G-RAGent: Dynamic Reasoning on Hypergraphs for Retrieval-Augmented Language Models

Extended Abstract

Yaoyang Hou
Hangzhou Institute for Advanced Study, UCAS
Institute of Software, Chinese Academy of Sciences
Hangzhou, China
houyaoyang23@mails.ucas.ac.cn

Chen Zheng*
Institute of Software, Chinese Academy of Sciences
University of Chinese Academy of Sciences, Nanjing
Hangzhou Institute for Advanced Study, UCAS
Beijing, China
zhengchen@iscas.ac.cn

ABSTRACT

Retrieval-Augmented Generation (RAG) with knowledge graphs improves factual grounding but is limited by binary relation modeling and non-adaptive retrieval. We propose G-RAGent, a dynamic framework integrating hypergraph-based knowledge with an adaptive retrieval-reasoning mechanism. G-RAGent captures multi-entity facts through hyperedges and employs a ReAct-style loop where the LLM selectively accesses sub-hypergraphs and halts early when internal knowledge is sufficient. Implementation results show G-RAGent outperforms Graph-CoT by 21.5 absolute points (a 60.9% relative improvement) while reducing end-to-end latency by over 28%, highlighting its effectiveness and superiority on retrieval and reasoning.

KEYWORDS

Retrieval-Augmented Generation; Reasoning; Hypergraphs; Large Language Models; Neuro-Symbolic AI

ACM Reference Format:
Yaoyang Hou and Chen Zheng. 2026. G-RAGent: Dynamic Reasoning on Hypergraphs for Retrieval-Augmented Language Models: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/FUHD9537

1 INTRODUCTION

Large Language Models (LLMs) exhibit strong reasoning capabilities but struggle with timeliness and hallucinations due to reliance on static parametric knowledge. Retrieval-Augmented Generation (RAG)[8] mitigates this by grounding generation in external documents. However, conventional chunk-based RAG overlooks the interdependencies among entities and passages, often introducing redundant or irrelevant evidence[9]. While GraphRAG[1] introduces graph-structured knowledge to further enhance retrieval grounding, it is typically constrained by binary relations and coarse graph constructions that fail to capture the prevalent n-ary interactions across multiple entities in real-world knowledge bases[12]. Moreover, static, query-agnostic over-retrieval injects redundant information that interferes with the LLM's intrinsic reasoning[2, 3, 5].

To address these concerns, we propose G-RAGent, a dynamic framework that integrates hypergraph-based knowledge with an adaptive, query-aware retrieval controller. G-RAGent constructs and updates a hypergraph knowledge base that encodes n-ary facts as hyperedges, preserving semantic integrity. It drives inference through an iterative ReAct-style loop: the LLM decomposes complex questions, predicts semantic topics to retrieve relevant sub-hypergraphs instead of issuing a single retrieval on the original query. Each iteration follows three atomic steps: Thought to specify the local objective and preconditions, Action to issue graph retrieval instructions and execute them, and Observation to incorporate retrieved results into subsequent reasoning. An early-stopping mechanism halts retrieval when internal knowledge suffices, which reduces redundancy and latency while preserving the model's intrinsic reasoning. This design yields fine-grained, scalable retrieval over heterogeneous graphs and a principled balance between external evidence and parametric knowledge.

2 METHODOLOGY

2.1 Knowledge Hypergraph Construction

Traditional knowledge graphs fragment complex n-ary relations into binary edges, losing semantic completeness and high-quality relational contexts. G-RAGent employs a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ as the atomic knowledge unit. Here, $\mathcal{V}$ represents entities with textual attributes, and each hyperedge $e_j \in \mathcal{E}$ connects a subset of nodes representing a complete multi-entity fact (e.g., "Paper A authored by B and C at Conf D"). This structure explicitly models high-order dependencies that binary graphs fail to capture.

To enable efficient access, we define a topic mapping function $\phi : \mathcal{E} \to \mathcal{T}$ that categorizes hyperedges into semantic topics T. During retrieval, G-RAGent extracts a topic-specific sub-hypergraph $\mathcal{H}_T = (\mathcal{V}_T, \mathcal{E}_T)$ where $\mathcal{E}_T = \{e_j \in \mathcal{E} \mid \phi(e_j) = T\}, \mathcal{V}_T = \bigcup_{e_j \in \mathcal{E}_T} e_j$. To bridge the modality gap, these sub-hypergraphs are linearized into structured text: $\text{HyperEdge}_j : \langle r_j \rangle$ involves $[v_1.\text{name}, \ldots, v_k.\text{name}]$. This format preserves the n-ary integrity of facts for the LLM.

2.2 Dynamic Retrieval-Reasoning Balance

G-RAGent transforms the conventional static GraphRAG pipeline into an autonomous agent unifying retrieval and reasoning under a refined ReAct-style loop, allowing continuous adaptation between internal and external knowledge sources.. At iteration

*Corresponding author.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/FUHD9537

t, the agent acts based on state $S_t = (Q, H_t, K_t)$, where $H_t = \{A_1, O_1, A_2, O_2, \ldots, A_{t-1}, O_{t-1}\}$ is the history up to step t, consisting of all previous action–observation pairs. $K_t \in \{0, 1\}$ is a binary early-stopping flag; $K_t = 1$ indicates that reasoning can be terminated at step t and an answer can be output. The agent performs three atomic operations:

1) Thought: The LLM analyzes the history H_t to generate an instruction $I_t = (D_t, T_t, F_t)$. It determines the decision variable $D_t \in \{\texttt{RETRIEVE}, \texttt{REASON}, \texttt{FINISH}\}$. If external knowledge is required ($D_t = \texttt{RETRIEVE}$), it predicts the most relevant semantic topic T_t. If the answer can be derived using only the current context and internal knowledge without retrieval, it sets $D_t = \texttt{REASON}$. If the answer is ready to be output, it sets $D_t = \texttt{FINISH}$ and prepares the final answer F_t.

2) Action: Based on D_t, the agent executes specific operations α_t. For retrieval, it invokes a set of predefined primitives such as `RetrieveSubgraph(T)` to fetch $\mathcal{H}_{T_t}$, or fine-grained queries like `NodeFeature(id,attr)` and `NeighborCheck(id,R)`. If $D_t =$ `REASON`, it performs internal deduction without graph access. If $D_t =$ `FINISH`, Output(F_t) finalizes the answer.

3) Observation: The system executes the command generated in the Action step and returns an observation result o_t (linearized subgraph text, internal parametric reasoning or termination of the current process) is appended to the history H_{t+1}. This feedback loop allows the model to refine its understanding and reasoning iteratively.

Early-Stopping Mechanism: To maximize efficiency and minimize unnecessary retrieval overhead, we incorporate an early-stopping condition K_t into the framework. At any point during the Thought or Observation step, if the LLM determines that the accumulated context—comprising all prior observations and its internal knowledge—is already sufficient to derive a final or sub-task answer reliably and directly ($S_{t+1} = (Q,\ H_t \cup \{(\alpha_t, o_t)\},\ K_{t+1}), K_{t+1} = 1$), it skips any remaining scheduled steps and immediately outputs "Finish[Answer]", terminating either the current iteration or the entire reasoning process. This mechanism empowers the LLM to autonomously balance internal reasoning with selective external retrieval.

3 EXPERIMENTS

3.1 Experimental Setup

We implemented G-RAGent using Qwen3-8B [10] as the backbone LLM. We evaluated performance on three benchmarks: GRBench [7], HotpotQA [13] , and 2WikiMultiHopQA [6]. Baselines include three representative categories: LLM-only, Graph-based RAG (GraphRAG [1] and LightRAG [4]), and CoT-based (CoT [11] and Graph-CoT [7]). Evaluation metrics are ROUGE-L and GPT4Score.

3.2 Performance and Efficiency Evaluation

G-RAGent achieves the best results across all benchmarks. On GRBench, it reaches a GPT-4 Score of 56.8, exceeding Graph-CoT (35.3) by 21.5 absolute points (+60.9% relative). On HotpotQA (62.4) and 2WikiMultiHopQA (63.9), it improves over Graph-CoT by 9.8 and 8.1 points, respectively. These gains validate that hypergraph modeling and dynamic retrieval improve multi-hop reasoning accuracy.

Regarding efficiency, G-RAGent reduces average end-to-end latency on GRBench to 224.0s, compared to 312.1s for Graph-CoT, a 28% reduction. This largely stems from the early-stopping mechanism and dynamic control, which effectively prune unnecessary retrieval steps.

3.3 Internal Dynamics and Error Analysis

We conducted a post-hoc analysis, focusing on the distribution of actions, the prediction of topic subgraphs and error breakdown. The internal action distribution is 56.5% `RETRIEVE`, 33.2% `REASON`, and 10.3% `FINISH`, showing adaptive control rather than fixed retrieval. Topic prediction achieves 65.1% Top-1 precision and 72.5% Top-2 precision. Error analysis indicates failures are mainly due to knowledge base incompleteness (58%), followed by topic prediction errors (21%), over-aggressive early stopping (14%), and intrinsic query ambiguity (7%).

3.4 Ablation Study

Table 1: Ablation study of G-RAGent components on GRBench

Method	GRBench	
	GPT4Score	Average Latency
G-RAGent	**56.8**	224.03s
w/o Hypergraph	47.5	448.56s
w/o Topic-guided Retrieval	35.0	457.83s
w/o Early Stopping	50.3	311.62s
w/o ReAct Loop	13.5	**123.12s**
w/o Decision Variable ($\mathcal{D}_t$)	33.6	542.79s

Table 1 confirms each component's value: Hypergraphs ensure structural integrity; binary graphs lag in accuracy and speed. Topic-guided retrieval effectively filters noise. Early stopping reduces latency by ~39% maintaining accuracy. The ReAct loop is vital for complex reasoning, as one-shot retrieval fails (13.5). The dynamic decision variable $\mathcal{D}_t$ prevents over-retrieval; forcing constant retrieval hurts both accuracy and speed.

4 CONCLUSION

In this work, we introduced G-RAGent, a dynamic reasoning framework that transforms the LLM into an autonomous agent capable of performing topic-guided retrieval, and integrating reasoning and retrieval within a closed-loop ReAct process. A hypergraph-based knowledge construction module enables complete modeling of n-ary relational facts, ensuring semantic integrity and contextual relevance during retrieval. Extensive experiments show that G-RAGent consistently surpasses strong baselines across graph-centric and multi-hop QA tasks, demonstrating its effectiveness in balancing external knowledge utilization with internal reasoning and advancing the capability of RAG-enhanced LLMs.

ACKNOWLEDGMENTS

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (XDA0360202).

REFERENCES

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

[2] Zengyi Gao, Yukun Cao, Hairu Wang, Ao Ke, Yuan Feng, Xike Xie, and S Kevin Zhou. 2025. Frag: A flexible modular framework for retrieval-augmented generation based on knowledge graphs. *arXiv preprint arXiv:2501.09957* (2025).

[3] Kai Guo, Harry Shomer, Shenglai Zeng, Haoyu Han, Yu Wang, and Jiliang Tang. 2025. Empowering graphrag with knowledge filtering and integration. *arXiv preprint arXiv:2503.13804* (2025).

[4] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779* (2024).

[5] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems* 37 (2024), 132876–132907.

[6] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 6609–6625. https://www.aclweb.org/anthology/2020.coling-main.580

[7] Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, et al. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103* (2024).

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

[9] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3580–3599.

[10] Qwen Team. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[12] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. *arXiv preprint arXiv:1604.08642* (2016).

[13] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.