Retrieval-Augmented Knowledge Completion for Graph-Constrained Reasoning

Wen Xu[1], Miao Fan[1,*], Wenjie Zhou[2], Chen Ma[3], Xiangzeng Liu[4], Haoyi Xiong[5]

[1]Beijing Institute of Graphic Communication, [2]Renmin University of China,
[3]City University of Hong Kong, [4]Xidian University, [5]Independent Scholar

***Abstract*—Graph-constrained reasoning (GCR) ensures factual correctness by enforcing structural constraints. However, it often leads to reasoning failures when the underlying knowledge graph (KG) is incomplete. To address this fundamental challenge, this paper presents a retrieval-augmented strategy to complete the KG for graph-constrained reasoning (abbr. as RA-GCR). Unlike traditional methods on GCR that tend to be limited by missing connections in KGs, RA-GCR continuously repairs the incomplete KGs with an LLM during inference. Specifically, it proceeds in two steps as follows. 1) Given the topic entities extracted from the user query, we construct a retrieval-augmented subgraph. Specifically, we synthesize this graph by merging explicit KG neighbors with implicit candidates retrieved from the global entity index. 2) On this subgraph, we perform graph-constrained reasoning. At each hop, the model selects the next entity by ranking candidates using a fused score. This score combines graph-based evidence with the LLM's semantic confidence derived from next-token predictions. We also employ vector retrieval to identify the target entity when explicit connections are missing, ensuring a faithful and continuous reasoning path. Extensive experiments on several KGQA benchmarks demonstrate that our approach not only achieves higher accuracy compared to SOTA methods, but also maintains stable performance in sparse graph settings, proving effective for reasoning over incomplete knowledge graphs.**

***Index Terms*—knowledge completion, graph reasoning, retrieval-augmented strategy**

I. Introduction

Large language models (LLMs) show strong capability in multi-hop reasoning and question answering [1], [2]. Knowledge graphs (KGs) provide explicit and verifiable relational facts [3], offering a natural substrate to ground and constrain LLM reasoning; accordingly, integrating LLMs with KGs is widely viewed as a way to enhance robustness and transparency in multi-hop reasoning [4]. Despite this, LLMs can still suffer from hallucinations, outdated knowledge, and limited interpretability [5], [6]. To address these issues, graph-constrained reasoning (GCR) constrains each step to valid entity–relation transitions in a knowledge graph, improving verifiability, traceable multi-hop paths, and faithfulness [7]–[9]. Follow-up work further enhances this paradigm by selecting paths or subgraphs to keep trajectories relevant and structurally consistent [10]–[13]. By contrast, retrieval-augmented generation (RAG) injects evidence as text without preserving

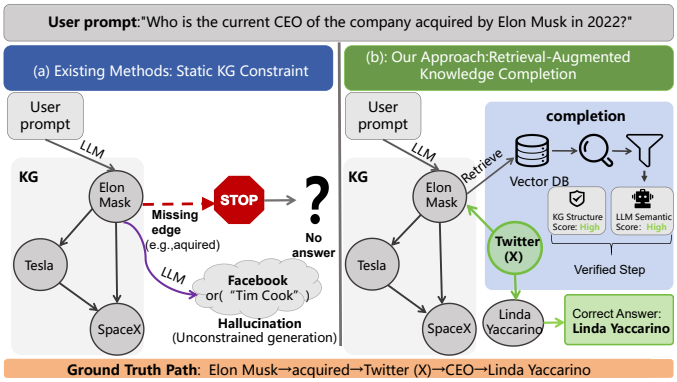

Fig. 1. An example illustrating the challenge of reasoning over incomplete knowledge graphs (KGs). (a) Existing static methods fail due to missing edges, leading to dead ends or hallucinations. (b) Our proposed method achieves robust reasoning by retrieval-augmented knowledge completion.

KG topology, making multi-hop control and hop-level verification harder [14]. However, as illustrated in Fig. 1, GCR is brittle on incomplete KGs: missing edges or sparse neighborhoods can block key transitions, leading to dead ends or uncontrolled generation. Existing fixes are limited—retrieval can be noisy, global completion is costly, and not query-specific [15], [16], and reflection methods mainly improve linguistic consistency with weak structural repair [17]. Therefore, enabling GCR to continue faithful reasoning despite missing links remains a core challenge. To this end, we introduce a retrieval-augmented strategy for graph-constrained reasoning (RA-GCR). By integrating semantic retrieval with structured decoding constraints, this paradigm avoids "dead end" failures on incomplete KGs, thereby ensuring robust reasoning. Inspired by the concept that missing structural links can be recovered through semantic correlation, we incorporate vector-based retrieval directly into the graph traversal process. This enables LLMs to reason on sparse graphs by bridging missing connections with semantically valid "soft edges" that lead to correct answers. In RA-GCR, we first encode all entities into a dense vector index to facilitate efficient semantic retrieval alongside the static KG structure. Unlike traditional GCR methods that restrict LLM output tokens strictly to existing explicit neighbors, RA-GCR expands the search space during inference. Then, we propose a candidate fusion mechanism

*Correspondence: fanmiao@bigc.edu.cn

that retrieves implicit entities relevant to the current question via vector similarity and fuses them with explicit neighbors from the KG, yielding a unified augmented subgraph. With that subgraph, we maintain reasoning continuity by leveraging LLMs' semantic knowledge to efficiently bridge structural gaps. Finally, we employ a dual-confidence scoring strategy to rank candidates from the subgraph, combining structural priors from the KG and semantic probabilities from the LLM to select the optimal next-hop entity. In this way, RA-GCR integrates the faithfulness of graph constraints with the flexible completion capabilities of semantic retrieval to achieve accurate, stable reasoning over incomplete KGs.

The main contributions of this work are as follows:

1) We propose retrieval-augmented graph-constrained reasoning (RA-GCR) to address the fundamental challenge posed by static constraints in incomplete KGs, allowing for robust reasoning via on-the-fly local augmentation.
2) We integrate the complementary strengths of explicit structural traversal and implicit semantic retrieval. By fusing symbolic neighbors with retrieved entities, we effectively bridge missing links during inference without the need for expensive offline global completion.
3) We conduct extensive experiments on two representative KGQA benchmarks under varying sparsity settings, demonstrating that RA-GCR not only significantly outperforms static GCR baselines, but also maintains stable performance even when a large portion of the graph structure is missing.

II. Related Work

Recent KG-augmented LLM approaches leverage KGs to ground generation and multi-hop inference. Graph-constrained frameworks, such as RoG, GCR, and ToG, enforce structural consistency directly during the decoding process [7]–[9], while path-based strategies focus on optimizing subgraph selection to filter noise [10]–[13]. Despite enhancing faithfulness, these methods fundamentally rely on the closed-set assumption of the static observed graph $\mathcal{G}_{obs}$, making them fragile in open-world settings, where missing edges often disrupt multi-hop paths and lead to early termination. To mitigate such errors, reflection mechanisms enable LLMs to refine outputs through iterative feedback [17]–[19]. However, reflection mainly operates at the semantic level and typically lacks an explicit mechanism for structural repair when the KG itself is incomplete. In contrast, traditional KG completion techniques focus on global structure refinement [15], [20]. However, a critical limitation remains: reflection methods perform well in semantic correction but cannot bridge missing links, while global completion methods are too costly for real-time inference. To address this, our RA-GCR strategy leverages retrieval to complete the graph structure during inference.

III. Problem Formulation

We formalize graph-constrained reasoning over incomplete knowledge graphs (KGs), where missing edges may break valid multi-hop paths. To address this, we model reasoning as graph-constrained traversal augmented with on-demand retrieval-based completion.

A. Preliminary

Knowledge Graph (KG). Formally, we denote a knowledge graph as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations, respectively. The set of factual knowledge is formalized as triples $\mathcal{T} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$.

Structural Incompleteness. In realistic scenarios, the observed graph $\mathcal{G}_{obs}$ is merely a subset of the ideal complete graph $\mathcal{G}_{ideal}$ ($\mathcal{G}_{obs} \subset \mathcal{G}_{ideal}$). This incompleteness manifests as missing edges, breaking valid inference chains, and leading to structural impasses.

Reasoning Path. A reasoning path is defined as a sequence of relations and entities originating from a start node e_0: $p = (e_0, r_1, e_1, \ldots, r_k, e_k)$. A path is considered valid if every transition (e_{i-1}, r_i, e_i) corresponds to an existing triple in $\mathcal{G}_{obs}$ or a retrieved implicit link.

KGQA Task. Given a natural language query q and an incomplete graph $\mathcal{G}_{obs}$, the goal of knowledge Graph question answering (KGQA) is to learn a mapping function $f : (q, \mathcal{G}_{obs}) \to \mathcal{E}$ that identifies the correct target entity a.

Topic Entity Linking. We denote the topic entity as $e_{topic} \in \mathcal{E}$, which is extracted from the query q. The reasoning path originates from e_{topic} to search for the target a.

Semantic Space. To bridge the structural gaps in $\mathcal{G}_{obs}$, we leverage an auxiliary semantic space. We assume a pre-trained encoder $\phi(\cdot)$ (e.g., BGE) that projects any textual query or entity into a d-dimensional dense vector $\mathbf{h} \in \mathbb{R}^d$. This embedding space serves as the foundation for constructing a dense retrieval index over all entities in $\mathcal{E}$.

Dense Retrieval. We define a generic retrieval function $\mathcal{M}_{ret}(x, k)$ to identify implicit neighbors based on semantic proximity. Specifically, it projects an input text x (e.g., a query or reasoning context) into the dense space and returns the top-k entities:

$$\mathcal{M}_{ret}(x, k) = \{e \in \mathcal{E} \mid \text{rank}(\cos(\phi(x), \phi(e))) \leq k\}. \quad (1)$$

This mechanism enables the model to traverse "soft edges" by matching semantic intent rather than strict symbol overlap.

B. Probabilistic Modeling

Our primary objective is to identify the optimal target entity $a^* \in \mathcal{E}$, maximizing the posterior probability given the query q and the observed graph $\mathcal{G}_{obs}$. To achieve this, we formulate the task as a multi-hop reasoning process, where the probability of an answer a is marginalized over all valid reasoning paths leading to it. Let $p = (e_0, \ldots, e_k)$ denote a reasoning path originating from the topic entity e_0 (identified in Sec. III-A). Assuming the Markov property, the probability of path p factorizes into the product of hop-wise transition probabilities:

$$P(p \mid q) = \prod_{t=0}^{k-1} P(e_{t+1} \mid e_t, q, \mathcal{G}_{obs}). \quad (2)$$

However, directly modeling these transitions on $\mathcal{G}_{obs}$ faces a twofold challenge: missing edges cause structural impasses,

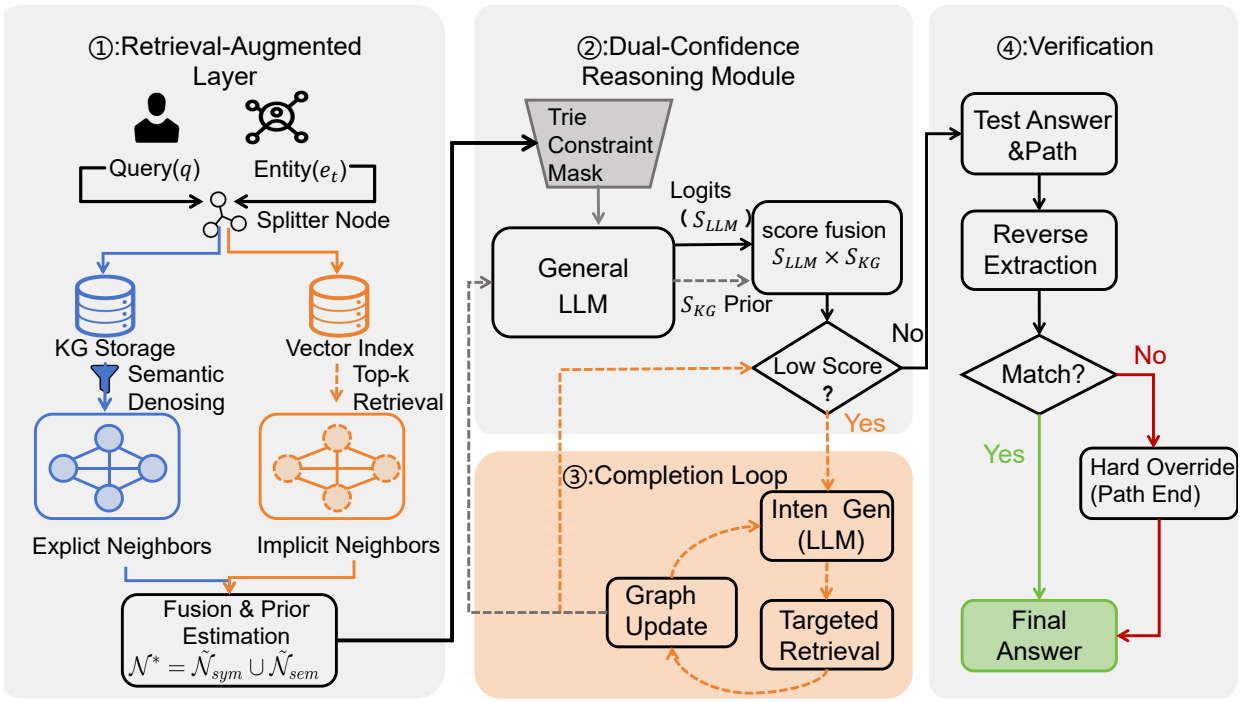

Fig. 2. Overview of the proposed framework. The process consists of four phases: (1) A Retrieval-Augmented Layer that fuses explicit (KG) and implicit (vector) neighbors; (2) A Dual-Confidence Reasoning Moudle leveraging combined LLM and KG priors with score detection; (3) A Completion Loop that dynamically updates the graph upon impasse; and (4) A Verification Module that ensures answer consistency via reverse extraction.

while irrelevant edges produce noise. To address this, we define a hybrid neighborhood $\mathcal{N}^*(e_t)$ by applying a semantic consistency check to filter both explicit and implicit candidates:

$$\mathcal{N}^*(e_t) = \{e \in \mathcal{N}_{obs} \cup \mathcal{M}_{ret} \mid \text{sim}(\mathbf{h}_{context}, \mathbf{h}_e) > \delta\}, \quad (3)$$

where $\mathbf{h}_e$ is the embedding of candidate entity e, $\mathbf{h}_{context}$ represents the query semantic vector, and δ serves as a confidence threshold. This formulation unifies completion (adding retrievable edges) and denoising (pruning irrelevant structural edges) into a mathematical operation, ensuring that the search space is both structurally augmented and semantically purified. Consequently, with this search space, we determine the optimal next hop by fusing structural priors with semantic confidence. We model the next-hop probability as a dual-confidence score:

$$P(e_{t+1} \mid e_t, q) \propto S_{KG}(e_t, e_{t+1}) \cdot S_{LLM}(e_{t+1} \mid q). \quad (4)$$

Here, S_{KG} encodes structural reliability (distinguishing facts from retrieval), while S_{LLM} measures the semantic confidence derived from the LLM.

IV. Proposed Solution

As illustrated in Fig. 2, RA-GCR consists of four main phases: 1) Retrieval-Augmented layer, 2)Dual-Confidence reasoning module, 3)Completion loop, 4)Verification. And our core design principles operate on two complementary dimensions:

1) We leverage the explicit structure of the KG to guide the LLM's generation strictly toward valid transitions. However, acknowledging that observed KGs ($\mathcal{G}_{obs}$) inevitably contain noise, we incorporate a semantic verification mechanism to prune irrelevant structural edges.
2) While we restrict the LLM to existing paths to prevent hallucinations, we recognize that missing edges can lead to reasoning impasses. Therefore, RA-GCR adopts an active completion strategy, treating "retrieval" as an on-the-fly mechanism to dynamically bridge these structural gaps.

In the following description, we merge the completion loop and the verification step (Fig 2) into a part termed impasse-aware adaptive completion.

A. Retrieval-Augmented Layer

To overcome the incompleteness of $\mathcal{G}_{obs}$, we introduce the retrieval-augmented layer that serves a dual purpose: semantic denoising and soft graph completion.

1) Hybrid Neighbor Retrieval: Given the query q, we first perform entity linking to identify the topic entity e_t, which serves as the anchor for hop-wise traversal. Then, starting from the current anchor entity e_t, we expand the search space via two parallel pathways:

Symbolic Pathway. We first retrieve the set of explicit 1-hop neighbors from the KG:

$$\mathcal{N}_{sym}(e_t) = \{e' \mid (e_t, r, e') \in \mathcal{G}_{obs}\}. \quad (5)$$

Semantic Pathway. Simultaneously, we leverage the vector index introduced in Sec III-A. We serialize the current entity e_t and the original query q into a single textual sequence, and encode it as $x_t = \phi([e_t; q])$. We then invoke the dense retrieval module to obtain top-k candidates:

$$\mathcal{N}_{sem}(e_t) = \mathcal{M}_{ret}(x_t, k), \tag{6}$$

where $\mathcal{M}_{ret}$ returns entities ranked by cosine similarity $\text{sim}(e_t, e') = \cos(\phi(x_t), \phi(e'))$.

2) Semantic Verification and Fusion: To generate a high-quality candidate set, we move beyond a naive union strategy by implementing a denoise-and-complete protocol, aligning with our design principles.

Step 1: Semantic Denoising. We verify the relevance of explicit neighbors. Let $\tilde{\mathcal{N}}_{sym}$ denote the validated set:

$$\tilde{\mathcal{N}}_{sym}(e_t) = \{e \in \mathcal{N}_{sym}(e_t) \mid \cos(\phi(x_t), \phi(e)) \geq \delta_{noise}\}. \tag{7}$$

Neighbors falling below δ_{noise} are discarded as structural noise.

Step 2: Implicit Completion. We identify high-confidence implicit nodes that are absent in the graph. Let $\tilde{\mathcal{N}}_{sem}$ denote the repair set:

$$\tilde{\mathcal{N}}_{sem}(e_t) = \{e \in \mathcal{N}_{sem}(e_t) \setminus \mathcal{N}_{sym}(e_t) \mid \cos(\phi(x_t), \phi(e)) > \delta_{complete}\}. \tag{8}$$

Unified Candidate Set. The final augmented neighborhood $\mathcal{N}^*$ is the union of these two refined sets: $\mathcal{N}^*(e_t) = \tilde{\mathcal{N}}_{sym}(e_t) \cup \tilde{\mathcal{N}}_{sem}(e_t)$. Accordingly, we assign a Structural Prior Score S_{KG} based on the source:

$$S_{KG}(e_{next}) = \begin{cases} 1.0 & \text{if } e_{next} \in \tilde{\mathcal{N}}_{sym}(e_t) \\ \text{sim}(e_t, e_{next}) & \text{if } e_{next} \in \tilde{\mathcal{N}}_{sem}(e_t) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

This ensures the system prioritizes validated facts (1.0) and relies on retrieved completions (< 1.0) only when they exhibit strong semantic relevance.

B. Dual-Confidence Constrained Reasoning

To robustly navigate $\mathcal{N}^*$, we employ a dual-confidence constrained decoding strategy, ensuring that generation aligns with both structural priors and semantic logic.

1) Trie-Guided Constraint Generation: We construct a prefix tree (Trie) [8] from the surface names of all entities in $\mathcal{N}^*(e_t)$. During generation, the LLM's vocabulary is masked by a vector $\mathbf{m} \in \mathbb{R}^V$:

$$m_i = \begin{cases} 0 & \text{if token } w_i \in \text{ValidNextTokens(Trie)} \\ -\infty & \text{otherwise} \end{cases} \tag{10}$$

2) Dual-Confidence Integration: We select the optimal path by fusing two sources of confidence:

- Structural Confidence (S_{KG}): Derived from Eq. 9.
- Semantic Confidence (S_{LLM}): derived from the LLM's token-level log-probabilities.

Reflection Prompt Template

Context: User asked "[Query]". Current entity: "[Current_Entity]".
Diagnosis: Neighbors [Neighbor_List] do not fit context.
Task: Generate a concise search intent for the missing link.
Output: [Search_Intent]

Fig. 3. Reflection prompt template used for dead-end handling.

$$S_{LLM}(e_{next}) = \frac{\exp(\text{Logit}(e_{next}))}{\sum_{e' \in \mathcal{N}^*(e_t)} \exp(\text{Logit}(e'))}. \tag{11}$$

Specifically, for an entity e with surface form tokens $(w_1, \ldots, w_n)$, we compute:

$$\text{Logit}(e) = \frac{1}{n} \sum_{i=1}^{n} \log p(w_i \mid \cdot), \tag{12}$$

where $\cdot$ denotes the LLM conditioning context at hop t. The step score is defined as $C_{step}(e_{next}) = S_{KG}(e_{next})^{\alpha} \cdot S_{LLM}(e_{next})$, where α is a calibration hyperparameter that controls the relative importance of structural confidence versus semantic confidence. A larger α biases the model toward explicit KG structure, while a smaller value allows more flexibility from semantic retrieval. which yields a bounded reliability score for ranking candidates at hop t. For retrieved implicit neighbors, relation types are not explicitly available in $\mathcal{G}_{obs}$. Instead, the semantic confidence S_{LLM} implicitly captures the relational compatibility between entities conditioned on the query and reasoning context, allowing relation-aware transitions without requiring explicit relation labels.

C. Impasse-Aware Adaptive Completion

1) Reasoning Impasse Detection: A reasoning state is identified as an impasse if the leading candidate's confidence falls below a tuned threshold τ:

$$\max_{e \in \mathcal{N}^*(e_t)} C_{step}(e) < \tau. \tag{13}$$

Such low confidence indicates that the current neighborhood lacks semantically relevant nodes, signaling a structural gap (missing edge) in $\mathcal{G}_{obs}$.

2) Adaptive Structural Augmentation: Upon detecting an impasse, the system prompts the LLM to generate a search intent I_{miss} describing the missing link as illustrated in Fig. 3. We encode I_{miss} to perform a targeted global retrieval. The top entities are dynamically injected into $\mathcal{N}^*$, and reasoning resumes.

3) Cycle-Consistency Verification: To prevent terminal hallucinations, we verify if the generated answer text A_{text} aligns with the reasoning path P_{final}. Let $\text{Tail}(P)$ denote the final entity of the path, $\mathcal{M}_{ext}$ be an entity extractor, and $\text{Norm}(\cdot)$ be a function. Faithfulness is satisfied iff:

$$\text{Norm}(\mathcal{M}_{ext}(A_{text})) = \text{Norm}(\text{Tail}(P_{final})). \tag{14}$$

Otherwise, the hard override is triggered, outputting the canonical name of $\text{Tail}(P_{final})$.

V. Experiments

In this section, we evaluate RA-GCR on incomplete KGs, covering effectiveness, robustness under sparsity, and ablation of key components.

A. Experimental Setup

Datasets. We evaluate RA-GCR on two widely-used KGQA benchmarks: WebQSP [21] and MetaQA [22]. WebQSP covers diverse relations with SPARQL-annotated queries, while MetaQA emphasizes multi-hop compositional reasoning (we report the 2-hop setting). Together, they test both real-world relational diversity and multi-hop reasoning.

Graph sparsification. To simulate incompleteness, we randomly remove a fraction of edges from the original KG to obtain the observed graph $\mathcal{G}_{obs}$. For reproducibility, we fix the random seed; for each sparsity ratio, all methods are evaluated on the same $\mathcal{G}_{obs}$.

Entity index for retrieval. For retrieval-based completion, we build a vector index over entities(names or definitions) from the full KG (before sparsification). During inference, RA-GCR still performs constrained traversal on $\mathcal{G}_{obs}$; the index is used only to propose candidates when local neighborhoods are insufficient, after which they are filtered or scored and injected into the local subgraph. This index is shared with other comparison methods.

Baselines. We compare RA-GCR against: 1) *LLM-only*: CoT prompting (LLaMA-3-8B-Instruct); 2) *Unconstrained Retrieval*: Graph-RAG; and 3) *Static GCR*: ToG and GCR, which operate under the closed-world assumption restricted to $\mathcal{G}_{obs}$.

Evaluation. We report Hits@1 (%) on both WebQSP and MetaQA. A question is counted as correct if the top-1 predicted answer entity matches any annotated correct answer provided by the dataset.

B. Implementation Details

We implement RA-GCR using LLaMA-3-8B-Instruct as the backbone model, performing step-wise constrained decoding via a Trie-based token mask. For retrieval-augmented completion, we adopt a pre-trained dense encoder $\phi(\cdot)$ (e.g., BGE-base) to embed entity names or short descriptions into a dense vector space, and construct a FAISS index over all entities from the full KG. At each reasoning step, we retrieve top-k semantic candidates and fuse them with explicit KG neighbors for constrained traversal. Unless otherwise specified, we set $k = 5$, $\delta_{\text{noise}} = 0.4$, $\delta_{\text{complete}} = 0.6$, $\tau = 0.25$, and $\alpha = 0.5$, and allow up to two iterations for adaptive completion. We use greedy decoding, and trigger the completion module only when the maximum step confidence falls below τ. Unless otherwise specified for particular experimental settings, all hyperparameters remain fixed across datasets, and the same retrieval index is shared across all compared methods for fair evaluation.

TABLE I
PERFORMANCE COMPARISON (HITS@1) UNDER **50% GRAPH SPARSITY**. RA-GCR OUTPERFORMS BASELINES BY DYNAMICALLY BRIDGING MISSING LINKS THAT CAUSE STATIC METHODS TO FAIL.

Category	Method	WebQSP	MetaQA
Parametric Knowledge	LLaMA-3 [23] (IO)	34.5	45.2
	CoT Prompting [2]	36.8	48.9
Unconstrained Retrieval	Graph-RAG [24]	48.2	58.6
Static Constraint	ToG [9]	49.5	61.4
	GCR [8]	<u>52.3</u>	<u>64.7</u>
Ours	**RA-GCR**	**65.2**	**78.5**

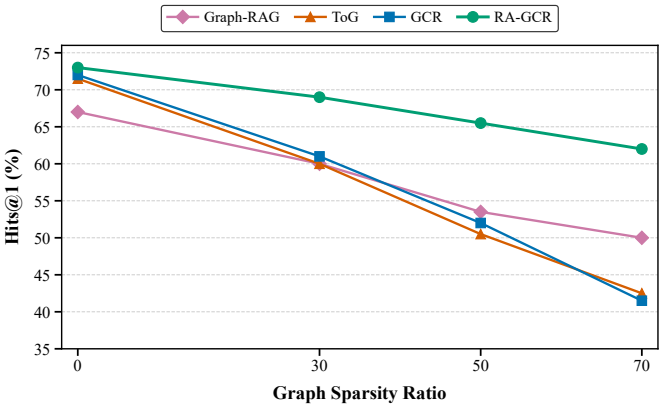

Fig. 4. Performance degradation under increasing graph sparsity on WebQSP. Our method exhibits significantly improved robustness compared to static graph-constrained baselines.

C. Main Results on Incomplete KGs

Table I presents Hits@1 under 50% graph sparsity. RA-GCR achieves the best performance among the compared baselines on both WebQSP and MetaQA. In particular, static graph-constrained baselines (e.g., ToG and GCR) degrade substantially under sparsification, as missing edges frequently induce structural dead ends that terminate multi-hop search. By contrast, RA-GCR improves over the strongest static baseline (GCR) by +12.9 on WebQSP and +13.8 on MetaQA, validating that the retrieval-Augmented completion can bridge missing links invisible to static decoding. Moreover, RA-GCR significantly outperforms Graph-RAG, indicating that the proposed dual-confidence scoring effectively suppresses retrieval noise compared with unconstrained augmentation.

D. Impact of Graph Sparsity

To further investigate the robustness of our strategy, we vary the graph sparsity ratio from 0% (Full Graph) to 70% (Highly Sparse) and record the performance trends on WebQSP. As illustrated in Fig. 4, static methods like GCR exhibit a substantial performance drop (declining by over 30%) as sparsity increases, confirming their brittleness. In sharp contrast, RA-GCR maintains a relatively stable performance curve, with only a minor 10% decrease even when 70% of the edges are missing. This resilience is attributed to our impasse-aware

adaptive completion module, which adaptively triggers targeted retrieval to patch structural gaps on demand, significantly reducing the dependency of reasoning performance on the static graph density.

E. Ablation Study

We conduct an ablation study to verify the contribution of each component in our retrieval-augmented strategy. As Table II, we create three variants: *w/o retrieval augmentation*, which degrades our model to a static GCR; *w/o semantic verification* (removing Eq. 7), using naive union for neighbor expansion; and *w/o adaptive augmentation* (removing Phase 3, detailed in Sec. IV-C), relying only on single-pass retrieval. The results in Table II show that removing the completion module causes the most severe drop (-12.9%), proving that the strategy of bridging gaps is the core driver of our performance. Additionally, removing the impasse-aware adaptive completion module results in a significant performance decline (-8.1%), indicating that impasse-driven intervention is crucial for resolving complex cases where single-pass retrieval fails.

TABLE II
ABLATION STUDY DEMONSTRATING THE CONTRIBUTION OF EACH MODULE.

Method Variant	**Hits@1**	Δ
RA-GCR (Full Strategy)	**65.2**	-
w/o Retrieval Augmentation	52.3	-12.9
w/o Semantic Verification	59.8	-5.4
w/o Adaptive Augmentation	57.1	-8.1

VI. Conclusion and Future Work

This paper addresses a important limitation of graph-constrained reasoning (GCR) on incomplete knowledge graphs, where missing edges break multi-hop reasoning paths. We propose RA-GCR, a retrieval-augmented completion framework that performs on-demand, query-conditioned augmentation during inference, enabling robust reasoning under structural sparsity. Experiments on WebQSP and MetaQA demonstrate consistent improvements over other baselines, particularly under high sparsity, while maintaining interpretable reasoning paths. Despite these gains, RA-GCR may introduce semantic drift due to retrieval noise, and relies on implicit relation modeling for soft edges, which can limit interpretability. Future work will extend evaluation to more heterogeneous datasets with structured sparsity patterns, conduct more systematic comparisons with alternative completion and retrieval frameworks, and perform more comprehensive sensitivity analysis of hyperparameters.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Municipal Universities of Beijing (No. 04190125001/015).

References

[1] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.
[2] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2021.
[4] W. X. Zhao *et al.*, "A survey of large language models," 2023.
[5] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.
[6] Y. Liu *et al.*, "Trustworthy llms: A survey and guideline for evaluating large language models' alignment," 2023.
[7] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable llm reasoning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.
[8] L. Luo *et al.*, "Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.
[9] J. Sun, C. Xu, L. Tang, *et al.*, "Think-on-graph: Deep and responsible reasoning of large language models on knowledge graph," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024, pp. 3868–3898.
[10] X. Tan *et al.*, "Paths-over-graph: Knowledge graph empowered large language model reasoning," in *Proceedings of The Web Conference (WWW)*, 2025, pp. 3505–3522.
[11] B. Jiang *et al.*, "Reasoning on efficient knowledge paths: knowledge graph guides large language model for domain question answering," in *Proceedings of the IEEE International Conference on Knowledge Graph (ICKG)*, 2024, pp. 142–149.
[12] H. Liu *et al.*, "Knowledge graph-enhanced large language models via path selection," in *Findings of the Association for Computational Linguistics (ACL Findings)*, 2024, pp. 6311–6321.
[13] Y. Wen, Z. Wang, and J. Sun, "Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
[14] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020, pp. 3929–3938.
[15] D. Xu *et al.*, "Multi-perspective improvement of knowledge graph completion with large language models," in *Proceedings of LREC-COLING*, 2024, pp. 11 956–11 968.
[16] N. Dong *et al.*, "Refining noisy knowledge graph with large language models," in *Proceedings of the GenAIK Workshop*, 2025, pp. 78–86.
[17] Y. Zhou *et al.*, "Reflection on knowledge graph for large language models reasoning," in *Findings of the Association for Computational Linguistics (ACL Findings)*, 2025, pp. 23 840–23 857.
[18] S. Yao *et al.*, "React: Synergizing reasoning and acting in language models," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
[19] A. Madaan *et al.*, "Self-refine: Iterative refinement with self-feedback," 2023.
[20] J. Youn and I. Tagkopoulos, "Kglm: Integrating knowledge graph structure in language models for link prediction," in *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, 2023, pp. 217–224.
[21] W.-t. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 201–206.
[22] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
[23] Llama Team, "The llama 3 herd of models," *arXiv:2407.21783*, 2024.
[24] Microsoft, "Graphrag: Unlocking llm discovery on narrative private data," https://github.com/microsoft/graphrag, 2024, accessed: 2026-01-21.