EVIDENCE IN THE LOOP: TRAINING SEARCH AGENTS FROM KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training search agents increasingly relies on synthetic data to reduce human annotation costs. Most prior methods generate training data offline, leaving the task distribution fixed and unable to adapt to the agent's evolving capabilities. Self-evolving frameworks address this limitation by synthesizing data online as the agent improves. However, optimization remains largely outcome-driven, resulting in sparse supervision for intermediate search behavior. Recent work therefore incorporates sampled knowledge graph (KG) paths into self-evolving training, using relational structure to guide question generation and path entities as intermediate supervision. However, this entity-centric design can weaken both the effectiveness of training data and the supervision signal: generated questions may still be answerable even when parts of the factual chain behind it are not explicitly supported by the target text corpus, while entity-based rewards can conflate mentioning path entities with genuine search progress, even when no relevant evidence is actually retrieved to support the solution. To this end, we propose EVIDENCE-IN-THE-LOOP, a self-evolving framework that grounds both question generation and process supervision in retrievable corpus evidence. We sample multi-hop KG paths and retain only those whose triplets are supported by verified corpus evidence. The retained paths guide online question generation, while the associated evidence is reused to supervise the agent's search process. Specifically, we introduce two process rewards: *evidence acquisition*, which rewards retrieving verified supporting passages, and *evidence utilization*, which rewards using newly acquired evidence to guide subsequent search toward additional relevant evidence. Experiments demonstrate the effectiveness of our framework in both problem solving and question generation: it consistently improves downstream QA performance across models and initialization settings, while the trained Proposer produces questions with higher validity and greater training effectiveness.

1 INTRODUCTION

Search agents have attracted increasing attention as a promising paradigm for complex information seeking, enabling large language models (LLMs) to iteratively formulate queries, retrieve external information, and reason over available information (Lewis et al., 2020; Yao et al., 2023; Schick et al., 2023; Asai et al., 2024; Jin et al., 2025; Zheng et al., 2025). Despite their practical potential, training such agents remains challenging because high-quality question-answer (QA) pairs and trajectories are expensive to annotate at scale.

To reduce reliance on human annotation, recent work has explored entity-centric synthetic data generation. One line of work constructs training data offline, typically by transforming existing QA pairs into more challenging questions with controlled difficulty. ASearcher (Gao et al., 2026), for example, progressively replaces named entities in simple seed questions with descriptive facts and further fuzzes them into more ambiguous and complicated descriptions. However, this offline synthesis paradigm has two major limitations. First, successive free-form transformations can accumulate errors, requiring rigorous validation to maintain data quality. Second, offline question synthesis makes it difficult to adapt question difficulty to the evolving capability of the search agent during training. More recently, another line of work has explored self-evolving training frameworks. A representative approach is Search Self-Play (SSP) (Lu et al., 2026), in which a question Proposer generates questions conditioned on given answer entities through multi-turn reasoning and search, while a

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

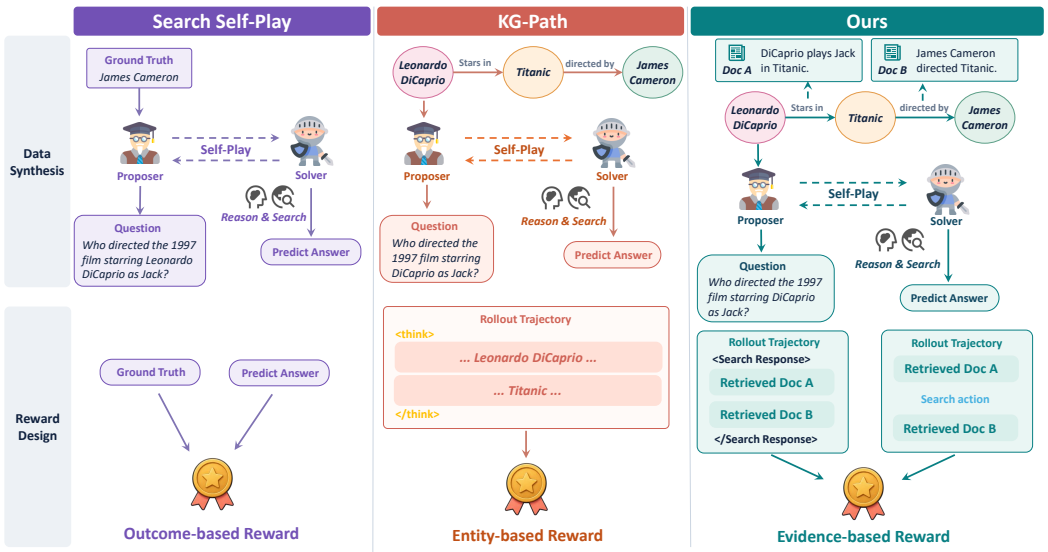

Figure 1: Comparison of Search Self-Play (SSP), KG-Path, and our framework in question generation and reward design. SSP generates questions conditioned on target answer entity and optimizes the Solver with outcome-based rewards. KG-Path incorporates sampled KG paths into question generation and derives intermediate supervision from entities along the corresponding KG paths. Our framework additionally associates each KG triplet with verified, retrievable corpus evidence and reuses this evidence to provide process-level supervision for evidence acquisition and utilization.

problem Solver learns to answer them. By jointly optimizing the two roles, SSP allows question generation to evolve with the Solver's capability and eliminates the need for seed questions.

Despite their different data synthesis paradigms, both remain heavily dependent on outcome-based rewards, which leads to a severe reward sparsity problem in agent training. To alleviate this issue, E-GRPO(Zhao et al., 2026) builds on ASearcher by retaining the entities introduced during question synthesis and assigning partial rewards to incorrect rollouts according to the occurrence of these entities in the `<think></think>` reasoning. KG-Path (Wu et al., 2026) further identifies poor question quality as a major issue in the early stage of SSP, with only 8.3% passing its validity check. To improve the validity and controllability of generated questions, it samples multi-hop KG paths ending at the answer entities and provides the corresponding subgraphs to the Proposer, anchoring question generation to explicit entity-relation chains. Following a similar idea to E-GRPO, it further assigns partial rewards during Solver training according to how many entities from the sampled path appear in the Solver's reasoning trajectories. Overall, these methods rely heavily on entities for both guiding question generation and providing intermediate supervision. However, this entity-centric formulation has two key limitations: First, entity-based question generation may rely on facts/relational without explicit support in the text corpus, reducing their effectiveness as training data even when they are answerable. Second, entity-based rewards are vulnerable to reward hacking, since agents can earn credit by only mention target entities in reasoning trace without retrieving evidence that contributes to solving the question.

To address these limitations, we introduce EVIDENCE-IN-THE-LOOP, built around a simple principle: KG structure specifies the relational structure of a question, while corpus evidence determines whether that structure is suitable for learning to search. We therefore verify each sampled multi-hop KG path against the target corpus by retrieving supporting evidence for every triplet, and retain only paths for which all triplets are supported. The retained paths provide a reliable basis for online question generation, with additional filtering to ensure validity, answerability, and sufficient search difficulty. Then the verified evidences directly supervise search progress through two process rewards: evidence acquisition, which rewards retrieving supporting passages, and evidence utilization, which rewards using newly acquired evidence to advance subsequent search. This grounds intermediate credit in observable retrieval behavior rather than just mention entities in the reasoning trace. Comprehensive experiments demonstrate that our method consistently improves downstream

QA performance across model families and initialization settings, with particularly strong gains on multi-hop QA benchmarks. The trained Proposer also generates higher-quality questions that are more effective for training search agents. The code is publicly available at here.

2 Related Work

Training Search Agents. Search agents have recently demonstrated strong capabilities in complex information seeking by interleaving reasoning with repeated interactions with external search systems (Jin et al., 2025; Zheng et al., 2025). Proprietary systems such as OpenAI Deep Research (OpenAI, 2025), Claude Research (Anthropic, 2025), Kimi-Researcher (Moonshot AI, 2025), and Grok DeepSearch (xAI, 2025) exemplify this progress, but their closed search infrastructure and largely undisclosed training procedures limit reproducibility, while reliance on live search APIs incurs prohibitive costs for large-scale data generation. Meanwhile, open-source methods such as Search-R1 (Jin et al., 2025), DeepResearcher (Zheng et al., 2025), and ZeroSearch (Sun et al., 2025) have established reinforcement learning (RL) as an effective paradigm for learning multi-turn search behavior. These methods, however, are often trained on human-curated datasets, making task availability a key bottleneck to further scaling. Recent work alleviates this limitation by automatically synthesizing challenging search problems, but such data is typically generated offline and preventing it from adapting to the agent's evolving capability (Gao et al., 2026).

Self-Evolving Training. Self-evolving training allows models to generate their own training tasks and adapt task difficulty as their problem-solving capabilities improve. Earlier work has largely focused on domains such as math and coding, where task correctness can be reliably verified using rule-based or executable verifiers (Lightman et al., 2024; Wang et al., 2024b; Zhao et al., 2025; Zhang et al., 2025; Zhou et al., 2025). Recently, Search Self-Play (Lu et al., 2026) extends this paradigm to open-domain QA, where a Proposer generates and verifies questions through a structured pipeline, while the Solver learns to answer them through search and reasoning. Dr.Zero (Yue et al., 2026) further removes the reliance on seed QA data by enabling the Proposer to synthesize both questions and answers through interaction with the search environment. However, fully unconstrained task generation remains challenging for multi-hop search agents, due in part to limited question diversity and insufficient structural guidance (Yu et al., 2025). More importantly, these advances primarily improve task generation rather than search supervision: optimization still relies predominantly on outcome-based supervision, leaving intermediate search steps without direct feedback and resulting in a sparse reward problem during RL (Zhao et al., 2026).

Knowledge Graphs as Training Data Source. Knowledge graphs (KGs) provide explicit relational structures and have been widely used to construct supervision for KGQA models (Zhang et al., 2018; Huang et al., 2025; Talmor & Berant, 2018; Zhang et al., 2026). However, KGQA operates over structured graphs, whereas open-domain QA requires retrieving evidence from unstructured corpora, motivating recent efforts to extend this training paradigm to open-domain QA. DeepDive (Lu et al., 2025) synthesizes challenging multi-hop questions from KG paths and uses the resulting QA pairs for multi-turn RL. KG-Path (Wu et al., 2026) further incorporates structure relational in KG into self-evolving training, using KG subgraphs to guide question generation and using entities along the construction path to provide intermediate supervision. However, KG-derived facts are not necessarily supported by retrievable evidence in the target corpus, which may compromise question quality. Entity-based rewards are also vulnerable to reward hacking, since mentioning path entities can yield intermediate rewards without retrieving useful supporting evidence.

3 Preliminaries

Search Agent. Let $\mathcal{N}$ denote the natural-language sequence space over vocabulary $\mathcal{V}$. Given an input prompt $\boldsymbol{x}$, an LLM policy π_θ interacts with a search environment through multiple rounds of generation and retrieval. A search trajectory is denoted as $\boldsymbol{\tau} = (\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{o}_1, \boldsymbol{y}_2, \boldsymbol{o}_2, \ldots, \boldsymbol{y}_T)$, where $\boldsymbol{y}_t \in \mathcal{N}$ is the model output at step t and $\boldsymbol{o}_t \in \mathcal{N}$ is the observation returned by the search tool. We use $\mathcal{A}(\boldsymbol{\tau})$ to denote the final answer extracted from a trajectory and $r(\mathcal{A}(\boldsymbol{\tau}), \boldsymbol{a}^*)$ to denote the outcome reward comparing the predicted answer with the ground-truth answer $\boldsymbol{a}^*$.

Search Self-Play. Search Self-Play (SSP) (Lu et al., 2026) trains the same policy π_θ to alternate between a question *Proposer* and a problem *Solver*. Given role-specific prompts $\boldsymbol{x}_{\text{propose}}$ and $\boldsymbol{x}_{\text{solve}}$, the corresponding policies are

$$u(\cdot \mid \boldsymbol{a}^*) = \pi_\theta(\cdot \mid \boldsymbol{x}_{\text{propose}}, \boldsymbol{a}^*), \qquad v(\cdot \mid \boldsymbol{q}) = \pi_\theta(\cdot \mid \boldsymbol{x}_{\text{solve}}, \boldsymbol{q}). \tag{1}$$

where $\boldsymbol{a}^*$ denotes the target answer provided to the Proposer, and $\boldsymbol{q}$ denotes the generated question. Let $\boldsymbol{\tau}$ and $\boldsymbol{\rho}$ denote the Proposer and Solver trajectories, respectively, and let $\mathcal{Q}(\boldsymbol{\tau})$ and $\mathcal{A}(\boldsymbol{\rho})$ extract the generated question and predicted answer. SSP formulates their interaction as

$$\min_u \max_v \mathbb{E}_{\boldsymbol{a}^* \sim \mathcal{D}, \boldsymbol{\tau} \sim u(\cdot|\boldsymbol{a}^*), \boldsymbol{\rho} \sim v(\cdot|\mathcal{Q}(\boldsymbol{\tau}))} \left[r(\mathcal{A}(\boldsymbol{\rho}), \boldsymbol{a}^*) \right]. \tag{2}$$

where $\mathcal{D}$ is the answer set. The Proposer is encouraged to generate increasingly challenging questions, while the Solver is optimized to answer them correctly.

However, the Proposer may generate questions that appear challenging because they are invalid or unanswerable, which can undermine the self-play objective. SSP therefore employs a two-stage filtering procedure before using generated questions for training: rule-based filtering and RAG-based verification. It first applies inexpensive rule-based checks to remove malformed outputs, empty or excessively short questions, questions that expose the target answer, and generations that do not invoke the search tool. The remaining questions are then subjected to RAG-based verification. Specifically, it collects the search observations $\boldsymbol{O}_T = (\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T)$ from the Proposer trajectory and verifies whether the Solver can recover $\boldsymbol{a}^*$ from $(\mathcal{Q}(\boldsymbol{\tau}), \boldsymbol{O}_T)$ without additional search. Only questions that pass this verification are retained for self-play training. For each retained question, the Solver samples n search trajectories and receives final-answer rewards.

$$r_{\text{solve}}^{(i)} = r\left(\mathcal{A}(\boldsymbol{\rho}^{(i)}), \boldsymbol{a}^*\right), \qquad i = 1, \ldots, n. \tag{3}$$

The Proposer is rewarded according to the Solver's failure rate,

$$r_{\text{propose}} = 1 - \frac{1}{n} \sum_{i=1}^{n} r_{\text{solve}}^{(i)}. \tag{4}$$

SSP optimizes the Solver with GRPO using the relative rewards within each rollout group, while the Proposer is optimized with REINFORCE using r_{propose}.

4 METHODOLOGY

Our method builds on the Proposer–Solver self-play framework of SSP. On the Proposer side, we construct self-play training data by grounding sampled KG paths in the target corpus and using the retained paths for online question generation and filtering. On the Solver side, the verified corpus evidence is used to provide process-level supervision, rewarding both evidence acquisition and evidence utilization. Next, we describe these two components in turn.

4.1 EVIDENCE-GROUNDED DATA SYNTHESIS

KG Path Sampling. We begin by sampling different multi-hop paths from a knowledge graph $\mathcal{G}$. Formally, let $P = (e_0, r_1, e_1, \ldots, r_K, e_K)$ denote a sampled path, where the terminal entity e_K serves as the target answer $\boldsymbol{a}^*$. We discard paths containing cycles, repeated entities, noisy relations, or overly generic relations.

Corpus Evidence Grounding. For each KG triplet $z_k = (e_{k-1}, r_k, e_k)$, we prompt an LLM (prompt template in Appendix B) to generate multiple retrieval queries and issue them to the target corpus. The retrieved passages are merged and deduplicated into a candidate pool, and each candidate is subsequently verified against z_k for relational support. We retain up to the top-3 verified passages as the evidence $\mathcal{E}_k = \{d_k^j\}, 1 \le j \le 3$ for each hop, ranked by their retriever similarity scores. Paths for which any hop lacks verified evidence are discarded.

Curriculum Scheduling. We construct the curriculum by presenting the retained KG paths to the Proposer in descending order of hop count, progressing from longer to shorter paths. Longer paths contain more intermediate entities and support a greater diversity of relational compositions, offering the Proposer more candidate structures for question synthesis early in training. As training proceeds, shorter paths provide fewer relational cues and require the Proposer to formulate questions from more limited structural context.

Question Generation and Filtering. For each retained KG path P, we take the terminal entity e_K as the target answer and provide P to the Proposer as relational context. Following SSP and KG-Path (Lu et al., 2026; Wu et al., 2026), the Proposer can invoke the search tool and perform multi-turn search and reasoning before producing a candidate question $\boldsymbol{q} = \mathcal{Q}(\boldsymbol{\tau})$. The KG path provides explicit relational structure for question synthesis, while the Proposer's search trajectory supplies additional context needed to formulate the final question. We then apply the same rule-based filtering and RAG-based verification to ensure that the synthesized questions are valid and answerable. Because every triplet in P is supported by retrievable corpus evidence, the Proposer starts from a more reliable factual basis for question synthesis. Consistent with this design, we observe a higher pass rate under the same verification procedure, as reported in Table 3.

However, passing the standard validity and answerability checks does not necessarily ensure that solving the question requires the intended multi-step search process. A synthesized question may still be substantially easier than the KG path suggests, for example when the Solver can answer it directly from parametric knowledge, when a single retrieval step is sufficient to reach the target answer, or when intermediate entities are explicitly revealed and thereby bypass part of the search chain. We therefore introduce additional difficulty-oriented filtering to remove these cases.

More specifically, we first ask the Solver to answer the question without access to the search tool and remove questions that can be reliably solved from parametric knowledge. For the remaining questions, we restrict the Solver to a single search interaction and discard those for which one retrieval step is sufficient to recover the target answer. We additionally check whether the synthesized question explicitly contains the names or aliases of intermediate entities in P, as such leakage can bypass intermediate retrieval steps. These checks remove questions with substantially simpler solution paths while placing no constraint on the trajectory used to solve the retained questions.

4.2 Evidence-Based Process Supervision

Instead of using the occurrence of path entities in the reasoning trace as a proxy for search progress, we directly supervise whether the Solver retrieves evidence supporting each relation and whether the acquired evidence is subsequently used to advance the search. For the i-th Solver rollout $\boldsymbol{\rho}^{(i)}$, let $\boldsymbol{s}_t^{(i)}$ denote the query issued at the t-th search step and $\mathcal{D}_t^{(i)}$ the passages returned by the search tool. Since the verified evidence and the Solver observations come from the same corpus, we determine evidence acquisition by passage identity. Specifically, let $\mathrm{ID}(\mathcal{S}) = \{\mathrm{id}(d) \mid d \in \mathcal{S}\}$ denote the set of unique identifiers associated with the passages d in corpus $\mathcal{S}$. We define a binary indicator for whether the retrieval at search step t contains verified evidence supporting triplet z_k:

$$h_{k,t}^{(i)} = \begin{cases} 1, & \text{if } \mathrm{ID}\left(\mathcal{D}_t^{(i)}\right) \cap \mathrm{ID}(\mathcal{E}_k) \neq \varnothing, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

We further define the cumulative evidence coverage for triplet z_k up to search step t as

$$c_{k,t}^{(i)} = \max_{1 \leq t' \leq t} h_{k,t'}^{(i)}. \tag{6}$$

Thus, $c_{k,t}^{(i)} = 1$ if the Solver has retrieved at least one verified evidence for z_k by step t.

Evidence Acquisition Reward. We quantify evidence acquisition by the coverage of evidence over the complete Solver rollout. Let T_i denote the number of search steps in $\boldsymbol{\rho}^{(i)}$. We define

$$r_{\mathrm{acq}}^{(i)} = \frac{1}{K} \sum_{k=1}^{K} c_{k,T_i}^{(i)}. \tag{7}$$

The reward measures the fraction of KG triplets for which the Solver retrieves at least one verified supporting passage. Each triplet contributes at most once, irrespective of how many supporting passages are retrieved, and evidence may be acquired in any order.

Evidence Utilization Reward. Evidence acquisition measures whether the Solver retrieves verified supporting evidence, but does not capture whether newly acquired evidence contributes to subsequent search progress. We therefore introduce an *evidence utilization* reward that captures local transitions in which evidence retrieved at one search step provides a new intermediate entity that is immediately used to retrieve verified evidence for an adjacent KG triplet.

Specifically, consider two adjacent triplets $z_k = (e_{k-1}, r_k, e_k)$ and $z_{k+1} = (e_k, r_{k+1}, e_{k+1})$, which share the intermediate entity e_k. We set $u_k^{(i)} = 1$ if there exist two consecutive search steps t and $t+1$ satisfying all of the following conditions: (1) step t retrieves verified evidence for one of the two triplets for the first time, and step $t+1$ retrieves verified evidence for the other for the first time; (2) the evidence retrieved at step t contains e_k, while neither the canonical name nor any alias of e_k appears in the question or in any preceding search observation; and (3) the subsequent query $\boldsymbol{s}_{t+1}^{(i)}$ explicitly contains the canonical name or an alias of e_k. Otherwise, $u_k^{(i)} = 0$. The evidence utilization reward is therefore denoted as:

$$r_{\text{util}}^{(i)} = \frac{1}{K-1} \sum_{k=1}^{K-1} u_k^{(i)}. \tag{8}$$

Each adjacent triplet pair contributes at most once, preventing repeated retrieval of the same evidence from accumulating additional reward. Entity matching uses normalized canonical names and aliases from the KG. The two triplets may be retrieved in either order, so $r_{\text{util}}^{(i)}$ rewards local evidence-driven progress without enforcing the direction or full ordering of the KG construction path.

Overall Solver Reward. We use evidence-based rewards only to differentiate incorrect trajectories, while all correct trajectories receive the same full reward. This avoids favoring a particular verified evidence path when multiple valid search strategies may lead to the correct answer. Among incorrect trajectories, evidence acquisition rewards retrieving task-relevant evidence, and evidence utilization further rewards using that evidence to make subsequent search progress. Specifically, let $r_{\text{ans}}^{(i)} = r\left(\mathcal{A}(\boldsymbol{\rho}^{(i)}), \boldsymbol{a}^*\right) \in \{0, 1\}$ denote whether the final answer of the i-th Solver rollout is correct. The overall Solver reward is defined as:

$$R_{\text{solve}}^{(i)} = \begin{cases} 1, & r_{\text{ans}}^{(i)} = 1. \\ \lambda_{\text{acq}}\, r_{\text{acq}}^{(i)} + \lambda_{\text{util}}\, r_{\text{util}}^{(i)}, & r_{\text{ans}}^{(i)} = 0. \end{cases} \tag{9}$$

where λ_{acq} and λ_{util} are the weighting coefficients, respectively.

5 Experiments

5.1 Experimental Setup

Datasets. Following prior work, we evaluate our method on widely used knowledge-intensive QA benchmarks. We consider Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023) for General QA, and HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023) for multi-hop QA. For training data construction, we use Wikidata (Vrandečić & Krötzsch, 2014) as the source knowledge graph and apply relation-level blocklist and allowlist filtering to remove overly generic or noisy relations. All KG paths are extracted once during offline preprocessing and remain fixed throughout training.

Baselines. For a fair comparison, we compare our method with standard SSP (Lu et al., 2026) and KG-Path (Wu et al., 2026) under strictly matched training settings. Specifically, to examine the robustness of our method across different starting points, we evaluate five model configurations spanning different model families, parameter scales, and initialization types. These configurations

Method	GeneralQA			Multi-HopQA				Avg
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Training from Base and Instruct Initializations								
Qwen2.5-7B-Base	19.8	24.0	18.2	11.6	5.8	4.8	23.2	15.3
+ SSP	**38.8**	**59.8**	**49.6**	30.0	19.6	12.2	24.8	33.5
+ KG-path	37.0	52.8	41.8	31.6	32.4	15.6	35.2	35.2
+ Ours	38.1	57.4	46.5	**33.4**	**35.3**	**15.8**	**37.0**	**37.6**
Qwen2.5-7B-Instruct	28.6	49.2	31.0	28.8	27.0	10.8	36.0	30.2
+ SSP	35.4	**57.6**	**42.8**	32.2	30.4	**15.4**	34.4	35.5
+ KG-path	35.2	56.8	40.8	33.0	32.2	12.8	40.0	35.8
+ Ours	**36.1**	57.3	40.6	**34.6**	**35.3**	15.0	**43.7**	**37.5**
Generalization Across Model Families								
LLaMA-3.1-8B	31.2	53.2	39.2	19.4	10.8	8.6	20.8	26.2
+ SSP	**38.4**	60.8	45.4	32.6	24.0	11.4	23.2	33.7
+ KG-path	37.0	60.4	46.0	30.2	28.8	10.4	**25.6**	34.1
+ Ours	38.0	**62.1**	**46.2**	**34.1**	**30.7**	**11.7**	25.4	**35.5**
Qwen3-8B	29.0	62.0	40.6	34.2	40.6	15.2	44.0	37.9
+ SSP	**32.2**	**65.2**	45.0	40.6	42.2	17.0	50.4	41.8
+ KG-path	31.6	64.4	44.8	42.0	43.2	18.8	49.6	42.1
+ Ours	31.7	65.1	**45.3**	**43.5**	**45.0**	**20.1**	**52.0**	**43.2**
Continual Training on Search Agent								
Search-R1-7B	**48.0**	61.2	52.0	43.6	38.6	21.0	42.4	43.8
+ SSP	47.8	63.2	**52.4**	44.0	40.4	21.6	46.4	45.1
+ KG-path	46.6	64.0	52.0	42.8	42.8	21.6	47.2	45.3
+ Ours	47.0	**64.5**	52.3	**45.2**	**43.1**	**21.8**	**49.8**	**46.2**

Table 1: Exact Match (EM) results comparing SSP, KG-Path, and our method under matched experimental settings across different initializations, model families, and continued-training settings. The best score within each group is shown in **bold**.

cover three complementary settings: training from relatively weak pretrained models (Qwen2.5-7B-Base and Qwen2.5-7B-Instruct (Qwen et al., 2025)), generalization across model families (LLaMA-3.1-8B Grattafiori et al. (2024) and Qwen3-8B (Yang et al., 2025)), and continued training from existing search agent checkpoints (Search-R1-7B (Jin et al., 2025)).

Training and Evaluation Setting. We use a local E5 retriever (Wang et al., 2024a) over the Wiki-2018 corpus (Karpukhin et al., 2020) as the search tool, retrieving the top-3 documents for each query. The same search setup is used for both training and evaluation, with a maximum of 10 search calls per trajectory. Since the ground-truth answers in the evaluated benchmarks are mainly short-form, we report Exact Match (EM) and F1 score as the primary evaluation metrics. We provide additional implementation details in Appendix C and the prompts in Appendix B.

5.2 Main Results

The main experimental results are reported in Tables 1 and 5. Our method achieves the highest average EM and F1 across all five training settings, spanning base and instruction-tuned initializations, different model families, and continued training from existing search agent. The gains are largest when starting from the weaker Qwen2.5-7B-Base initialization and become more modest for stronger starting points such as Qwen3-8B and Search-R1-7B, while remaining positive in every setting. This pattern suggests that the proposed training framework provides consistent benefits across different initial capabilities, with less headroom once the underlying model already exhibits stronger search performance.

The improvement is most pronounced on multi-hop QA. Across the five training settings and four datasets, our method achieves the best EM and F1 in 18 of 20 comparisons. KG-Path already improves substantially over SSP on many multi-hop tasks, while offering smaller or occasionally negative gains on GeneralQA, suggesting that KG-based supervision is particularly useful when solving the question requires structured multi-step retrieval. Building on this setting, our method further improves performance on most multi-hop benchmarks, with especially consistent gains on 2Wiki-MultiHopQA and Bamboogle. This trend is consistent with the intended role of evidence-grounded

process supervision in supporting search over dependent pieces of information; the specific contributions of evidence acquisition and utilization are examined separately in the ablation study.

The results on GeneralQA are more mixed and notably different from that on multi-hop QA. Unlike the consistent gains observed on multi-hop benchmarks, no method dominates across NQ, TriviaQA, and PopQA, particularly under the Qwen2.5-7B-Base initialization. One possible explanation is a difference in the induced training-task distribution: SSP may generate relatively shorter or less compositional questions that are better aligned with GeneralQA, while path-based synthesis places greater emphasis on multi-step relational structure. Such a difference could be particularly consequential for a weaker base model, whose learned search behavior may be more sensitive to the difficulty and structure of the training tasks. Since we do not directly control for training-question difficulty in the main comparison, we treat this as a possible contributing factor rather than attributing the GeneralQA gap solely to the supervision design. Overall, the primary advantage of our method lies in its more consistent improvements on multi-hop QA, while performance on GeneralQA remains broadly competitive.

	GeneralQA			Multi-HopQA				
Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	**Avg**
KG-path	37.0	52.8	41.8	31.6	32.4	<u>15.6</u>	35.2	35.2
Ours *w/ Entity*	<u>37.1</u>	53.3	43.1	31.6	33.0	15.0	<u>35.6</u>	35.5
Ours *w/ Acq*	**38.1**	<u>57.2</u>	<u>46.0</u>	<u>32.8</u>	<u>34.5</u>	<u>15.6</u>	**37.0**	<u>37.3</u>
Ours	**38.1**	**57.4**	**46.5**	**33.4**	**35.3**	**15.8**	**37.0**	**37.6**

Table 2: Ablation of data synthesis and reward design on Qwen2.5-7B-Base, evaluated with Exact Match (EM). We compare entity-based reward, evidence acquisition, and the full evidence-based reward. The best and second-best results are shown in **bold** and <u>underlined</u>, respectively.

5.3 Ablation Study

We further disentangle the contributions of evidence-grounded question generation and evidence-based reward design. Specifically, we consider three variants. *Ours w/ Entity* replaces our evidence-based reward with the entity-based waypoint reward used in KG-Path, computed over the intermediate entities along the retained KG path. Comparing this variant with KG-Path provides an approximate comparison of the two task construction pipelines under the same form of intermediate supervision. *Ours w/ Acq* replaces entity-based supervision with the evidence acquisition reward r_{acq}, while removing evidence utilization. This variant isolates the effect of supervising the retrieval of verified supporting evidence rather than the occurrence of path entities. Finally, *Ours* uses both evidence acquisition and evidence utilization rewards, corresponding to the complete method.

Tables 2 and 6 show a consistent progression across the ablation variants. Under entity-based intermediate supervision, *Ours w/ Entity* outperforms KG-Path on average. Since the two methods are trained on different underlying KG paths, the comparison does not isolate task construction alone, but it nevertheless supports the effectiveness of our evidence-grounded construction pipeline. Replacing entity-based supervision with evidence acquisition yields the largest improvement among the reward ablations, indicating that direct supervision of verified evidence retrieval is more effective than using entity occurrence as a proxy for search progress. Adding evidence utilization provides a further, smaller gain, particularly on multi-hop benchmarks, consistent with its role in rewarding whether newly acquired evidence is used to advance subsequent retrieval. Overall, the results suggest that evidence acquisition provides the primary benefit of the proposed reward design, while evidence utilization offers additional supervision for multi-step search.

5.4 Analysis of Proposer Capability

The effectiveness of self-evolving training depends critically on the quality of the tasks generated by the Proposer. We examine this capability from two perspectives: whether the Proposer can reliably generate questions that survive the training-time verification pipeline, and whether those questions provide useful supervision for training a new search agent. The former captures generation validity, while the latter measures the downstream training effectiveness of the resulting QA pairs.

Validity of Generated Questions. We first examine how reliably each Proposer produces valid training instances. Following the same generation protocol, each Proposer generates 20K trajectories from 10,000 shared seeds, with two rollouts per seed, and all trajectories are then processed by the same training-time filtering pipeline. As shown in Table 3, Qwen2.5-7B-Instruct and Search-R1-7B retain 13.9% and 22.3% of the generated trajectories, respectively. Self-play substantially improves this rate, reaching 55.4% for SSP and 61.8% for KG-Path. Our Proposer further increases the pass rate to 70.9%, yielding 14,174 valid QA pairs. This improvement is consistent with the role of evidence grounding in task construction. Rather than relying on KG structure alone, our pipeline filters out paths containing relations without verified support in the target corpus before they are used for question generation. As a result, the Proposer starts from relational structures that are better aligned with the available search environment, increasing the likelihood that the resulting questions satisfy the downstream validity and answerability checks.

Model	**Trajectories**	**Valid QA**	**Pass Rate (%)**
Qwen2.5-7B-Instruct	20K	2,774	13.9
Search-R1-7B	20K	4,458	22.3
Search-R1-7B + SSP	20K	11,077	55.4
Search-R1-7B + KG-Path	20K	12,357	61.8
Search-R1-7B + Ours	20K	14,174	70.9

Table 3: Statistics of the validity of generated questions.

Training Effectiveness of Generated Questions. To assess whether the advantage of our Proposer carries over to the training value of its generated questions, we conduct a controlled transfer experiment. Starting from the final Proposer of SSP and our method, we construct two equally sized datasets of 10K valid QA pairs using the same filtering procedure. Each dataset is then used to train a fresh Qwen2.5-7B-Base Solver under an identical search and optimization setup, with outcome reward as the only learning signal. This removes the influence of method-specific process rewards and allows the comparison to focus on the questions themselves. Table 4 shows that our generated questions lead to stronger Solvers on all four multi-hop benchmarks, raising the average F1 from 30.4 to 35.3. The gain persists even after evidence-based supervision is removed from downstream training, indicating that the benefit of our framework is reflected not only in how the Solver is trained, but also in the training value of the questions produced by the Proposer.

	Multi-HopQA				
Method	HotpotQA	2Wiki	MuSiQue	Bamboogle	**Avg**
Qwen2.5-7B-Base	17.0	10.1	9.0	28.9	16.3
+ SSP_{10K}	37.9	25.3	19.2	39.1	30.4
+ $Ours_{10K}$	**41.5**	**34.5**	**21.1**	**44.2**	**35.3**

Table 4: Training effectiveness of generated questions on multi-hop QA benchmarks. Each method samples 10K valid QA pairs generated by its final Proposer and trains a new Solver from the same Qwen2.5-7B-Base checkpoint using the same outcome-only reward and training setting.

6 Conclusion

We present Evidence-in-the-Loop, an evidence-centric framework for self-evolving search-agent training. Our central idea is to connect the relational structure provided by knowledge graphs with evidence that is actually available in the target text corpus. By grounding KG relations in verified, retrievable evidence, our framework establishes a more reliable basis for question generation and enables process supervision based on the information an agent actually retrieves and uses during search, rather than on entity occurrence alone. This design preserves the flexibility of open-domain search without requiring the agent to reproduce a fixed KG trajectory. Across model families and initialization settings, our framework consistently improves downstream QA performance, with particularly strong gains on multi-hop benchmarks. We further find that the resulting Proposer generates higher-quality questions that are more effective for training search agents. Together, these results highlight corpus evidence as a useful interface between structured task construction and search-process supervision, providing a promising direction for scalable self-evolving search agents.

REFERENCES

Anthropic. Claude takes research to new places, April 2025. Official product announcement, April 15, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9112–9141, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Unlocking long-horizon agentic search with large-scale end-to-end RL. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MfPDdPUGKi.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Xiang Huang, Jiayu Shen, Shanshan Huang, Sitao Cheng, Xiaxia Wang, and Yuzhong Qu. TARGA: Targeted synthetic data generation for practical reasoning over structured data. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2704–2726, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.137. URL https://aclanthology.org/2025.acl-long.137/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

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 Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Jiaqi Guo, Haotian Xu, Chutian Wang, Haonan Chen, xiaoxi jiang, and guanjunjiang. Search self-play: Pushing the frontier of agent capability without supervision. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZmGirmNJqE.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl. *arXiv preprint arXiv:2509.10446*, 2025.

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 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Moonshot AI. Kimi-Researcher: End-to-end rl training for emerging agentic capabilities, June 2025. Technical release, June 20, 2025.

OpenAI. Introducing deep research, February 2025. Official product announcement, February 2, 2025.

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

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings*

of the Twentieth European Conference on Computer Systems, EuroSys '25, pp. 1279–1297, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

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.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 641–651, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL https://aclanthology.org/N18-1059/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

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, 2024a. URL https://arxiv.org/abs/2212.03533.

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 (Volume 1: Long Papers)*, pp. 9426–9439, 2024b.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Huyu Wu, Jun Liu, Xiaochi Wei, Yan Gao, Yi Wu, and Yao Hu. Knowledge-graph paths as intermediate supervision for self-evolving search agents. *arXiv preprint arXiv:2605.05702*, 2026.

xAI. Grok 3 Beta—the age of reasoning agents, February 2025. Official product announcement, February 19, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Wenhao Yu, Zhenwen Liang, Chengsong Huang, Kishan Panaganti, Tianqing Fang, Haitao Mi, and Dong Yu. Guided self-evolving llms with minimal human supervision, 2025. URL https://arxiv.org/abs/2512.02472.

Zhenrui Yue, Kartikeya Upasani, Xianjun Yang, Suyu Ge, Shaoliang Nie, Yuning Mao, Zhe Liu, and Dong Wang. Dr. zero: Self-evolving search agents without training data. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=3UH5bILEBI.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Qianchun Lu. Learning like humans: Advancing LLM reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6619–6633, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.336. URL https://aclanthology.org/2025.emnlp-main.336/.

Mike Zhang, Johannes Bjerva, and Russa Biswas. Follow the path: Reasoning over knowledge graph paths to improve large language model factuality. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11574–11590, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.561. URL https://aclanthology.org/2026.findings-acl.561/.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=neZSGqhxDa.

Yida Zhao, Kuan Li, Xixi Wu, Liwen Zhang, Ding-Chu Zhang, Baixuan Li, Maojia Song, Zhuo Chen, Chenxi Wang, Xinyu Wang, et al. Repurposing synthetic data for fine-grained search agent supervision. In *International Conference on Learning Representations*, volume 2026, pp. 14467–14492, 2026.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: efficient execution of structured language model programs. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.22. URL https://aclanthology.org/2025.emnlp-main.22/.

Yifei Zhou, Sergey Levine, Jason E Weston, Xian Li, and Sainbayar Sukhbaatar. Self-challenging language model agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9yusqX9DpR.

AI USE STATEMENT

In this paper, we use Large Language Models (LLMs) as a general-purpose assistance tool for improving writing quality and accelerating the code implementation. Specific applications include: (1) Language and Grammar Correction (2) Paraphrasing and Style Refinement (3) Generating code snippets and providing debugging support to help implement the proposed algorithms and experimental setups. It should be noted that all core research concepts, experimental design, data analysis, and conclusions are developed exclusively by the human authors. Any content or suggestions generated by the LLM, including code, are critically checked, and substantially edited by the authors to ensure accuracy. The authors take full responsibility for the final content of this paper.

A ADDITIONAL EXPERIMENT RESULTS

	GeneralQA			Multi-HopQA				
Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	**Avg**
Training from Base and Instruct Initializations								
Qwen2.5-7B-Base	26.9	28.8	21.7	17.0	10.1	9.0	28.9	20.3
+ SSP	49.0	**69.4**	**56.4**	41.2	27.1	21.1	40.2	43.5
+ KG-path	47.0	63.9	49.2	45.7	40.7	23.1	47.9	45.4
+ Ours	**49.3**	66.8	52.0	**46.6**	**42.3**	**24.0**	**49.4**	**47.2**
Qwen2.5-7B-Instruct	37.6	58.9	36.4	40.3	33.6	18.4	50.0	39.3
+ SSP	45.2	66.5	49.9	44.4	39.1	**22.9**	44.3	44.6
+ KG-path	**46.0**	68.2	48.8	45.8	40.8	22.6	53.7	46.6
+ Ours	45.6	**70.4**	**50.3**	**46.7**	**41.5**	22.7	**54.4**	**47.4**
Generalization Across Model Families								
LLaMA-3.1-8B	39.7	61.9	45.3	29.8	17.7	13.2	28.5	33.7
+ SSP	48.4	69.1	50.7	43.6	29.7	17.5	34.7	42.0
+ KG-path	46.3	70.5	**51.2**	42.6	36.7	20.6	35.6	43.4
+ Ours	**48.8**	**71.0**	50.0	**43.8**	**38.0**	**21.2**	**36.7**	**44.2**
Qwen3-8B	37.3	71.5	48.0	48.1	50.8	23.2	57.1	48.0
+ SSP	42.0	74.2	53.3	52.4	51.2	25.3	61.3	51.4
+ KG-path	42.1	73.6	53.1	55.4	53.3	27.3	62.0	52.4
+ Ours	**42.7**	**74.5**	**53.6**	**56.0**	**53.8**	**28.1**	**64.2**	**53.3**
Continual Training on Search Agent								
Search-R1-7B	55.3	71.3	56.5	57.0	46.3	30.2	54.9	53.1
+ SSP	**55.4**	**73.6**	57.4	**57.1**	49.3	31.3	58.3	54.6
+ KG-path	55.1	73.1	57.4	56.7	51.0	32.7	59.2	55.0
+ Ours	**55.4**	73.3	**57.6**	56.7	**52.4**	**33.1**	**61.5**	**55.7**

Table 5: F1 score results comparing SSP, KG-Path, and our method under matched experimental settings across different initializations, model families, and continued-training settings. The best score within each group is shown in **bold**.

	GeneralQA			Multi-HopQA				
Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	**Avg**
KG-path	47.0	63.9	49.2	<u>45.7</u>	40.7	23.1	47.9	45.4
Ours *w/ Entity*	47.9	65.7	51.0	45.4	41.1	23.6	48.0	46.1
Ours *w/ Acq*	<u>49.0</u>	<u>66.2</u>	<u>51.9</u>	**46.6**	<u>42.1</u>	<u>23.7</u>	<u>48.8</u>	<u>46.9</u>
Ours	**49.3**	**66.8**	**52.0**	**46.6**	**42.3**	**24.0**	**49.4**	**47.2**

Table 6: Ablation of data synthesis and reward design on Qwen2.5-7B-Base, evaluated with F1 score. We compare entity-based reward, evidence acquisition, and the full evidence-based reward. The best and second-best results are shown in **bold** and <u>underlined</u>, respectively.

B PROMPT TEMPLATES

Prompts used in our experiments are listed as follows.

Proposer Prompt

System Prompt:
You are an expert multi-hop question creator for training a search agent. You will be given an ordered knowledge-graph path $e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_K$, where the terminal entity e_K is the target answer. Each relation on the path has already been verified to have supporting evidence that is retrievable from the same corpus available to the search agent. Your task is to construct **one challenging, fact-based, and uniquely answerable question** whose answer is the provided target answer. The KG path defines the intended relational dependencies underlying the question, but neither the path nor its intermediate entities should be directly exposed to the solver.
Core Objective.
The final question should require the solver to resolve multiple dependent pieces of information through search. It should not merely combine several independent attributes of the target answer. Instead, information obtained from an earlier retrieval step should be useful for identifying what needs to be searched next.

Question Construction Process.
1. Understand the relational path. First inspect the complete KG path and identify: (i) the starting entity e_0, (ii) the intermediate bridge entities $e_1, \ldots, e_{K-1}$, and (iii) the target answer e_K. Determine how each relation connects one entity to the next and formulate an intended sequence of information discovery from e_0 toward e_K.
2. Plan a dependent search chain. Translate the KG path into a natural sequence of search dependencies. For each intermediate entity, ask what information a solver would need to recover it from the preceding step and how that entity could subsequently lead to the next step. The resulting question should preserve these dependencies rather than presenting all clues independently. For example, avoid constructing a question of the form "Which entity has properties A, B, and C?" when all properties can be searched together. Prefer a question in which resolving one clue identifies an entity whose information is needed to resolve the next clue.
3. Use search to construct reliable clues. You MUST use the search tool before producing the final question. Search for concrete and verifiable information about entities and relations along the KG path, such as distinctive events, roles, affiliations, dates, locations, works, or other attributes that can be expressed as natural-language clues. Every factual clue added to the final question must be supported by retrieved information. Do not invent facts or rely on uncertain background knowledge. Prefer clues that are sufficiently distinctive to identify the intended entity but do not directly reveal its canonical name.
4. Hide intermediate entities. Do NOT mention the target answer e_K in the question. Do NOT directly reveal the canonical names or aliases of intermediate entities $e_1, \ldots, e_{K-1}$. Intermediate entities should instead be referred to through descriptive clues that can be recovered through search. The starting entity e_0 may be mentioned explicitly when it provides a natural entry point to the question; otherwise, it may also be described indirectly.
Do not copy KG relation labels verbatim when doing so would expose the intended path.
5. Prevent search shortcuts. The question must not be answerable through one obvious search query. In particular:

- Do not directly combine a named entity with the final relation leading to the target answer.
- Do not include an intermediate entity together with the exact relation needed to reach the next entity.
- Do not use highly distinctive verbatim sentences, long quotations, identifiers, URLs, or other strings that can be copied directly into a search engine to reveal the answer.
- Do not reduce a multi-hop path to a flat list of independent properties of the target answer.

The intended solution should require the solver to obtain new information during search and use that information to determine subsequent search actions.
6. Ensure unique answerability. All clues must jointly identify the provided target answer unambiguously. If a clue could refer to multiple entities, use additional corpus-supported information to disambiguate it. However, do not resolve ambiguity by explicitly revealing intermediate entities. The final answer must remain exactly the provided target answer and must match the specified answer type.
7. Preserve trajectory flexibility. The KG path is used to construct the relational dependencies of the question, but it is not a required search trajectory. Do not write the question in a way that explicitly instructs the solver to follow the KG path step by step. A solver may reach the correct answer through any valid sequence of searches.

Self-Check Before Finalizing. Before returning the final question, verify all of the following:

- The question has the provided target answer as its unique answer.
- All factual clues are supported by information obtained through search.
- The question requires multiple dependent information-discovery steps.
- The target answer and intermediate entities are not explicitly revealed.
- The question does not admit an obvious single-search shortcut.
- The question does not collapse into a conjunction of independent clues about the target answer.
- The wording is natural, concise, self-contained, and unambiguous.

If any condition is violated, revise the question before producing the final output.

Tool and Output Format. You must conduct reasoning inside <think> and </think> whenever you receive new information. If additional information is needed, call the search engine using <search> query </search>. The search results will be returned inside <information> and </information>. You may perform multiple searches as needed. When sufficient information has been collected and all requirements are satisfied, output only the final question inside <question> and </question>. Do not include the target answer, KG path, intermediate entities, explanations, options, or any other text in the final output.
User Prompt:
Construct one multi-hop search question from the following verified KG path.
Target Answer: {target_answer}
KG Path: {kg_path}
The path is ordered from the starting entity to the target answer. Its relations have already been verified to have supporting evidence in the target corpus. Use the path as the latent relational structure of the question and use corpus search to construct natural and discriminative clues. Return exactly one question whose unique answer is the target answer.

Solver Prompt

System Prompt: You are a helpful and harmless assistant.

User Prompt: Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. You can search as many times as you want. If you find that no further external knowledge is needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> xxx </answer>. Question: {question}

Evidence Retrieval Query Generation Prompt

Role. You are a retrieval-query generator for corpus evidence grounding.

Goal. Given a knowledge graph triplet, generate search queries that are likely to retrieve passages from a Wikipedia corpus that explicitly state or clearly support the corresponding fact.

Knowledge Graph Triplet.

- Head entity: {`head_entity`}
- Head aliases: {`head_aliases`}
- Relation: {`relation`}
- Relation description: {`relation_description`}
- Tail entity: {`tail_entity`}
- Tail aliases: {`tail_aliases`}

Generate exactly {`num_queries`} diverse retrieval queries for finding textual evidence supporting:

({`head_entity`}, {`relation`}, {`tail_entity`}).

Requirements.

1. Every query must target evidence for the specified relation between the two entities. Do not broaden the query to general information about either entity.
2. Preserve the semantics of the relation. You may paraphrase the relation in natural language, but do not replace it with a related, broader, or weaker relation.
3. Use complementary formulations across queries, including:
   - formulations centered on the head entity;
   - formulations centered on the tail entity when a natural inverse expression exists;
   - alternative natural-language expressions of the relation; and
   - supplied entity aliases when useful for retrieval.
4. Most queries should mention both endpoint entities. A query may focus on only one endpoint when this is likely to retrieve a passage in which the other endpoint appears as the value of the target relation.
5. Keep queries concise and retrieval-oriented. Prefer short natural-language phrases or declarative formulations over long questions or explanations.
6. Use only information provided in the triplet, entity aliases, and relation description. Do not introduce additional dates, locations, occupations, events, or other facts.
7. Do not invent entity aliases or uncertain relation paraphrases.
8. Avoid duplicate or near-duplicate queries. Each query should provide a distinct lexical formulation for retrieving evidence for the same fact.
9. Do not answer the fact, explain your reasoning, or generate evidence. Output retrieval queries only.

Output Format.
Return only:

```
{
  "queries": [
    "...",
    "...",
    ...
  ]
}
```

C Implementation Details

We use Wikidata as the source graph and perform extraction in QA mode. We sample 50,000 seed nodes, generating one 3–7-hop target path per seed. Labeled attributes are included when available, and relation expansion is guided by an LLM selector over up to 10 candidate relations per step, with at most five retries. After filtering and post-processing, the resulting subgraphs form the training set of 50,000 instances used in the main experiments.

We implement our method with SGLang (Zheng et al., 2024) for asynchronous multi-turn tool-integrated rollouts in VeRL (Sheng et al., 2025). For a controlled comparison with SSP, we optimize the Proposer with REINFORCE (Williams, 1992) and the Solver with GRPO (Shao et al., 2024), while matching the main rollout and optimization settings. Process rewards are used only during training, whereas validation relies solely on final-answer correctness. We use the same search environment and RAG-based verification throughout, with a fixed maximum number of interaction rounds. Unless otherwise specified, we use a learning rate of 1×10^{-6} with 5 warmup steps and a global batch size of 256. The maximum prompt and response lengths are 4,096 and 8,192 tokens, respectively, and each training run lasts approximately 150–200 optimization steps. The main hyperparameter settings are summarized in Table 7.

Table 7: Key experimental hyperparameter configuration.

Parameter	**Value**
— *Basic* —	
Global Batch Size	256
Max Prompt Length	4096
Max Response Length	8192
Learning Rate	1e-6
Learning Rate Warmup Steps	5
KL Loss Coefficient	0.01
Validation Temperature	0.0
Rollout Temperature	1.0
Use RAG Verification	True
Noisy RAG Documents	4
— *Proposer* —	
Proposer Advantage Estimator	REINFORCE
Proposer Samples	1
— *Solver* —	
Solver Advantage Estimator	GRPO
Solver Samples	5
— *Reward Coefficient* —	
λ_{acq}	0.7
λ_{util}	0.1

D Case Study

We present representative failure cases to examine where search trajectories break down despite containing apparently relevant intermediate information. Each case highlights a distinct search failure that is weakly distinguished by outcome- and entity-based supervision. Together, the examples illustrate the gap between mentioning relevant entities and making effective progress through evidence acquisition and utilization.

D.1 Search Drift after Identifying the Correct Bridge Entity

The Solver correctly identifies *Richard J. Lewis* as the director of *K-9: P.I.* in the first retrieval, recovering the key intermediate entity needed to solve the question. However, it does not use this information to formulate the next search around Lewis's birthplace. Instead, subsequent queries drift toward the film's production background and general Canadian film context, eventually leading to an

unsupported inference of Vancouver. This trajectory highlights a limitation of entity-level progress signals: the presence of the correct intermediate entity in the reasoning trace does not indicate whether that information actually influences subsequent search. A supervision signal based only on entity occurrence may therefore treat the trajectory as making meaningful intermediate progress even though the retrieved information is not productively used to resolve the remaining question.

Case Study on 2Wiki: Search Drift on Multi-Hop QA

Question
What is the place of birth of the director of film *K-9: P.I.*?

Ground Truth
Toronto

Solver Trajectory
<think> I need to find the birthplace of the director of *K-9: P.I.*. I should first identify who directed the film. </think>
<search> K-9 P.I. director </search>
<information> *Doc 1 (Title: "K-9: P.I."):* K-9: P.I. is a 2002 film directed by Richard J. Lewis and starring James Belushi.
Doc 2 (Title: "K-9"): K-9 is a 1989 American buddy-cop comedy film starring James Belushi.
Doc 3 (Title: "James Belushi"): James Belushi is an American actor known for several film and television roles. </information>
<think> The director is Richard J. Lewis. Now I need to determine where he was born.
I remember that Richard J. Lewis is a Canadian film and television director. Since *K-9: P.I.* also had Canadian production connections, perhaps his career was centered in Vancouver. I should investigate the film's Canadian production background first. </think>
<search> K-9 P.I. Canadian production filming location </search>
<information> *Doc 1 (Title: "K-9: P.I."):* The film was produced as a direct-to-video continuation of the K-9 series.
Doc 2 (Title: "Film production in British Columbia"): British Columbia has hosted many American and Canadian film productions.
Doc 3 (Title: "Vancouver film industry"): Vancouver is a major center for film and television production in Canada. </information>
<think> These results provide general information about Canadian film production, but they do not establish where Richard J. Lewis was born.
I should look for biographical information about his career that might identify where he is from. </think>
<search> Richard J. Lewis Canadian television director career </search>
<information> *Doc 1 (Title: "Richard J. Lewis"):* Richard J. Lewis is a Canadian film and television director whose credits include several television dramas.
Doc 2 (Title: "Canadian television directors"): Canadian television directors have worked across productions in Toronto, Vancouver, and the United States.
Doc 3 (Title: "CSI: Crime Scene Investigation"): The series employed numerous directors during its run. </information>
<think> This confirms that Richard J. Lewis is Canadian, but it still does not identify his birthplace.
Toronto and Vancouver are both plausible. Since he worked extensively in television, perhaps he was associated with Vancouver. </think>
<search> Richard J. Lewis Vancouver Canadian director </search>
<information> *Doc 1 (Title: "Canadian television production"):* Vancouver is one of Canada's major television production centers.
Doc 2 (Title: "Richard Lewis director"): Several people named Richard Lewis have worked as directors and producers.
Doc 3 (Title: "Vancouver filmmakers"): A number of Canadian filmmakers have worked in Vancouver. </information>
<think> The results remain ambiguous and still do not state where Richard J. Lewis was born.

Maybe the film itself has a stronger connection to Toronto. I should check whether *K-9: P.I.* was produced or filmed there. </think>
<search> K-9 P.I. Toronto filming </search>
<information> *Doc 1 (Title: "K-9: P.I."):* K-9: P.I. is the third installment in the K-9 film series.
Doc 2 (Title: "Toronto film production"): Toronto is one of Canada's largest centers for film and television production.
Doc 3 (Title: "Canadian film locations"): Many American productions have been filmed in Canadian cities. </information>
<think> I still have not found a source that explicitly states the birthplace of Richard J. Lewis.
The retrieved information repeatedly points to Canadian film and television production, and Vancouver is a major production center. I will therefore choose Vancouver. </think>
<answer> Vancouver </answer>

D.2 Correct Entities but Relation-Mismatched Retrieval

The Solver correctly identifies both directors, and its reasoning contains several entities that are highly relevant to the question, including *Stuart Rosenberg*, *Jean-Daniel Pollet*, *American*, and *French*. Nevertheless, the crucial retrievals establish the countries of the two *films*, whereas the question requires the nationalities of the two *directors*. The trajectory therefore accumulates substantial entity-level relevance without resolving the relation needed for the final comparison. This failure exposes a fundamental ambiguity in entity-based supervision: matching a relevant entity does not specify which relation involving that entity has been established. Consequently, a trajectory can receive substantial intermediate credit while retrieving facts that are topically related but insufficient for answering the question.

Case Study on 2Wiki: Relation-Mismatched Search

Question
Are director of film *Move* (1970 Film) and director of film *Méditerranée* (1963 Film) from the same country?

Ground Truth
No

Solver Trajectory
<think> The question asks whether the directors of the two films are from the same country. I first need to identify the director of each film. </think>
<search> Move 1970 film director </search>
<information> *Doc 1 (Title: "Move (1970 film)"):* Move is a 1970 comedy film directed by Stuart Rosenberg.
Doc 2 (Title: "Stuart Rosenberg"): Stuart Rosenberg was a film and television director known for *Cool Hand Luke* and other films.
Doc 3 (Title: "Move"): Move may refer to several films and other works. </information>
<think> The first director is Stuart Rosenberg. I now need to identify the director of *Méditerranée*. </think>
<search> Méditerranée 1963 film director </search>
<information> *Doc 1 (Title: "Méditerranée (1963 film)"):* Méditerranée is a 1963 experimental film directed by Jean-Daniel Pollet.
Doc 2 (Title: "Jean-Daniel Pollet"): Jean-Daniel Pollet was associated with post-war French cinema.
Doc 3 (Title: "French experimental cinema"): Experimental cinema developed strongly in France during the twentieth century. </information>
<think> The two directors are Stuart Rosenberg and Jean-Daniel Pollet. I now need to determine whether they are from the same country.
Move is an American film, so I should check its country first. </think>

<search> Move 1970 film country </search>
<information> *Doc 1 (Title: "Move (1970 film)"):* Move is a 1970 American comedy film.
Doc 2: The film was released in the United States in 1970.
Doc 3: Move stars American actor Elliott Gould. </information>
<think> Move is American, which suggests that Stuart Rosenberg was American.
Now I should check the country of Méditerranée. </think>
<search> Méditerranée 1963 film country </search>
<information> *Doc 1 (Title: "Méditerranée (1963 film)"):* Méditerranée is a 1963 French experimental film.
Doc 2: The film is associated with French experimental cinema.
Doc 3: Jean-Daniel Pollet worked extensively in French cinema. </information>
<think> The films are American and French respectively, but this does not necessarily mean that their directors had different nationalities. Both directors worked across international productions, so the nationality of the films may not directly determine where the directors themselves were from.
Perhaps they shared the same nationality despite working in different national film industries. </think>
<search> Stuart Rosenberg Jean-Daniel Pollet international directors </search>
<information> *Doc 1:* Stuart Rosenberg worked across film and television productions in several contexts.
Doc 2: Jean-Daniel Pollet was associated with French and European experimental cinema.
Doc 3: Directors often participate in international film productions. </information>
<think> Both had international careers, and the retrieved results do not directly establish their nationalities. Since their film careers crossed national boundaries, I think they may have been from the same country. </think>
<answer> Yes </answer>