

Agent-Enhanced Heterogeneous Graph RAG for Academic Question Answering

Runsong Jia
University of Technology Sydney
Sydney, Australia
runsong.jia@student.uts.edu.au

Mengjia Wu
University of Technology Sydney
Sydney, Australia
mengjia.wu@uts.edu.au

Ying Ding
University of Texas at Austin
Austin, United States
ying.ding@ischool.utexas.edu

Jie Lu
University of Technology Sydney
Sydney, Australia
jie.lu@uts.edu.au

Yi Zhang*
University of Technology Sydney
Sydney, Australia
yi.zhang@uts.edu.au

Abstract

Academic question answering requires reasoning over heterogeneous scholarly graphs, where queries range from simple attribute lookups to multi-hop inference across author–paper–venue structures. Existing retrieval-augmented generation (RAG) systems struggle in this setting due to three limitations: (1) fixed retrieval strategies that do not adapt to varying query complexity, (2) the absence of sufficiency evaluation leading to incomplete or misaligned evidence, and (3) a lack of structured verification against graph facts. To address these issues, we propose an agentic heterogeneous graph RAG method that transforms the three core stages of the RAG pipeline into explicit agentic decision steps. A query-aware retrieval agent analyzes query type and selects an appropriate graph traversal strategy; a sufficiency-aware reranking agent assesses evidence completeness and adaptively expands the retrieved subgraph; and a graph-grounded verification agent checks entity, relation, and attribute correctness before finalizing the answer. Experiments on heterogeneous graphs constructed from OpenAlex and DBLP suggest that our method consistently outperforms strong LLM, graph-augmented RAG, and agent-based baselines.

CCS Concepts

• Information systems → Information retrieval; • Computing methodologies → Natural language processing.

Keywords

Agentic Retrieval-Augmented Generation; Heterogeneous Graph; Academic Question Answering

ACM Reference Format:

Runsong Jia, Mengjia Wu, Ying Ding, Jie Lu, and Yi Zhang. 2026. Agent-Enhanced Heterogeneous Graph RAG for Academic Question Answering. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*, April 13–17, 2026, Dubai, United Arab Emirates. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3774904.3792960>

*Corresponding author.



This work is licensed under a Creative Commons Attribution 4.0 International License. *WWW '26, Dubai, United Arab Emirates.*

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2307-0/2026/04
<https://doi.org/10.1145/3774904.3792960>

1 Introduction

Academic question answering (QA) requires combining natural-language understanding with structured knowledge encoded in heterogeneous scholarly graphs. RAG has become a standard paradigm for knowledge-intensive QA [10], and recent studies explore combining LLMs with knowledge graphs or heterogeneous graphs to enhance multi-hop reasoning and factual grounding [3].

Despite these advances, current graph-augmented RAG methods still exhibit three limitations. They rarely model query complexity, applying uniform traversal depth to both simple attribute queries and multi-hop reasoning queries [14]. They lack evidence sufficiency control, typically retrieving a one-shot subgraph without ensuring that it satisfies the entities, constraints, or aggregation signals implied by the query. Finally, they generally do not perform graph-grounded answer verification: once evidence is retrieved, generated answers are seldom checked against the graph for entity existence, relation validity, or attribute correctness. In contrast, text-only hallucination detection approaches such as SelfCheckGPT [12] and self-consistency methods [17] primarily rely on self-consistency rather than structured verification grounded in graph facts. This motivates our central question:

How can we develop an agentic RAG method for Academic QA in heterogeneous scholarly graphs?

We address this question by introducing an agent-enhanced heterogeneous graph RAG method in which the three stages of the RAG pipeline are explicitly agenticized. A query-aware retrieval agent analyzes the type of the question and produces a retrieval plan tailored to attribute, direct relation, aggregation, or multi-hop queries. A sufficiency-aware evidence control agent evaluates whether the retrieved subgraph adequately supports the query and selectively expands the graph neighborhood when necessary, rather than relying on fixed-depth traversal. A graph-grounded verification agent checks entity existence, relation validity, attribute correctness, and answer completeness before returning the final answer, inspired by recent agentic frameworks that decompose complex tasks into specialized modules [16, 18].

We evaluate the method on heterogeneous graphs constructed from OpenAlex and DBLP, comparing it with pure LLM baselines, representative graph-augmented RAG models, and agent-based retrieval methods. Across all evaluation settings, the agentic decomposition leads to consistent improvements in answer accuracy,

demonstrating the effectiveness of integrating agentic query-aware planning, adaptive retrieval, and graph-grounded verification.

Our main contributions are summarized as follows:

- (1) We propose an agentic method for the three core stages of the Academic QA RAG pipeline: retrieval planning, evidence control, and answer verification, where each stage is handled by a dedicated agent that explicitly leverages heterogeneous scholarly graph structure.
- (2) We develop a query and sufficiency-aware graph retrieval method that classifies questions into four academic query types and adaptively guides retrieval and reranking based on query structure.
- (3) We propose a graph-grounded verification method that validates answers against retrieved graph evidence, which helps improve the robustness of the overall method.

2 Related Works

Academic QA and Heterogeneous Scholarly Graphs. Scholarly data—comprising authors, papers, venues, and citation relations—naturally forms heterogeneous graphs. Prior research in academic recommendation, citation analysis, and structured QA has leveraged graph neural networks (GNNs) and knowledge-graph reasoning to encode such structures. Representative methods, including Heterogeneous Graph Transformer (HGT) [4], learn type-aware representations by aggregating information across nodes and edges. While this line of work has advanced structured reasoning over academic graphs, most methods focus on embedding learning or executing logical operators, and relatively fewer studies explore integrating LLMs as generators or incorporating explicit mechanisms that adapt retrieval to the query’s structure or complexity.

Retrieval-Augmented Generation and Graph-RAG. RAG has become a standard paradigm for knowledge-intensive QA, where retrieved evidence guides an LLM during generation [10]. Extending this idea, graph- and KG-augmented RAG variants incorporate graph traversal into retrieval pipelines, enabling the use of structured relations as evidence [6]. Recent systems such as G-Retriever [3] retrieves subgraphs to support textual graph understanding and integrate graph foundation models into RAG to improve retrieval over structured relations. These graph RAG systems highlight the benefits of combining structured knowledge with LLMs for knowledge-intensive QA.

Agentic LLMs and Answer Verification. Agentic LLM paradigms view language models as controllers capable of invoking tools, planning actions, or coordinating subtasks. ReAct [18] demonstrates interleaved reasoning and acting, while MAIN-RAG [1] proposes a multi-agent filtering mechanism for retrieval-augmented generation. Parallel to this, answer verification and reliability assessment have received growing attention, with methods such as self-consistency [17] and FActScore [13] offering fine-grained evaluation over generated text. Recent efforts have also explored agent-based or complexity-aware RAG methods [5] [9], which demonstrate the potential of using agents to control retrieval behavior. These approaches primarily operate in unstructured or text-centric settings, whereas our work focuses on verification against structured evidence from heterogeneous scholarly graphs.

3 Methodology

This section introduces the academic heterogeneous graph setting and describes each agent and its decision rules. As shown in Figure 1, we propose an agentic heterogeneous graph RAG method that decomposes Academic QA into three decision stages: (1) query-aware retrieval planning, (2) graph retrieval with sufficiency-aware evidence control, and (3) graph-grounded answer verification.

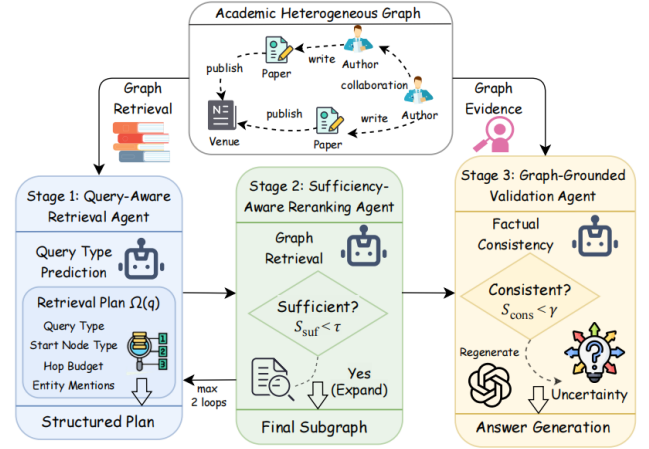


Figure 1: Overall architecture of our method.

3.1 Academic Heterogeneous Graph QA Setting

We consider a heterogeneous scholarly graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with three node types: Authors, Papers, and Venues. Author nodes store attributes such as name and organization, Paper nodes include title, abstract, keywords, citation counts, and year, and Venue nodes include venue name and rank. The graph includes two relation types: Paper–Author, representing authorship, and Paper–Venue, representing publication venues.

Given a natural-language query q , the goal is to generate an answer a grounded in a task-specific subgraph $\mathcal{G}_q \subseteq \mathcal{G}$. Motivated by prior analyses of complex Knowledge Base QA and complex logical queries over knowledge graphs [8], we classify queries into four structural categories: Attribute (0-hop), Direct Relation (1-hop), Aggregation (1–2 hops with numerical reasoning), and Multi-hop (2+ hop relational chains). Each type induces different requirements for start-node selection and traversal depth.

3.2 Query-Aware Retrieval Agent

The Retrieval Agent converts a natural-language query into a structured retrieval plan. Specifically, given query q , the agent predicts: (1) query type $t \in \{\text{Attr, Rel, Agg, Multi}\}$, (2) start-node type $s \in \{\text{Author, Paper, Venue}\}$, (3) hop budget h determined by t , (4) entity mentions $E(q)$, and (5) traversal plan π_t (node lookup, single-hop, multi-hop, aggregation).

We formalize the plan as: $\Omega(q) = (t, s, h, E(q), \pi_t)$. The agent uses an LLM classifier to infer t with distribution:

$$p(t | q) = \text{softmax}(f_\theta(q)) \quad (1)$$

The predicted query type is selected as:

$$t^* = \arg \max_t p(t \mid q) \quad (2)$$

after which (s, h, π_t) are determined through a fixed mapping.

In practice, f_θ is implemented as a prompted GPT-4-turbo classifier: we provide a system prompt that describes the academic graph schema and the four query types, together with a few in-context examples, and let the model directly output a description of $\Omega(q)$ without additional supervised fine-tuning. This step makes retrieval planning explicit and agentic.

3.3 Graph Retrieval and Sufficiency-Aware Reranking Agent

Given $\Omega(q)$, the system performs (1) entity lookup, (2) hop constrained traversal, and (3) sufficiency-aware evidence control. The start node is determined by embedding similarity:

$$v^* = \arg \max_{v \in V_s} \text{sim}(\phi(q), \phi(v)) \quad (3)$$

where $\phi(\cdot)$ is a text encoder. We instantiate ϕ using a Sentence-Transformers model (all-MiniLM-L6-v2), which produces 384-dimensional embeddings suitable for efficient semantic similarity matching between queries and node metadata. Traversal then yields the initial subgraph as $\mathcal{G}_q^{(0)} = \text{Traverse}(v^*, h, \pi_t)$.

The Reranking Agent evaluates whether $\mathcal{G}_q^{(i)}$ is adequate to answer q through a sufficiency score:

$$S_{\text{suf}}(q, \mathcal{G}_q^{(i)}) = \alpha \text{Cov}(q, \mathcal{G}_q^{(i)}) + (1 - \alpha) \text{Rel}(q, \mathcal{G}_q^{(i)}) \quad (4)$$

where Cov measures entity/attribute coverage and Rel measures semantic relevance. Concretely, $\text{Cov}(q, \mathcal{G}_q^{(i)})$ is computed as the fraction of entity mentions in $E(q)$ that can be matched to nodes in $\mathcal{G}_q^{(i)}$ via string and embedding-based matching, while $\text{Rel}(q, \mathcal{G}_q^{(i)})$ averages the embedding similarity between the query and textual descriptions of nodes in the subgraph. If the score is below threshold ($S_{\text{suf}} < \tau$), the agent triggers controlled expansion:

$$\mathcal{G}_q^{(i+1)} = \text{Expand}(\mathcal{G}_q^{(i)}, \pi_t) \quad (5)$$

with at most two expansions. In our experiments, we cap expansion at two steps to control latency and keep the retrieved subgraph compact, introducing explicit evidence control into the RAG pipeline.

3.4 Answer Generation and Graph-Grounded Validation Agent

The retrieved subgraph $\mathcal{G}_q$ is linearized and supplied to an LLM to generate answer a . The Validation Agent then checks whether the answer aligns with graph facts with factual consistency score defined as follows:

$$S_{\text{cons}}(a, \mathcal{G}_q) = \beta_1 \text{EntAlign}(a, \mathcal{G}_q) + \beta_2 \text{RelAlign}(a, \mathcal{G}_q) + \beta_3 \text{AttrMatch}(a, \mathcal{G}_q) \quad (6)$$

where the three components evaluate entity existence, relation validity, and attribute correctness.

Answer entities, relations, and attribute claims are extracted from a using lightweight LLM-based tagging prompts. $\text{EntAlign}(a, \mathcal{G}_q)$ measures the fraction of entity mentions in the answer that can be linked to nodes in $\mathcal{G}_q$ via string matching and embedding similarity;

$\text{RelAlign}(a, \mathcal{G}_q)$ checks whether relations described in the answer (e.g., head–relation–tail triples) correspond to existing edges in the retrieved subgraph; and $\text{AttrMatch}(a, \mathcal{G}_q)$ verifies that numerical and categorical attributes in the answer (such as years, venues, or counts) agree with the corresponding node attributes.

If $S_{\text{cons}} < \gamma$, the agent attempts regeneration (up to two times) or outputs an uncertainty signal. We limit regeneration attempts to two to bound computation cost. This graph-grounded check helps detect unsupported or hallucinated content.

4 Experimental Setup

Datasets. We evaluate our method on heterogeneous academic graphs derived from OpenAlex and DBLP. Each graph contains three node types (Authors, Papers, Venues) and two relations (writes, published_in). OpenAlex includes 76,569 nodes and 105,290 edges, while DBLP contains 62,443 nodes and 79,697 edges. Following our query taxonomy, we generate 400 queries for each dataset (100 per type: attribute, direct-relation, aggregation, multi-hop), with ground-truth answers automatically extracted from graph facts.

Backbone LLM and Implementation Details. All agentic controllers and the final answer generator use GPT-4-turbo. Entity lookup is performed using sentence-transformers for semantic similarity matching. For each query, the system allows at most two retrieval expansions and at most two verification-based regenerations. We report three standard metrics: (1) Accuracy, which measures exact match with the gold answer; (2) F1, computed via entity-level precision and recall; and (3) Hit@1, which evaluates whether the top-ranked predicted answer is correct.

Baselines. We compare our method against three categories of baselines: (1) pure LLM models, including Qwen-2.5-7B and GPT-o3; (2) graph-augmented RAG methods, including Vanilla RAG, Graph-Prompter [11], GraphRAG [2], KGRAG [15], and GraphCoT [7]; and (3) agent-based or complexity-aware RAG methods, including AdaptiveRAG [5] (agentic, complexity-aware retrieval) and Agent-G [9] (agentic retrieval control). These baselines cover the major types of RAG, graph-RAG, and agent-based QA methods.

5 Results

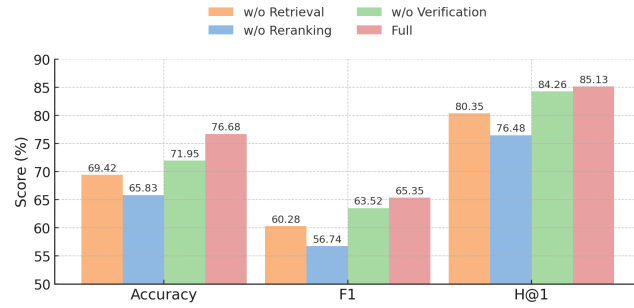
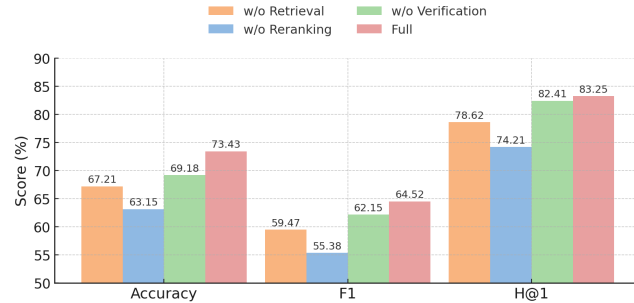
Overall Performance. Table 1 shows that our method achieves the strongest performance on both OpenAlex and DBLP across all metrics. Compared with graph-augmented approaches such as GraphCoT, GraphRAG, and KGRAG, as well as agent-based methods like AdaptiveRAG and Agent-G, our method delivers clear performance improvements, reaching 76.68% accuracy on OpenAlex and 73.43% accuracy on DBLP. The improvements are particularly notable in Hit@1, indicating that query-aware planning and sufficiency-aware retrieval lead to more reliable answer selection.

Ablation Study. Figure 2 and figure 3 report ablations for the three agents in our method. Removing the Retrieval Agent yields the largest performance drop, underscoring the importance of explicit query-type analysis and hop-budget planning. The Reranking Agent also contributes substantially by enabling adaptive evidence expansion when the initial subgraph is insufficient. Finally, the Verification Agent improves F1 and Hit@1 by checking entity, relation, and attribute correctness against the retrieved subgraph. Overall,

Table 1: Performance comparison of different models on OpenAlex and DBLP.

Model	OpenAlex			DBLP		
	Accuracy	F1	H@1	Accuracy	F1	H@1
Qwen 2.5 7B	48.63	43.46	50.26	45.93	39.82	51.52
GPT o3	53.58	46.45	51.65	51.65	45.59	50.67
Vanilla RAG	54.36	48.92	63.27	52.14	46.82	65.43
GraphPrompter	59.28	52.46	62.23	57.63	49.25	67.21
GraphRAG	61.87	57.39	74.86	60.27	53.32	72.59
KGRAG	62.49	55.28	68.51	62.35	54.71	73.48
GraphCoT	64.28	56.43	75.62	68.52	59.43	78.26
AdaptiveRAG	72.58	62.95	82.63	70.23	61.29	80.48
Agent-G	73.14	61.74	81.15	69.74	60.82	79.86
Ours	76.68	65.35	85.13	73.43	64.52	83.25

each agent contributes meaningfully to the method, and the full agentic pipeline provides the most robust performance.

**Figure 2: Ablation study on OpenAlex****Figure 3: Ablation study on DBLP**

6 Conclusion

In conclusion, we introduced an agentic heterogeneous graph RAG method for Academic QA that incorporates query-aware retrieval planning, sufficiency-aware evidence control, and graph-grounded answer verification. By explicitly agenticizing the three major steps of the RAG pipeline, our method more effectively aligns retrieval and verification with the structure of heterogeneous scholarly graphs. Experiments on OpenAlex and DBLP demonstrate consistent improvements over pure LLM baselines, graph-augmented RAG systems, and agent-based retrieval methods.

Acknowledgments

This work was supported by the Australian Commonwealth Scientific and Industrial Research Organization (CSIRO) in conjunction with the National Science Foundation (NSF) of the United States, under CSIRO-NSF #2303037.

References

- [1] Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, and Na Zou. 2025. Main-rag: Multi-agent filtering retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2607–2622.
- [2] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, 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).
- [3] 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.
- [4] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*. 2704–2710.
- [5] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 7029–7043.
- [6] Runsong Jia, Mengjia Wu, Ying Ding, Jie Lu, and Yi Zhang. 2025. HetGCoT: Heterogeneous Graph-Enhanced Chain-of-Thought LLM Reasoning for Academic Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. Association for Computational Linguistics, 15950–15963. doi:10.18653/v1/2025.findings-emnlp.864
- [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. In *Findings of the Association for Computational Linguistics ACL 2024*. 163–184.
- [8] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 11 (2022), 11196–11215.
- [9] Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vasilis N. Ioannidis, Huzefa Rangwala, and Christos Faloutsos. 2024. Agent-G: An Agentic Framework for Graph Retrieval Augmented Generation. (2024). <https://openreview.net/forum?id=g2C947jjjQ> Openreview Preprint.
- [10] 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.
- [11] Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024. Can we soft prompt llms for graph learning tasks?. In *Companion Proceedings of the ACM Web Conference 2024*. 481–484.
- [12] Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 9004–9017.
- [13] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 12076–12100.
- [14] 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.
- [15] Diego Sanmartin. 2024. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035* (2024).
- [16] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136* (2025).
- [17] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations* (2023).
- [18] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafraan, K. Narasimhan, and Y. Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.