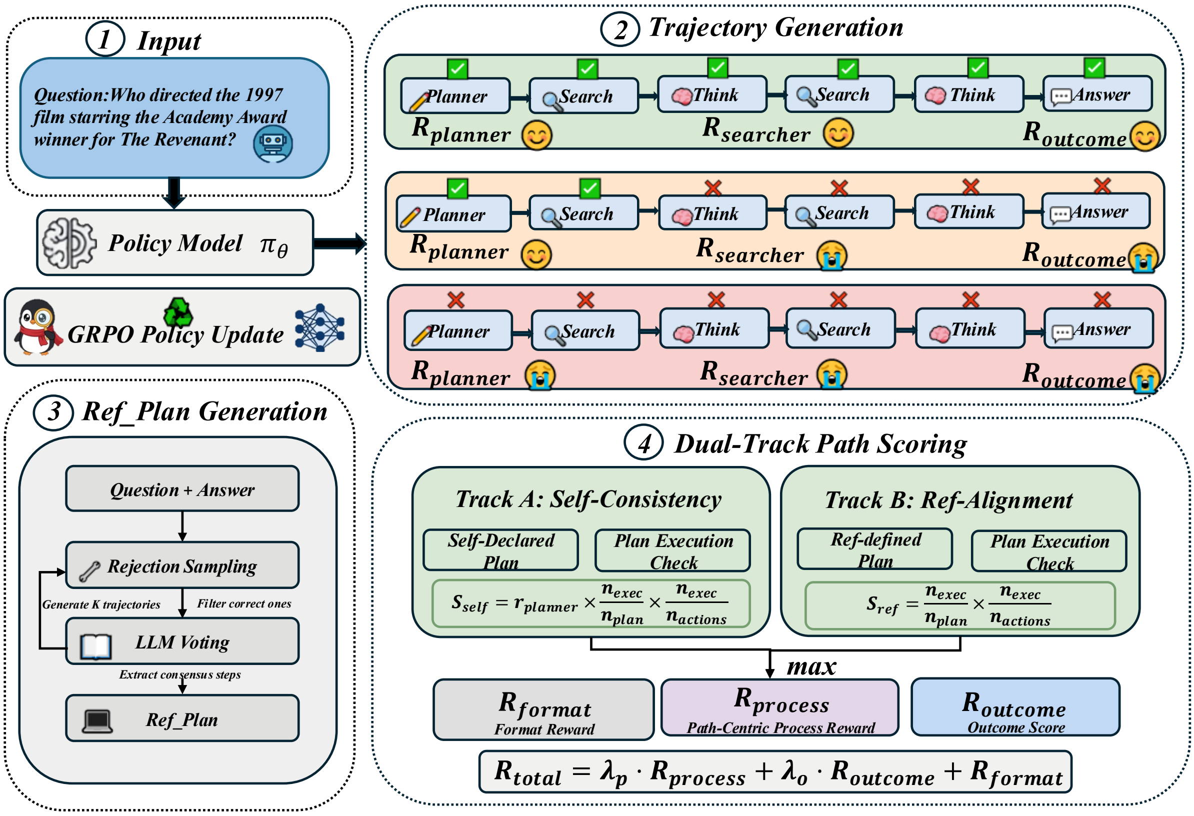


Figure 2: Overview of SEARCH-P1 framework. Our approach introduces path-centric reward shaping for agentic RAG training, comprising: (1) Dual-Track Path Scoring that evaluates trajectories from both self-consistency and reference-alignment perspectives, and (2) Soft Outcome Scoring that extracts training signals even from incorrect answers.

where $R_{\text{path}}$ is the path-centric reward computed via dual-track evaluation, $R_{\text{outcome}}$ is the soft outcome score that extracts signals even from incorrect answers, $R_{\text{format}}$ encourages well-structured outputs, and $\lambda_p$, $\lambda_a$, $\lambda_f$ are balancing coefficients.

#### 3.2.1 Reference Planner Generation

We generate reference planners offline through rejection sampling and LLM voting. For each training sample $(q, a^*)$, we generate $K$ candidate trajectories using a high-capability LLM, filter for correct answers, and apply LLM voting to distill an optimized reference planner $P_{\text{ref}}$:

$$P_{\text{ref}} = \text{Vote}(\{T_i\}_{i=1}^{K} | \text{correct}(T_i)) \tag{5}$$

The voting identifies the minimal set of essential steps across successful trajectories, yielding a reference reasoning path $\mathcal{R}_{\text{ref}} = \{s_1, s_2, \ldots, s_m\}$.

#### 3.2.2 Dual-Track Path Scoring

We evaluate trajectory quality from two complementary perspectives. Track A (Self-Consistency) assesses whether the model effectively executes its own stated plan:

$$S_{\text{self}} = r_{\text{planner}} \times \frac{n_{\text{exec}}^{\text{self}}}{n_{\text{plan}}} \times \frac{n_{\text{exec}}^{\text{self}}}{n_{\text{actions}}} \tag{6}$$

where $r_{\text{planner}}$ rates the plan quality, $n_{\text{exec}}^{\text{self}}$ counts executed steps, $n_{\text{plan}}$ is the total planned steps, and $n_{\text{actions}}$ is the total actions in the trajectory. Track B (Reference-Alignment) measures coverage of essential steps from the reference planner using order-agnostic matching:

$$S_{\text{ref}} = \frac{n_{\text{covered}}}{|\mathcal{R}_{\text{ref}}|} \times \frac{n_{\text{covered}}}{n_{\text{actions}}} \tag{7}$$

where $n_{\text{covered}}$ counts accomplished reference steps regardless of execution order. Both tracks incorporate an efficiency ratio $\frac{n_{\text{effective}}}{n_{\text{actions}}}$ to prevent reward hacking through excessive redundant steps and encourage concise reasoning trajectories. The concrete criteria for determining effective steps and covered steps—including the LLM-based semantic matching procedure—are detailed in Appendix D.3. The final path-centric reward $R_{\text{path}} =$

$\max(S_{\text{self}}, S_{\text{ref}})$ takes the maximum rather than a weighted combination, so that when the reference plan is suboptimal or the model discovers a better strategy, the self-consistency track can dominate without being diluted by a low reference score (and vice versa).

#### 3.2.3 Soft Outcome Scoring

To improve sample efficiency, we extract learning signals from trajectories with incorrect final answers through soft scoring:

$$R_{\text{outcome}} = \begin{cases} 1.0 & \text{if correct} \\ \alpha \cdot r_{\text{acc}} + (1-\alpha) \cdot r_{\text{reason}} & \text{otherwise} \end{cases} \tag{8}$$

where $\alpha = 0.8$, $r_{\text{acc}}$ indicates partial answer correctness and $r_{\text{reason}}$ evaluates reasoning quality independent of the final answer. This converts previously zero-reward failed samples into useful training signals based on their path quality.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** Following prior work, we evaluate on seven public QA benchmarks spanning two categories: (1) **General QA**: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023); (2) **Multi-Hop QA**: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), Musique (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Additionally, we evaluate on **AD-QA**, a fully anonymized proprietary advertising QA dataset containing 1,000 multi-hop test instances from an internal business to assess real-world applicability (details in Appendix A). Following Search-R1, we merge the training sets of NQ and HotpotQA to form a unified training dataset. Evaluation is conducted on all datasets to assess both in-domain (NQ, HotpotQA) and out-of-domain (TriviaQA, PopQA, 2WikiMultiHopQA, Musique, Bamboogle, AD-QA) generalization.

**Models.** We conduct experiments with Qwen2.5-7B-Instruct and Qwen2.5-3B-Instruct (Qwen et al., 2025), denoted as 7B and 3B for brevity. For retrieval, we use the 2018 Wikipedia dump as the knowledge source and E5 as the retriever, with top-3 passages returned per search step.

**Evaluation Metric.** We use Accuracy (ACC) as the primary evaluation metric, which checks whether the ground-truth answer is contained in the model's generated response.

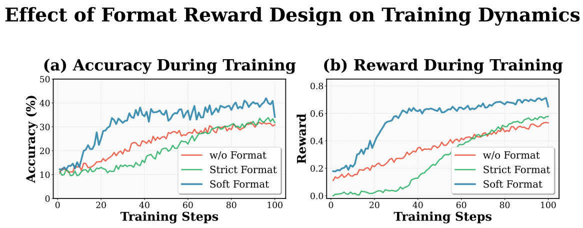


Figure 3: Training dynamics comparison of different format reward strategies. Soft Format (our buffered design) achieves faster ACC improvement and higher stable rewards compared to Strict Format (zero reward for invalid format) and Without Format baseline.

**Baselines.** We compare against the following methods: (1) **Direct Inference**: Generation without retrieval, including direct prompting and Chain-of-Thought (CoT); (2) **Standard RAG**: Single-round retrieval before generation; (3) **Prompt-Based Agentic RAG**: IRCoT and Search-o1 that use prompting for multi-step retrieval; (4) **RL-Based Agentic RAG**: Search-R1 and HiPRAG that use reinforcement learning for training. All RL-based methods share identical training and retrieval configurations (detailed in Appendix B); the only difference is the reward function.

### 4.2 Main Results

As shown in Table 1, SEARCH-P1 achieves the highest average accuracy across both model sizes, outperforming all baselines by a clear margin (+7.7 Avg. ACC over Search-R1 on 7B). The gains are especially pronounced on the internal AD-QA benchmark (+20.6 over Search-R1 on 7B), a real-world advertising QA dataset with complex multi-hop queries, confirming the practical value of path-centric rewards in industrial settings. Notably, the improvements are consistent across model scales, with the 3B model achieving +7.9 Avg. ACC over Search-R1, demonstrating that path-centric rewards are effective even for smaller models.

### 4.3 Ablation Study

We conduct ablation studies to validate the contribution of each reward component in SEARCH-P1: format reward, path-centric reward, and outcome reward.

#### 4.3.1 Format Reward

As shown in Figure 3, we compare three strategies: (1) Soft Format (our buffered design), (2)

| Method | General QA | | | Multi-Hop QA | | | | Avg. | Internal |
|---|---|---|---|---|---|---|---|---|---|
| | NQ† | TriviaQA | PopQA | HotpotQA† | 2Wiki | Musique | Bamboogle | | AD-QA |
| | | | | *Qwen2.5-7B* | | | | | |
| Direct | 13.4 | 40.8 | 14.0 | 18.3 | 25.0 | 3.1 | 12.0 | 18.1 | 10.3 |
| CoT | 4.8 | 18.5 | 5.4 | 9.2 | 11.1 | 2.2 | 23.2 | 10.6 | 8.7 |
| RAG | 34.9 | 58.5 | 39.2 | 29.9 | 23.5 | 5.8 | 20.8 | 30.4 | 60.4 |
| IRCoT | 22.4 | 47.8 | 30.1 | 13.3 | 14.9 | 7.2 | 22.4 | 23.9 | 52.3 |
| Search-o1 | 15.1 | 44.3 | 13.1 | 18.7 | 17.6 | 5.8 | 29.6 | 20.6 | 48.5 |
| Search-R1 | 42.9 | 62.3 | 42.7 | 38.6 | 34.6 | 16.2 | 40.0 | 39.6 | 65.6 |
| HiPRAG | 46.5 | 65.8 | 45.8 | 42.0 | <u>46.1</u> | 14.0 | 40.0 | 42.9 | 75.6 |
| **SEARCH-P1** | **56.6** | **78.6** | **47.5** | **42.9** | 39.8 | **21.8** | **44.0** | **47.3** | **86.2** |
| | | | | *Qwen2.5-3B* | | | | | |
| Direct | 10.6 | 28.8 | 10.8 | 14.9 | 24.4 | 2.0 | 2.4 | 13.4 | 7.8 |
| CoT | 2.3 | 3.2 | 0.5 | 2.1 | 2.1 | 0.2 | 0.0 | 1.5 | 5.2 |
| RAG | 34.8 | 54.4 | 38.7 | 25.5 | 22.6 | 4.7 | 8.0 | 27.0 | 54.7 |
| IRCoT | 11.1 | 31.2 | 20.0 | 16.4 | 17.1 | 6.7 | 24.0 | 18.1 | 45.8 |
| Search-o1 | 23.8 | 47.2 | 26.2 | 22.1 | 21.8 | 5.4 | 32.0 | 25.5 | 42.1 |
| Search-R1 | 39.7 | 56.5 | 39.1 | 33.1 | 31.0 | 12.4 | 23.2 | 33.6 | 58.3 |
| HiPRAG | 43.0 | 59.8 | 42.0 | 36.0 | <u>40.5</u> | 10.8 | 24.0 | 36.6 | 70.2 |
| **SEARCH-P1** | **53.0** | **74.5** | **47.9** | **36.2** | 36.6 | **13.3** | **28.8** | **41.5** | **79.5** |

Table 1: Main results (ACC %) on seven public QA benchmarks and one internal dataset. Best results are in **bold**, second best are <u>underlined</u>. † denotes in-domain datasets used for training; others are out-of-domain. AD-QA is a proprietary advertising QA dataset. HiPRAG results are from our reproduction using the same retrieval setup.

| Strategy | In-D | OOD | Avg. |
|---|---|---|---|
| **Max** (Ours) | **49.8** | **46.3** | **47.3** |
| Weighted (0.5+0.5) | 47.9 | 45.0 | 45.8 |
| Ref-Only (w/o Self-Cons.) | 46.4 | 43.4 | 44.2 |
| Self-Only (w/o Ref-Align.) | 44.2 | 41.3 | 42.1 |
| Search-R1 (no path reward) | 40.8 | 39.2 | 39.6 |

Table 2: Path score aggregation ablation (ACC %, Qwen2.5-7B). "Ref-Only" / "Self-Only" remove the other track (i.e., w/o Self-Consistency and w/o Reference-Alignment, respectively). In-D = avg(NQ, HotpotQA); OOD = avg of the five other public datasets. Per-dataset results are in Appendix F.4.

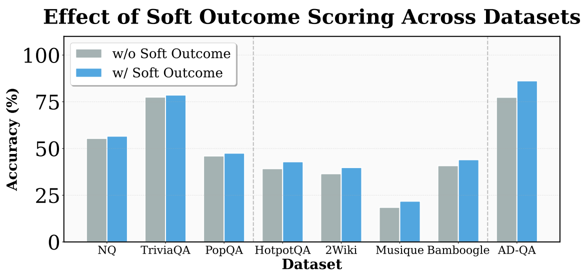


Figure 4: Effect of soft outcome scoring across datasets. Gray bars show accuracy without soft scoring (binary outcome), blue bars show accuracy with soft scoring. Per-dataset results are in Appendix F.6.

Strict Format (zero reward for violations), and (3) Without Format. Our soft format achieves significantly faster convergence by providing continuous gradient feedback, while the strict approach yields near-zero rewards in early training steps due to frequent formatting errors.

#### 4.3.2 Path-Centric Reward

Table 2 studies both the contribution of each track and the choice of aggregation operator. Removing reference-alignment (*Self-Only*) causes a 5.2-point drop and removing self-consistency (*Ref-Only*) a 3.1-point drop, confirming that both external guidance and internal consistency are complementary signals. Crucially, we find that **taking the max is preferable to a weighted average** (+1.5 Avg. ACC over $0.5S_{\text{self}}+0.5S_{\text{ref}}$): when the reference plan is suboptimal but the model discovers a valid alternative path, averaging dilutes the stronger signal with the weaker one, whereas max acts as a logical-OR that preserves whichever track is more informative. This effect is most pronounced on AD-QA (86.2 vs. 82.5), where the diversity of valid reasoning paths is largest.

#### 4.3.3 Outcome Reward

As shown in Figure 4, soft outcome scoring provides modest gains for single-hop tasks (+1.2%), larger improvements for multi-hop QA (+3.5%),

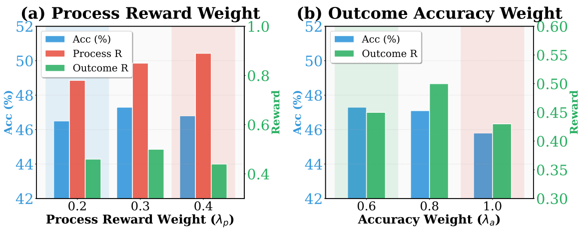


Figure 5: Hyperparameter sensitivity analysis. All rewards are averaged over steps 195–205. (a) Effect of path reward weight $\lambda_p$. (b) Effect of accuracy weight $\lambda_a$. Per-dataset results are in Appendix F.7.

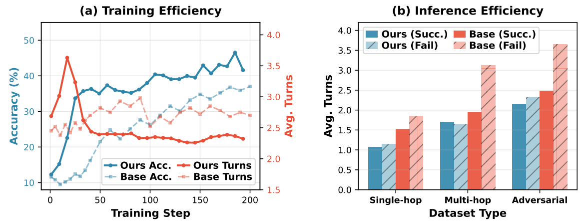


Figure 6: Efficiency analysis. (a) Training efficiency: accuracy and interaction turns comparison between SEARCH-P1 and Search-R1 during training. (b) Inference efficiency: turns by outcome across dataset types.

and the highest gain for AD-QA (+8.8%), confirming that complex scenarios benefit most from partial credit signals.

## 5 Analysis

### 5.1 Hyperparameter Sensitivity

We investigate the impact of two critical hyperparameters in our reward formulation: the path reward weight $\lambda_p$ and the accuracy weight $\lambda_a$.

As shown in Figure 5, both $\lambda_p$ and $\lambda_a$ exhibit clear sweet spots. Too little path weight provides insufficient supervision, while too much induces reward overfitting where path metrics improve but accuracy drops. Similarly, over-weighting accuracy neglects reasoning quality and leads to reward hacking. The optimal configuration ($\lambda_p$=0.3, $\lambda_a$=0.6) balances accuracy as the primary objective with reasoning quality as a regularizer.

### 5.2 Efficiency Analysis

**Training Efficiency** Figure 6(a) compares training dynamics. SEARCH-P1 converges significantly faster, reaching Search-R1's final accuracy (~40%) within 60 steps versus over 150. Meanwhile, SEARCH-P1's interaction turns steadily decrease, indicating path-centric rewards guide toward higher accuracy and more concise reasoning, while Search-R1's turns remain flat or increase.

**Inference Efficiency** Figure 6(b) compares turn distributions across dataset types. Two key findings emerge: (1) Both methods require more turns for complex adversarial queries. (2) SEARCH-P1 maintains consistent turn counts between successful and unsuccessful cases, while Search-R1 exhibits larger gaps for multi-hop (+60%) and adversarial (+47%) tasks.

| Model | RL | Single | Multi | AD |
|---|---|---|---|---|
| Qwen2.5-3B | GRPO | **58.5** | **28.7** | **79.5** |
| Qwen2.5-3B | PPO | 57.2 | 27.5 | 77.8 |
| Llama-3.2-3B | GRPO | 56.8 | 27.1 | 76.3 |
| Llama-3.2-3B | PPO | 55.6 | 26.2 | 74.6 |

Table 3: ACC (%) across base models and RL algorithms. All models use Instruct versions. Per-dataset results are in Appendix F.5.

### 5.3 Model and RL Algorithm Analysis

Table 3 examines the impact of base models and RL algorithms. Qwen2.5-3B-Instruct (Qwen et al., 2025) slightly outperforms Llama-3.2-3B-Instruct (Grattafiori et al., 2024) across all task types, likely due to stronger instruction-following and reasoning capabilities in the base model. GRPO (Shao et al., 2024) achieves marginally higher accuracy than PPO (Schulman et al., 2017); however, PPO exhibits more stable training dynamics with lower variance across runs. Importantly, path-centric rewards yield consistent gains across all model–algorithm combinations, suggesting that our approach is orthogonal to the choice of base model and RL algorithm.

### 5.4 LLM Evaluator Analysis

Our dual-track scoring and soft outcome scoring rely on an external LLM evaluator during training (at inference time, no evaluator calls are needed). To examine sensitivity, we extend our evaluator study to six models across three capability tiers: frontier proprietary models (DeepSeek-V3.2, HY 2.0-Instruct, GLM-5, GPT-4o) and open-source models (Qwen3-32B-Instruct, Qwen3-8B-Instruct). For each evaluator, we retrain SEARCH-P1 from scratch on Qwen2.5-7B and sample 200 trajectories per run to measure human agreement on plan quality, step coverage, and outcome scoring.

As shown in Table 4, three findings emerge. (1)

| Evaluator | ACC (%) In-D | OOD | Avg. | Human Agree. (%) Plan | Step | Outc. |
|---|---|---|---|---|---|---|
| DS V3.2 | **50.5** | **46.6** | **47.8** | 92.0 | **95.5** | **90.0** |
| HY 2.0 (Ours) | 49.8 | 46.3 | 47.3 | 91.2 | 94.5 | 88.7 |
| GLM-5 | 49.4 | 45.8 | 46.9 | 90.0 | 94.0 | 88.0 |
| GPT-4o | 48.6 | 45.9 | 46.7 | **93.0** | 94.0 | 89.5 |
| Qwen3-32B | 48.7 | 45.2 | 46.2 | 86.5 | 90.0 | 82.5 |
| Qwen3-8B | 46.0 | 42.2 | 43.3 | 80.0 | 85.5 | 74.0 |
| *Search-R1* | *40.8* | *39.2* | *39.6* | — | | |

Table 4: Sensitivity to the LLM evaluator (Qwen2.5-7B). In-D = avg(NQ, HotpotQA); OOD = avg of the five other public datasets. *Search-R1* uses no evaluator. Per-dataset results are in Appendix F.8.

**Frontier models form a tight top tier.** DS V3.2, HY 2.0, GLM-5, and GPT-4o span only 1.1 Avg. ACC points (47.8 vs. 46.7), with no single model dominating all datasets. This cross-run stability ($\sigma = 0.3$) confirms that SEARCH-P1 is not tightly coupled to a specific evaluator. (2) **Open-source evaluators remain effective.** Qwen3-32B reaches 46.2 Avg. ACC (only $-1.1$ relative to our default HY 2.0, and $+6.6$ over Search-R1), making SEARCH-P1 fully reproducible without proprietary APIs. Even Qwen3-8B retains a $+3.7$ gain over Search-R1, though its lower outcome-level human agreement (74.0%) introduces noisier reward signals on multi-hop tasks. (3) **Step coverage is the most robust signal.** Even the weakest 8B evaluator maintains 85.5% step-level agreement, indicating that structural matching in our path-centric reward degrades gracefully while outcome judgment is most sensitive to evaluator capacity. The step/outcome agreement gap widens from 2.5 (DS V3.2) to 11.5 points (Qwen3-8B), so In-D accuracy is more evaluator-sensitive than OOD. In practice, 8B evaluators suffice when step coverage dominates the training signal, and frontier models are only needed when outcome correctness is the bottleneck.

### 5.5 Case Study

To qualitatively illustrate SEARCH-P1's advantages, we present case studies comparing reasoning trajectories with baseline methods. Appendix E provides a representative example from multi-hop QA, demonstrating how path-centric rewards lead to more structured decomposition, precise query formulation, and effective information synthesis.

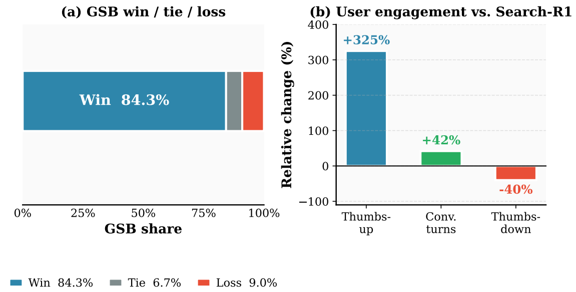


Figure 7: Online A/B test of SEARCH-P1 against the previous Search-R1-based production model on an industrial conversational search product (expert-graded queries over millions of documents). (a) GSB win/tie/loss distribution. (b) User-engagement relative changes.

### 5.6 Industrial Deployment

Beyond academic benchmarks, SEARCH-P1 has been deployed as the core reasoning engine of a conversational search product serving millions of documents on a major industrial platform. Because the LLM evaluator is used *only during training*, the deployed policy incurs zero inference-time overhead relative to Search-R1-style agents. Figure 7 reports a side-by-side online A/B test against the previous Search-R1-based production model, with queries graded by human experts under a Good/Same/Bad (GSB) protocol. SEARCH-P1 achieves a **Win/Tie/Loss of 84.3% / 6.7% / 9.0%**; on user-engagement signals, the thumbs-up rate relatively increases by 325% (to 13.17%), conversational turns increase by $+42\%$, and the thumbs-down rate drops by $-40\%$. These online gains corroborate the AD-QA results in Table 1 and confirm that path-centric reward shaping transfers from academic QA to enterprise-scale agentic retrieval. A live production trace is shown in Appendix E (Figure 12).

## 6 Conclusion

We presented SEARCH-P1, a framework that introduces path-centric reward shaping for agentic RAG training. By evaluating the structural quality of entire reasoning paths rather than isolated elements, our approach provides fine-grained supervision while respecting the inherent diversity of multi-step reasoning. Extensive experiments on public QA benchmarks and an internal advertising dataset demonstrate significant improvements in accuracy and efficiency, validating path-centric rewards in both academic and industrial settings.

## Ethics Statement

Our work focuses on improving the training of AI systems for information retrieval and reasoning. We use publicly available datasets for training and evaluation. The internal AD-QA dataset is fully anonymized with all personally identifiable information removed prior to use. The improved efficiency of agentic RAG systems could reduce computational resources required for deployment, contributing to more sustainable AI.

## References


Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieval step by step for large language models. *Preprint*, arXiv:2502.01142.

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

Lingxiang Hu, Chenyang Hei, Fuliang Li, Chengxi Gao, Jiaxing Shen, and Xingwei Wang. 2025. Smarttc: A real-time ml-based traffic classification with smartnic. In *2025 IEEE/ACM 33rd International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE.

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Hao Peng, Julia Hockenmaier, and Tong Zhang. 2025. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *Preprint*, arXiv:2503.12759.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jian Li, Xiaoxi Li, Yan Zheng, Yizhang Jin, Shuo Wang, Jiafu Wu, Yabiao Wang, Chengjie Wang, and Xiaotong Yuan. 2025a. A survey on ai search with large language models. *Preprints*.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. Search-o1: Agentic search-enhanced large reasoning models. *Preprint*, arXiv:2501.05366.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zeyang Sha, Shiwen Cui, and Weiqiang Wang. 2025. Sem: Reinforcement learning for search-efficient large language models. *Preprint*, arXiv:2505.07903.

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

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

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

Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025b. R1-searcher++: Incentivizing the dynamic knowledge acquisition of llms via reinforcement learning. *Preprint*, arXiv:2505.17005.

Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Yang Song, and Han Li. 2025. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 1251–1261, New York, NY, USA. Association for Computing Machinery.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025. Chain-of-retrieval augmented generation. *Preprint*, arXiv:2501.14342.

Zhao Wang, Ziliang Zhao, and Zhicheng Dou. 2026. ProRAG: Process-supervised reinforcement learning for retrieval-augmented generation. *Preprint*, arXiv:2601.21912.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. 2025a. Hiprag: hierarchical process rewards for efficient agentic retrieval augmented generation. *arXiv preprint arXiv:2510.07794*.

Peilin Wu, Mian Zhang, Xinlu Zhang, Xinya Du, and Zhiyu Zoey Chen. 2025b. Search wisely: Mitigating sub-optimal agentic searches by reducing uncertainty. *Preprint*, arXiv:2505.17281.

Tianle Xia, Liang Ding, Guojia Wan, Yibing Zhan, Bo Du, and Dacheng Tao. 2025. Improving complex reasoning over knowledge graph with logic-aware curriculum tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12881–12889.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, and 6 others. 2025a. The landscape of agentic reinforcement learning for llms: A survey. *Preprint*, arXiv:2509.02547.

Qi Zhang, Shouqing Yang, Lirong Gao, and Hao Chen. 2025b. LeTS: Learning to think-and-search via process-and-outcome reward hybridization. *Preprint*, arXiv:2505.17447.

Tianhua Zhang, Kun Li, Junan Li, and Yunxiang Li. 2026. TreePS-RAG: Tree-based process supervision for reinforcement learning in agentic rag. *Preprint*, arXiv:2601.06922.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, and 1 others. 2025c. Process vs. outcome reward: Which is better for agentic rag reinforcement learning. *arXiv preprint arXiv:2505.14069*.

# A AD-QA Dataset

AD-QA is a fully anonymized multi-hop QA benchmark from a real-world advertising domain, containing 1,000 test instances requiring multi-step reasoning across domains such as campaign configuration, bidding strategies, audience targeting, and conversion tracking. All instances are derived from authentic user queries with all personally identifiable information removed.

Each question requires synthesizing information from at least two distinct knowledge domains, making it a challenging benchmark for multi-hop reasoning in enterprise settings. Ground-truth answers are curated by domain experts and verified through cross-validation.

# B Implementation Details

## B.1 Training Configuration

For GRPO training, we set the policy learning rate to $1 \times 10^{-6}$ with a warm-up ratio of 0.1. Training is conducted on 8×H20 GPUs using a total batch size of 512, with a mini-batch size of 256. The micro-batch size per GPU is set to 8 for 7B models and 16 for 3B models.

The maximum prompt length and response length are both set to 4,096 tokens, with a maximum model context length of 8,192 tokens. We enable gradient checkpointing for memory efficiency and use Fully Sharded Data Parallel (FSDP) with reference model parameter offloading.

For efficient rollout generation, we use SGLang with tensor parallel size of 1 and GPU memory utilization of 0.8 (7B) or 0.75 (3B). Rollout sampling uses temperature $\tau = 0.6$, top-$k = 20$, and top-$p = 0.95$. We sample 16 candidate responses per prompt for 7B models and 32 for 3B models with an over-sample rate of 0.1. The KL divergence coefficient $\beta$ is set to 0.001 with low-variance KL loss, and the clip ratio ranges from 0.2 to 0.28.

## B.2 Reward Computation

The path-centric reward combines three components with the following default weights: format reward weight $\lambda_f = 0.1$, path reward weight $\lambda_p = 0.3$, and outcome accuracy weight $\lambda_a = 0.6$. The reference planner uses a proprietary instruction-tuned model (anonymized as HY 2.0-Instruct) to generate guidance trajectories, which are cached offline before training to avoid runtime overhead.

For self-consistency scoring, we sample 3 independent reasoning paths per query and compute pairwise agreement using Jaccard similarity on extracted evidence spans. The soft outcome scoring applies a decay factor of 0.5 for partial matches when the final answer is incorrect but the reasoning path demonstrates high path quality.

## B.3 Computational Cost

Reference planners are generated offline for all 90K training samples using HY 2.0-Instruct, with each sample requiring on average 1.91 LLM calls. This is a one-time cost cached before RL training and amortized over all subsequent runs.

## B.4 Inference Settings

During inference, we set the maximum action budget $B = 4$, allowing up to 4 search-reason iterations per query. The retriever returns top-3 passages per search step. We use sampling with temperature 0.6 and top-$p$ 0.95 for validation. Model checkpoints are saved every 10 steps, and we select the checkpoint with the highest validation accuracy for final evaluation.

# C Algorithms

This section provides algorithmic descriptions of the key components in SEARCH-P1: (1) offline reference planner generation, (2) agentic RAG inference, and (3) path-centric reward computation.

Algorithm 1 describes reference planner generation using a high-capability LLM (HY 2.0-Instruct) to produce structured plans and reference reasoning paths, cached offline for training.

Algorithm 2 illustrates agentic RAG inference: the model iteratively generates reasoning, issues search queries via <tool_call>, and receives retrieved passages as <tool_response> until the action budget is exhausted or an answer is produced.

Algorithm 3 details the reward computation combining format, dual-track path-centric, and soft outcome signals.

# D Prompt Templates

This section presents the prompt templates used in SEARCH-P1 for inference, reference planner generation, and reward evaluation.

## D.1 Agentic RAG Inference Prompt

Figure 8 shows the prompt template used during both training rollouts and inference. The prompt instructs the model to decompose questions into sub-tasks, execute searches iteratively,

**Algorithm 1** Reference Planner Generation

**Require:** Training dataset $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{N}$, reference LLM $\mathcal{M}_{\text{ref}}$
**Ensure:** Reference trajectories $\mathcal{T}_{\text{ref}} = \{(p_i, r_i)\}_{i=1}^{N}$
**for** each $(q, a) \in \mathcal{D}$ **do**
  $\text{prompt}_p \leftarrow \textsc{PlannerPrompt}(q)$ {Generate planning prompt}
  $p \leftarrow \mathcal{M}_{\text{ref}}(\text{prompt}_p)$ {Generate reference plan}
  $\text{prompt}_r \leftarrow \textsc{ReasoningPrompt}(q, p)$ {Generate reasoning prompt}
  $r \leftarrow \mathcal{M}_{\text{ref}}(\text{prompt}_r)$ {Generate reference reasoning path}
  $\mathcal{T}_{\text{ref}} \leftarrow \mathcal{T}_{\text{ref}} \cup \{(p, r)\}$
**end for**
**return** $\mathcal{T}_{\text{ref}}$

**Algorithm 2** Agentic RAG Inference

**Require:** Question $q$, policy model $\pi$, retriever $\mathcal{R}$, action budget $B$, top-$K$
**Ensure:** Generated trajectory $y$ with final answer
$y \leftarrow$ `<reasoning>`; $t \leftarrow 1$
**while** $t \leq B$ **do**
  $\Delta \leftarrow \textsc{Generate}(\pi, y)$ **until** `</tool_call>` or `</answer>`
  $y \leftarrow y \,\|\, \Delta$
  **if** $\textsc{Contains}(y,$ `</answer>`$)$ **then**
    **break** {Final answer generated}
  **end if**
  **if** $\textsc{Contains}(\Delta,$ `<tool_call>`$)$ **then**
    $\text{query} \leftarrow \textsc{Extract}(\Delta,$ `<tool_call>`$)$
    $\text{docs} \leftarrow \mathcal{R}(\text{query}, K)$ {Retrieve top-$K$ passages}
    $y \leftarrow y \,\|$ `<tool_response>` $\|$ docs $\|$ `</tool_response>`
    $t \leftarrow t + 1$
  **end if**
**end while**
**if not** $\textsc{Contains}(y,$ `</answer>`$)$ **then**
  $y \leftarrow y \,\|$ `<answer>` $\| \; \textsc{Generate}(\pi, y)$ **until** `</answer>`
**end if**
**return** $y$

and produce structured outputs with `<reasoning>`, `<tool_call>`, and `<answer>` tags.

### D.2 Reference Planner Generation Prompt

Figure 9 shows the prompt used to generate reference plans and reasoning paths from HY 2.0-Instruct. Given a question and its correct answer, the reference LLM produces an optimized search strategy that serves as guidance during path reward computation.

### D.3 Dual-Track Evaluation Prompt

Figure 10 presents the prompt used for dual-track path evaluation. An evaluator LLM assesses the model's trajectory along two dimensions: self-consistency (execution of its own plan) and reference-alignment (coverage of expert reference steps), along with outcome quality scoring.

## E Case Study

To qualitatively illustrate SEARCH-P1's advantages, we present two representative cases: an academic multi-hop example from MuSiQue (Appendix E.1) and a live trace from our industrial deployment (Appendix E.2).

### E.1 Multi-Hop Reasoning Comparison

Figure 11 compares Search-R1 and SEARCH-P1 on a multi-hop question. Without explicit planning, Search-R1 misinterprets "rock & roll" as a genre descriptor, retrieving information about the wrong entity. In contrast, SEARCH-P1's planning

**Algorithm 3** SEARCH-P1 Reward Computation

**Require:** Trajectory $y$, ground truth $a^*$, reference plan $p_{\text{ref}}$, reference path $r_{\text{ref}}$
**Ensure:** Total reward $R(y)$

*// Format Reward*
**if** $\text{VALIDFORMAT}(y)$ **and** $\text{HASANSWER}(y)$ **and** $\text{HASTOOLCALL}(y)$ **then**
  $r_f \leftarrow 0.1$
**else if** $\text{HASANSWER}(y)$ **and** $\text{HASTOOLRESPONSE}(y)$ **then**
  $r_f \leftarrow 0.05$
**else**
  **return** 0 {Invalid trajectory}
**end if**

*// Path-Centric Reward via Dual-Track Evaluation*
eval $\leftarrow \text{LLMEVALUATE}(y, p_{\text{ref}}, r_{\text{ref}})$ {Call evaluator LLM}
$r_{\text{planner}} \leftarrow$ eval.planner_score {Plan quality: 0.2/0.6/1.0/1.2}

*// Track A: Self-Consistency*
$s_{\text{self}} \leftarrow r_{\text{planner}} \times \frac{\text{eval.eff\_steps\_self}}{\text{eval.model\_plan\_steps}}$

*// Track B: Reference-Alignment*
$s_{\text{ref}} \leftarrow \frac{\text{eval.eff\_steps\_ref}}{|\text{steps}(r_{\text{ref}})|}$

$r_p \leftarrow \max(s_{\text{self}}, s_{\text{ref}})$ {Best of dual tracks}

*// Outcome Reward with Soft Scoring*
**if** $\text{EXACTMATCH}(\text{GETANSWER}(y), a^*)$ **then**
  $r_o \leftarrow 1.0$
**else**
  $r_o \leftarrow 0.8 \times \text{eval.acc\_score} + 0.2 \times \text{eval.reason\_score}$
**end if**

$R(y) \leftarrow \lambda_f \cdot r_f + \lambda_p \cdot r_p + \lambda_o \cdot r_o$
**return** $R(y)$

correctly identifies "Bang Bang Rock & Roll" as a complete album title, leading to the correct answer.

### E.2 Online Industrial Deployment Case

Figure 12 shows a live trace captured from the production conversational search system described in §5.6, which serves millions of documents on a major industrial platform. Because the product operates in Chinese, both the user query and the agent's outputs are in the native language. This example was withheld from the original anonymous submission to preserve double-blind review and is included here now that anonymization constraints no longer apply.

The user asks a compound advertising configuration question that chains three knowledge domains: *Moments ad* $\rightarrow$ *mini-program landing page* $\rightarrow$ *WeChat-Shop purchase*. The left panel shows the agent's multi-turn reasoning trace. In Round 1, the planner formulates a focused query about the jump-path configuration, retrieves three relevant documents, and extracts concrete configuration constraints (e.g., choosing *Moments* placement and *product sales* campaign objective). In Round 2, the planner drills down into follow-up configuration details (account authorization, conversion-tracking setup) identified in Round 1 as missing. The agent terminates after two tool calls, indicating that path-centric reward shaping leads to efficient, non-redundant search.

The right panel shows the final answer, which the agent organizes into a structured three-stage workflow (*1. Prerequisites*, *2. Ad Creation*, *3. Landing-page Configuration*, *4. Testing & Opti-*

**Agentic RAG Inference Prompt**

You are a meticulous **Deep Research Agent**. Your goal is to provide a comprehensive and accurate answer by conducting multiple rounds of search.
**## CRITICAL INSTRUCTIONS**
**1. Detailed Planning (<reasoning>):**
- In the first turn, you MUST break the question down into **multiple dependent sub-questions**.
- Focus on one sub-question at a time.

**2. Step-by-Step Execution (<tool_call>):**
- Execute only ONE search query per turn.
- After receiving results, verify: "Is this sufficient? Do I need more details?"

**3. No Guessing:**
- If results are incomplete, issue another search. Do NOT hallucinate.

**4. Final Answer (<answer>):**
- Only output <answer> when ALL necessary information is gathered.

**## CURRENT TASK**
Question: `{question}`

Figure 8: Prompt template for agentic RAG inference. The model is instructed to plan, search iteratively, and provide structured outputs.

*mization*) with actionable bullet-level instructions. This mirrors the explicit-planning and structured-synthesis pattern observed in the MuSiQue case (Figure 11) and corroborates the AD-QA accuracy gains in Table 1 as well as the online A/B test outcomes in §5.6.

# F Additional Results

## F.1 Impact of Retrieved Documents per Search

Table 5 shows how the number of retrieved documents per search iteration affects model performance. Retrieving too few documents may miss relevant information, while retrieving too many can introduce noise and increase context length.

## F.2 Effect of Format Reward on Output Compliance

Table 6 analyzes the relationship between the format reward component and the model's ability to produce properly formatted outputs.

## F.3 Search Iterations Analysis

Table 7 presents the distribution of search iterations for successful and failed cases across different datasets.

**Key Observations.** (1) General QA datasets achieve most successes with single-iteration searches. (2) Multi-hop datasets show successful cases concentrated at 2 iterations. (3) Failed cases consistently show higher 3+ iteration rates, suggesting excessive searching indicates difficulty. (4) The 3B model requires slightly more iterations than 7B.

## F.4 Detailed Ablation on Path Score Aggregation

Table 8 provides the complete per-dataset breakdown extending Table 2 in the main paper, including the weighted-average variant for the 7B model.

## F.5 Detailed Model and RL Algorithm Analysis

Table 9 extends Table 3 with per-dataset accuracy for different base models and RL algorithms.

## F.6 Detailed Soft Outcome Scoring Analysis

Table 10 provides the per-dataset breakdown of the soft outcome scoring ablation (corresponding to Figure 4 in the main paper).

## F.7 Detailed Hyperparameter Sensitivity Analysis

Tables 11 and 12 provide the per-dataset breakdown for hyperparameter sensitivity analysis (corresponding to Figure 5 in the main paper).

## F.8 Detailed LLM Evaluator Analysis

Table 13 provides the per-dataset breakdown for the six LLM evaluators studied in Table 4. The Qwen3-

**Reference Planner Generation Prompt**

You are an expert planner and reasoning optimizer.
**Current Question:** `{question}`
**Correct Answer:** `{golden_answers}`
Your task is to generate:
**1. Optimized Reasoning Path:** A sequence of search queries that would lead directly to the correct answer in the most efficient way. Format as a numbered list.
**2. Optimized Planner:** A concise, step-by-step instruction on how a reasoning agent should solve this question correctly and efficiently.
**Important:**

- Focus on the minimal set of queries needed.
- Avoid redundant or inefficient steps.

**Output format:**

```
<correct_reasoning_path>
1. query 1
2. query 2
</correct_reasoning_path>
<optimized_planner>
To solve this, first search for... then...
</optimized_planner>
```

Figure 9: Prompt template for reference planner generation. HY 2.0-Instruct generates optimal search strategies for each training sample.

8B evaluator shows larger drops on multi-hop datasets (e.g., −4.0 on 2Wiki, −4.0 on MuSiQue) where accurately counting covered reasoning steps is more challenging; nevertheless, it still maintains a +3.7 Avg. ACC advantage over Search-R1, confirming that path-centric rewards provide value even with modest evaluator capacity.

**Dual-Track Evaluation Prompt**

You are an expert RL researcher evaluating an AI agent's trajectory.
Your task is to conduct a **Dual-Track Evaluation**:
1. **Self-Consistency Track**: How well did the agent execute its OWN plan?
2. **Reference-Alignment Track**: How well did the agent follow the Expert plan?
3. **Outcome Evaluation**: Assess accuracy and reasoning quality.

**Evaluation Inputs:**
- **Question**: `{question}`
- **Correct Answer**: `{golden_answers}`
- **Reference Planner**: `{ref_planner}`
- **Reference Path**: `{ref_reasoning_path}`
- **Model Trajectory**: `{trajectory}`

**Scoring Criteria:**
- **Planner Score**: 0.2 (Bad) / 0.6 (Average) / 1.0 (Good) / 1.2 (Excellent)
- **Outcome Accuracy**: 0.0 (Wrong) / 0.5 (Partial) / 1.0 (Correct)
- **Reasoning Quality**: 0.0 / 0.5 / 0.8 / 1.0

**Output:** JSON with `planner_score`, `model_plan_steps`, `effective_steps_self`, `effective_steps_ref`, `outcome_accuracy_score`, `outcome_reasoning_score`.

Figure 10: Prompt template for dual-track evaluation. The evaluator LLM assesses both self-consistency and reference-alignment of model trajectories.

**Query: Who is the lead singer of the band performing Bang Bang Rock & Roll? Answer: Eddie Argos**

**Search-R1 (Incorrect)**

<think> I need to find who performs "Bang Bang Rock & Roll." </think>
<search> *song Bang Bang rock and roll genre* </search>
<result> In 2017, Dan Auerbach collaborated with Robert Finley on "Bang Bang," featuring a classic rock and roll sound... </result>
<search> *lead singer of The Black Keys* </search>
<result> Dan Auerbach serves as the lead vocalist... </result>
**Final: ✗ Dan Auerbach**

**SEARCH-P1 (Correct)**

<reasoning> I need to: 1. Find the band performing "Bang Bang Rock & Roll". 2. Find the lead singer. </reasoning>
<tool_call> *"Bang Bang Rock & Roll" album* </tool_call>
<tool_response> "Bang Bang Rock & Roll" is the debut album by British rock band **Art Brut**... </tool_response>
<tool_call> *lead singer of Art Brut* </tool_call>
<tool_response> ...frontman **Eddie Argos**' enthusiastic vocal delivery... </tool_response>
**Final: ✓ Eddie Argos**

Figure 11: Comparison of reasoning trajectories. Search-R1's imprecise query retrieves valid but irrelevant results; SEARCH-P1's planning-driven query retrieves the correct information. Highlighted text shows search queries.


Figure 12: A live production case of SEARCH-P1 serving a real advertising query on a major industrial conversational search platform. **Left:** the agent's two-round reasoning trace (planner + search queries + retrieved evidence). **Right:** the final structured answer organized as a four-stage setup-ads-landing-optimization workflow. Text is in Chinese, the product's native language. The behavior mirrors Figure 11: explicit planning, precise query formulation, and structured synthesis.

| Model | # Docs | NQ | General QA TriviaQA | PopQA | HotpotQA | Multi-Hop QA 2Wiki | MuSiQue | Bamboogle | Avg. | Internal AD-QA |
|---|---|---|---|---|---|---|---|---|---|---|
| *7B* | 1 | 45.8 | 67.2 | 38.5 | 35.2 | 32.0 | 15.5 | 36.0 | 38.6 | 72.8 |
| | 2 | 52.0 | 74.2 | 44.0 | 39.5 | 36.8 | 18.8 | 41.0 | 43.8 | 80.5 |
| | 3 | **56.6** | **78.6** | **47.5** | **42.9** | **39.8** | 21.8 | 44.0 | **47.3** | **86.2** |
| | 5 | 55.8 | 77.8 | 46.8 | 42.5 | 39.2 | **22.4** | **45.2** | 47.1 | 85.5 |
| | 10 | 53.5 | 74.5 | 44.5 | 40.5 | 37.2 | 20.1 | 42.8 | 44.7 | 82.5 |
| *3B* | 1 | 42.8 | 63.5 | 38.2 | 28.8 | 28.5 | 9.2 | 21.2 | 33.2 | 64.5 |
| | 2 | 48.5 | 69.8 | 43.6 | 32.8 | 33.2 | 11.5 | 25.2 | 37.8 | 73.2 |
| | 3 | **53.0** | **74.5** | **47.9** | 36.2 | **36.6** | **13.3** | 28.8 | **41.5** | **79.5** |
| | 5 | 52.2 | 73.8 | 47.2 | **36.5** | 36.0 | 12.8 | **30.4** | 41.3 | 78.8 |
| | 10 | 49.8 | 71.2 | 45.0 | 33.8 | 34.2 | 11.8 | 27.6 | 39.1 | 75.5 |

Table 5: Performance (ACC %) with different numbers of retrieved documents per search. Retrieving 3 documents achieves the best average performance. While 5 documents shows advantages on specific datasets (MuSiQue, Bamboogle for 7B; HotpotQA, Bamboogle for 3B), the overall best configuration is 3 documents.

| Model | Method | NQ | General QA TriviaQA | PopQA | HotpotQA | Multi-Hop QA 2Wiki | MuSiQue | Bamboogle | Internal AD-QA |
|---|---|---|---|---|---|---|---|---|---|
| *7B* | w/o Format Reward | 82.8 | 84.2 | 81.5 | 79.1 | 76.8 | 75.2 | 83.6 | 88.2 |
| | w/ Format Reward | **95.1** | **95.6** | **94.2** | **91.5** | **88.8** | **87.6** | **96.4** | **97.5** |
| *3B* | w/o Format Reward | 72.8 | 75.5 | 72.2 | 67.1 | 61.8 | 62.4 | 65.2 | 78.8 |
| | w/ Format Reward | **87.2** | **88.1** | **85.4** | **80.2** | **74.8** | **75.1** | **78.0** | **91.5** |

Table 6: Format compliance rate (%) with and without format reward. Adding format reward significantly improves the model's ability to produce properly structured responses with parseable answers.

| Model | Dataset | Successful Cases | | | Failed Cases | | |
|---|---|---|---|---|---|---|---|
| | | 1 iter | 2 iter | 3+ iter | 1 iter | 2 iter | 3+ iter |
| *7B* | NQ | 68.5% | 22.3% | 9.2% | 45.2% | 28.6% | 26.2% |
| | TriviaQA | 72.1% | 19.8% | 8.1% | 48.3% | 26.4% | 25.3% |
| | PopQA | 65.8% | 24.5% | 9.7% | 42.1% | 29.8% | 28.1% |
| | HotpotQA | 28.5% | 48.2% | 23.3% | 35.4% | 30.1% | 34.5% |
| | 2Wiki | 25.1% | 50.6% | 24.3% | 33.8% | 29.5% | 36.7% |
| | MuSiQue | 18.5% | 52.3% | 29.2% | 31.2% | 28.6% | 40.2% |
| | Bamboogle | 22.4% | 45.6% | 32.0% | 28.3% | 25.4% | 46.3% |
| | AD-QA | 35.2% | 42.5% | 22.3% | 22.8% | 28.5% | 48.7% |
| | **Average** | 42.0% | 38.2% | 19.8% | 35.9% | 28.4% | 35.7% |
| *3B* | NQ | 62.3% | 26.1% | 11.6% | 40.5% | 30.2% | 29.3% |
| | TriviaQA | 66.8% | 23.4% | 9.8% | 44.1% | 28.5% | 27.4% |
| | PopQA | 60.2% | 28.3% | 11.5% | 38.6% | 31.2% | 30.2% |
| | HotpotQA | 22.4% | 45.8% | 31.8% | 30.2% | 28.5% | 41.3% |
| | 2Wiki | 20.3% | 47.2% | 32.5% | 28.5% | 27.8% | 43.7% |
| | MuSiQue | 14.2% | 48.6% | 37.2% | 26.4% | 26.2% | 47.4% |
| | Bamboogle | 16.8% | 42.1% | 41.1% | 24.5% | 23.8% | 51.7% |
| | AD-QA | 28.5% | 40.2% | 31.3% | 18.2% | 25.6% | 56.2% |
| | **Average** | 36.4% | 37.7% | 25.9% | 31.4% | 27.7% | 40.9% |

Table 7: Distribution of search iterations for successful and failed cases. General QA datasets (NQ, TriviaQA, PopQA) show high success rates with single-iteration searches, while Multi-Hop QA datasets require more iterations. Failed cases consistently show higher proportions of 3+ iterations, suggesting that excessive searching indicates difficulty in finding relevant information.

| Aggregation Strategy | In-Domain | | Out-of-Domain | | | | | Avg. | Internal |
|---|---|---|---|---|---|---|---|---|---|
| | NQ | HotpotQA | TriviaQA | PopQA | 2Wiki | MuSiQue | Bamboogle | | AD-QA |
| *Qwen2.5-7B* | | | | | | | | | |
| **Max** (Ours) | **56.6** | **42.9** | **78.6** | **47.5** | 39.8 | **21.8** | **44.0** | **47.3** | **86.2** |
| Weighted (0.5+0.5) | 54.2 | 41.5 | 77.2 | 46.0 | **40.2** | 19.8 | 41.6 | 45.8 | 82.5 |
| Ref-Only (w/o Self-Cons.) | 53.0 | 39.8 | 75.5 | 44.8 | 37.2 | 19.2 | 40.2 | 44.2 | 82.5 |
| Self-Only (w/o Ref-Align.) | 50.8 | 37.5 | 73.0 | 43.2 | 34.0 | 16.8 | 39.5 | 42.1 | 78.8 |
| Search-R1 (no path reward) | 42.9 | 38.6 | 62.3 | 42.7 | 34.6 | 16.2 | 40.0 | 39.6 | 65.6 |
| *Qwen2.5-3B* | | | | | | | | | |
| **Max** (Ours) | **53.0** | **36.2** | **74.5** | **47.9** | **36.6** | **13.3** | **28.8** | **41.5** | **79.5** |
| Ref-Only (w/o Self-Cons.) | 49.8 | 33.8 | 71.8 | 45.2 | 34.0 | 11.8 | 26.0 | 38.9 | 75.2 |
| Self-Only (w/o Ref-Align.) | 47.8 | 31.2 | 68.5 | 43.4 | 30.5 | 10.1 | 24.0 | 36.5 | 70.5 |
| Search-R1 (no path reward) | 39.7 | 33.1 | 56.5 | 39.1 | 31.0 | 12.4 | 23.2 | 33.6 | 58.3 |

Table 8: Detailed path score aggregation ablation (ACC %). Weighted averaging slightly outperforms Max on 2Wiki, where both tracks are already close in quality and averaging provides smoothing, but Max is uniformly best everywhere else by avoiding dilution from the weaker track. Removing reference-alignment causes larger drops on multi-hop datasets where external guidance is more critical, while removing self-consistency affects general QA more where the model's own planning suffices.

| Model | RL | General QA | | | Multi-Hop QA | | | | Avg. | Internal |
|---|---|---|---|---|---|---|---|---|---|---|
| | | NQ | TriviaQA | PopQA | HotpotQA | 2Wiki | MuSiQue | Bamboogle | | AD-QA |
| Qwen2.5-3B-Inst. | GRPO | **53.0** | **74.5** | **47.9** | **36.2** | **36.6** | **13.3** | **28.8** | **41.5** | **79.5** |
| Qwen2.5-3B-Inst. | PPO | 51.6 | 73.1 | 46.8 | 34.8 | 35.4 | 12.6 | 27.6 | 40.3 | 78.1 |
| Llama-3.2-3B-Inst. | GRPO | 50.2 | 71.8 | 45.4 | 33.9 | 34.1 | 11.8 | 26.0 | 39.0 | 75.8 |
| Llama-3.2-3B-Inst. | PPO | 48.9 | 70.2 | 44.2 | 32.8 | 33.0 | 10.9 | 24.8 | 37.8 | 74.2 |

Table 9: Detailed accuracy (%) across different base models and RL algorithms on all datasets. Qwen2.5 consistently outperforms Llama-3.2, and GRPO achieves slightly higher accuracy than PPO across all datasets.

| Model | Method | General QA | | | Multi-Hop QA | | | | Avg. | Internal AD-QA |
|---|---|---|---|---|---|---|---|---|---|---|
| | | NQ | TriviaQA | PopQA | HotpotQA | 2Wiki | MuSiQue | Bamboogle | | |
| *7B* | w/o Soft Scoring | 55.4 | 77.5 | 46.0 | 39.2 | 36.5 | 18.5 | 40.8 | 44.8 | 77.4 |
| | w/ Soft Scoring | **56.6** | **78.6** | **47.5** | **42.9** | **39.8** | **21.8** | **44.0** | **47.3** | **86.2** |
| | $\Delta$ | *+1.2* | *+1.1* | *+1.5* | *+3.7* | *+3.3* | *+3.3* | *+3.2* | *+2.5* | *+8.8* |
| *3B* | w/o Soft Scoring | 51.8 | 73.2 | 46.5 | 32.6 | 33.1 | 10.4 | 24.5 | 38.9 | 68.5 |
| | w/ Soft Scoring | **53.0** | **74.5** | **47.9** | **36.2** | **36.6** | **13.3** | **28.8** | **41.5** | **79.5** |
| | $\Delta$ | *+1.2* | *+1.3* | *+1.4* | *+3.6* | *+3.5* | *+2.9* | *+4.3* | *+2.6* | *+11.0* |

Table 10: Effect of soft outcome scoring (ACC %). Multi-hop QA datasets benefit more from soft scoring (+3.0–3.7%) compared to general QA datasets (+1.1–1.5%), while the internal AD-QA dataset shows the largest improvement (+8.8–11.0%), confirming that complex enterprise queries benefit most from partial credit signals.

| $\lambda_p$ | General QA | | | Multi-Hop QA | | | | Avg. | Internal AD-QA |
|---|---|---|---|---|---|---|---|---|---|
| | NQ | TriviaQA | PopQA | HotpotQA | 2Wiki | MuSiQue | Bamboogle | | |
| 0.2 | 55.5 | 77.8 | 46.8 | 41.8 | 38.5 | 20.8 | **44.5** | 46.5 | 83.5 |
| **0.3** | **56.6** | **78.6** | **47.5** | **42.9** | **39.8** | **21.8** | 44.0 | **47.3** | **86.2** |
| 0.4 | 55.8 | 78.0 | 47.2 | 42.2 | 39.2 | 21.5 | 43.8 | 46.8 | 85.0 |

Table 11: Effect of path reward weight $\lambda_p$ on performance (ACC %, Qwen2.5-7B). The optimal value is $\lambda_p = 0.3$, which achieves the best average performance. While $\lambda_p = 0.2$ shows slight advantage on Bamboogle, $\lambda_p = 0.3$ provides the best overall balance.

| $\lambda_a$ | General QA | | | Multi-Hop QA | | | | Avg. | Internal AD-QA |
|---|---|---|---|---|---|---|---|---|---|
| | NQ | TriviaQA | PopQA | HotpotQA | 2Wiki | MuSiQue | Bamboogle | | |
| **0.6** | **56.6** | **78.6** | **47.5** | **42.9** | **39.8** | 21.8 | 44.0 | **47.3** | **86.2** |
| 0.8 | 56.0 | 78.2 | 47.0 | 42.5 | 39.2 | **22.2** | **44.8** | 47.1 | 85.5 |
| 1.0 | 54.8 | 76.8 | 45.8 | 41.2 | 38.0 | 20.8 | 43.2 | 45.8 | 83.2 |

Table 12: Effect of outcome accuracy weight $\lambda_a$ on performance (ACC %, Qwen2.5-7B). The optimal value is $\lambda_a = 0.6$, which achieves the best average performance. While $\lambda_a = 0.8$ shows slight advantages on MuSiQue and Bamboogle, $\lambda_a = 0.6$ provides better overall results.

| LLM Evaluator | General QA | | | Multi-Hop QA | | | | Avg. | Internal AD-QA |
|---|---|---|---|---|---|---|---|---|---|
| | NQ | TriviaQA | PopQA | HotpotQA | 2Wiki | MuSiQue | Bamboogle | | |
| *Frontier proprietary models* | | | | | | | | | |
| DeepSeek-V3.2 | **57.8** | 78.5 | **48.2** | 43.2 | **41.0** | 22.0 | 43.5 | **47.8** | **87.0** |
| HY 2.0-Instruct (Ours) | 56.6 | **78.6** | 47.5 | **42.9** | 39.8 | **21.8** | **44.0** | 47.3 | 86.2 |
| GLM-5 | 56.0 | 77.8 | 47.2 | 42.8 | 40.5 | 20.5 | 43.2 | 46.9 | 85.8 |
| GPT-4o | 55.2 | 78.0 | 46.5 | 42.0 | 39.5 | 21.5 | **44.0** | 46.7 | 83.8 |
| *Open-source models* | | | | | | | | | |
| Qwen3-32B-Instruct | 55.5 | 77.5 | 46.2 | 41.8 | 38.8 | 20.8 | 42.5 | 46.2 | 84.2 |
| Qwen3-8B-Instruct | 52.8 | 74.2 | 43.5 | 39.2 | 35.8 | 17.8 | 39.8 | 43.3 | 79.5 |
| Search-R1 (no evaluator) | 42.9 | 62.3 | 42.7 | 38.6 | 34.6 | 16.2 | 40.0 | 39.6 | 65.6 |

Table 13: Detailed accuracy (%) across LLM evaluators (Qwen2.5-7B + GRPO). Frontier proprietary models (DeepSeek-V3.2, HY 2.0, GLM-5, GPT-4o) form a tight top tier with complementary strengths across datasets (spread $\leq$1.1 Avg.). Qwen3-32B shows modest degradation ($-1.1$ Avg.), while Qwen3-8B exhibits larger drops on multi-hop tasks where step coverage evaluation is more challenging.