DR. FREE: YOU DON'T NEED DIFFICULTY REWARDS FOR SELF-EVOLVING SEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A central limitation of current data-free self-evolution methods for training search agents is their reliance on difficulty-based proposer rewards. These methods reward a proposer for generating questions that challenge a co-evolving solver, using solver difficulty as a proxy for question quality. Yet difficulty alone is insufficient to distinguish questions that require cross-passage evidence from those that are answerable via simpler shortcuts. In addition, measuring difficulty demands repeated solver rollouts for every candidate question, leading to substantial computational costs. In this paper, we introduce Dr. Free, the first self-evolving search framework that eliminates difficulty-based proposer rewards and directly optimizes for evidence necessity relative to shortcut contexts. Dr. Free samples relational chains from a knowledge graph and pairs them with aligned passages, giving question generation an explicit multi-hop structure. A generated question receives a positive information-gain reward only when the likelihood of the target answer under the complete evidence passages exceeds the maximum likelihood under all evaluated shortcut contexts. Because this signal is computed from teacher-forced likelihoods, it removes the need for pass-rate estimation and reduces proposer training time by over $7\times$. Experiments on seven open-domain QA benchmarks show that Dr. Free outperforms prior data-free search agents and the supervised baseline, with large improvements on multi-hop QA benchmarks.

1 INTRODUCTION

Training search agents typically requires curated question-answer pairs, incurring high data costs (Jin et al., 2025; Kwiatkowski et al., 2019; Yang et al., 2018; Trivedi et al., 2023; Li et al., 2025; Qian et al., 2026).Self-evolving methods reduce this dependence through proposer-solver self-play, in which a proposer continually generates questions, a solver learns to answer them, and solver performance in turn provides the training signal for subsequent question generation, enabling the two models to co-evolve over successive rounds (Zhao et al., 2026; Huang et al., 2025). Search Self-Play (Lu et al., 2025a) generates multi-turn search questions sampled from public-QA datasets and filters the collected evidence with RAG, whereas Dr. Zero (Yue et al., 2026) removes curated QA pairs by generating questions from corpus passages and rewarding proposer difficulty. Subsequent methods incorporate knowledge graphs for question construction (Wu et al., 2026; Tan et al., 2026) or reward the marginal utility of grounded evidence (Arai & Ichikawa, 2026). Despite these advances, these methods still rely on solver difficulty as the primary training signal, even though difficulty does not reveal whether a question genuinely requires the intended evidence.

The core issue is that solver difficulty and evidence necessity measure fundamentally different properties of a question-answer pair. Difficulty is inherently solver-relative: It indicates whether the current model can produce a correct answer (Yue et al., 2026), rather than assessing the construction quality of the question itself. This distinction is particularly important for multi-hop QA, where models can often reach the correct answer through reasoning shortcuts or disconnected evidence rather than the intended reasoning chain (Jiang & Bansal, 2019; Trivedi et al., 2020; Guo et al., 2023; Li et al., 2026b). A well-constructed multi-hop question should instead require the integration of information across multiple passages, such that no individual passage alone provides sufficient evidence for the answer (Ho et al., 2020; Trivedi et al., 2022; Press et al., 2023; Feng et al., 2026).

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.

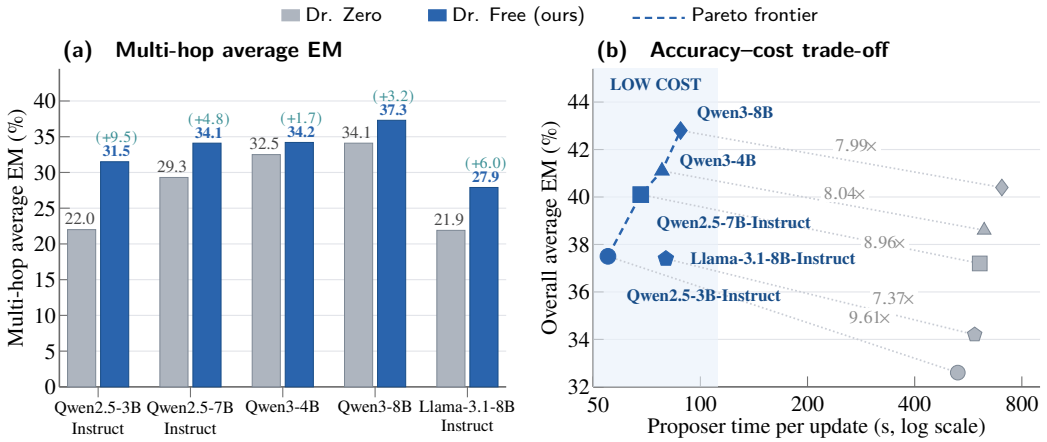


Figure 1: **Dr. Free outperforms Dr. Zero across five backbones while training faster.**

Our audit of Dr. Zero reproductions further confirms this failure mode in practice. Across two backbone models, more than 80% of questions labeled as multi-hop are generated without search, and the target answer for most questions already appears in the seed passage (Table 4). As shown in Figure 2, a representative Dr. Zero trace illustrates this mismatch. The question is labelled as two-hop, although the proposer performs no search and the target answer is directly available in the seed passage. Because the difficulty reward is computed solely from solver rollout outcomes, it cannot detect this evidence shortcut. Consequently, hop labels often reflect the requested generation format rather than demonstrated cross-document evidence composition, leaving the resulting question-answer pairs susceptible to single-passage shortcuts. Moreover, solver difficulty introduces substantial computational overhead because its estimation requires multiple multi-turn solver rollouts for each generated question. Together, these limitations motivate replacing solver difficulty with a training signal that can efficiently determine whether the full evidence context increases the likelihood of the target answer relative to shortcut-only contexts.

To address these limitations, we introduce Dr. Free, which changes both how training questions are constructed and how the proposer is rewarded (Figure 2).Whereas Dr. Zero relies on proposer-side search to construct questions from seed passages, Dr. Free provides relational chains from Wikidata5M together with aligned Wikipedia passages as explicit inputs for question construction. The proposer therefore starts from a predefined evidence structure and generates questions without online search. In contrast to difficulty-based rewards, which evaluate questions according to the performance of the current solver, our reward measures how much the full evidence context increases the likelihood of the target answer relative to the strongest evaluated shortcut context, including closed-book, source-only, one-shot retrieval, and individual-passage contexts. This formulation shifts proposer optimization from solver-relative difficulty to evidence dependence under explicit shortcut comparisons. Importantly, the proposed objective completely replaces rollout-based difficulty rather than supplementing it with an additional reward signal. By computing the likelihood of a fixed target answer under teacher forcing, Dr. Free eliminates solver rollouts from proposer reward estimation. Combined with structured question construction, this design substantially reduces the training time of the proposer per update.

As shown in Figure 1, Dr. Free improves both effectiveness and efficiency. Across five backbone models, it improves the average EM on multi-hop benchmarks over Dr. Zero by 1.7–9.5 points while accelerating proposer training by 7.37×–9.61× per update. Our contributions are as follows.

- We identify the mismatch problem between difficulty and evidence necessity and document severe pseudo-multi-hop degeneration in a representative self-evolving search agent.
- We introduce Dr. Free, featuring a rollout-free, shortcut-aware information-gain reward that directly measures the dependence of generated questions on the full evidence relative to evaluated shortcut contexts, promoting the construction of genuine multi-hop questions.
- Across seven QA benchmarks, Dr. Free improves multi-hop accuracy while reducing the proposer training time by more than 7×. Controlled ablations further confirm the benefits of structured question construction and the information-gain-based reward.

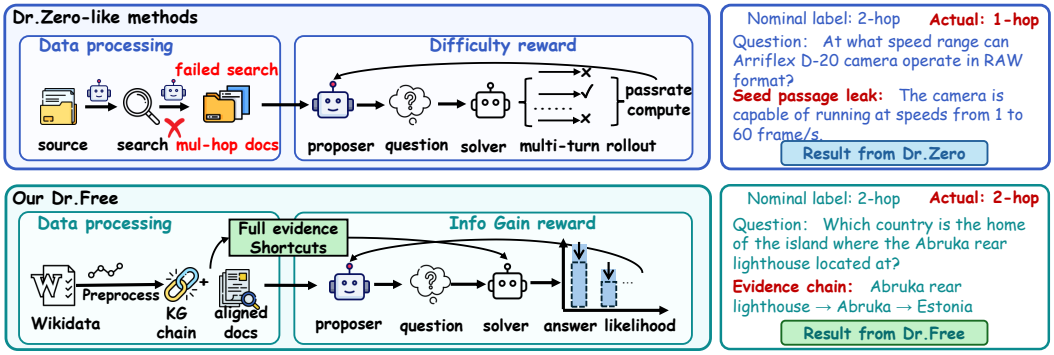


Figure 2: Comparison between Dr. Zero-like methods and Dr. Free. The key differences lie in question construction and proposer reward design. Dr. Zero-like methods construct questions through proposer-side search and optimize the proposer with rollout-based difficulty rewards, which may permit zero-search generation with target answers exposed in seed passages. In contrast, Dr. Free constructs questions from verified knowledge-graph chains with aligned passages and replaces difficulty-based rewards with a shortcut-aware information-gain objective that directly promotes dependence on cross-passage evidence.

2 Related Work

2.1 Self-evolving task generation

Self-evolving methods reduce reliance on human-written training examples through iterative self-training or proposer-solver self-play (Zhang et al., 2025; Li et al., 2026a). Absolute Zero (Zhao et al., 2026) established zero-data self-play with an execution-based oracle, while R-Zero (Huang et al., 2025) decoupled the proposer and solver to support broader reasoning tasks. In the search setting, SSP (Lu et al., 2025a) generates questions from answers sampled from public QA datasets and filters them for evidence sufficiency. Dr. Zero (Yue et al., 2026) further removes curated QA pairs by generating questions from corpus passages, but continues to use solver pass rate as the proposer difficulty signal. Recent methods introduce stronger structural and verification signals. KG Paths (Wu et al., 2026) uses Wikidata subgraphs for question construction and intermediate solver supervision, CoEvoKG (Li et al., 2026c) maintains an evolving KG memory, and EVE-Agent (Arai & Ichikawa, 2026) augments difficulty with evidence utility. Related analysis also emphasizes the role of data gating in controlling which self-generated examples enter training (Pu et al., 2026). However, these proposer-based methods retain solver-relative difficulty in their objectives. Dr. Free instead constructs questions from KG chains and evaluates them using teacher-forced full-versus-shortcut likelihood contrasts. As shown in Figure 2, this removes difficulty-based proposer rewards and requires neither annotated QA pairs nor solver rollouts for proposer reward computation.

2.2 Structured multi-hop construction and evidence verification

Apparent compositionality does not guarantee genuine multi-hop dependence. Both benchmark and synthetic questions can often be answered from a single passage or through disconnected reasoning (Jiang & Bansal, 2019; Trivedi et al., 2020; Guo et al., 2023; Chen et al., 2025b). Prior work addresses this problem through connected question construction (Trivedi et al., 2022; Lu et al., 2025b; Zhang et al., 2026b), graph-derived intermediate supervision (Wu et al., 2026), or evidence-aware verification and rewards (Lu et al., 2025a; Arai & Ichikawa, 2026; Zhang et al., 2026a). These mechanisms establish evidence structure, sufficiency, or utility but do not explicitly compare target-answer support under the full context against specific shortcut contexts. Dr. Free makes this comparison explicit: it scores a fixed answer under the full chain and contrasts it with closed-book, source-only, one-shot, and individual-passage contexts. The same knowledge-graph chain therefore defines both the question structure and the contexts used for shortcut-aware evaluation.

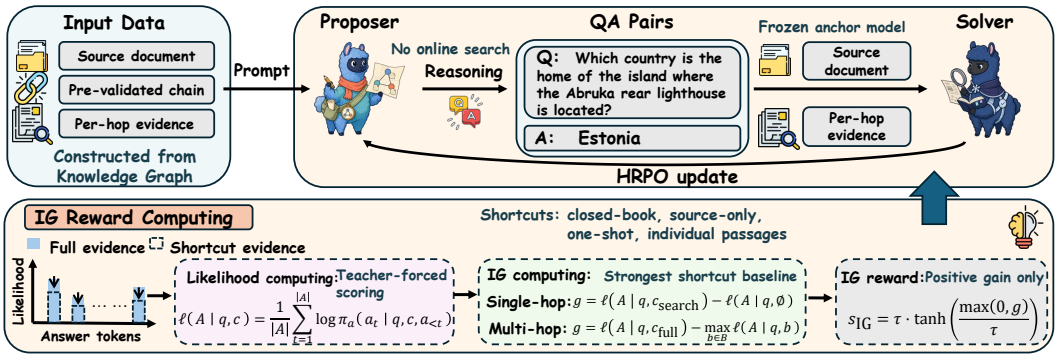


Figure 3: Overview of Dr. Free. We first sample pre-validated relational chains from a knowledge graph and align each hop with supporting evidence, providing an explicit multi-hop structure for question construction. Given the validated chain and its aligned evidences, the proposer generates a question without performing online multi-turn search. The shortcut-aware information-gain reward then evaluates the generated question by comparing the likelihood of the target answer under the full evidence context against that under the strongest evaluated shortcut context, directly rewarding questions that require cross-passage evidence.

3 METHOD

We adopt the proposer-solver paradigm for data-free self-evolution (Yue et al., 2026). A proposer π_θ synthesizes question-answer pairs (x, y), while a solver π_ϕ learns to answer the generated questions. Both models are initialized from the same base LLM and optimized through an alternating procedure without annotated data. The two models maximize their respective expected rewards:

$$\text{Proposer:} \quad \mathbb{E}_{(x,y)\sim\pi_\theta(\cdot|\mathcal{R})}\big[\, r(x,y)\,\big], \tag{1}$$

$$\text{Solver:} \quad \mathbb{E}_{(x,y)\sim\pi_\theta(\cdot|\mathcal{R}),\, \hat{y}\sim\pi_\phi(\cdot|x,\mathcal{R})}\big[\, \mathbb{I}(y=\hat{y})\,\big], \tag{2}$$

where r denotes the proposer reward, and $\mathbb{I}$ denotes the indicator function used for the solver reward. In the absence of annotated data, every question-answer pair available to the solver is generated by the proposer. The central challenge is therefore to synthesize pairs that provide effective learning signals. Accordingly, we adopt the solver as given and use the standard outcome-reward formulation (Jin et al., 2025; Yue et al., 2026). As shown in Figure 3, we focus on the two proposer components that determine data quality: the information on which the proposer conditions (Section 3.1) and the reward used to optimize the proposer (Section 3.2). Together, these two components address complementary aspects of proposer optimization. The knowledge-graph-based construction provides an explicit multi-hop evidence structure for question generation, while the information-gain reward evaluates whether the full evidence provides additional support beyond the evaluated shortcut contexts.

3.1 KNOWLEDGE-GRAPH CHAIN SAMPLER

In Dr. Zero, the proposer is expected to retrieve intermediate evidence starting from a seed passage and compose the retrieved information into a multi-hop question. In practice, however, this process often collapses into single-passage question generation. Across two backbone models, most question-answer pairs labeled as multi-hop are generated without search, and the target answer is already present in the seed passage (see Sec.4.4). As a result, the assigned hop count just reflect the intended generation format rather than actual cross-document reasoning. Moreover, because pass-rate-based difficulty measures whether the current solver can answer a question rather than whether the supporting passages form a necessary and coherent evidence chain, it cannot reliably distinguish genuine multi-hop questions from such shortcut cases.

Building on the use of knowledge-graph structures to guide question construction (Wu et al., 2026), Dr. Free makes the intended evidence structure an explicit input to the proposer rather than relying on proposer-side search to discover it implicitly. We represent a relational chain as $\mathcal{C} = (e_0, r_1, e_1, \ldots, r_h, e_h)$, where each relation r_i connects e_{i-1} to e_i, e_0 is the source entity, and

e_h is the target answer. Each entity e_i is further associated with an aligned Wikipedia passage. We sample such chains from Wikidata5M (Wang et al., 2021) through random walks over the subgraph induced by entities covered by Wiki-18.

To obtain chains with unambiguous targets and retrievable intermediate entities, we apply three filters. First, we remove cyclic paths so that each hop introduces a new entity. Second, we retain relations for which a source entity maps to a unique or nearly unique target and discard entities with ambiguous surface forms, thereby reducing the likelihood of multiple valid answers. Third, we retain a chain only when every entity e_i is aligned with at least one Wiki-18 passage and at least one passage aligned with the source entity e_0 mentions the first bridge entity e_1. These filters jointly reduce entity ambiguity and enforce basic retrieval constraints, yielding structured inputs for question generation. Given a retained chain and its aligned passages, the proposer generates a question whose target answer is e_h, without performing multi-turn search during question construction. The information-gain reward is then applied after generation to determine whether the full evidence context supports the target answer more strongly than the evaluated shortcut contexts. Additional details of the knowledge-graph construction are provided in Appendix A.5.

3.2 Shortcut-Aware Information-Gain Reward

Given a structured evidence chain, the proposer should favor question-answer pairs whose intended evidence contributes information that cannot be obtained from shortcut contexts alone. We operationalize this principle with a rollout-free, shortcut-aware information-gain reward. For a fixed question and target answer, a frozen anchor model evaluates the likelihood of the target answer under the full intended evidence and under each evaluated shortcut context. The reward is defined as the margin between the target-answer likelihood under the full evidence and the maximum likelihood across all shortcut contexts. A positive margin therefore indicates that the full evidence provides additional support for the target answer beyond any evaluated shortcut. We describe the reward formulation in detail below.

Answer likelihood. Let q denote the question generated by the proposer, and let $A = (a_1, \ldots, a_{|A|})$ denote the gold answer to q, represented as a sequence of $|A|$ tokens. Let c denote a context containing zero or more evidence passages. We define a frozen anchor model π_{a}, initialized as the base LLM in the first iteration and updated to the solver from the immediately preceding iteration thereafter. Given q and c, the anchor model computes the length-normalized log-likelihood of the target answer:

$$\ell(A \mid q, c) \;=\; \frac{1}{|A|} \sum_{t=1}^{|A|} \log \pi_{\text{a}}\big(a_t \mid q,\, c,\, a_{<t}\big), \tag{3}$$

where a_t is the t-th answer token and $a_{<t}$ denotes all preceding answer tokens. Normalization by the answer length $|A|$ makes the scores comparable across answers of different lengths. Computing $\ell(A \mid q, c)$ requires only a single teacher-forced forward pass over the fixed answer A. Unlike the pass rate, this computation does not require sampling an answer from π_{a}, thereby avoiding the repeated solver rollouts that make the baseline reward expensive to compute. Across all evidence comparisons below, the question q and answer A are held fixed; only the evidence context c is varied.

Single-hop questions. The simplest case is a single-hop chain, in which the question should be answerable from the retrieved evidence but not from parametric knowledge alone. We retrieve using the question itself, $c_{\text{search}} = \text{retrieve}(q)$, and measure how much the retrieved evidence raises the likelihood of the answer conditioned on the same question relative to the closed-book context,

$$g \;=\; \ell(A \mid q, c_{\text{search}}) - \ell(A \mid q, \varnothing). \tag{4}$$

A large gain indicates that the retrieved passage substantially increases the likelihood assigned by the anchor model to the target answer relative to the closed-book context. In contrast, a small gain indicates either that the answer already has a high closed-book likelihood or that the retrieved passage provides little additional support.

Multi-hop questions. Multi-hop questions must additionally resist predefined shortcuts that allow the answer to be recovered without cross-passage composition. Here the full-evidence context is the concatenation of all passages along the chain, denoted c_{full}, and we compare it against a set B of shortcut contexts, each corresponding to one way in which the chain could be bypassed. For a chain with h search hops, the set of shortcut contexts is

$$B = \{\varnothing,\, c_{\text{src}},\, c_{\text{1-shot}},\, c_1, \ldots, c_h\}\,, \tag{5}$$

where:

- $\ell(A \mid q, \varnothing)$ is the closed-book likelihood, testing whether the question is answerable from parametric knowledge alone;
- $\ell(A \mid q, c_{\text{src}})$ conditions on the seed passage alone, testing whether the question can be answered without searching beyond the source;
- $\ell(A \mid q, c_{\text{1-shot}})$ conditions on the result of a single retrieval using q, testing whether the question is answerable within one retrieval hop;
- $\ell(A \mid q, c_k)$ conditions on the evidence passage associated with hop k in isolation, testing whether any single passage suffices without cross-passage composition.

The gain is defined as the likelihood margin of the full evidence chain over the strongest shortcut, with the generated question held fixed:

$$g \;=\; \ell(A \mid q, c_{\text{full}}) - \max_{b \in B} \ell(A \mid q, b). \tag{6}$$

A question therefore receives credit only when the full evidence context assigns a higher likelihood to the target answer than every evaluated shortcut context. Finally, in both the single-hop and multi-hop cases, the gain is converted into a reward through a saturating transformation that retains only positive gains,

$$s_{\text{IG}} \;=\; \tau \cdot \tanh\!\left(\frac{\max(0,\, g)}{\tau}\right), \tag{7}$$

so that a question receives credit only when the intended evidence strictly increases the likelihood of the answer relative to the strongest applicable comparison context. The $\tanh$ function bounds the reward by $\tau = 3$ and stabilizes the reward scale across chains with different numbers of hops.

3.3 Reward Composition and Optimization

Reward. We combine a graded format score with a gated information-gain reward. The format score s_{fmt} assigns partial credit to a parseable question-answer pair that does not reveal the answer, includes the required reasoning block, and provides an answer that matches the target entity of the chain. A separate grounding check verifies that the target answer occurs in an admissible supporting passage. For a multi-hop chain, the supporting evidence must include at least one non-seed passage from the chain rather than relying exclusively on the seed passage. The total proposer reward is

$$r = \lambda s_{\text{fmt}} + \mathbb{I}[s_{\text{fmt}} = 1 \wedge \text{grounded}]\, s_{\text{IG}}, \tag{8}$$

where λ is the format weight, set to 0.2. Thus, the information-gain reward contributes only when a pair receives the full format score and passes the grounding check; all other pairs receive only the graded format reward. Appendix A.6 provides the detailed parsing, scoring, and grounding criteria.

Optimization. We train the proposer with hop-grouped relative policy optimization (Yue et al., 2026) (HRPO), which standardizes the advantage within each hop group so that questions compete only against those of the same hop count,

$$A_{i,h} \;=\; \frac{r_i - \text{mean}_{j \in \mathcal{I}_h}(r_j)}{\text{std}_{j \in \mathcal{I}_h}(r_j) + \delta}, \tag{9}$$

where $\mathcal{I}_h = \{j : \text{hop}(j) = h\}$ is the index set of samples with hop count h, and δ is a small constant that ensures numerical stability.

Solver. We train the solver as in prior work using GRPO (Shao et al., 2024) with an outcome reward based on answer correctness. The proposer and solver are updated alternately. At each iteration, the proposer is updated and generates a batch of question–answer pairs, after which the solver is trained on the generated batch. The resulting solver then serves as the anchor model for the next iteration.

Table 1: Performance on seven open-domain QA benchmarks measured by Exact Match (EM). For each model size, methods are grouped as few-shot/supervised or data-free. Baseline results are taken from the original papers under their reported setups.

Methods	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-3B-Instruct								
Few-shot / Supervised								
IRCoT (Trivedi et al., 2023)	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1 (Li et al., 2025)	0.238	0.472	0.262	0.221	0.218	0.054	0.320	0.255
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-Instruct (Guo et al., 2025)	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
Search-R1-Instruct (Jin et al., 2025)	0.341	0.545	0.378	0.324	0.319	0.103	0.264	0.325
Data-Free								
SQLM (Chen et al., 2025a)	0.264	0.432	0.258	0.226	0.238	0.060	0.158	0.233
R-Zero (Huang et al., 2025)	0.389	0.513	0.370	0.243	0.128	0.052	0.096	0.256
Dr. Zero (Yue et al., 2026)	**0.397**	**0.572**	**0.431**	0.298	0.291	0.091	0.200	0.326
Dr. Free (Ours)	0.380	**0.572**	0.416	**0.353**	**0.390**	**0.172**	**0.344**	**0.375**
Qwen2.5-7B-Instruct								
Few-shot / Supervised								
IRCoT (Trivedi et al., 2023)	0.224	0.478	0.301	0.133	0.149	0.072	0.224	0.239
Search-o1 (Li et al., 2025)	0.151	0.443	0.131	0.187	0.176	0.058	0.296	0.206
SFT	0.318	0.354	0.121	0.217	0.259	0.066	0.112	0.207
R1-Instruct (Guo et al., 2025)	0.270	0.537	0.199	0.237	0.292	0.072	0.293	0.271
Search-R1-Instruct (Jin et al., 2025)	0.393	**0.610**	0.397	0.370	**0.414**	0.146	0.368	0.385
Data-Free								
SSP (Lu et al., 2025a)	0.354	0.576	0.428	0.322	0.304	0.154	0.344	0.355
KG Paths (Wu et al., 2026)	0.352	0.568	0.408	0.330	0.322	0.128	0.400	0.358
Dr. Zero (Yue et al., 2026)	**0.406**	0.608	0.416	0.362	0.347	0.104	0.360	0.372
Dr. Free (Ours)	0.397	**0.610**	**0.438**	**0.387**	0.389	**0.164**	**0.424**	**0.401**

4 Experiments

4.1 Experimental Setup

Training. Training proceeds for three self-evolution iterations, each comprising 50 proposer updates, data generation, and 50 solver updates. In each iteration, we uniformly sample one-, two-, and three-hop chains from a pool of approximately 100,000 verified chains, using only the number required by the training budget rather than the entire pool. The IG anchor is the frozen base model in the first iteration and the solver from the preceding iteration thereafter. Complete training and sampling configurations are provided in Appendix A.1.

Datasets and metrics. We evaluate on three single-hop benchmarks—Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)—and four multi-hop benchmarks—HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). We use the Search-R1 test splits (Jin et al., 2025), totaling 51,713 questions, and we report exact match (EM) in Table 1, more details are provided in Appendix A.3.

4.2 Main Results

Table 1 reports exact-match performance on seven benchmarks. Dr. Free achieves the highest average performance at both model scales, outperforming Dr. Zero and the matched Search-R1-Instruct baseline. The improvements are most pronounced on multi-hop QA, where Dr. Free leads on nearly all benchmarks across both scales. This pattern aligns with the shortcut analysis in Table 4 and suggests that our explicit evidence chains and the IG reward are particularly beneficial for multi-hop reasoning. More results of Dr. Free on different model scales are shown in Table 8 in the Appendix.

4.3 Ablation Study

Module Ablation. Table 2 evaluates the contribution of the KG chain sampler and the proposer reward. Removing proposer optimization causes the largest performance drop, reducing the average score from 0.375 to 0.306, with most of the degradation occurring on multi-hop QA benchmarks.

Table 2: Ablation of Dr. Free with Qwen2.5-3B-Instruct. *Difficulty Only* replaces the IG reward with the pass-rate difficulty reward; *IG + Difficulty* combines both rewards; *w/o KG* restores Dr. Zero-style proposer search; and *w/o Proposer Training* uses the base proposer to generate questions from KG chains without reward-based optimization. Best results are shown in bold.

	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Dr. Free	**0.380**	**0.572**	**0.416**	**0.353**	**0.390**	**0.172**	**0.344**	**0.375**
Difficulty Only	0.362	0.568	0.405	0.324	0.357	0.121	0.264	0.343
IG + Difficulty	0.358	0.563	0.389	0.325	0.357	0.137	0.248	0.340
w/o KG	0.366	0.536	0.365	0.319	0.356	0.129	0.272	0.335
w/o Proposer Training	0.377	0.570	0.393	0.289	0.281	0.063	0.168	0.306

Table 3: Leave-one-shortcut-type-out ablation of the shortcut set B. Each row removes one type of shortcut context from the full set. All four shortcut types contribute positively on average, with the largest degradation occurring when the source-only hop passage contexts are removed.

	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Dr. Free (full B)	**0.380**	**0.572**	**0.416**	**0.353**	**0.390**	**0.172**	0.344	**0.375**
w/o one-shot	0.366	0.564	0.390	0.324	0.373	0.142	**0.360**	0.360
w/o closed-book	0.341	0.556	0.370	0.315	0.326	0.117	0.288	0.330
w/o per-hop	0.321	0.535	0.355	0.290	0.311	0.114	0.320	0.321
w/o source-only	0.326	0.527	0.356	0.295	0.303	0.104	0.304	0.316

This result indicates that structured evidence chains alone are insufficient to produce effective training questions without reward-guided proposer optimization. Replacing the information-gain (IG) reward with the difficulty reward reduces the average score to 0.343, while combining the two yields a similar score of 0.340. These results suggest that, under our setting, the evaluated difficulty signal neither provides an effective substitute for the IG reward nor offers additional benefit when combined with it. Removing the KG scaffold also reduces the average score to 0.335, demonstrating the importance of explicit relational structure for question construction. Overall, both components contribute to the performance of Dr. Free, with IG-based proposer optimization accounting for the largest improvement.

Shortcut Ablation. Table 3 removes each shortcut context from B individually. Every removal reduces average downstream performance, indicating that the shortcut contexts provide complementary training signals. The source-only shortcut has the largest effect, reducing the average from 0.375 to 0.316 (-0.059). Without this comparison, the reward no longer directly penalizes cases in which the seed passage alone assigns high likelihood to the target answer. Removing the per-hop shortcut yields a similarly large drop to 0.321 (-0.054), because the reward no longer contrasts the full-context likelihood against each individual evidence passage. Removing the closed-book and one-shot shortcuts lowers the average to 0.330 (-0.045) and 0.360 (-0.015), respectively. The reductions are generally larger on the multi-hop benchmarks, suggesting that the complete shortcut set helps construct training questions that improve downstream multi-hop QA performance. Concrete reward traces illustrating how the source-only and per-hop contexts prevent false-positive rewards are provided in Appendix A.7.

4.4 Analysis of Proposer Search and Shortcut Behavior

Prior data-free proposers may generate questions with pseudo-multi-hop labels without performing genuine multi-document search. To determine whether this behavior persists across backbones, we reproduce Dr. Zero (Yue et al., 2026) with Qwen2.5-3B-Instruct and Qwen3-4B under the same retrieval configuration using E5-base-v2 and Wiki-18. Each replication generates 33,923 question–answer pairs for solver training.

Table 4 shows that the shortcut occurs with both backbones. Among questions labeled as multi-hop, the zero-search rates are 99.995% for Qwen2.5-3B-Instruct and 84.73% for Qwen3-4B, while 96.4% and 83.9% of the corresponding answers occur in the seed passages. The stronger backbone reduces but does not eliminate the shortcut. Representative cases in Appendix A.8 further illustrate that the

Table 4: Behavior of the Dr. Zero proposer under two backbone replications. Across both models, most questions labeled as multi-hop are generated without any search, and their answers are frequently recoverable from the seed passage.

(a) Qwen2.5-3B-Instruct

Hop label	# Pairs	Zero-search	Answer in seed
1	13,722	—	96.9%
2	10,131	100.0%	96.4%
3	6,630	99.98%	96.3%
4	3,440	100.0%	96.0%
≥ 2	**20,201**	**99.995%**	**96.4%**

(b) Qwen3-4B

Hop label	# Pairs	Zero-search	Answer in seed
1	13,602	—	93.2%
2	10,192	79.34%	83.1%
3	6,791	90.21%	85.4%
4	3,338	90.05%	83.1%
≥ 2	**20,321**	**84.73%**	**83.9%**

Table 5: Gold-blind counterfactual audit of four observable reasoning shortcuts. Each method contributes 100 questions with a matched 51/49 two-/three-hop distribution. Brackets report Wilson 95% confidence intervals. The final two rows report one-sided Fisher exact-test p-values for Dr. Free.

Method	**Closed-book shortcut ↓**	**Source-only shortcut ↓**	**One-shot shortcut ↓**	**Single-component shortcut ↓**
Dr. Zero	17% [10.9, 25.5]	81% [72.2, 87.5]	65% [55.3, 73.6]	88% [80.2, 93.0]
KG + Difficulty	48% [38.5, 57.7]	20% [13.3, 28.9]	45% [36.0, 55.2]	51% [41.3, 60.6]
Dr. Free	**5%** [2.2, 11.2]	**3%** [1.0, 8.5]	**2%** [0.6, 7.0]	**5%** [2.2, 11.2]
Dr. Free vs. baseline (p-value)				
vs. Dr. Zero	5.7×10^{-3}	3.1×10^{-33}	3.8×10^{-24}	1.4×10^{-36}
vs. KG + Difficulty	7.0×10^{-13}	1.1×10^{-4}	1.5×10^{-14}	3.9×10^{-14}

hop label often reflects the requested output format rather than actual cross-document composition. These findings provide a plausible explanation for the limited multi-hop gains of Dr. Zero.

We further evaluate whether Dr. Free suppresses shortcut behavior through a controlled, gold-blind counterfactual audit over the four contexts in $\mathcal{B}$: closed-book, source-only, one-shot retrieval, and individual chain components. Each method contributes 100 questions with the same two-/three-hop distribution, and each counterfactual context is evaluated through an independent judge call.

Table 5 reveals distinct shortcut profiles across methods. For Dr. Zero, the source-only and single-component shortcut rates reach 81% and 88%, respectively, whereas the closed-book shortcut rate is only 17%. This pattern suggests that the dominant shortcuts arise from local evidence contained in the provided passages rather than from parametric knowledge alone. In contrast, KG + Difficulty, corresponding to the *Difficulty Only* ablation in Table 2, exhibits a substantially higher closed-book shortcut rate of 48%, indicating that the difficulty reward does not explicitly penalize questions that can be answered from parametric knowledge. Dr. Free reduces all four shortcut rates to between 2% and 5%, with statistically significant improvements over both baselines in every comparison. These results align with the IG objective, which explicitly rewards a likelihood advantage of the full evidence over the strongest evaluated shortcut context.

5 Conclusions

We introduced Dr. Free, a self-evolving search framework that replaces solver-relative difficulty with a shortcut-aware objective for measuring evidence dependence. Across five backbone models and seven open-domain QA benchmarks, Dr. Free consistently improves overall performance, with particularly strong gains on multi-hop tasks, while reducing proposer training time by more than $7\times$. Ablation studies verify the contributions of both structured chain construction and the shortcut-aware information-gain reward. Overall, Dr. Free provides a principled and practical approach for generating higher-quality training questions while improving both the effectiveness and efficiency of data-free search-agent self-evolution.

AI USE STATEMENT

We used AI tools to assist with language polishing, including correcting grammar, improving clarity, and refining phrasing. All AI-assisted edits were reviewed and revised by the authors to ensure that they accurately reflected the intended scientific meaning. In addition, we used an LLM as an evaluator in the counterfactual shortcut audit described in Appendix A.12. Specifically, GPT-4o was used as a gold-blind judge to assess whether a question was answerable under different evidence conditions. The judge was used only for evaluation and analysis, following a fixed prompting and decision protocol, and did not have access to the gold answer during generation.

REFERENCES

Yamato Arai and Yuma Ichikawa. Eve-agent: Evidence-verifiable self-evolving agents. *arXiv preprint arXiv:2605.22905*, 2026. 1, 3

Lili Chen, Mihir Prabhudesai, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Self-questioning language models. *arXiv preprint arXiv:2508.03682*, 2025a. 7

Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Hang Yan, Kai Chen, and Dahua Lin. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27129–27151, 2025b. 3

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026. 1

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 7

Wangzhen Guo, Qinkang Gong, Yanghui Rao, and Hanjiang Lai. Counterfactual multihop qa: A cause-effect approach for reducing disconnected reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4214–4226, 2023. 1, 3

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020. 1, 7

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025. 1, 3, 7

Yichen Jiang and Mohit Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2726–2736, 2019. 1, 3

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516. 1, 4, 7

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. 7

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020. 16

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. 1, 7

Jian Li, Yizhang Jin, Dongqi Liu, Hang Ding, Jiafu Wu, Dongsheng Chen, Yunhang Shen, Yulei Qin, Ying Tai, Chengjie Wang, et al. Se-search: Self-evolving search agent via memory and dense reward. *arXiv preprint arXiv:2603.03293*, 2026a. 3

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025. 1, 7

Yunheng Li, Guohong Mu, Hao Li, Shengsheng Qian, Dingwen Zhang, Qibin Hou, and Ming-Ming Cheng. Annotations as rollouts: Efficient and scalable reinforcement learning for video mllms. *arXiv preprint arXiv:2608.20492*, 2026b. 1

Zhaoyang Li, Zenghuang Fu, Qiuyuan Ai, Ping Jiang, Haoyu Wu, Minghui Wu, Chenxu Zhao, Jie Song, and Guannan He. Coevokg: Co-evolving knowledge graphs with self-evolving search agents. *arXiv preprint arXiv:2608.01904*, 2026c. 3

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 13

Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Jiaqi Guo, Haotian Xu, Chutian Wang, Haonan Chen, Xiaoxi Jiang, and Guanjun Jiang. Search self-play: Pushing the frontier of agent capability without supervision. *arXiv preprint arXiv:2510.18821*, 2025a. 1, 3, 7

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*, 2025b. 3

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023. 7

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. 1, 7

Sophia Xiao Pu, Zhaotian Weng, Chengzhi Liu, Jayanth Srinivasa, Gaowen Liu, William Yang Wang, and Xin Eric Wang. Survive or collapse: The asymmetric roles of data gating and reward grounding in self-play rl. *arXiv preprint arXiv:2605.22217*, 2026. 3

Zhipeng Qian, Zihan Liang, Yufei Ma, Ben Chen, Huangyu Dai, Jiayi Ji, Chenyi Lei, Wenwu Ou, Xiaoshuai Sun, and Qibin Hou. Plan before search: Search agents need plan. *arXiv preprint arXiv:2605.28354*, 2026. 1

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. 6, 13

Wentao Tan, Qiong Cao, Jiaqi Wang, and Nan Duan. Searchmaster: Grounded and regulated self-play for search agents. *arXiv preprint arXiv:2608.01822*, 2026. 1

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Is multihop qa in dire condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8846–8863, 2020. 1, 3

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. 1, 3, 7

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023. 1, 7

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021. 5, 16

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. 1, 3, 4, 7

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018. 1, 7

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. *arXiv preprint arXiv:2601.07055*, 2026. 1, 3, 4, 6, 7, 8, 13, 21

Ding-Chu Zhang, Yida Zhao, Jialong Wu, Liwen Zhang, Baixuan Li, Wenbiao Yin, Yong Jiang, Yu-Feng Li, Kewei Tu, Pengjun Xie, et al. Evolvesearch: An iterative self-evolving search agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13134–13147, 2025. 3

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20744–20765, 2026a. 3

Xichen Zhang, Ziyi He, Yinghao Zhu, Sitong Wu, Shaozuo Yu, Meng Chu, Wenhu Zhang, Haoru Tan, and Jiaya Jia. Searchgym: Bootstrapping real-world search agents via cost-effective and high-fidelity environment simulation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18633–18665, 2026b. 3

Andrew Zhao, Yiran Wu, 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. *Advances in Neural Information Processing Systems*, 38:105816–105879, 2026. 1, 3

Table 6: Representative proposer and solver configurations for Dr. Free.

(a) Proposer (challenger)

Hyperparameter	Setting
Algorithm	HRPO
Maximum training steps	50
Optimizer	AdamW
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
Learning rate	1×10^{-6}
Warmup ratio	0.03
LR scheduler	Constant with warmup
Weight decay	0.01
Maximum gradient norm	0.1
Group size in HRPO	1
Total train batch size	256
PPO mini-batch size	128
Precision	BF16-mixed
Maximum prompt length	2560
Multi-turn tool use	Disabled

(b) Solver

Hyperparameter	Setting
Algorithm	GRPO
Maximum training steps	50
Optimizer	AdamW
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
Learning rate	1×10^{-6}
Warmup ratio	0.03
LR scheduler	Constant with warmup
Weight decay	0.01
Maximum gradient norm	0.1
Group size in GRPO	5
Total train batch size	256
PPO mini-batch size	128
Precision	BF16-mixed
Maximum prompt length	512
Multi-turn tool use	Enabled (search tool)

A Appendix

A.1 Implementations

We evaluate Dr. Free across multiple backbone sizes and initialize both the proposer and solver from the same backbone. The construction pipeline produces approximately 100,000 verified chains, which form a candidate pool rather than a training set used exhaustively. In each self-evolution iteration, we uniformly sample one-, two-, and three-hop chains according to the training budget. With a total batch size of 256, the 50 proposer updates require 12,800 chain instances. Subsequent data generation is restricted to the question–answer pairs required for 50 solver updates.

Dr. Free inherits Dr. Zero's Wiki-18 index, E5-base-v2 retriever, search-turn limit and retrieval top-k. Following Dr. Zero (Yue et al., 2026), we also optimize the proposer using HRPO with one generation per prompt and the information-gain reward. Proposer optimization runs for at most 50 steps with a maximum prompt length of 2,560 tokens and no tool interaction. We then train the solver on the generated question–answer pairs using GRPO (Shao et al., 2024). The solver samples five trajectories per question, interacts with the search tool over multiple turns, and is optimized for at most 50 steps with a maximum prompt length of 512 tokens.

Both stages use AdamW (Loshchilov & Hutter, 2017) with a learning rate of 1×10^{-6}, a 3% warmup ratio, a weight decay of 0.01, and a maximum gradient norm of 0.1. We use a total training batch size of 256, a PPO mini-batch size of 128, a constant learning-rate schedule with warmup, and BF16 mixed precision. Table 6 summarizes the complete configurations.

A.2 Proposer efficiency comparison

Qwen2.5-3B-Instruct. Under the same 8×H800 setup, Dr. Free reduces the runtime of 50 proposer updates from 7.88 to 0.82 hours (9.6×). Generating the complete solver-training pool decreases from 6.30 to 1.04 hours (6.1×), or from 0.669 to 0.094 seconds per question (7.1×). This reduction comes from replacing Dr. Zero's live retrieval with five samples per prompt by direct question generation from a grounded KG path. Overall, proposer training and data generation decrease from 14.18 to 1.86 hours (7.6×). Including 50 solver updates, data conversion, and evaluation, the estimated end-to-end runtime decreases from 22.02 to 8.61 hours (2.56×), reducing GPU-hours by 60.9%. The KG-chain pool is constructed and validated only once, taking less than one hour, and is reused in all subsequent iterations. We therefore exclude this one-time preprocessing cost from the per-iteration comparison.

Standardized cross-backbone comparison. We additionally standardize data generation to the 12.8k questions required by 50 solver updates. For Qwen2.5-3B-Instruct, generation time decreases from 4.75 to 0.67 hours ($7.14\times$), while the combined proposer-side runtime decreases from 12.63 to 1.49 hours ($8.48\times$), reducing GPU-hours by 88.2%. For Qwen2.5-7B-Instruct, proposer training decreases from 8.42 to 0.94 hours ($8.96\times$), generation from 3.93 to 0.89 hours ($4.42\times$), and the combined runtime from 12.35 to 1.83 hours ($6.75\times$), reducing GPU-hours by 85.2%.

Overall trend. Qwen3-4B/8B and Llama-3.1-8B-Instruct show the same pattern: proposer training is accelerated by $7.37\times$–$8.04\times$, data generation by $3.42\times$–$5.30\times$, and the combined proposer-side pipeline by $5.61\times$–$6.92\times$. Their per-iteration GPU-hour reductions range from 82.2% to 85.6%. Across all five backbones, the combined proposer-side speedup averages $6.8\times$, with no systematic dependence on model scale.

A.3 More Ablations

Table 7: Effect of the hop-sampling ratio used to construct proposer training data. All columns report EM under the same training and evaluation protocol. Values are rounded to three decimal places, and bolding is determined using the unrounded values.

Setting	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Ratio 1:1:1:1	0.377	0.570	0.403	0.338	0.352	0.141	0.264	0.349
Ratio 4:3:2	0.375	0.572	0.408	0.324	0.354	0.129	0.296	0.351
Ratio 1:1:1	**0.380**	**0.572**	**0.416**	**0.353**	**0.390**	**0.172**	**0.344**	**0.375**

A.3.1 Ablations on hop ratio

Table 7 examines how the hop composition of the proposer training data affects downstream performance. Ratio 1:1:1, which assigns equal proportions to one-, two-, and three-hop examples, achieves the highest EM on all seven benchmarks. Compared with Ratio 4:3:2, increasing the relative proportion of multi-hop examples produces particularly clear gains on HotpotQA, 2Wiki, MuSiQue, and Bamboogle, demonstrating the effectiveness of Dr. Free in constructing useful multi-hop supervision. However, extending the mixture to four-hop examples with Ratio 1:1:1:1 reduces the average EM from 0.375 to 0.349. Constructing well-posed questions with unambiguous intermediate references and complete evidence dependencies becomes increasingly difficult as chain length grows. The performance degradation may therefore reflect the lower reliability of the four-hop question–answer pairs. We accordingly adopt Ratio 1:1:1 as the default configuration.

A.3.2 Performance across model scales

Table 8 compares Dr. Free with Dr. Zero across five model backbones and seven QA benchmarks. Dr. Free consistently achieves higher average EM across all backbones and improves average F1 whenever comparable results are available. The gains are particularly consistent on multi-hop benchmarks such as HotpotQA, MuSiQue, and Bamboogle, while Dr. Zero remains competitive on several single-hop datasets. Overall, these results demonstrate that the effectiveness of Dr. Free generalizes across model families and scales, with its main advantage arising from improved performance on multi-hop QA benchmarks.

A.3.3 Effect of the Anchor-Model Update Strategy

We ablate the default anchor update by retaining the base-model anchor in Iteration 2 while keeping all other settings fixed. Figure 4 plots Iteration 1 once over steps 1–50 and places Iteration 2 step k at cumulative step $50+k$. The frozen-anchor solver achieves a higher training reward, but its validation reward drops sharply from 0.3626 to 0.2434. In contrast, the updated-anchor solver attains a lower training reward while largely preserving validation performance, from 0.3780 to 0.3734, and reaches a peak of 0.3915. Because Panel (a) reports the solver reward rather than the proposer IG reward, this discrepancy cannot be attributed to anchor-dependent reward calibration. Instead, the train–validation divergence is consistent with a distributional effect induced by proposer optimization: the proposer has already been trained to favor questions that receive high reward under the fixed anchor,

Table 8: Evaluation with Exact Match (EM) and token-level F1. Each cell reports EM / F1 in percent. Boldface marks the best comparable value within each backbone group. F1 is compared only when both methods report it; blank entries denote results that have not yet been provided.

	GeneralQA			Multi-HopQA				
Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-3B-Instruct								
Dr. Zero	**39.7** /	**57.2** /	**43.1** /	29.8 /	29.1 /	9.1 /	20.0 /	32.6 /
Dr. Free(Ours)	38.0 / 47.5	**57.2** / 65.0	41.6 / 45.8	**35.3** / 46.1	**39.0** / 45.2	**17.2** / 25.0	**34.4** / 47.9	**37.5** / 46.1
Qwen2.5-7B-Instruct								
Dr. Zero	**40.6** /	60.8 /	41.6 /	36.2 /	34.7 /	10.4 /	36.0 /	37.2 /
Dr. Free(Ours)	39.7 / 50.1	**61.0** / 69.0	**43.8** / 48.5	**38.7** / 49.8	**38.9** / 46.2	**16.4** / 23.5	**42.4** / 53.0	**40.1** / 48.6
LLaMA-3.1-8B-Instruct								
Dr. Zero	**42.9 / 52.3**	**64.3 / 72.3**	45.2 / **51.7**	31.7 / 42.0	**31.4 / 37.4**	7.5 / 14.7	16.8 / 27.8	34.2 / 42.6
Dr. Free(Ours)	41.5 / 50.8	62.5 / 70.8	**46.2** / 49.3	**32.6 / 44.2**	31.0 / 37.1	**12.6 / 20.7**	**35.2 / 44.8**	**37.4 / 45.4**
Qwen3-4B								
Dr. Zero	38.7 / 49.8	61.6 / 69.8	40.3 / 47.5	36.2 / 47.2	38.8 / 45.5	14.0 / 22.0	40.8 / 52.4	38.6 / 47.7
Dr. Free(Ours)	**42.0 / 51.3**	**64.5 / 71.4**	**44.9 / 49.8**	**38.4 / 49.0**	**41.1 / 46.5**	**15.5 / 23.4**	**41.6 / 53.0**	**41.1 / 49.2**
Qwen3-8B								
Dr. Zero	39.5 / 51.3	65.2 / 73.1	42.0 / 48.7	38.5 / 49.9	42.8 / 49.1	13.5 / 21.9	41.6 / 53.4	40.4 / 49.6
Dr. Free(Ours)	**39.9 / 51.4**	**66.7 / 74.1**	**43.8 / 49.7**	**40.3 / 51.8**	**46.5 / 52.1**	**16.6 / 24.8**	**45.6 / 60.5**	**42.8 / 52.0**

making the resulting training questions easier for that same anchor and therefore yielding inflated training rewards without comparable gains in held-out performance. Updating the anchor changes this reference model and consequently shifts proposer optimization toward questions that remain informative for a stronger solver. In Iteration 3, the anchor is further updated to the Iteration 2 solver. The continued improvement in held-out performance across all three iterations (Table 9) indicates that, at this training scale, progressively updating the anchor does not cause IG signal compression to accumulate into reward collapse.

A.4 Effect of Self-Evolution Iterations

Table 9: Performance of Dr. Free across self-evolution iterations. The best result in each column within each backbone is shown in bold. Both backbones improve across iterations, with the largest gains occurring on multi-hop benchmarks.

	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-3B-Instruct								
Dr. Free Iter 1	0.358	0.563	0.399	0.336	0.364	0.138	0.320	0.354
Dr. Free Iter 2	0.365	0.565	0.413	0.344	0.379	0.141	0.320	0.362
Dr. Free Iter 3	**0.380**	**0.572**	**0.416**	**0.353**	**0.390**	**0.172**	**0.344**	**0.375**
Qwen2.5-7B-Instruct								
Dr. Free Iter 1	0.382	**0.616**	0.404	0.376	0.345	0.165	**0.432**	0.389
Dr. Free Iter 2	**0.410**	0.608	0.429	**0.392**	0.376	**0.169**	0.392	0.397
Dr. Free Iter 3	0.397	0.610	**0.438**	0.387	**0.389**	0.164	0.424	**0.401**

Table 9 traces Dr. Free across three self-evolution iterations. Average EM increases monotonically from 0.354 to 0.362 and 0.375 for the 3B backbone, and from 0.389 to 0.397 and 0.401 for the 7B backbone. From the first to the third iteration, the largest 3B gains occur on MuSiQue ($+0.034$), 2Wiki ($+0.026$), and Bamboogle ($+0.024$), while HotpotQA improves by 0.017. For the 7B model, 2Wiki shows the largest gain ($+0.044$), followed by PopQA ($+0.034$) and HotpotQA ($+0.011$). Individual benchmarks do not necessarily improve monotonically: for example, the 7B MuSiQue and Bamboogle scores change by -0.001 and -0.008 from the first to the third iteration, and several 7B benchmarks attain their best performance before the final iteration. Overall, repeated self-evolution consistently improves aggregate performance, although its effects vary across individual benchmarks and model scales.

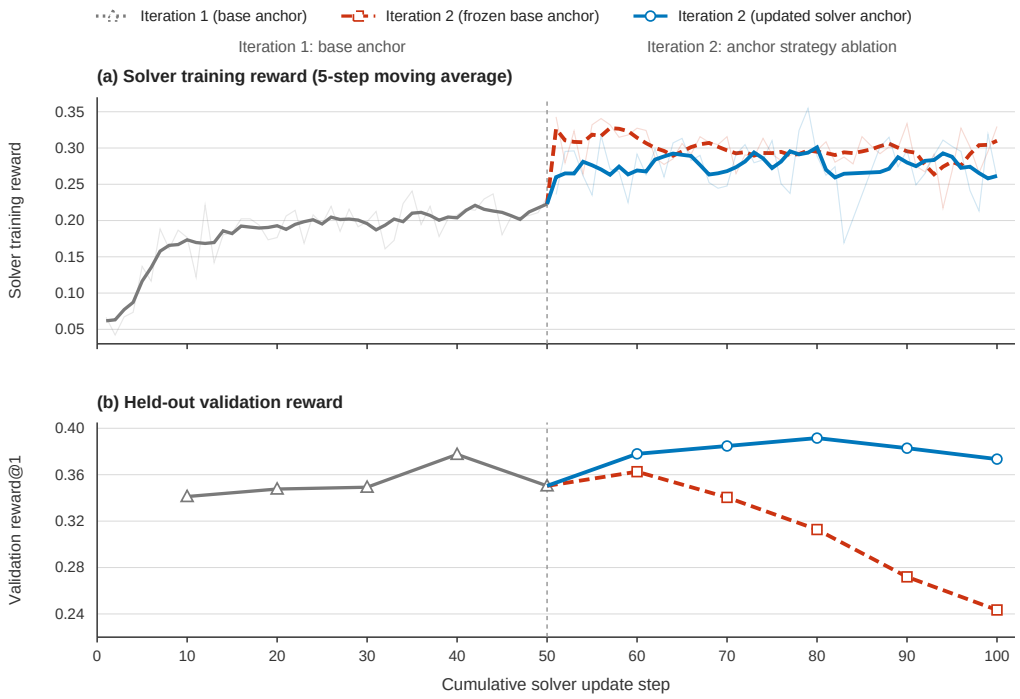


Figure 4: Anchor-update ablation across two iterations. Panel (a) shows the solver's raw training rewards (faint) and five-step moving averages; Panel (b) shows held-out validation reward. Iteration 1 is the shared prefix, and the two Iteration-2 strategies branch at cumulative step 50. The isolated excursion at updated-anchor steps 34–36 is omitted from the plot but retained in the accompanying data.

A.5 Knowledge-Graph Chain Sampler Details

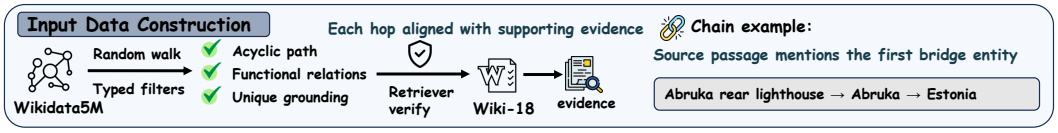


Figure 5: Knowledge-graph chain construction pipeline. Random walks on Wikidata5M generate candidate chains, which are filtered for acyclicity, functional or near-functional relations, and unique entity grounding. Retrieval verification in Wiki-18 checks that the source passage mentions the first bridge entity and that subsequent entities are reachable through per-hop retrieval.

This appendix provides the full technical specification of the relational chain construction pipeline. The pipeline samples typed relational paths from Wikidata5M and grounds every entity in the Wiki-18 corpus, ensuring that each chain is both semantically well-typed and executable by the retriever that the solver uses.

Data sources. We use two resources. Wikidata5M (Wang et al., 2021) provides the relational structure: approximately 20.6M triples over 4.8M entities and roughly 800 relation types, with each entity aligned to a Wikipedia article and annotated with a label and a set of aliases. The Wiki-18 corpus (Karpukhin et al., 2020), which contains about 21M passages, serves two purposes: it is the retrieval index that the solver queries, and it provides a mapping from article titles to passage identifiers. The relational edges are drawn from Wikidata5M; Wiki-18 contributes only the title-to-passage mapping and the retrieval index, and it is not used as an independent source of graph structure.

Table 10: Representative functional and near-functional relation types retained in the whitelist for chain sampling. The whitelist contains approximately 40 relation types across five categories.

Category	Representative Relations
Authorship & creation	author (P50), creator (P170), composer (P86), director (P57), screenwriter (P58), performer (P175)
Geographic & institutional	country (P17), headquarters location (P159), country of origin (P495), location of formation (P740)
Affiliations & roles	occupation (P106), educated at (P69), employer (P108), position held (P39)
Production & distribution	manufacturer (P176), record label (P264), publisher (P123), production company (P272)
Works & achievements	notable work (P800), award received (P166), field of work (P101), genre (P136)

Entity grounding and normalization. We build a lookup from surface strings to Wikidata entity identifiers by normalizing every label and alias in Wikidata5M: we strip parenthetical disambiguation markers (for example, converting "Paris (city)" to "Paris"), apply Unicode NFKD normalization, and lowercase the result. An entity is considered groundable if its aligned article title is present in the title-to-passage mapping of Wiki-18. Entities whose normalized alias maps to more than one Wikidata identifier are marked as ambiguous and excluded during chain construction, which removes chains such as "Paris $\rightarrow$ France", where "Paris" could refer either to the city or to a mythological figure.

Subgraph induction and random walks. We first restrict Wikidata5M to the subgraph induced by groundable entities, retaining only triples whose head and tail both have articles in Wiki-18. On this induced subgraph, we perform random walks to sample chains of one to three hops. A walk is rejected if it revisits an entity (thereby forming a cycle), if it uses two or more geographic-nesting relations consecutively, or if any entity along the path lacks a retrievable source passage.

Typed-relation filtering. Wikidata5M contains many relations that are one-to-many or that otherwise do not determine a unique answer. We retain only a whitelist of functional or near-functional relation types, so that each hop maps a head entity to a well-defined target. The whitelist contains approximately 40 relation types, organized by category in Table 10. Each candidate chain is assigned a verdict. A chain is retained only if every relation belongs to the whitelist and every entity is unambiguously grounded; chains labeled as having no grounded identifier, ambiguous grounding, no typed relation, or excessive geographic nesting are discarded.

Two-phase retrieval verification. A well-typed chain is useful only if the solver can traverse it through retrieval. We verify each chain in two phases, using the same retriever and corpus that are used during training and evaluation.

Phase 1: Source-passage grounding. For each chain, we scan the passages associated with the source entity e_0 (retrieved using its article title) and require that at least one passage mentions, as a word-boundary substring, the entity e_1 that bridges to the second hop. This guarantees that a solver starting from the source passage can observe the first bridging entity in context.

Phase 2: Per-hop retrieval probes. For every intermediate hop $k \in \{1, \ldots, h-1\}$, we issue a retrieval query using the bridging entity e_k and examine the top-3 retrieved passages. We require that at least one of these passages belongs to the article of the expected target entity e_{k+1}. This ensures that the retrieval path the solver follows can reach every entity in the chain.

A chain is retained only if the source-passage check succeeds and all per-hop retrieval probes succeed. This two-phase verification ensures that every hop of the chain is reachable through the retrieval operations that the solver actually performs.

Pool statistics and training usage. Running the construction pipeline across multiple sampling batches yields approximately $100{,}000$ verified chains. For computational and storage efficiency, we retain a fixed subset of 76,000 verified chains for the experiments reported in this paper. This subset is sampled once using a fixed random seed and shared across all methods and ablations. It is substantially larger than the number of chains consumed during training, so the additional constructed chains are not required for the reported results. During training, we sample inputs from this pool on the fly, mixing one-, two-, and three-hop chains uniformly at a $1{:}1{:}1$ ratio. The proposer is trained for 50 update steps with a batch size of 256, so it consumes roughly $12{,}800$ chains per iteration, which is well below the size of the retained subset. The retained pool therefore provides ample headroom, and the uniform hop sampling ensures that each batch contains a balanced mix of hop counts (with an empirical standard deviation of about 3% per hop). Each sampled chain is rendered into a single or multi-turn prompt that presents the source passage together with the relational chain and its aligned evidence passages, from which the proposer generates a question and its target answer. The proposer objective and the information-gain reward are described in Sections 3.2 and 3.3.

A.6 Format Score and Validity Criteria

This appendix specifies the format score s_{fmt}, the grounding check, and the gate that jointly control when the information-gain reward is applied.

Parsing. Each proposer response is expected to contain a reasoning block followed by a question and its answer. From a response we extract an indicator has_think of whether a `<think>...</think>` block is present, together with the final `<question>` and `<answer>` spans, denoted q and a.

Integrity gate. A well-formed pair must satisfy

$$\text{integrity} = \mathbb{I}\big[\,|q| > 0 \;\wedge\; |a| > 0 \;\wedge\; \text{norm}(a) \not\subseteq \text{norm}(q)\,\big], \tag{10}$$

where $\text{norm}(\cdot)$ lowercases the text and removes punctuation and articles. This requires non-empty spans and forbids the answer from appearing inside the question. A pair failing integrity receives $s_{\text{fmt}} = 0$.

Score aggregation. For a pair passing the integrity gate, the score averages a base term, the presence of a reasoning block, and answer correctness:

$$s_{\text{fmt}} = \begin{cases} 0, & \text{integrity} = 0, \\ \dfrac{1 + \text{has_think} + \text{ans_correct}}{3}, & \text{integrity} = 1, \end{cases} \tag{11}$$

where $\text{ans_correct} = \mathbb{I}[\text{norm}(a) = \text{norm}(y)]$ compares the generated answer against the target answer y fixed by the chain. The score thus takes values in $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$.

Grounding check. Beyond structural correctness, we apply a grounding check that verifies where the answer can be found. For a single-hop question, the answer must appear in either the source passage or one of the evidence passages. For a multi-hop question, the answer must appear in a searched evidence passage along the chain, and a pair whose answer is supported only by the seed passage is rejected. This check discourages the proposer from posing nominally multi-hop questions whose answer is already available in the source.

Information-gain gate and total reward. The information-gain reward is granted only when the pair attains the maximum format score and passes the grounding check. Operationally, the IG term is retained only when $s_{\text{fmt}} = 1$ (enforced by a threshold $\gamma = 0.7$ on the discrete score) and the grounding check succeeds; otherwise it is set to zero. The full proposer reward is

$$r = \lambda\, s_{\text{fmt}} + \mathbb{I}[s_{\text{fmt}} = 1 \wedge \text{grounded}] \cdot s_{\text{IG}}, \tag{12}$$

with format weight $\lambda = 0.2$, so that the format signal contributes at most 0.2 and the information-gain reward dominates once a pair is valid. This composition ensures that the full-versus-shortcut likelihood comparison is applied only to structurally valid, correctly answered, and grounded question–answer pairs.

Table 11: Representative cases showing how the evaluated shortcut contexts prevent false-positive information-gain rewards. Here, Δ_{-b} denotes the gain computed after excluding shortcut context b, whereas Δ_B denotes the gain computed using the complete shortcut set defined in equation 5. Adding the source-only or individual-passage comparison changes an otherwise positive gain into a negative one, revealing that the target answer can already be supported by the corresponding shortcut context.

Case	Shortcut context	Gain without context	Gain with full set B	Effect
Lake Ontario → Saint Lawrence	Source-only	$\Delta_{-\mathrm{src}} = +1.422$	$\Delta_B = -0.953$	False positive → reject
Hypatima → Macquarie River	Individual passage	$\Delta_{-\mathrm{hop}} = +9.754$	$\Delta_B = -0.117$	False positive → reject

Table 12: Representative Dr. Zero proposer failures. Despite their nominal multi-hop labels, all three questions are generated without search and can be answered directly from the seed passage. Questions are reproduced verbatim from the generated data.

Case	Hop label	Searches	Generated question	Answer	Evidence in seed passage
A1	3	0	What is the fundamental group of a topological group G?	Abelian	"The fundamental group of a topological group G is abelian."
A2	2	0	At what speed range can the Arriflex D-20 camera operate in RAW format?	1 to 60 frame/s	"The camera is capable of running at speeds from 1 to 60 frame/s."
A3	4	0	On which beach did the 2/17th Infantry Battalion first land during the Allied invasion of Lae?	Red Beach	"The 2/17th Infantry Battalion came ashore on Red Beach behind the 2/15th."

A.7 Numerical Cases for Shortcut Ablation

Table 11 provides representative numerical cases showing how the evaluated shortcut comparisons prevent false-positive rewards. For the Lake Ontario case, excluding the source-only context gives a positive gain of $+1.422$, whereas including it changes the gain to -0.953. This result indicates that the seed passage already provides sufficient support for the target answer, so the question should not receive a positive information-gain reward. Similarly, the Hypatima case has a gain of $+9.754$ when the individual per-hop contexts are excluded, but the gain decreases to -0.117 after these contexts are included. This change reveals that a single evidence passage already provides sufficient answer information. Together, these cases illustrate how the source-only and per-hop comparisons suppress two common forms of shortcut: reliance on the seed passage and reliance on an individual evidence passage.

A.8 Representative Failure Cases of the Dr. Zero Proposer

Table 12 presents representative traces underlying the aggregate diagnosis in Section 4.4. The generated questions are reproduced verbatim from the proposer outputs, and the evidence column quotes the corresponding seed passages. Although these examples are labeled as requiring two to four hops, the proposer issues no search in any case, and each answer is stated directly in the seed passage. The examples therefore illustrate the semantic nature of the measured shortcut: the declared hop count reflects the requested generation format rather than the evidence actually required to answer the question.

A.9 Representative Multi-Hop Generations

Table 13 presents qualitative examples of question–answer pairs generated by the proposer from two- and three-hop relational chains. For each example, the knowledge-graph sampler first selects a relational path and retrieves an aligned evidence passage for each entity along the path. Given the resulting chain and evidence passages, the proposer generates a question whose target answer

Table 13: Representative unedited proposer generations and their corresponding construction chains. The chains are included for exposition and are not part of the generated outputs.

	Generated question	Construction chain	Answer
2-hop generations			
(1)	Which high-speed rail line serves the station that is a terminal for the Ōfunato Line?	Ōfunato Line → Ichinoseki Station → Tōhoku Shinkansen	**Tōhoku Shinkansen**
(2)	Which family of the payload specialist that flew on STS-41-D?	STS-41-D → Charles D. Walker → Walker family	**Walker family**
(3)	Which country is the home of the island where the Abruka rear lighthouse is located?	Abruka rear lighthouse → Abruka → Estonia	**Estonia**
3-hop generations			
(4)	In which country is the shipwreck discussed in Hugh Edwards's book located?	Hugh Edwards → *Islands of Angry Ghosts* → Batavia → Australia	**Australia**
(5)	Who is the creator of the James Bond novels that an airport from the country where Lacena Golding-Clarke represented in the Olympics is named after?	Lacena Golding-Clarke → Jamaica → Ian Fleming International Airport → Ian Fleming	**Ian Fleming**
(6)	Which fighting game marks the development company of the PlayStation game in the Mana series as its first venture into the fighting game genre?	Mana series → *Legend of Mana* → Square → *Tobal No. 1*	***Tobal No. 1***

Table 14: Training–test overlap across 51,713 evaluation questions. A similar Q–A pair has a question Jaccard similarity of at least 0.6 and at least one shared normalized answer alias.

Training source	Similar Q–A pairs	Exact questions
Dr. Free	25 (0.048%)	0 (0.000%)
Search-R1	1,108 (2.143%)	14 (0.027%)

is the terminal entity. The source entity is typically mentioned explicitly in the generated question, whereas intermediate entities remain implicit and are referred to through their relational descriptions.

For each example, we report the generated question, the target answer, and a compact representation of the underlying relational chain. The chain serves as generation metadata provided to the proposer rather than as part of the generated output, and the aligned evidence passages are omitted for readability. These examples demonstrate how the proposer verbalizes structured relational chains across different domains and hop depths, and allow direct inspection of whether the generated questions preserve the intended sequence of relations.

A.10 Training–Test Overlap Analysis

We examine whether the performance of Dr. Free may be attributed to overlap between its generated training data and the evaluation sets. For each of the 51,713 test questions across the seven benchmarks, we search for training questions with a normalized token-set Jaccard similarity of at least 0.6 and determine whether the matched questions share at least one normalized answer alias. We apply the same procedure to the 169,615 supervised training examples used by Search-R1 for reference. As shown in Table 14, only 25 test questions (0.048%) have both substantial lexical similarity and a shared answer with a Dr. Free training example. Manual inspection further shows that only four of these pairs ask for the same fact, and none contains an identical question. For comparison, the supervised Search-R1 training set contains 1,108 similar question–answer pairs and 14 exact question matches. These results indicate that direct training–test leakage is unlikely to account for the performance gains of Dr. Free.

Table 15: Robustness across independent training runs. Dr. Zero results are taken from the single runs reported by Yue et al. (2026). Dr. Free results are reported for three independent runs, followed by the mean and sample standard deviation. At both model scales, every Dr. Free run exceeds the corresponding average EM reported by Dr. Zero.

Method / Run	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen2.5-3B-Instruct								
Dr. Zero (reported)	0.397	0.572	0.431	0.298	0.291	0.091	0.200	0.326
Dr. Free (Run 1)	0.3765	0.5684	0.4057	0.3545	0.4008	0.1671	0.3440	0.3739
Dr. Free (Run 2)	0.3740	0.5777	0.4166	0.3357	0.3713	0.1514	0.3200	0.3638
Dr. Free (Run 3)	0.3795	0.5724	0.4159	0.3530	0.3903	0.1721	0.3440	0.3753
Dr. Free (Mean ± Std.)	0.3767 ± 0.0028	0.5728 ± 0.0047	0.4127 ± 0.0061	0.3477 ± 0.0104	0.3875 ± 0.0150	0.1635 ± 0.0108	0.3360 ± 0.0139	$\mathbf{0.3710 \pm 0.0063}$
Qwen2.5-7B-Instruct								
Dr. Zero (reported)	0.406	0.608	0.416	0.362	0.347	0.104	0.360	0.372
Dr. Free (Run 1)	0.3970	0.6104	0.4377	0.3869	0.3887	0.1638	0.4240	0.4012
Dr. Free (Run 2)	0.4011	0.6173	0.4238	0.3730	0.3727	0.1692	0.4480	0.4007
Dr. Free (Run 3)	0.3950	0.6212	0.4175	0.3822	0.3915	0.1746	0.4160	0.3997
Dr. Free (Mean ± Std.)	0.3977 ± 0.0031	0.6163 ± 0.0055	0.4263 ± 0.0103	0.3807 ± 0.0071	0.3843 ± 0.0101	0.1692 ± 0.0054	0.4293 ± 0.0167	$\mathbf{0.4005 \pm 0.0008}$

Table 16: Context construction and decision rules for the gold-blind counterfactual audit.

Condition	**Context and judge mode**	**Shortcut decision rule**
Closed-book	No evidence is supplied. The judge may answer using its parametric knowledge.	The judge returns `answerable=true`, and its answer matches the hidden gold.
Source-only	Only the seed document is supplied under the evidence-grounded prompt.	The judge returns `answerable=true` and `unique=true`, and its answer matches the hidden gold.
One-shot	The judge receives the top three passages from one retrieval using the training retriever. Each passage is truncated to 340 whitespace-delimited words.	The judge returns `answerable=true` and `unique=true`, and its answer matches the hidden gold.
Single-component	Each evidence-chain passage is supplied independently under the evidence-grounded prompt.	A shortcut is recorded if any component independently yields a unique answer matching the hidden gold.

A.11 Robustness Across Independent Runs

To assess sensitivity to training randomness, we run Dr. Free with three independent random seeds using Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, keeping the training budget, data-construction pipeline, retrieval setup, and evaluation protocol fixed. Table 15 reports individual runs and their mean and sample standard deviation. We include the single-run results reported for Dr. Zero (Yue et al., 2026) as a reference; since repeated Dr. Zero measurements are unavailable, this comparison serves as a robustness check rather than a formal significance test.

Across all seeds, Dr. Free exceeds the reported Dr. Zero average EM at both model scales. It achieves 0.3710 ± 0.0063 with the 3B backbone, compared with 0.326 for Dr. Zero, and 0.4005 ± 0.0008 with the 7B backbone, compared with 0.372. Even the lowest-performing runs retain absolute margins of 0.0378 and 0.0277, respectively. Although the 3B model exhibits greater seed sensitivity, particularly on multi-hop benchmarks, these results show that the improvements are not driven by a single favorable seed.

A.12 Gold-Blind Counterfactual Judge

We use GPT-4o through Azure OpenAI as the gold-blind judge for the counterfactual audit in Table 5. All calls use temperature 0 and JSON-constrained output. Each question–context condition is evaluated in an independent call, and the judge never observes the gold answer. For evidence-based conditions, the judge must answer solely from the supplied context and abstain unless the answer is uniquely supported. The closed-book condition instead permits the judge to use its parametric knowledge. Predicted answers are compared with the hidden gold answers only after generation. Table 16 summarizes the prompts, context construction, and shortcut decision rules.

A.13 Proposer Prompt

Table 17 presents the prompt template used for proposer-side question construction. For each sampled knowledge-graph chain, the source document, relational path, and aligned evidence passages are inserted into the corresponding placeholders. These materials are available only to the proposer: the proposer generates a question–answer pair without invoking search, whereas the solver receives only the generated question and must retrieve the required evidence independently. The

prompt also specifies hop-dependent composition constraints to preserve sequential dependence and prevent intermediate-entity leakage.

Table 17: Proposer prompt for question construction. The proposer receives a relational chain and aligned evidence, generates the question without search, and hides intermediate entities while preserving sequential dependence.

Complete proposer prompt. Placeholders in braces are instantiated for each sampled knowledge-graph chain.

Task definition

You are an expert question writer. You are given a source document and the evidence for every hop of a pre-validated relational chain. This evidence is available only to the proposer and will not be shown to the solver. Compose ONE search-oriented question (Q) and its single unambiguous answer (A), grounded in the supplied evidence. A one-hop question should require retrieval of the fact connecting the source entity to the answer; a multi-hop question should require sequential retrieval and composition of the intermediate evidence. Output exactly one assistant turn in the form `<think>...</think><question>...</question><answer>...</answer>`.

Dynamic input

Source document (Hop 0)	**Pre-validated chain**	**Per-hop evidence**
{document}	{chain_summary}	{evidence_block}

Treat the passages as the complete construction evidence available to the proposer.

Question rules

Q1. **Real question.** Begin with a wh-word or auxiliary verb, end with exactly one question mark, and match the wh-word to the answer type.

Q2. **No imperative.** Never begin with *Identify*, *Name*, *List*, *Describe*, *Explain*, *State*, *Provide*, or *Give*.

Q3. **One question.** Do not use compound constructions such as "and what" or "and who."

Q4. **Self-contained.** Do not refer to the document, source, passage, text, or article, or use expressions such as "according to" or "as mentioned in."

Q5. **Length and leakage.** Follow the hop-specific length limit, use only one question mark, and do not reveal the final answer.

Q6. **Hidden bridges.** Mention only Hop 0 by name. Hops $1, \ldots, n-1$ must remain implicit and appear only through descriptions grounded in the preceding hop.

Q7. **Surface form.** Follow the hop-specific form constraint given below.

Q8. **Supported modifiers.** Ground every modifier in Hop 0 or a resolved hop; do not add unsupported details to force uniqueness.

Chain-composition rules

Let Hop $0 \to$ Hop $1 \to \cdots \to$ Hop n, where Hop n is the answer and Hops $1, \ldots, n-1$ are hidden bridges.

C1. **Bounded depth.** Use zero subordinate clauses for one hop, at most one for two hops, and at most two for three hops. Never use three nested clauses or three consecutive "of the X" phrases.

C2. **Sequential dependence.** Resolve each entity from the preceding entity. Use narrow entity types and concrete relations rather than vague connectors such as *associated with*, *related to*, or *linked to*.

C3. **No attribute stacking.** Do not flatten multiple hops into parallel attributes.

C4. **No mixed-hop clauses.** Each clause may describe at most one hop.

C5. **No single-hop disguise.** Do not phrase an intermediate clue so that the answer is directly recoverable from Hop 0.

C6. **Functional resolution.** Each relation must uniquely determine the next entity. Reject underdetermined endpoints and reverse-functional relations.

Hop-specific form constraint

One hop. Use 4–12 words. Begin with a wh-word or auxiliary verb, mention only the starting entity, and use no subordinate clause introduced by *whose*, *which*, or *that*. Reject declarative fragments.

Two or three hops. Use 8–18 words for two hops or 12–22 words for three hops. Mention Hop 0 exactly once and compose hidden bridges using possessives, prepositional phrases, or at most two subordinate clauses. Reject repetition, excessive nesting, overlength questions, declarative fragments, and bridge attributes stacked with *and* or *with*.

Answer rules

Output only the canonical name of Hop n in 1–5 words. Do not include an article, explanation, alternative answer, alias list, confidence statement, or terminal punctuation. The answer must be grounded in the supplied chain and must not appear in the question.

Output format (exact)

```
<think>
  Hop k | relation:  ...  | referent:  ...  | unique?  yes/no
  ...
  Composing:  ...  (compose from the innermost relation outward)
</think>
<question> the generated question </question>
<answer> the final answer </answer>
```