EXPLORER-R1: EFFICIENT ITERATIVE RETRIEVAL VIA STEPWISE EVIDENCE-STATE REWARDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Iterative retrieval allows large language models to collect supporting evidence in stages, offering capabilities that a one-shot search cannot. Prior iterative retrieval techniques have leveraged Reinforcement Learning (RL) to enhance multi-hop RAG retrieval for reasoning over questions and producing answers. However, many methods rely on end-to-end optimization where the reward is based only on the final answer, overlooking rewards tied to intermediate reasoning steps and making efficient training challenging. Moreover, optimizing solely for the final answer limits the model's ability to serve as a standalone retrieval component that other models can reuse. To address these challenges, we introduce *Explorer-R1* (ExR1), an efficient RL framework that recasts multi-hop retrieval as a sequential decision-making problem to train a plug-and-play multi-hop RAG retriever. Specifically, ExR1 breaks the original end-to-end pipeline into a series of state transitions. By monitoring transitions in the evidence state, ExR1 gathers data to improve later model decisions. In the RL phase, ExR1 uses a dense, state-level reward to guide evidence updates, search behavior, and termination decisions, keeping the evidence state compact. Consequently, ExR1 separates retrieval from answering, allowing it to operate as a standalone, robust retrieval component that can adapt smoothly and flexibly to downstream collaborative settings with other large models. Extensive experiments on well-known benchmarks show that ExR1 outperforms existing iterative RAG baselines in recall, converges faster during training, and can effectively collaborate with different answer models.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in complex reasoning, yet their incomplete parametric knowledge often leads to ungrounded claims on knowledge-intensive tasks (Wei et al., 2022; Achiam et al., 2023; Ji et al., 2023). Retrieval-Augmented Generation (RAG) mitigates this issue by grounding generation in external documents (Lewis et al., 2020). However, real-world multi-hop question answering (QA) usually cannot be completed with a single retrieval step (Ji et al., 2026). For example, a question asking *What one star of The Newcomers is known for* requires the model to first retrieve the cast list of the movie, then retrieve information about the cast members, and finally identify the representative work of a specific cast member. The required evidence emerges progressively and cannot be achieved through single-step retrieval. Consequently, iterative retrieval systems that interleave evidence access with model inference have become important for open-domain question answering, enterprise knowledge bases, and academic assistants (Cheng et al., 2025; Li et al., 2026; Zheng et al., 2025). These applications require an effective iterative retrieval system to select evidence accurately, propose new queries, and decide when to terminate.

Recent iterative retrieval systems have evolved from heuristic prompting toward learned search policies (Shao et al., 2023; Jiang et al., 2023). Reinforcement learning (RL) approaches (Ouyang et al., 2022) such as Search-R1 (Jin et al., 2025) and RAG-R1 (Tan et al., 2026) internalize the search policy into model parameters. These prior works use the final answer score as the reward signal and optimize the full pipeline from question input to answer output. As shown in Figure 1 (a), the traditional end-to-end framework lets the model run the entire retrieval chain and then updates all model actions along the chain based on the reward of the final answer. However, this end-to-end, one-shot optimization has three clear defects. ❶ Credit assignment is ambiguous (Li & Li, 2025; Setlur et al., 2025). A complete rollout contains multiple actions, yet only the final answer provides a

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

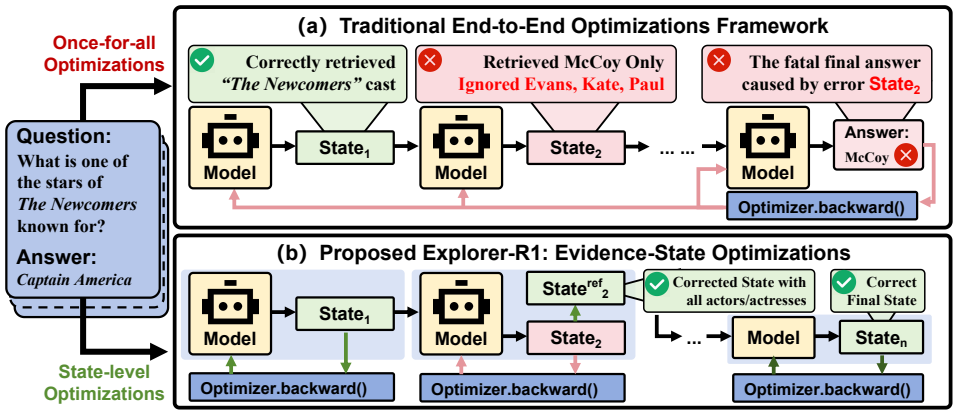


Figure 1: Motivation for learning evidence-state transitions.

delayed reward. Consequently, the model cannot easily determine which retrieval step is decisive. In Figure 1(a), the model misses key actors at the second step, which causes subsequent states to fail. Because the penalty applies to the entire generation chain, the model cannot attribute this failure to the second step and thus cannot precisely correct the erroneous state. ❷ Training efficiency is low (Wu et al., 2026; Xie et al., 2026). Because the reward depends on the final answer, the model must complete the full multi-turn generation and retrieval process before updating. As a result, optimization frequency is limited, and substantial computation is wasted on unrecoverable paths. ❸ Retrieval is coupled with answer generation (Fang et al., 2026). In end-to-end training, the retrieval policy is optimized alongside the model's answer generation, so the retrieval behavior is shaped to serve the answer generation rather than produce transferable evidence states. Consequently, the learned retrieval policy cannot be directly reused by other answer models (Shi et al., 2023).

To address these issues, this paper proposes Explorer-R1 (ExR1), a reinforcement learning framework that frames multi-hop RAG retrieval as a sequential decision process over evidence-state transitions. Unlike end-to-end answer optimization, ExR1 decomposes the iterative retrieval process into stepwise evidence-state updates. Within this framework, a lightweight policy model, which we call the explorer, takes the current evidence state as input and outputs structured actions that transform it into the next evidence state. Figure 1 (b) illustrates the training procedure of ExR1. After each evidence state, an optimizer update is triggered, so the explorer is optimized without waiting for the final answer. When the explorer produces an incorrect retrieval at a step, the system immediately penalizes that action and replaces the next input with the reference evidence state $State_t^{ref}$. Furthermore, the reward is computed over evidence states rather than the final answer, and the training output is a compact evidence state. This procedure provides immediate feedback for each step, prevents error accumulation, and keeps the learned retrieval policy independent of answer generation. Specifically, ExR1 realizes this stepwise evidence-state optimization through three coordinated designs. ❶ For ambiguous credit assignment, ExR1 gives a stepwise evidence reward for each evidence state that evaluates evidence updates, search behavior, and termination decisions, providing immediate feedback for every action. ❷ For low training efficiency, ExR1 optimizes the model at the end of each evidence state, so the model updates without completing the full multi-turn generation and retrieval process. ❸ For the coupling between retrieval and answer generation, ExR1 optimizes the explorer to refine the evidence state rather than generate the final answer, making the learned policy an independent, plug-and-play retrieval module.

- We reformulate multi-hop retrieval as a sequential decision process and standardize its formulation by decomposing the end-to-end pipeline into evidence-state transitions.
- We propose ExR1, an RL framework that trains a retrieval model through stepwise evidence-state optimization, and design a stepwise reward to evaluate model actions, including evidence updates, search behavior, and termination decisions.
- We decouple retrieval from answer generation, creating a plug-and-play, robust module that adapts to downstream collaboration with other LLMs.
- We conduct extensive experiments on HotpotQA, MuSiQue, and 2WikiMultihopQA, showing that ExR1 outperforms existing iterative RAG baselines in recall, converges faster during training, and collaborates effectively with different answer models.

2 Related Work

Iterative Retrieval. Retrieval-augmented language models combine parametric generation with evidence retrieved from an external collection (Lewis et al., 2020; Guu et al., 2020; Karpukhin et al., 2020). Sparse retrievers such as BM25 remain competitive and transparent, while dense retrievers improve semantic matching at the cost of additional training and indexing choices (Robertson & Zaragoza, 2009; Karpukhin et al., 2020). This paradigm has proven effective in mitigating hallucinations on knowledge-intensive tasks, but single-step retrieval is often inadequate for complex multi-hop questions, where the required evidence emerges progressively. To address this, iterative retrieval methods interleave search with model inference. IRCoT (Trivedi et al., 2023) first couples chain-of-thought reasoning with retrieval, using partial reasoning steps to formulate subsequent queries. Search-o1 (Li et al., 2025) performs an additional refinement step after each retrieval to improve evidence quality. These methods generalize well across multi-hop tasks, but they substantially increase token cost per question. This efficiency concern has motivated researchers to internalize the retrieval policy into model parameters rather than relying on prompt engineering.

Reinforcement Learning for Retrieval. Reinforcement learning has been widely adopted for LLM tuning since RLHF (Ouyang et al., 2022). Recent work has turned to RL to internalize search policies into model parameters. LeReT (Hsu et al., 2024) is an early attempt to optimize a single-turn retrieval strategy with RL. Search-R1 (Jin et al., 2025) then jointly optimizes reasoning and retrieval, demonstrating that policy optimization can empower LLMs to autonomously execute multi-turn, multi-query retrieval behaviors. SPARKLE (Fang et al., 2026) trains a lightweight proxy model to control retrieval. It computes its reward at the trajectory level from final-context recall and answer F1, then assigns it back to individual agent decisions via Monte Carlo estimation, and it relies on an additional model to extract knowledge triples from retrieved documents. More recently, RAG-R1 (Tan et al., 2026) provides a stable engineering recipe for single- and multi-retrieval training. Across these approaches, supervision remains at the final answer or trajectory level. However, these approaches provide supervision through end-to-end post-credit assignment, without directly rewarding the quality of each evidence-state transition. To the best of our knowledge, ExR1 is the first to optimize iterative retrieval by rewarding evidence-state quality at each transition, yielding a single policy to perform evidence selection, query formulation, and termination in one action.

3 Methodology

Preliminary. Given a multi-hop question Q, an external corpus $\mathcal{D}$, and a retriever $\mathrm{Search}(\cdot)$, ExR1 represents the information available at transition t as $\mathrm{State}_t = (E_t, D_t)$. Here, E_t is the compact evidence accumulated from previous retrieval transitions, while D_t contains the raw documents returned in the current transition. Rather than appending all retrieved text, E_t retains selected evidence anchors with their query provenance and records unproductive retrieval attempts. This compact representation therefore preserves the information required for subsequent exploration without carrying the full reasoning trajectory. Starting from $(Q, \mathrm{State}_{t-1})$, the student produces a structured action $a_t^{\mathrm{stu}} = (E_t^{\mathrm{stu}}, q_t^{\mathrm{stu}})$, where E_t^{stu} is the updated evidence and q_t^{stu} is a set of follow-up queries. The retriever returns $D_t^{\mathrm{stu}} = \mathrm{Search}(q_t^{\mathrm{stu}})$, forming the next state $\mathrm{State}_t^{\mathrm{stu}} = (E_t^{\mathrm{stu}}, D_t^{\mathrm{stu}})$. An empty query set terminates exploration. The finalized evidence E_T is then supplied to the reasoner $\mathcal{M}_{\mathrm{ans}}$, which produces $A = \mathcal{M}_{\mathrm{ans}}(Q, E_T)$. This explicit evidence-state formalization exposes the recurring transition structure within a multi-hop QA episode, thereby enabling stepwise rewarding.

Overview of ExR1. ExR1 optimizes the student policy over the designed evidence-state transition structure, so that the resulting transition-level policy composes the model's overall policy for multi-hop QA. As illustrated in Figure 2, the framework consists of three steps. Data generation converts each QA episode into a trajectory composed of multiple evidence-state transitions, producing the transition dataset. Supervised fine-tuning (SFT) then aligns the base model with the task format and provides a roughly capable student initialization. Finally, transition-level RL further refines the student policy under the designed retrieval-grounded reward. In actual deployment, the trained student becomes an explorer that updates the evidence state and provides the finalized evidence E_T to a separately chosen reasoner. This decoupling makes the explorer plug-and-play, allowing it to be paired with different reasoners. The following subsections detail these steps and provide a theoretical analysis demonstrating the feasibility of the proposed design.

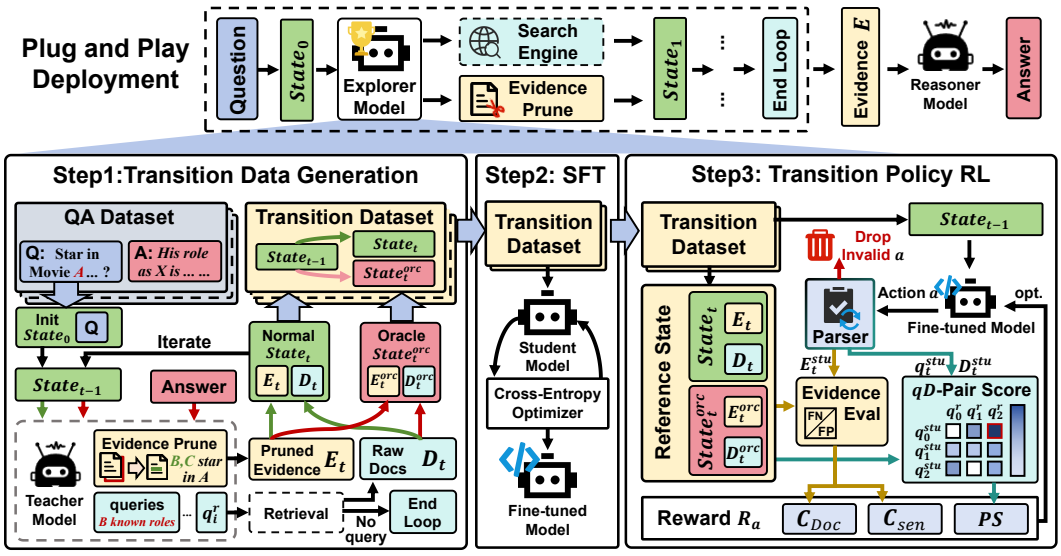


Figure 2: **Framework and workflow of our ExR1 approach.**

3.1 Transition Data Generation (Step1)

Transition Data Generation is the offline component that converts each QA episode into transition-level supervision. To expose the recurring evidence-state structure while avoiding hallucinated teacher decisions, we query the same teacher twice at every visited normal state State_{t-1}. As illustrated in Figure 2, the first inference provides the primary supervision. The teacher selects evidence anchors and proposes queries under the same information condition available to the student, yielding the normal next state State_t. The second inference provides an optional oracle scoring reference. With the ground-truth answer additionally provided, the teacher produces a one-step oracle target $\mathrm{State}_t^{\mathrm{orc}}$ under the same condition. The oracle target is not rolled forward, and only State_t continues the normal trajectory. Each normal trajectory is then decomposed into transition-level records $(Q, \mathrm{State}_{t-1}, \mathrm{State}_t, \mathrm{State}_t^{\mathrm{orc}})$, where State_t serves as the main target and $\mathrm{State}_t^{\mathrm{orc}}$ serves as the corrective target. The resulting transition dataset $\mathcal{D}_{\mathrm{tr}}$ is used for SFT and transition-level RL.

3.2 Supervised Fine-Tuning (Step 2)

SFT is the first training stage, initializing the student policy on the transition dataset. As illustrated in Step 2 of Figure 2, the transition dataset is fed into the student model, and a cross-entropy optimizer updates the student to reproduce teacher actions. This stage aligns the student with the structured action format required by the downstream parser and provides a capable initialization for evidence selection, query generation, and termination decision. We optimize the student by minimizing:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\mathrm{tr}}}\left[\sum_{u=1}^{L} \log \pi_\theta\big(a_{t,u} \mid a_{t,<u}, Q, \mathrm{State}_{t-1}\big)\right], \tag{1}$$

where $a_t = (a_{t,1}, \ldots, a_{t,L})$ is the serialized teacher action associated with $(Q, \mathrm{State}_{t-1})$, and $\mathcal{D}_{\mathrm{tr}}$ denotes the transition dataset. However, SFT only reproduces teacher actions and does not distinguish alternative student actions by their retrieval consequences. To address this limitation, the next stage, transition-level RL, optimizes the student at the transition level.

3.3 Transition Policy RL (Step 3)

Transition Policy RL optimizes the student from the SFT initialization using grouped transition rewards. As illustrated in Step 3 of Figure 2, the fine-tuned student generates G actions from each evidence state, the parser filters invalid structured actions, and the reward R_a is computed for each valid action. We standardize these rewards within the group to obtain advantages $\hat{A}_i$. We maximize

the clipped Group Relative Policy Optimization (GRPO)(Shao et al., 2024) objective as:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\rho_i \hat{A}_i, \mathrm{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\hat{A}_i\right) - \beta_{\mathrm{KL}}\mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})\right)\right], \quad (2)$$

where $\rho_i = \pi_\theta(a_i \mid Q, \mathrm{State}_{t-1})/\pi_{\mathrm{old}}(a_i \mid Q, \mathrm{State}_{t-1})$. Group-relative advantages compare candidate transitions from the same evidence state, while the Kullback–Leibler (KL) term limits deviation from the supervised reference. Each transition therefore provides direct optimization feedback without requiring a full exploration-and-answer rollout.

3.4 Transition-Level Reward Design

The transition policy RL stage above requires computing the reward R_a for each valid student action. To provide a computable supervision signal at the transition level, we design R_a with four components, namely an evidence cost, a query-matching score, a gated transition score, and an optional oracle reference. As illustrated in Step 3 of Figure 2, after a valid action a_t^{stu} is established, it turns into an evidence update E_t^{stu}, a query set q_t^{stu}, and retrieved documents D_t^{stu}. The reward is computed against reference targets $r \in \{\mathrm{nor}, \mathrm{orc}\}$, where nor corresponds to the normal next state State_t and orc corresponds to the oracle target $\mathrm{State}_t^{\mathrm{orc}}$.

Evidence Cost. The evidence cost evaluates the evidence-update part E_t^{stu} against the reference evidence E_t^r. As shown in the Evidence Eval block of Figure 2, the parser output is compared with the reference state to compute normalized false-negative (FN) and false-positive (FP) rates. The asymmetric document cost is defined as:

$$C_{\mathrm{doc}}^r = \frac{w_{\mathrm{FN}}\mathrm{FN}_r + w_{\mathrm{FP}}\mathrm{FP}_r}{w_{\mathrm{FN}} + w_{\mathrm{FP}}}, \quad (3)$$

where $w_{\mathrm{FN}} > w_{\mathrm{FP}} > 0$ are tunable weights controlling the preference between evidence coverage and compactness. The sentence-level cost C_{sen}^r is computed analogously over the retained anchors using the same weighted form, and its relative weight is denoted by $\lambda > 0$. Both costs lie in $[0, 1]$.

Query Matching. To encourage the student to generate retrieval-equivalent queries that may differ in wording but retrieve similar evidence, we design the query-matching reward based on retrieval outcomes rather than raw query strings. Since direct query-string comparison drives the student to imitate surface text rather than learn a retrieval strategy, we compare the prefixes of retrieved rankings. As shown in the query-Document pair (qD-Pair) score block of Figure 2, for the i-th student query and the j-th reference query, we compute:

$$S_{ij}^{qD,r} = \frac{1}{k}\sum_{h=1}^{k}\frac{\left|D_{i,1:h}^{\mathrm{stu}} \cap D_{j,1:h}^{r}\right|}{h}, \quad (4)$$

to form the $S_{ij}^{qD,r}$ matrix. Maximum-weight one-to-one qD matching yields the final pair score (PS):

$$PS_r = \frac{1}{\max(m,n)}\max_{\mathcal{M}}\sum_{(i,j)\in\mathcal{M}} S_{ij}^{qD,r}, \quad (5)$$

where m and n are the numbers of student and reference queries, and $\mathcal{M}$ ranges over one-to-one matchings. This construction rewards retrieval-equivalent queries, assigns zero contribution to unmatched queries, and prevents one student query from explaining multiple reference retrievals.

Gated Transition Reward. We use evidence preservation as a gate before rewarding the next retrieval. Since a better retrieval direction cannot compensate for an evidence update that loses reference information, any evidence mismatch yields a negative reward. The transition score is:

$$R_r(a_t^{\mathrm{stu}}) = \begin{cases} -\left(C_{\mathrm{doc}}^r + \lambda C_{\mathrm{sen}}^r\right), & C_{\mathrm{doc}}^r > 0, \\ \alpha\left(1 - C_{\mathrm{sen}}^r\right) + \beta PS_r + \gamma\,\mathrm{Stop}_r, & C_{\mathrm{doc}}^r = 0, \end{cases} \qquad \alpha + \beta + \gamma = 1. \quad (6)$$

Here $\mathrm{Stop}_r \in \{0, 1\}$ indicates whether the student termination decision matches reference r. Since an empty query set terminates exploration, this term rewards stopping when the reference also stops

and penalizes premature or delayed termination. The weights α, β, and γ balance anchor preservation, retrieval-direction alignment, and termination correctness.

Oracle Reference and Final Reward. Since normal teacher decisions may contain hallucinations, we do not penalize a student action that follows the oracle target rather than the normal target. Therefore, we boost the positive oracle reward as:

$$R_{\text{orc}}^{+} = R_{\text{orc}} + w_{\text{orc}} \max(R_{\text{orc}}, 0), \tag{7}$$

where $w_{\text{orc}} \geq 0$ is a hyperparameter controlling the strength of the boost. The final reward is then:

$$R_a = \max\left(R_{\text{nor}}, R_{\text{orc}}^{+}\right). \tag{8}$$

3.5 Theoretical Justification

End-to-end optimization conditions the policy on a delayed final-answer reward, obscuring contribution assignment and limiting training efficiency. This limitation motivates optimizing individual evidence-state transitions rather than the full retrieval-and-answer pipeline. We now analyze whether such transition-level optimization can provably improve the complete multi-hop QA process. For a fixed reasoner, let $U(Q, E_T) \in [0, 1]$ be the terminal QA utility and $J(\pi) = \mathbb{E}_\pi[U(Q, E_T)]$ the end-to-end performance of explorer π over at most T transitions. Let $\pi^\star$ be a reference policy. For transition reward $R_t(s, a)$, define the local regret $\ell_t(s, a) = R_t^\star(s) - R_t(s, a)$, where $R_t^\star(s) = \sup_a R_t(s, a)$. We use three conditions. *(i) Approximate state sufficiency.* Replacing the history by $s = \text{State}_{t-1}$ changes reference continuation values by at most ε_s. *(ii) Reward calibration.* An action's downstream disadvantage is upper-bounded by $\kappa\ell_t(s, a) + \varepsilon_r$. *(iii) State coverage.* Let μ_t and d_t^π be the transition-data and rollout distributions. Then $d_t^\pi(s) \leq C\mu_t(s)$ for some $C \geq 1$. *Theorem 1 (Transition-to-task guarantee).* Let $\overline{\mathcal{L}}_{\text{tr}}(\pi) = T^{-1} \sum_{t=1}^{T} \mathbb{E}_{s\sim\mu_t, a\sim\pi(\cdot|s)}[\ell_t(s, a)]$ be the average transition regret. Under the conditions above, we have:

$$J(\pi^\star) - J(\pi) \leq \kappa C T\, \overline{\mathcal{L}}_{\text{tr}}(\pi) + T\big(\varepsilon_r + 2\varepsilon_s\big). \tag{9}$$

Thus, reducing transition regret reduces an explicit upper bound on the complete QA performance gap. Evidence-state construction targets ε_s, retrieval-grounded rewarding targets ε_r, and SFT initialization with KL-regularized GRPO help control the coverage factor C. Reward shaping is designed to make local reward differences reflect the larger downstream risk of losing necessary evidence. The proof, extensions, and scope of the conditions are provided in Appendix C. This analysis justifies the overall design: the evidence-state formulation, transition data generation, and retrieval-grounded reward jointly reduce the upper bound on the complete QA performance gap.

4 Experiments

We conducted experiments to address four Research Questions (**RQs**). **RQ1:** Does ExR1 outperform existing iterative RAG methods? **RQ2:** Is ExR1 efficient under a small budget? **RQ3:** How does each component work? **RQ4:** How effective is ExR1 as an independent retrieval module?

4.1 Experimental Setup

Datasets and Retrieval. We sampled 3,000 HotpotQA training questions and used Qwen3.5-Plus (Qwen Team, 2026) to generate a transition dataset containing 5,600 training, 700 validation, and 700 test transition records. Baselines followed their original training data configurations. Evaluation used fixed 1,000-question subsets of HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultihopQA (Ho et al., 2020). We tested every method with frozen BM25 (Robertson & Zaragoza, 2009) and DPR (Karpukhin et al., 2020) indexes. Five documents are returned per query.

Models and Baselines. The prompt-based group contains Direct Inference, CoT, Direct RAG, IRCoT, and Search-o1. These methods use the original Qwen2.5-1.5B-Instruct (Qwen Team, 2024) model without task-specific SFT or RL. The learned group contains Search-R1, RAG-R1-SQ, RAG-R1-MQ, and ExR1, all initialized from Qwen2.5-1.5B-Instruct and following their respective SFT and RL logic. ExR1 used Qwen2.5-1.5B-Instruct as the answering model.

Evaluation Metrics. We assessed answer correctness by measuring ExactMatch (EM) and token F1. We measured retrieval quality using Recall@k, which measures whether the gold evidence appears

Table 1: **Main performance comparisons on three multi-hop QA benchmarks.**

RAG Engine	Method	HotpotQA				MuSiQue				2WikiMultihopQA			
		R@3	R@10	EM	F1	R@3	R@10	EM	F1	R@3	R@10	EM	F1
None	Direct	n/a	n/a	13.0	18.8	n/a	n/a	1.7	7.6	n/a	n/a	21.2	25.6
	CoT	n/a	n/a	10.8	18.6	n/a	n/a	3.2	9.6	n/a	n/a	16.7	21.4
BM25	Direct RAG	62.1	70.9	31.1	40.9	35.1	40.6	7.5	13.7	59.8	63.9	24.5	30.2
	IRCoT	62.1	80.4	31.2	42.1	35.1	49.0	9.5	16.1	59.9	79.4	30.3	37.0
	Search-o1	57.4	72.7	31.1	40.1	34.9	42.9	7.6	14.3	54.2	65.2	24.7	30.1
	Search-R1 (1.5B)	18.0	19.1	3.3	7.8	10.2	11.1	1.0	3.2	17.0	18.5	5.5	8.1
	Search-R1 (3B)	55.9	78.6	43.3	53.9	34.3	52.8	15.3	22.5	48.1	80.2	**40.5**	**47.1**
	RAG-R1-SQ	51.9	72.3	40.4	49.8	32.5	49.1	13.9	21.2	47.5	80.3	35.2	43.9
	RAG-R1-MQ	45.5	72.6	37.5	47.5	29.8	46.9	10.2	18.0	46.0	77.2	36.1	44.3
	ExR1	**79.6**	79.6	**48.8**	**61.1**	**52.2**	52.3	**18.5**	**26.0**	**81.4**	82.7	39.0	46.6
	ExR1 (w/o pruning)	51.9	**89.1**	34.3	43.9	38.9	**66.8**	8.8	14.1	47.5	**84.5**	28.7	35.8
DPR	Direct RAG	65.5	74.0	29.0	39.2	42.7	49.3	10.9	18.4	56.0	61.8	23.7	28.8
	IRCoT	65.6	80.4	31.1	41.6	42.6	56.4	14.8	23.0	56.0	73.8	28.8	35.6
	Search-o1	58.7	72.3	29.2	39.8	39.1	48.9	10.1	17.8	51.9	64.3	24.2	29.4
	Search-R1 (1.5B)	20.6	21.4	5.5	9.5	11.1	12.2	0.4	2.9	17.7	19.8	6.0	8.4
	Search-R1 (3B)	56.4	76.7	39.9	49.3	37.5	56.7	18.9	25.7	48.1	78.0	**40.8**	**47.2**
	RAG-R1-SQ	51.8	70.4	34.9	46.1	36.3	51.9	15.4	23.3	47.6	79.5	33.3	40.5
	RAG-R1-MQ	46.7	73.3	38.0	47.9	33.2	51.4	10.9	18.3	44.6	76.6	34.4	41.3
	ExR1	**78.7**	78.7	**47.0**	**58.3**	**54.6**	54.7	**19.5**	**27.9**	**80.3**	81.2	39.4	46.5
	ExR1 (w/o pruning)	52.0	**85.7**	32.3	42.0	39.9	**68.1**	14.5	21.8	46.2	**82.6**	26.0	33.3

in the top-k retrieved documents, and Docs, which denotes the number of documents in the final evidence set. We also measured efficiency using Tok., the tokenizer-level token count.

Training Configurations. All reinforcement learning methods shared the same runtime protocol: each run used two RTX 4090 GPUs, was capped at 36 wall-clock hours, and saved the nearest completed optimizer state every six hours. PPO-based methods used batch size 8, actor learning rate 1×10^{-6}, critic learning rate 1×10^{-5}, and rollout group size 1. Our GRPO-based explorer used batch size 8, a learning rate of 5×10^{-6}, and a rollout group size of 8. We train the explorer with LoRA (Hu et al., 2022). Appendix D provides detailed hyperparameters for each method.

4.2 RQ1: Overall Effectiveness, Retrieval Behavior, and Inference Cost

Main Effectiveness. We compared all methods under a unified 1.5B backbone. For ExR1, we reported two variants. ExR1 (w/o pruning) includes every unique document retrieved along the trajectory, while ExR1 passes only the final retained evidence set to the answerer. We adopted Search-R1's released 3B checkpoint as an upper bound on its attainable performance. Table 1 shows that ExR1 achieves the strongest overall answer quality across all evaluated same-size learned systems. On HotpotQA with BM25, it reaches 61.1 F1, outperforming every same-size baseline by a clear margin. This advantage remains under DPR, indicating that the gains are not tied to a single retrieval backend. On MuSiQue, where evidence chains are longer and more compositional, ExR1 retains a consistent lead over both prompted and trajectory-trained baselines. On 2WikiMultihopQA, it achieves the best recall and answer quality among all learned systems. Notably, ExR1 with only 1.5B parameters achieves performance comparable to the released 3B Search-R1 checkpoint. The ablation ExR1 (w/o pruning) further isolates the effect of pruning, which we analyze in detail below.

Evidence Flow and Inference Cost. To understand where the gains came from, we collect evidence-flow and efficiency metrics on MuSiQue with BM25. **Final R.** and **Final P.** measure gold-document recall and evidence precision in the evidence set that is ultimately used for answering. **L. Stop** is the fraction of questions terminated by the policy's explicit synthesis or answer action. Table 2 reports the results. The unpruned variant reveals a clear limitation of pure exploration. Although it achieves a high gold-document recall of 73.9, its evidence precision is only 17.5, as it retains 13.02 documents per question. The recall performance does not translate into answer quality, yielding only 38.9 R@3 and 14.1 F1. In contrast, ExR1 prunes the candidate pool to 1.88 documents per question, which raises evidence precision to 74.4. Although this pruning reduces gold-document recall to 56.6, the

Table 2: **Evidence flow and efficiency comparisons on MuSiQue with BM25.**

Method	Final R.	Final P.	Docs	Queries	R@3	F1	Tok. In	Invalid	L. Stop
IRCoT	54.0	20.3	8.01	4.39	35.1	16.1	5484	0%	n/a
Search-o1	48.4	20.7	7.24	7.08	34.9	14.3	7267	0%	n/a
Search-R1 (1.5B)	12.5	10.1	1.59	0.67	10.2	3.2	628	26.2%	55.7%
RAG-R1-MQ	55.1	20.7	8.13	3.64	29.8	18.0	3300	2.6%	91.2%
ExR1 (w/o pruning)	**73.9**	17.5	13.02	3.97	38.9	14.1	6777	3.5%	81.2%
ExR1	56.6	**74.4**	**1.88**	3.97	**52.2**	**26.0**	5828	3.5%	81.2%

Table 3: **Evaluation of training trajectories on HotpotQA with BM25.**

Method	Origin		6h		12h		24h		36h		AUC	Rollout Tok.
	F1	R@5	F1	R@5	F1	R@5	F1	R@5	F1	R@5		
Search-R1	10.99	26.85	10.16	25.75	8.50	25.85	10.20	24.45	7.96	23.25	9.46	0.76M
RAG-R1-SQ	49.30	68.20	48.58	67.80	47.49	67.80	49.79	69.05	48.97	67.40	48.84	0.40M
RAG-R1-MQ	47.15	63.20	46.01	61.50	47.24	62.30	47.21	62.55	47.49	63.45	47.06	0.40M
ExR1 (full.)	56.67	77.35	57.11	77.25	56.20	76.50	58.07	77.40	58.32	77.40	57.37	0.41M
ExR1	56.67	77.35	61.14	79.60	60.03	79.85	60.49	80.20	59.51	80.40	60.00	1.95M

final evidence set becomes substantially more compact and precise, improving R@3 to 52.2 and F1 to 26.0. This trade-off shows that high recall alone is insufficient in noisy contexts, and that a compact, high-precision evidence set is more answer-friendly. The efficiency metrics further support this picture. Pruning reduces input tokens from 6777 to 5828, lowering inference cost. The low invalid-action rate and the relatively high termination rate indicate that the explorer has learned when to stop and synthesize an answer. Taken together, these results confirm that the gains come from combining strong exploration with a compact, high-precision evidence set.

4.3 RQ2: Convergence and Training Efficiency

To assess whether stepwise evidence-state rewarding improves training efficiency, we collect training trajectories for all methods and evaluate them at aligned checkpoints on HotpotQA with BM25. Since Search-R1 has no SFT stage, we treat its actual 6-hour checkpoint as Origin and shift later displayed points by six hours. Other methods use their post-SFT checkpoints as Origin, and we recompute AUC on the displayed relative-time axis. Under this unified setting, Search-R1 collapses during training, with AUC only 9.46 and persistently low F1 and R@5, because its rollouts increasingly answer directly instead of executing retrieval, and search behavior is not stably learned. We attribute this to the absence of SFT initialization, since the end-to-end reward is too sparse to teach basic search behavior and the policy exploits direct answering as a shortcut. Our method outperforms all baselines under the same training budget, considering Rollout Tok. and GPU time. The full-parameter variant quickly identifies the optimization direction and, relative to its own initialization, shows an overall upward trend in F1 while steadily strengthening evidence collection, whereas the other methods fluctuate without consistent gains. The LoRA variant converges much faster and reaches its maximum F1 of 61.14 within six hours, after which F1 remains stable while R@5 continues to rise to 80.40, indicating that later training mainly improves retrieval coverage rather than answer quality. These results reveal that SFT is necessary and efficient for learning a stable basic policy, while evidence-state rewarding enables stable optimization.

4.4 RQ3: Component Contributions

To understand how each reward component contributes to evidence preservation and retrieval-equivalent query matching, we conducted an ablation study on MuSiQue with BM25. Each trainable variant received the same six-hour LoRA-GRPO budget. Ever R. and Final R. are gold-document recall before and after pruning. Final P. is the precision of the final evidence set. Early Stop is the fraction of early stops with missing gold evidence. Zero-Var. is the fraction of groups with no within-group variance. Table 4 reports the results. The three components serve distinct roles. Removing the evidence-preservation gate raises early stopping to 61.5%, which indicates that the

Table 4: **Ablation study of reward components on MuSiQue with BM25.**

Variant	EM	F1	Ever R.	Final R.	Final P.	Docs	Calls	Early Stop	Zero-Var.
ExR1	18.5	26.0	73.9	56.6	74.4	1.88	3.97	53.6%	23.0%
w/o evidence gate	18.1	25.5	71.5	55.7	73.1	1.90	3.51	61.5%	20.8%
w/o qD-Pair score	18.4	25.6	73.8	56.2	74.0	1.83	4.22	47.0%	62.4%
w/o oracle reference	17.4	24.8	73.6	54.3	73.2	1.79	4.71	50.2%	22.3%

Table 5: **Collaboration performance with different answer models (BM25).**

Method	HotpotQA						MuSiQue						2WikiMultihopQA					
	Q2.5-7B		Q3.5-Plus		4o-mini		Q2.5-7B		Q3.5-Plus		4o-mini		Q2.5-7B		Q3.5-Plus		4o-mini	
	F1	Tok.	F1	Tok.	F1	Tok.	F1	Tok.	F1	Tok.	F1	Tok.	F1	Tok.	F1	Tok.	F1	Tok.
IRCoT	55.1	1262	56.2	1278	60.0	1190	19.1	1371	20.3	1387	25.5	1295	42.6	1371	46.3	1373	47.8	1282
Search-o1	49.4	1121	50.9	1136	54.4	1055	14.1	1238	14.1	1250	21.6	1163	32.8	1127	32.8	1129	35.5	1052
Search-R1	19.0	432	13.4	432	15.9	398	4.1	454	3.6	452	4.7	415	13.0	404	8.6	405	11.1	373
RAG-R1-MQ	52.4	1003	55.5	1018	56.1	944	19.5	1329	22.9	1343	27.4	1244	46.1	1059	48.9	1065	50.5	993
ExR1	**69.9**	348	69.0	457	**69.4**	413	**26.8**	506	27.3	521	31.1	467	**53.9**	560	54.0	567	**58.1**	521
ExR1 (w/o pruning)	66.9	1379	**69.9**	1467	68.0	1371	25.6	2041	**30.9**	1960	**33.9**	1822	49.7	1529	**56.6**	1359	56.0	1268

gate mainly prevents the policy from terminating before the evidence state is complete. Removing the qD-Pair Score leaves answer quality nearly unchanged but raises the zero-variance rate to 62.4%, showing that its primary contribution is dense credit assignment rather than direct answer improvement. Removing the oracle reference reduces final recall and increases the mean number of Calls, which indicates that the alternative continuation improves evidence retention. These results match the motivation of stepwise evidence-state rewarding. Three components each shape a different aspect of the per-transition signal. Together, they give the policy fine-grained guidance at every evidence-state construction step, which the end-to-end final-answer reward cannot provide.

4.5 RQ4: Plug-and-Play Collaboration with Different Answer Models

To test how effectively ExR1 serves as an independent retrieval module, we used Qwen2.5-7B-Instruct(Q2.5-7B), Qwen3.5-Plus(Q3.5-Plus), and GPT-4o-mini(4o-mini) to answer on the evidence retrieved by each method. Table 5 reports the results. ExR1 delivers the strongest overall collaboration quality across all three datasets and all three answer models. On HotpotQA with Qwen2.5-7B, it reaches 69.9 F1, far above every baseline. This advantage holds under Qwen3.5-Plus and GPT-4o-mini, and extends to MuSiQue and 2WikiMultihopQA. The gains confirm that the learned retrieval strategy transfers across model families and scales. The token cost further supports the plug-and-play claim. ExR1 uses substantially fewer reasoner-side tokens than IRCoT, Search-o1, and RAG-R1-MQ in every setting. Its compact final evidence set lets smaller answer models focus on relevant context and achieve strong performance with low token consumption. The w/o pruning variant retains all retrieved documents and therefore offers higher recall. This higher recall benefits stronger answer models, which can filter noise more effectively. The two variants thus serve different regimes, with ExR1 favoring efficiency and ExR1 (w/o pruning) favoring recall when the answer model is sufficiently strong. Taken together, these results confirm that ExR1 decouples retrieval from answering. The same frozen retrieval trajectory remains effective when consumed by different answer models, and the learned evidence state is both compact and transferable.

5 Conclusion

We introduced Explorer-R1(ExR1), an RL framework that recasts multi-hop retrieval as a sequential decision process to train a plug-and-play retrieval module. Specifically, we decompose the end-to-end pipeline into evidence-state transitions and optimize each transition with a dense evidence-state reward that guides evidence updates, search behavior, and termination decisions. Extensive experiments show that ExR1 outperforms existing iterative RAG baselines in recall, converges faster during training, and collaborates effectively with different answer models. Future work includes applying this method to other fine-tuning tasks and developing more efficient data generation schemes.

AI USAGE STATEMENT

We used generative AI tools to assist with transition dataset generation and manuscript preparation, including language polishing and grammar correction. Because the teacher model used for data generation may produce hallucinated content, we designed targeted mechanisms to mitigate these errors. All AI-assisted outputs were reviewed, verified, and edited by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available multi-hop QA datasets and does not involve human subjects, personal data, or privacy-sensitive information. We adhere to the ICLR Code of Ethics and confirm that no ethical violations are associated with this study.

REPRODUCIBILITY STATEMENT

We provide detailed training and evaluation configurations in Appendix D, the theoretical assumptions and complete proof in Appendix C, and the transition data generation protocol in Appendix A. The code and data generation scripts will be released as supplementary materials.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ili Akkaya, Florencia L. Aleman, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Mingyue Cheng, Yucong Luo, Jie Ouyang, Qi Liu, Huijie Liu, Li Li, Shuo Yu, Bohou Zhang, Jiawei Cao, Jie Ma, Daoyu Wang, and Enhong Chen. A survey on knowledge-oriented retrieval-augmented generation. *ACM Transactions on Information Systems*, 2025. URL https://arxiv.org/abs/2503.10677.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. Sparkle: A structured and plug-and-play agentic retrieval policy for adaptive rag models. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pp. 38699–38719, 2026.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

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

Sheryl Hsu, Omar Khattab, Chelsea Finn, and Archit Sharma. Grounding by trying: Llms with reinforcement learning-enhanced retrieval. arXiv preprint arXiv:2410.23214, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Yuelyu Ji, Zhuochun Li, Rui Meng, and Daqing He. Retrieval–reasoning processes for multi-hop question answering: A four-axis design framework and empirical trends. arXiv preprint arXiv:2601.00536, 2026.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *In Proceedings of the Conference on Language Modeling (COLM)*, 2025.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K"uttler, Mike Lewis, Wen-tau Yih, Tim Rockt"aschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, volume 2025, pp. 14708–14726, 2025.

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 *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Xingyu Li, Rongguang Wang, Yuying Wang, Mengqing Guo, Chenyang Li, Tao Sheng, Sujith Ravi, and Dan Roth. Par2-rag: Planned active retrieval and reasoning for multi-hop question answering. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Industry Track*, volume 6, pp. 1727–1737, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-industry.118.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Qwen Team. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024.

Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, 2026.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, volume 2025, pp. 60808–60838, 2025.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. arXiv preprint arXiv:2305.15294, 2023.

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. arXiv preprint arXiv:2402.03300, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Sch"arli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *In Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Zhiwen Tan, Jiaming Huang, Qintong Wu, Hongxuan Zhang, Chenyi Zhuang, and Jinjie Gu. Rag-r1: Incentivizing the search and reasoning capabilities of llms through multi-query parallelism. In *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

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, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

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

Yutao Xie, Nathaniel Thomas, Nick Hansen, Yang Fu, Li Li, and Xiaolong Wang. Tips: Turn-level information-potential reward shaping for search-augmented llms. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, volume 2026, pp. 156549–156584, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Liwen Zheng, Chaozhuo Li, Litian Zhang, Haoran Jia, Senzhang Wang, Zheng Liu, and Xi Zhang. Mrr-fv: Unlocking complex fact verification with multi-hop retrieval and reasoning. In *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025. doi: 10.1609/aaai.v39i24.34802.

A Teacher Data Generation Details

We generate the transition data with Qwen3.5-Plus accessed through an OpenAI-compatible API. Decoding is deterministic ($T = 0$), with a maximum of 1,400 generated tokens per request and up to three retries. The environment uses a local BM25 index over the 10,000-question HotpotQA training pool, returns the top five documents per query, permits at most four queries per transition and six retrieval calls per episode, and limits each selected document to at most two source sentences. The teacher produces two references for every transition: *Normal*, which receives the question and current retrieval state, and *Oracle*, which additionally receives the gold answer and gold supporting documents. The student does not see the answer-only teacher prompt; it retains only the structured state-transition action.

For the raw completed episodes, the teacher's final answers obtain 79.12% EM and 86.91% token F1 (2,277 exact matches out of 2,878 episodes). Normal actions issue 8,493 retrieval queries and select 5,280 document items; Oracle actions issue 7,581 queries and select 5,522 document items. All 7,005 Normal and Oracle actions pass the structural validator before release cleaning. The remaining filtering is therefore a data-release operation rather than a model-validation failure.

We measure evidence recall and pruning on the Normal trajectory. For each episode, "ever retrieved" is the union of all documents returned by any Normal query, while "final" is the deduplicated evidence state the environment passes on after the teacher stops. Macro-averaged ever-retrieved gold-document recall is 91.16%, and the final retained state preserves 81.97% recall. Final evidence precision is 93.35%, whereas the broad ever-retrieved pool has only 18.69% precision. The trajectory retrieves an average of 10.72 unique documents and retains 1.77 documents in the final state. Computing the final-to-ever ratio separately for each episode and then macro-averaging gives 17.75% retention, or the exactly complementary 82.25% reduction. For reference, the ratio of the two corpus-level means is 16.54%; it differs because a mean of per-episode ratios is not the ratio of means. Thus, the generated data exhibits the intended recall-first pattern: broad search coverage followed by a substantially smaller, high-precision evidence state.

Table 6: Counts in the teacher-generation pipeline. The five-transition difference between the raw and exported counts reflects malformed transitions removed by the release exporter; four additional contaminated Oracle references appear as null in the exported training data.

Generation quantity	**Value**
Candidate questions in local pool	10,000
Questions selected for generation	3,000
Completed and audited episodes	2,878
Failed episodes	122
Raw transitions in completed episodes	7,005
Exported transitions	7,000
Normal references in export	7,000
Oracle references in export	6,996
Oracle references removed during cleaning	4
Export split (train/validation/test)	5,600 / 700 / 700

Table 7: Provider-reported token usage for the 2,878 completed episodes. Input tokens include the serialized question, state, and retrieved documents sent to the teacher; output tokens include the structured action or final answer. These are generation-time API tokens and are separate from the RL rollout-token accounting reported in RQ2.

Call type	**Calls**	**Input tok.**	**Output tok.**	**Total tok.**
Normal transition calls	7,005	18,007,554	6,473,250	24,480,804
Oracle transition calls	7,005	17,974,020	9,044,384	27,018,404
Final answer calls	2,878	810,390	3,490,697	4,301,087
All API calls	**16,888**	**36,791,964**	**19,008,331**	**55,800,295**

B Transition Protocol, Data Generation Prompts, and Reward Cases

This appendix first specifies the exact transition interface used by the explorer and the prompts used to construct its supervision. It then traces two groups sampled during GRPO training. Unlike an evaluation-time case study, each case fixes one training state and compares three actions sampled from the same group. The reported rewards and group-relative advantages are taken from the training log rather than recomputed from an evaluation trajectory.

B.1 Transition Protocol

Explorer-R1 learns a local knowledge-state transition rather than an entire search-and-answer episode. At transition t, its visible input is defined as:

$$x_t = (Q, \mathcal{S}_{t-1}, \mathcal{D}_t), \tag{10}$$

where Q is the root question, $\mathcal{S}_{t-1}$ is the compact history of completed query nodes, and $\mathcal{D}_t$ contains only the documents returned in the current retrieval round. A FOUND node stores verbatim sentence anchors selected in an earlier round, whereas a NO-NEW-INFO node records a query from which no document was retained. Historical raw documents are not repeatedly inserted into the prompt.

The policy emits one structured action as:

$$a_t = (\texttt{thought}_t, \texttt{use}_t, \texttt{search}_t). \tag{11}$$

The `thought` field is a short local analysis and is not copied into the next state. The `use` field points to one or two visible sentences in a current document; the environment copies those sentences verbatim into a new FOUND node. The `search` field contains at most four novel entity-rich queries. An empty `search` list terminates retrieval and hands the retained evidence to a separate answer reasoner.

Table 8: The four transition types induced by the compact Explorer-R1 protocol. These labels are derived deterministically from the visible input and the normal reference action; they are not additional tokens predicted by the policy.

Transition	Visible condition	Expected action	Environment outcome
`initial`	Both `state` and `results` are empty.	Set `use=[]` and issue one to four initial entity-rich queries.	Execute the queries and expose their top-k results in the next transition.
`use_continue`	Current results contain useful evidence and further information is required.	Point to the minimal useful sentence anchors and issue novel continuation queries.	Create FOUND or NO-NEW-INFO nodes and continue retrieval.
`no_info_continue`	The current results add no selectable evidence, but a productive untried branch remains.	Set `use=[]` and backtrack to entities or relations exposed by historical anchors.	Record the failed current queries as NO-NEW-INFO and execute the new branch.
`terminal`	The retained anchors are sufficient, or no productive untried branch remains.	Retain any useful current evidence and set `search=[]`.	Stop retrieval and pass the retained documents to the answer model.

Table 9: The structured action predicted during SFT, GRPO, and inference.

Explorer Action Schema

```
{
"thought": "brief analysis of productive and failed branches",
"use": [{"d": 7, "sentences": [0, 2]}],
"search": ["next entity-rich retrieval query"]
}
```

Validity constraints. The output must contain exactly the three keys above. A document pointer may refer only to the current `results`; its sentence list must contain one or two visible sentence identifiers. Document identifiers cannot be repeated inside `use`. Queries are normalized by lowercasing and removing punctuation, and a query is invalid if it repeats an item in either `state`, current `results`, or the same action. The complete action is rejected if any pointer, field, or query violates the protocol.

B.2 PROMPTS FOR TRANSITION DATA GENERATION

We generate the transition corpus before SFT or GRPO. Neither training stage calls a teacher model or an LLM judge. For every visible state, the generator requests a *normal* action from an unguided transition teacher and an *oracle* action from a teacher that additionally receives gold-only guidance. Both teachers must obey the same sentence-pointer protocol that is later imposed on the student.

Student prompt. The student uses the same system instruction in Table B.2, but without the teacher few-shot messages and without any answer or gold field. Both SFT and GRPO serialize the student interaction as:

$$[\texttt{system: protocol}, \texttt{user: } x_t, \texttt{assistant: } a_t]. \tag{12}$$

SFT applies supervision only to the assistant action. GRPO supplies the same system and user messages, samples G assistant actions, validates each action, executes its `search` queries with the frozen retriever, and computes the transition reward.

B.3 VALIDATION AND TRAINING SERIALIZATION

After teacher generation, we normalize document identifiers to a transition-local contiguous range. The exported corpus removes question identifiers, gold answers, teacher API traces, and oracle thoughts. A compact training row has the following form in Table B.3.

Every exported row is checked for the exact top-level schema; a nonempty root question; list-valued state and results; contiguous visible document IDs; existing document and sentence pointers; and

Table 10: The exact system instruction used by the normal transition teacher.

normal-Teacher System Prompt

```
You are Explorer-R1, a knowledge-state transition policy.

Never answer the root question.  A separate reasoner answers once retrieval stops.

Input JSON:
- question:  the root question.
- state:  completed query nodes.  FOUND nodes contain verbatim sentence anchors copied
by the environment; NO_NEW_INFO nodes record queries that added no selected evidence.
Historical raw documents are never shown again.
- results:  only the current round's documents, grouped by the exact query q.  Each
document has a temporary id and numbered sentences.

Output exactly one JSON object:
{
"thought":  "brief analysis of productive and failed branches",
"use":  [{"d":  7, "sentences":  [0, 2]}],
"search":  ["next entity-rich retrieval query"]
}

Rules:
1.  Select only documents whose pointed sentences add useful evidence.  For every current
query with no selected document, the environment automatically creates a NO_NEW_INFO node;
do not report negative outcomes yourself.
2.  Use may reference only document IDs in current results.  Each sentences array MUST
contain exactly one or two visible sentence ids from that document, never three or more.
Select the smallest set of sentences that preserves useful entities and relations.
3.  A document is useful only if its selected sentences support its source query or a
still-productive branch.  Ignore distractors even when their words resemble the query.
4.  Never invent or paraphrase an anchor.  You only point to sentences; the environment
copies their verbatim text into state.
5.  Expand all productive branches in parallel.  When one branch adds no new information
but a historical FOUND anchor exposes candidate entities or works, backtrack to those
anchors and search the candidates instead of repeatedly focusing on the failed branch.
6.  Generate at most four novel entity-rich keyword queries.  Before emitting search,
compare every candidate against every q in both state and results after lowercasing and
removing punctuation.  Delete every repeated query; one repeated query invalidates the
entire action.
7.  On the first call, results is empty:  use=[], then search contains initial queries.
8.  search=[] hands the retained documents to the final reasoner.  Stop when the retained
evidence is sufficient, or when current results add no useful evidence and the state
exposes no untried entity or relationship angle.  Rephrasing an exhausted intent is not
a novel action.
9.  Thought is allowed but is not copied into state.  Return JSON only, without markdown.
```

Table 11: Runtime envelope used to request a normal reference action.

normal-Teacher Runtime Message

```
{
"question":  "{root question}",
"state":  [{
"q":  "{completed query}",
"status":  "FOUND|NO_NEW_INFO",
"anchors":  [{"title":"{title}","sentences":["{verbatim sentence}"]}]
}],
"results":  [{
"q":  "{current query}",
"docs":  [
{"id":0,"title":"{title}",
"sentences":[{"id":0,"text":"{sentence}"}]}
]
}]
}
```

The generator prepends three fixed few-shot trajectories illustrating a two-hop bridge, backtracking after an unproductive query, and termination from an indirect but sufficient constraint. The final user message is the visible runtime object above.

Table 12: The normal-teacher system instruction in Table B.2 is followed by this oracle-only addendum. The lower block is the oracle runtime envelope.

Oracle-Teacher Prompt Addendum and Runtime Message

```
You are producing an Oracle reference. Teacher-only Gold guidance
is appended to the visible input. Use all Gold information to choose
a shorter, more productive action than unguided exploration. Gold
may directly identify the best document, entity, branch, or next
query. The action must still obey the visible sentence-pointer
schema: use can only select current documents, while search may use
Gold guidance.
```

```
{
"visible_input": {
"question": "{root question}",
"state": [...],
"results": [...]
},
"teacher_only_gold": {
"answer": "{gold answer}",
"gold_titles": ["{title}", ...],
"supporting_facts": [["{title}", {sentence id}], ...]
}
}
```

The oracle message sequence uses parallel bridge, backtracking, and constraint demonstrations. Gold information is available only to this offline teacher; it is absent from the student input used by SFT, GRPO, and evaluation.

Table 13: Compact interface between offline data generation and model training. SFT supervises `normal.thought/use/search`; GRPO compares the sampled action against `normal.use/retrieved` and, when present, `oracle.use/retrieved`.

Compact Transition Record

```
{
"input": {
"question": "...", "state": [...], "results": [...]
},
"normal": {
"thought": "...",
"use": [{"d":0,"sentences":[0]}],
"search": ["next query"],
"retrieved": [[12,35,71,90,103]]
},
"oracle": {
"use": [{"d":0,"sentences":[0]}],
"retrieved": [[12,35,71,90,103]]
}
}
```

one retrieval list per reference query. An oracle reference that fails deterministic structural checks is stored as `null`, in which case training falls back to the normal reference. In the retrieval field, `retrieved=[]` denotes stopping, whereas `retrieved=[[]]` denotes an executed query with no returned document.

B.4 Training-Time Reward Cases

We now inspect two actual GRPO groups from the training run. Each group contains eight actions sampled from the same transition prompt. The tables show three representative actions, while the reported group mean, standard deviation, and advantages are computed over all eight samples. Green and red mark reward-relevant correct and incorrect decisions, respectively.

Case 1: evidence preservation gates search and stopping rewards. Table B.4 uses a `use_continue` state. The normal reference retains Documents 0 and 2 and requests an additional query about Louise of Hesse-Kassel. The oracle reference retains the same documents but stops because the gold-guided teacher recognizes that the visible Louise of Sweden evidence already supports the intended answer. The highest-reward rollout follows this shorter oracle continuation. The first error stops after dropping Document 2; although its stop behavior exactly matches the oracle, the document mismatch closes the gate. The second error preserves both documents but issues a query whose retrieved ranking matches neither reference continuation.

Table 14: Aligned reward computation for three sampled actions. Error A shows that a perfect oracle stop cannot compensate for dropping a required document.

Highest-Reward Rollout	**Error A: Dropped Evidence**	**Error B: Wrong Continuation**
Question: Louise Bille-Brahe was a Danish courtier to the wife of what King?		**Training group:** sweep 1, position 1344, $G = 8$.
Step 0: Shared visible state. The student receives two top-5 result groups, not only the two documents later retained. For query *Louise Bille-Brahe*: **Doc 0, Louise Bille-Brahe** states that she served Louise of Hesse-Kassel and later Louise of Sweden; Doc 1 is the similar Danish courtier Ida Marie Bille; **Doc 2, Louise of Sweden** identifies her as the spouse of King Frederick VIII; Docs 3–4 concern actress Louise Brooks and astronomer Paul Wittich. For query *Louise Bille-Brahe courtier*: Doc 5 repeats the Louise Bille-Brahe article, Doc 6 repeats Ida Marie Bille, and Docs 7–9 concern Arthur Collins, Enevold Brandt, and Ingeborg Christiane Rosenorn. Thus, Docs 0 and 2 are selected from ten visible result records; the other eight are duplicate or irrelevant candidates.		
Normal reference: retain Docs 0 and 2; search *Louise of Hesse-Kassel husband King Denmark*.		
Oracle reference: retain Docs 0 and 2; stop.		
Step 1: Sampled Action **Thought:** Docs 0 and 2 establish the courtier–queen relation and that Louise of Sweden was married to **Frederick VIII**; the chain is sufficient. **Use:** **Doc 0, sent. 0**; **Doc 2, sent. 0**. **Search:** **[]**.	**Step 1: Sampled Action** **Thought:** Doc 0 identifies Louise of Sweden and the answer is already available. **Use:** **Doc 0, sent. 0**; **Doc 2 omitted**. **Search:** [].	**Step 1: Sampled Action** **Thought:** The husband must still be retrieved. **Use:** **Doc 0, sent. 0**; **Doc 2, sent. 0**. **Search:** **Frederick VIII King of Denmark**.
Step 2: Evidence Comparison **Doc FN / FP:** 0 / 0. **Sentence FN / FP:** 0 / 0. C_{DOC} / C_{sen}**:** 0 / 0. **Gate:** **OPEN**.	**Step 2: Evidence Comparison** **Doc FN / FP:** 0.5 / 0. **Sentence FN / FP:** 0.5 / 0. C_{DOC} / C_{sen}**:** 0.40 / 0.40. **Gate:** **CLOSED**.	**Step 2: Evidence Comparison** **Doc FN / FP:** 0 / 0. **Sentence FN / FP:** 0 / 0. C_{DOC} / C_{sen}**:** 0 / 0. **Gate:** **OPEN**.
Step 3: Reference Scoring **Normal:** $PS = 0$, stop mismatch; $R_{\text{nor}} = 0.15$. **Oracle:** $PS = 1$, stop match; $R_{\text{orc}} = 1.00$. **Oracle shaping:** $1.00 + 0.2(1.00) = 1.20$. **Final reward:** $R_a = 1.20$ (oracle).	**Step 3: Reference Scoring** **Normal:** Gate closed; $R_{\text{nor}} = -(0.40 + 0.25 \times 0.40) = -0.50$. **Oracle:** The stop matches and $PS = 1$, but the same evidence gate is closed. **Final reward:** $R_a = -0.50$.	**Step 3: Reference Scoring** **Retrieved top-5:** Louise of Sweden; Frederick II; Frederick I; Princess Eleonore; King John. **Normal:** $PS = 0$, continue match; $R_{\text{nor}} = 0.15 + 0 + 0.15 = 0.30$. **Oracle:** $PS = 0$, stop mismatch; shaped score 0.18. **Final reward:** $R_a = 0.30$ (normal).
Step 4: Group-Relative Signal **Group statistics:** $\mu = 0.7931$, $\sigma = 0.5921$. **Advantage:** $\hat{A} = +0.687$. **Update:** Reinforce the oracle-aligned stop.	**Step 4: Group-Relative Signal** **Group statistics:** $\mu = 0.7931$, $\sigma = 0.5921$. **Advantage:** $\hat{A} = -2.184$. **Update:** Strongly suppress evidence-dropping termination.	**Step 4: Group-Relative Signal** **Group statistics:** $\mu = 0.7931$, $\sigma = 0.5921$. **Advantage:** $\hat{A} = -0.833$. **Update:** Suppress the unsupported continuation.

Case 2: qD-Pair rewards retrieval-equivalent queries rather than surface forms. Table B.4 is an `initial` transition, so all valid actions correctly use no current document and continue retrieval. Their reward difference therefore comes almost entirely from the ranked documents produced by

their queries. The best rollout recovers both reference retrieval groups. Error A emits only one of the two required searches and receives zero credit for the unmatched group. Error B uses broad producer- and actor-oriented queries; although the strings contain relevant names, their retrieved rankings drift away from the film-specific references.

Table 15: Aligned qD-Pair scoring for three sampled actions. The score depends on retrieved rankings and maximum-weight one-to-one matching.

Highest-Reward Rollout	**Error A: Missing Query Branch**	**Error B: Retrieval Drift**
Question: In what year was the American musical drama film, produced by Tim Rice and Andrew Lloyd Webber and featuring Antonio Banderas, released? **Training group:** sweep 1, position 93, $G = 8$.		
Step 0: Shared visible state. This is the initial transition: `state=[]`, `results=[]`, and all three actions correctly set `use=[]`. No document can yet be selected.		
Normal reference operation. Set `use=[]`; issue two unguided searches joining Tim Rice, Andrew Lloyd Webber, Antonio Banderas, and the film relation; retain both resulting top-5 ranking groups as the normal qD target; continue retrieval.		
Oracle reference operation. Set `use=[]`; use teacher-only gold guidance to issue two Evita-focused searches, one targeting the film identity/release year and one targeting the creator–cast relation; retain their two top-5 ranking groups as the oracle qD target; continue retrieval. The compact GRPO record stores these retrieved rankings rather than requiring the student to copy either teacher's query wording.		
Step 1: Sampled Queries **Query 1: Tim Rice Andrew Lloyd Webber Antonio Banderas film.** **Query 2: Antonio Banderas Tim Rice Andrew Lloyd Webber.**	**Step 1: Sampled Queries** **Query 1: Antonio Banderas Tim Rice Andrew Lloyd Webber film.** **Query 2: Missing.**	**Step 1: Sampled Queries** **Query 1: Tim Rice Andrew Lloyd Webber musical film.** **Query 2: Antonio Banderas production company.**
Step 2: Retrieved Rankings **Ranking 1: Evita accolades; Evita soundtrack; Evita film;** Wizard of Oz; Jesus Christ Superstar. **Ranking 2: Evita accolades; Evita soundtrack; Evita film;** Jesus Christ Superstar; Evita album.	**Step 2: Retrieved Rankings** **Ranking 1: Evita accolades; Evita soundtrack; Evita film;** Wizard of Oz; Jesus Christ Superstar. **Ranking 2: Unmatched reference group.**	**Step 2: Retrieved Rankings** **Ranking 1:** Wizard of Oz; Joseph; Jesus Christ Superstar; Evita accolades; Evita album. **Ranking 2:** Antonio Banderas; Havana Quartet; Play It to the Bone; Summer Rain; Haywire.
Step 3: qD-Pair Matching PS_{nor}**: 0.935.** PS_{orc}: 0.577. **Interpretation:** Both student groups are matched one-to-one to retrieval-equivalent normal groups.	**Step 3: qD-Pair Matching** PS_{nor}**: 0.500.** PS_{orc}: 0.285. **Interpretation:** The one recovered group scores well, but the unmatched second reference contributes zero.	**Step 3: qD-Pair Matching** PS_{nor}**: 0.110.** PS_{orc}**: 0.150.** **Interpretation:** Relevant words do not rescue rankings dominated by unrelated films and actor pages.
Step 4: Reference Scoring **Normal:** $0.15 + 0.70(0.935) + 0.15 = 0.9545$. **Oracle shaped:** $[0.15 + 0.70(0.577) + 0.15] \times 1.2 = 0.8444$. **Final reward:** $R_a = 0.9545$ (normal).	**Step 4: Reference Scoring** **Normal:** $0.15 + 0.70(0.500) + 0.15 = 0.6500$. **Oracle shaped:** $[0.15 + 0.70(0.285) + 0.15] \times 1.2 = 0.5994$. **Final reward:** $R_a = 0.6500$ (normal).	**Step 4: Reference Scoring** **Normal:** $0.15 + 0.70(0.110) + 0.15 = 0.3770$. **Oracle shaped:** $[0.15 + 0.70(0.150) + 0.15] \times 1.2 = 0.4860$. **Final reward:** $R_a = 0.4860$ (oracle).
Step 5: Group-Relative Signal **Group statistics:** $\mu = 0.6733$, $\sigma = 0.1396$. **Advantage:** $\hat{A} = +2.015$. **Update:** Strongly reinforce both productive query branches.	**Step 5: Group-Relative Signal** **Group statistics:** $\mu = 0.6733$, $\sigma = 0.1396$. **Advantage:** $\hat{A} = -0.167$. **Update:** Suppress incomplete branch coverage.	**Step 5: Group-Relative Signal** **Group statistics:** $\mu = 0.6733$, $\sigma = 0.1396$. **Advantage:** $\hat{A} = -1.342$. **Update:** Strongly suppress retrieval drift.

B.5 Worked Example of qD-Pair Scoring

The qD-Pair score compares the documents induced by two queries, rather than their surface strings. For a transparent arithmetic example (separate from the logged aggregate value $PS_{\text{nor}} = 0.935$), consider one student query and one reference query with the following top-5 rankings, using titles from Case 2:

$$D^{\text{stu}} = [A, B, C, D, E],$$
$$D^{\text{ref}} = [A, B, C, E, D],$$

where $A =$ *Evita accolades*, $B =$ *Evita soundtrack*, $C =$ *Evita film*, $D =$ *The Wizard of Oz*, and $E =$ *Jesus Christ Superstar*. The two queries may be lexically different, but their rankings contain the same five documents with a small order disagreement. At each prefix depth, their overlap precision is:

$$\left[\frac{1}{1}, \frac{2}{2}, \frac{3}{3}, \frac{3}{4}, \frac{5}{5}\right] = [1, 1, 1, 0.75, 1]. \tag{13}$$

Therefore, with $k = 5$, the pair score is:

$$S_{11}^{qD} = \frac{1}{5}(1 + 1 + 1 + 0.75 + 1) = 0.95. \tag{14}$$

The fourth prefix receives only $3/4$ because the student has introduced D while the reference has introduced E; the score recovers at depth five when both documents are present. Hence qD is sensitive to early ranking quality without requiring identical query text.

For multiple queries, all pair scores form a bipartite matching matrix. For example, suppose two student queries and two reference queries produce as:

$$\mathbf{S}^{qD} = \begin{bmatrix} 0.95 & 0.30 \\ 0.25 & 0.90 \end{bmatrix}. \tag{15}$$

Maximum-weight one-to-one matching chooses $(1, 1)$ and $(2, 2)$, giving:

$$PS = \frac{0.95 + 0.90}{\max(2, 2)} = 0.925. \tag{16}$$

If the student emits only the first query, the best matched weight remains 0.95, but normalization by the larger query-set size yields $PS = 0.95/2 = 0.475$. The absent branch therefore contributes zero instead of allowing one strong query to explain both reference searches. This is the mechanism behind the lower score of Error A in Table B.4; Error B is penalized for a different reason, namely low prefix overlap for every possible match.

Together, the two cases expose the intended reward ordering. Evidence selection is evaluated first: once a reference document is lost, a favorable query or stopping decision cannot compensate for the loss. When document selection is exact, qD-Pair distinguishes complete, incomplete, and drifting search directions through their retrieval consequences. Finally, the optional oracle reference permits a shorter valid continuation without forcing the student to imitate a single teacher trajectory.

C Reward Details and Theoretical Analysis

C.1 Details of the qD-Pair Score

For student queries $\{q_i^{\text{stu}}\}_{i=1}^m$ and reference queries $\{q_j^r\}_{j=1}^n$, Eq. 4 produces an $m \times n$ matrix. A feasible matching $\mathcal{M}$ contains pairs (i, j) with no repeated student or reference index. Equation 5 selects the maximum-weight matching and divides its weight by $\max(m, n)$. Hence, unmatched queries contribute zero, and a single student query cannot receive credit for several reference searches.

The score operates on stable document identities rather than query strings, rewarding retrieval equivalence under different query formulations. If both sides terminate, their empty query sets represent consistent termination; if only one side terminates, the unmatched queries contribute zero.

C.2 Formal Setup

Let h_t denote the complete interaction history before action a_t, and let $s_t = \phi(h_t)$ be its compact evidence-state representation. An episode contains at most T transitions and ends with evidence E_T. For a fixed answer model, the terminal utility $U(Q, E_T) \in [0, 1]$ may represent answer accuracy or another bounded downstream QA metric. The end-to-end objective is:

$$J(\pi) = \mathbb{E}_\pi[U(Q, E_T)]. \tag{17}$$

For reference policy $\pi^\star$, define the history-dependent continuation values as:

$$V_t^\star(h_t) = \mathbb{E}[U(Q, E_T) \mid h_t, \pi^\star], \tag{18}$$

$$Q_t^\star(h_t, a) = \mathbb{E}[U(Q, E_T) \mid h_t, a, \pi^\star]. \tag{19}$$

We use $\bar{V}_t^\star(s)$ and $\bar{Q}_t^\star(s, a)$ for their counterparts defined on the compact state.

For local reward $R_t(s, a)$, let

$$R_t^\star(s) = \sup_a R_t(s, a), \qquad \ell_t(s, a) = R_t^\star(s) - R_t(s, a) \geq 0. \tag{20}$$

The quantity ℓ_t measures how far an action falls below the best transition reward available at the same state. Maximizing the group-relative reward therefore reduces an empirical counterpart of this local regret.

C.3 Assumptions

C.3.1 Approximate State Sufficiency

For every reachable history h_t and action a, assume:

$$\left|V_t^\star(h_t) - \bar{V}_t^\star(\phi(h_t))\right| \le \varepsilon_s, \qquad \left|Q_t^\star(h_t, a) - \bar{Q}_t^\star(\phi(h_t), a)\right| \le \varepsilon_s. \tag{21}$$

This condition does not require lossless compression. It requires only that information omitted from the evidence state have a bounded effect on downstream decisions under the reference continuation.

C.3.2 Reward Calibration

For every compact state and action, assume:

$$\bar{V}_t^\star(s) - \bar{Q}_t^\star(s, a) \le \kappa \ell_t(s, a) + \varepsilon_r. \tag{22}$$

The left-hand side is the loss in final QA utility caused by taking a before returning to the reference policy. Thus, Eq. 22 is the formal connection between the retrieval-grounded transition reward and the downstream task. The scale κ permits the two quantities to use different units, while ε_r captures reward misspecification and imperfect reference supervision.

C.3.3 State-Distribution Coverage

Let d_t^π be the distribution of compact states encountered by π and μ_t the distribution represented by the transition dataset. Assume absolute continuity with a bounded density ratio as:

$$d_t^\pi(s) \le C\mu_t(s) \quad \text{for all } t, s, \tag{23}$$

where $C \ge 1$. When transition optimization uses states drawn from the current policy rollout, $C = 1$. In offline transition learning, C quantifies distribution shift between training and deployment.

C.4 Performance-Difference Decomposition

For completeness, we derive the finite-horizon identity used in the proof. By definition,

$$Q_t^\star(h_t, a_t) = \mathbb{E}\left[V_{t+1}^\star(h_{t+1}) \mid h_t, a_t\right]. \tag{24}$$

Taking expectations over trajectories generated by π gives:

$$\mathbb{E}_{h_t \sim d_t^\pi, a_t \sim \pi}\left[V_t^\star(h_t) - Q_t^\star(h_t, a_t)\right] \tag{25}$$

$$= \mathbb{E}_{h_t \sim d_t^\pi}[V_t^\star(h_t)] - \mathbb{E}_{h_{t+1} \sim d_{t+1}^\pi}[V_{t+1}^\star(h_{t+1})]. \tag{26}$$

Summing from $t = 1$ to T telescopes. Since the initial-state value is $J(\pi^\star)$ and the terminal value under trajectories from π is $J(\pi)$, we obtain:

$$J(\pi^\star) - J(\pi) = \sum_{t=1}^{T} \mathbb{E}_{h_t \sim d_t^\pi, a_t \sim \pi}\left[V_t^\star(h_t) - Q_t^\star(h_t, a_t)\right]. \tag{27}$$

This identity retains the sequential dependence of the task: earlier actions affect all later terms through the rollout distribution d_t^π.

C.5 Proof of Theorem 1

Let $s_t = \phi(h_t)$. By approximate state sufficiency,

$$V_t^\star(h_t) - Q_t^\star(h_t, a_t) \le \bar{V}_t^\star(s_t) - \bar{Q}_t^\star(s_t, a_t) + 2\varepsilon_s \tag{28}$$

$$\le \kappa \ell_t(s_t, a_t) + \varepsilon_r + 2\varepsilon_s, \tag{29}$$

where the second inequality applies reward calibration. Substituting Eq. 29 into Eq. 27 yields

$$J(\pi^\star) - J(\pi) \le \kappa \sum_{t=1}^{T} \mathbb{E}_{s \sim d_t^\pi, a \sim \pi}[\ell_t(s, a)] + T(\varepsilon_r + 2\varepsilon_s). \tag{30}$$

Because $\ell_t(s,a) \geq 0$, the coverage condition implies

$$\mathbb{E}_{s\sim d_t^\pi, a\sim\pi}[\ell_t(s,a)] \leq C\, \mathbb{E}_{s\sim\mu_t, a\sim\pi}[\ell_t(s,a)]. \tag{31}$$

Applying this inequality to Eq. 30, summing over t, and using the definition of $\overline{\mathcal{L}}_{\mathrm{tr}}(\pi)$ gives

$$J(\pi^\star) - J(\pi) \leq \kappa C T \overline{\mathcal{L}}_{\mathrm{tr}}(\pi) + T(\varepsilon_r + 2\varepsilon_s), \tag{32}$$

which proves Theorem 1. □

C.6 Extensions

If the compact state is sufficient ($\varepsilon_s = 0$), the reward is exactly calibrated ($\varepsilon_r = 0$), and optimization uses the rollout distribution ($C = 1$), then we have:

$$J(\pi^\star) - J(\pi) \leq \kappa T \overline{\mathcal{L}}_{\mathrm{tr}}(\pi). \tag{33}$$

Consequently, zero transition regret recovers the reference policy's end-to-end utility under these idealized conditions. More generally, the uniform constants in Theorem 1 are used only for readability. If calibration, compression, and coverage vary with the transition, the same proof gives:

$$J(\pi^\star) - J(\pi) \leq \sum_{t=1}^{T} \left[\kappa_t C_t \mathbb{E}_{\mu_t,\pi}[\ell_t] + \varepsilon_{r,t} + 2\varepsilon_{s,t}\right]. \tag{34}$$

This form highlights that errors in early bridge-finding transitions may have different scales from errors near termination. The density-ratio condition can be replaced by total-variation control. If $0 \leq \ell_t \leq L_{\max}$ and $\mathrm{TV}(d_t^\pi, \mu_t) \leq \delta_t$, then:

$$\mathbb{E}_{d_t^\pi,\pi}[\ell_t] \leq \mathbb{E}_{\mu_t,\pi}[\ell_t] + L_{\max}\delta_t. \tag{35}$$

Substitution into Eq. 30 produces an additive distribution-shift penalty $\kappa L_{\max} \sum_t \delta_t$ instead of the multiplicative factor C.

C.7 Connection to the Reward Design

The theorem does not assume that a particular reward construction is uniquely correct. Instead, the reward components are intended to make the calibration residual ε_r small. Omitting a necessary bridge may remove all high-value continuations, whereas retaining an additional document generally incurs a smaller context cost. Choosing $w_{\mathrm{FN}} > w_{\mathrm{FP}}$ reflects this directional asymmetry without prescribing a fixed ratio. The gate further prevents query quality from compensating for evidence loss, preserving the ordering between severe downstream disadvantage and local reward. Surface-different queries may induce the same document transition. The qD-Pair Score evaluates retrieved rankings, avoiding artificial regret for semantically different but operationally equivalent searches. Taking the better score under the Normal and Oracle targets expands the set of accepted transitions. This reduces the chance that a valid alternative path receives high local regret merely because it differs from one teacher realization. SFT initializes the policy on the transition distribution, while the KL term discourages abrupt deviation during GRPO. These mechanisms do not guarantee coverage, but they are consistent with controlling the effective distribution-shift factor C.

C.8 Scope and Limitations

Theorem 1 is a sufficient-condition result rather than an unconditional guarantee. Transition optimization may fail to improve final QA when the state removes decision-critical information (large ε_s), the local reward is poorly aligned with downstream utility (large ε_r), or the policy visits states absent from the transition data (large or unbounded C). Moreover, the guarantee is relative to $\pi^\star$. If $\pi^\star$ is a teacher policy, the theorem bounds imitation of its end-to-end performance; interpreting it as a bound to the globally optimal policy requires the stronger assumption that the reference itself is optimal.

D TRAINING AND EVALUATION DETAILS

Table 16: Detailed configuration for the Explorer-R1 experiments.

Category	Parameter	Value
Hardware	Training runtime	PyTorch/HuggingFace + PEFT, bf16
	GRPO devices	2 RTX 4090 GPUs, 36-hour wall-clock cap
Backbone	Explorer initialization	Qwen2.5-1.5B-Instruct
	SFT/GRPO adaptation	LoRA; rank 16, alpha 32, dropout 0.05
	LoRA targets	q, k, v, o and gate/up/down projections
Transition data	Teacher and source	Qwen3.5-Plus; HotpotQA transitions
	Split	5,600 / 700 / 700 transition records (train/validation/test)
Retrieval	Training retriever	Frozen SQLite BM25, top-$k = 5$
	Evaluation retrievers	Frozen BM25 and DPR; DPR uses all-MiniLM-L6-v2
	Retrieval multiplier	4 during training; evaluation uses fixed benchmark index
Protocol	Queries per action	At most 4
	Evidence anchors	At most 2 sentences/document, 700 characters/document
	Thought budget	1,200 characters
	Evaluation episode length	5 evidence-state transitions + 1 answer turn
SFT	Epochs / learning rate	2 / 2×10^{-4}
	Batch	Micro batch 1, gradient accumulation 8 (effective batch 8)
	Warmup / scheduler	0.05 / cosine
	Sequence length	6,144 tokens
GRPO	Group size G	8 actions/state
	Batch size	8
	Learning rate / optimizer	5×10^{-6} / AdamW (LoRA)
	Clip ϵ / KL β_{KL}	0.2 / 0.05
	Update epochs / advantage clip	1 / 3.0
	Sampling	Temperature 1.0, top-p 0.95, max 384 new tokens
	Runtime budget	Up to 36 hours, checkpoints every 6 hours
Reward	Document weights	$w_{\mathrm{FN}} = 4.0$, $w_{\mathrm{FP}} = 1.0$
	Sentence cost λ	0.25
	Positive mixture	$\alpha = 0.15$, $\beta = 0.70$, $\gamma = 0.15$
	Oracle shaping	$w_{\mathrm{orc}} = 0.20$
Evaluation	Main sets	1,000 questions each from HotpotQA, MuSiQue, and 2WikiMultihopQA
	Main reasoner	Frozen Qwen2.5-1.5B-Instruct
	RQ4 reasoners	Local Qwen2.5-7B-Instruct, Qwen3.5-Plus, GPT-4o-mini
	Metrics	EM, F1, Recall@3/10, document count, calls, tokens, invalid rate

Training Setup. The base model is Qwen2.5-1.5B-Instruct in bfloat16. SFT inserts LoRA adapters with rank 16, alpha 32, and dropout 0.05 on the attention projections $(q_{\mathrm{proj}}, k_{\mathrm{proj}}, v_{\mathrm{proj}}, o_{\mathrm{proj}})$ and the MLP projections $(\mathrm{gate_proj}, \mathrm{up_proj}, \mathrm{down_proj})$. GRPO first merges the SFT adapter into the base model and then optimizes a fresh LoRA adapter; the reference policy used in the KL term is the frozen pre-update policy. We generate transition records from HotpotQA training questions using the Qwen3.5-Plus teacher and split them into 5,600 training, 700 validation, and 700 held-out records. SFT trains only on the assistant action portion of each serialized transition, masking prompt tokens from the loss, using AdamW with cosine decay, two epochs, learning rate 2×10^{-4}, micro batch size 1, gradient accumulation 8, warmup ratio 0.05, and a maximum sequence length of 6,144 tokens; the best validation-loss checkpoint is retained and reused by GRPO. For each evidence state, GRPO samples $G = 8$ candidate actions, standardizes rewards within the group, clips advantages to $[-3, 3]$, and uses one update epoch, AdamW learning rate 5×10^{-6}, clip range $\epsilon = 0.2$, KL coefficient $\beta_{\mathrm{KL}} = 0.05$, gradient norm clipping at 1.0, and a minimum reward standard deviation of 10^{-6}. Sampling uses temperature 1.0, top-p 0.95, and at most 384 new tokens. Training is capped at 36 wall-clock hours, with checkpoints saved every six hours.

Reward Configuration. The retrieval-grounded reward uses $w_{\mathrm{FN}} = 4$, $w_{\mathrm{FP}} = 1$, $\lambda = 0.25$, and positive mixture weights $(\alpha, \beta, \gamma) = (0.15, 0.70, 0.15)$, where S_{qD} is the one-to-one query–document prefix matching score and S_{stop} is an exact match of the termination decision. When an oracle reference is available, its positive score is shaped with $w_{\mathrm{orc}} = 0.20$.