# EVIDENCEGATE: EVIDENCE-GUIDED COMMIT-OR-REPAIR FOR FIRST-HOP ANSWER BINDING IN MULTI-HOP QUESTION ANSWERING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reinforcement learning has recently shown promise in improving search-augmented reasoning. However, existing methods for open-domain multi-hop question answering (QA) may prematurely commit to a first-hop answer based on relevant yet insufficient evidence, causing errors to propagate through subsequent queries. To address this issue, we introduce EvidenceGate, an evidence-guided Commit/Repair framework that formulates first-hop answer commitment as a decision between committing the current candidate and performing additional retrieval for repair. To support this decision, we use a controllable search simulator to generate evidence with varying levels of support and train an evidence verifier to assess whether retrieved evidence supports a candidate answer. The verifier resolves clear evidence states, while a Commit/Repair value model handles ambiguous verifier rejections by predicting the cost-adjusted repair advantage from rollout-level final QA rewards. Extensive experiments on five multi-hop QA benchmarks demonstrate that EvidenceGate achieves state-of-the-art overall performance.

## 1 INTRODUCTION

Retrieval-augmented generation (RAG) enables large language models to answer knowledge-intensive questions by accessing external information beyond their parametric memory (Guu et al., 2020; Karpukhin et al., 2020; Lewis et al., 2020; Izacard & Grave, 2021). Recent reinforcement-learning-based search-augmented methods further advance this paradigm by training models to generate search queries, use retrieved evidence, and perform multi-step reasoning in interactive retrieval environments, substantially improving open-domain question answering performance (Jin et al., 2025; Sun et al., 2025; Zheng et al., 2025; Luo et al., 2026). These studies demonstrate that reinforcement learning can effectively enhance autonomous search and multi-hop reasoning.

Despite this progress, existing search-augmented reasoning methods remain vulnerable to an under-explored failure mode: premature commitment to an intermediate answer. In open-domain multi-hop question answering, the answer produced at an early hop is often reused as a variable in later search queries. If the model prematurely commits to an insufficiently supported first-hop candidate, subsequent retrieval is then conditioned on the wrong entity or relation. As a result, the reasoning trajectory may continue along an erroneous path even when later retrieved documents are individually relevant. We focus on this earliest binding point because it provides a high-leverage, bounded-cost intervention: the first-hop answer can influence the longest remaining portion of the reasoning trajectory, whereas applying verification at every hop would introduce overhead that grows with trajectory length. We refer to this phenomenon as *premature first-hop answer binding*. Fine-grained trajectory analysis shows that more than 28% of analyzed trajectories involve premature first-hop binding (details in Appendix B.1).

A natural way to mitigate this problem is to verify whether the retrieved evidence supports the current candidate before it is written into the reasoning state. However, treating this step only as an evidence classification problem is insufficient. A verifier score reflects the local relationship between a candidate answer and the currently retrieved documents, but the decision to repair also depends on downstream utility: whether additional retrieval is likely to improve the final answer enough to justify its cost. Therefore, first-hop answer binding should be modeled not merely as a local verification



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.

task, but as a task-oriented decision problem that balances commitment and repair under final-answer supervision.

Motivated by this observation, we propose **EvidenceGate**, an evidence-guided Commit/Repair framework for open-domain multi-hop question answering. EvidenceGate treats the first-hop answer as tentative before commitment. It first uses a controllable search simulator to construct evidence with varying support levels, including fully supportive, partially relevant, and misleading evidence, and trains an evidence verifier to assess whether retrieved documents support a candidate answer. Based on the verifier representations, a Commit/Repair value model is trained from paired Commit/Repair rollout rewards to predict repair advantage and decide whether to COMMIT the current candidate or trigger REPAIR through additional focused retrieval. When repair is selected, the model retrieves new evidence without conditioning on the unverified candidate, thereby reducing the risk of propagating an incorrect first-hop entity into later queries.

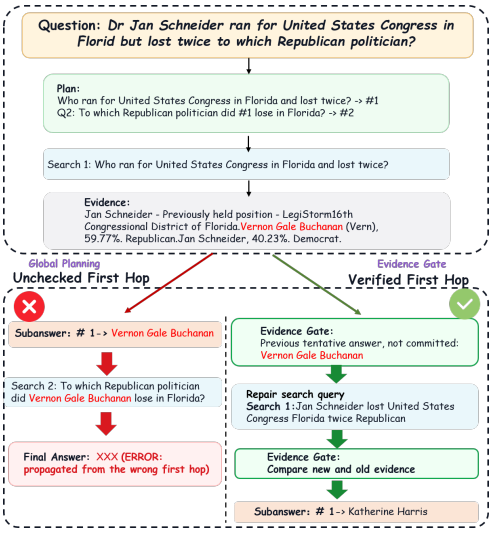



Figure 1: Illustration of premature first-hop answer binding and EvidenceGate. Without evidence verification, an insufficiently supported candidate may be committed and propagated through subsequent queries. EvidenceGate instead treats the first-hop candidate as tentative, assesses whether the retrieved evidence supports it, and selectively triggers repair retrieval before commitment.

This design converts first-hop answer binding from a one-shot generation step into a verifiable and repairable decision. By connecting local evidence assessment with final QA outcomes, EvidenceGate can selectively allocate repair computation to high-risk states rather than applying unconditional re-retrieval. Experiments on five multi-hop QA benchmarks show that EvidenceGate consistently improves over strong search-augmented reasoning baselines.

Our contributions are summarized as follows:

- We identify premature first-hop answer binding as a key failure mode in search-augmented multi-hop question answering, where an insufficiently supported intermediate answer can propagate errors through subsequent retrieval steps.
- We construct a controllable simulated retrieval setting that generates first-hop evidence with different support levels, enabling the training of an evidence verifier for candidate-level support assessment.
- We formulate first-hop binding as a Commit/Repair decision and train a Commit/Repair value model with final QA rewards to predict repair advantage and determine whether direct commitment or repair retrieval is more beneficial.

## 2 RELATED WORK

**Search-Augmented Multi-Hop Reasoning.** Retrieval-augmented generation grounds language-model outputs in external evidence (Karpukhin et al., 2020; Lewis et al., 2020; Izacard et al., 2023). Iterative and adaptive methods coordinate retrieval with reasoning or generation (Yao et al., 2023b; Trivedi et al., 2023; Shao et al., 2023; Jiang et al., 2023; Jeong et al., 2024), while Search-o1 (Li et al., 2025) integrates iterative search into long-form reasoning. Recent reinforcement-learning methods further optimize retrieval and reasoning behaviors through task rewards. Search-R1 (Jin et al., 2025) trains models to interact with search engines using final-answer rewards, StepSearch (Zheng et al., 2025) introduces step-wise optimization for search behavior, ZeroSearch (Sun et al., 2025) employs a controllable simulated search engine, and GlobalRAG (Luo et al., 2026) strengthens global planning and subgoal execution. Despite these advances, existing methods primarily optimize how evidence is acquired and used, while typically treating generated intermediate answers as ordinary reasoning

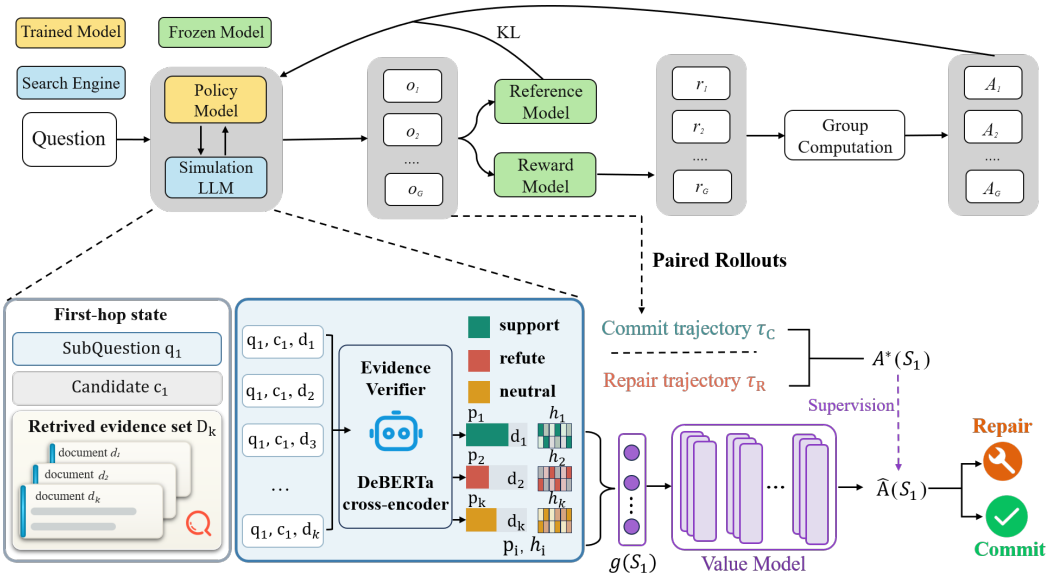



Figure 2: Overview of EvidenceGate. The base search policy is first optimized with GRPO in a simulated retrieval environment. The evidence verifier then provides first-hop verification signals for further policy optimization with final QA and repair-behavior rewards. Paired rollouts generated by the refined policy supervise Commit/Repair value learning.

states. In contrast, we introduce an evidence-guided intervention at the first-hop binding point to prevent insufficiently supported candidates from being propagated through subsequent queries.

**Evidence Verification and Corrective Retrieval.** Verification, revision, and reflection mechanisms have been introduced to improve generation reliability (Gao et al., 2023; Dhuliawala et al., 2024; Shinn et al., 2023; Gou et al., 2024; He et al., 2024). Self-RAG (Asai et al., 2024) uses reflection tokens to assess retrieved content and generated outputs, whereas CRAG (Yan et al., 2024) evaluates retrieval quality and triggers corrective retrieval when necessary. These approaches primarily assess the relevance or quality of retrieved documents, or the reliability of a generated response as a whole. However, in multi-hop reasoning, a document may be relevant to the current subquestion without providing sufficient evidence for the specific candidate generated by the policy. Unlike these approaches, our verifier directly assesses whether the retrieved evidence supports the policy-generated first-hop candidate before it is committed.

**Value-Guided Reasoning.** Process supervision and value-guided methods evaluate intermediate reasoning steps or states to guide subsequent generation and exploration (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024; Yao et al., 2023a). Process reward models provide step-level correctness signals, while value-guided search uses state evaluations to prioritize promising reasoning trajectories. These methods generally estimate the overall quality or correctness of an intermediate state rather than the relative utility of a specific intervention. We instead formulate first-hop answer binding as a choice between committing the current candidate and acquiring additional evidence. By contrast, the Commit/Repair value model predicts the cost-adjusted repair advantage, allowing additional retrieval to be selected according to its expected downstream utility.

## 3 Method

### 3.1 Retrieval-Augmented GRPO Base Framework

Figure 2 summarizes the staged training procedure of EvidenceGate. We begin by describing the retrieval-augmented GRPO framework used to optimize the base search policy before introducing the first-hop evidence gate.

Given a question $x \sim \mathcal{D}$, the search policy $\pi_\theta$ interacts with a parameter-frozen simulated search model $\pi_\psi$ to generate a complete search-reasoning trajectory:

$$\tau \sim \pi_\theta(\cdot \mid x; \pi_\psi). \tag{1}$$

The search policy generates reasoning steps, search queries, and the final answer, while the simulated search model returns query-conditioned documents. We optimize the base policy with GRPO using final-answer rewards and KL regularization (Shao et al., 2024). Retrieved documents are treated as policy context, and their tokens are masked in the policy loss. The complete objective and group-normalized advantage are provided in Appendix A.3.

### 3.2 Multi-Hop Reasoning and First-Hop Binding

Given a question $x$, the search policy first constructs an ordered decomposition containing $K$ mutually dependent subquestions:

$$\mathcal{P}(x) = [(q_k, \#k)]_{k=1}^{K}, \tag{2}$$

where $q_k$ denotes the $k$-th subquestion and $\#k$ denotes its intermediate-answer slot. The policy sequentially retrieves evidence and answers each subquestion, binding the resulting answer $c_k$ to $\#k$ for use in subsequent reasoning.

For the first hop, the policy retrieves a document set

$$D_1 = \{d_i\}_{i=1}^{m} \tag{3}$$

and generates a candidate intermediate answer $c_1$. Because $c_1$ is subsequently incorporated into later queries, an incorrect first-hop binding can redirect the remaining search process. Before committing the candidate, we therefore construct the decision state

$$s_1 = (q_1, c_1, D_1), \tag{4}$$

and introduce two possible actions:

$$a \in \{\text{Commit}, \text{Repair}\}. \tag{5}$$

Commit directly binds $\#1 \leftarrow c_1$, whereas Repair postpones the binding and performs an additional retrieval.

### 3.3 Controllable Evidence Simulation

To cover representative first-hop evidence conditions, the simulated search model assigns each generated document $d_i$ a hidden evidence role $\rho_i$:

**Support**: documents containing the key facts required to answer the subquestion. **Partial**: documents related to the subquestion but missing at least one necessary constraint. **Wrong**: documents containing misleading entities or conflicting information.

The gold answer is used only to control the content and role composition of the simulated evidence. It is not provided to either the search policy or the evidence verifier.

Bundles that are malformed, incomplete, duplicate, highly similar, or inconsistent with the requested role composition are excluded from the labeled verifier data. Within each retained bundle, documents are randomly shuffled to prevent positional shortcuts.

The hidden role $\rho_i$ provides a simulator-side prior about the intended evidence condition. Following distant-supervision practices in open-domain QA (Chen et al., 2017; Karpukhin et al., 2020), we use normalized lexical matching to associate each generated document with the first-hop candidate $c_1$. We then combine this association with the simulator-provided role to construct task-specific weak-supervision labels. Let $M(c_1, d_i) \in \{0, 1\}$ indicate whether $c_1$ matches $d_i$. The candidate-level evidence label is defined as

$$y_i = \begin{cases} \text{support}, & \rho_i = \text{Support} \wedge M(c_1, d_i) = 1, \\ \text{refute}, & \rho_i = \text{Wrong} \wedge M(c_1, d_i) = 1, \\ \text{neutral}, & \text{otherwise.} \end{cases} \tag{6}$$

Partial documents are labeled as neutral because topical relevance alone is insufficient to justify safe intermediate-answer binding. The support and refute labels in Eq. (6) are operational pseudo-labels rather than guaranteed semantic entailment and contradiction labels. In particular, refute denotes candidate-associated evidence generated under the Wrong role; its agreement with semantic contradiction is assessed through the LLM-assisted human audit in Appendix A.4. The matching function is used only to construct candidate-level supervision. Its implementation and the evidence-generation configurations are also detailed in Appendix A.4.

### 3.4 Evidence Verifier

The evidence verifier employs a DeBERTa cross-encoder (He et al., 2021) to jointly encode the first-hop subquestion, candidate answer, and each retrieved document:

$$(h_i, z_i) = f_\omega(q_1, c_1, d_i), \qquad p_i = \mathrm{softmax}(z_i), \tag{7}$$

where $h_i$ is the final-layer [CLS] representation and $p_i \in \mathbb{R}^3$ denotes the probabilities of support, refute, and neutral.

The verifier is trained using candidate-level evidence labels:

$$\mathcal{L}_{\mathrm{evi}} = -\frac{1}{N}\sum_{i=1}^{N} \log p_i(y_i). \tag{8}$$

where $N$ is the number of document-candidate training instances.

The verifier acts as the primary evidence-sufficiency gate. We aggregate document-level verifier predictions into an evidence-gate score and commit when sufficient support is detected; verifier-rejected states are repaired directly or routed to the Commit/Repair value model when the gate decision is ambiguous. The complete decision rule is provided in Appendix A.3. Its outputs characterize the evidence state for repair-advantage estimation and provide first-hop verification signals during policy refinement. During refinement, GRPO combines the final QA reward with a repair-behavior reward:

$$R_{\mathrm{EG}}(\tau) = R_{\mathrm{QA}}(\tau) + \gamma R_{\mathrm{repair}}(\tau), \tag{9}$$

where $R_{\mathrm{repair}}(\tau)$ rewards the completion of verifier-triggered repair search and rebinding of the first-hop answer slot, penalizes final answers emitted before repair is complete, and $\gamma$ controls its contribution. Its implementation is detailed in Appendix A.3. During this refinement, the GRPO advantage is computed using $R_{\mathrm{EG}}$.

### 3.5 Commit/Repair Value Learning

Since the ordering of retrieved documents should not affect the decision, we apply mean and max pooling to both the document representations and evidence probabilities:

$$g_\omega(s_1) = \left[\mathrm{Mean}_i(h_i); \mathrm{Max}_i(h_i); \mathrm{Mean}_i(p_i); \mathrm{Max}_i(p_i)\right]. \tag{10}$$

To construct supervision for value learning, the rollout controller branches from the same first-hop state and fixes one of two environment transitions:

$$\tau_C \sim \mathrm{Rollout}(\pi_\theta; s_1, \mathrm{Commit}), \qquad \tau_R \sim \mathrm{Rollout}(\pi_\theta; s_1, \mathrm{Repair}). \tag{11}$$

Commit and Repair are implemented as controller interventions, while the underlying policy remains unchanged. The Commit branch immediately binds the current candidate, whereas the Repair branch performs one additional retrieval before regenerating the first-hop answer. Each branch then continues until a final answer is produced.

We define the observed repair advantage as

$$A^*(s_1) = R_{\mathrm{QA}}(\tau_R) - \lambda C_R - R_{\mathrm{QA}}(\tau_C), \tag{12}$$

where $R_{\mathrm{QA}}$ denotes the final QA reward, $C_R$ is a unit action cost assigned to the single additional retrieval in the Repair rollout, taking the value 1 when the repair retrieval is executed and 0 otherwise, and $\lambda$ controls the contribution of this retrieval cost to the repair advantage. This unit action cost does not represent actual token usage, API charges, or runtime latency. A positive $A^*(s_1)$ indicates that Repair produces greater task utility than Commit after accounting for its additional cost.

The paired rollouts are used only to construct training targets and are not generated at inference time.

The Commit/Repair value model receives only the pooled verifier representation and predicts the repair advantage:

$$\widehat{A}_{\omega,\phi}(s_1) = V_\phi(g_\omega(s_1)) = \mathrm{MLP}_\phi(g_\omega(s_1)). \tag{13}$$

The trajectory text is not provided to the Commit/Repair value model. Additional rollout-construction details are provided in Appendix A.5.

We first regress the magnitude of the observed repair advantage using a smooth $L_1$ loss:

$$\mathcal{L}_{\mathrm{mag}} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{SmoothL1}\left(\widehat{A}_b, A_b^*\right). \tag{14}$$

Because inference depends on whether Repair is more valuable than Commit, we additionally introduce a decision loss that directly supervises the Commit/Repair choice:

$$\mathcal{L}_{\mathrm{dec}} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{BCEWithLogits}\left(\frac{\widehat{A}_b}{T}, \mathbb{I}[A_b^* > 0]\right), \tag{15}$$

where $T$ is a temperature parameter. The value-learning objective is

$$\mathcal{L}_{\mathrm{value}} = \mathcal{L}_{\mathrm{mag}} + \eta\mathcal{L}_{\mathrm{dec}}, \tag{16}$$

where $\eta$ controls the decision-loss contribution. The SmoothL1 term fits the rollout-derived repair advantage, while the auxiliary decision loss supervises its sign.

### 3.6 Inference and Repair

During inference, the verifier first resolves clear evidence states, while ambiguous verifier-rejected states are routed to the trained Commit/Repair value model. For each routed state, the value model predicts the cost-adjusted repair advantage and selects the action as follows:

$$a^* = \begin{cases} \mathrm{Repair}, & \widehat{A}_{\omega,\phi}(s_1) > 0, \\ \mathrm{Commit}, & \text{otherwise}. \end{cases} \tag{17}$$

Verifier-supported states are committed directly, whereas clear unrouted rejections trigger Repair. When Commit is selected, the candidate $c_1$ is bound to $\#1$, and the search policy continues with the remaining subquestions. When Repair is selected, the current candidate is not committed. The policy constructs a repair query using the original question and the first-hop subquestion and performs one additional retrieval. The unverified candidate is excluded from the repair query to prevent potentially incorrect entities from being propagated into the subsequent search process.

After obtaining repair evidence, the policy jointly considers the original and newly retrieved documents, regenerates the first-hop answer, and resumes multi-hop reasoning. This transforms first-hop binding into an evidence-guided, repairable decision while leaving intermediate-answer generation to the search policy.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We evaluate EvidenceGate on five open-domain multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023), and WikiHop (Welbl et al., 2018). Dataset construction and evaluation protocols are detailed in Appendix A.1.

**Baselines.** We compare with non-retrieval reasoning methods (Zero-Shot and R1), retrieval-augmented methods (NaiveRAG, IRCoT, RA-Agent, and Search-o1), and RL-based search methods (Search-R1, StepSearch, ZeroSearch, and GlobalRAG) (Brown et al., 2020; Lewis et al., 2020; Trivedi et al., 2023; Li et al., 2025; Jin et al., 2025; Zheng et al., 2025; Sun et al., 2025; Luo et al., 2026). We additionally include **w/o Verifier** as a matched control that uses the same GRPO-optimized policy and retrieval configuration as the Full Model but disables evidence verification and Commit/Repair intervention.

Table 1: Main results of different methods on five open-domain multi-hop question answering datasets. The best result is shown in bold, and the second-best result is underlined.

| | HotpotQA | | 2Wiki | | MuSiQue | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| *Qwen2.5-3B* | | | | | | | | | | |
| Zero-Shot | 1.2 | 3.3 | 1.6 | 2.7 | 0.0 | 1.2 | 0.0 | 1.9 | 0.1 | 0.7 |
| NaiveRAG | 6.5 | 13.8 | 2.2 | 4.1 | 0.5 | 1.9 | 4.0 | 8.5 | 2.8 | 7.6 |
| IRCoT | 4.8 | 11.2 | 3.0 | 8.6 | 0.9 | 3.2 | 5.6 | 10.1 | 2.8 | 6.3 |
| Search-o1 | 14.6 | 22.1 | 14.0 | 17.0 | 3.6 | 7.7 | 26.4 | 34.9 | 5.3 | 10.3 |
| Search-R1 | 33.2 | 41.5 | 32.0 | 36.5 | 11.2 | 13.2 | 30.6 | 34.7 | 9.3 | 16.5 |
| StepSearch | 29.1 | 40.3 | 33.5 | 38.7 | 10.4 | 17.7 | 32.8 | 43.8 | 13.5 | 19.6 |
| GlobalRAG | 32.9 | 44.2 | 43.1 | 48.3 | 11.3 | 18.6 | 40.0 | **49.5** | 12.9 | 20.7 |
| Ours | **33.7** | **46.6** | **44.3** | **50.9** | **13.1** | **20.3** | **41.3** | 48.2 | **13.6** | **20.8** |
| *Qwen2.5-7B* | | | | | | | | | | |
| RA-Agent | 19.6 | 27.2 | 19.6 | 19.1 | 7.6 | 15.9 | 28.0 | 36.0 | 12.7 | 22.4 |
| Search-o1 | 17.0 | 25.4 | 27.0 | 29.2 | 8.6 | 16.5 | 30.4 | 39.3 | 13.4 | 22.9 |
| R1 | 21.6 | 33.5 | 27.8 | 29.8 | 8.4 | 14.5 | 25.0 | 33.6 | 15.1 | 24.5 |
| Search-R1 | 32.8 | 47.5 | 37.2 | 39.5 | 18.2 | 16.7 | 30.6 | 52.7 | 15.5 | 23.0 |
| StepSearch | 34.8 | 46.9 | 39.1 | 44.8 | 13.4 | 22.1 | 42.4 | 53.1 | 16.9 | 24.6 |
| ZeroSearch | 34.6 | 48.9 | 35.2 | 42.4 | 18.4 | 23.6 | 33.3 | 44.7 | 17.0 | 27.9 |
| GlobalRAG | 43.8 | **50.7** | 50.1 | 53.2 | 17.8 | 21.2 | 50.5 | 61.6 | 23.9 | 28.0 |
| Ours | **46.0** | 50.6 | **62.0** | **62.0** | **20.6** | **25.3** | **53.6** | **64.3** | **38.0** | **34.6** |

**Evaluation Metrics.** Following ZeroSearch (Sun et al., 2025), we report answer-containment Exact Match (EM) and token-level F1; metric definitions are provided in Appendix A.2.

**Training and Implementation.** We instantiate EvidenceGate with Qwen2.5-3B and Qwen2.5-7B backbones (Qwen Team, 2024) and train it on HotpotQA, 2WikiMultiHopQA, MuSiQue, and WikiHop using the staged procedure in Figure 2. Full configurations and hyperparameters are provided in Appendix A.3.

### 4.2 Main Results

Table 1 compares EvidenceGate with non-retrieval reasoning, retrieval-augmented reasoning, and reinforcement-learning search baselines across five multi-hop QA benchmarks. We make the following observations.

**Overall performance.** Under the Qwen2.5-7B setting, EvidenceGate achieves the highest EM on all five datasets and the best F1 on 2Wiki, MuSiQue, Bamboogle, and WikiHop, while remaining within 0.1 F1 points of the best result on HotpotQA. Compared with GlobalRAG, EvidenceGate improves average EM from 37.2 to 44.0 and average F1 from 42.9 to 47.4, corresponding to relative gains of 18.3% and 10.3%, respectively. It also improves EM over ZeroSearch on all five datasets, with gains ranging from 2.2 to 26.8 points.

**Comparison with search and reasoning baselines.** EvidenceGate improves EM over GlobalRAG on all five datasets, with the largest gains observed on 2Wiki and WikiHop: 11.9/8.8 and 14.1/6.6 EM/F1 points, respectively. A plausible explanation is the alignment between their task structures and first-hop verification. 2Wiki explicitly constructs compositional, inference, and bridge-comparison chains in which an intermediate answer determines a subsequent relation or comparison (Ho et al., 2020), while WikiHop requires traversal over entity–document paths among type-consistent candidate answers (Welbl et al., 2018). In both settings, an incorrect early binding can redirect the remaining reasoning toward a plausible but incompatible entity path, making the first-hop intervention particularly consequential. EvidenceGate also improves average EM/F1 over Search-R1 and StepSearch by 17.2/11.5 and 14.7/9.1 points, respectively, and outperforms the retrieval-free R1 model by 24.5/20.2 points. These results show that candidate-level verification provides benefits beyond search-policy optimization or retrieval-free reasoning alone.

Table 2: Ablation and controlled comparison results on five open-domain multi-hop QA benchmarks using Qwen2.5-7B.

| | HotpotQA | | 2Wiki | | MuSiQue | | Bamboogle | | WikiHop | | Avg. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| w/o Verifier | 43.7 | 49.3 | 56.3 | 56.4 | 16.7 | 22.5 | 51.2 | 62.2 | 36.0 | 34.9 | 40.8 | 45.1 |
| Random Repair | 43.3 | 48.6 | 57.0 | 55.2 | 18.7 | 22.9 | 52.0 | 65.5 | 36.3 | **36.8** | 41.5 | 45.8 |
| Verifier Only | 45.3 | 50.3 | 59.0 | 57.0 | 20.3 | 25.0 | 52.8 | **65.6** | 36.7 | 35.3 | 42.8 | 46.6 |
| Always Repair | 40.6 | 49.7 | 48.2 | 52.3 | 15.1 | 20.8 | 49.6 | 62.1 | 34.5 | 32.7 | 37.6 | 43.5 |
| Full Model | **46.0** | **50.6** | **62.0** | **62.0** | **20.6** | **25.3** | **53.6** | 64.3 | **38.0** | 34.6 | **44.0** | **47.4** |

**Generalization across model scales.** Under Qwen2.5-3B, EvidenceGate achieves the highest EM on all five datasets and improves average EM/F1 over GlobalRAG from 28.0/36.3 to 29.2/37.4. The consistent gains across the 3B and 7B settings indicate that the Commit/Repair mechanism is not tied to a particular backbone scale.

### 4.3 Ablation and Controlled Analysis

To examine the contributions of the evidence verifier, Commit/Repair value model, and selective repair strategy, we consider five experimental settings: (1) **w/o Verifier**, which directly commits the first-hop candidate without evidence verification or repair; (2) **Random Repair**, which randomly triggers Repair using the same repair ratio as Verifier Only; (3) **Verifier Only**, which makes Commit/Repair decisions using only the verifier outputs, without the Commit/Repair value model; (4) **Always Repair**, which performs an additional retrieval for every first-hop candidate; and (5) **Full Model**, which uses the verifier to resolve clear evidence states and applies the Commit/Repair value model only to ambiguous verifier-rejected states. All settings use the same base search policy and retrieval configuration. The results are reported in Table 2.

**Effect of the evidence verifier.** Compared with w/o Verifier, Verifier Only improves EM across all five datasets, yielding average gains of 2.0 EM and 1.6 F1. The improvement is most pronounced on MuSiQue, reaching 3.6 EM and 2.5 F1. These results indicate that evidence relevant to a subquestion does not necessarily provide sufficient support for the current candidate. Verifying candidate-level evidence before commitment reduces the risk of propagating incorrect intermediate answers into subsequent queries.

**Evidence-guided repair outperforms random repair.** To determine whether the improvement arises merely from additional retrieval, we compare Verifier Only with Random Repair under the same repair ratio. Verifier Only improves average EM and F1 by 1.4 and 0.8, respectively, and achieves consistent gains on HotpotQA, 2WikiMultiHopQA, and MuSiQue. This demonstrates that the verifier does more than increase retrieval frequency: it identifies first-hop states with insufficient evidence and allocates the repair budget to candidates that are more likely to cause erroneous binding.

**Effect of the Commit/Repair value model.** Adding the Commit/Repair value model to the verifier improves average EM from 42.8 to 44.0 and average F1 from 46.6 to 47.4. The largest improvement occurs on 2WikiMultiHopQA, with gains of 3.0 EM and 5.0 F1. Whereas Verifier Only relies solely on local evidence assessments, the Commit/Repair value model is trained using paired Commit/Repair outcomes and final QA rewards, enabling it to estimate the downstream utility of Repair relative to Commit. Compared with Verifier Only, the Full Model improves EM on all five datasets, although F1 decreases on Bamboogle and WikiHop. These results show more consistent gains in exact match than in token-level F1.

**Selective repair outperforms unconditional re-retrieval.** Always Repair obtains average scores of 37.6 EM and 43.5 F1, which are 6.4 EM and 3.9 F1 lower than those of the Full Model. Unconditional re-retrieval can introduce irrelevant or conflicting evidence into states that are already sufficiently supported. By allocating additional retrieval only to states judged to benefit from Repair, the Full Model achieves a better balance between answer quality and retrieval overhead.

### 4.4 First-Hop Error Detection and Repair Analysis

To determine whether the proposed modules act on the intended failure mode, we separately assess the verifier's detection of first-hop binding errors and the value model's targeting of Repair. Verifier precision is measured among flagged samples, while verifier recall is measured over all annotated first-hop errors. For ambiguity-routed states evaluated by the value model, precision measures the fraction of Repair decisions applied to first-hop errors, and conditional recall measures the fraction of routed first-hop errors selected for Repair. Count-based statistics are provided in Appendix B.2.

Table 3: First-hop error detection and repair-targeting performance.

| | Verifier | | Value Model | |
|---|---|---|---|---|
| Dataset | Prec. | Rec. | Prec. | Cond. Rec. |
| Bamboogle | 75.0% | 62.5% | 50.0% | 66.7% |
| 2Wiki | 72.9% | 58.3% | 66.7% | 72.7% |
| WikiHop | 88.1% | 60.7% | 57.1% | 36.4% |
| MuSiQue | 70.6% | 55.8% | 64.3% | 47.4% |
| HotpotQA | 71.4% | 37.2% | 54.5% | 60.0% |
| **Overall** | **74.3%** | **52.8%** | **60.4%** | **53.7%** |

**Verifier detection of first-hop errors.** Across the five datasets, verifier precision ranges from 70.6% to 88.1%, with an overall precision of 74.3%, while recall ranges from 37.2% to 62.5%, with an overall recall of 52.8%. The relatively high precision indicates that the verifier does not indiscriminately flag uncertain samples, but identifies states with insufficient evidence or elevated binding risk. Its moderate recall is consistent with a conservative gate that prioritizes reliable intervention over maximizing coverage. This filtering strategy therefore covers a meaningful portion of first-hop errors while limiting intervention in sufficiently supported states.

**Value-model repair targeting.** Among ambiguity-routed states evaluated by the value model, precision ranges from 50.0% to 66.7%, with an overall precision of 60.4%, while conditional recall ranges from 36.4% to 72.7%, with an overall recall of 53.7%. Although the values vary across datasets, the results consistently show that Repair is preferentially directed toward erroneous bindings rather than allocated uniformly. This behavior reflects dataset-dependent repair conservativeness while concentrating additional retrieval on states more likely to benefit from it.

Combined with the consistent EM improvements of the Full Model over Verifier Only in Table 2, these findings show that the proposed risk estimates and repair decisions target the intended error type and improve final-answer correctness. Appendix D illustrates this process in a representative case study.

## 5 Conclusion

We propose EvidenceGate, an evidence-guided Commit/Repair framework for open-domain multi-hop question answering. It targets error propagation caused by premature first-hop answer binding under insufficient evidence. By training an evidence verifier with controllable simulated retrieval and learning cost-adjusted repair advantages from rollout-level final QA rewards, EvidenceGate connects candidate-level evidence assessment with downstream answer correctness. This enables the model to decide whether to commit a first-hop answer or perform repair retrieval before the answer is reused in later queries. Experiments on five multi-hop QA benchmarks show consistent improvements, and ablation and controlled analyses validate the contributions of evidence verification, value-based repair decisions, and selective repair. These results suggest that effective multi-hop search reasoning requires not only generating intermediate answers, but also deciding whether the available evidence is sufficient to support their commitment. While first-hop gating offers a high-leverage intervention with bounded overhead, future work can extend EvidenceGate to later intermediate steps and study how to allocate verification and repair budgets across the full reasoning chain.

## AI USE DISCLOSURE

OpenAI ChatGPT and Codex were used to assist with literature retrieval, manuscript drafting and language editing, numerical consistency checks, and the candidate-level label audit reported in Appendix A.4. All AI-assisted outputs were reviewed by the authors, who take responsibility for the final content of the paper.

## REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1870–1879, 2017.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, 2024.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 16477–16508, 2023.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3929–3938, 2020.

Bolei He, Nuo Chen, Xinran He, Lingyong Yan, Zhenkai Wei, Jinchang Luo, and Zhen-Hua Ling. Retrieving, rethinking and revising: The chain-of-verification can improve retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10371–10393, 2024.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*, 2021.

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, Barcelona, Spain, December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 874–880, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 7036–7050, 2024.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, 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.

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*, pp. 6769–6781, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://iclr.cc/virtual/2024/poster/17549.

Jinchang Luo, Mingquan Cheng, Fan Wan, Ni Li, Xiaoling Xia, Shuangshuang Tian, Tingcheng Bian, Haiwei Wang, Haohuan Fu, and Yan Tao. GlobalRAG: Enhancing global reasoning in multi-hop question answering via reinforcement learning. *arXiv preprint arXiv:2510.20548*, 2026.

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

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9248–9274, 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.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang, and Yan Zhang. 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: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 10014–10037, 2023.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9426–9439, 2024.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a.

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

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21805–21830, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1106. URL https://aclanthology.org/2025.emnlp-main.1106/.

# APPENDIX

## A EXPERIMENTAL DETAILS

### A.1 DATASETS

Our experiments use five multi-hop question answering datasets: HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle, and WikiHop. To prevent data leakage, we remove potentially duplicated instances between the training and evaluation sets, ensuring that evaluation samples do not appear in the training data.

For EvidenceGate-specific fine-tuning, we use a mixed training set constructed from HotpotQA, 2WikiMultiHopQA, MuSiQue, and WikiHop, containing 1,000 samples in total. Specifically, we sample 250 instances from each dataset. Bamboogle is not used for training and serves as an additional evaluation dataset.

For datasets with non-public test sets, including HotpotQA, 2WikiMultiHopQA, MuSiQue, and WikiHop, we use the corresponding development sets as test proxies for evaluation. During evaluation, we call the real Google search engine through the Serper API to obtain retrieval evidence.

### A.2 EVALUATION METRICS

Following the ZeroSearch evaluation protocol (Sun et al., 2025), we evaluate answer quality using answer-containment Exact Match (EM) and token-level F1. Precision and recall are additionally used to analyze first-hop error detection and repair decisions.

#### ANSWER-CONTAINMENT EXACT MATCH

Let $\mathcal{N}(\cdot)$ denote the normalization used by ZeroSearch, which lowercases text, removes punctuation and the English articles *a*, *an*, and *the*, and collapses consecutive whitespace. For a prediction $\hat{y}_i$ and a set of acceptable gold answers $\mathcal{A}_i$, the per-example score is

$$\mathrm{EM}_i = \max_{a \in \mathcal{A}_i} \mathbf{1}[\mathcal{N}(a) \text{ is a contiguous substring of } \mathcal{N}(\hat{y}_i)]\,. \tag{18}$$

The dataset-level score is

$$\mathrm{EM} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{EM}_i, \tag{19}$$

where $\mathbf{1}[\cdot]$ is the indicator function.

#### F1 SCORE

F1 is the harmonic mean of token-level precision $P$ and recall $R$ between the predicted and ground-truth answers:

$$\mathrm{F1} = \frac{2PR}{P+R}. \tag{20}$$

#### PRECISION AND RECALL

Precision and recall evaluate the correctness and coverage of positive predictions:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{21}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

### A.3 TRAINING AND IMPLEMENTATION DETAILS

**Retrieval-Augmented GRPO Objective.** The base search policy is optimized using the final QA reward with KL regularization:

$$\max_{\theta} \quad \mathbb{E}_{x,\tau}[R(\tau)] - \beta_{\mathrm{KL}} D_{\mathrm{KL}}\left[\pi_\theta(\cdot \mid x; \pi_\psi) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x; \pi_\psi)\right], \tag{22}$$

where $R(\tau)$ is computed from the final QA result and $\pi_{\text{ref}}$ is the reference policy. For each question, GRPO samples a group of $G$ trajectories $\{\tau_i\}_{i=1}^{G}$ and computes

$$\widetilde{A}_i^{\text{GRPO}} = \frac{R(\tau_i) - \mu_R}{\sigma_R + \epsilon}, \qquad \mu_R = \frac{1}{G}\sum_{j=1}^{G} R(\tau_j), \tag{23}$$

where $\sigma_R$ is the standard deviation of the group rewards.

**Policy Initialization.** For both the Qwen2.5-3B and Qwen2.5-7B experiments, we initialize the base search policies from checkpoints trained with ZeroSearch (Sun et al., 2025). Starting from these checkpoints, we continue GRPO optimization on the 1,000-example training mixture described in Appendix A.1. The evidence verifier and Commit/Repair value model are initialized separately and trained using the objectives described in Sections 3.4 and 3.5 of the main paper.

**Computing Infrastructure.** All experiments were conducted on a server equipped with four NVIDIA A800 GPUs. Two GPUs were allocated to serving the parameter-frozen simulated search model during policy rollouts, while the other two GPUs were used for model training and fine-tuning.

**Training Hyperparameters.** For GRPO fine-tuning, we use a learning rate of $1 \times 10^{-6}$, a clipping ratio of $0.2$, and a KL coefficient of $0.02$. For each question, we sample five trajectories with a temperature of $0.8$ and top-$p$ of $0.95$. The maximum prompt and response lengths are set to 4,096 and 2,048 tokens, respectively. Each trajectory is allowed at most six interaction turns, and the top five documents are returned for each query. The simulated retrieval environment uses the publicly released 14B Google-search simulator from ZeroSearch (Sun et al., 2025), referred to as `Simulation_LLM_google_14B`. The simulator and download instructions are available in the official ZeroSearch repository.[1]

After initial GRPO fine-tuning, we train the evidence verifier for five epochs with a mini-batch size of 8 candidate–document pairs, a learning rate of $2 \times 10^{-5}$, a maximum sequence length of 384, and a weight decay of $0.01$. The cross-encoder is initialized from MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli. During verifier-guided policy refinement, we set the repair-behavior reward weight to $\gamma = 1.0$ and allow at most one repair retrieval per sample.

We train the Commit/Repair value model using paired rollouts generated by the refined policy, with a mini-batch size of 2 first-hop decision states and a learning rate of $1 \times 10^{-3}$, where each state is supervised by the cost-adjusted return difference between paired Commit and Repair rollouts from that same state. The transition parameter of the SmoothL1 loss is set to $0.25$, the decision temperature to $T = 0.2$, the decision-loss weight to $\eta = 0.5$, and the repair cost coefficient to $\lambda = 0.05$. For ambiguity-routed states at inference time, Repair is selected when the predicted repair advantage is greater than zero. Unless otherwise specified, the same hyperparameter settings are used across all datasets.

**Evidence-Gate Routing.** Given the document-level verifier probabilities, we define the evidence-gate score and the support–refute conflict as

$$s_{\text{evi}}(s_1) = \max_i p_i^{\text{sup}}, \qquad \kappa(s_1) = \left(\max_i p_i^{\text{sup}}\right)\left(\max_i p_i^{\text{ref}}\right).$$

We use a support boundary of $\tau_s = 0.25$ and define the signed verifier margin as $m_v(s_1) = s_{\text{evi}}(s_1) - \tau_s$. The complete routing rule is

$$r(s_1) = \begin{cases} \text{Commit}, & s_{\text{evi}}(s_1) \geq \tau_s, \\ \text{Value}, & s_{\text{evi}}(s_1) < \tau_s \ \wedge\ (|m_v(s_1)| \leq \delta \ \vee\ \kappa(s_1) \geq \tau_c), \\ \text{Repair}, & \text{otherwise.} \end{cases}$$

We set $\delta = 0.10$ and $\tau_c = 0.20$. Thus, clearly supported states are committed, ambiguous verifier rejections are routed to the Commit/Repair value model and follow Eq. (17), and the remaining clear rejections trigger Repair directly.

[1] https://github.com/Alibaba-NLP/ZeroSearch

**Repair-Behavior Reward.** The repair-behavior term in Eq. (9) is instantiated as

$$R_{\text{repair}}(\tau) = I_{\text{search}}(\tau) + I_{\text{rebind}}(\tau) - I_{\text{premature}}(\tau).$$

Here, $I_{\text{search}}(\tau) = 1$ if the policy performs the required repair search after the verifier blocks a first-hop binding; $I_{\text{rebind}}(\tau) = 1$ if, after repair evidence returns, the policy regenerates and binds the same first-hop answer slot; and $I_{\text{premature}}(\tau) = 1$ if the policy emits a final answer while repair is still pending. Each indicator is zero otherwise. This reward supervises only completion of the verifier-triggered repair protocol; answer correctness after repair is still evaluated by the final $R_{\text{QA}}$. Since at most one repair is allowed per sample, these indicators are binary in the current setting. Any uniform scaling of this reward is absorbed into $\gamma$ in Eq. (9).

### A.4 Controllable Evidence Simulation Details

The role-controlled evidence simulation is applied to first-hop retrieval, while subsequent retrieval steps follow the standard simulated-search setting. For each first-hop query, we sample a clean configuration with probability 0.3 and a mixed configuration with probability 0.7. Each retrieval returns five documents. The clean configuration contains five Support documents, whereas the mixed configuration contains one Support document, three Partial documents, and one Wrong document. We assign a larger sampling probability to the mixed configuration to expose the verifier to more first-hop states dominated by incomplete or misleading evidence, while retaining clean configurations to prevent the training distribution from becoming uniformly adversarial.

The simulator generates each document as a concise, search-style evidence snippet together with a hidden role annotation. The annotation is retained as supervision metadata and is not presented to the search policy. A bundle contributes labeled verifier examples only when its document count and role composition match the requested configuration and its documents satisfy the quality requirements. Bundles that fail these checks are excluded from the labeled verifier data, although valid document content may still be used as unlabeled retrieval context during policy rollout.

We instantiate the matching function $M(c_1, d_i)$ in Eq. (6) using normalized token coverage. Let $T(c_1)$ and $T(d_i)$ denote the normalized tokens of the first-hop candidate $c_1$ and document $d_i$, respectively. Candidate coverage is defined as

$$\text{Cov}(c_1, d_i) = \frac{\sum_{t \in T(c_1)} \mathbb{I}[t \in T(d_i)]}{|T(c_1)|}. \tag{24}$$

The candidate is associated with the document when

$$M(c_1, d_i) = \mathbb{I}\left[\text{Cov}(c_1, d_i) \geq \xi\right], \tag{25}$$

where $\xi$ is a fixed matching threshold shared across all datasets. The resulting match is combined with the hidden document role according to Eq. (6) to construct the support, refute, and neutral supervision used to train the evidence verifier.

**Candidate-Level Label Audit.** To assess the semantic quality of the supervision produced by Eq. (6), we conduct a post-hoc, LLM-assisted human audit of a class-balanced sample of 150 candidate–document pairs, comprising 50 automatically labeled examples from each class. Adjudication considers the textual relation among the first-hop subquestion, policy candidate, and synthetic document: *support* indicates that the document entails the candidate as an answer to the subquestion, *refute* indicates that it contradicts the candidate or supplies a mutually incompatible answer, and *neutral* indicates that it is irrelevant, ambiguous, or insufficient by itself.

The audit yields 66.0% overall agreement. The main source of disagreement is the semantic boundary between neutral and refute: 19 of the 50 automatically labeled neutral examples are adjudicated as refute. This pattern indicates that lexical candidate coverage often recognizes the absence of positive support but does not reliably distinguish incomplete evidence from evidence that supplies a conflicting answer. Because neutral and refute both represent a lack of positive support, this confusion preserves much of the binary support-sufficiency signal. However, it can shift probability mass from the refute class to the neutral class, potentially weakening the support–refute conflict score used for ambiguity routing.

| Audit Target | Sample Size | Correct Labels | Accuracy |
|---|---|---|---|
| Support | 50 | 38 | 76.0% |
| Refute | 50 | 35 | 70.0% |
| Neutral | 50 | 26 | 52.0% |
| Overall | 150 | 99 | 66.0% |

Table 4: LLM-assisted human audit of candidate-conditioned verifier labels on a class-balanced sample. Correct Labels counts agreement between the automatically constructed label and the adjudicated label.

| Automatic Label | Adjudicated Label | | | Total |
|---|---|---|---|---|
| | Support | Refute | Neutral | |
| Support | **38** | 4 | 8 | 50 |
| Refute | 6 | **35** | 9 | 50 |
| Neutral | 5 | 19 | **26** | 50 |
| Total | 49 | 58 | 43 | 150 |

Table 5: Confusion matrix for the candidate-conditioned label audit. Rows are labels constructed by Eq. (6), and columns are manually adjudicated labels.

### A.5 Commit/Repair Rollout Construction

We instantiate the interventions in Eq. (11) from the same first-hop state $s_1 = (q_1, c_1, D_1)$. The two branches share the original question, first-hop subquestion, candidate answer, initially retrieved evidence, search policy, and subsequent reasoning procedure. They differ only in the transition applied by the rollout controller at the first-hop binding point.

For the Commit rollout $\tau_C$, the controller binds $c_1$ to the first-hop answer slot $\#1$. The policy then continues the remaining subquestions using the committed intermediate answer.

For the Repair rollout $\tau_R$, the controller applies the same Repair transition used at inference time (Section 3.6). It suspends the binding of $c_1$, constructs a focused query from the original question and $q_1$ without including $c_1$, and executes one additional retrieval. The policy then conditions on both $D_1$ and the newly retrieved evidence, regenerates the first-hop answer, binds the regenerated answer to $\#1$, and continues the remaining reasoning process.

After both branches produce final answers, their QA rewards are substituted into Eq. (12) to obtain the cost-adjusted repair-advantage target used to train the Commit/Repair value model.

## B First-Hop Error Analysis

### B.1 Annotation and Error Prevalence

To quantify premature first-hop answer binding, we sample up to 300 instances from each dataset, including HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle, and WikiHop. Since Bamboogle contains only 125 instances, we evaluate its full set, resulting in 1,325 analyzed instances in total.

For each instance, we inspect the first-hop candidate together with its subsequent use in the search-reasoning trajectory. A case is labeled as a premature first-hop binding error when the candidate is incorrect or insufficiently supported and its commitment causes subsequent retrieval or reasoning to proceed along an erroneous path. This criterion distinguishes premature binding from general final-answer errors by requiring the unreliable intermediate answer to affect later reasoning steps.

We identify 377 premature first-hop binding errors, accounting for 28.45% of all analyzed instances. Specifically, we observe 94 cases on HotpotQA, 60 on 2WikiMultiHopQA, 138 on MuSiQue, 24 on Bamboogle, and 61 on WikiHop.

### B.2 Detailed Detection and Repair Statistics

Tables 6 and 7 provide the count statistics underlying Table 3. Applying the deterministic routing rule in Appendix A.3 to the 268 verifier-flagged states yields 102 ambiguity-routed states evaluated by the Commit/Repair value model; the remaining 166 clear rejections bypass the value model and trigger Repair directly. Routing depends only on verifier outputs, while the manual first-hop error annotations are used only for the subsequent error analysis. In Table 6, Error denotes all first-hop binding errors, Flagged denotes verifier-flagged samples, and Flagged Error denotes their intersection. In Table 7, Routed denotes states evaluated by the value model, Repair denotes states selected for repair, and Routed Error and Repaired Error denote the corresponding first-hop errors.

Table 6: Count statistics for verifier detection of first-hop binding errors.

| Dataset | Error | Flagged | Flagged Error |
|---|---|---|---|
| Bamboogle | 24 | 20 | 15 |
| 2Wiki | 60 | 48 | 35 |
| WikiHop | 61 | 42 | 37 |
| MuSiQue | 138 | 109 | 77 |
| HotpotQA | 94 | 49 | 35 |
| **Overall** | **377** | **268** | **199** |

Table 7: Count statistics for value-model repair targeting among ambiguity-routed states.

| Dataset | Routed | Repair | Routed Error | Repaired Error |
|---|---|---|---|---|
| Bamboogle | 10 | 4 | 3 | 2 |
| 2Wiki | 16 | 12 | 11 | 8 |
| WikiHop | 23 | 7 | 11 | 4 |
| MuSiQue | 35 | 14 | 19 | 9 |
| HotpotQA | 18 | 11 | 10 | 6 |
| **Overall** | **102** | **48** | **54** | **29** |

At the count level, 199 of the 268 verifier-flagged states are genuine first-hop errors, covering 199 of the 377 annotated errors. Among the 102 ambiguity-routed states, the value model selects Repair in 48 cases, including 29 of the 54 routed errors. These results show a selective cascade in which the verifier identifies risky first-hop bindings and the value model concentrates repair on ambiguous states with predicted downstream benefit.

## C Gate Scope and Inference Overhead

Table 8 compares three gating scopes. No Gate disables the verifier and all Commit/Repair interventions, First-Hop Gate applies the gate only to the first-hop binding point, and All-Hop Gate applies the same mechanism to every intermediate binding point with at most one repair per answer slot. Gate/Q reports candidate-level verifier evaluations per question; repeated candidate attempts can therefore make it greater than one under First-Hop Gate. Repair/Q reports triggered repair interventions per question, and Search/Q reports all regular and repair retrieval calls per question.

Table 8: Gate-scope comparison on the full Bamboogle evaluation set of 125 questions.

| Setting | EM | F1 | Gate/Q | Repair/Q | Search/Q |
|---|---|---|---|---|---|
| No Gate | 51.2 | 62.2 | 0.000 | 0.000 | 1.992 |
| First-Hop Gate | 53.6 | 64.3 | 1.096 | 0.144 | 2.160 |
| All-Hop Gate | 38.4 | 50.0 | 1.976 | 0.392 | 2.592 |

First-Hop Gate improves EM from 51.2 to 53.6 and F1 from 62.2 to 64.3 over No Gate, while Search/Q rises only from 1.992 to 2.160 because verifier evaluations add retrieval only when Repair is selected. All-Hop Gate increases Gate/Q to 1.976, Repair/Q to 0.392, and Search/Q to 2.592, but

lowers EM and F1 to 38.4 and 50.0. The comparison therefore favors first-hop gating: it improves answer quality at modest retrieval cost, whereas extending the intervention to later hops increases inference operations without improving accuracy.

## D CASE STUDY

We illustrate the Commit/Repair process using a representative Bamboogle example.

Table 9: Representative EvidenceGate Commit/Repair trajectory on Bamboogle.

---

**Case 1:** What league was the team that Jose Cancela was a member of in?
**Ground Truth:** Major League Soccer (MLS)

---

**EvidenceGate (Full Model).**
<think> To answer this question, I need to identify the team that Jose Cancela was a member of and then determine the league in which that team competed. </think>

<plan>
{ "Q1": ["What team was Jose Cancela a member of?", "#1"], "Q2": ["What league does #1 play in?", "#2"] }
</plan>

<subPlan>
<think> First, I need to identify the team that Jose Cancela was a member of. </think>
<search> Jose Cancela team membership </search>
<information>
**Doc 1:** Jose Cancela has been named president of Telemundo Station Group.
**Doc 2:** Jose Cancela is a media executive with extensive experience at Telemundo and Univision.
...
</information>
<think> The retrieved documents describe a media executive and do not identify a sports team. I should refine the query to locate the athlete with the same name. </think>
<search> Jose Cancela sports team </search>
<information>
**Doc 1:** Jose Cancela is the president of Telemundo Station Group.
**Doc 2:** Jose Cancela is associated with the United Football League as a composer.
**Doc 3:** Jose Carlos Cancela is a midfielder from Santa Lucia, Uruguay. His profile is listed on MLSsoccer.com.
...
</information>
<candidate> $c_1$ **= United Football League** </candidate>

**Verifier.**
The evidence does not safely support the candidate binding. The retrieved documents conflate a media executive, a media credit associated with the United Football League, and a professional soccer player. The candidate is therefore not committed.
**Value Model.**
Decision: **Repair**.

---

**Case 1 Continued:** Repair retrieval and final reasoning.

<search> Jose Cancela soccer team </search>
<information>
**Doc 1:** Jose Cancela played for Peñarol, Cruz Azul, Deportivo Saprissa, and the New England Revolution.
**Doc 2:** Jose Cancela is a professional midfielder listed by MLSsoccer.com.
**Doc 3:** Jose Cancela played for the Colorado Rapids.
...
</information>
<think> The refined evidence identifies the relevant Jose Cancela as a professional soccer player and directly associates him with the Colorado Rapids. </think>
<subAnswer> **#1 = Colorado Rapids** </subAnswer>
</subPlan>

<subPlan>
<think> Now that the correct team has been identified, I need to determine which league the Colorado Rapids compete in. </think>
<search> Colorado Rapids league </search>
<information>
**Doc 1:** The Colorado Rapids are an American professional soccer club based in the Denver metropolitan area.
**Doc 2:** The Colorado Rapids compete in Major League Soccer (MLS).
**Doc 3:** The official Colorado Rapids channel identifies the club as a Major League Soccer team.
...
</information>
<think> The retrieved evidence clearly states that the Colorado Rapids compete in Major League Soccer.
</think>
<subAnswer> **#2 = Major League Soccer (MLS)** </subAnswer>
</subPlan>

<think> Jose Cancela was a member of the Colorado Rapids, which compete in Major League Soccer.
</think>

<answer> **Major League Soccer (MLS)** </answer>

---

The initial retrieval conflates multiple identities and yields an unsupported first-hop binding. EvidenceGate blocks the candidate and uses candidate-independent repair to rebind Jose Cancela to the Colorado Rapids, preventing the error from propagating to the final answer.