Graph-R1: Towards Agentic GraphRAG Framework via End-to-end Reinforcement Learning

Haoran Luo [1 2] **Haihong E** [1] **Guanting Chen** [1] **Qika Lin** [3] **Yikai Guo** [4] **Fangzhi Xu** [5] **Zemin Kuang** [6] **Meina Song** [1] **Xiaobao Wu** [7] **Yifan Zhu** [1] **Luu Anh Tuan** [2 8]

Abstract

Retrieval-Augmented Generation (RAG) mitigates hallucination in LLMs by incorporating external knowledge, but relies on chunk-based retrieval that lacks structural semantics. GraphRAG methods improve RAG by modeling knowledge as entity-relation graphs, but still face challenges in high construction cost, fixed one-time retrieval, and reliance on long-context reasoning and prompt design. To address these challenges, we propose **Graph-R1**, the first agentic GraphRAG framework via end-to-end reinforcement learning (RL). It introduces lightweight knowledge hypergraph construction, models retrieval as a multi-turn agent-environment interaction, and optimizes the agent process via an end-to-end reward mechanism. Experiments on standard RAG datasets show that Graph-R1 outperforms traditional GraphRAG and RL-enhanced RAG methods in reasoning accuracy, retrieval efficiency, and generation quality. Our software and data are publicly available at https://github.com/LHRLAB/Graph-R1.

1. Introduction

Large Language Models (LLMs) (Zhao et al., 2025a) have achieved widespread success in NLP tasks. However, when applied to knowledge-intensive or proprietary knowledge-dependent applications, they still suffer from the hallucination problem (Zhang et al., 2025), generating inaccurate content. To improve credibility and factual consistency, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) introduces external knowledge sources as references, alleviating the knowledge bottleneck of pure language modeling. Nevertheless, existing RAG methods mostly rely on chunk-based text blocks (Gao et al., 2024), which makes it difficult to capture complex knowledge structures among entities. To address this, GraphRAG methods (Edge et al., 2025; Guo et al., 2025b; Luo et al., 2025a) represent knowledge as entity-relation graphs, enhancing retrieval efficiency.

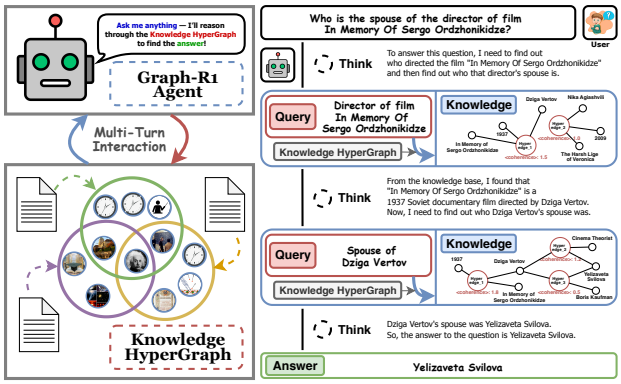


Figure 1. An illustration of Graph-R1. Graph-R1 formulates GraphRAG as a multi-turn agentic retrieval process over a knowledge hypergraph, optimized via end-to-end reinforcement learning.

Generally, GraphRAG methods consist of three processes: *knowledge graph construction*, *graph retrieval*, and *answer generation*. First, knowledge graphs are typically constructed by LLMs to extract entities and relations from text, forming a graph structure (Xu et al., 2024). Second, the retrieval process queries relevant subgraphs or paths through subgraph retrieval or path pruning strategies (Chen et al., 2026; Gutiérrez et al., 2025). Finally, the generation process prompts LLMs to generate answers based on the retrieved graph-based knowledge (Xiao et al., 2025).

However, current GraphRAG methods still face three key challenges: **(i) High cost and semantic loss in *knowledge construction* process.** Compared to standard RAG, GraphRAG methods convert natural language knowledge into graph structures using LLMs, which results in high cost and often causes semantic loss relative to the original content (Luo et al., 2024; 2025a). **(ii) Fixed retrieval process**

[1]Beijing University of Posts and Telecommunications [2]Nanyang Technological University [3]National University of Singapore [4]Beijing Institute of Computer Technology and Application [5]Xi'an Jiaotong University [6]Beijing Anzhen Hospital, Capital Medical University [7]Shanghai Jiao Tong University [8]VinUniversity. Correspondence to: Haihong E <ehaihong@bupt.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

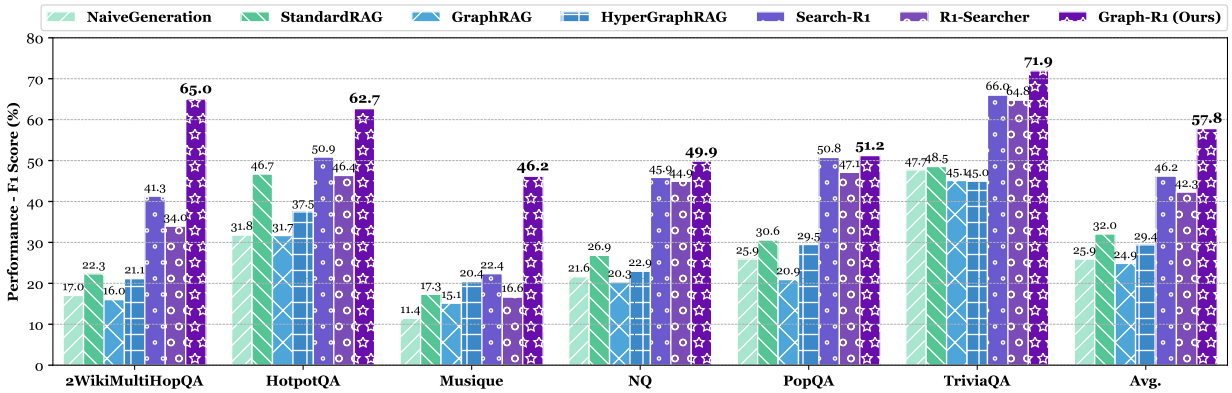


Figure 2. Comparison of F1 scores across RAG benchmarks. Using a graph as the knowledge environment enables RL to achieve a higher performance ceiling compared to chunk-based knowledge. Graph-structured knowledge boosts accuracy by richer representation.

with only one-time interaction in *graph retrieval* process. Although existing GraphRAG methods design various retrieval strategies to improve efficiency, most of them aim to gather sufficient knowledge in a single fixed retrieval (Chen et al., 2026), which limits performance in complex queries. **(iii) Dependence on large LLMs for long-context analysis and prompt quality in *answer generation* process.** Generation based on retrieved graph-structured knowledge often requires strong long-context reasoning ability, making the output quality highly dependent on the LLM's parameter size and prompt design (Guo et al., 2025b).

To address these challenges, we propose **`Graph-R1`**, as illustrated in Figure 1, the first agentic GraphRAG framework enhanced by end-to-end reinforcement learning (RL), inspired by DeepSeek-R1 (Guo et al., 2025a). First, we propose a lightweight knowledge hypergraph construction method to establish a standard agent environment for the query action space. Moreover, we model the retrieval process as a multi-turn agentic interaction process, enabling LLMs to repeatedly perform the reasoning loop of "think-retrieve-rethink-generate" within the knowledge hypergraph environment. Furthermore, we design an end-to-end reward mechanism that integrates generation quality, retrieval relevance, and structural reliability of graph paths into a unified optimization objective. Using RL, the agent learns a generalizable graph reasoning strategy and achieves tighter alignment between structured knowledge and language.

We perform experiments on various standard RAG datasets (Jin et al., 2025b). Experimental results demonstrate that Graph-R1 outperforms traditional GraphRAG methods and RAG combined with RL methods (Jin et al., 2025a; Song et al., 2025) in reasoning accuracy, retrieval efficiency, and generation quality. As shown in Figure 2, the end-to-end RL strategy guides the agent through multiple turns of interaction and goal-driven exploration in the graph, effectively bridging the gap between knowledge representation and language generation. This work lays a foundation for building the next generation of knowledge-driven and strategy-optimized agent-based generation systems.

2. Related Work

RAG and GraphRAG. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) improves LLM factuality by retrieving external knowledge, but struggles with data silos and limited structural reasoning. GraphRAG (Edge et al., 2025) mitigates these issues by incorporating graph-structured knowledge, inspiring enterprise systems (Wu et al., 2025; Liang et al., 2025; Wang et al., 2025a) and efficient variants like LightRAG (Guo et al., 2025b). Recent works further enhance representation power by diverse graph structures (Luo et al., 2025a; Feng et al., 2026a; Lin et al., 2025; Wang et al., 2025b; Xu et al., 2025; Zhuang et al., 2025), while retrieval optimization explores path-based exploration and pruning techniques (Chen et al., 2026; Gutiérrez et al., 2025; Liu et al., 2025; Wang & Han, 2025; Zhao et al., 2025b; Luo et al., 2025c; 2026; Lee et al., 2025; Dong et al., 2025; Yang et al., 2025b). In this work, we propose Graph-R1, the first agentic GraphRAG framework that learns a multi-turn retrieval policy via end-to-end RL.

Reinforcement Learning for LLMs. Reinforcement learning (RL) is increasingly adopted to enhance LLM reasoning (Wu, 2025; Luo et al., 2025b), as demonstrated by OpenAI's o1/o3/o4 (OpenAI et al., 2024b). DeepSeek-R1 (Guo et al., 2025a) achieves comparable capabilities and introduces the Group Relative Policy Optimization (GRPO) (Shao et al., 2024) for scalable end-to-end training, which has been extended to various downstream tasks (Ma et al., 2025; Feng et al., 2026b; Jiang et al., 2026). RL-enhanced agents have also shown strong performance in multi-turn interaction with different environments (Luo et al., 2025b; Feng et al., 2025; Jin et al., 2025a; Song et al., 2025; Liu et al., 2026), highlighting RL's potential in advancing agentic GraphRAG framework (Gao et al., 2025).

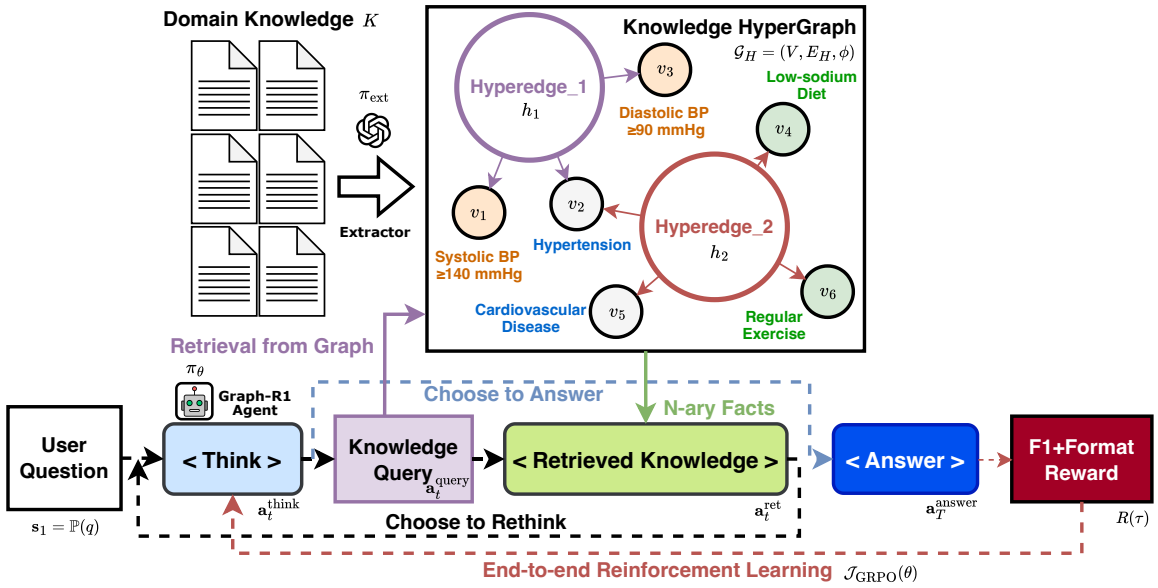


Figure 3. Overview of the Graph-R1 framework: an RL-enhanced reasoning trajectory over knowledge hypergraph, where the agent iteratively decides to think, query from graph environment, retrieve n-ary facts as candidate knowledge, and answer the user question.

3. Preliminaries

We formalize the GraphRAG pipeline into three stages:

(a) Knowledge Graph Construction. Given a knowledge collection $K = \{d_1, d_2, \ldots, d_N\}$, the goal is to extract facts f_d from each semantic unit $d \in K$ into a unified graph $\mathcal{G}_K$:

$$\mathcal{G}_K \sim \sum_{d \in K} \pi_{\text{ext}}(f_d \mid d), \tag{1}$$

where π_{ext} denotes an LLM-based extractor that parses each d into a set of relation-entity pairs $f_d = \{(r_i, \mathcal{V}_{r_i})\}$, with r_i as the relation and $\mathcal{V}_{r_i} = \{v_1, \ldots, v_n\}$ the entities.

(b) Graph Retrieval. Graph retrieval is formulated as a two-step process over $\mathcal{G}_K$: (1) retrieving candidate reasoning paths and (2) pruning irrelevant ones. Conditioned on a query q, the model first retrieves a candidate set $\mathcal{X}_q = \{x_1, \ldots, x_m\}$ and then selects a relevant subset $\mathcal{Z}_q \subseteq \mathcal{X}_q$. The overall objective is to maximize the joint likelihood:

$$\max_\theta \ \mathbb{E}_{\mathcal{Z}_q \sim P(\mathcal{Z}_q \mid q, \mathcal{G}_K)} \left[\prod_{t=1}^{T_x} P_\theta(x_t \mid x_{<t}, q, \mathcal{G}_K) \cdot \prod_{t=1}^{T_z} P_\theta(z_t \mid z_{<t}, \mathcal{X}_q, q) \right], \tag{2}$$

where T_x & T_z are numbers of retrieved & selected paths.

(c) Answer Generation. Given a query q and selected paths $\mathcal{Z}_q$, answer generation produces a natural language answer y grounded in graph-based evidence, formulated as:

$$P(y \mid q, \mathcal{G}_K) = \sum_{\mathcal{Z}_q \subseteq \mathcal{X}_q} P(y \mid q, \mathcal{Z}_q) \cdot P(\mathcal{Z}_q \mid q, \mathcal{G}_K), \tag{3}$$

where $P(y \mid q, \mathcal{Z}_q)$ is generation likelihood and $P(\mathcal{Z}_q \mid q, \mathcal{G}_K)$ is retrieval-pruning distribution.

4. Methodology: Graph-R1

In this section, as illustrated in Figure 3, we introduce Graph-R1, including agent initialization, multi-turn graph interaction, and outcome-directed end-to-end RL optimization.

4.1. Agent Initialization

Graph-R1 adopts an LLM-driven agent, initialized with a knowledge hypergraph environment $\mathcal{G}_H$, the action space $\mathcal{A}$, the state space $\mathcal{S}$, and the answer target y_q, given q.

Graph Environment $\mathcal{G}_H$. We propose a lightweight method for constructing a knowledge hypergraph $\mathcal{G}_H$ from given domain knowledge $K = \{d_1, d_2, \ldots, d_N\}$. For each chunk unit $d \in K$, an LLM-based extractor π_{ext} identifies m n-ary relational facts, where each comprises a semantic segment h_i and a set of participating entities $\mathcal{V}_{h_i} = \{v_1, \ldots, v_n\}$. A shared encoder $\phi(\cdot)$ is then used to generate semantic embeddings for both entities & relations:

$$\begin{aligned} &\mathcal{G}_H = (V, E_H, \phi), \ \text{where} \ \pi_{\text{ext}}(d) \to \{(h_i, \mathcal{V}_{h_i})\}_{i=1}^m, \\ &\text{and} \ \phi(v) = \text{Enc}(v), \ \phi(h_i) = \text{Enc}(h_i), \end{aligned} \tag{4}$$

where each h_i defines a hyperedge $h_i \in E_H$ connecting its associated entities $\mathcal{V}_{h_i}$ as $v \in V$. The resulting hypergraph $\mathcal{G}_H$ encodes n-ary relations with rich semantic grounding.

The Agent Action Space $\mathcal{A}$. In Graph-R1, each agent action $\mathbf{a}_t \in \mathcal{A}$ comprises four sub-actions: *Thinking* $\mathbf{a}_t^{\text{think}}$, which decides whether to continue or terminate reasoning; *Query Generation* $\mathbf{a}_t^{\text{query}}$, which formulates a retrieval query; *Graph Retrieval* $\mathbf{a}_t^{\text{ret}}$, which extracts relevant knowledge from the hypergraph; and *Answering* $\mathbf{a}_t^{\text{ans}}$, which produces a final response if reasoning ends.

Table 1. Template for Graph-R1. question will be replaced with the specific user query. Note that the knowledge retrieved is placed within <knowledge>...</knowledge> after <query>...</query>.

You are a helpful assistant. Answer the given question. You can query from knowledge base provided to you to answer the question. You can query knowledge as many times as you want. You must first conduct reasoning inside <think>...</think>. If you need to query knowledge, you can set a query statement between <query>...</query> to query from knowledge base after <think>...</think>. When you have the final answer, you can output the answer inside <answer>...</answer>. Question: question. Assistant:

The agent action $\mathbf{a}_t$ has two compositional forms, and the joint action log-likelihood $\log \pi(\mathbf{a}_t \mid \mathbf{s}_t)$ is defined as:

$$\begin{cases} \log \mathcal{G}_H(\mathbf{a}_t^{\text{ret}} \mid \mathbf{s}_t, \mathbf{a}_t^{\text{think}}, \mathbf{a}_t^{\text{query}}) + \log \pi(\mathbf{a}_t^{\text{query}} \mid \mathbf{s}_t, \mathbf{a}_t^{\text{think}}) + \\ \log \pi(\mathbf{a}_t^{\text{think}} \mid \mathbf{s}_t), & \text{if } \mathbf{a}_t^{\text{think}} \rightarrow \text{continue}, \\ \log \pi(\mathbf{a}_t^{\text{ans}} \mid \mathbf{s}_t, \mathbf{a}_t^{\text{think}}) + \log \pi(\mathbf{a}_t^{\text{think}} \mid \mathbf{s}_t), & \text{if } \mathbf{a}_t^{\text{think}} \rightarrow \text{terminate}, \end{cases} \tag{5}$$

where, at each step, the agent first performs *Thinking*, and then chooses between continuing reasoning (*Query Generation* and *Graph Retrieval*) or terminating via *Answering*.

The Agent State Space $\mathcal{S}$ and Target y_q. At each step t, the state $\mathbf{s}_t \in \mathcal{S}$ is defined as $\mathbf{s}_t = (\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{a}_{t-1})$, with $\mathbf{s}_1$ initialized from the input query q. Once a termination action $\mathbf{a}_T$ is issued, the agent reaches final state $\mathbf{s}_T$, where T is the total number of reasoning steps, and an answer $y_q \sim \mathbf{a}_T^{\text{ans}}$ is produced to address q.

Proposition 1. *Graph-structured knowledge boosts agent accuracy by richer representation.*

Proof. We provide quantitative experimental results in Section 5.2 and qualitative proofs in Appendix B.1. □

4.2. Reasoning via Multi-turn Graph Interaction

We model reasoning as a multi-turn interaction between an agent π_θ and a hypergraph $\mathcal{G}_H$. We first define the step-wise policy $\pi_\theta(\cdot \mid \mathbf{s}_t)$ prompted by Table 1, then describe how to retrieve knowledge $\mathcal{G}_H(\mathbf{a}_t^{\text{ret}} \mid \cdot, \mathbf{a}_t^{\text{query}})$ based on $\mathbf{a}_t^{\text{query}}$, and finally present the objective to optimize $P(y_q \mid \cdot)$.

Modeling the Step-wise Reasoning Policy. At each reasoning step t, the LLM governs the agent's behavior by generating a structured output consisting of: (i) a thinking reflection $\mathbf{a}_t^{\text{think}}$ that summarizes the current state and highlights potential knowledge gaps; (ii) a composition indicator $\alpha_t \in \mathcal{A}_{\text{type}} = \{\texttt{(query, retrieve)}, \texttt{(answer)}\}$ that determines the sub-action structure; and (iii) a content output $\mathbf{a}_t^{\text{out}} \in \mathcal{A}_{\text{content}}$, representing either a retrieval query or a final answer. We model this decision-making process as a hierarchical policy conditioned on the agent state $\mathbf{s}_t \in \mathcal{S}$. The policy is factorized as:

$$\begin{aligned} &\pi_\theta(\mathbf{a}_t^{\text{think}}, \alpha_t, \mathbf{a}_t^{\text{out}} \mid \mathbf{s}_t) = \\ &\pi_\theta(\mathbf{a}_t^{\text{out}} \mid \alpha_t, \mathbf{a}_t^{\text{think}}, \mathbf{s}_t) \cdot \pi_\theta(\alpha_t \mid \mathbf{a}_t^{\text{think}}, \mathbf{s}_t) \cdot \pi_\theta(\mathbf{a}_t^{\text{think}} \mid \mathbf{s}_t), \end{aligned} \tag{6}$$

where π_θ denotes the LLM-parameterized policy, which encourages three aligned behaviors: generating reflections $\mathbf{a}_t^{\text{think}}$ that assess knowledge sufficiency, selecting α_t to balance exploration and termination, and producing $\mathbf{a}_t^{\text{out}}$ that advances retrieval $\mathbf{a}_t^{\text{query}}$ or yields a direct answer $\mathbf{a}_t^{\text{ans}}$.

Knowledge Interaction via Hypergraph Retrieval. Given a query $\mathbf{a}_t^{\text{query}}$ generated by the reasoning LLM, we retrieve relevant knowledge $\mathbf{a}_t^{\text{ret}}$ from the hypergraph $\mathcal{G}_H = (V, E_H)$ through a dual-path interaction process: entity-based retrieval and direct hyperedge retrieval. The resulting n-ary relational facts are then aggregated via rank-based fusion to support downstream reasoning.

(i) Entity-based Hyperedge Retrieval. We first identify a set of top-ranked entities based on their similarity to the extracted entities $V_{\mathbf{a}_t^{\text{query}}}$, and collect hyperedges that connect to any retrieved entity:

$$\begin{aligned} &\mathcal{R}_V(\mathbf{a}_t^{\text{query}}) = \underset{v \in V}{\text{argmax}}^{k_V} \left(\text{sim}(\phi(V_{\mathbf{a}_t^{\text{query}}}), \phi(v)) \right), \\ &\text{and } \mathcal{F}_V^* = \bigcup_{v_i \in \mathcal{R}_V} \{(e_H, V_{e_H}) \mid v_i \in V_{e_H}, e_H \in E_H\}, \end{aligned} \tag{7}$$

where $\phi(V_{\mathbf{a}_t^{\text{query}}})$ is the aggregated embedding of entities extracted from $\mathbf{a}_t^{\text{query}}$, $\phi(v)$ is the entity embedding, k_V is the number of retrieved entities, and V_{e_H} denotes the entity set of hyperedge e_H.

(ii) Direct Hyperedge Retrieval. In parallel, we directly retrieve hyperedges based on query-hyperedge similarity, and collect their associated relational facts:

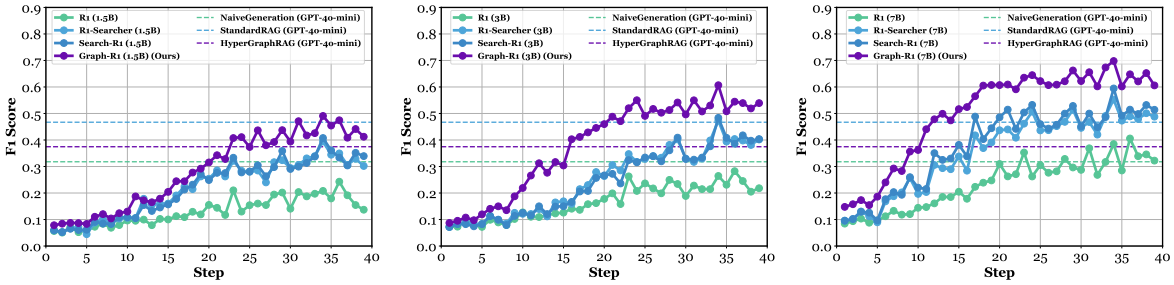
$$\begin{aligned} &\mathcal{R}_H(\mathbf{a}_t^{\text{query}}) = \underset{e_H \in E_H}{\text{argmax}}^{k_H} \left(\text{sim}(\phi(\mathbf{a}_t^{\text{query}}), \phi(e_H)) \right), \\ &\text{and } \mathcal{F}_H^* = \bigcup_{e_i \in \mathcal{R}_H} \{(e_i, V_{e_i}) \mid V_{e_i} \subseteq V\}, \end{aligned} \tag{8}$$

where $\phi(\mathbf{a}_t^{\text{query}})$ is the query embedding, $\phi(e_H)$ is the hyperedge embedding, k_H is the number of retrieved hyperedges, and V_{e_i} denotes the entity set of hyperedge e_i.

(iii) Fusion via Reciprocal Rank Aggregation. To produce the final knowledge set, we merge results from both retrieval paths using reciprocal rank aggregation over hyperedges:

$$\mathbf{a}_t^{\text{ret}} = \mathcal{F}_{\mathbf{a}_t^{\text{query}}}^* = \text{Top-}k \left(\mathcal{F}_V^* \cup \mathcal{F}_H^*, \ \text{Score}(f) = \frac{1}{r_V} + \frac{1}{r_H} \right), \tag{9}$$

where r_V and r_H are the ranks of n-ary relational fact f in $\mathcal{F}_V^*$ and $\mathcal{F}_H^*$ respectively (set to ∞ if absent), and k is the number of retrieved facts $\mathbf{a}_t^{\text{ret}}$ returned to the agent.



(a) Qwen2.5-1.5B-Instruct *(b)* Qwen2.5-3B-Instruct *(c)* Qwen2.5-7B-Instruct

Figure 4. Step-wise F1 score based on Qwen2.5 (1.5B, 3B, 7B), where Graph-R1 outperforms baselines and GPT-4o-mini variants.

Optimization Objective for Agent Trajectories. The agent aims to learn a reasoning trajectory $\tau \in \mathcal{T}_q$ that yields a faithful and contextually grounded answer y_q. Each trajectory $\tau = ((\mathbf{s}_1, \mathbf{a}_1), (\mathbf{s}_2, \mathbf{a}_2), \ldots, (\mathbf{s}_T, \mathbf{a}_T))$ comprises a sequence of actions executed over $\mathcal{G}_H$, defined as:

$$\max_{\theta} \ \mathbb{E}_{\tau \sim \pi_\theta(\mathcal{T}_q | q; \mathcal{G}_H)} \left[\log P(y_q \mid \tau) \right], \tag{10}$$

where $P(y_q \mid \tau)$ denotes the likelihood of the correct answer $y_q \sim \mathbf{a}_t^{\text{ans}}$ under trajectory τ, guiding π_θ toward answer-consistent reasoning.

Proposition 2. *Multi-turn interaction with the graph environment improves retrieval efficiency.*

Proof. We provide quantitative experimental results in Section 5.5 and qualitative proofs in Appendix B.2. □

4.3. Outcome-directed End-to-end RL Optimization

To optimize the reasoning policy π_θ toward generating faithful and well-structured answers, we adopt an end-to-end reinforcement learning objective based on Group Relative Policy Optimization (GRPO) (Shao et al., 2024) $\mathcal{J}_{\text{GRPO}}(\theta)$ and design an outcome-directed reward function $R(\tau)$.

End-to-end RL Objective $\mathcal{J}_{\text{GRPO}}(\theta)$. Given a dataset question $q \in \mathcal{D}_Q$, the agent interacts with the knowledge hypergraph $\mathcal{G}_H$ to generate a group of multi-turn reasoning trajectories $\{\tau_i\}_{i=1}^N \subseteq \mathcal{T}_q$, where each $\tau_i = ((\mathbf{s}_1^{(i)}, \mathbf{a}_1^{(i)}), \ldots, (\mathbf{s}_T^{(i)}, \mathbf{a}_T^{(i)}))$ denotes a sequence of state-action pairs sampled from the environment. We optimize the policy π_θ using the GRPO-based objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\left[\mathbf{s}_1 \sim \{\mathbb{P}(q) | q \in \mathcal{D}_Q\},\ \{\tau_i\}_{i=1}^N \sim \pi_{\theta_{\text{old}}}(\mathcal{T}_q | \mathbf{s}_1; \mathcal{G}_H)\right]}$$

$$\left[\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\tau_i|} \sum_{t=1}^{|\tau_i|} \min\left(\rho_\theta(\mathbf{a}_t^{(i)}) \hat{A}(\tau_i), \text{clip}\left(\rho_\theta(\mathbf{a}_t^{(i)}), 1 \pm \epsilon \right) \hat{A}(\tau_i) \right) \right.$$

$$\left. - \beta \, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right], \ \text{where} \ \rho_\theta(\mathbf{a}_t^{(i)}) = \frac{\pi_\theta(\mathbf{a}_t^{(i)} \mid \mathbf{s}_{t-1}^{(i)}; \mathcal{G}_H))}{\pi_{\theta_{\text{old}}}(\mathbf{a}_t^{(i)} \mid \mathbf{s}_{t-1}^{(i)}; \mathcal{G}_H))},$$

$$\text{and} \ \hat{A}(\tau_i) = \frac{R(\tau_i) - \text{mean}\left(\{R(\tau_j)\}_{j=1}^N\right)}{F_{\text{norm}}\left(\{R(\tau_j)\}_{j=1}^N\right)}. \tag{11}$$

Here, π_θ is the current policy, and $\pi_{\theta_{\text{old}}}$ is the behavior policy used for sampling. The importance ratio $\rho_\theta(\mathbf{a}_t^{(i)})$ adjusts for distribution shift, while the advantage $\hat{A}(\tau_i)$ normalizes the reward using a scaling function $F_{\text{norm}}(\cdot)$ (e.g., standard deviation). The clip$(\cdot)$ operator stabilizes updates by constraining policy shifts. A KL term $\mathrm{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})$ regularizes toward a reference policy π_{ref}, with β controlling its strength. This objective encourages high-reward, stable reasoning over $\mathcal{G}_H$.

Outcome-directed Reward Function $R(\tau)$. To meet outcome requirements, we define a reward function $R(\tau)$ composed of two parts: a *format reward* $R_{\text{format}}(\tau)$ and an *answer reward* $R_{\text{answer}}(\mathbf{a}_T^{\text{ans}})$, promoting both thoughtful retrieval and accurate answer generation.

(i) Format Reward. The format reward $R_{\text{format}}(\tau)$ encourages the agent to follow the intended reasoning structure. At each step $(\mathbf{s}_t, \mathbf{a}_t)$, we check whether the output includes a well-formed block $(\mathbf{a}_t^{\text{think}}, \alpha_t, \mathbf{a}_t^{\text{out}})$. Each valid step receives 0.5 reward, capped at 1.0 overall:

$$R_{\text{format}}(\tau) = \min\left(1.0,\ 0.5 \cdot \sum_{t=1}^{T} \mathbb{I}\left\{(\mathbf{a}_t^{\text{think}}, \alpha_t, \mathbf{a}_t^{\text{out}})\right\}\right), \tag{12}$$

where $\mathbb{I}\{\cdot\}$ is an indicator function that returns 1 if the step output matches the expected format.

(ii) Answer Reward. The answer reward $R_{\text{answer}}(\mathbf{a}_T^{\text{ans}})$ measures the semantic correctness of the generated answer $\mathbf{a}_T^{\text{ans}}$ by comparing it with the ground-truth answer y_q^* using a token-level F1 score $R_{\text{answer}}(\mathbf{a}_T^{\text{ans}})$.

(iii) Overall Outcome Reward. The total reward for a reasoning trajectory τ is defined as:

$$R(\tau) = -1.0 + R_{\text{format}}(\tau) + \mathbb{I}\{R_{\text{format}}(\tau) = 1.0\} \cdot R_{\text{answer}}(\mathbf{a}_T^{\text{ans}}), \tag{13}$$

where $\mathbf{a}_T^{\text{ans}} \in \tau$, ensuring that answer correctness is only rewarded when the format is structurally valid.

Proposition 3. *End-to-end RL bridges the gap between graph-based knowledge and language.*

Proof. We provide quantitative experimental results in Section 5.6 and qualitative proofs in Appendix B.3. □

Table 2. Main results with best in **bold**. means prompt engineering, means training, means no knowledge interaction, means chunk-based knowledge, and means graph-based knowledge.

Method	2Wiki.		HotpotQA		Musique		NQ		PopQA		TriviaQA		Avg.			
	F1	G-E	F1	G-E	F1	G-E	F1	G-E	F1	G-E	F1	G-E	EM	F1	R-S	G-E
GPT-4o-mini																
NaiveGeneration	17.03	74.86	31.79	78.48	11.45	76.61	21.59	84.64	25.95	**72.75**	47.73	83.33	11.36	25.92	-	78.45
StandardRAG	**22.31**	73.02	**46.70**	**81.88**	17.31	74.93	**26.85**	84.55	**30.58**	69.42	**48.55**	84.63	**18.10**	**32.05**	52.68	78.07
GraphRAG	16.02	72.81	31.67	77.37	15.14	74.43	20.31	82.36	20.92	65.88	45.13	82.76	12.50	24.87	32.48	75.94
LightRAG	16.59	71.94	30.70	73.42	14.39	73.75	19.09	80.20	20.47	67.76	40.18	81.60	9.77	23.57	47.42	74.78
PathRAG	12.42	67.19	23.12	71.81	11.49	69.94	20.01	81.99	15.65	60.58	37.44	80.94	7.03	20.02	46.71	72.08
HippoRAG2	16.27	68.78	31.78	76.43	12.37	73.05	24.56	**84.65**	21.10	63.31	46.86	83.55	13.80	25.49	36.41	74.96
HyperGraphRAG	21.14	**76.76**	37.46	80.50	**20.40**	**79.29**	22.95	81.22	29.48	70.55	44.95	**85.20**	13.15	29.40	**61.82**	**78.92**
Qwen2.5-1.5B-Instruct																
NaiveGeneration	7.78	49.13	4.27	45.77	2.35	46.63	6.03	46.74	10.06	42.67	8.10	52.92	1.17	6.43	-	47.31
StandardRAG	11.46	55.38	9.93	52.91	3.18	39.46	11.39	59.73	13.08	50.29	17.43	60.52	5.73	11.08	52.84	53.05
SFT	13.26	34.72	13.61	38.93	5.14	28.50	11.56	46.61	15.61	31.35	26.18	46.66	9.83	14.23	-	37.80
R1	26.28	47.48	20.07	44.43	4.84	39.12	16.75	45.95	21.36	44.50	34.78	48.59	14.19	20.68	-	45.01
Search-R1	28.43	60.61	39.99	64.16	4.69	39.32	20.26	59.93	39.63	58.19	44.16	63.01	23.18	29.53	50.45	57.54
R1-Searcher	28.01	58.81	**41.50**	61.54	6.26	38.31	**36.86**	**60.79**	38.37	56.02	42.57	61.24	23.70	32.26	50.68	56.12
Graph-R1 (ours)	**35.13**	**65.73**	40.62	**65.30**	**28.28**	**58.82**	35.62	59.13	**43.55**	**66.46**	**57.36**	**70.83**	**31.90**	**40.09**	**59.35**	**64.38**
Qwen2.5-3B-Instruct																
NaiveGeneration	7.59	55.00	11.16	53.75	3.67	54.00	8.90	57.18	10.89	49.08	10.89	48.16	3.26	8.85	-	52.86
StandardRAG	12.52	60.01	15.41	62.51	2.92	50.40	10.69	65.13	14.70	57.25	21.92	68.43	3.39	13.03	52.69	60.62
SFT	12.40	52.31	16.48	51.35	5.04	51.31	11.23	58.20	16.95	46.42	33.02	59.98	9.64	15.85	-	53.26
R1	28.45	56.92	25.33	55.38	8.07	47.53	21.51	55.11	27.11	48.65	47.91	60.74	19.66	26.40	-	54.06
Search-R1	38.04	54.39	43.84	69.32	7.65	46.43	37.96	52.90	38.67	63.74	47.99	60.37	28.65	35.69	49.99	57.86
R1-Searcher	23.50	55.86	42.44	64.60	12.81	50.07	36.53	63.33	40.18	66.23	54.00	60.52	27.08	34.91	49.98	60.10
Graph-R1 (ours)	**57.56**	**76.45**	**56.75**	**77.46**	**40.51**	**67.84**	**44.75**	**69.92**	**45.65**	**71.27**	**62.31**	**75.01**	**42.45**	**51.26**	**60.19**	**72.99**
Qwen2.5-7B-Instruct																
NaiveGeneration	12.25	66.75	16.58	65.31	4.06	65.47	13.00	69.56	12.82	60.50	24.51	72.65	3.12	13.87	-	66.71
StandardRAG	12.75	60.06	21.10	66.13	4.53	59.84	15.97	70.49	16.10	60.86	24.90	73.71	5.34	15.89	52.67	65.18
SFT	20.28	63.85	27.59	65.65	10.02	63.50	19.02	68.19	27.93	56.31	39.21	70.25	15.57	24.01	-	64.63
R1	30.99	59.19	37.05	60.12	14.53	49.39	28.45	57.63	30.35	53.38	57.33	66.73	25.91	33.12	-	57.74
Search-R1	41.29	70.26	50.85	73.85	22.35	57.68	45.88	67.58	50.76	66.08	65.98	76.15	38.54	46.19	51.60	68.60
R1-Searcher	33.96	69.61	46.36	74.56	16.63	59.05	44.93	68.54	47.12	66.74	64.76	75.95	34.51	42.29	51.26	69.08
Graph-R1 (ours)	**65.04**	**82.42**	**62.69**	**80.03**	**46.17**	**71.42**	**49.87**	**70.97**	**51.22**	**73.43**	**71.93**	**79.11**	**48.57**	**57.82**	**60.40**	**76.23**

5. Experiments

This section presents the experimental setup, main results, and analysis. We answer the following research questions (RQs): **RQ1:** Does Graph-R1 outperform other methods? **RQ2:** Does the main component of Graph-R1 work, and how is its comparative analysis? **RQ3-5:** How are construction cost, retrieval efficiency, and generation quality?

5.1. Experimental Setup

Datasets. To evaluate the performance of Graph-R1, we conduct experiments across six standard RAG datasets (Jin et al., 2025b): 2WikiMultiHopQA (**2Wiki.**) (Ho et al., 2020), **HotpotQA** (Yang et al., 2018), **Musique** (Trivedi et al., 2022), Natural Questions (**NQ**) (Kwiatkowski et al., 2019), **PopQA** (Mallen et al., 2023), and **TriviaQA** (Joshi et al., 2017). More details are in Appendix D.

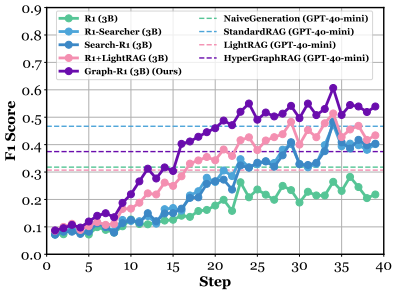
Baselines. We mainly compare Graph-R1 with **NaiveGeneration**, **StandardRAG** (Lewis et al., 2020), **SFT** (Zheng et al., 2024), **R1** (Shao et al., 2024), **Search-R1** (Jin et al., 2025a), and **R1-Searcher** (Song et al., 2025) at three `Qwen2.5` (Qwen et al., 2025) scales: 1.5 B, 3 B, and 7 B. We also compare **GraphRAG** (Edge et al., 2025), **LightRAG** (Guo et al., 2025b), **PathRAG** (Chen et al., 2026), **HippoRAG2** (Gutiérrez et al., 2025), and **HyperGraphRAG** (Luo et al., 2025a) based on `GPT-4o-mini` (OpenAI et al., 2024a) as a reference. More details are in Appendix E.

Evaluation Metrics. We evaluate Graph-R1 and baselines with four metrics: Exact Match (**EM**), **F1**, Retrieval Similarity (**R-S**), and Generation Evaluation (**G-E**), which together assess answer correctness, retrieval relevance, and overall generation quality. These metrics provide a comprehensive evaluation of both retrieval and generation components of the framework. More details are in Appendix F.

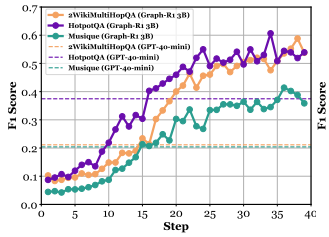
Implementation Details. We use `GPT-4o-mini` for knowledge construction in Graph-R1 and GraphRAG baselines. For retrieval, we use `bge-large-en-v1.5` (Chen et al., 2023) in all variants. All experiments are done on 4 NVIDIA A100 GPUs (80GB) with identical training and inference settings. More details are in Appendix G.

Method	2Wiki.				HotpotQA				Avg.			
	EM	F1	R-S	G-E	EM	F1	R-S	G-E	EM	F1	R-S	G-E
Qwen2.5-3B-Instruct												
Graph-R1	**50.00**	**57.56**	**55.78**	**76.45**	**50.78**	**56.75**	**54.74**	**77.46**	**50.39**	**57.16**	**55.26**	**76.96**
w/o K.C.	36.33	44.94	53.46	65.83	40.63	47.27	53.23	72.69	38.48	46.11	53.35	69.26
w/o M.I.	21.88	34.34	54.70	66.59	30.86	37.64	53.34	64.75	26.37	35.99	54.02	65.67
w/o R.L.	0.78	8.91	10.16	47.14	5.47	12.56	14.44	58.60	3.13	10.74	12.30	52.87
Qwen2.5-7B-Instruct												
Graph-R1	**55.47**	**65.04**	**55.24**	**82.42**	**57.03**	**62.69**	**56.27**	**80.03**	**56.25**	**63.87**	**55.76**	**81.23**
w/o K.C.	44.14	51.81	54.10	75.90	49.22	55.93	54.14	76.78	46.68	53.87	54.12	76.34
w/o M.I.	37.50	44.78	54.54	69.98	40.63	47.04	54.58	69.63	39.07	45.91	54.56	69.81
w/o R.L.	0.00	18.25	54.63	75.81	3.12	17.33	53.80	78.92	1.56	17.79	54.22	77.37

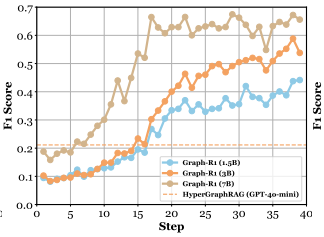
(a) Ablation Study



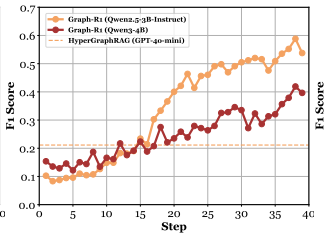
(b) Representations



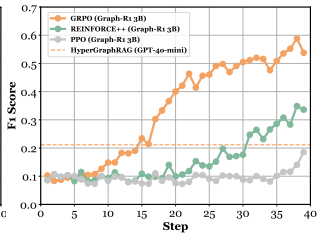
(c) Datasets



(d) Parameters



(e) Qwen3



(f) Algorithms

Figure 5. (a) Ablation study of Graph-R1. (b–f) Performance comparison across different kinds of knowledge representations, RAG datasets, model parameters, Qwen versions, and RL algorithms.

5.2. Main Results (RQ1)

As shown in Table 2, we compare Graph-R1 with baselines across different base models, and observe that Graph-R1 consistently outperforms all baselines. In addition, we have two key observations.

RL Unlocks the Power of Graph Representations. Prompt-only GraphRAG methods often underperform StandardRAG, showing that graph structures alone are not sufficient. Graph-R1, with multi-turn RL optimization, fully exploits structural signals, achieving 57.28 F1 under Qwen2.5-7B-Instruct, surpassing StandardRAG (32.05), HyperGraphRAG (29.40) and Search-R1 (46.19).

Larger Base Model Further Enhances Performance. As base model size increases from 1.5B to 3B and 7B, Graph-R1 achieves steadily higher F1 scores: 40.09, 51.26, and 57.82. Moreover, its gap over other RL-enhanced baselines such as Search-R1 and R1-Searcher becomes increasingly evident. This shows that larger models better exploit the synergy between graph structures and RL.

5.3. Ablation Study and Comparative Analysis (RQ2)

Figures 6 shows an ablation study and comparative analysis.

Ablation Study. We remove three core components of Graph-R1: knowledge construction (K.C.), multi-turn interaction (M.I.), and reinforcement learning (R.L.), to assess their individual contributions. As shown in Figure 5a, removing any module leads to performance degradation.

Comparison with Different Knowledge Representations. As shown in Figures 4 and 5b, models without external knowledge (green) perform the worst. Chunk-based knowledge with RL (blue) performs better, but is still inferior to graph-based methods using binary relations (pink), while hypergraph-based knowledge with RL (red) achieves the highest ceiling. This demonstrates that, when combined with RL, stronger knowledge representations yield higher performance potential.

Comparison across Datasets and Base Models. As shown in Figures 5c and 5d, Graph-R1 consistently outperforms baselines across different datasets and parameter sizes, showcasing strong scalability. Interestingly, Figure 5e shows that when Graph-R1 is trained on Qwen3 (4B) (Yang et al., 2025a), which is already well trained by RL, the model tends to over-rely on its own internal reasoning. Despite a stronger starting point, its overall performance ceiling appears slightly lower.

Comparison with Different RL Algorithms. Figure 5f compares different RL strategies. GRPO significantly outperforms REINFORCE++ (Hu et al., 2025) and PPO (Schulman et al., 2017), achieving the highest F1. This confirms that GRPO facilitates more stable training and stronger multi-turn graph reasoning, making it a favorable choice for training agentic GraphRAG models.

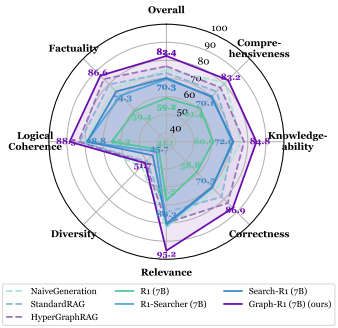
5.4. Analysis of Graph-R1's Construction Cost (RQ3)

As shown in Table 6a, we utilize metrics: time per 1K tokens (TP1KT), cost per 1M tokens (CP1MT), number of nodes & edges, time per query (TPQ), cost per 1K queries (CP1KQ), and final F1 score.

Construction Cost. Graph-R1 requires only 5.69 seconds

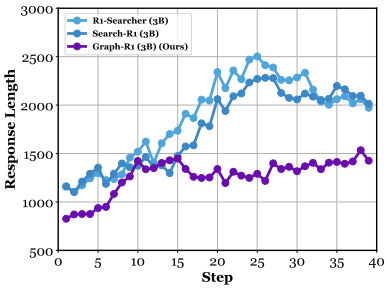
Method	Knowledge Construction				Retrieval & Generation		
	TP1KT	CP1MT	#Node	#Edge	TPQ	CP1KQ	F1
NaiveGeneration	0 s	0 $	-	-	3.7 s	0.16 $	17.0
StandardRAG	0 s	0 $	-	-	4.1 s	1.35 $	22.3
GraphRAG	8.04 s	3.35 $	7,771	4,863	7.4 s	3.97 $	16.0
LightRAG	6.84 s	4.07 $	59,197	24,596	12.2 s	8.11 $	16.6
PathRAG	6.84 s	4.07 $	59,197	24,596	15.8 s	8.28 $	12.4
HippoRAG2	3.25 s	1.26 $	11,819	40,654	8.8 s	7.68 $	16.3
HyperGraphRAG	6.76 s	4.14 $	173,575	114,426	9.6 s	8.76 $	21.1
Graph-R1 (7B) (ours)	5.69 s	2.81 $	120,499	98,073	7.0 s	0 $	65.0

(a)

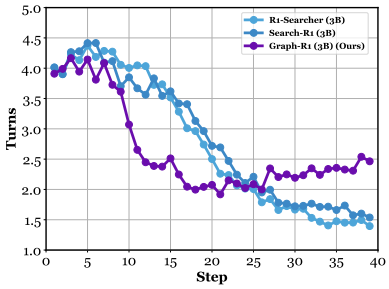


(b)

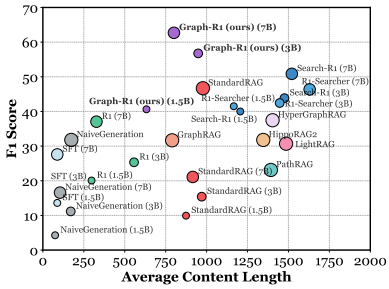
Figure 6. (a) Time & Cost Comparisons on 2Wiki. (b) Generation Evaluations.



(a) Response Length



(b) Turns of Interaction



(c) Efficiency Comparison

Figure 7. Step-wise response length & turns of interaction, and efficiency comparison on HotpotQA.

and $2.81 per 1K tokens for knowledge construction, lower than GraphRAG (8.04s, $3.35) and HyperGraphRAG (6.76s, $4.14). Generating over 120K nodes and 98K edges, Graph-R1 maintains a semantically rich structure.

Generation Cost. By leveraging end-to-end RL and localized knowledge retrieval, Graph-R1 achieves not only the best F1 but also a response time of 7.0s per query and a generation cost of $0, outperforming baselines such as HyperGraphRAG (9.6s, $8.76), highlighting its superior potential for real-world deployment.

5.5. Analysis of Graph-R1's Retrieval Efficiency (RQ4)

As shown in Figure 7, to evaluate Graph-R1's retrieval efficiency, we analyze it from (a) response length, (b) number of interaction turns, and (c) performance with average retrieval content lengths.

Tendency toward More Concise Thinking and Adequate Interaction. As shown in Figures 7a and 7b, Graph-R1 generates shorter responses and conducts more interaction turns, averaging around 1200-1500 tokens and 2.3-2.5 turns, leading to more stable and accurate retrieval.

Balancing Performance and Retrieved Content Length. As shown in Figure 7c, Graph-R1 achieves the highest F1 scores with a moderate amount of average retrieved content compared to other methods, balancing input length and performance through its multi-turn interaction strategy.

5.6. Analysis of Graph-R1's Generation Quality (RQ5)

As shown in Figure 6b, we evaluate the generation quality in seven dimensions and present a case study in Table 4.

High-Quality Generation Performance. Graph-R1 outperforms all RL-based baselines and achieves generation quality comparable to GPT-4o-mini-based methods like HyperGraphRAG, with strong results in Correctness (86.9), Relevance (95.2), and Logical Coherence (88.5).

RL Bridges the Gap Between Graph & Language. HyperGraphRAG performs similarly to StandardRAG, indicating limited gains from graph structure alone. In contrast, Graph-R1 achieves a much higher Overall score (82.4 vs. 70.3) than Search-R1, showing that graph-based reasoning becomes truly effective when combined with RL.

6. Conclusion

In this work, we introduce Graph-R1, the agentic GraphRAG framework powered by end-to-end RL. By introducing lightweight knowledge hypergraph construction and modeling retrieval as a multi-turn interaction process, Graph-R1 bridges graph-structured knowledge with natural language generation. A unified reward mechanism enables outcome-directed reasoning. Experiments across six benchmarks demonstrate Graph-R1's superiority in accuracy, retrieval efficiency, and generation quality.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62473271, Grant No. 62176026, and Grant No. 62406036) and the Fundamental Research Funds for the Beijing University of Posts and Telecommunications (Grant No. 2025AI4S03). We also acknowledge the support from the Engineering Research Center of Information Networks, Ministry of Education, China. This work is also supported by the National Research Foundation, Singapore, under its Frontier Competitive Research Programme (NRF-F-CRP-2024-0005). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

Impact Statement

This work poses no significant ethical concerns, as it relies solely on publicly available datasets, and its broader societal impact is expected to be positive by advancing reliable and structured knowledge-grounded reasoning for retrieval-augmented generation systems.

References

Chen, B., Guo, Z., Yang, Z., Chen, Y., Chen, J., Liu, Z., Shi, C., and Yang, C. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30183–30191, Mar. 2026. doi: 10.1609/aaai.v40i36.40268. URL https://ojs.aaai.org/index.php/AAAI/article/view/40268.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2023.

Dong, J., An, S., Yu, Y., Zhang, Q.-W., Luo, L., Huang, X., Wu, Y., Yin, D., and Sun, X. Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning, 2025. URL https://arxiv.org/abs/2508.19855.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. From local to global: A graph rag approach to query-focused summarization, 2025. URL https://arxiv.org/abs/2404.16130.

Feng, L., Xue, Z., Liu, T., and An, B. Group-in-group policy optimization for llm agent training. In Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., and Chen, N. (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 46375–46408. Curran Associates, Inc., 2025.

Feng, Y., Hu, H., Ying, S., Hou, X., Liu, S., Yang, M., Li, J., Du, S., Zheng, N., Hu, H., and Gao, Y. Hyper-rag: combating llm hallucinations using hypergraph-driven retrieval-augmented generation. *Nature Communications*, 2026a. ISSN 2041-1723. doi: 10.1038/s41467-026-71411-1. URL https://doi.org/10.1038/s41467-026-71411-1.

Feng, Y., Luo, H., Feng, L., Zhao, S., and Luu, A. T. From stimuli to minds: Enhancing psychological reasoning in llms via bilateral reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1):274–282, Mar. 2026b. doi: 10.1609/aaai.v40i1.36988. URL https://ojs.aaai.org/index.php/AAAI/article/view/36988.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Gao, Y., Xiong, Y., Zhong, Y., Bi, Y., Xue, M., and Wang, H. Synergizing rag and reasoning: A systematic review, 2025. URL https://arxiv.org/abs/2504.15909.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025a.

Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. LightRAG: Simple and fast retrieval-augmented generation. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., and Su, Y. From RAG to memory: Non-parametric continual learning for large language models. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (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.

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

Hu, J., Liu, J. K., Xu, H., and Shen, W. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization, 2025. URL https://arxiv.org/abs/2501.03262.

Jiang, Y., Wang, J., Shen, Z., Lin, Z., Wang, J., Yang, Y., Dai, K., and Luo, H. Rethinking scientific modeling: Toward physically consistent and simulation-executable programmatic generation, 2026. URL https://arxiv.org/abs/2602.07083.

Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S. O., Wang, D., Zamani, H., and Han, J. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025a. URL https://openreview.net/forum?id=Rwhi91ideu.

Jin, J., Zhu, Y., Dou, Z., Dong, G., Yang, X., Zhang, C., Zhao, T., Yang, Z., and Wen, J.-R. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 737–740, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715313. URL https://doi.org/10.1145/3701716.3715313.

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

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Lee, M.-C., Zhu, Q., Mavromatis, C., Han, Z., Adeshina, S., Ioannidis, V. N., Rangwala, H., and Faloutsos, C. HybGRAG: Hybrid retrieval-augmented generation on textual and relational knowledge bases. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 879–893, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.43. URL https://aclanthology.org/2025.acl-long.43/.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020.

Liang, L., Bo, Z., Gui, Z., Zhu, Z., Zhong, L., Zhao, P., Sun, M., Zhang, Z., Zhou, J., Chen, W., Zhang, W., and Chen, H. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 334–343, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715240. URL https://doi.org/10.1145/3701716.3715240.

Lin, Z., Tan, Y., Chen, J., Zhang, H., Chen, C., Wang, S., and Yang, C. Unified heterogeneous hypergraph construction for incomplete multimedia recommendation. *ACM Transactions on Information Systems*, 43(5):1–31, 2025.

Liu, H., Wang, Z., Chen, X., Li, Z., Xiong, F., Yu, Q., and Zhang, W. HopRAG: Multi-hop reasoning for logic-aware retrieval-augmented generation. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1897–1913, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.97. URL https://aclanthology.org/2025.findings-acl.97/.

Liu, W., Luo, H., Lin, X., Liu, H., Shen, T., Wang, J., Mao, R., and Cambria, E. Prompt-r1: Collaborative automatic prompting framework via end-to-end reinforcement learning, 2026. URL https://arxiv.org/abs/2511.01016.

Luo, H., E, H., Yang, Y., Yao, T., Guo, Y., Tang, Z., Zhang, W., Peng, S., Wan, K., Song, M., Lin, W., Zhu, Y., and Luu, A. T. Text2nkg: Fine-grained n-ary relation extraction for n-ary relational knowledge graph construction. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 27417–27439. Curran Associates, Inc., 2024.

Luo, H., E, H., Chen, G., Zheng, Y., Wu, X., Guo, Y., Lin, Q., Feng, Y., Kuang, Z., Song, M., Zhu, Y., and Luu, A. T. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. In Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., and Chen, N. (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 152206–152234. Curran Associates, Inc., 2025a.

Luo, H., E, H., Guo, Y., Lin, Q., Wu, X., Mu, X., Liu, W., Song, M., Zhu, Y., and Luu, A. T. KBQA-o1: Agentic knowledge base question answering with Monte Carlo tree search. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41177–41199. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/luo25d.html.

Luo, L., Zhao, Z., Haffari, R., Phung, D., Gong, C., and Pan, S. Gfm-rag: Graph foundation model for retrieval augmented generation. In Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., and Chen, N. (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 36371–36405. Curran Associates, Inc., 2025c.

Luo, L., Zhao, Z., Liu, J., Qiu, Z., Dong, J., Panev, S., Gong, C., Vu, T.-T., Haffari, G., Phung, D., Liew, A. W.-C., and Pan, S. G-reasoner: Foundation models for unified reasoning over graph-structured knowledge. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zJm9nmoahk.

Ma, P., Zhuang, X., Xu, C., Jiang, X., Chen, R., and Guo, J. Sql-r1: Training natural language to sql reasoning model by reinforcement learning. In Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., and Chen, N. (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 174505–174537. Curran Associates, Inc., 2025.

Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., and Hajishirzi, H. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

OpenAI, :, Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card, 2024a. URL https://arxiv.org/abs/2410.21276.

OpenAI, :, Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card, 2024b. URL https://arxiv.org/abs/2412.16720.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Song, H., Jiang, J., Min, Y., Chen, J., Chen, Z., Zhao, W. X., Fang, L., and Wen, J.-R. R1-searcher: Incentivizing the search capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2503.05592.

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

Wang, J. and Han, J. PropRAG: Guiding retrieval with beam search over proposition paths. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6212–6227, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.317. URL https://aclanthology.org/2025.emnlp-main.317/.

Wang, J., Fu, J., Wang, R., Song, L., and Bian, J. Pikerag: specialized knowledge and rationale augmented generation, 2025a. URL https://arxiv.org/abs/2501.11551.

Wang, N., Han, X., Singh, J., Ma, J., and Chaudhary, V. CausalRAG: Integrating causal graphs into retrieval-augmented generation. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22680–22693, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1165. URL https://aclanthology.org/2025.findings-acl.1165/.

Wu, J., Zhu, J., Qi, Y., Chen, J., Xu, M., Menolascina, F., Jin, Y., and Grau, V. Medical graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28443–28467, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1381. URL https://aclanthology.org/2025.acl-long.1381/.

Wu, X. A comprehensive survey on learning from rewards for large language models: Reward models and learning strategies. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17847–17875, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.970. URL https://aclanthology.org/2025.findings-emnlp.970/.

Xiao, Y., Dong, J., Zhou, C., Dong, S., Zhang, Q., Yin, D., Sun, X., and Huang, X. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2506.02404.

Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Wang, Y., and Chen, E. Large language models for generative information extraction: a survey. *Front. Comput. Sci.*, 18(6), November 2024. ISSN 2095-2228. doi: 10.1007/s11704-024-40555-y. URL https://doi.org/10.1007/s11704-024-40555-y.

Xu, T., Zheng, H., Li, C., Chen, H., Liu, Y., Chen, R., and Sun, L. Noderag: Structuring graph-based rag with heterogeneous nodes, 2025. URL https://arxiv.org/abs/2504.11544.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Yang, C., Wu, X., Lin, X., Xu, C., Jiang, X., Sun, Y., Li, J., Xiong, H., and Guo, J. Graphsearch: An agentic deep searching workflow for graph retrieval-augmented generation, 2025b. URL https://arxiv.org/abs/2509.22009.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., Wang, L., Luu, A. T., Bi, W., Shi, F., and Shi, S. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418, 12 2025. ISSN 0891-2017. doi: 10.1162/COLI.a.16. URL https://doi.org/10.1162/COLI.a.16.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. A survey of large language models, 2025a. URL https://arxiv.org/abs/2303.18223.

Zhao, Y., Zhu, J., Guo, Y., He, K., and Li, X. E2graphrag: Streamlining graph-based rag for high efficiency and effectiveness, 2025b. URL https://arxiv.org/abs/2505.24226.

Zheng, Y., Zhang, R., Zhang, J., Ye, Y., and Luo, Z. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In Cao, Y., Feng, Y., and Xiong, D. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 400–410, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.38. URL https://aclanthology.org/2024.acl-demos.38/.

Zhuang, L., Chen, S., Xiao, Y., Zhou, H., Zhang, Y., Chen, H., Zhang, Q., and Huang, X. Linearrag: Linear graph retrieval augmented generation on large-scale corpora, 2025. URL https://arxiv.org/abs/2510.10114.

Appendix

A. Prompts Used in Graph-R1

A.1. Knowledge HyperGraph Construction Prompt

As shown in Figure 8, we use the same n-ary relation-extraction prompt as HyperGraphRAG (Luo et al., 2025a). Moreover, we streamline knowledge-hypergraph construction by skipping confidence-score calculations and adopting a simpler semantic-retrieval method, reducing construction costs while maintaining equivalent knowledge-representation.

```
-Goal-
Given a text document that is potentially relevant to this activity and a list of entity types, identify all entities of those types from the text and all relationships among the identified entities.
Use {language} as output language.

-Steps-
1. Divide the text into several complete knowledge segments.  For each knowledge segment, extract the following information:
-- knowledge_segment: A sentence that describes the context of the knowledge segment.
Format each knowledge segment as ("hyper-relation"{tuple_delimiter}<knowledge_segment>)

2. Identify all entities in each knowledge segment. For each identified entity, extract the following information:
- entity_name: Name of the entity, use same language as input text. If English, capitalized the name.
- entity_type: Type of the entity.
- entity_description: Comprehensive description of the entity's attributes and activities.
Format each entity as ("entity"{tuple_delimiter}<entity_name>{tuple_delimiter}<entity_type>{tuple_delimiter}<entity_description>)

3. Return output in {language} as a single list of all the entities and relationships identified in steps 1 and 2. Use **{record_delimiter}** as the list delimiter.

4. When finished, output {completion_delimiter}

######################
-Examples-
######################
{examples}

##############################
-Real Data-
######################
Text: {input_text}
######################
```

Figure 8. Prompt for n-ary relation extraction π_{ext} in Equation 4.

A.2. Agentic Knowledge Reasoning Prompt

The initial knowledge-reasoning prompt has been shown in Table 1. Through several rounds of interaction with the environment, the Graph-R1 agent keeps adding information, readying the prompt for the next exchange. As shown in Figure 9, we present a case of the final prompt produced by the full agentic knowledge-reasoning process.

```
<|im_start|>system
You are a helpful assistant.
<|im_end|>
<|im_start|>user
Answer the given question. You can query from knowledge base provided to you to answer the question. You can query knowledge as many times as you want.
You must first conduct reasoning inside <think>...</think>. If you need to query knowledge, you can set a query statement between <query>...</query> to query from knowledge base after <think>...</think>.
When you have the final answer, you can output the answer inside <answer>...</answer>.

Output format for query:
<think>
...
</think>
<query>
...
</query>

Output format for answer:
<think>
...
</think>
<answer>
...
</answer>
Question: Who is the spouse of the director of film In Memory Of Sergo Ordzhonikidze?<|im_end|>
<|im_start|>assistant
<think>
To answer this question, I need to find out who directed the film \"In Memory Of Sergo Ordzhonikidze\" and then find out who that director's spouse is.
</think>
<query>
{
 \"query\": \"Director of film In Memory Of Sergo Ordzhonikidze\"
}
</query><|im_end|><|im_start|>user
<knowledge>
{\"results\": [{\"<knowledge>\": \"\\\"In Memory of Sergo Ordzhonikidze is a 1937 Soviet documentary film directed by Dziga Vertov.\\\"\", \"<coherence>\": 1.5}, {\"<knowledge>\": \"\\\"The Harsh Life of Veronica Lambert (also released as Five Nights in Hollywood) is a 2009 independent drama film directed and written by Nika Agiashvili.\\\"\", \"<coherence>\": 1.0}, {\"<knowledge>\": \"\\\"Tengiz Abuladze (31 January 1924 \\u2013 6 March 1994) was a Georgian film director, screenwriter, theatre teacher and People's Artist of the USSR.\\\"\", \"<coherence>\": 0.5}, {\"<knowledge>\": \"\\\"The film was created in memory of Sergo Ordzhonikidze, who had died in 1937.\\\"\", \"<coherence>\": 0.393}, {\"<knowledge>\": \"\\\"Leonid Kvinikhidze was a Russian screenwriter and film director who died in St. Petersburg on 13 March 2018.\\\"\", \"<coherence>\": 0.367}, {\"<knowledge>\": \"\\\"The Harsh Life of Veronica Lambert is a 2009 independent drama film directed and written by Nika Agiashvili.\\\"\", \"<coherence>\": 0.333}, {\"<knowledge>\": \"\\\"Tengiz Abuladze was a Georgian film director, screenwriter, theatre teacher and People's Artist of the USSR.\\\"\", \"<coherence>\": 0.333}, {\"<knowledge>\": \"\\\"A Green Story is a 2012 film written and directed by Nika Agiashvili.\\\"\", \"<coherence>\": 0.25}, {\"<knowledge>\": \"\\\"The Secret of Two Oceans' is a 1956 Soviet drama film directed by Konstantine Pipinashvili.\\\"\", \"<coherence>\": 0.2}]}
</knowledge><|im_end|>
<|im_start|>assistant
<think>
From the knowledge base, I found that \"In Memory Of Sergo Ordzhonikidze\" is a 1937 Soviet documentary film directed by Dziga Vertov. Now, I need to find out who Dziga Vertov's spouse was.
</think>
<query>
{
 \"query\": \"Spouse of Dziga Vertov\"
}
</query><|im_end|><|im_start|>user
<knowledge>
{\"results\": [{\"<knowledge>\": \"\\\"In Memory of Sergo Ordzhonikidze is a 1937 Soviet documentary film directed by Dziga Vertov.\\\"\", \"<coherence>\": 1.833}, {\"<knowledge>\": \"\\\"Dziga Vertov was a Russian Soviet pioneer documentary film and newsreel director, as well as a cinema theorist.\\\"\", \"<coherence>\": 1.167}, {\"<knowledge>\": \"\\\"Yulia Meltzer was the wife of Yakov Dzhugashvili and the daughter-in-law of Joseph Stalin.\\\"\", \"<coherence>\": 0.5}, {\"<knowledge>\": \"\\\"Vertov's younger brothers Boris Kaufman and Mikhail Kaufman were also noted filmmakers, as was his wife, Yelizaveta Svilova.\\\"\", \"<coherence>\": 0.45}, {\"<knowledge>\": \"\\\"Vadim Abdrashitov has been enjoying a happy family life with his wife, Natella Toidze, and their two children, son Oleg (born in 1973) and daughter Naina (born in 1980).\\\"\", \"<coherence>\": 0.333}, {\"<knowledge>\": \"\\\"Alexandra Andreyevna Voronin (\n\\u00e9e' Voronina, later Yourieff; 20 August 1905 \\u2014 1 October 1993) was the Soviet wife of Norwegian fascist Vidkun Quisling, the leader of Nasjonal Samling (NS).\\\"\", \"<coherence>\": 0.25}, {\"<knowledge>\": \"\\\"The film was directed by Yuli Raizman and Yelizaveta Svilova.\\\"\", \"<coherence>\": 0.2}]}
</knowledge><|im_end|>
<|im_start|>assistant
<think>
Dziga Vertov's spouse was Yelizaveta Svilova. So, the answer to the question is Yelizaveta Svilova.
</think>
<answer>
Yelizaveta Svilova
</answer><|im_end|>
```

Figure 9. Prompt for agentic knowledge reasoning π_θ in Equation 6.

B. Theoretical Proof

B.1. Proof of Proposition 1

Proposition 1. *Graph-structured knowledge boosts agent accuracy by richer representation.*

Proof. Let the knowledge base $\mathcal{K}$ be encoded into two forms: a graph R_G and a linear chunk set R_C, where $R_C = g(R_G)$ is a deterministic transformation that discards edge information. For a query Q and ground-truth answer $A^\star$, the agent's internal belief at step t is h_t. Each step performs retrieval $\mathcal{E}_t(R)$ and Bayesian update, forming the recurrence:

$$h_{t+1} = f(h_t, R). \tag{14}$$

Define the Lyapunov function as $V_R(h_t) = -\log P(A^\star \mid h_t)$, measuring how far the agent is from certainty. Its update is:

$$\Delta V_R(h_t) = -\log \frac{P(\mathcal{E}_t(R) \mid A^\star)}{\sum_a P(a \mid h_t) P(\mathcal{E}_t(R) \mid a)}. \tag{15}$$

Graphs can capture more relevant facts in shorter contexts due to explicit edges, leading to higher information density δ_R and more negative $\Delta V_R(h_t)$ in expectation. Thus, $V_R(h_t)$ decreases faster with graphs, indicating faster convergence. From an information-theoretic view, the mutual information evolves as:

$$I(A^\star; h_{t+1} \mid Q) = I(A^\star; h_t \mid Q) + I(A^\star; \mathcal{E}_t(R) \mid h_t, Q), \tag{16}$$

and since graphs provide denser evidence, we have $I_{R_G}(A^\star; h_T \mid Q) \geq I_{R_C}(A^\star; h_T \mid Q)$. Then by Fano's inequality,

$$P_e(R) \leq \frac{H(A^\star \mid Q) - I_R(A^\star; h_T \mid Q) + 1}{\log |\mathcal{A}|}, \tag{17}$$

which implies $P_e(R_G) \leq P_e(R_C)$, i.e., $\mathrm{Acc}(R_G) \geq \mathrm{Acc}(R_C)$, with strict inequality when the graph contains structural relations not recoverable from text.

In summary, the graph-structured representation offers higher information density per retrieval, accelerates belief convergence via Lyapunov descent, and accumulates more mutual information, leading to provably higher answer accuracy. □

B.2. Proof of Proposition 2

Proposition 2. *Multi-turn interaction with the graph environment improves retrieval efficiency.*

Proof. Let the graph-structured knowledge base be denoted by R_G, and let $A^\star$ be the ground-truth answer. Suppose the retrieval cost is measured by the number of tokens retrieved, and we fix a total budget of B tokens. A single-turn retrieval strategy selects a fixed token set $\mathcal{E}_{\text{static}}$ of size B, independent of any intermediate reasoning. This leads to a posterior belief $P(A^\star \mid Q, \mathcal{E}_{\text{static}})$ and yields total information gain

$$I_{\text{static}} = I(A^\star; \mathcal{E}_{\text{static}} \mid Q) = H(A^\star \mid Q) - H(A^\star \mid Q, \mathcal{E}_{\text{static}}), \tag{18}$$

where Q is the query and $H(\cdot)$ denotes entropy. In contrast, an adaptive multi-turn strategy π divides the budget across T rounds as $B = \sum_{t=1}^{T} B_t$. At each round t, the agent uses prior evidence $\mathcal{H}_{t-1} = \{\mathcal{E}_1, \ldots, \mathcal{E}_{t-1}\}$ to update its internal belief h_{t-1} and selects new evidence $\mathcal{E}_t$ of size B_t by actively exploring the graph based on current uncertainty. The updated belief h_t is obtained via Bayesian inference, and the entire process forms a dynamic system:

$$h_t = f(h_{t-1}, \mathcal{E}_t, R_G). \tag{19}$$

To evaluate retrieval progress, we define a Lyapunov-style potential function $V_t = H(A^\star \mid Q, \mathcal{H}_t)$, which quantifies the remaining uncertainty after round t. Each retrieval step reduces entropy by:

$$V_{t-1} - V_t = I(A^\star; \mathcal{E}_t \mid Q, \mathcal{H}_{t-1}), \tag{20}$$

which is precisely the mutual information from $\mathcal{E}_t$ given past evidence. Let ρ_t denote information gain per token at round t:

$$\rho_t = \frac{I(A^\star; \mathcal{E}_t \mid Q, \mathcal{H}_{t-1})}{B_t}, \tag{21}$$

and define the average information density of the static strategy as:

$$\rho_{\text{static}} = \frac{I_{\text{static}}}{B}. \tag{22}$$

Since the adaptive agent tailors each $\mathcal{E}_t$ to the current belief state and selects high-impact graph regions, it is expected that $\rho_t \geq \rho_{\text{static}}$ each round. When the graph allows pruning irrelevant branches via early evidence, this inequality becomes strict with non-zero probability. Summing over all rounds, the total information gain of the adaptive strategy satisfies:

$$\mathbb{E}_\pi \left[\sum_{t=1}^{T} I(A^\star; \mathcal{E}_t \mid Q, \mathcal{H}_{t-1}) \right] = \mathbb{E}_\pi \left[I(A^\star; \mathcal{H}_T \mid Q) \right] \geq I_{\text{static}}, \tag{23}$$

with strict inequality under the above conditions. From a Bayesian viewpoint, retrieval efficiency can be seen as how much uncertainty is reduced per token. Because the adaptive policy achieves a greater entropy reduction under the same budget, or requires fewer tokens to reach the same posterior certainty, it is strictly more efficient. Moreover, by Fano's inequality,

$$P_e \leq \frac{H(A^\star \mid Q) - I(A^\star; \mathcal{H}_T \mid Q) + 1}{\log |\mathcal{A}|}, \tag{24}$$

we conclude that the lower the conditional entropy, the lower the expected error. Therefore, greater mutual information directly translates into improved answer accuracy.

In conclusion, multi-turn interaction enables the agent to reason over what has already been retrieved, selectively expanding into the most informative parts of the graph, leading to more efficient and accurate question answering. □

B.3. Proof of Proposition 3

Proposition 3. *End-to-end RL bridges the gap between graph-based knowledge and language.*

Proof. Let G_H denote the graph-structured knowledge base, and q a given query. The agent (parameterized by θ) interacts over multiple steps, forming a trajectory $\tau = (s_1, a_1, \ldots, s_T, a_T)$ where each a_t is either a graph query or a natural-language output. The policy induces an answer distribution:

$$P_\theta(y \mid q, G_H) = \sum_{\tau:\, \text{answer}(\tau)=y} \pi_\theta(\tau \mid q, G_H). \tag{25}$$

To align graph usage with answer generation, we define a trajectory-level reward:

$$R(\tau) = r_{\text{fmt}}(\tau) + \mathbb{I}\{r_{\text{fmt}}(\tau) = 1\} \cdot r_{\text{ans}}(y_T, y_q^\star) - 1, \tag{26}$$

where r_{fmt} ensures proper structure (e.g., retrieve before answer), and r_{ans} measures answer quality. Only valid, grounded answers receive positive reward. The expected reward is maximized via policy gradient:

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t; G_H) \cdot \hat{A}(\tau) \right], \tag{27}$$

where $\hat{A}(\tau)$ derives from $R(\tau)$. Trajectories that retrieve the right subgraph and generate correct answers are reinforced, linking graph retrieval to accuracy. As training progresses, the expected log-likelihood of the gold answer increases:

$$\mathcal{L}_\theta = -\log \sum_{\tau} \pi_\theta(\tau \mid q, G_H) P(y_q^\star \mid \tau), \tag{28}$$

which lower-bounds the ideal $\log P^\star(y_q^\star \mid q, G_H)$. In the limit, we approach:

$$P_\theta(\cdot \mid q, G_H) \to P^\star(\cdot \mid q, G_H). \tag{29}$$

This also manifests as a reduction in conditional entropy:

$$H_\theta(Y \mid Q, G_H) < H(Y \mid Q), \tag{30}$$

since ungrounded answers are discouraged and graph-consistent ones are promoted. By Fano's inequality, lower entropy implies lower error.

Thus, end-to-end RL not only learns to query the graph but also binds retrieved knowledge to answer generation, effectively bridging the gap between structure and language. □

C. Graph-R1 Algorithm Details

To illustrate the mechanism of Graph-R1, we present its full workflow in Algorithm 1, comprising three key phases: **(1) Hypergraph Construction.** An LLM-based extractor π_{ext} extracts n-ary relational facts from the corpus K to build a semantic hypergraph $\mathcal{G}_H = (V, E_H, \phi)$, where all elements are encoded via $\text{Enc}(\cdot)$. **(2) Multi-turn Agentic Reasoning.** Given query q, the agent performs reflection, intent selection, and action generation over T steps under policy π_θ. Queries trigger dual-path hypergraph retrieval; answers terminate reasoning. **(3) End-to-end RL Optimization.** The policy is optimized using GRPO, guided by format and answer rewards. The objective $\mathcal{J}_{\text{GRPO}}$ is computed from sampled trajectories $\{\tau_i\}$ with clipped advantage-weighted updates.

Algorithm 1 Graph-R1: Agentic GraphRAG via End-to-end RL

Require: Query q, knowledge corpus $K = \{d_1, \ldots, d_N\}$, policy π_θ, reward function $R(\tau)$
Ensure: Final answer y_q

// 1: Knowledge Hypergraph Construction
Initialize hypergraph $\mathcal{G}_H = (V, E_H, \phi)$
for each document $d \in K$ **do**
  Extract relational facts: $\{(h_i, \mathcal{V}_{h_i})\} \sim \pi_{\text{ext}}(d)$
  for each $(h_i, \mathcal{V}_{h_i})$ **do**
    $E_H \leftarrow E_H \cup \{h_i\}, \quad V \leftarrow V \cup \mathcal{V}_{h_i}$
    $\phi(h_i) \leftarrow \text{Enc}(h_i), \quad \phi(v) \leftarrow \text{Enc}(v)$ for $v \in \mathcal{V}_{h_i}$
  end for
end for
// 2: Multi-turn Graph Reasoning
Initialize state $s_1 \leftarrow q$, trajectory $\tau \leftarrow \emptyset$
for $t = 1$ to T **do**
  Generate reasoning plan: $\mathbf{a}_t^{\text{think}} \sim \pi_\theta(\cdot \mid s_t)$
  Choose intent: $\alpha_t \sim \pi_\theta(\cdot \mid \mathbf{a}_t^{\text{think}}, s_t)$
  if $\alpha_t =$ `(answer)` **then**
    Output answer: $\mathbf{a}_t^{\text{ans}} \sim \pi_\theta(\cdot \mid \mathbf{a}_t^{\text{think}}, s_t)$
    $\tau \leftarrow \tau \cup \{(s_t, \mathbf{a}_t^{\text{query}}, \mathbf{a}_t^{\text{ans}})\}$; **return** $y_q = \mathbf{a}_t^{\text{ans}}$
  else if $\alpha_t =$ `(query, retrieve)` **then**
    Generate query: $\mathbf{a}_t^{\text{query}} \sim \pi_\theta(\cdot \mid \mathbf{a}_t^{\text{think}}, s_t)$
    Entity retrieval: $\mathcal{R}_V = \text{argmax}_{v \in V}^{k_V} \text{sim}(\phi(v), \phi(V_{\mathbf{a}_t^{\text{query}}}))$
    Hyperedge retrieval: $\mathcal{R}_H = \text{argmax}_{h \in E_H}^{k_H} \text{sim}(\phi(h), \phi(\mathbf{a}_t^{\text{query}}))$
    Rank fusion: $\mathbf{a}_t^{\text{ret}} = \text{Top-}k\left(\mathcal{F}_V^* \cup \mathcal{F}_H^*,\ \text{Score}(f) = \frac{1}{r_V(f)} + \frac{1}{r_H(f)}\right)$
    Update state $s_{t+1} \leftarrow s_t \cup \{(s_t, \mathbf{a}_t^{\text{think}}, \mathbf{a}_t^{\text{query}}, \mathbf{a}_t^{\text{ret}})\}$
    $\tau \leftarrow \tau \cup \{(s_t, \mathbf{a}_t^{\text{think}}, \mathbf{a}_t^{\text{query}}, \mathbf{a}_t^{\text{ret}})\}$
  end if
end for
// 3: End-to-end Policy Optimization (GRPO)
Sample N trajectories $\{\tau_i\} \sim \pi_{\theta_{\text{old}}}$
for each τ_i **do**
  Compute reward: $R(\tau_i) = -1 + R_{\text{format}}(\tau_i) + \mathbb{I}\{R_{\text{format}} = 1\} \cdot R_{\text{answer}}(y_T, y_q^\star)$
  Compute advantage: $\hat{A}(\tau_i) = \frac{R(\tau_i) - \text{mean}(\{R(\tau_j)\})}{\text{std}(\{R(\tau_j)\})}$
end for
Update policy via GRPO: $\mathcal{J}_{\text{GRPO}} \sim \sum_{i=1}^{N} \sum_{t=1}^{|\tau_i|} \min\left(\rho_\theta(a_t^{(i)})\hat{A}(\tau_i), \text{clip}(\rho_\theta(a_t^{(i)}), 1 \pm \epsilon)\hat{A}(\tau_i)\right)$
where $\rho_\theta(a_t^{(i)}) = \frac{\pi_\theta(a_t^{(i)} \mid s_{t-1}^{(i)})}{\pi_{\theta_{\text{old}}}(a_t^{(i)} \mid s_{t-1}^{(i)})}$

Complexity Analysis. Graph-R1 involves three computational components corresponding to phases. First, hypergraph construction scales with the total token count T_K of the knowledge corpus and the number of extracted relational facts F, yielding complexity $O(T_K)+O(F)$. Second, during multi-turn reasoning, the agent performs T steps of action sampling and dual-path retrieval. At each step, similarity computations over $|V|$ nodes and $|E_H|$ hyperedges with embedding dimension d yield $O((|V| + |E_H|)d)$ per step. Third, for policy optimization, GRPO processes N sampled trajectories of max length T, with gradient updates costing $O(NTd)$. Each component is computationally tractable and benefits from parallelization and localized retrieval over compact hypergraph subsets.

D. Dataset Details

We conduct experiments on six widely-used RAG benchmarks selected from the FlashRAG toolkit (Jin et al., 2025b), covering both single-hop and multi-hop question answering tasks:

- **2WikiMultiHopQA (2Wiki.)** (Ho et al., 2020): A multi-hop dataset requiring reasoning across two Wikipedia documents.
- **HotpotQA** (Yang et al., 2018): A challenging multi-hop QA dataset with sentence-level supporting facts and diverse question types.
- **Musique** (Trivedi et al., 2022): Multi-hop questions needing chains of inference, often involving three or more reasoning steps.
- **Natural Questions (NQ)** (Kwiatkowski et al., 2019): A large-scale single-hop QA dataset grounded in real Google search questions with Wikipedia passages.
- **PopQA** (Mallen et al., 2023): An open-domain QA dataset focused on popular culture questions sourced from Wikipedia.
- **TriviaQA** (Joshi et al., 2017): A large-scale dataset containing trivia-style questions with distantly supervised evidence documents.

To ensure consistency across datasets and maintain manageable training and evaluation workloads, we uniformly sample 5,120 instances per dataset for training and 128 instances for testing.

E. Baseline Details

Our experiments compare Graph-R1 with two groups of baselines using different backbone LLMs:

E.1. Baselines with `GPT-4o-mini`

- **NaiveGeneration (GPT-4o-mini)**: Zero-shot generation using GPT-4o-mini without retrieval, evaluating base model capacity.
- **StandardRAG (GPT-4o-mini)** (Lewis et al., 2020): Chunk-based RAG using GPT-4o-mini as the generator with retrieval over text chunks.
- **GraphRAG** (Edge et al., 2025): Graph-structured retrieval baseline that constructs entity graphs and performs one-shot retrieval with GPT-4o-mini for answer generation.
- **LightRAG** (Guo et al., 2025b): A lightweight GraphRAG variant that builds compact graphs for more efficient retrieval and GPT-4o-mini generation.
- **PathRAG** (Chen et al., 2026): Retrieval via path-based pruning on entity graphs, followed by GPT-4o-mini answer synthesis.
- **HippoRAG2** (Gutiérrez et al., 2025): Hierarchical path planner over knowledge graphs to improve retrieval efficiency, with GPT-4o-mini used for generation.
- **HyperGraphRAG** (Luo et al., 2025a): Constructs n-ary relational hypergraphs to support a single retrieval step, and uses GPT-4o-mini for answer writing.

E.2. Baselines with `Qwen2.5` (1.5B, 3B, 7B)

- **NaiveGeneration**: Direct generation by Qwen2.5 given the question prompt, without any retrieval, serving as a lower bound baseline.
- **StandardRAG** (Lewis et al., 2020): Classic chunk-based retrieval-augmented generation pipeline with semantic retriever and Qwen2.5 decoder.
- **SFT** (Zheng et al., 2024): Supervised fine-tuning of Qwen2.5 on QA pairs, without multi-turn reasoning or reinforcement optimization.

- **R1** (Shao et al., 2024): A GRPO-trained policy that generates final answers directly from question prompts without retrieval, optimized only on answer quality.
- **Search-R1** (Jin et al., 2025a): A multi-turn chunk-based retrieval method trained with GRPO, capable of iterative query refinement and retrieval under a unified policy.
- **R1-Searcher** (Song et al., 2025): A two-stage GRPO-based method with chunk-based retrieval: first using only format-level rewards to produce structured traces, then adding answer rewards.

F. Evaluation Details

Inspired by Luo et al. (2025a); Jin et al. (2025a), we evaluate model performance using four metrics:

(i) Exact Match (EM). EM measures whether the predicted answer exactly matches the ground truth. Let $\text{norm}(\cdot)$ denote the normalization function:

$$\text{EM} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left\{\text{norm}(y_i) = \text{norm}(y_i^\star)\right\}. \tag{31}$$

(ii) F1 Score. The F1 score measures the token-level overlap between the predicted answer y_i and the ground-truth answer $y^\star$ using the harmonic mean of precision and recall:

$$\text{F1} = \frac{1}{N}\sum_{i=1}^{N} \frac{2 \cdot |\text{tokens}(y_i) \cap \text{tokens}(y_i^\star)|}{|\text{tokens}(y_i)| + |\text{tokens}(y_i^\star)|}. \tag{32}$$

(iii) Retrieval Similarity (R-S). R-S assesses semantic similarity between retrieved $k_{\text{retr}}^{(i)}$ and gold knowledge $k_{\text{retr}}^{(i)}$. Let $\text{Enc}(\cdot)$ be the semantic embedding function:

$$\text{R-S} = \frac{1}{N}\sum_{i=1}^{N} \cos\left(\text{Enc}(k_{\text{retr}}^{(i)}), \text{Enc}(k_{\text{gold}}^{(i)})\right). \tag{33}$$

(iv) Generation Evaluation (G-E). G-E reflects generation quality. Let $s_{i,d}$ be GPT-4o-mini scores across 7 criteria (Figure 10):

$$\text{G-E} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{7}\sum_{d=1}^{7} s_{i,d}\right). \tag{34}$$

```
"comprehensiveness": (
        "comprehensiveness",
        "whether the thinking considers all important aspects and is thorough",
        """Scoring Guide
(0–10):
- 10: Extremely thorough, covering all relevant angles and considerations with depth.
- 8–9: Covers most key aspects clearly and thoughtfully; only minor omissions.
- 6–7: Covers some important aspects, but lacks depth or overlooks notable areas.
- 4–5: Touches on a few relevant points, but overall lacks substance or completeness.
- 1–3: Sparse or shallow treatment of the topic; misses most key aspects.
- 0: No comprehensiveness at all; completely superficial or irrelevant."""
    )
```

```
"knowledgeability": (
        "knowledgeability",
        "whether the thinking is rich in insightful, domain-relevant knowledge",
        """Scoring Guide
(0–10):
- 10: Demonstrates exceptional depth and insight with strong domain-specific knowledge.
- 8–9: Shows clear domain knowledge with good insight; mostly accurate and relevant.
- 6–7: Displays some understanding, but lacks depth or has notable gaps.
- 4–5: Limited knowledge shown; understanding is basic or somewhat flawed.
- 1–3: Poor grasp of relevant knowledge; superficial or mostly incorrect.
- 0: No evidence of meaningful knowledge."""
    )
```

```
"correctness": (
        "correctness",
        "whether the reasoning and answer are logically and factually correct",
        """Scoring Guide
(0–10):
- 10: Fully accurate and logically sound; no flaws in reasoning or facts.
- 8–9: Mostly correct with minor inaccuracies or small logical gaps.
- 6–7: Partially correct; some key flaws or inconsistencies present.
- 4–5: Noticeable incorrect reasoning or factual errors throughout.
- 1–3: Largely incorrect, misleading, or illogical.
- 0: Entirely wrong or nonsensical."""
    )
```

```
"relevance": (
        "relevance",
        "whether the reasoning and answer are highly relevant and helpful to the question",
        """Scoring Guide
(0–10):
- 10: Fully focused on the question; highly relevant and helpful.
- 8–9: Mostly on point; minor digressions but overall useful.
- 6–7: Generally relevant, but includes distractions or less helpful parts.
- 4–5: Limited relevance; much of the response is off-topic or unhelpful.
- 1–3: Barely related to the question or largely unhelpful.
- 0: Entirely irrelevant."""
    )
```

```
"diversity": (
        "diversity",
        "whether the reasoning is thought-provoking, offering varied or novel perspectives",
        """Scoring Guide
(0–10):
- 10: Exceptionally rich and original; demonstrates multiple fresh and thought-provoking ideas.
- 8–9: Contains a few novel angles or interesting perspectives.
- 6–7: Some variety, but generally safe or conventional.
- 4–5: Mostly standard thinking; minimal diversity.
- 1–3: Very predictable or monotonous.
- 0: No diversity or originality at all."""
    )
```

```
"logical_coherence": (
        "logical coherence",
        "whether the reasoning is internally consistent, clear, and well-structured",
        """Scoring Guide
(0–10):
- 10: Highly logical, clear, and easy to follow throughout.
- 8–9: Well-structured with minor lapses in flow or clarity.
- 6–7: Some structure and logic, but a few confusing or weakly connected parts.
- 4–5: Often disorganized or unclear; logic is hard to follow.
- 1–3: Poorly structured and incoherent.
- 0: Entirely illogical or unreadable."""
    ),
```

```
"factuality": (
        "factuality",
        "whether the reasoning and answer are based on accurate and verifiable facts",
        """Scoring Guide
(0–10):
- 10: All facts are accurate and verifiable.
- 8–9: Mostly accurate; only minor factual issues.
- 6–7: Contains some factual inaccuracies or unverified claims.
- 4–5: Several significant factual errors.
- 1–3: Mostly false or misleading.
- 0: Completely fabricated or factually wrong throughout."""
    )
```

Figure 10. Seven Dimensions for Generation Evaluation.

G. Implementation Details

As shown in Table 3, we summarize the detailed hyperparameter configurations used throughout our experiments, including model backbone, input limits, training configuration, and retrieval setup.

Table 3. Hyperparameter settings for baselines and Graph-R1.

Method	Backbone	Batch Size	Max Length	Top-K	Algo	Epochs
NaiveGeneration	Qwen2.5 / GPT-4o-mini	–	∞	N/A	–	–
StandardRAG	Qwen2.5 / GPT-4o-mini	–	∞	5 Chunks	–	–
GraphRAG	GPT-4o-mini	–	∞	60	–	–
LightRAG	GPT-4o-mini	–	∞	60	–	–
PathRAG	GPT-4o-mini	–	∞	60	–	–
HippoRAG2	GPT-4o-mini	–	∞	60	–	–
HyperGraphRAG	GPT-4o-mini	–	∞	60	–	–
SFT	Qwen2.5 (1.5B, 3B, 7B)	16	4096	N/A	LoRA	3
R1	Qwen2.5 (1.5B, 3B, 7B)	128	4096	N/A	GRPO	1
Search-R1	Qwen2.5 (1.5B, 3B, 7B)	128	4096	5 Chunks / Turn	GRPO	1
R1-Searcher	Qwen2.5 (1.5B, 3B, 7B)	128	4096	5 Chunks / Turn	GRPO	1
Graph-R1 (ours)	Qwen2.5 (1.5B, 3B, 7B)	128	4096	5 / Turn	GRPO	1

H. Case Study Analysis

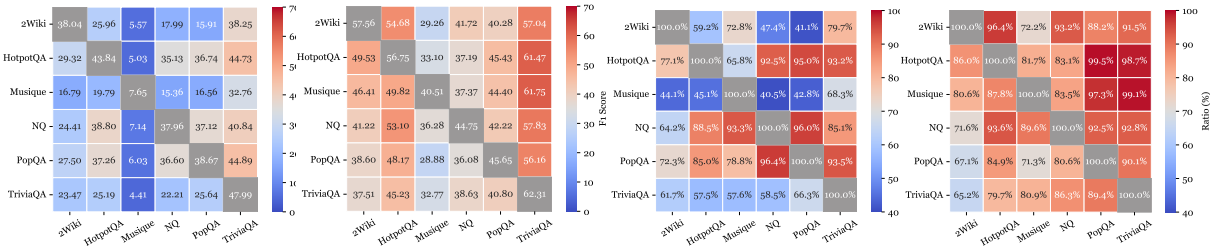
Table 4. Case study on generation quality under a query, comparing NaiveGeneration, StandardRAG, HyperGraphRAG based on GPT-4o-mini, with R1 (7B), Search-R1 (7B), and Graph-R1 (7B).

Query	When was the director of film Ingmar'S Inheritance born?								
Golden Answers	['18 November 1888']								
GPT-4o-mini	**NaiveGeneration**			**StandardRAG**			**HyperGraphRAG**		
Generation	<think> \nTo answer this question, I first need to identify who the director of "Ingmar\'s Inheritance" is. The film "Ingmar\'s Inheritance" is directed by a notable filmmaker. Researching the film, it turns out that it was directed by the Swedish director, "Ingmar Bergman." I need to check the birth date of Ingmar Bergman to provide the correct answer. Ingmar Bergman was born on July 14, 1918. Since the question is specifically asking about the birth of the director of the film rather than the film\'s release date or other details, I will now formulate the answer based on this information.\n </think> \n<answer> \n**Ingmar Bergman was born on July 14, 1918.**\n</answer>			<think> \nThe question asks for the birth year of the director of the film "Ingmar\'s Inheritance." However, the provided knowledge does not include information about the film "Ingmar\'s Inheritance" or its director..."\n </think> \n<answer> \n**The information necessary to answer the question about the director of "Ingmar\'s Inheritance" is not available in the provided knowledge. Therefore, I cannot provide a specific birth date for that director. However, Bille August, a noted director with connections to Ingmar Bergman, was born on November 9, 1948. It's unclear if he is associated with "Ingmar\'s Inheritance."** \n</answer>			<think> \nTo answer the question, I need to identify the director of the film "Ingmar\'s Inheritance." The information available mentions that "Ingmar\'s Inheritance" is a silent drama film directed by Gustaf Molander. The provided data does not specifically give a birthdate for Gustaf Molander. However, I find that the knowledge base also contains information about Ernst Ingmar Bergman, a prominent director, with dates of birth and death, but he is not related to "Ingmar\'s Inheritance." Since only Gustaf Molander is mentioned as the director of that specific film, and I don\'t have his birthdate recorded in the provided knowledge, I cannot provide an answer to this question.\n </think> \n<answer> \n**The birthdate of Gustaf Molander, the director of "Ingmar\'s Inheritance," is not provided in the knowledge base, so I cannot answer the question.** \n</answer>		
Evaluation Score	**F1**	**R-S**	**G-E**	**F1**	**R-S**	**G-E**	**F1**	**R-S**	**G-E**
	0.00	-	55.71	3.70	39.52	55.71	0.00	38.93	60.00
Qwen2.5-7B-Instruct	**R1 (7B)**			**Search-R1 (7B)**			**Graph-R1 (7B)**		
Generation	<think> \nTo answer when the director of the film "Ingmar\'s Inheritance" was born, I first need to find out who the director of this film is.\nThe film "Ingmar\'s Inheritance" was directed by Garth Davis. Now I need to find out when Garth Davis was born.\nGarth Davis was born on December 26, 1970. Therefore, the director of the film "Ingmar\'s Inheritance" was born on December 26, 1970.\n </think> \n<answer> \n**December 26, 1970**\n</answer>			<think> \nTo answer this question, I need to find the director of the film "Ingmar\'s Inheritance" and then determine their birth date.\n </think> \n<query> director of Ingmar\'s Inheritance </query> \n<knowledge> \n{"results": ["Weizsäcker studied at the University of Tübingen. While serving in the administration of King William II of Württemberg, he was raised to personal nobility on 24 February 1897...",...]}\n </knowledge> \n\n<think> \nFrom the information provided, Baldvin Zophoníasson is the director of "Ingmar\'s Inheritance" and he was born in 1978 in Akureyri.\n </think> \n<answer> \n**1978**\n</answer>			<think> \nTo answer this question, I need to find out the director of the film "Ingmar\'s Inheritance" and then determine the director\'s birth year. Let\'s start by finding the director of the film.\n </think> \n<query> director of Ingmar\'s Inheritance </query> \n<knowledge> \n{"results": [{"<knowledge>": "\\"Ernst Ingmar Bergman (14 July 1918 \\u2013 30 July 2007) was a Swedish director, writer, and producer who worked in film, television, theatre and radio.\\"", "<coherence>": 1.25}, {...}, ...]}\n </knowledge> \n\n<think> \nFrom the knowledge query, it seems that "Ingmar\'s Inheritance" was directed by Gustaf Molander. Now, I need to find out Gustaf Molander\'s birth year to answer the question.\n </think> \n<query> Gustaf Molander birth year </query> \n<knowledge> \n{"results": [{"<knowledge>": "\\"Gustaf Molander was born in Helsingfors (now Helsinki) in the Grand Duchy of Finland (in the Russian Empire), where his father was working at the Swedish Theatre.\\"", "<coherence>": 1.5}, {...}, ...]}\n </knowledge> \n\n<think> \nFrom the knowledge query, we found that Gustaf Molander was born on November 18, 1888. This is the director of the film "Ingmar\'s Inheritance".\n </think> \n<answer> \n**The director of the film "Ingmar\'s Inheritance", Gustaf Molander, was born on November 18, 1888.**\n</answer>		
Evaluation Score	**F1**	**R-S**	**G-E**	**F1**	**R-S**	**G-E**	**F1**	**R-S**	**G-E**
	0.00	-	41.43	0.00	40.02	64.29	37.50	45.83	88.57

As shown in Table 4, NaiveGeneration and R1 fail to provide the correct answer, and both StandardRAG and HyperGraphRAG also fall short despite using structured prompts. Search-R1, though RL-enhanced, shows limited improvement due to weak retrieval grounding. In contrast, Graph-R1 accurately identifies both the director and birthdate, achieves the highest G-E score (88.57), and demonstrates that RL is more effective with graph-based knowledge interaction.

I. Analysis of Graph-R1's Generalizability on O.O.D. Settings

As shown in Figure 11, to verify generalization, we conduct O.O.D. cross-validation for Search-R1 (3B) & Graph-R1 (3B) across six datasets: (a-b) F1 comparison, and (c-d) O.O.D.-to-I.I.D. ratios.



(a) Datasets *(b)* Parameters *(c)* Qwen3 *(d)* Algorithms

Figure 11. F1 comparison and performance ratios across six datasets under O.O.D. cross-validation.

F1 Performance Across Datasets. Figures 11a and 11b show that Graph-R1 outperforms Search-R1 on six datasets in O.O.D. validation, with notable gains on NQ and TriviaQA. Its multi-turn interaction with hypergraph retrieval ensures more stable performance under distribution shifts.

Robust Generalization Ability. Figures 11c and 11d show that Graph-R1 achieves higher O.O.D.-to-I.I.D. ratios than Search-R1, often above 85% and exceeding 90% in some cases, reflecting its strong robustness and cross-domain generalizability via end-to-end RL over knowledge hypergraph.

J. Details of Knowledge HyperGraph Construction

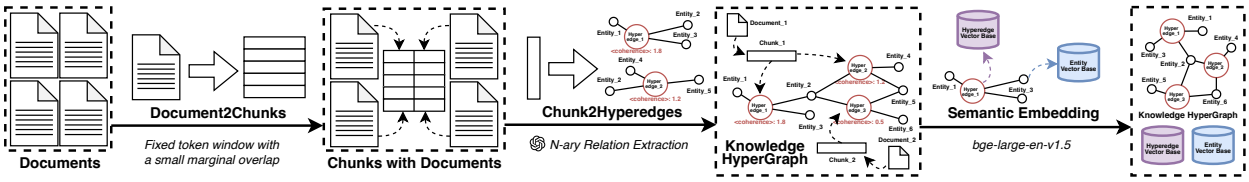


Figure 12. Knowledge HyperGraph Construction in Graph-R1.

As shown in Figure 12, Graph-R1 constructs the Knowledge HyperGraph through a streamlined yet expressive three-stage pipeline.

First, we segment raw documents into fixed-size textual units to maintain stable local semantics. Specifically, each document is divided using a 1200-token window with a 50-token overlap, ensuring that adjacent chunks retain sufficient contextual continuity. This windowing strategy preserves semantically meaningful boundaries while preventing the loss of cross-sentence information.

Second, each chunk is processed using a GPT-4o-mini–based n-ary relation extraction module. We adopt the carefully designed prompt shown in Appendix A.1, Figure 8, which guides the model to identify sets of entities and the higher-order relations binding them. Each extracted relational fact is represented as a hyperedge, and we assign a coherence score that reflects the internal semantic consistency and reliability of the extracted n-ary relation. This step transforms linear text into a structured hypergraph where nodes represent entities and hyperedges capture rich n-ary facts.

Third, to enable semantic retrieval and reasoning over the constructed hypergraph, we generate dense embeddings for both entities and hyperedges using the large-scale embedding model bge-large-en-v1.5. The resulting entity vector base and hyperedge vector base form the foundation of Graph-R1's retrieval module, allowing subsequent reasoning steps.

Compared with the original HyperGraphRAG implementation, we introduce several architectural and engineering optimizations, particularly in parallelization, prompt efficiency, storage layout, and vectorization. These improvements reduce token consumption and computational overhead, yielding a more efficient hypergraph construction pipeline, as evidenced in Table 6a of Section 5.4.

K. More Experimental Results on O.O.D. Settings

To rigorously assess the out-of-domain (O.O.D.) robustness and cross-domain generalization of Graph-R1, we evaluate the model under three distinct training regimes, each designed to isolate a different aspect of domain transferability. These settings provide a comprehensive picture of how retrieval structures (chunks vs. hypergraphs) behave when training and testing domains differ.

(i) In-Domain (I.I.D.) Training Setting. The first setting follows the same configuration as in Table 2. Here, each model is trained exclusively on the training split of its own dataset, meaning the supervision is fully aligned with the target domain. This represents the upper-bound scenario in which rich in-domain training data are available, allowing us to evaluate whether Graph-R1 can surpass Search-R1 even without requiring cross-domain generalization.

(ii) Single O.O.D. Training Setting. The second setting corresponds to the cross-dataset training paradigm illustrated in Figures 11. For each target dataset, the model is trained not on its own data, but on the training split of a different dataset. We use the results and compute the average performance. This setting stresses the ability to extract transferable information and tests whether hypergraph-based retrieval yields more robust inductive biases than chunk-based retrieval under dataset shift.

(iii) Combined-Dataset Training Setting. The third regime evaluates the model in a mixed-domain learning scenario. We first merge all six datasets into a single training pool and then uniformly downsample 1/6 of the combined data to ensure that the total training data volume remains identical to the I.I.D. setting. The resulting training set contains balanced signals drawn from diverse domains while strictly controlling for data quantity. This design allows us to examine whether Graph-R1 can exploit heterogeneous multi-domain relational structures to improve generalization, while ruling out gains attributable merely to increased training size.

K.1. Results and Analysis on Six Open-Domin Datasets

Table 5. Comparison between Search-R1 (3B) and Graph-R1 (3B) across six multi-hop and open-domain datasets under three training regimes. Best results are in **bold** and second in underline.

Method	2Wiki.		HotpotQA		Musique		NQ		PopQA		TriviaQA		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Qwen2.5-3B-Instruct *(Results When Trained on Its Own I.I.D. Dataset)*														
Search-R1 (3B)	31.25	38.04	38.28	43.84	3.91	7.65	24.22	37.96	33.59	38.67	40.62	47.99	28.65	35.69
Graph-R1 (3B)	**50.00**	**57.56**	**50.78**	**56.75**	**32.81**	**40.51**	**30.47**	**44.75**	37.50	**45.65**	**53.13**	**62.31**	**42.45**	**51.26**
Qwen2.5-3B-Instruct *(Average Performance When Trained on a Single O.O.D. Dataset)*														
Search-R1 (3B)	–	24.30	–	29.40	–	5.64	–	25.46	–	26.59	–	40.29	–	25.28
Graph-R1 (3B)	–	42.65	–	50.20	–	32.06	–	38.20	–	42.63	–	58.85	–	44.10
Qwen2.5-3B-Instruct *(Results When Trained on Combined Datasets)*														
Search-R1 (3B)	25.00	29.68	36.72	41.59	10.16	15.48	21.09	34.78	35.16	39.07	42.97	50.92	28.52	35.25
Graph-R1 (3B)	42.97	51.07	46.88	54.97	32.03	38.29	27.34	41.00	**38.28**	45.32	50.00	58.49	39.58	48.19

Table 5 reports the performance of Search-R1 and Graph-R1 under three training regimes (I.I.D., single O.O.D., and combined), covering six multi-hop and open-domain datasets.

First, Graph-R1 achieves consistent and clear improvements over Search-R1 across all datasets and all training settings. This confirms the advantage of hypergraph-structured retrieval, which provides more coherent and relationally grounded evidence than chunk-based retrieval, leading to stronger reasoning and higher EM/F1 scores.

Second, comparing the three regimes, the results follow a stable pattern: I.I.D. training performs best, combined-dataset training ranks second, and single O.O.D. training performs worst. This indicates that dataset-aligned supervision remains the most effective, but exposure to balanced multi-dataset data is still substantially better than training on a completely mismatched dataset.

K.2. Additional O.O.D. Results on Five Specialized Domains

Table 6. Domain-wise results on five specialized datasets. Results of GPT-4o-mini baselines are reported from the HyperGraphRAG paper (Luo et al., 2025a). Best results are in **bold** and second in underline.

Method	Medicine		Agriculture		CS		Legal		Mix		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
GPT-4o-mini												
NaiveGeneration	–	12.89	–	12.74	–	18.65	–	21.64	–	16.93	–	16.57
StandardRAG	–	27.90	–	27.43	–	28.93	–	37.34	–	43.20	–	32.96
GraphRAG	–	17.60	–	21.28	–	23.33	–	30.11	–	19.27	–	22.32
LightRAG	–	12.79	–	18.24	–	22.72	–	31.64	–	27.03	–	22.48
PathRAG	–	14.94	–	21.30	–	26.73	–	31.29	–	37.07	–	26.27
HippoRAG2	–	21.34	–	12.63	–	17.34	–	18.53	–	21.53	–	18.27
HyperGraphRAG	–	35.35	–	33.89	–	31.30	–	**43.81**	–	48.71	–	38.61
Qwen2.5-3B-Instruct *(Zero-Shot Results Without Domain Training)*												
Search-R1 (3B)	12.70	21.84	11.33	17.65	17.97	25.39	14.65	24.84	23.83	31.90	16.10	24.32
Graph-R1 (3B)	**24.22**	**37.18**	**25.98**	**37.39**	**27.54**	**37.45**	**21.48**	33.66	**38.48**	**50.60**	**27.54**	**39.26**

To further assess zero-shot generalization, Table 6 evaluates the model, trained only under the combined setting, on five specialized domain datasets introduced in HyperGraphRAG. For GPT-4o-mini baselines, we report the original results from the HyperGraphRAG paper.

Across Medicine, Agriculture, CS, Legal, and the Mixed domain, Graph-R1 (3B) consistently surpasses Search-R1 by sizeable margins in both EM and F1. Notably, Graph-R1 delivers strong improvements even in knowledge-intensive fields (e.g., Medicine and CS), where structured multi-entity relationships are crucial. These findings suggest that the relational inductive biases encoded by the hypergraph representation transfer effectively across specialized verticals not seen during training.

Collectively, these experiments demonstrate that Graph-R1 offers superior cross-domain robustness, strong transferability to unseen fields, and stable zero-shot performance, significantly outperforming chunk-based RAG baselines across both horizontal (general) and vertical (specialized) domains.