CAN RETRIEVAL FAILURES DRIVE GRAPH EVOLUTION? SELF-IMPROVING GRAPHS FOR EVIDENCE RECOVERY IN MULTI-HOP QA

Anonymous authors
Paper under double-blind review

ABSTRACT

Frontier models are increasingly expected to be self-improving, refining their reasoning, tools, and environments rather than remaining static after training. However, optimization typically targets only the policy, while the environment remains fixed. Multi-hop question answering makes this gap concrete: answering requires retrieving, connecting, and verifying evidence across documents, yet existing retrieval-augmented and graph-based systems operate over fixed retrieval environments, and reinforcement learning (RL) approaches typically rely on terminal answer rewards. Such rewards provide limited guidance on which retrieval decisions contribute to success or failure, and policy optimization alone cannot repair deficiencies in how the environment represents and connects evidence. We present SIG-R1, a self-improving graph method that combines retrieval-step rewards with failure-driven adaptation of a searchable hypergraph. SIG-R1 assigns process-aware credit to individual retrieval steps, diagnoses recurring failures, and admits only corpus-supported edits. Experiments show that, with Qwen3.5-9B, SIG-R1 improves macro-F1 by **6.4 points** over Graph-R1, including a **12.0-point gain** on HotpotQA. Three-seed ablations show higher mean F1 with graph evolution and dataset-dependent additional gains from process rewards. These results support coupling retrieval-policy learning with evidence-constrained environment adaptation. Our code is available at https://anonymous.4open.science/status/SIG-R1-DC07.

1 INTRODUCTION

Self-evolving large language model (LLM) agents are promising: they refine their reasoning, reward signals, and tool-using workflows from interaction feedback (Zelikman et al., 2023; Yuan et al., 2024; Tao et al., 2024; Zhang et al., 2026), frontier laboratories such as Anthropic[1] and OpenAI[2] now pursue such recursive self-improvement (RSI) explicitly. Multi-hop question answering (QA) is a natural testbed for this agenda: a system must identify, connect, and verify evidence distributed across multiple documents or entities (Ho et al., 2020), and durable improvement requires advancing not only the model but also the environment it searches. Retrieval-augmented generation (RAG) has substantially improved the factuality and extensibility of language models (Lewis et al., 2020; Asai et al., 2024), and graph-based retrieval provides an explicit structure for representing entities and relations (Edge et al., 2024; Gutiérrez et al., 2024; Gutiérrez et al., 2025); nevertheless, reliable multi-hop reasoning remains difficult because a system must jointly determine what to retrieve, how to connect evidence, and when the accumulated evidence is sufficient to support an answer.

Existing approaches can be organized into three lines: dense retrieval, graph-structured retrieval, and agentic reasoning. Dense retrieval selects passages that appear relevant to the question and grounds generation in an external index (Mallen et al., 2023; Lewis et al., 2020). Graph-structured retrieval exposes entity or relation structure explicitly, coupling vector search with entity graphs, community summaries, or personalized PageRank over a knowledge graph (Gutiérrez et al., 2024; Guo et al., 2025b; Edge et al., 2024). Agentic reasoning interleaves free-form reasoning with retrieval actions and, more recently, trains the search behavior itself with outcome-based reinforcement learning (Jin et al., 2025; Song et al., 2025; Asai et al., 2024; Yao et al., 2022; Luo et al., 2025).

[1] https://www.anthropic.com/institute/recursive-self-improvement
[2] https://blog.samaltman.com/the-gentle-singularity

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

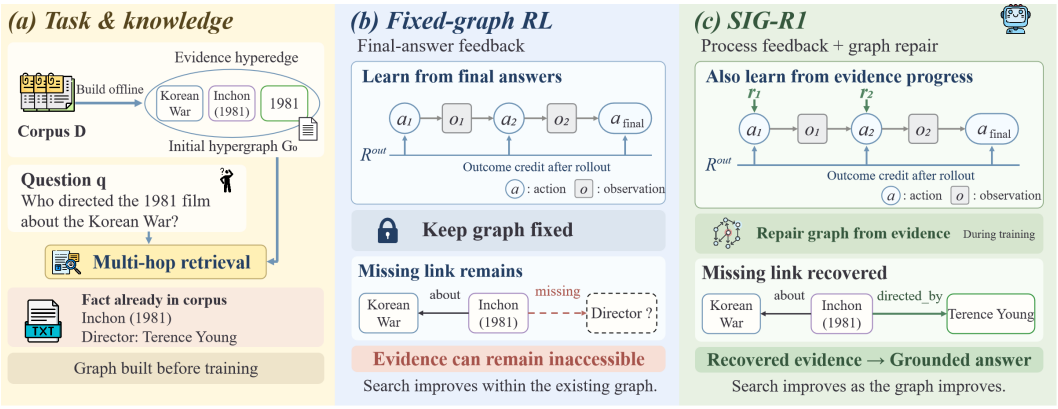


Figure 1: **Motivation and overview of SIG-R1.** (a) Answering the question requires chaining two facts; the answer-bearing fact is in the corpus, but the offline-built hypergraph misses the link. (b) Fixed-graph RL improves search *within* the existing graph, the missing evidence remains inaccessible. (c) SIG-R1 adds process feedback and corpus-supported graph repair during training, thus recovering the missing link.

Despite these advances, graph construction and retrieval are typically fixed before policy learning, leaving the retrieval environment unable to adapt to downstream failures. As illustrated in Figure 1, offline construction may omit answer-bearing facts, incomplete retrieval descriptions, or bridge relations required for multi-hop reasoning, making such evidence unrecoverable through query-policy optimization alone. Meanwhile, terminal rewards provide only sequence-level feedback, offering limited credit assignment for intermediate retrieval decisions and obscuring the difference between partial progress and uniformly poor trajectories.

We therefore study multi-hop QA in a setting where both search policy and retrieval environment may be imperfect. Given an initial hypergraph, a corpus, and gold evidence available only for training examples, our objective is to learn a retrieval policy while selectively repairing persistent graph defects in a way that preserves factual grounding, avoids introducing information unsupported by the corpus, and generalizes beyond the questions that trigger an edit. This setting exposes a central challenge because failures may originate in the policy, in unsupported or shortcut answers, in an incorrect multi-hop path, or in the graph itself, and treating every failure as a policy error is insufficient when further optimization cannot resolve a defect in the retrieval environment.

To address this problem, we propose *SIG-R1* (*Self-Improving Graph Reasoning with Reinforcement Learning*), a closed-loop framework that co-evolves the retrieval policy and a verifiable hypergraph environment. SIG-R1 tracks the entities, hyperedges, and queries involved at each retrieval step and computes process-aware rewards from evidence gain, graph connectivity, answer-entity matching, and local support. These rewards are assigned only to the corresponding policy-generated retrieval tokens and normalized within each question and step. In parallel, SIG-R1 attributes persistent failures to policy, grounding, path, or graph defects, with only graph defects triggering environment evolution. Candidate hyperedge additions or retrieval-text rewrites are constructed from training evidence and accepted only after passing source-fidelity, entity-ownership, and retrievability checks. Accepted edits are incrementally applied to the graph and index, allowing subsequent policy rollouts to operate in the improved environment. Policy updates alternate with graph checks every five updates, forming successive evolution rounds.

We evaluate SIG-R1 on three QA benchmarks with matched policy-update budgets and development-selected checkpoints. Three-seed component ablations compare process-aware rewards and graph evolution separately and jointly. Graph evolution is the larger standalone contributor on both ablation datasets, while their combination achieves the highest mean F1 and exact match (EM). Evolution trajectories show an early batch of accepted repairs followed by smaller updates, and evidence-conditioned outcomes locate most of the accuracy gain in correct answers accompanied by retrieved evidence. Zero-shot evaluation on fixed target graphs further characterizes how the learned policy transfers beyond the source training environment.

Our contributions are as follows:

① ***Verifiable graph evolution.*** We develop a self-improving hypergraph mechanism in which candidate edits are constructed from training evidence and inserted into the graph and retrieval

index only after satisfying quote-level, entity-ownership, and pre-commit retrieval-score constraints, thereby enabling environment adaptation while limiting unsupported knowledge injection. The graph is held fixed at test time.

② ***Step-level retrieval credit.*** We introduce a structured credit-assignment mechanism for graph retrieval trajectories that combines evidence novelty, graph connectivity, answer-entity matching, and local graph support to provide dense supervision for individual retrieval decisions in addition to the terminal reward shared across the trajectory.

③ ***Persistent-failure routing.*** We propose an attribution and action-routing scheme that separates policy failures, grounding failures, path failures, and graph defects by aggregating persistent failures across multiple rollouts before intervention, reserving environment modification for evidence-verified graph defects, and retaining the other categories as channels for training signals and diagnostics.

④ ***Periodic graph updates and evaluation.*** We establish a closed-loop training procedure that updates the retrieval policy after each batch and checks the retrieval environment every five updates. We evaluate the framework through three-seed component ablations, graph-edit trajectories, evidence-conditioned analysis, and zero-shot transfer, disentangling the effects of policy learning and environment adaptation while assessing whether the learned policy generalizes to new datasets without transferring source-graph edits.

2 Related Work

Retrieval-augmented and graph-structured question answering. Retrieval-augmented generation (RAG) couples a language model with an external index (Lewis et al., 2020), which is useful when parametric memory is incomplete or unreliable (Mallen et al., 2023). For multi-hop QA, graph-based systems expose part of the required connectivity explicitly: GraphRAG builds entity graphs and community summaries (Edge et al., 2024), LightRAG combines graph and vector retrieval (Guo et al., 2025b), and HippoRAG propagates evidence over a knowledge graph with personalized PageRank (Gutiérrez et al., 2024). Node- and path-oriented variants alter the representation or construction of the question-specific evidence structure (Xu et al., 2025; Zhuang et al., 2026; Huang et al., 2026a;b). These approaches improve access to multi-hop context, but the graph and its retrieval descriptions are normally fixed after construction; omitted entities, relations, or answer-bearing text therefore remain inaccessible to the retriever.

Agentic retrieval and reinforcement learning. Agentic retrieval interleaves reasoning with external actions (Yao et al., 2022), while SELF-RAG trains retrieval requests and evidence critique (Asai et al., 2024). Graph-R1, GraphRAG-R1, and HyperGraphPro optimize graph or hypergraph tool use with end-to-end, process-constrained, or progress-aware reinforcement learning (Luo et al., 2025; Yu et al., 2026; Park et al., 2026); GraphRAG-Router and PathRouter further shape routing decisions with cost or path-quality signals (Fan et al., 2026; Wang et al., 2026a). Process-supervision methods such as TIPS and MatchTIR provide finer credit for search or tool-call steps (Xie et al., 2026; Qu et al., 2026). Such work improves what the policy does during a rollout, but leaves the underlying evidence index unchanged.

Adaptive retrieval environments. Adaptive environments relax this fixed-index assumption: Agentic-KGR and SAGE update graph structure or memory from interaction feedback (Li et al., 2025; Wang et al., 2026b), and EvoGraph-R1 edits multimodal hypergraphs during retrieval (Lin et al., 2026). These systems establish environment adaptation as a useful direction, but do not jointly provide graph-semantic step credit, explicit attribution of persistent failures, and provenance-gated edits. SIG-R1 closes this gap by routing only graph-attributed failures to corpus-grounded repairs while retaining policy, grounding, and path failures as training and diagnostic signals.

3 Methodology

SIG-R1 augments a hypergraph retrieval RL agent with process-aware, per-step credit assignment and failure-driven, gated graph evolution. The first mechanism supplies structural rewards for retrieval actions, and the second repairs persistent retrieval defects through verified edits during training. Figure 2 summarizes the design.

3.1 Problem Formulation

Let G_0 be the initial hypergraph and $G = (\mathcal{V}, \mathcal{E}, \mathcal{I})$ its working copy, with entities $\mathcal{V}$, hyperedges $\mathcal{E}$, and retrieval index $\mathcal{I}$. Each entity v has retrieval text $\mathrm{rt}(v)$; each hyperedge e stores evidence $\mathrm{ct}(e)$

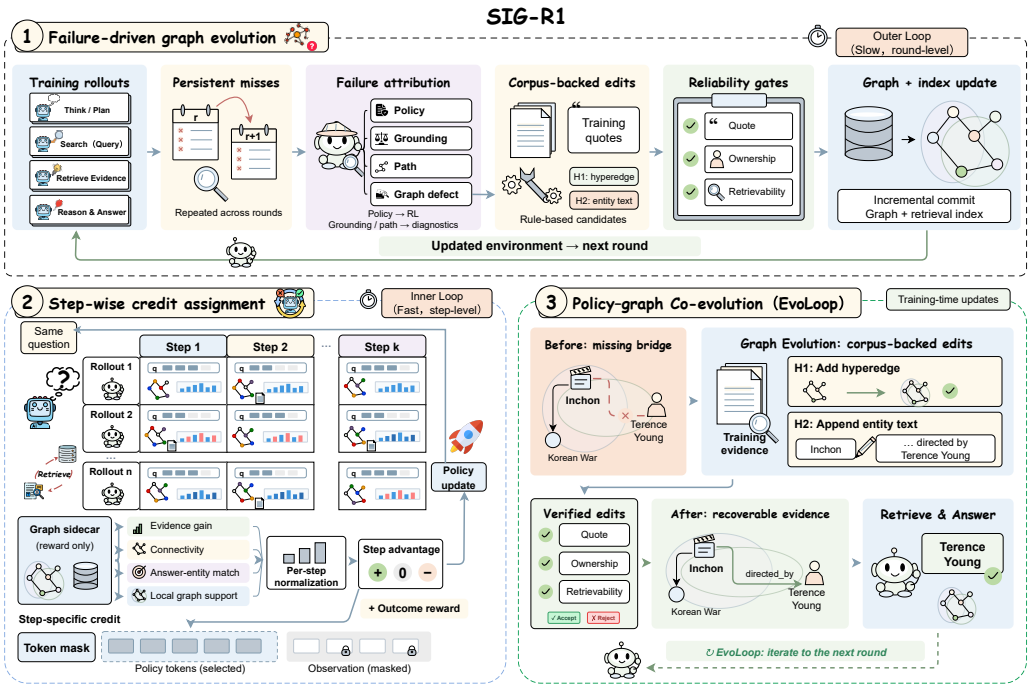


Figure 2: **SIG-R1 framework.** For each retrieval step, SIG-R1 records the queried and retrieved entities and hyperedges and assigns a process reward to the associated policy-generated segment. Every five policy updates, an evolution round aggregates persistent failures, diagnoses their causes, and routes graph-defect candidates to evidence-constrained editing. Candidate edits are accepted only after quote, ownership, and retrievability verification, after which the graph and retrieval index are incrementally updated for subsequent rollouts.

linking entities $\text{link}(e) \subseteq \mathcal{V}$. Retrieval combines entity and hyperedge search, graph traversal, and corpus passages to return ten evidence texts.

For question q, policy π_θ with parameters θ generates a rollout (interaction trajectory) $\tau = (a_1, o_1, \ldots, a_K, o_K, a_{\text{final}})$, where a_k is a query, o_k its retrieved observation, K the number of retrieval calls, and a_{final} the answer. Gold answer aliases $\mathcal{A}_q$ are reference strings used for scoring, never policy inputs. The terminal reward $R^{\text{out}} \in [-1, 1]$ combines format validity and token-level F1. We sample n rollouts per question and use group-relative policy optimization (GRPO) (Shao et al., 2024) with a clipped proximal policy optimization (PPO) objective (Schulman et al., 2017); retrieved tokens receive no policy loss. Retrieval and optimization details are in Appendix B.1.

3.2 Process-Aware Credit Assignment

Step reward components. At retrieval step k, a sidecar (metadata hidden from the policy) records query-linked entities Q_k, retrieved entities E_k, and hyperedges H_k. Let $S_{k-1} = \bigcup_{j<k} E_j$ ($S_0 = \emptyset$), $F_k = S_{k-1} \cup Q_k$, and $D_k = \max\{1, |E_k|\}$. We reward entity gain, overlap with this reference set, answer matching, and local structural support:

$$\text{gain}_k = \frac{|E_k \setminus F_k|}{D_k}, \quad (1) \qquad \text{conn}_k = \frac{|E_k \cap F_k|}{D_k}, \quad (2)$$

$$\text{supp}_k = \mathbf{1}\{\text{match}(E_k, \mathcal{A}_q)\}, \quad (3) \qquad \text{valid}_k = \mathbf{1}\{H_k \neq \emptyset \wedge E_k \cap F_k \neq \emptyset\}. \quad (4)$$

Here $|\cdot|$, $\setminus$, and $\cap$ denote cardinality, set difference, and intersection; $\mathbf{1}$ is the indicator function. $\text{match}(E_k, \mathcal{A}_q)$ tests whether any retrieved entity name matches an answer alias after normalization (false for yes/no questions). The four terms are heuristic signals; valid_k checks local support, not an entire reasoning path. Their weighted sum is

$$r_k = w_p \,\text{gain}_k + w_c \,\text{conn}_k + w_r \,\text{supp}_k + w_{\text{path}} \,\text{valid}_k, \quad (5)$$

with weights $(w_p, w_c, w_r, w_{\text{path}}) = (0.04, 0.03, 0.05, 0.03)$. Outcome rewards are standardized within each question group, and step rewards within the subgroup containing step k. The resulting

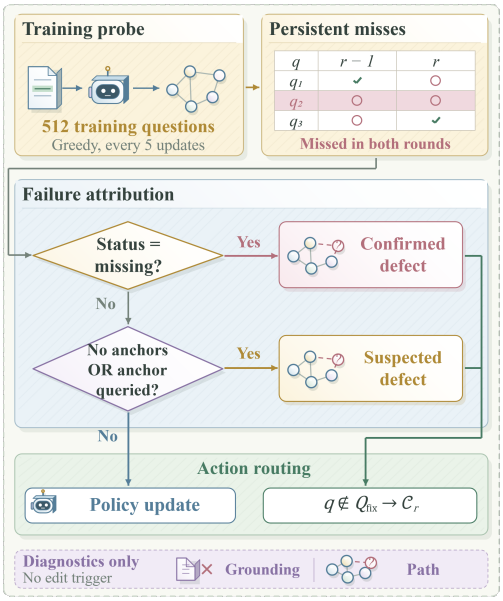


Figure 3: **Failure attribution and routing.** Checks/circles mark probe hits/misses. Persistent graph-defect candidates outside the previously repaired set Q_{fix} enter $\mathcal{C}_r$; grounding and path labels remain diagnostics.

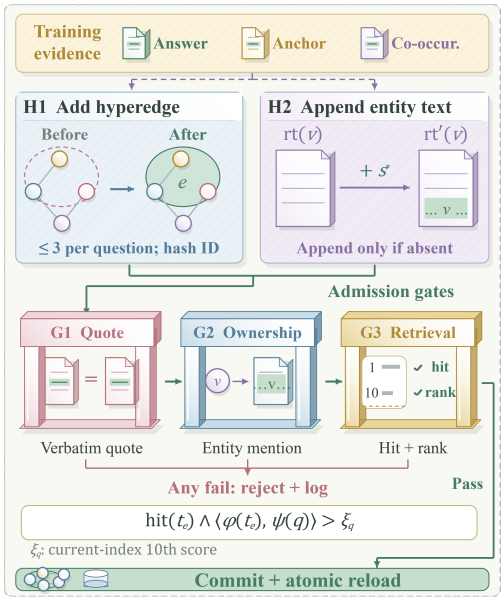


Figure 4: **Evidence-constrained graph evolution.** H1 adds hyperedges; H2 appends entity text. All three gates precede commit. G3 tests candidate text t_e against the current tenth-ranked score ξ_q; graph motifs are schematic.

step advantage applies only to policy tokens in that retrieval segment. Appendix B.6 gives reward properties, token placement, and normalization details.

3.3 Failure Attribution and Action Routing

Process rewards improve retrieval decisions; persistent evidence misses can also require graph repair. Every I policy updates, SIG-R1 runs a fixed training probe P of 512 questions using greedy decoding. A question becomes eligible after m consecutive probes retrieve no gold answer alias. Evolution round r denotes the check after update rI.

Figure 3 gives the ordered attribution rules, which use training-evidence status and entity-name hints to distinguish confirmed or suspected graph defects from query-policy failures. Only graph-defect candidates enter the edit set $\mathcal{C}_r$, excluding questions already repaired (Q_{fix}). Query failures are handled by RL; grounding (failure to use retrieved evidence) and path (failure to connect reachable facts) remain diagnostics. Appendix B.2 formalizes persistence and routing.

3.4 Gated Graph Self-Evolution

SIG-R1 constructs two edit types from training-split gold evidence (Figure 4): **H1** adds hyperedges connecting the relevant entities, and **H2** appends a verbatim answer-bearing sentence s^* to an entity's retrieval text, producing $\text{rt}'(v)$. Neither operation generates new facts with a language model.

Each edit e must pass three Boolean gates: $g_1(e)$ checks verbatim source quotes, $g_2(e)$ checks that the fact mentions its associated entity, and $g_3(e)$ checks answer relevance and retrieval visibility. Let t_e be the proposed hyperedge or updated entity text, ϕ and ψ the BAAI General Embedding (BGE) text and query encoders (Xiao et al., 2024), and ξ_q the current index's tenth-ranked score for the triggering question q. Then

$$g_3(e) \iff \text{hit}(t_e) \wedge \langle \phi(t_e), \psi(q) \rangle > \xi_q, \tag{6}$$

where hit detects an answer alias or its key evidence span. This pre-commit score filter does not guarantee a hit after hybrid retrieval. Accepted edits update the per-run graph and index together; test-time graphs remain fixed. Construction rules and span matching are detailed in Appendix B.3, and data isolation in Appendix B.7.

3.5 The Closed Training Loop

Across $N = 40$ policy updates, each batch trains the policy on the current graph; every fifth update probes failures and admits verified edits. An edit accepted in round r affects rollouts from update

$rI + 1$ onward. Algorithm 1 gives the complete loop, including reward masking and graph reloads; run constants appear in Appendix B.

4 EXPERIMENTS

We evaluate overall QA performance, isolate the two training mechanisms, and analyze when graph edits occur, how evidence accompanies correct answers, and which gains transfer beyond the training environment. Five research questions (RQs) organize these experiments:

RQ1: Does SIG-R1 improve QA across scales and model families at a fixed RL budget?

RQ2: How much do process rewards and graph evolution each contribute?

RQ3: When are graph repairs accepted, and how selective are the gates?

RQ4: Are accuracy gains concentrated in answers with retrieved evidence?

RQ5: Do policy gains transfer to new datasets without transferring graph edits?

4.1 EXPERIMENTAL SETUP

4.1.1 DATASETS AND EVALUATION.

We evaluate Natural Questions (NQ) (Kwiatkowski et al., 2019), 2WikiMultiHopQA (Ho et al., 2020), and HotpotQA (Yang et al., 2018) using Graph-R1's released partitions (Luo et al., 2025): 5,120 training and 128 test examples per dataset, with development-based checkpoint selection (Appendix B.7). Metrics are normalized exact match (EM) and token-level F1, both in percent. Out-of-distribution (OOD) transfer uses MuSiQue (Trivedi et al., 2022), PopQA (Mallen et al., 2023), and TriviaQA (Joshi et al., 2017), with fixed target graphs and no additional training.

4.1.2 BASELINES.

Controls include direct generation (NaiveGeneration), standard retrieval-augmented generation (StandardRAG), and supervised fine-tuning (SFT). RL comparisons include R1 (Guo et al., 2025a), Search-R1 (Jin et al., 2025), R1-Searcher (Song et al., 2025), and Graph-R1 (Luo et al., 2025), all with equal policy-update budgets. The latter is the direct fixed-graph comparison; the other baselines place graph-based RL in the broader context of generation, retrieval, and search training. We use Qwen3.5-2B/4B/9B and Llama-3.1-8B, with 40 policy updates and five sampled rollouts per question. Full implementation details appear in Appendix B.

4.2 MAIN RESULTS (RQ1)

Table 1 compares four policy models on three benchmarks under a shared evaluation protocol and equal policy-update budgets for RL methods. Average F1 is the unweighted mean over the three datasets, so each benchmark contributes equally. The 4B SIG-R1 row reports three-seed means; Appendix B.4 gives the individual runs, while the other main-table rows report single runs.

Obs. 1: SIG-R1 achieves the highest average F1 for all four policy models. The aggregate advantage holds across model sizes and both model families, with the largest dataset-level margins on multi-hop QA at the larger scales. For Qwen3.5-9B, SIG-R1 reaches 65.66 average F1 versus 59.28 for Graph-R1; on HotpotQA, F1 rises from 59.64 to 71.66 and EM from 52.34 to 64.84. The agreement between F1 and EM on this benchmark shows that the improvement includes more exact answers, rather than token overlap alone.

Obs. 2: The observed margin over Graph-R1 widens with Qwen model scale. The average-F1 gap grows from $+1.75$ at 2B to $+3.39$ at 4B and $+6.38$ at 9B. At 2B, SIG-R1 remains below Graph-R1 on 2WikiMultiHopQA F1 (54.91 versus 55.66), whereas at 9B it improves F1 on all three datasets. Thus, the combined training approach becomes more effective relative to the fixed-graph baseline as the evaluated Qwen models grow. This pattern is consistent with stronger policies making better use of an adapted retrieval environment; the component ablation below tests the contribution of that adaptation separately.

Obs. 3: The gains extend to Llama-3.1-8B on every benchmark. SIG-R1 improves both EM and F1 over Graph-R1 across all three Llama evaluations. The larger margins on the two multi-hop benchmarks mirror the stronger Qwen results, supporting applicability beyond the Qwen family.

4.3 COMPONENT ABLATION (RQ2)

We train four Qwen3.5-4B variants on NQ and 2WikiMultiHopQA: outcome rewards on a fixed graph (w/o Both), process rewards only (w/o Graph Evol.), graph evolution only (w/o Proc. RL), and full SIG-R1. All variants use three seeds, 40 policy updates, and development-selected checkpoints

Table 1: Test EM/F1 (%). Average is the unweighted mean over NQ, 2WikiMultiHopQA (2Wiki), and HotpotQA. Bold/underline mark the best/second-best scores per model. SIG-R1 4B uses three-seed means (Appendix B.4); other rows are single runs.

Model	Method	NQ		2Wiki		HotpotQA		Average	
		EM	F1	EM	F1	EM	F1	EM	F1
Qwen3.5-2B	NaiveGeneration	10.16	19.40	15.62	19.36	16.41	21.50	14.06	20.09
	StandardRAG	22.66	38.05	24.22	28.28	41.41	46.77	29.43	37.70
	SFT	10.16	17.38	35.16	39.75	15.62	21.23	20.31	26.12
	R1	10.94	21.02	25.00	28.68	19.53	25.82	18.49	25.17
	Search-R1	15.63	23.48	29.69	32.40	17.97	21.32	21.10	25.73
	R1-Searcher	9.38	15.75	24.22	29.60	33.59	39.71	22.40	28.35
	Graph-R1	25.00	35.36	**50.78**	**55.66**	**46.09**	51.28	**40.62**	47.43
	SIG-R1 (Ours)	**27.34**	**39.50**	47.66	54.91	**46.09**	**53.13**	40.36	**49.18**
Qwen3.5-4B	NaiveGeneration	13.28	22.63	9.38	13.33	21.88	23.33	14.85	19.76
	StandardRAG	20.31	30.85	17.97	22.28	32.03	36.63	23.44	29.92
	SFT	16.41	27.50	37.50	42.44	24.22	29.79	26.04	33.24
	R1	17.97	30.50	27.34	32.34	28.13	33.59	24.48	32.14
	Search-R1	24.22	34.53	33.59	36.83	26.56	31.31	28.12	34.22
	R1-Searcher	17.97	31.02	34.38	38.57	42.19	49.40	31.51	39.66
	Graph-R1	37.50	**51.87**	58.59	65.85	50.00	58.09	48.70	58.60
	SIG-R1 (Ours)	**38.02**	51.05	**62.76**	**70.48**	**56.77**	**64.44**	**52.52**	**61.99**
Qwen3.5-9B	NaiveGeneration	18.75	27.86	15.62	22.41	19.53	23.59	17.97	24.62
	StandardRAG	27.34	38.52	23.44	29.53	38.28	44.87	29.69	37.64
	SFT	22.66	33.20	35.94	39.34	32.81	39.28	30.47	37.27
	R1	21.09	35.85	27.34	33.27	27.34	32.04	25.26	33.72
	Search-R1	27.34	39.59	29.69	33.24	42.19	48.93	33.07	40.59
	R1-Searcher	21.09	32.76	20.31	24.31	33.59	40.81	24.99	32.63
	Graph-R1	32.03	46.46	62.50	71.75	52.34	59.64	48.96	59.28
	SIG-R1 (Ours)	**35.16**	**48.57**	**71.09**	**76.76**	**64.84**	**71.66**	**57.03**	**65.66**
Llama-3.1-8B	NaiveGeneration	19.53	39.77	17.19	23.55	30.47	37.62	22.40	33.65
	StandardRAG	32.03	46.19	27.34	39.35	51.56	58.66	36.98	48.07
	SFT	23.44	37.18	41.41	45.88	32.81	39.78	32.55	40.95
	R1	28.12	47.80	26.56	32.58	33.59	41.75	29.42	40.71
	Search-R1	21.88	37.59	25.78	29.93	32.03	36.48	26.56	34.67
	R1-Searcher	21.09	38.08	27.34	32.32	35.16	40.42	27.86	36.94
	Graph-R1	35.94	49.23	57.03	64.28	50.78	56.80	47.92	56.77
	SIG-R1 (Ours)	**36.72**	**50.89**	**62.50**	**67.92**	**54.69**	**61.64**	**51.30**	**60.15**

Table 2: Component ablation (4B, three-seed test means, %). Proc. RL: process-aware RL; Graph Evol.: graph evolution; w/o Both: Graph-R1.

Method	2Wiki		NQ	
	F1	EM	F1	EM
SIG-R1	**70.48**	**62.76**	**51.05**	**38.02**
w/o Proc. RL	68.49	60.94	50.77	37.50
w/o Graph Evol.	67.35	59.90	48.40	35.94
w/o Both	67.06	59.90	46.85	35.16

Table 3: Held-out outcomes (%). Corr./Incorr.: correct/incorrect; Evid./No evid.: detected/absent evidence in retrieved text. Measurement details: Appendix B.8.

	4B		9B	
Outcome	Graph-R1	SIG-R1	Graph-R1	SIG-R1
Corr. + Evid.	47.4	53.7	46.9	53.9
Corr. + No evid.	1.3	1.3	2.1	2.6
Incorr. + Evid.	22.7	16.4	23.2	18.0
Incorr. + No evid.	28.6	28.6	27.9	25.5

(Table 2). The w/o Both arm is the three-seed Graph-R1 control for this experiment; its means are distinct from the single-run Graph-R1 row in Table 1.

Obs. 4: Process-aware rewards improve F1 even when the graph is fixed. Adding step credit raises NQ F1 from 46.85 to 48.40 and 2WikiMultiHopQA F1 from 67.06 to 67.35. NQ EM also improves, while 2WikiMultiHopQA EM remains 59.90. These comparisons isolate a benefit from supervising retrieval segments without changing graph content. The unequal gains indicate that dense credit is more useful for some retrieval tasks than others; by itself, it cannot supply evidence missing from the fixed environment.

Obs. 5: Graph evolution is the larger standalone contributor. With outcome rewards unchanged, enabling evolution improves F1 by 3.92 points on NQ and 1.43 on 2WikiMultiHopQA, exceeding the respective process-only gains of 1.55 and 0.29. Evolution alone also attains higher EM than

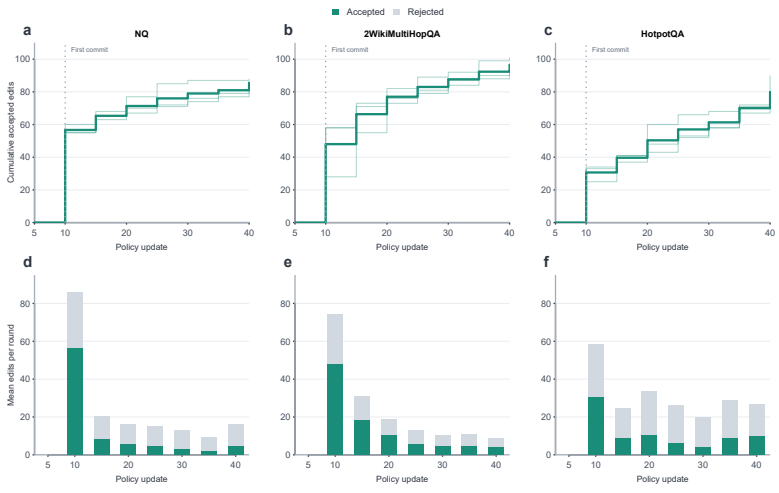


Figure 5: **Failure-driven graph evolution across training.** Top: cumulative accepted edits on NQ, 2WikiMultiHopQA, and HotpotQA; thin lines show individual runs and the dark line is their mean. Bottom: mean accepted and rejected candidate edits at each evolution round. The dotted line marks the first eligible commit after two consecutive failure probes.

process rewards alone on both datasets. Adapting the searchable environment therefore adds value beyond improving retrieval-step credit under the same policy-update budget. The admitted content comes from training evidence through the quote, ownership, and retrieval-score gates, tying this intervention to source-supported repair rather than free-form generation of new facts.

4.4 Failure-Driven Evolution Dynamics (RQ3)

Figure 5 shows the three runs per dataset, their mean cumulative accepted edits, and mean accepted and rejected counts per round. Probes occur every five updates; two consecutive misses make update 10 the first eligible commit

Obs. 6: The largest repair batch occurs early enough to affect most of policy training. At update 10, the mean accepted counts are 56.7 on NQ, 48.0 on 2WikiMultiHopQA, and 30.7 on HotpotQA, the largest single-round batch in each dataset. These changes are available for the remaining 30 policy updates; mean cumulative counts reach 85.7, 96.7, and 80.0 by update 40. Evolution thus changes the evidence available to subsequent learning, rather than operating only as a correction after policy training has ended.

Obs. 7: Later evolution rounds remain selective instead of admitting every candidate. Accepted batches become smaller after the initial round, while rejected candidates persist throughout training. On NQ, for example, update 40 accepts 4.7 candidates and rejects 11.3 on average, compared with 56.7 accepted at update 10. The graph continues to adapt, but candidate generation does not automatically become graph growth. Together with the ablation, these trajectories show when the beneficial evolution component acts and that its admission gates actively filter proposed changes; the counts measure selectivity, not an independent factuality audit.

4.5 Evidence-Conditioned Answer Quality (RQ4)

Table 3 partitions 384 held-out answers per setting into four joint outcomes according to normalized-EM correctness and evidence presence in the retrieved text. Evidence is detected by an offline checker applied to the model-visible retrieval blocks, including a separate check for yes/no questions. Appendix B.8 details the measurement protocol.

Obs. 8: Most of the accuracy gain occurs in correct answers accompanied by evidence. The correct-with-evidence share rises from 47.4% to 53.7% at 4B and from 46.9% to 53.9% at 9B. Correct answers without detected evidence remain at 1.3% at 4B and increase by only 0.5 percentage points at 9B. Thus, the evidence-accompanied category accounts for the full gain in this 4B profile and most of the gain at 9B. SIG-R1's advantage is concentrated in answers for which relevant evidence is also present in the visible context.

Obs. 9: The improvement extends beyond a higher aggregate evidence-hit rate. At 4B, the total evidence-present share stays at 70.1%, while the incorrect-with-evidence share falls from 22.7% to 16.4%. Calculated from these joint percentages, correctness among evidence-present answers

rises from 67.6% to 76.6%. At 9B, evidence presence grows by 1.8 percentage points, smaller than the 7.0-point increase in correct-with-evidence answers. These patterns locate the benefit in the quality of evidence-associated answers as well as access to evidence. They describe aggregate output behavior, rather than paired corrections of the same questions.

4.6 Transfer to Unseen Datasets (RQ5)

We evaluate source-trained policies zero-shot on MuSiQue, PopQA, and TriviaQA, averaging scores across the three targets (Table 6; Appendix B.10). Each target uses its own fixed graph, with no source-graph edits and no further training. This separates transfer of the learned policy from direct reuse of repaired source evidence.

Obs. 10: Policy gains can transfer without carrying over the edited source graph. The largest positive transfer differences occur after 2WikiMultiHopQA training at 4B and HotpotQA training at 9B. Their mean target F1 improves from 44.9 to 51.3 (+6.4) and from 50.4 to 58.4 (+8.0), respectively, with higher mean target EM in both settings. Because the target graphs remain fixed, these gains reflect policies that retain useful behavior in new retrieval environments, rather than direct access to source-specific edits.

Obs. 11: Transfer is conditional on the training source and model scale. The reported source-averaged F1 differences are -0.6 at 4B and $+1.3$ at 9B, and NQ-trained policies decline at both scales. In-domain improvement therefore does not imply uniformly better zero-shot transfer. SIG-R1 can learn transferable retrieval behavior, but its most consistent advantage remains in the environments where policy learning and evidence-constrained graph evolution are coupled.

4.7 Case Studies: What Graph Evolution Changes

We now show *what* graph evolution changes for the agent through three auditable trajectories—two successful repairs and one boundary failure—each comparing the same policy and retrieval budget on the fixed versus the evolved graph. Recorded final answers are color-coded, each box highlights the decisive difference, and full retrieval tables are deferred to Appendix C.

Case 1 (HotpotQA, success): a missing EP→track bridge.
Question: *Which track is included in Alexandra Burke's Chridtmas Gift EP?* **Gold answer:** *Silent Night*.
Graph-R1 (fixed graph): three successive queries (`tracklist`, `tracks`, `song list`) return only the EP description—the initial graph contains *no EP→track hyperedge*. After exhausting the retrieval budget, the agent guesses **Christmas Gift, All I Want for Christmas Is You, Merry Christmas** (EM/F1 = 0/0).
SIG-R1 (evolved graph): an evolution round commits a `bridge_chain` hyperedge "Christmas Gift is the first EP … The EP contains a version **'Silent Night'**, …", which ranks #3 in the hyperedge index and enters the first-round fused evidence; the agent answers **Silent Night** (EM/F1 = 1/1).
Key difference: the repair creates a new structural path (Alexandra Burke → Christmas Gift → track), not a better query policy: the fixed-graph agent already queried the right EP, and more retrieval rounds cannot surface a nonexistent edge.

Case 2 (2WikiMultiHopQA, success): two documents composed into one unit.
Question: *What is the date of birth of the composer of film Destination Miami: Objective Murder?* **Gold answer:** *24 January 1933*.
Graph-R1 (fixed graph): three queries alternate among the film, its director, and its composer without ever forming the *film → composer → birth date* chain; the agent answers **Not determinable from available search results** (EM/F1 = 0/0).
SIG-R1 (evolved graph): a `connected` hyperedge concatenates the film paragraph with Nico Fidenco's biography ("born **24 January 1933** in Rome"), so two queries (`film composer`; `Nico Fidenco date of birth`) complete both hops; the agent answers **24 January 1933** within three turns (EM/F1 = 1/1).
Key difference: one hyperedge carries both hops *and* the verbatim answer span, so the policy need not maintain the intermediate entity across turns.

Case 3 (NQ, failure): reachable evidence, unresolved conflicts.
Question: *when does season 2 of seven deadly sins* **Gold answer:** *February 17, 2017*.
Both systems answer **January 13, 2018** (EM/F1 = 0/0). Evolution did commit a `bridge_chain` hyperedge containing "the *Signs of Holy War* arc, labeled as 'Season 2', … was released on **February 17, 2017**", yet the full-agent run still records no evidence hit: the generic entity Season 2 retrieves conflicting dates from unrelated shows, and the model never aligns *when does* with the answer-bearing sentence.
Key difference vs. Cases 1–2: graph evolution repairs *evidence reachability*, not *answer selection* under conflicting candidates; entity disambiguation and evidence-consistent ranking remain open (Appendix C).

5 Conclusion

SIG-R1 couples retrieval-policy learning with evidence-constrained evolution of the retrieval environment, using process-aware rewards to supervise search decisions and verified failures to trigger corpus-grounded repairs of the hypergraph and its index. More broadly, SIG-R1 provides a verifiable framework for recursive self-improvement in which environment defects become explicit training signals rather than fixed limitations. We hope this framework offers a foundation for studying jointly adaptive policies and knowledge environments. Future work will extend environment evolution to richer knowledge structures and improve generalization to unseen graphs and domains.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing of early drafts (grammar, phrasing, and clarity). We have not used generative AI tools to generate research ideas, to design the method, to produce or analyze experimental results, or to author the experimental code that produced the reported numbers. The core technical contributions, experimental design, analysis, and conclusions were developed entirely by the authors, and we take responsibility for the final content, including all text, scientific claims, and accompanying research artifacts.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Dongzhe Fan, Chuanhao Ji, Zimu Wang, Tong Chen, and Qiaoyu Tan. Graphrag-router: Learning cost-efficient routing over graphrags and llms with reinforcement learning. arXiv preprint arXiv:2604.16401, 2026. URL https://arxiv.org/abs/2604.16401.

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*, 2025a.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *EMNLP (Findings)*, pp. 10746–10761, 2025b.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

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.

Manzong Huang, Chenyang Bu, Yi He, Xingrui Zhuo, and Xindong Wu. Relink: Constructing query-driven evidence graph on-the-fly for graphrag. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31202–31210, 2026a.

Yuqi Huang, Ning Liao, Kai Yang, Anning Hu, Shengchao Hu, Xiaoxing Wang, and Junchi Yan. Help: Hypernode expansion and logical path-guided evidence localization for accurate and efficient graphrag. arXiv preprint arXiv:2602.20926, 2026b. URL https://arxiv.org/abs/2602.20926.

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

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

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Jing Li, Zhijie Sun, Zhicheng Zhou, Suming Qiu, Junjie Huang, Haijia Sun, and Linyuan Qiu. Agentic-kgr: co-evolutionary knowledge graph construction through multi-agent reinforcement learning. *arXiv preprint arXiv:2510.09156*, 2025.

Jiashi Lin, Changhong Jiang, Xiangru Lin, Ruifei Zhang, Xinyi Zhu, Jiyao Liu, Cheng Tang, Ye Du, Shujian Gao, Junzhi Ning, et al. Evograph-r1: Self-evolving multimodal knowledge hypergraphs for agentic retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 756–765, 2026.

Haoran Luo, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, Luu Anh Tuan, et al. Graph-r1: Towards agentic graphrag framework via end-to-end reinforcement learning. *arXiv preprint arXiv:2507.21892*, 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 *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Jinyoung Park, Sanghyeok Lee, Omar Zia Khan, Hyunwoo J Kim, and Joo-Kyung Kim. Hypergraphpro: Progress-aware reinforcement learning for structure-guided hypergraph rag. *arXiv preprint arXiv:2601.17755*, 2026.

Changle Qu, Sunhao Dai, Hengyi Cai, Jun Xu, Shuaiqiang Wang, and Dawei Yin. Matchtir: Fine-grained supervision for tool-integrated reasoning via bipartite matching. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11953–11968, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

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

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

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.

Bo Wang, Heyan Huang, Yaolin Li, Wei Tang, Yuan Zhang, Wenbo Li, Mingze Gao, Ge Shi, and Chong Feng. Pathrouter: Aligning rewards with retrieval quality in agentic graph retrieval-augmented generation. *arXiv preprint arXiv:2606.16409*, 2026a.

Juntong Wang, Haoyue Zhao, Guanghui Pan, Xiyuan Wang, Yanbo Wang, Qiyan Deng, and Muhan Zhang. Sage: A self-evolving agentic graph-memory engine for structure-aware associative memory. arXiv preprint arXiv:2605.12061, 2026b. URL https://arxiv.org/abs/2605.12061.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

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

Tianyang Xu, Haojie Zheng, Chengze Li, Haoxiang Chen, Yixin Liu, Ruoxi Chen, and Lichao Sun. Noderag: Structuring graph-based rag with heterogeneous nodes. *arXiv preprint arXiv:2504.11544*, 2025.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang, Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li, Yuzhi Zhang, Qingyun Sun, et al. Graphrag-r1: Graph retrieval-augmented generation with process-constrained reinforcement learning. In *Proceedings of the ACM Web Conference 2026*, pp. 1398–1409, 2026.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (stop): Recursively self-improving code generation. *arXiv preprint arXiv:2310.02304*, 2023.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations*, volume 2026, pp. 147053–147075, 2026.

A Supplementary Analyses

Scale, seed, retrieval-turn, transfer, and robustness analyses are collected here to keep the main narrative focused on the core causal comparisons.

A.1 Evidence-Access Motivation

Figure 6 provides a compact conceptual view of the evidence-access gap addressed by SIG-R1. It distinguishes policy improvement within a fixed graph from the recovery of corpus-supported evidence that is not yet represented or exposed through the retrieval index.

B Implementation Details

Policies are Qwen3.5-2B/4B/9B and Llama-3.1-8B. The default RL runs use 40 policy updates, a global batch of 48 questions, five sampled trajectories per question, and learning rate 5×10^{-7}. Main-table numbers use greedy decoding; component ablations report means over three seeds (1, 20260811, and 42). Table 4 additionally reports sample standard deviations. The evolution probe contains 512 training questions and runs every five updates with persistence m=2. Injected observations are excluded from the policy loss; step rewards are attached to the final policy token that issued each query. Edits are generated only from training-split gold evidence, stored in a per-run graph copy, indexed incrementally, and atomically reloaded. The initial graph remains read-only and no development or test artifact enters the edit pipeline.

B.1 Retrieval and Policy Optimization Details

BAAI General Embedding (BGE) encoders (Xiao et al., 2024) map corpus text and query text to vectors via ϕ and ψ, respectively. The inner product $\langle \phi(t), \psi(x) \rangle$ scores evidence text t for query x. The operator $\mathrm{Retr}(x; G) = (t_1, \ldots, t_{10})$ combines entity and hyperedge retrieval, graph traversal, and raw corpus chunks to return ten texts. The interaction loop uses a configured retrieval budget T. Token-level F1 is the harmonic mean of answer-token precision and recall. GRPO standardizes rewards within each question's rollout group without a learned value estimator (critic). PPO clips action-probability ratios in its surrogate objective, and retrieved observations are excluded from the loss.

B.2 Persistent Failure Attribution and Routing Rules

Let $\mathcal{D}_{\text{train}}$ be the training questions and $P \subset \mathcal{D}_{\text{train}}$ the fixed probe with $|P| = 512$. Every $I = 5$ updates, greedy decoding selects the highest-probability token at each generation step. At evolution round r, after update rI, define $\mathcal{M}_r = \{q \in P : h(\tau_q, \mathcal{A}_q) = 0\}$, where τ_q is the probe trajectory and h indicates whether any retrieved observation contains a gold answer alias. The persistent-miss set is $\mathcal{M}_r^{\text{persist}} = \bigcap_{j=0}^{m-1} \mathcal{M}_{r-j}$ for $r \geq m$, and empty otherwise, with $m = 2$.

Attribution uses the training-evidence field `evidence_status` and entity-name hints: question anchors, bridge entities connecting intermediate facts, and answer entities. These are name lists, not probabilistic labels. Two attribution passes incorporate the current probe before routing, making a newly persistent miss eligible within that round. For a persistent miss, the first matching rule below assigns its edit-routing label; `missing` abbreviates `missing_answer_evidence`:

$$\text{label}(q) = \begin{cases} \text{defect/confirmed}, & \texttt{evidence_status}(q) = \texttt{missing} \\ \text{defect/suspected}, & \text{no anchor names or anchor queried} \\ \text{policy/query form.}, & \text{otherwise.} \end{cases} \tag{7}$$

Grounding labels describe failure to use retrieved evidence, path labels describe failure to connect reachable evidence, and graph-defect labels describe missing or unretrievable evidence. Grounding and path remain separate diagnostics. With $\mathcal{L}_{\text{def}} = \{\text{defect/confirmed}, \text{defect/suspected}\}$, the candidate questions are

$$\mathcal{C}_r = \{q \in \mathcal{M}_r^{\text{persist}} : \text{label}(q) \in \mathcal{L}_{\text{def}},\ q \notin Q_{\text{fix}}\}, \tag{8}$$

where Q_{fix} contains questions repaired in earlier rounds. Only these graph-defect candidates can modify the environment.

B.3 Edit Construction and Admission Details

H1 deterministically constructs at most three hyperedges per question from training-split gold evidence. Their content concatenates answer-bearing sentences, anchor/bridge titles, and sentences mentioning relevant entities together; each hyperedge links the participating entities. Content-derived identifiers prevent duplicate insertion. H2 appends a verbatim answer-bearing sentence s^* to $\mathrm{rt}(v)$ only when absent, producing $\mathrm{rt}'(v)$. The sentence must mention v itself.

For each edit e, the Boolean gates $g_1(e)$, $g_2(e)$, and $g_3(e)$ must all pass before commit (writing the edit to the graph); rejections are logged. The quote gate requires the content to equal its declared quotes, each verbatim in its declared context. The ownership gate requires the quoted fact to mention its associated entity. In Eq. 6, t_e is the new hyperedge content for H1 or updated entity text $\mathrm{rt}'(v)$ for H2. $\mathrm{hit}(t_e)$ is true when the text contains a normalized answer alias or its key span: the ± 5-word window around the earliest answer alias, or the full sentence when shorter than 12 words. The score comparison uses the current index's tenth-ranked score for q, not the output of the full hybrid pipeline after insertion. Accepted edits update a per-run graph copy and its indexes, followed by an atomic reload that switches both together (Appendix B.7).

B.4 Seed-Level Results

Table 4 gives the individual evaluation results for the three Qwen3.5-4B seeds.

B.5 Complete Training Procedure

Algorithm 1 gives the training procedure. Here $\mathrm{GenEdits}(\mathcal{C}_r)$ produces the H1/H2 candidate edits, ΔG is the set that passes all three gates, and $\mathrm{questions}(\Delta G)$ returns their originating questions. $\mathcal{H}$ stores previous miss sets; M^{loss} masks policy tokens (1) versus retrieved tokens (0). FAISS is the library used for vector similarity search; `POST /reload` requests a joint graph/index refresh.

Table 4: SIG-R1 test results across three training seeds on Qwen3.5-4B. Scores are percentages; means carry green upper and red lower annotations for $+1$ and -1 standard deviation across seeds.

	Seed 1		Seed 20260811		Seed 42		Mean ± Std.	
Dataset	F1	EM	F1	EM	F1	EM	F1	EM
2WikiMultiHopQA	68.95	62.50	73.36	64.84	69.14	60.94	$\mathbf{70.48}^{+2.49}_{-2.49}$	$\mathbf{62.76}^{+1.97}_{-1.97}$
NQ	52.12	39.84	49.88	36.72	51.14	37.50	$\mathbf{51.05}^{+1.12}_{-1.12}$	$\mathbf{38.02}^{+1.63}_{-1.63}$
HotpotQA	65.63	57.81	64.37	56.25	63.33	56.25	$\mathbf{64.44}^{+1.15}_{-1.15}$	$\mathbf{56.77}^{+0.90}_{-0.90}$

Algorithm 1 SIG-R1: process-aware RL with gated graph self-evolution

Require: graph copy $G \leftarrow G_0$, policy π_θ, retrieval budget T, probe $P \subset \mathcal{D}_{\text{train}}$ ($|P|$=512), interval I=5, persistence m=2, steps N=40

miss history $\mathcal{H} \leftarrow []$; committed set $Q_{\text{fix}} \leftarrow \emptyset$
for policy update $u = 1$ **to** N **do**
  sample 48 questions; repeat $\times 5$; roll out on $\text{Retr}(\cdot; G)$ under retrieval budget T, then answer
  strip graph sidecars from observations into trajectory metadata
  compute outcome reward R^{out} and step rewards r_k (Eqs. 1 through 5); place r_k on the last policy token of segment k
  form advantages using Eqs. 12–14; PPO-clip update of π_θ under mask M^{loss}
  if $u \bmod I = 0$ **then**
    $r \leftarrow u/I$; run greedy probe rollouts on P; $\mathcal{M}_r \leftarrow \{q \in P : h(\tau_q, \mathcal{A}_q) = 0\}$
    $\mathcal{M}_r^{\text{persist}} \leftarrow \emptyset$ if $r < m$, otherwise $\bigcap_{i=0}^{m-1} \mathcal{M}_{r-i}$; two-pass attribution (Eq. 7)
    $\mathcal{C}_r \leftarrow$ graph-defect candidates (Eq. 8)
    $\Delta G \leftarrow \{e \in \text{GenEdits}(\mathcal{C}_r) : g_1(e) \wedge g_2(e) \wedge g_3(e)\}$ {deterministic, no LLM; Eq. 6}
    if $\Delta G \neq \emptyset$ **then**
      commit ΔG to the stores and graph structure; rebuild FAISS incrementally; $Q_{\text{fix}} \leftarrow Q_{\text{fix}} \cup \text{questions}(\Delta G)$
      `POST /reload` {atomic swap; updates from $rI + 1$ use the evolved graph}
    end if
    append edit log; $\mathcal{H} \leftarrow \mathcal{H} + [\mathcal{M}_r]$
  end if
end for
return π_θ, evolved graph G

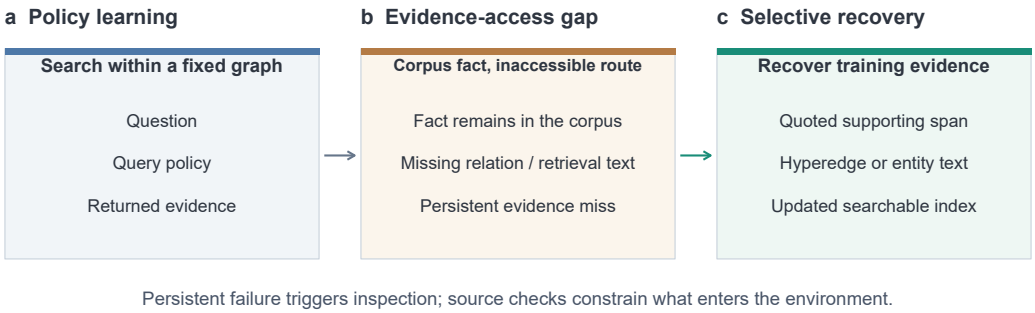


Figure 6: **Evidence-access gap and selective recovery.** (a) Policy learning refines search within a fixed graph. (b) Relevant corpus facts can remain inaccessible when relations or retrieval text are missing. (c) Persistent training misses trigger source-constrained recovery as a hyperedge or entity-text update, followed by index refresh.

B.6 Token Masking and Step-Sliced Advantages

Reward semantics and bounds. Each retrieval segment pairs a `<query>` block with its returned `<knowledge>` block. Its sidecar stores entity identifiers and names; identifiers distinguish similarly named entities. The policy cannot write sidecars, although its queries determine their contents. Missing sidecars yield zero structural reward. In Eq. 3, $\text{match}(E_k, \mathcal{A}_q)$ abbreviates $\exists x \in \text{names}(E_k) : \text{match}(x, \mathcal{A}_q)$, where names returns surface names and the latter predicate tests normalized alias matching; the signal is set to zero for yes/no questions.

All four components lie in $[0, 1]$ and vanish for $E_k = \emptyset$. For nonempty E_k, $\text{gain}_k + \text{conn}_k = 1$, so their weighted contribution is $0.03 + 0.01\,\text{gain}_k$. New entities receive modest additional credit; repeated retrieval can still receive positive reward from the other terms. The definitions imply $0 \leq r_k \leq 0.12$, below the configured cap $R_{\max} = 0.15$. The local-support term requires a retrieved hyperedge and overlap with F_k; it does not validate an entire reasoning path.

Reward placement and observation masking. For trajectory i and token position t, let $M^{\text{loss}}_{i,t}$ be one on policy-generated tokens and zero on injected observation segments and their wrappers. Let $c_i(t) \in \{1, \ldots, K_i\} \cup \{-1\}$ map a token position to its retrieval-step identifier, where K_i is the number of retrieval calls in trajectory i, with $c_i(t) = -1$ for the final answer segment. The scalar reward $r_{i,k}$ is stored at the last policy token of retrieval segment k, which issued the corresponding query, and is zero at other positions in the step-reward tensor. This storage convention is distinct from the advantage assignment: after normalization, the step advantage is applied to the policy tokens belonging to that retrieval segment. Injected text receives neither policy loss nor step advantage.

Comparison groups and normalization. For a fixed question q, denote its sampled trajectory group by $\mathcal{B}_q$ and the subgroup that possesses retrieval-step identifier k by $\mathcal{B}_{q,k}$. Missing steps are excluded from the subgroup; they are not entered as zero-reward observations. For scalar scores x_i and trajectory indices i in a comparison group $\mathcal{B}$, define its mean $\mu_{\mathcal{B}}$, population standard deviation $\sigma_{\mathcal{B}}$, and standardized score $\widetilde{x}_{i;\mathcal{B}}$:

$$\mu_{\mathcal{B}}(x) = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} x_j, \tag{9}$$

$$\sigma_{\mathcal{B}}(x) = \sqrt{\frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \big(x_j - \mu_{\mathcal{B}}(x)\big)^2}, \tag{10}$$

$$\widetilde{x}_{i;\mathcal{B}} = \begin{cases} \dfrac{x_i - \mu_{\mathcal{B}}(x)}{\sigma_{\mathcal{B}}(x) + 10^{-6}}, & |\mathcal{B}| > 1 \text{ and } \sigma_{\mathcal{B}}(x) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The constant 10^{-6} stabilizes the denominator. The standard deviation here is the population standard deviation within the comparison group, rather than the sample standard deviation used to summarize results across training seeds.

Outcome and step channels. Let $s_i = R^{\text{out}}_i$ be the terminal trajectory score and $s^k_i = r_{i,k}$ its score at retrieval step k. Normalize s_i over $\mathcal{B}_q$ and s^k_i over $\mathcal{B}_{q,k}$, denoting the resulting values by $\widetilde{s}_i$ and $\widetilde{s}^k_i$, respectively. With $\lambda = 0.5$ weighting step supervision relative to the terminal score, the token-level advantages (relative reward signals for policy optimization) are

$$A^{\text{out}}_{i,t} = \widetilde{s}_i \, M^{\text{loss}}_{i,t}, \tag{12}$$

$$A^{\text{step}}_{i,t} = \begin{cases} \lambda \, \widetilde{s}^{\,c_i(t)}_i M^{\text{loss}}_{i,t}, & c_i(t) \geq 1, \\ 0, & c_i(t) = -1, \end{cases} \tag{13}$$

$$A_{i,t} = \big(A^{\text{out}}_{i,t} + A^{\text{step}}_{i,t}\big) M^{\text{loss}}_{i,t}. \tag{14}$$

The outcome channel is broadcast over policy-generated tokens. The step channel supplies local credit only to the corresponding retrieval segment, so the final answer segment receives outcome advantage alone. A singleton or zero-variance comparison group contributes zero advantage to its channel. These advantages are used in the PPO-clip policy update. Each update uses five trajectories per question sampled at temperature 1.0 and one PPO-clip epoch. Evolution checks occur every five updates; with persistence $m = 2$, the first eligible commit is at the second check, after update 10.

B.7 Graph Updates and Data Isolation

Commit and atomic reload. Accepted edits are applied to a per-run working copy of the graph, while the offline-built graph G_0 remains read-only. H1 adds hyperedges and H2 appends source-supported text to entity retrieval descriptions; neither operation deletes existing graph content. The stores and graph structure are updated, edited entity rows are re-encoded, and new hyperedge vectors are appended to the FAISS indexes. The training process then issues `POST /reload`. The search service atomically swaps the indexes, stores, and graph under the same lock that serializes search requests. Subsequent training and probe queries use the evolved graph. Injected items carry provenance flags that preserve traceability to their quoted sources.

Training, selection, and evaluation boundaries. Edit generation uses training-split gold evidence, and the 512-question probe is sampled from the training split. Development and test questions, answer annotations, and associated gold-evidence contexts are excluded from the probe and edit-generation inputs. Development data are used for checkpoint selection, while test data are used for evaluation. This restriction concerns the use of benchmark supervision in the editing pipeline; it does not assert that documents in a shared retrieval corpus are disjoint from all evaluation evidence. During zero-shot evaluation, each target dataset retains its static graph and receives no edits from the source training run.

Dataset construction and partitions. We reuse Graph-R1's processed QA data and released partitions (Luo et al., 2025). Graph-R1 draws its benchmarks from FlashRAG and uniformly samples 5,120 training examples and 128 test examples per dataset. Its preprocessing script reads `qa_train.json`, `qa_dev.json`, and `qa_test.json` separately and converts them into the corresponding training, development, and test files. We preserve these partitions, including the development examples, without resampling. Training data supply policy updates and the 512-question evolution probe; the development partition selects checkpoints; the test partition evaluates the selected checkpoints. NQ covers open-domain QA, while 2WikiMultiHopQA and HotpotQA require evidence from multiple documents.

B.8 Outcome-Profile Measurement

Table 3 pools 128 held-out questions from each of NQ, 2WikiMultiHopQA, and HotpotQA, yielding 384 answers per model setting. "Correct" denotes normalized EM ≥ 0.99. "Valid evidence" is the offline checker's indicator computed from the visible `<knowledge>` blocks, including a separate retrieved-segment check for yes/no answers, whose answer words alone are not informative evidence matches. The saved `evidence_hit` field depends on a reward-time graph sidecar that is absent from the final JSON files, so the profile recomputes evidence presence from visible text. Format errors are a separate diagnostic and can overlap the four mutually exclusive outcome cells. Their rates are 6.3%/5.2% for 4B Graph-R1/SIG-R1 and 0.3%/0.5% for 9B.

The percentages describe joint outcomes, not paired transitions for individual questions. Detected evidence does not by itself establish that an answer was derived from that evidence, and these cells do not validate the four heuristic failure-attribution labels. Likewise, the accepted and rejected edit counts in Figure 5 describe gate activity, rather than label accuracy or a measured rate of unsupported content.

B.9 Retrieval-turn Dynamics

Table 5: Retrieval-turn dynamics on Qwen3.5-4B test (pooled over seeds and datasets). A turn is one retrieval call; Share is the fraction of answers in each turn-count group, and EM is accuracy within that group. Δ denotes SIG-R1 minus Graph-R1 in percentage points.

	Share (%)			**EM (%)**		
Turns	Graph-R1	SIG-R1	Δ	Graph-R1	SIG-R1	Δ
0	5.1	5.1	+0.0	0.0	1.7	+1.7
1	14.8	20.9	+6.1	67.8	66.4	-1.4
2	37.7	34.4	-3.3	67.3	71.0	+3.7
3	39.6	36.5	-3.0	32.7	36.6	+3.9
4+	2.8	3.0	+0.3	0.0	0.0	+0.0

B.10 Zero-Shot Out-of-Distribution Transfer (RQ5)

This experiment evaluates checkpoints trained on one source dataset on three unseen target datasets (MuSiQue, PopQA, and TriviaQA). Each target uses its own fixed graph, without importing edits from the source training graph or performing test-time graph updates. The comparison therefore measures cross-dataset policy transfer rather than transfer of the edited graph itself. Table 6 reports the mean test score across the three OOD datasets.

Interpretation. Transfer depends on the source dataset and model scale. The reported overall F1 difference is -0.6 points for Qwen3.5-4B and $+1.3$ points for Qwen3.5-9B. The largest positive differences occur after 2WikiMultiHopQA training at 4B ($+6.4$ points) and HotpotQA training at 9B ($+8.0$ points), whereas NQ-trained policies decrease on the aggregate target score at both scales. Thus, the in-domain gains do not translate into uniform cross-dataset improvements. These results

Table 6: **Cross-dataset policy transfer.** Each model is trained on the source dataset in "Trained on" and evaluated zero-shot on MuSiQue, PopQA, and TriviaQA. F1 and EM are mean scores across these three targets (%); each target graph stays fixed and receives no source-graph edits. Ret. denotes the reported retention ratio, $100 \times \mathrm{F1}_{\mathrm{OOD}}/\mathrm{F1}_{\mathrm{ID}}$, where OOD is target-dataset evaluation and ID is in-domain source-dataset evaluation. A value of 100% indicates equal F1. Δ is SIG-R1 minus Graph-R1 OOD F1 in percentage points; Overall summarizes the three source settings. The comparison tests whether policy improvements generalize across datasets.

Model	Trained on	Graph-R1			SIG-R1			Δ F1
		F1	EM	Ret.	F1	EM	Ret.	
Qwen3.5-4B	NQ	50.7	46.5	112%	46.8	46.7	97%	-3.9
	2Wiki	44.9	44.7	98%	**51.3**	**48.4**	83%	+6.4
	HotpotQA	46.7	42.5	85%	43.7	42.4	84%	-3.0
	Overall	47.3	44.3	96%	46.8	45.3	87%	-0.6
Qwen3.5-9B	NQ	48.8	40.6	105%	44.8	38.8	95%	-4.1
	2Wiki	59.4	51.8	88%	59.4	52.1	85%	-0.0
	HotpotQA	50.4	42.4	88%	**58.4**	**51.0**	83%	+8.0
	Overall	52.9	45.0	94%	**54.2**	**47.3**	88%	+1.3

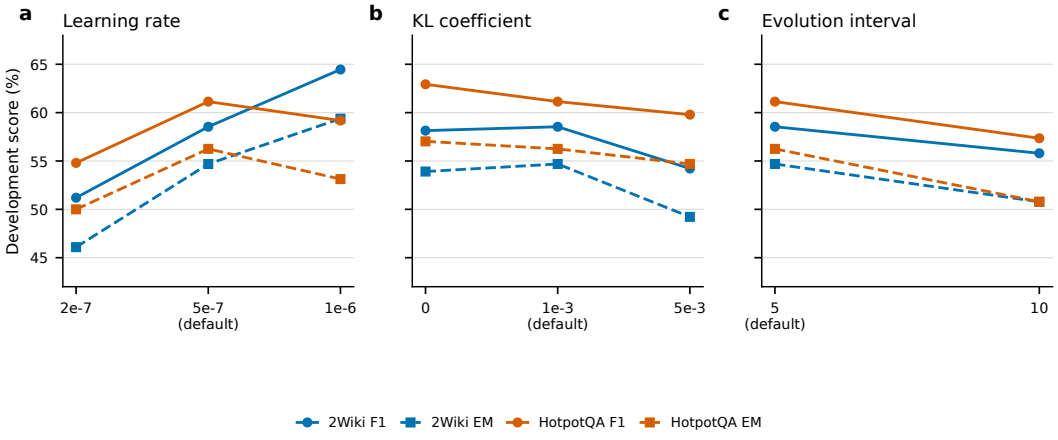


Figure 7: **Hyperparameter sensitivity.** One parameter is varied at a time on Qwen3.5-2B; values are development-set scores.

characterize policy transfer under fixed target environments; they do not isolate the effect of source-graph edits on new questions.

B.11 Robustness to Hyperparameters and Decoding (RQ1)

These analyses test sensitivity to optimization and decoding choices. The KL coefficient weights Kullback–Leibler divergence regularization in the policy objective. Greedy decoding chooses the highest-probability next token; sampled decoding draws tokens from the predicted distribution. Table 7 varies one hyperparameter at a time on Qwen3.5-2B (also visualized in Figure 7), and Table 8 compares greedy decoding with sampled decoding at individual NQ and 2WikiMultiHopQA checkpoints.

C Case Study Details

This appendix expands the three cases of Section 4.7 with the retrieval states that precede each final answer. All quoted strings are transcribed verbatim from the saved run artifacts (prediction files and edit logs). The examples come from the development split and illustrate how the repair mechanism reorganizes corpus facts into queryable hyperedge paths; they are mechanism evidence, not standalone proof of test-time generalization.

Table 7: Single-variable hyperparameter sensitivity on Qwen3.5-2B (development scores, %). Center denotes the reference configuration; Best step is the selected policy-update index.

Hyperparameter	Value	2WikiMultiHopQA (dev)			HotpotQA (dev)		
		F1	EM	Best step	F1	EM	Best step
Learning rate	2×10^{-7}	51.20	46.09	30	54.80	50.00	40
Learning rate	5×10^{-7} (center)	58.54	54.69	30	61.14	56.25	40
Learning rate	1×10^{-6}	**64.46**	**59.38**	20	59.18	53.13	30
KL coefficient	0	58.14	53.91	40	**62.93**	**57.03**	40
KL coefficient	10^{-3} (center)	58.54	54.69	30	61.14	56.25	40
KL coefficient	5×10^{-3}	54.22	49.22	40	59.79	54.69	40
Evolution interval	5 (center)	58.54	54.69	30	61.14	56.25	40
Evolution interval	10	55.80	50.78	40	57.35	50.78	40

Note: Interval 1 is omitted because the implementation requires an evolution boundary to coincide with a validation rollout, while validation is performed every five steps. These are single-seed development results and should not be compared directly with the three-seed 4B test results.

Table 8: Decoding results of SIG-R1 on Qwen3.5-4B for NQ and 2WikiMultiHopQA (%). Greedy scores are from individual checkpoints; sampled scores report means over three decoding runs, with green upper and red lower annotations denoting $+1$ and -1 standard deviation.

	Greedy		**Sampled ×3**	
Dataset	EM	F1	EM	F1
NQ	39.84	52.12	$36.72^{+1.35}_{-1.35}$	$48.63^{+1.62}_{-1.62}$
2Wiki	64.84	73.36	$64.06^{+2.34}_{-2.34}$	$72.55^{+2.50}_{-2.50}$

C.1 Case 1 (HotpotQA): *Silent Night*

Question and outcome. *Which track is included in Alexandra Burke's Chridtmas Gift EP?* Gold answer: *Silent Night*. Both runs use the same policy, greedy decoding, and four retrieval turns; only the graph differs.

Table 9: Case 1: what each retrieval view returns for the shared first query `Alexandra Burke Christmas Gift EP tracklist`.

Retrieval view	**Granularity**	**Returns *Silent Night*?**	**Diagnosis**
Raw chunks (vector)	5230-char chunk	No (invisible)	Answer at char. 2463, past the visible prefix
Entities (fixed graph)	entity text	No	Christmas Gift ranked #1 (0.8416), EP description only
Hyperedges (fixed graph)	hyperedge text	No	EP→track edge missing from top-10
Evolved graph (SIG-R1)	entity + hyperedge	**Yes**	`bridge_chain` hyperedge at rank #3 (0.7860)

Why the fixed graph fails. The failure is structural rather than a query-policy error. The top-1 raw chunk (BGE score 0.7107) does contain the answer sentence, but at character 2463 of a 5230-character chunk; because one tool response must pack ten evidence items under the response-length cap, the agent-visible excerpt ends before the answer. Chunk ranks 2–10 drift to other artists (Dua Lipa, Nick Jonas, Paramore, . . .). Entity retrieval already ranks the correct intermediate entity Christmas Gift first, but its retrieval text contains only the EP description; hyperedge retrieval likewise returns only "Christmas Gift is the first EP . . . ". Notably, the initial graph *does* contain a single-sentence hyperedge mentioning *Silent Night*, yet it never enters this query's top-10, so the fact is present in the corpus but unreachable through the retrieval path. The baseline agent issues three paraphrased queries with near-identical results and finally guesses **Christmas Gift, All I Want for Christmas Is You, Merry Christmas**.

What evolution commits. For this persistent failure, the edit generator constructs edits from the gold-evidence sentence *"The EP contains a version 'Silent Night', which was hugely popular with Burke's fans as she performed this on the final of 'The X Factor' when she was a contestant."* Two

hyperedges pass all three gates and are committed (a third `bridge_fact` candidate is proposed in the edit log but not committed to the final graph):

Table 10: Case 1: committed edits. Both hyperedges share the same six-node incidence; *Silent Night* is an answer span inside the hyperedge text, not a standalone entity node.

Edit	Kind	Content / effect
H1 edge `rel-evonline4-2d25...73045`	`bridge_chain`	EP description + *contains a version "Silent Night"*
H1 edge `rel-evonline4-ba72...817d`	`cooccur`	EP description (short auxiliary edge)
H2 on ALEXANDRA BURKE	entity text	EP fact appended to the anchor's retrieval text
Incidence of both H1 edges: ALEXANDRA BURKE, CHRISTMAS GIFT, THE X FACTOR, JASON DERULO, BAD BOYS, ADAM LAMBERT (context-window co-occurring entities)		

After the edit, ALEXANDRA BURKE rises from outside the entity top-10 to rank #2 (0.7432), and the `bridge_chain` hyperedge enters the hyperedge top-10 at rank #3. In the fused evidence actually shown to the agent, the answer-bearing edge appears as item #5 of round 1; rounds 2–3 return the same edge, and the final answer is **Silent Night** (EM/F1 $= 1/1$, evidence hit). The incidence spans six nodes because the edit attaches all entities co-occurring in the evidence window; only ALEXANDRA BURKE $\rightarrow$ CHRISTMAS GIFT $\rightarrow$ track is necessary for this question, indicating that current edits operate at the context-window rather than minimal-chain granularity.

C.2 CASE 2 (2WIKIMULTIHOPQA): *24 January 1933*

Question and outcome. *What is the date of birth of the composer of film Destination Miami: Objective Murder?* Gold answer: *24 January 1933*. The required chain is DESTINATION MIAMI: OBJECTIVE MURDER $\rightarrow$ NICO FIDENCO $\rightarrow$ *date of birth*, spanning two documents: the answer does not occur in the film's description but in the composer's biography.

Why the fixed graph fails. The baseline's three queries (`film composer`, `composer name`, `Piero Regnoli ... composer`) circle among the film, its director PIERO REGNOLI, and cast member CLAUDIO GORA without establishing that NICO FIDENCO composed the soundtrack, and it never reaches the biography document. It terminates with **Not determinable from available search results** (EM/F1 $= 0/0$).

What evolution commits. Three edits pass the gates:

Table 11: Case 2: committed edits and their roles.

Edit	Kind	Content / effect
`rel-evonline4-3a06...7f58`	`connected`	Film paragraph concatenated with the NICO FIDENCO biography ("born **24 January 1933** in Rome ... film soundtrack composer"); links the film, NICO FIDENCO, DATE OF BIRTH, and context entities PIERO REGNOLI, CLAUDIO GORA
`rel-evonline4-313f...3c27`	`bridge_fact`	Short path carrying only NICO FIDENCO $\rightarrow$ *born 24 January 1933*
`rel-evonline4-e733...05f8`	`cooccur`	First-hop fact: the film co-occurs with NICO FIDENCO
H2 on DESTINATION MIAMI	entity text	Film description + NICO FIDENCO's mention appended to the anchor

How the answer is found. With the evolved graph, the first query `Destination Miami Objective Murder film composer` identifies NICO FIDENCO as soundtrack composer; the second query `Nico Fidenco date of birth` directly hits the biography sentence inside the committed hyperedge. The agent answers **24 January 1933** in three turns (one fewer than the baseline), EM/F1 $= 1/1$ with an evidence hit. The case shows the hyperedge carrying all three roles at once: first-hop entity recognition, second-hop attribute location, and verbatim answer span.

C.3 CASE 3 (NQ): A BOUNDARY FAILURE, *February 17, 2017*

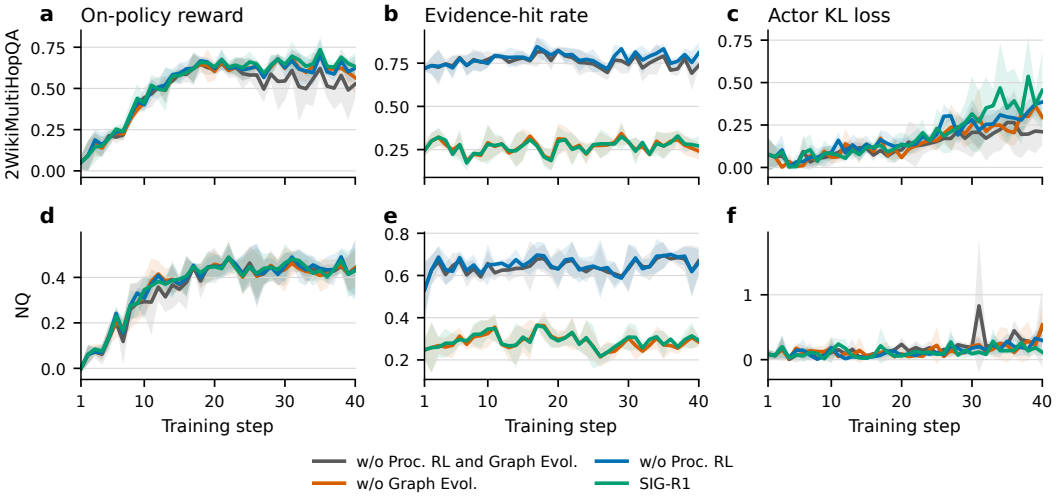
Question and outcome. *when does season 2 of seven deadly sins*. Gold answer: *February 17, 2017*. Both the fixed-graph and evolved-graph runs answer **January 13, 2018** (EM/F1 $= 0/0$, no evidence hit in either run).

What evolution commits. The `bridge_chain` hyperedge `rel-evonline4-6965 ...6847c` quotes: *"The first 'The Seven Deadly Sins' anime series was licensed for English release by Netflix ... The 'Signs of Holy War' arc, labeled as 'Season 2', has also been licensed by Netflix and was released on* ***February 17, 2017****."* It links SEVEN DEADLY SINS, KNIGHTS OF SIDONIA, HOBBY JAPAN, and SEASON 2; a `bridge_fact` edge carries the short SEASON 2 → date path, and H2 appends the licensing sentence to the anchor SEVEN DEADLY SINS.

Why the answer is still wrong. Stage-1 diagnostics confirm that the answer-bearing path now exists, yet the complete agent run still records `evidence_hit=false`, so the repair cannot be credited with retrieval success. Four obstacles remain: (1) SEASON 2 is a high-reuse generic entity that pulls in dates of unrelated shows; (2) the search results mix the Japanese broadcast date, the Netflix release date, and other series' dates (*January 13, 2018*, *April 27, 2018*, *May 18, 2018*); (3) the model never aligns *when does* with the *Signs of Holy War* sentence; (4) an existing answer-bearing sentence therefore does not imply a correct final answer. This case delimits the mechanism: SIG-R1 repairs *evidence reachability*, while conflicting-evidence discrimination and answer extraction—for example through entity disambiguation, temporal-granularity distinction, and evidence-consistent candidate ranking—remain future work.

D TRAINING DIAGNOSTICS

Figure 8 reports reward on trajectories generated by the current policy (on-policy rollouts), evidence-hit rate, and actor KL loss across the four ablation variants. The actor is the trainable policy; its KL loss measures divergence from the reference policy. Evidence-hit rate is the fraction of rollouts with a detected answer-evidence match.



Mean across three seeds; shaded bands denote ±1 s.d. These are on-policy rollout diagnostics, not held-out test metrics.

Figure 8: **Training diagnostics across ablation arms.** On-policy rollout reward, evidence-hit rate, and actor KL loss for the four ablation arms on 2WikiMultiHopQA and NQ. Curves are means across three seeds with one-standard-deviation bands. These quantities diagnose optimization behavior and are distinct from held-out QA performance.

Figure 9 complements these ablation diagnostics with the complete on-policy rollout-F1 trajectories across model scales and datasets. The figure makes the training progression visible without substituting for the held-out results in the main experiments.

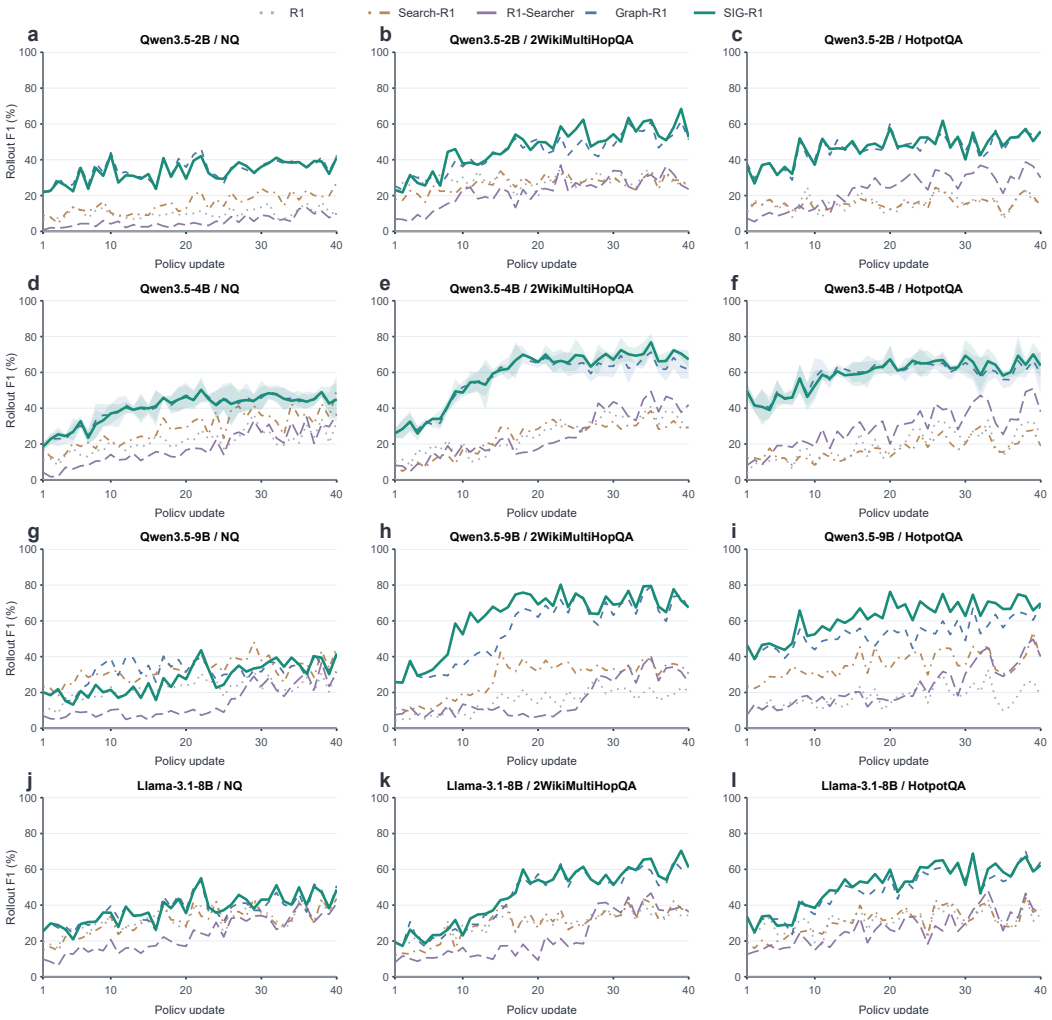


Figure 9: **Training coverage across model scales and datasets.** Each panel reports on-policy rollout F1 over all 40 policy updates for the five compared methods. For the Qwen3.5-4B Graph-R1 and SIG-R1 runs, solid curves are three-seed means and shaded regions denote one sample standard deviation; the remaining curves are individual training trajectories.

E Detailed Training Workflow

Figure 10 separates policy updates from periodic evolution rounds for failure attribution, edit verification, and graph/index refresh.

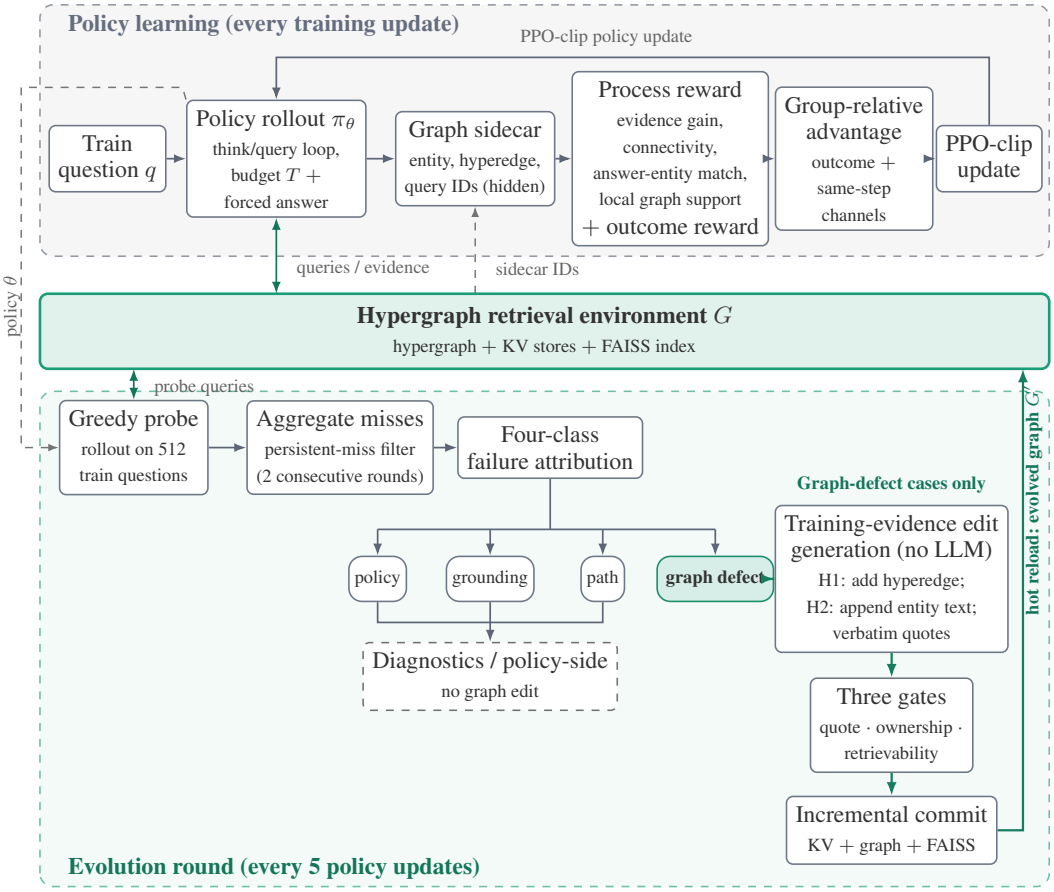


Figure 10: **SIG-R1 overview.** Within a trajectory, SIG-R1 assigns a process reward to each retrieval segment using query and graph-hit metadata. Every five policy updates, an evolution round aggregates persistent probe failures and routes graph-defect candidates to evidence-constrained editing. The quote, ownership, and retrievability gates precede the joint graph/index update. KV denotes key–value storage and G' the evolved working graph.