EVIDENCE-R1: MULTI-TURN SEARCH THROUGH AN EVIDENCE-COMMIT PROTOCOL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn search enables language models to progressively acquire the information needed for complex reasoning, but the continual accumulation of retrieved content also increases context and inference costs. Existing methods compress retrieved information through summarization or memory; however, when subsequent actions can still access the original content, the model may continue to rely on the uncompressed context, making it difficult for the explicitly maintained evidence state to truly serve as the medium for information transfer across turns. To address this issue, we propose Evidence-R1, a framework for multi-turn search based on an evidence-commit protocol. The framework explicitly maintains a cross-turn evidence state throughout the search process. After each retrieval step, the model first updates this state through an independent evidence-commit segment, then removes the raw retrieved content and temporary reasoning, and carries the updated evidence state into the next turn. To further improve evidence commitment, Provenance-Routed Segment Credit assigns local rewards to the evidence-commit segments that construct the relevant state versions, based on the provenance of the finally cited evidence and source-support verification results. Across seven open-domain question answering benchmarks, the 3B and 7B variants of Evidence-R1 achieve average accuracies of 44.1% and 49.1%, respectively, both exceeding the strongest same-scale baseline MemSearcher (43.8% and 48.9%). Under a unified efficiency evaluation setting, Evidence-R1 again achieves higher accuracy than MemSearcher while reducing peak context length by 37.5%–49.7% and inference time by 16.9%–21.2%. These results show that a mandatory and traceable evidence state can reduce the cost of multi-turn search while preserving question answering performance.

1 INTRODUCTION

Multi-turn search allows language models to refine queries, gather evidence, and integrate information across documents for open-domain question answering and multi-hop reasoning. Reinforcement learning methods such as Search-R1 show that a model can learn when to search, what to search for, and when to answer from a final task reward (Jin et al., 2025). Yet as the number of search turns grows, retaining raw retrieved documents, temporary reasoning, and action history continually expands the context and introduces noise. Later actions may therefore depend on any token in the history, including outdated, irrelevant, or unverified content. Once such content enters query or answer generation, its reliability becomes difficult to distinguish; errors can accumulate across turns, and the basis for the final decision becomes difficult to audit.

Existing approaches manage information across turns with periodic summaries, compact memories, or answer-oriented evidence notes (Wu et al., 2025; Yuan et al., 2025; Dai et al., 2025). These approaches reduce context size, but compression alone does not ensure that subsequent actions obtain newly retrieved information through the compressed memory. If a memory update and the next action are produced in the same generation segment while the raw retrieval remains visible, the action can use the raw text directly without reading the newly written memory. We call this interaction pattern, in which actions can bypass compressed memory, *Soft Commit*. In this setting, memory is an accompanying representation rather than a required information channel for the subsequent policy. Inspecting the memory alone cannot reveal which retrieved information the next action actually used or where that information came from. Systems such as MemSearcher reconstruct the next-turn con-

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.

text from the question and updated memory (Yuan et al., 2025), but facts in their free-form memory are not individually bound to sources, and they do not retain a verifiable version lineage and update log. As a result, source support for the memory used by the final answer is difficult to verify, and errors cannot be attributed to specific memory updates.

Learning to construct evidence also requires more specific feedback than a final answer reward provides. Multiple commits in one trajectory share the same success signal, which cannot identify the commits that built the evidence eventually cited or determine whether the declared sources support that evidence. Existing evidence-supervision methods generally evaluate how well evidence supports an answer or estimate the value of an evidence state to the answer (Dai et al., 2025; Gao & Marttinen, 2026). These signals act on final evidence or an overall state; they do not establish whether an individual statement remains supported by its source after each addition, deletion, or revision. An error can consequently persist through evidence revisions, while the final answer reward cannot identify the specific version that needs correction.

We propose Evidence-R1, which combines mandatory evidence commitment with local credit assignment for evidence construction. The **Evidence-Commit Protocol** separates post-retrieval evidence updates from the next action into independent generation segments. After retrieval, the model enters a dedicated Evidence Commit Segment and outputs only a candidate evidence-state update. The environment validates and atomically applies a legal update, removes the raw retrieval and temporary reasoning, and reconstructs the next Action Segment from the question, a query-only Search History, and the current Evidence State. This *commit-before-use* constraint requires newly retrieved information to pass through an explicit commit before it can affect the next search or answer. It prevents the current raw retrieval from bypassing the state update and creates an auditable boundary for information transfer. To train evidence construction, we introduce **Provenance-Routed Segment Credit**. Starting from the active evidence explicitly cited in the final answer, it traces the corresponding ADD/UPDATE version lineages and checks whether the declared source supports each version. Local credit for source-supported versions is routed to the Evidence Commit Segment that constructed them, while a global task reward continues to supervise final answer quality. Final citations thus define the scope of credit tracing, version lineages locate the responsible commits, and source support determines the local feedback. On Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, Evidence-R1 obtains average exact match (EM) scores of 44.1% and 49.1% across seven open-domain question answering benchmarks, both above the strongest same-scale baseline, MemSearcher. In a unified 3B efficiency setting, it also achieves higher average EM while reducing peak context by 37.5%–49.7% and inference time per question by 16.9%–21.2%. Controlled ablations further show that the hard commit protocol reduces context cost without degrading answer quality, query history supplied to action generation reduces redundant retrieval, and provenance credit improves the source faithfulness of committed and finally cited evidence. Our main contributions are as follows:

- We propose an Evidence-Commit Protocol with a versioned evidence state. It separates post-retrieval commitment from the next action, establishes a *commit-before-use* boundary through environment-controlled context reconstruction, and retains the source and revision history of each evidence item.
- We introduce Provenance-Routed Segment Credit. Final citations define the scope of credit tracing, evidence-version lineages identify the relevant commits, and per-version source support assigns local credit, providing traceable training feedback for multi-turn evidence construction.
- We evaluate on seven open-domain question answering benchmarks. Evidence-R1 improves average EM over the strongest same-scale baseline at both 3B and 7B scales, and achieves higher average EM with lower context and inference costs in a unified 3B efficiency setting. An independent evidence evaluation further measures source support for the evidence cited in final answers.

2 Related Work

Two central challenges in search-augmented reasoning are managing context across turns and assigning credit to intermediate steps. We review both lines of work and clarify how evidence commitment and source-lineage modeling differ from prior approaches.

2.1 Context Management in Search-Augmented Reasoning

Multi-turn search must retain information from earlier retrievals, but accumulating raw documents increases context cost. Search-R1 retains retrieved results in the interaction context for subsequent reasoning and search (Jin et al., 2025). ReSum periodically compresses the history into summaries, but actions between summary boundaries can still access newly retrieved content directly; it does not impose a commit constraint after each retrieval (Wu et al., 2025). MemSearcher updates a compact memory after each retrieval and builds the next-turn context from the question and updated memory. It thus replaces the raw interaction history with turn-by-turn memory updates, but its free-form memory does not explicitly maintain source bindings and version lineages for individual evidence items. The basis and revision history of retained information are consequently difficult to trace (Yuan et al., 2025). We therefore combine an Evidence-Commit Protocol with a versioned evidence state: newly retrieved information must be explicitly committed before the next action can use it, while source and revision records support subsequent evidence auditing and credit assignment.

2.2 Process Rewards and Credit Assignment

Fine-grained rewards in search-augmented reasoning supplement final answer feedback by supervising intermediate behavior. StepSearch uses information gain and a redundancy penalty to supervise search steps; HiPRAG uses hierarchical process rewards to assess the necessity of search and non-search decisions (Zheng et al., 2025; Wu et al., 2026). EviNote-RAG uses entailment supervision to evaluate whether final evidence notes support a reference answer (Dai et al., 2025). MAVEN estimates the answer-conditioned value of an evidence state and combines online gains with retrospective contributions to assign rewards to the corresponding action segments (Gao & Marttinen, 2026). Although MAVEN assigns credit at the action-segment level, its reward primarily measures how state changes affect answer support; it does not directly verify the source faithfulness of individual evidence statements. Its retrospective assessment of final retained items also does not explicitly track successive versions of an item and their changing sources. Provenance-Routed Segment Credit instead traces the ADD/UPDATE lineages of finally cited evidence back to the relevant Evidence Commit Segments and assigns local credit according to the source support of each version. Alongside the global answer reward, this mechanism provides source-faithfulness feedback to the specific commits that constructed and revised cited evidence.

3 Method

Evidence-R1 combines two mechanisms (Figure 1). The **Evidence-Commit Protocol** makes an explicit state update the route from each retrieval to the next action. **Provenance-Routed Segment Credit** checks the sources of versions leading to final citations and credits the Evidence Commit Segments that created them. We describe the interaction protocol and training signal in turn.

3.1 Evidence-Commit Protocol

An explicit state alone cannot prevent a later action from reading the raw observation directly. If the state update and the next query or answer are produced in one continuous generation, the model can still use retrieval results that have not been evicted, yielding a *Soft Commit* that bypasses the explicit state. Evidence-R1 therefore separates evidence commitment and the next policy action into independent generation segments. Between them, the environment applies the state update, evicts information, and reconstructs the context.

Action–Commit Interaction Protocol. Evidence-R1 organizes multi-turn search as alternating action decisions and evidence commits. In an Action Segment, the model decides whether to search or answer from the question, a query-only Search History, and the current Evidence State. If it searches, the environment executes the query and returns retrieved results. In a separate Evidence Commit Segment, the model then selects relevant information and proposes an evidence-state update. After a legal update is applied, the environment removes the raw results, temporary reasoning, and generation cache, and reconstructs the next Action Segment using only the question, query history, and current evidence state. This repeats until the model produces a final answer citing evidence IDs. Algorithm 1 gives the complete procedure.

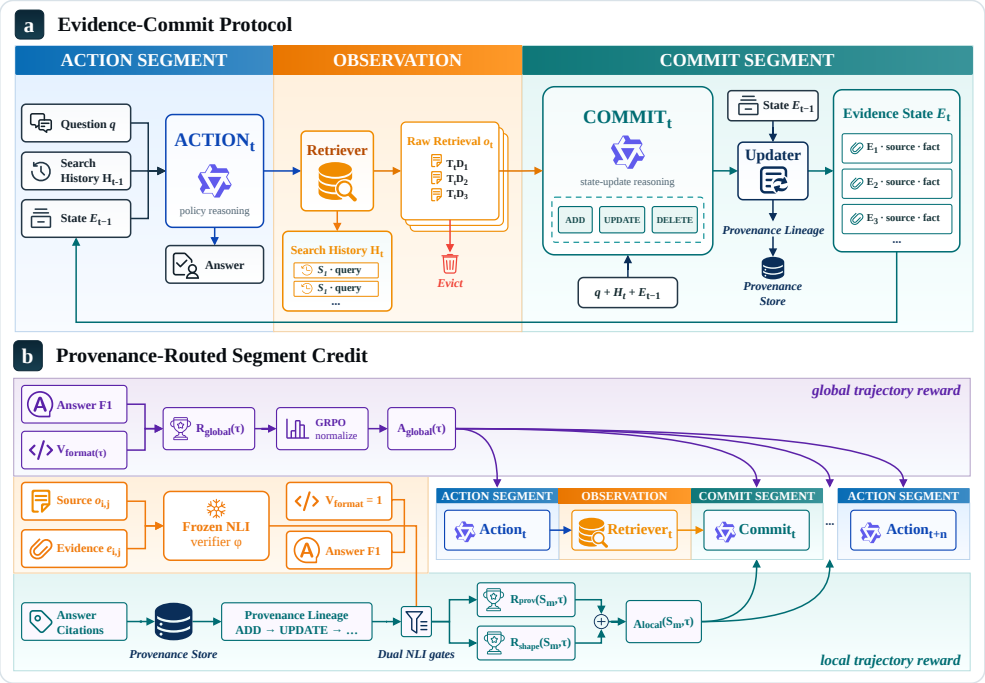


Figure 1: Evidence-R1 overview. (a) Commit retrieved evidence, evict the raw observation, and generate the next action. (b) Route source-support credit along cited evidence lineages to the responsible COMMIT segments.

Formally, the input and output of the t-th Action Segment are

$$C_t^{\mathrm{A}} = q \oplus H_{t-1} \oplus E_{t-1},$$
$$(r_t^{\mathrm{A}}, a_t) \sim \pi_\theta\left(\cdot \mid C_t^{\mathrm{A}}, \mathrm{phase} = \mathrm{A}\right), \qquad a_t \in \{\mathrm{SEARCH}, \mathrm{ANSWER}\}.$$

Here, $\oplus$ denotes concatenation according to the phase-specific template. If $a_t = \mathrm{ANSWER}$, the model produces a final answer listing active evidence IDs in the `evidence` attribute, and the environment terminates the trajectory after validation. If $a_t = \mathrm{SEARCH}$, the model produces query s_t; the environment updates Search History as $H_t = H_{t-1} \parallel s_t$, where $\parallel$ denotes chronological appending, and retrieves K documents $o_t = \mathcal{R}(s_t)$. Once the search budget B is reached, the next Action Segment may only answer.

After retrieval, the model enters a separate Evidence Commit Segment, generates nonempty reasoning r_t^{C} about evidence selection and state revision using the current observation, and outputs a candidate state update ΔE_t:

$$C_t^{\mathrm{C}} = q \oplus H_t \oplus E_{t-1} \oplus o_t, \qquad (r_t^{\mathrm{C}}, \Delta E_t) \sim \pi_\theta\left(\cdot \mid C_t^{\mathrm{C}}, \mathrm{phase} = \mathrm{C}\right).$$

The two segment types share policy parameters θ, but have different input contexts, phase prompts, and stopping conditions. The Evidence Commit Segment stops immediately after producing the closing tag `</state_update>`. The next Action Segment is generated in a separate call, without reusing the current generation cache or accessing the raw observation.

Versioned Evidence State and Atomic Update. To add and revise evidence within a bounded context while retaining source and version histories, Evidence-R1 maintains a versioned Evidence State that separates policy-visible active evidence from historical versions stored by the environment.

Specifically, Evidence State E_t is a set of active evidence subject to a fixed token budget; each item has a stable, unique ID, a source identifier, and a factual statement. The model can propose three operations in ΔE_t: ADD creates a new item with a source; UPDATE retains its ID but creates a new version that replaces the active one; and DELETE removes an item from the active state. At the

hard commit boundary, the environment validates the entire candidate update. It accepts the update as a unit only if every operation is legal and the resulting state meets the budget; otherwise, it rejects the entire update and leaves the state unchanged. An empty update is allowed when the observation contains no task-relevant information.

Subsequent Action Segments receive only the current active versions. Outside the policy context, the environment maintains a provenance store recording successfully committed versions, sources, and UPDATE lineages. Replacing or deleting active evidence does not destroy this history. These records support version tracing and the credit routing described in Section 3.2. Appendix B.1 details the representation and maintenance of versions, and Algorithm 2 specifies the update rules.

Commit-before-use Property. At a hard commit boundary, the environment executes the following steps in order:

$$\text{STOP} \longrightarrow \text{VALIDATE/APPLY} \longrightarrow \text{EVICT} \longrightarrow \text{REGENERATE}.$$

After the model stops generating, the environment validates and atomically applies the candidate update to obtain E_t. It then removes the current observation, both types of temporary reasoning, and the generation cache, and reconstructs the next Action Segment context solely from $q \oplus H_t \oplus E_t$. If the update is rejected, $E_t = E_{t-1}$, but the same reset still occurs. Because H_t is fixed before the current retrieval returns, new retrieval content must first enter E_t through an evidence commit to affect the immediately following action. We call this local structural property *commit-before-use*: newly retrieved information must be explicitly committed before the next policy action can use it.

This property constrains the information path between o_t and a_{t+1} only. Later queries may restate committed facts in E_t and preserve them in Search History. Even if the corresponding evidence is subsequently updated or deleted, those facts may still affect later actions through historical queries. Evidence State is thus the mandatory channel for current retrieval content before the next action, but it is not the only way factual information can persist over an entire trajectory.

3.2 Provenance-Routed Segment Credit

Global Answer F1 supervises only the final answer. It cannot distinguish the Evidence Commit Segments that constructed finally cited evidence or determine whether the declared sources support it. We therefore introduce provenance-routed segment credit. Starting from active evidence cited in the final answer, it traces the relevant commits along ADD/UPDATE version lineages and assigns local credit to the corresponding Evidence Commit Segments according to source-support checks. This provides fine-grained feedback for evidence construction alongside global task supervision.

Global Trajectory Reward. Evidence-R1 uses a global trajectory reward to supervise final task success. Let $V_{\text{format}}(\tau) \in \{0, 1\}$ indicate whether the full trajectory satisfies the complete interaction protocol, including atomic acceptance of every candidate update by the updater, rather than only valid tag syntax. If any update is rejected, $V_{\text{format}}(\tau) = 0$, even if the rollout continues and the final answer is correct. Let $R_{\text{F1}}(\tau) \in [0, 1]$ be token-level F1 between the predicted and reference answers. The global trajectory reward is

$$R_{\text{global}}(\tau) = V_{\text{format}}(\tau)\big[0.1 + 0.9R_{\text{F1}}(\tau)\big].$$

GRPO normalizes sampled trajectories for the same question within a group to obtain global advantage $A_{\text{global}}(\tau)$, which is assigned to every policy segment in the trajectory.

Citation-Lineage-Routed Source Credit. To assign local training signal to the commits that constructed finally cited evidence, we parse the set $C(\tau)$ of still-active citations from the final answer's `evidence` attribute. For each $E_i \in C(\tau)$, the provenance store traces its successful version lineage

$$L_i = \left(u_{i,0}^{\text{ADD}}, u_{i,1}^{\text{UPDATE}}, \ldots, u_{i,K_i}^{\text{UPDATE}}\right),$$

Here $u_{i,j}$ is the successful operation that created the corresponding evidence version, and u_i^{final} created the final active version. The lineage contains only ADD and UPDATE operations accepted by the atomic updater.

For each operation $u_{i,j}$, we use the source text declared by that version as the premise and the committed evidence statement as the hypothesis. A frozen NLI model ϕ computes source support as

$$s_{\mathrm{NLI}}(u_{i,j}) = \max\left(0,\, p_\phi^{\mathrm{ent}}(u_{i,j}) - \max\left[p_\phi^{\mathrm{neutral}}(u_{i,j}), p_\phi^{\mathrm{contra}}(u_{i,j})\right]\right),$$

The three probabilities correspond to entailment, neutrality, and contradiction. Given threshold θ_{NLI}, we define a gate for the final version and the set of operations passing the source check:

$$v_i^{\mathrm{final}} = \mathbf{1}\left[s_{\mathrm{NLI}}(u_i^{\mathrm{final}}) > \theta_{\mathrm{NLI}}\right], \qquad L_i^+ = \{u_{i,j} \in L_i \mid s_{\mathrm{NLI}}(u_{i,j}) > \theta_{\mathrm{NLI}}\}.$$

Only when the final version passes the gate can valid operations in its lineage receive credit. Operation-level source credit is

$$r_{\mathrm{prov}}(u_{i,j}, \tau) = \begin{cases} \dfrac{R_{\mathrm{F1}}(\tau)\, V_{\mathrm{format}}(\tau)\, v_i^{\mathrm{final}}\, s_{\mathrm{NLI}}(u_{i,j})}{|C(\tau)|\,|L_i^+|}, & u_{i,j} \in L_i^+, \\ 0, & u_{i,j} \notin L_i^+. \end{cases}$$

Here R_{F1} scales source credit by final answer quality, while V_{format} gives malformed trajectories zero source credit. If $L_i^+ = \varnothing$ or $C(\tau) = \varnothing$, the corresponding source credit is zero. The denominator terms $|C(\tau)|$ and $|L_i^+|$ normalize by the number of final citations and the length of the valid lineage, respectively, limiting reward inflation through redundant citations or repeated UPDATE operations.

We aggregate operation-level credit over the Evidence Commit Segment that produced each operation. Let $\mathrm{seg}(u_{i,j})$ denote that segment. Then

$$R_{\mathrm{prov}}(S_t^{\mathrm{C}}, \tau) = \sum_{E_i \in C(\tau)} \sum_{\substack{u_{i,j} \in L_i \\ \mathrm{seg}(u_{i,j}) = S_t^{\mathrm{C}}}} r_{\mathrm{prov}}(u_{i,j}, \tau).$$

Joint Training Advantage. To control evidence-state size and editing cost, we define a capacity cost from the number of active evidence items and their serialized token count. Combining the difference in cost before and after a commit with an operation cost yields the state-shaping reward R_{shape}, derived in Appendix B.2. We combine provenance credit with this reward to define the local advantage increment for policy segment S_m:

$$A_{\mathrm{local}}(S_m, \tau) = \mathbf{1}[S_m \text{ is a COMMIT segment}]\big(R_{\mathrm{prov}}(S_m, \tau) + R_{\mathrm{shape}}(S_m, \tau)\big).$$

This increment applies only to Evidence Commit Segments; other policy segments receive zero local advantage. Adding it to the global trajectory advantage with a weight gives the training advantage of segment S_m:

$$A_{\mathrm{total}}(S_m, \tau) = A_{\mathrm{global}}(\tau) + w_{\mathrm{local}} A_{\mathrm{local}}(S_m, \tau).$$

Here A_{global} is obtained by within-group normalization of the global trajectory reward, and w_{local} is the local advantage weight. The local term is not normalized separately and applies only to policy tokens in the corresponding Evidence Commit Segment; it is not broadcast to other segments. During training, the policy probability of each segment is computed under the context in which it was actually generated, matching information visibility between training and rollout. Appendix B.3 gives the full trajectory likelihood factorization.

4 Experiments

We first evaluate the overall question answering performance of Evidence-R1 on seven open-domain benchmarks. We then compare context and inference costs under different search budgets and analyze changes in answer quality and evidence source support during training. Finally, ablations test how the main components affect answer accuracy, evidence quality, search behavior, and context efficiency.

4.1 Experimental Setup

Datasets. We follow the data protocol of Search-R1 (Jin et al., 2025), using the training sets of Natural Questions (NQ) (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) for reinforcement learning. We evaluate on the single-hop datasets NQ, TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023), and the multi-hop datasets HotpotQA, 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023).

Table 1: **Exact match (%) on seven QA benchmarks.** Single-hop: NQ, TriviaQA, PopQA; multi-hop: HotpotQA, 2Wiki, MuSiQue, Bamboogle. Results are grouped by backbone and size. Avg. is the unweighted mean. Bold/underlined: best/second-best at each size; shaded: ours.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Qwen-2.5-3B-Base/Instruct								
Search-o1	16.6	31.0	8.2	14.8	22.4	5.2	22.4	17.2
IRCoT	11.1	31.2	20.0	16.4	17.1	6.7	24.0	18.1
Search-R1-base	40.6	58.7	43.5	28.4	27.3	4.9	8.8	30.3
Search-R1-instruct	34.1	54.5	37.8	32.4	31.9	10.3	26.4	32.5
AutoRefine-base	46.7	62.0	45.0	40.5	39.3	15.7	34.4	40.5
AutoRefine-instruct	43.6	59.7	44.7	40.4	38.0	16.9	33.6	39.6
ReSearch	20.4	33.5	17.3	35.6	39.3	17.3	37.6	28.7
ZeroSearch	41.4	57.4	44.8	27.4	30.0	9.8	11.1	31.7
O^2-Searcher	44.4	59.7	42.9	38.8	37.4	16.0	34.4	39.1
MemSearcher	<u>47.0</u>	<u>63.8</u>	**47.9**	<u>43.9</u>	<u>43.5</u>	<u>17.9</u>	<u>42.4</u>	<u>43.8</u>
Evidence-R1 (Ours)	**47.2**	**64.7**	<u>47.0</u>	**44.9**	**43.8**	**18.2**	**43.2**	**44.1**
Qwen-2.5-7B-Base/Instruct								
Search-o1	19.4	40.6	11.4	17.0	27.0	8.6	30.4	22.1
IRCoT	22.4	47.8	30.1	13.3	14.9	7.2	22.4	23.9
Search-R1-base	48.0	63.8	45.7	43.3	38.2	19.6	43.2	43.1
Search-R1-instruct	39.3	61.0	39.7	37.0	41.4	14.6	36.8	38.5
ReSearch	40.9	63.7	44.6	43.5	47.6	<u>22.3</u>	42.4	43.6
ZeroSearch	43.6	65.2	**48.8**	34.6	35.2	18.4	27.8	39.1
R1-Searcher	40.4	52.2	41.0	44.2	**51.3**	15.8	36.8	40.2
MemSearcher	**52.7**	<u>68.1</u>	47.8	<u>50.8</u>	48.6	**25.8**	**48.8**	<u>48.9</u>
Evidence-R1 (Ours)	<u>52.0</u>	**68.5**	**48.8**	**52.5**	<u>49.9</u>	<u>25.5</u>	<u>46.4</u>	**49.1**

Baselines. We compare Evidence-R1 with IRCoT (Trivedi et al., 2023) and Search-o1 (Li et al., 2025), which prompt models to retrieve interactively during reasoning, and with Search-R1 (Jin et al., 2025), AutoRefine (Shi et al., 2026), ReSearch (Chen et al., 2025), ZeroSearch (Sun et al., 2025), O^2-Searcher (Mei et al., 2025), R1-Searcher (Song et al., 2025), and MemSearcher (Yuan et al., 2025), which use reinforcement learning to optimize search and reasoning trajectories, evidence selection, or memory management.

Following Search-R1, Table 1 uses exact match (EM, ↑) as the primary metric. Appendix D.1 specifies the normalization and matching rules. All results are percentages, and Avg. is the unweighted mean EM across the seven datasets. Evaluation uses deterministic decoding, whereas training rollouts sample trajectories for within-group comparisons.

Models and Training. Following Search-R1, we initialize the policies from Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Qwen et al., 2025). Training and evaluation use the 2018 English Wikipedia snapshot (wiki-18) (Karpukhin et al., 2020) and an E5-base-v2 retriever (Wang et al., 2022).

We train the policies with full-parameter GRPO, independently sampling five complete search trajectories per question to compute relative advantages within each group. During training, each retrieval returns the top five documents, with at most five searches. The test budget is top-20 retrieval with at most ten searches. Context and inference efficiency experiments instead fix top-5 retrieval and use search budgets of five and ten. Appendix D.2 gives the complete retrieval, generation, and evidence-state budgets; Appendix D.3 details the rewards and source-support settings.

4.2 Main Results

Table 2: **Answer quality and inference efficiency under a unified setup.** Avg. Searches is per-question mean; other metrics are averaged within each dataset, then across seven datasets.

B	Method	EM ↑ (%)	Avg. Searches ↓ (searches/question)	Peak Context ↓ (tokens)	Cumulative Tokens ↓ (tokens)	Inference Time ↓ (s/question)
5	Search-R1	33.8	1.62	1786	5154	16.5
	MemSearcher	41.1	2.05	1942	4152	15.9
	Evidence-R1	**41.7**	**2.07**	**1214**	**3793**	**12.53**
10	Search-R1	32.6	1.84	1896.5	6,308.4	16.81
	MemSearcher	41.4	2.12	2413.4	4749.7	15.28
	Evidence-R1	**41.6**	**2.07**	**1215**	**3802**	**12.70**

Table 3: **Evidence source faithfulness (%) judged by `GLM-5.3-Flash`.** $n_{\text{commit}}/n_{\text{cited}}$: evaluated committed/cited versions; Avg.: unweighted mean across seven datasets.

Metric	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Total/Avg.
n_{commit}	5,785	19,350	24,140	16,368	34,808	6,915	3,489	110,855
Commit Faithfulness	79.4	78.6	86.3	82.4	78.8	80.2	83.8	81.3
n_{cited}	4,418	14,869	17,932	10,646	18,186	3,489	162	69,702
Cited Faithfulness	77.7	77.2	85.4	81.4	75.4	77.8	79.0	79.1

Overall Question Answering Performance. Table 1 shows that Evidence-R1 obtains average EM scores of 44.1% and 49.1% at the 3B and 7B scales, respectively, above the strongest same-scale baseline, MemSearcher, at 43.8% and 48.9%. These are gains of 0.3 and 0.2 percentage points. At 3B, Evidence-R1 has the best score on six of seven datasets and outperforms MemSearcher on all four multi-hop datasets. At 7B, it obtains the best or tied-best score on TriviaQA, PopQA, and HotpotQA; its overall performance is comparable to MemSearcher, with a slightly higher average EM. Although the average gains are small, the higher overall accuracy at both scales supports the effectiveness of the method.

Context and Inference Efficiency. Table 2 compares Search-R1, MemSearcher, and Evidence-R1 under a unified configuration. All three use Qwen2.5-3B-Instruct and top-5 retrieval; only the search limit B varies. Appendix E.2 gives the full configuration and metric definitions.

Table 2 shows that Evidence-R1 reaches average EM scores of 41.7% and 41.6% for $B = 5$ and $B = 10$, respectively, slightly above both baselines, while performing about as many searches as MemSearcher. It has the lowest peak context, cumulative tokens, and inference time. Increasing the search limit from five to ten leaves EM, actual search count, and resource costs nearly unchanged, indicating no further answer gain from the larger budget. The efficiency gain is therefore primarily associated with the bounded Evidence State limiting the accumulation of raw retrieved content, rather than with additional retrieval.

Evidence Source Faithfulness. We use `GLM-5.3-Flash`, provided by Zhipu AI (API model identifier: `glm-5.3-flash`), as an independent LLM judge to assess whether each evidence statement is supported by its declared source. This judge is used only for test-time source-faithfulness evaluation and is independent of the frozen NLI model that computes source-support rewards during training. Appendix E.3 details the evaluation protocol and metrics.

As Table 3 shows, the mean Commit Faithfulness and Cited Faithfulness of Evidence-R1 are 81.3% and 79.1%, respectively. Thus, most committed and finally cited evidence is fully supported by its declared source. Cited Faithfulness is slightly lower than Commit Faithfulness on all seven datasets, a mean difference of 2.2 percentage points. The final citation set is therefore no more concentrated on fully supported versions than the full set of commits. Citation selection, version updates, and retention in the final state may all contribute to this pattern; the result alone does not show that the citation process reduces source faithfulness.

Table 4: **Ablation configurations.** B4 is the full model; B2–B4 use Hard Commit. Raw access concerns the next Action's access to o_t; history access concerns direct access to query-only Search History. R_{shape}/R_{prov}: state shaping/provenance credit. All use global reward and $w_{\text{local}} = 1$.

Configuration	COMMIT–Action transition	Raw observation access by the next Action	History access		Local rewards	
			Action	COMMIT	R_{shape}	R_{prov}
B1: Soft Commit	Continuous output within a single generation	✓	✓	✓	✓	✓
B2: w/o Action History	Stop, commit, and rebuild context	×	×	✓	✓	✓
B3: w/o Provenance Credit	Stop, commit, and rebuild context	×	✓	✓	✓	×
B4: Full Model	Stop, commit, and rebuild context	×	✓	✓	✓	✓

4.3 Ablation Studies

To isolate the roles of Evidence-R1's components, we conduct three controlled ablations on NQ and HotpotQA: (1) the effect of the hard evidence-commit boundary on answer quality and context cost; (2) the effect of giving query history directly to action generation; and (3) the incremental contribution of provenance credit to source faithfulness and answer quality when the state-shaping reward is retained. Table 4 defines four configurations. B4 is the full model, while B1–B3 vary the commit protocol, Action history access, or local reward composition. Appendix F provides the full setup and aggregation rules.

Effect of the Hard Evidence-Commit Boundary. Relative to Soft Commit (B1), Hard Commit (B4) raises Answer EM by 2.0 and 3.0 percentage points on NQ and HotpotQA while reducing cumulative tokens by 330 and 1,167 (10.6% and 23.8%), respectively. F1 also rises and peak context falls on both datasets. The full stop/apply/evict/rebuild transition thus reduces context cost without sacrificing answer quality; this comparison does not isolate its individual steps.

Effect of Query History on Action Generation. B2 omits query-only Search History from Action input while retaining it for COMMIT. Relative to B2, the full model reduces semantic duplicate queries by 3.2 and 7.8 percentage points and searches per question by 0.63 and 1.42 on NQ and HotpotQA, respectively; Answer EM rises by 0.2 and 2.6 points. Direct history access for Action therefore reduces redundant search without lowering answer quality.

Effect of Provenance-Routed Credit. B3 removes R_{prov} but retains state shaping. Adding provenance credit in B4 increases Commit Faithfulness by 7.3 and 10.8 points and Cited Faithfulness by 5.9 and 10.5 points on NQ and HotpotQA, respectively; Answer EM rises by 1.3 and 0.6 points. This supports an incremental source-faithfulness benefit beyond state shaping. These metrics assess support by the declared source, not its objective truth.

5 Conclusion

We proposed Evidence-R1, which uses an Evidence-Commit Protocol to require the model to commit retrieved information to a versioned Evidence State before using it, and Provenance-Routed Segment Credit to assign source-support credit to the COMMIT segments that constructed the finally cited evidence. Experiments across open-domain question answering benchmarks and controlled ablations show that Evidence-R1 maintains competitive answer performance while reducing redundant search, context and inference costs, and increasing source support for committed and cited evidence. These results suggest that a constrained, explicit evidence state can provide a common interface for multi-turn search, evidence maintenance, source auditing, and fine-grained credit assignment.

AI use statement

This research used OpenAI GPT-5.6 only for language polishing and expression assistance in paper writing, including improving the grammar, clarity, concision, and readability of text written by the authors. The tool was not used to formulate or refine research questions or hypotheses, design methods or experiments, implement code, generate or process data, analyze or interpret results, retrieve or generate references, or create or modify figures or other research outputs. The authors reviewed all

AI-assisted text and checked relevant statements against the original manuscript, method implementation, experimental records, and cited literature. Suggestions that were inaccurate, unsupported, or changed the intended meaning were not adopted. The authors take responsibility for all final content, including text edited with generative AI assistance.

Ethics statement

The authors have read and comply with the ICLR Code of Ethics. This study aims to improve the traceability and context efficiency of evidence use in multi-turn search agents. Explicit source identifiers, version lineages, and a hard evidence-commit boundary help audit how retrieved information enters later decisions, but do not make the system inherently truthful, unbiased, or safe. The experiments use existing public question answering datasets and an archived English Wikipedia snapshot; they involve no automated decisions about individuals and no private personal data. Like other search-augmented language models, Evidence-R1 may inherit factual errors, social biases, and uneven coverage from base models, evaluation data, and retrieved corpora. It may retrieve text containing personal information, harmful content, or copyrighted material. The source-faithfulness gate only checks whether a declared source supports a committed evidence statement; it does not verify the source against real-world facts or assess its fairness, legality, or appropriateness. Deployment outside a research setting therefore still requires corpus and license governance, privacy and safety filtering, source-quality controls, access controls, and human review tailored to the application. Releasing code and pretrained model weights can support reproducible research but may also enable misuse; users should respect applicable data, model, and content licenses and reassess risks for each application.

Reproducibility statement

We are committed to making this study reproducible. Section 3 fully defines the Evidence-Commit Protocol, versioned Evidence State, Provenance-Routed Segment Credit, reward functions, and the inputs, outputs, and stopping conditions of each generation segment. Section 4 describes the training and evaluation datasets, baselines, metrics, policy models, retrieval corpus and retriever, and principal training and interaction settings. Appendix C provides pseudocode for rollout, atomic state updates, and segment credit assignment, while Appendix G includes the complete Action Segment and Evidence Commit Segment prompts. To further support independent reproduction, we will release the full source code and pretrained model weights after publication, along with scripts and configurations for data preprocessing, training, inference, and evaluation. Public datasets, base models, and the Wikipedia corpus remain subject to their respective licenses and access terms.

References

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. Research: Learning to reason with search for llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2503.19470.

Yuqin Dai, Guoqing Wang, Yuan Wang, Kairan Dou, Kaichen Zhou, Zhanwei Zhang, Shuo Yang, Fei Tang, Jun Yin, Pengyu Zeng, et al. Evinote-rag: Enhancing rag models via answer-supportive evidence notes. *arXiv preprint arXiv:2509.00877*, 2025.

Ya Gao and Pekka Marttinen. Evidence-state rewards for long-context reasoning. *arXiv preprint arXiv:2607.02073*, 2026.

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

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

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

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 *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, et al. O^2-searcher: A searching-based agent model for open-domain open-ended question answering. *arXiv preprint arXiv:2505.16582*, 2025.

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

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, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. *Advances in Neural Information Processing Systems*, 38:155930–155958, 2026.

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

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, 2023. doi: 10.18653/v1/2023.acl-long.557.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 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 *International Conference on Learning Representations*, volume 2026, pp. 82317–82347, 2026.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

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

Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. Memsearcher: Training llms to reason, search and manage memory via end-to-end reinforcement learning. *arXiv preprint arXiv:2511.02805*, 2025.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21816–21841, 2025.

A NOTATION

Table 5: Main notation used in this paper.

Symbol	Meaning
q, y^*	Input question and reference answer.
π_θ	Policy model with parameters θ.
$\mathcal{R}, B, K$	Retriever, maximum number of searches, and documents returned per search.
s_t, o_t	Search query and retrieved result at turn t, where $o_t = \mathcal{R}(s_t)$.
H_t, E_t	Query-only Search History and active Evidence State after turn t.
ΔE_t	State update produced by the Evidence Commit Segment at turn t.
$S_t^{\mathrm{A}}, S_t^{\mathrm{C}}$	Action Segment and Evidence Commit Segment at turn t.
$C_t^{\mathrm{A}}, C_t^{\mathrm{C}}$	Input contexts actually visible to the two segment types at turn t.
$r_t^{\mathrm{A}}, r_t^{\mathrm{C}}$	Temporary reasoning generated within the two segment types.
τ	A complete multi-turn search and answer trajectory.
$e_i^{(v_i)}$	Current version of evidence item i, with version number v_i.
$C(\tau), L_i$	Set of active evidence cited in the final answer and successful version lineage of item i.
ϕ, s_{NLI}	Frozen NLI verifier and source-support score.
θ_{NLI}	NLI source-support gate threshold.
$R_{\mathrm{F1}}, V_{\mathrm{format}}$	Answer F1 reward and trajectory-format validity indicator.
$A_{\mathrm{global}}, A_{\mathrm{local}}$	Within-group-normalized global advantage and segment-routed local advantage increment.
$T_{\mathrm{budget}}, w_{\mathrm{local}}$	Evidence State token budget and local reward weight.

B ADDITIONAL METHOD DETAILS

B.1 EVIDENCE-VERSION REPRESENTATION AND SOURCE RECORDS

Version Representation and Policy Visibility. The current version of evidence item i is represented as

$$e_i^{(v_i)} = (\mathrm{id}_i, \mathrm{src}_i, \mathrm{text}_i)\,,$$

where v_i is the current version number, id_i is an evidence ID that is stable and unique within a trajectory, src_i identifies the visible source cited when the version was created, and text_i is the task-relevant fact committed from that source. The source identifier and factual statement are recorded with each version. UPDATE retains the evidence ID and creates a new version with a new source identifier and statement.

The Evidence State after commit t is defined as

$$E_t = \left\{ e_i^{(v_i)} \;\middle|\; e_i^{(v_i)} \text{ is active after commit } t \right\}.$$

The serialized token count of E_t is constrained by a fixed budget T_{budget}. Subsequent Action Segments receive only the current active versions in Evidence State, not historical versions that have been replaced or deleted.

Provenance-Store Maintenance. The environment maintains a provenance store P outside the policy context. For every evidence version produced by ADD or UPDATE in a successful atomic commit, it records the version and source identifiers, the corresponding visible source text, the committed statement, and the Evidence Commit Segment that generated the operation. For UPDATE, it also records the lineage link to the preceding version. Sources cited by ADD and UPDATE must belong to the current observation. If a retrieved document is truncated by the observation budget, only the fragment actually visible to the model in the current Evidence Commit Segment is stored and used in later source-support evaluation.

Version records and the active state take effect in the same atomic commit. If a candidate update is rejected, $E_t = E_{t-1}$ and the provenance store remains unchanged; no candidate versions from that update are retained. UPDATE replaces the old version in the active state but does not overwrite historical records in the provenance store. DELETE only removes an item from the active state;

it preserves its version history and creates no new evidence statement. Rejected operations and DELETE therefore do not enter the ADD/UPDATE lineages used for source credit. Algorithm 2 gives the full validation and commit procedure.

After the trajectory ends, the system starts from the active evidence IDs cited in the final answer, follows version links to reconstruct each successful update lineage from its initial ADD to its final active version, and identifies each version's source text and generating Evidence Commit Segment for local credit routing in Section 3.2. The provenance store is used only for source auditing, version tracing, and credit assignment. It is absent from both Action Segment and Evidence Commit Segment input contexts, so it does not create another path around Evidence State.

B.2 State-Compactness Shaping

In addition to source-support credit, each Evidence Commit Segment that completes an atomic commit receives a difference-based shaping signal for changes in Evidence State capacity. Let $N_{\text{active}}(E_t)$ and $T_{\text{active}}(E_t)$ denote the number of active evidence items and their serialized token count after commit t, respectively, and let T_{budget} be the token budget of Evidence State. The capacity cost of E_t is

$$\mathcal{C}(E_t) = \lambda_n N_{\text{active}}(E_t) + \lambda_{\text{tok}} \frac{T_{\text{active}}(E_t)}{T_{\text{budget}}},$$

where λ_n and λ_{tok} control the contributions of evidence count and serialized length to capacity cost.

For the t-th Evidence Commit Segment, we define a compactness reward from the difference in capacity cost before and after commitment and impose an additional cost for state operations performed in that segment:

$$r_{\text{compact},t} = \mathcal{C}(E_{t-1}) - \mathcal{C}(E_t), \qquad r_{\text{opcost},t} = -\lambda_{\text{op}} n_{\text{operations},t},$$

where $n_{\text{operations},t}$ is the number of evidence operations committed in this Evidence Commit Segment, and λ_{op} is the cost coefficient per operation. The resulting state-shaping reward is

$$R_{\text{shape}}(S_t^{\text{C}}, \tau) = r_{\text{compact},t} + r_{\text{opcost},t}.$$

Under these definitions, ADD usually increases both the number and serialized length of active evidence, yielding a negative compactness reward. UPDATE leaves the number of active items unchanged, so its compactness reward depends on the difference in serialized token counts before and after revision. DELETE yields a positive compactness reward only when it actually shrinks the active state. For an empty update satisfying $E_t = E_{t-1}$ the compactness reward is 0.

The difference term telescopes across consecutive commits:

$$\sum_{t=1}^{T} r_{\text{compact},t} = \mathcal{C}(E_0) - \mathcal{C}(E_T).$$

Thus, if an item is added and later deleted, and the final capacity returns to its pre-ADD value, the compactness rewards for those operations sum to 0. Both ADD and DELETE nevertheless incur operation costs, making an edit cycle that leaves the final state unchanged locally disadvantageous.

B.3 Segmented Policy Likelihood

The Evidence-Commit Protocol gives different generation segments different information visibility. During training, each segment's policy probability is therefore computed under the context it actually saw during rollout. For a trajectory τ with T retrievals, the policy-generated content consists of T Evidence Commit Segments and $T+1$ Action Segments. Its log-likelihood factorizes as

$$\begin{aligned}\log P_\theta(\tau) = &\sum_{t=1}^{T} \log P_\theta\left(r_t^{\text{C}}, \Delta E_t \mid C_t^{\text{C}}\right) \\ &+ \sum_{t=1}^{T+1} \log P_\theta\left(r_t^{\text{A}}, a_t \mid C_t^{\text{A}}\right),\end{aligned}$$

where the two segment types receive contexts

$$C_t^{\mathrm{C}} = q \oplus H_t \oplus E_{t-1} \oplus o_t,$$
$$C_t^{\mathrm{A}} = q \oplus H_{t-1} \oplus E_{t-1}.$$

The notation omits the fixed prompt templates for each phase. This factorization includes only probabilities of policy-generated content; retrieved results are conditional inputs supplied by the environment and are not counted as policy-generated tokens.

C Core Algorithms

This section presents the three core procedures of Evidence-R1, from interaction to training-signal assignment. Algorithm 1 specifies the order of action decisions, evidence commits, environment eviction, and context reconstruction in a complete rollout. Algorithm 2 gives the atomic update rules for the versioned evidence state. Algorithm 3 shows how segment credit is assigned along the version lineages of finally cited evidence. Together, they define the interaction protocol, state updates, and optimization procedure.

Algorithm 1 Evidence-Commit Protocol Rollout

Require: Question q, policy π_θ, retriever $\mathcal{R}$, search budget B, state budget T_{budget}
Ensure: Trajectory τ, final evidence state E, provenance store P

$H_0, E_0, P, \tau \leftarrow \varnothing$
for $t = 1, \ldots, B+1$ **do**
  $C_t^{\mathrm{A}} \leftarrow q \oplus H_{t-1} \oplus E_{t-1}$
  Independently sample $(r_t^{\mathrm{A}}, a_t) \sim \pi_\theta(\cdot \mid C_t^{\mathrm{A}}, \mathrm{phase} = \mathrm{A})$ ▷ Force ANSWER if $t = B+1$
  Record $S_t^{\mathrm{A}} = (r_t^{\mathrm{A}}, a_t)$ and C_t^{A} in τ
  if $a_t = \mathrm{ANSWER}$ **then**
    return (τ, E_{t-1}, P)
  end if
  $s_t \leftarrow \textsc{Query}(a_t)$; $H_t \leftarrow H_{t-1} \,\|\, s_t$
  $o_t \leftarrow \textsc{VisibleObservation}(\mathcal{R}(s_t))$
  $C_t^{\mathrm{C}} \leftarrow q \oplus H_t \oplus E_{t-1} \oplus o_t$
  Independently sample $(r_t^{\mathrm{C}}, \Delta E_t) \sim \pi_\theta(\cdot \mid C_t^{\mathrm{C}}, \mathrm{phase} = \mathrm{C})$
  Stop immediately after `</state_update>` ▷ STOP
  $S_t^{\mathrm{C}} \leftarrow (r_t^{\mathrm{C}}, \Delta E_t)$
  $(E_t, P, _) \leftarrow \textsc{AtomicApply}(E_{t-1}, P, \Delta E_t, o_t, S_t^{\mathrm{C}}, T_{\mathrm{budget}})$ ▷ VALIDATE/APPLY
  Record S_t^{C}, C_t^{C}, and the commit outcome in τ
  Evict o_t, r_t^{A}, r_t^{C}, and generation caches ▷ EVICT
  Rebuild the next Action context as $q \oplus H_t \oplus E_t$ ▷ REGENERATE
end for

Algorithm 2 Atomic Versioned Evidence-State Update

Require: State E, store P, operations ΔE, visible sources o, Commit Segment S^{C}, state budget T_{budget}
Ensure: Updated state, provenance store, and validity v

$(\widetilde{E}, \widetilde{P}) \leftarrow \textsc{Copy}(E, P)$ ▷ Including ID allocation state
$A_0 \leftarrow \textsc{ActiveIDs}(E)$
for operation u in ΔE, in order **do**
  if $u = \mathrm{ADD}(z, x)$ **then**
    If $z \notin \textsc{SourceIDs}(o)$ or x is empty, **return** $(E, P, 0)$
    Allocate a fresh ID i and create its initial version in $\widetilde{E}$
  else if $u = \mathrm{UPDATE}(i, z, x)$ **then**
    If $i \notin A_0 \cap \textsc{ActiveIDs}(\widetilde{E})$, $z \notin \textsc{SourceIDs}(o)$, or x is empty, **return** $(E, P, 0)$
    Create a new version of i in $\widetilde{E}$, linked to its previous version
  else if $u = \mathrm{DELETE}(i)$ **then**
    If $i \notin A_0 \cap \textsc{ActiveIDs}(\widetilde{E})$, **return** $(E, P, 0)$
    Remove i from $\widetilde{E}$, preserving its history; **continue**
  else
    return $(E, P, 0)$
  end if
  Append the new version, its lineage link, source z, visible text $o[z]$, statement x, and segment S^{C} to $\widetilde{P}$
end for
If $T_{\mathrm{active}}(\widetilde{E}) > T_{\mathrm{budget}}$, **return** $(E, P, 0)$
return $(\widetilde{E}, \widetilde{P}, 1)$ ▷ Atomic commit; empty updates are valid

Algorithm 3 Provenance-Routed Segment Credit Assignment

Require: Trajectory group $\mathcal{G}_q$ with state and provenance records, reference answer y^*, frozen NLI model ϕ
Require: Threshold θ_{NLI}, capacity cost $\mathcal{C}$, operation cost λ_{op}, local weight w_{local}, $\varepsilon > 0$
Ensure: Segment advantages $A_{\mathrm{total}}(S, \tau)$

for $\tau_k \in \mathcal{G}_q$ **do**
  $F_k \leftarrow R_{\mathrm{F1}}(\tau_k; y^*)$; $V_k \leftarrow V_{\mathrm{format}}(\tau_k)$
  $R_k^{\mathrm{global}} \leftarrow V_k(0.1 + 0.9F_k)$
  Initialize $R_{k,S}^{\mathrm{local}} \leftarrow 0$ for all $S \in \tau_k$
  for each successfully committed segment S_t^{C} **do**
    $R_{k,S_t^{\mathrm{C}}}^{\mathrm{local}} \leftarrow \mathcal{C}(E_{t-1}) - \mathcal{C}(E_t) - \lambda_{\mathrm{op}} n_{\mathrm{operations},t}$
  end for
  $C_k \leftarrow$ the set of active evidence IDs cited by the final answer
  for $i \in C_k$ **do**
    Recover the successful ADD/UPDATE lineage L_i and score each version against its recorded visible source using s_{NLI}
    $L_i^+ \leftarrow \{u \in L_i : s_{\mathrm{NLI}}(u) > \theta_{\mathrm{NLI}}\}$
    If $s_{\mathrm{NLI}}(u_i^{\mathrm{final}}) \le \theta_{\mathrm{NLI}}$, **continue**
    for $u \in L_i^+$ **do**
      $R_{k,\mathrm{seg}(u)}^{\mathrm{local}} \leftarrow R_{k,\mathrm{seg}(u)}^{\mathrm{local}} + \dfrac{F_k V_k s_{\mathrm{NLI}}(u)}{|C_k|\,|L_i^+|}$
    end for
  end for
end for
$\mu_q \leftarrow \mathrm{mean}_k\, R_k^{\mathrm{global}}$; $\sigma_q \leftarrow \mathrm{std}_k\, R_k^{\mathrm{global}}$
for $\tau_k \in \mathcal{G}_q$ **do**
  $A_k^{\mathrm{global}} \leftarrow (R_k^{\mathrm{global}} - \mu_q)/(\sigma_q + \varepsilon)$
  $A_{\mathrm{total}}(S, \tau_k) \leftarrow A_k^{\mathrm{global}} + w_{\mathrm{local}} R_{k,S}^{\mathrm{local}}$ for all $S \in \tau_k$
  Apply each advantage only to its segment's policy tokens under its recorded context
end for
return $\{A_{\mathrm{total}}(S, \tau_k)\}$

D Additional Experimental Details

D.1 EM Normalization and Matching

Before computing EM, we lowercase predicted and all acceptable answers, remove punctuation and English articles, and collapse extra spaces. A prediction is correct if it exactly matches any reference answer after this normalization.

D.2 Retrieval, Generation, and Evidence-State Budgets

The observation budget for each retrieval is 4096 tokens. The maximum generation length is 512 tokens for both Action Segment and Evidence Commit Segment. The cross-turn Evidence State budget is 1024 tokens; any update that would make the serialized state exceed this limit is rejected in full.

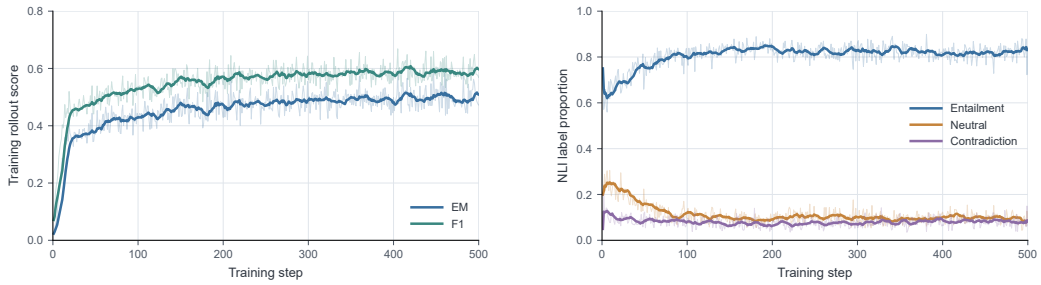
D.3 Reward and Source-Support Settings

The global reward is defined in Section 3, and the local reward is added to the corresponding Evidence Commit Segment with $w_{\text{local}} = 1.0$. Source support is computed by frozen RoBERTa-large-MNLI (Liu et al., 2019), with NLI soft-score threshold $\theta_{\text{NLI}} = 0$; only evidence versions with a positive source-faithfulness score receive source credit. The evidence-count, token-count, and operation-cost coefficients of the compactness reward are $\lambda_n = 0.1$, $\lambda_{\text{tok}} = 0.01$, and $\lambda_{\text{op}} = 0.001$, respectively, discouraging redundant evidence, overly long states, and unproductive repeated edits.

E Additional Evaluation Details

E.1 Training Dynamics

Figure 2 shows rollout answer quality and NLI label proportions during the first 500 recorded steps of a single Qwen2.5-3B-Instruct training run. All curves come from the same run; light lines show raw values at each step, and dark lines show trailing means over at most 11 steps.



(a) Answer quality (EM / F1)

(b) Evidence source support (NLI)

Figure 2: **Training dynamics of Evidence-R1.** Answer quality and NLI label proportions over the first 500 steps of one Qwen2.5-3B-Instruct training run. Light lines show raw values by step, and dark lines show trailing means over at most 11 steps.

Answer Quality. In Figure 2(a), rollout EM and F1 rise rapidly early in training, then improve more slowly with continued fluctuations. Averaged equally by training step, EM and F1 increase from 31.59% and 41.54% over the first 50 steps to 49.66% and 58.68% over the last 50. These observations indicate improving answer quality in sampled trajectories during this run. Table 1 reports test performance; this curve does not estimate stability across random seeds.

Evidence Source Support. Figure 2(b) shows the proportions of NLI entailment, neutral, and contradiction labels in the same training run. These correspond to evidence supported by its source, evidence for which the source is insufficient to decide, and evidence contradicted by the source.

Comparing the first and last 50 steps, the stepwise mean entailment proportion rises from 70.24% to 82.16%, while neutral falls from 20.31% to 9.55% and contradiction from 9.45% to 8.29%. The main shift is from neutral to entailment; the change in contradiction is smaller. Rollout answer metrics and the share of source-supported NLI labels both increase during this run, but their joint movement alone does not establish causality. The effect of local credit assignment must still be identified by comparing B3 and B4. This analysis concerns textual support of evidence statements by their sources, not the objective truth of the source documents.

These training proportions are first aggregated by step and then averaged equally over the selected intervals; they are not an overall support rate computed by pooling all scored evidence.

E.2 Context and Inference Efficiency Metrics

Peak Context is the maximum number of input tokens received by any single generation call in a complete trajectory for one question; for Evidence-R1, this includes both Action and Evidence Commit Segments. Cumulative Tokens sums input and output tokens across all generation calls, including repeated persistent-state inputs and tokens generated during evidence commits. Inference time per question runs from the first policy call to the final answer and includes retrieval, model generation, and state updates. Model and index initialization and training-time NLI reward computation are excluded. Timing uses the same warm-up and concurrency conditions. For any metric m, we first average across questions within each dataset and then take the unweighted mean of the seven dataset values:

$$\overline{m}(B) = \frac{1}{7} \sum_{d=1}^{7} \frac{1}{|\mathcal{D}_d|} \sum_{q \in \mathcal{D}_d} m(q; B),$$

Here $\mathcal{D}_d$ is the evaluation question set for dataset d. Statistics cover every question in each fixed evaluation set, without filtering by answer correctness. Timed-out or otherwise incomplete trajectories must be reported separately with their inclusion rules to avoid bias from counting only successful trajectories.

E.3 Independent LLM Judge and Human Calibration

Evaluation Population and Setup. We use `GLM-5.3-Flash`, provided by Zhipu AI (API model identifier `glm-5.3-flash`), as an independent LLM judge of evidence source faithfulness. The judge does not participate in policy training, source-support reward calculation, or checkpoint selection. For each evidence version produced by a successful ADD or UPDATE, we retrieve the source text actually visible when it was created from the provenance store and pair it with the corresponding evidence statement. The Commit Faithfulness pool contains all 110,855 successfully committed versions; the Cited Faithfulness pool contains the 69,702 versions that remain active at the end of a trajectory and are cited in the final answer. Rejected candidate updates and DELETE operations produce no evaluable statement and are excluded.

The judge receives only the source text and evidence statement, without the method or dataset name, question, final or reference answer, or other evidence. Evaluation uses deterministic decoding at temperature 0, and a fixed parser reads the four-class output. A version whose source is missing or cannot be recovered from the provenance store, or whose output remains unparseable after one format-correction retry, is conservatively labeled Unsupported and retained in the denominator.

Decision Rules and Full Prompts. The judge labels each source–statement pair Fully Supported, Partially Supported, Unsupported, or Contradicted. It assigns Fully Supported only if the source supports every material fact, qualifier, relation, and numerical detail in the statement. The complete evaluation prompts follow; braced fields are serialized from the provenance store at runtime.

Source-Faithfulness Judge: System Prompt

You are a source-support evaluator. Determine whether the SOURCE supports the EVIDENCE_STATEMENT. Use only the supplied SOURCE; do not use outside knowledge, the question, or a final answer.

Assign exactly one label:
- FULLY_SUPPORTED: every material factual claim in the statement is supported by the source, with no unsupported qualifier, relation, or numerical detail.
- PARTIALLY_SUPPORTED: the source supports at least one material factual claim, but another material part is not sufficiently supported.
- UNSUPPORTED: the source does not provide enough information to support the statement and does not clearly contradict it.
- CONTRADICTED: the source explicitly conflicts with at least one material factual claim in the statement.

If the source is empty, missing, or unreadable, choose UNSUPPORTED. Treat the source as the object being checked, not as guaranteed world truth. Return exactly one JSON object with keys "label" and "rationale". The rationale must be one concise sentence grounded in the supplied source.

Source-Faithfulness Judge: Serialized Input

<source>
{source_text_visible_at_commit_time}
</source>

<evidence_statement>
{evidence_version_text}
</evidence_statement>

Metric Computation. Let V_d^{commit} and V_d^{cited} be the Commit Faithfulness and Cited Faithfulness evaluation pools for dataset d, respectively, and let $I_{\text{full}}(v)$ indicate whether version v is labeled FULLY SUPPORTED. For $x \in \{\text{commit}, \text{cited}\}$, source faithfulness is

$$F_d^x = \frac{1}{|V_d^x|} \sum_{v \in V_d^x} I_{\text{full}}(v), \qquad \overline{F}^x = \frac{1}{7} \sum_{d=1}^{7} F_d^x.$$

Thus, Avg. in Table 3 is the unweighted mean of seven dataset proportions, and Total is the corresponding number of evaluated versions. The evaluation does not weight datasets by question count or select trajectories based on final answer correctness.

Human Calibration. To assess the reliability of the automated judge, we stratify-sample 250 source–statement pairs from each of the commit and cited pools, for 500 total, covering all four automatic labels. Versions included in the cited subsample are excluded from the commit subsample. Human annotators see only the source text and evidence statement; the automatic label, method, dataset, question, and final answer remain hidden. Disagreements are independently adjudicated to form human reference labels. The LLM judge exactly agrees with the human reference on 454/500 pairs, a strict four-class agreement rate of 90.8%.

F ADDITIONAL ABLATION DETAILS

This section provides implementation details, metric aggregation rules, and complete numerical results for the ablations in Section 4.3. Table 4 defines configurations B1–B4. Using the same numbering, we describe generation and training under Soft Commit, the removal of Action history, and the ablation of provenance credit.

F.1 Statistics and Metric Aggregation

All ablations initialize the policy from Qwen2.5-3B-Instruct and train on NQ and HotpotQA, using the same training data as the main experiments. The retriever is the same E5-base-v2 model with the same wiki-18 index; each search returns the top five documents, with at most five searches, and the cross-turn Evidence State capacity is 1024 tokens. All configurations use the same evidence-state representation and capacity budget, training budget, checkpoint-selection rules, and evaluation protocol. Each is run independently with the same set of five random seeds; matched seeds use the same parameter initialization. NQ and HotpotQA are evaluated and summarized separately. For each seed, dataset-level metrics are first computed on a fixed evaluation set; the tables report the mean over five independent runs. Each Δ is the difference between the B4 mean and the corresponding ablation mean. Differences in percentage metrics are in percentage points; other metrics retain their original units. For metrics where lower is better, a negative difference indicates improvement.

Answer Quality and Search Cost. Answer EM uses the answer normalization and matching rules in the main text. Answer F1 measures token-level overlap between the predicted and reference answers. Peak Context and Cumulative Tokens use the definitions in Appendix E.2: the maximum input length of a generation segment in each trajectory and the total input and output tokens over all generation segments, respectively, averaged by question within each dataset. Avg. Searches is the dataset mean number of retrieval calls actually triggered per question.

Semantic Duplicate Query Rate. Semantic Duplicate measures the proportion of queries in a search trajectory that differ as strings from an earlier query for the same question but have a highly similar retrieval intent. We use a frozen `intfloat/e5-base-v2` query encoder, consistent with the retriever encoder. Each query receives the prefix `query: ` ; we mean-pool last-layer hidden states over non-padding tokens and L2-normalize the resulting embedding. The similarity threshold is $\theta_{\mathrm{dup}} = 0.9$ across all configurations, datasets, and random seeds.

Suppose question q triggers T_q searches, and let $d_t \in \{0, 1\}$ indicate whether query t is a semantic duplicate under this rule. Its duplicate rate is

$$\mathrm{SD}(q) = \begin{cases} \dfrac{100}{T_q} \displaystyle\sum_{t=2}^{T_q} d_t, & T_q \geq 2, \\ 0, & T_q < 2. \end{cases}$$

The denominator includes all searches actually performed for that question. For each seed, rates are first averaged equally across questions in a dataset, then averaged across the five seeds.

Evidence Source Faithfulness. Commit and Cited Faithfulness use the evaluation populations, independent LLM judge prompts, four-class decision rules, and error handling from Appendix E.3. Within each seed, we divide the number of versions labeled FULLY SUPPORTED on the current dataset by the total evaluated versions. The tables report the mean of these five proportions rather than pooling versions across seeds into one ratio. The n_{commit} and n_{cited} columns sum the corresponding evaluated versions across five runs; they are neither question counts nor numbers of independent runs. Neither evaluation pool filters trajectories by final answer correctness. Versions with missing or unrecoverable sources, or still-unparseable output after one format-correction retry, are counted as UNSUPPORTED and remain in the denominator.

F.2 B1 versus B4: Commit-Protocol Implementation

This section details the input, update validation, and training-token mask for Soft Commit. Apart from the commit protocol, the two configurations use the same evidence state, query history, update operations, and rewards, with the same observation and total generation-token budgets for each logical search turn.

Soft Commit: Continuous Generation and Update Validation. Following the indexing convention of Section 3.1, the current query s_t has already been appended to history when retrieval returns:

$H_t = H_{t-1} \parallel s_t$. B1 uses the following input and joint output:

$$C_t^{\text{soft}} = q \oplus H_t \oplus E_{t-1} \oplus o_t,$$
$$\left(r_t^{\text{C}}, \Delta E_t, r_{t+1}^{\text{A}}, a_{t+1}\right) \sim \pi_\theta\left(\cdot \mid C_t^{\text{soft}}\right).$$

Here, </state_update> only separates output fields; it does not trigger a stop or context reconstruction. The next action can therefore access both the generated update text ΔE_t and the current raw observation o_t, although the update has not yet been committed by the environment. After the full joint segment returns, the environment validates and atomically applies ΔE_t. An illegal update invalidates both the update and its subsequent action.

B1's format checker handles joint segments, requiring one state update and one subsequent action per segment while retaining B4's update-legality constraints. The global trajectory advantage applies to all policy tokens in the joint segment. A token mask restricts the local training signal to commit reasoning and the token span of state_update, excluding tokens of the subsequent action.

Hard Commit: Environment Boundary. B4 follows the STOP–Apply–Evict–Regenerate protocol in Section 3.1. It stops after generating the state update; the environment applies the update, removes the raw observation and temporary reasoning, and starts next-action generation from $q \oplus H_t \oplus E_t$. Each segment's training likelihood is computed under its actual generation context, as factorized in Appendix B.3. The B1–B4 comparison therefore includes changes to the generation boundary, timing of environmental commitment, and context reconstruction as a whole; it cannot isolate any one of those operations.

Table 6 gives the complete answer-quality and token-cost results discussed in the main text.

Table 6: **Soft versus Hard Commit on NQ and HotpotQA.** Datasets are summarized separately. Values are means of five independent runs, and Δ is B4−B1.

Dataset	Method	Answer EM ↑	Answer F1 ↑	Peak Context ↓	Cum. Tokens ↓
NQ	B1: Soft Commit	43.5	52.1	1,238	3,111
	B4: Hard Commit Full	45.5	54.7	1,149	2,781
	Δ	**+2.0**	**+2.6**	**−89**	**−330**
HotpotQA	B1: Soft Commit	40.3	51.8	1,685	4,904
	B4: Hard Commit Full	43.3	55.3	1,233	3,737
	Δ	**+3.0**	**+3.5**	**−452**	**−1,167**

F.3 B2 versus B4: Removing Action History

The two configurations share the model, data, Hard Commit protocol, rewards, and evaluation setting. B2's environment still maintains the complete query history and supplies it to Evidence Commit Segment; it omits this field only when constructing Action Segment input. Their input structures are

$$\text{B2}: \quad C_t^{\text{C}} = q \oplus H_t \oplus E_{t-1} \oplus o_t, \quad C_{t+1}^{\text{A}} = q \oplus E_t,$$
$$\text{B4}: \quad C_t^{\text{C}} = q \oplus H_t \oplus E_{t-1} \oplus o_t, \quad C_{t+1}^{\text{A}} = q \oplus H_t \oplus E_t.$$

Here $H_t = (s_1, \ldots, s_t)$ records only executed queries, not historical reasoning, raw observations, retrieval summaries, or evidence entries. The first Action Segment starts with empty history and evidence state in both configurations. After each later COMMIT, the environment removes the raw observation and temporary reasoning and reconstructs Action input according to the corresponding structure.

This comparison fixes input fields and their visibility; it does not require the two configurations to generate identical queries, evidence states, or retrieval trajectories. Because COMMIT in B2 still sees history, the ablation tests only the incremental value of supplying query history directly to Action and cannot establish whether the whole system needs history. Table 7 gives the complete answer-quality, semantic-duplicate, and search-count results.

Table 7: **Ablating Action access to query-only Search History on NQ and HotpotQA.** COMMIT retains history in both B2 and B4; only Action in B2 lacks historical queries. Datasets are summarized separately. Values are means of five independent runs, and Δ is B4−B2.

Dataset	Method	Answer EM ↑	Answer F1 ↑	Semantic Duplicate ↓	Avg. Searches ↓
NQ	B2: No Action History	45.3	54.4	12.3	2.20
	B4: Query-only History	45.5	54.7	9.1	1.57
	Δ	**+0.2**	**+0.3**	**−3.2**	**−0.63**
HotpotQA	B2: No Action History	40.7	52.2	20.0	3.42
	B4: Query-only History	43.3	55.3	12.2	2.00
	Δ	**+2.6**	**+3.1**	**−7.8**	**−1.42**

F.4 B3 versus B4: Local-Credit Ablation

B3 removes provenance credit only and retains the state-shaping reward. Using the local-advantage notation of Section 3.2, the two configurations differ as follows:

$$A^{\mathrm{B3}}_{\mathrm{local}}(S_m,\tau) = \mathbf{1}[S_m \text{ is a COMMIT segment}]R_{\mathrm{shape}}(S_m,\tau),$$
$$A^{\mathrm{B4}}_{\mathrm{local}}(S_m,\tau) = \mathbf{1}[S_m \text{ is a COMMIT segment}]\big(R_{\mathrm{shape}}(S_m,\tau) + R_{\mathrm{prov}}(S_m,\tau)\big).$$

For $k \in \{\mathrm{B3}, \mathrm{B4}\}$, both use training advantage

$$A^{k}_{\mathrm{total}}(S_m,\tau) = A_{\mathrm{global}}(\tau) + w_{\mathrm{local}} A^{k}_{\mathrm{local}}(S_m,\tau).$$

The interaction protocol, history access, evidence-citation requirement, global trajectory reward, and local reward weight are the same in both configurations. Both use $w_{\mathrm{local}} = 1$; the local term is not separately normalized within groups and applies only to policy tokens in the corresponding Evidence Commit Segment. The state-shaping reward is defined in Appendix B.2; λ_n, λ_{tok}, and λ_{op} are held fixed. B3 therefore does not remove all local training signal.

B4 follows Section 3.2 for version-lineage tracing, source-support gating, and credit aggregation. B3 retains the same evidence state, update protocol, and final citation requirement, but assigns no additional credit from version lineages or source-support scores. The independent LLM judge used for evaluation (`GLM-5.3-Flash`) and its decision rules are identical for both configurations; it does not contribute to training rewards or checkpoint selection.

Table 8 provides the complete means and evaluated version counts. Both faithfulness metrics measure only whether source text supports the evidence statement. They do not directly measure relevance to the question, sufficiency for the answer, or evidence coverage, and do not verify the source's objective truth.

Table 8: **Ablation of Provenance-Routed Segment Credit on NQ and HotpotQA.** Datasets are summarized separately. Except for count columns, values are means of five independent runs, and Δ is B4−B3. n_{commit} and n_{cited} sum evaluated versions across five runs.

Dataset	Method	Answer EM ↑	Answer F1 ↑	n_{commit}	Commit Faith. ↑	n_{cited}	Cited Faith. ↑
NQ	B3: No Provenance Credit	44.2	53.4	4,258	72.10	3,586	71.80
	B4: Provenance Credit	45.5	54.7	5,758	79.40	4,497	77.70
	Δ	**+1.3**	**+1.3**	**–**	**+7.30**	**–**	**+5.90**
HotpotQA	B3: No Provenance Credit	42.7	54.9	27,739	71.60	16,713	70.90
	B4: Provenance Credit	43.3	55.3	16,368	82.40	10,755	81.40
	Δ	**+0.6**	**+0.4**	**–**	**+10.80**	**–**	**+10.50**

G PROMPT TEMPLATES

This section gives the complete prompt templates used by the two generation-segment types during training and inference. Each subsection shows the corresponding system prompt and user input template. The question, search history, evidence state, retrieved documents, and other fields in the latter are serialized by the environment at runtime.

G.1 ACTION SEGMENT

This subsection presents the Action Segment prompt template used to choose between another search and a final answer at each turn. The phase receives only the question, query-only Search History, and current active Evidence State, without previous raw retrieved documents or temporary reasoning. The model must output nonempty reasoning followed by exactly one `<search>` or `<answer>`, and an answer may cite only current active evidence IDs. The template limits action generation to information visible after commitment, implementing commit-before-use at the prompt level.

System Prompt

You are an Evidence-R1 search agent. Follow the protocol exactly.

You receive a question, a search-only history, and an active evidence state. Previous retrieval documents and previous temporary thoughts are intentionally unavailable. Facts from retrieval may influence this action only through the active evidence state.

First generate exactly one non-empty <think>...</think>. Then generate exactly one action:
- <search>one non-empty search query</search> when more information is needed; or
- <answer evidence="E1,E2">short final answer</answer> when the evidence state supports an answer.

Do not emit any other text. Cite only active evidence IDs. If the evidence state is non-empty, a final answer must cite at least one active evidence ID.

Serialized User Input

```
<question>
{question}
</question>

<search_history>
  <search>{query_1}</search>
  <search>{query_2}</search>
</search_history>

<evidence_state>
  <evidence id="E1" source="T1D1">
  {evidence_text}
  </evidence>
</evidence_state>
```

G.2 Evidence Commit Segment

This subsection presents the Evidence Commit Segment prompt template used to select information from the current observation and revise the evidence state after retrieval. This phase can additionally access documents retrieved for the current query, but may output only one `<state_update>` containing ADD, UPDATE, DELETE, or a legal empty update. The template also constrains source references, target versions, and stopping position, and forbids search or answer generation in this phase. It thus directs the model to explicitly commit task-relevant content from the current observation to Evidence State.

System Prompt

You are the Evidence-R1 evidence updater. Follow the protocol exactly.

You receive a question, search-only history including the current query, the previous active evidence state, and only the current retrieval documents. First generate exactly one non-empty <think>...</think> explaining evidence selection/revision. Then generate exactly one <state_update>...</state_update> and stop.

Allowed operations are:
<add source="T1D1">one concise source-supported fact</add>
<update target="E1" source="T2D1">a corrected or more precise fact</update>
<delete target="E1"/>

The environment assigns ADD IDs in operation order; never write an ID on ADD. UPDATE/DELETE may target only evidence that was active at the start of this commit (not evidence added earlier in the same update). ADD/UPDATE sources must be listed retrieval documents. Keep each evidence atomic, concise, task-relevant, and supported by its source. An empty <state_update></state_update> is allowed when no document provides useful evidence. Do not search or answer in this phase and do not emit any other text.

Serialized User Input

<question>
{question}
</question>

<search_history>
<search>{query_1}</search>
<search>{query_2}</search>
</search_history>

<evidence_state>
<evidence id="E1" source="T1D1">
{evidence_text}
</evidence>
</evidence_state>

<information>
[T1D1]
{retrieved_document_1}

[T1D2]
{retrieved_document_2}
</information>

H CASE STUDY

Evidence-R1 Reasoning Example

Question: Who wrote the play in which the actress who portrayed old Murph in *Interstellar* made her Broadway debut?

Search Round 1

Action Segment

<think>I first need to identify the actress who portrayed old Murph in *Interstellar*.</think>
<search>Interstellar Old Murph actress</search>

Evidence Commit Segment

<information>
[T1D1] Title: Interstellar. The film stars Matthew McConaughey, Anne Hathaway, Jessica Chastain and Michael Caine.
[T1D2] Title: Interstellar Full Cast. Jessica Chastain played the adult Murph; Mackenzie Foy played the young Murph; Ellen Burstyn played the old Murph.
[T1D3] Title: Ellen Burstyn. Ellen Burstyn is an American actress who has received numerous awards during her career.
</information>
<think>T1D2 explicitly states that Ellen Burstyn portrayed old Murph; T1D1 and T1D3 do not support this intermediate fact.</think>
<state_update>
<add source="T1D2">Ellen Burstyn played Old Murph in Interstellar.</add>
</state_update>

Search Round 2

Action Segment

<think>E1 identifies Ellen Burstyn as the target actress. I next need to find her Broadway debut play and its playwright.</think>
<search>Ellen Burstyn Broadway debut play playwright</search>

Evidence Commit Segment

<information>
[T2D1] Title: Ellen Burstyn Theatre Credits. Burstyn made her Broadway debut in 1957 in the comedy Fair Game.
[T2D2] Title: Fair Game. Fair Game was a 1957 Broadway comedy written by Sam Locke.
[T2D3] Title: Ellen Burstyn Awards. Burstyn won a Tony Award for Same Time, Next Year.
</information>
<think>T2D1 identifies Ellen Burstyn's Broadway debut play, and T2D2 identifies its playwright; T2D3 is irrelevant to the question.</think>
<state_update>
<add source="T2D1">Ellen Burstyn made her Broadway debut in Fair Game.</add>
<add source="T2D2">Fair Game was written by Sam Locke.</add>
</state_update>

Final Action Segment

<think>E1 identifies Ellen Burstyn as the target actress. E2 shows that she made her Broadway debut in *Fair Game*, and E3 shows that Sam Locke wrote the play. Therefore, the answer is Sam Locke.</think>
<answer evidence="E1,E2,E3"> **Sam Locke** </answer>