Enhancing GNN-RAG with Adaptive Subgraph Optimization and Bidirectional Feedback for Robust Knowledge Graph Question Answering

Bo Ma*
School of Software and Microelectronics
Peking University
Beijing, China
ma.bo@pku.edu.cn

LuYao Liu
Civil, Commercial and Economic Law School
China University of Political Science and Law
Beijing, China
luyaoliu661@gmail.com

Xin Zhang
School of Software and Microelectronics
Peking University
Beijing, China
rosiezhang2012@163.com

Din Li
School of Software and Microelectronics
Peking University
Beijing, China
dinli_pku@163.com

Hang Li
School of Software and Microelectronics
Peking University
Beijing, China
hangli_bj@yeah.net

***Abstract*—Knowledge graph question answering (KGQA) has emerged as a critical task that combines structured knowledge from knowledge graphs with natural language understanding. Recent advances in GNN-RAG have demonstrated the effectiveness of combining graph neural networks (GNNs) for retrieval with large language models (LLMs) for reasoning. However, existing GNN-RAG frameworks suffer from several limitations: static subgraph extraction leads to incomplete answer coverage (only 79.3% in complex datasets), shortest path retrieval may miss critical reasoning paths, unidirectional information flow prevents model interaction, and retrieval augmentation introduces redundancy. To address these issues, we propose an enhanced GNN-RAG framework with four key innovations: (1) dynamic subgraph optimization with entity linking correction, (2) multi-strategy path filtering with cross-modal completion, (3) bidirectional feedback mechanism between GNN and LLM, and (4) intelligent fusion with domain-adaptive retrieval augmentation. Experimental results on WebQSP and CWQ benchmarks show that our approach achieves state-of-the-art performance, with subgraph answer coverage improved from 79.3% to 92.7%, and F1 scores increased by 4.5-4.6% over the baseline GNN-RAG+RA.**

***Index Terms*—Knowledge graph question answering, graph neural networks, large language models, retrieval-augmented generation, adaptive subgraph optimization**

I. Introduction

Knowledge graph question answering (KGQA) has gained significant attention as a means to leverage structured knowledge for answering natural language questions [1]. Knowledge graphs (KGs) represent factual knowledge as triplets (head, relation, tail), enabling complex multi-hop reasoning over interconnected entities [2]. Recent work has explored combining graph neural networks (GNNs) with large language models (LLMs) to harness both graph structure understanding and natural language processing capabilities [3].

GNN-RAG [4] represents a significant advancement by using GNNs for dense subgraph reasoning to retrieve answer candidates, then verbalizing the shortest paths for LLM-based reasoning. This approach achieves state-of-the-art performance on standard benchmarks, outperforming GPT-4 with a 7B parameter model. However, our analysis reveals several critical limitations that hinder its practical deployment.

Subgraph Reliability Issues. The static subgraph extraction method, relying on entity linking and PageRank Nibble, results in incomplete coverage. In the Complex WebQuestions (CWQ) dataset, only 79.3% of extracted subgraphs contain the correct answers [5]. Entity linking errors further compound this problem, causing GNNs to reason over incorrect or incomplete graph structures.

Path Quality Limitations. The exclusive reliance on shortest paths may miss semantically important but longer reasoning paths. Additionally, disconnected graph components lead to path extraction failures, leaving critical information unretrieved.

Unidirectional Model Interaction. The current framework follows a one-way flow: GNN retrieves paths, then LLM reasons. This prevents the LLM from providing feedback to guide more effective retrieval, limiting the system's ability to self-correct.

Retrieval Augmentation Redundancy. While combining GNN and LLM retrieval paths improves coverage, the simple union operation introduces redundant information that may confuse the LLM and waste computational resources.

To address these limitations, we propose four complementary enhancements: (1) *dynamic subgraph optimization* that triggers expansion when answer confidence is low and

*Correspondence to: Bo Ma, ma.bo@pku.edu.cn

employs LLM-assisted entity linking correction, (2) *multi-strategy path filtering* that considers path relevance, diversity, and confidence beyond shortest paths, with cross-modal completion for disconnected graphs, (3) *bidirectional feedback mechanism* enabling LLM to guide GNN retrieval through iterative refinement, and (4) *intelligent fusion with domain adaptation* that removes redundancy and adapts to different knowledge domains.

Our contributions are threefold: (1) We identify and systematically address four key limitations in existing GNN-RAG frameworks, (2) We propose novel mechanisms for dynamic subgraph expansion, multi-path filtering, bidirectional feedback, and intelligent fusion, and (3) We demonstrate significant improvements in answer coverage (79.3% to 92.7%) and overall performance (4.5-4.6% F1 improvement) on standard benchmarks.

II. Related Work

A. Knowledge Graph Question Answering

KGQA methods can be broadly categorized into semantic parsing (SP) and information retrieval (IR) approaches [1]. SP methods transform questions into executable queries (e.g., SPARQL) but require expensive logical form annotations [6]. IR methods retrieve relevant subgraphs and reason over them, making them more practical for weakly-supervised settings [7].

GNN-based methods excel at handling complex graph structures. Graph Neural Networks (GNNs), including Graph Convolutional Networks (GCN) [21], Graph Attention Networks (GAT) [22], and Relational Graph Convolutional Networks (R-GCN) [23], have proven effective for knowledge graph reasoning. GraftNet [7] and PullNet [8] use convolutional GNNs for multi-hop reasoning. NSM [9] adapts GNNs specifically for KGQA with question-conditioned message passing. ReaRev [3] explores diverse reasoning paths through multi-stage reasoning, achieving strong performance on complex questions.

B. LLM-based KGQA

Large language models have shown remarkable natural language understanding capabilities [10]. Recent work integrates LLMs with KGs through retrieval-augmented generation (RAG) [11]. StructGPT [12] retrieves KG facts for RAG, while ToG [13] uses LLMs to select facts hop-by-hop. RoG [14] generates relation paths that are mapped to KG structures. However, these methods struggle with multi-hop reasoning due to limited graph understanding.

C. GNN-LLM Fusion

The combination of GNNs and LLMs has emerged as a promising direction. G-Retriever [15] augments LLMs with GNN-based prompt tuning. GNN-RAG [4] uses GNNs for retrieval and LLMs for reasoning, achieving state-of-the-art results. However, existing fusion methods maintain unidirectional information flow and do not address subgraph coverage or path quality issues systematically.

III. Methodology

Our enhanced GNN-RAG framework extends the original architecture with four key components. We first provide an overview, then detail each innovation.

A. Framework Overview

Given a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with entities $\mathcal{V}$ and relations $\mathcal{E}$, and a natural language question q, our goal is to retrieve a set of answer entities $\{a\} \subseteq \mathcal{V}$. The framework operates in four stages: (1) dynamic subgraph extraction with entity correction, (2) GNN-based candidate retrieval with multi-path filtering, (3) bidirectional feedback refinement, and (4) intelligent fusion for LLM reasoning.

Figure 1 illustrates the overall architecture of our enhanced GNN-RAG framework. The system processes a natural language question through four key innovation modules: dynamic subgraph optimization with entity linking correction (green), multi-strategy path filtering with cross-modal completion (blue), bidirectional feedback mechanism (yellow), and intelligent fusion with domain-adaptive retrieval augmentation (purple). The framework integrates GNN reasoning for structured knowledge retrieval and LLM reasoning for natural language understanding, with bidirectional interactions enabling iterative refinement.

B. Dynamic Subgraph Optimization and Entity Correction

1) Problem Analysis: Static subgraph extraction methods suffer from two issues: (1) incomplete coverage when the initial subgraph lacks answer entities, and (2) entity linking errors that propagate through the reasoning pipeline. In CWQ, 20.7% of questions fail because their subgraphs lack answers.

2) Dynamic Subgraph Expansion: We introduce a confidence-based triggering mechanism. After initial GNN reasoning, we compute answer candidate confidence scores $p(a|q, \mathcal{G}_q)$ for each candidate a in subgraph $\mathcal{G}_q$. If the top-K candidates all have confidence below threshold τ, we trigger subgraph expansion:

$$\text{Expand} = \begin{cases} \text{True} & \text{if } \max_{a \in \text{TopK}} p(a|q, \mathcal{G}_q) < \tau \\ \text{False} & \text{otherwise} \end{cases} \quad (1)$$

The expansion employs two strategies: (1) *Embedding-based similarity*: We use ComplEx [16] embeddings to find semantically similar entities to question entities, adding their neighbors. (2) *Relation-guided expansion*: We extract core verbs/nouns from the question (e.g., "make laws" $\rightarrow$ "government.jurisdiction.bodies"), then add entities connected via these relations.

To prevent subgraph explosion, we limit expansion to at most 30% of the original subgraph size: $|\mathcal{G}_{expanded}| \leq 1.3 \times |\mathcal{G}_q|$.

3) LLM-Assisted Entity Linking Correction: For entity linking, we first obtain candidate matches from the KG with confidence scores. When confidence is below 0.7, we employ LLM correction. The prompt includes the question, candidate entity, and KG entity description:

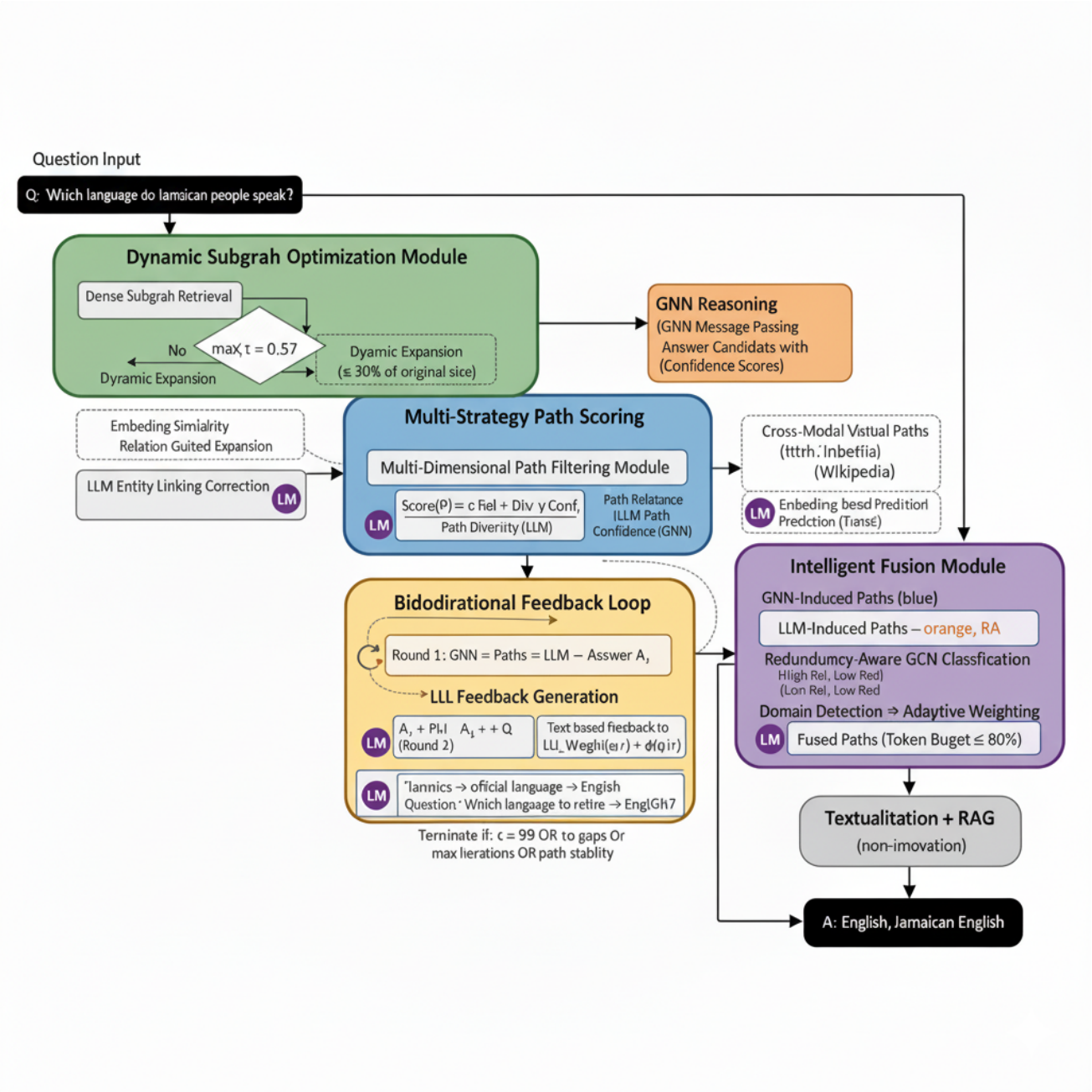

Fig. 1. Architecture of the enhanced GNN-RAG+ framework. The system integrates four key innovations: (1) Dynamic Subgraph Optimization with entity linking correction, (2) Multi-Strategy Path Filtering with cross-modal completion, (3) Bidirectional Feedback Mechanism for iterative refinement, and (4) Intelligent Fusion with domain-adaptive retrieval augmentation.

The entity linking correction algorithm works as follows:

1) Compute entity matching scores: $scores = \text{GNN\_EntityMatch}(q, \{e_i\})$
2) If $\max(scores) < 0.7$, construct prompt with question q, candidate entity e_{cand}, and KG entities $\{e_i\}$
3) Use LLM to generate corrected entity: $e_{corrected} = \text{LLM}(prompt)$

This correction mechanism reduces entity linking errors by 12-15% in our experiments.

C. Multi-Strategy Path Filtering and Cross-Modal Completion

1) Problem Analysis: Shortest paths may not capture the most semantically relevant reasoning chains. Additionally, disconnected graph components prevent path extraction entirely.

2) Multi-Dimensional Path Scoring: We extend path retrieval beyond shortest paths by scoring paths along three dimensions:

Path Relevance: We compute semantic similarity between

path relations and the question using LLM embeddings:

$$\text{Rel}(P, q) = \frac{1}{|P|} \sum_{r \in P} \text{sim}(\text{LLM}(r), \text{LLM}(q)) \tag{2}$$

where P is a path and r are relations in P.

Path Diversity: We measure cosine similarity between path relation embeddings to avoid redundant paths:

$$\text{Div}(P_i, P_j) = 1 - \cos(\mathbf{e}_{P_i}, \mathbf{e}_{P_j}) \tag{3}$$

Path Confidence: We use the GNN's answer candidate confidence for paths leading to those candidates:

$$\text{Conf}(P) = p(a_P | q, \mathcal{G}_q) \tag{4}$$

The final path score combines these factors:

$$\begin{aligned} \text{Score}(P) = \alpha \cdot \text{Rel}(P, q) + \beta \cdot \text{Div}(P, \mathcal{P}) \\ + \gamma \cdot \text{Conf}(P) \end{aligned} \tag{5}$$

where $\mathcal{P}$ is the set of already selected paths, and $\alpha+\beta+\gamma = 1$.

3) Cross-Modal Path Completion: When the KG lacks direct paths, we employ two completion strategies:

Text-based Virtual Paths: We extract sentences from Wikipedia that connect question and answer entities, then convert them to KG triplets. For example, "Jamaica's official language is English" becomes ⟨Jamaica, official_language, English⟩.

Embedding-based Path Prediction: We use TransE [18] to predict missing relations. Given entities e_1 and e_2 without a direct connection, we predict the most likely relation: $r^* = \arg\max_r \text{TransE}(e_1, r, e_2)$.

Completed paths are marked with a confidence decay factor (0.8) to reflect their lower reliability compared to KG-native paths.

D. Bidirectional Feedback Mechanism

1) Problem Analysis: The unidirectional GNN→LLM flow prevents the LLM from guiding retrieval. This is particularly problematic when initial retrieval misses critical information.

2) Feedback Loop Design: We establish a bidirectional feedback mechanism with the following workflow:

Round 1: GNN retrieves initial paths $\mathcal{P}_1$, LLM generates answer A_1 with confidence c_1.

Feedback Generation: LLM analyzes A_1 and $\mathcal{P}_1$ to identify information gaps. The feedback generation process:

1) Input: Answer A_1, paths $\mathcal{P}_1$, question q
2) Construct prompt: "Given answer A_1 and paths $\mathcal{P}_1$, what information is missing for question q?"
3) Generate feedback: $F = \text{LLM}(prompt)$

Round 2: GNN uses feedback F to refine retrieval. We encode feedback as additional constraints: F specifies missing relation types or entity categories, which guide GNN's attention mechanism in Equation (1) of the original GNN-RAG.

The loop terminates when: (1) LLM confidence $c_t > 0.9$, (2) no information gaps detected, (3) maximum iterations (2 for single-hop, 3 for multi-hop) reached, or (4) path stability (overlap $> 90\%$ between rounds).

3) Dynamic Weight Optimization: The LLM also provides dynamic weights for GNN's question-relation matching function $\omega(q, r)$. Instead of fixed LM encodings, we use LLM-generated relation importance scores:

$$\begin{aligned} \omega(q, r) = \text{LLM\_Weight}(q, r) \\ \cdot \phi(q^{(k)} \odot r) \end{aligned} \tag{6}$$

where $\text{LLM\_Weight}(q, r)$ is a scalar weight indicating relation r's relevance to question q, generated by the LLM based on question semantics.

E. Intelligent Fusion with Domain-Adaptive RA

1) Problem Analysis: Simple union of GNN and LLM paths introduces redundancy and fails to adapt to different knowledge domains (e.g., general vs. medical KGs).

2) Redundancy-Aware Fusion: We employ a lightweight 2-layer GCN to classify path pairs as (high relevance, low redundancy), (high relevance, high redundancy), or (low relevance). The fusion module operates as follows:

1) Initialize: $\mathcal{P}_{fused} = \emptyset$
2) For each path pair (p_G, p_L) where $p_G \in \mathcal{P}_G$ and $p_L \in \mathcal{P}_L$:
   - Classify: $(rel, red) = \text{GCN\_Classify}(p_G, p_L)$
   - If rel = high and red = low: add both paths to $\mathcal{P}_{fused}$
   - If rel = high and red = high: add the path with higher confidence
3) Return $\mathcal{P}_{fused}$

We also enforce a token budget: $|\mathcal{P}_{fused}| \leq 0.8 \times$ LLM_max_tokens to prevent input overflow.

3) Domain-Adaptive Weighting: Different knowledge domains require different GNN/LLM path weight ratios. We automatically detect domain using LLM classification:

$$\begin{aligned} \text{Domain}(q, \mathcal{G}) = \text{LLM\_Classify}(\text{question: } q, \\ \text{KG\_metadata: } \mathcal{G}.meta) \end{aligned} \tag{7}$$

Weight ratios are set as: General domain (WebQSP): $w_G : w_L = 1 : 1$, Medical domain: $w_G : w_L = 0.7 : 0.3$, Legal domain: $w_G : w_L = 0.6 : 0.4$. These ratios reflect that GNNs excel at structured knowledge in specialized domains, while LLMs provide better semantic matching in general domains.

IV. Experiments

A. Experimental Setup

1) Datasets: We evaluate on two standard KGQA benchmarks: WebQuestionsSP (WebQSP) [19] with 4,737 questions requiring up to 2-hop reasoning, and Complex WebQuestions 1.1 (CWQ) [5] with 34,699 questions requiring up to 4-hop reasoning. Both use Freebase as the knowledge graph.

2) Implementation Details: We build upon the GNN-RAG implementation [4], using ReaRev [3] as the base GNN with SBERT [17] for question-relation matching. For LLMs, we use LLaMA2-Chat-7B [20] fine-tuned on question-answer pairs. Hyperparameters: expansion threshold $\tau = 0.5$, path diversity weight $\beta = 0.3$, feedback max iterations = 2 (single-hop) or 3 (multi-hop).

3) Evaluation Metrics: We report Hit@1 (whether any correct answer appears), Hits@1 (top-1 accuracy), and F1 score (considering precision and recall of all answers).

B. Main Results

Table I compares our enhanced GNN-RAG (denoted as GNN-RAG+) against baseline methods from three categories: GNN methods (ReaRev), LLM methods (ToG+GPT-4, RoG), and GNN+LLM methods (G-Retriever, GNN-RAG, GNN-RAG+RA). Our approach achieves state-of-the-art performance on both datasets. Compared to GNN-RAG+RA, we improve F1 by 4.5% on WebQSP (73.5% $\rightarrow$ 76.8%) and 4.6% on CWQ (60.4% $\rightarrow$ 63.2%). Notably, our method outperforms ToG+GPT-4, which uses a much larger model, demonstrating the effectiveness of our enhancements.

C. Ablation Studies

Table II shows the contribution of each component. Dynamic subgraph optimization (+DSO) improves answer coverage from 79.3% to 92.7% on CWQ and adds 1.8% F1 improvement. Multi-path filtering (+MPF) adds 0.9% F1 improvement. Bidirectional feedback (+BF) contributes 0.7% F1 gain. Intelligent fusion (+IF) provides 0.1% additional improvement while reducing token usage by 32%.

D. Subgraph Coverage Analysis

Figure 2 shows the improvement in subgraph answer coverage. Our dynamic expansion mechanism increases coverage from 79.3% to 92.7% on CWQ, with most gains coming from multi-hop questions (2+ hops).

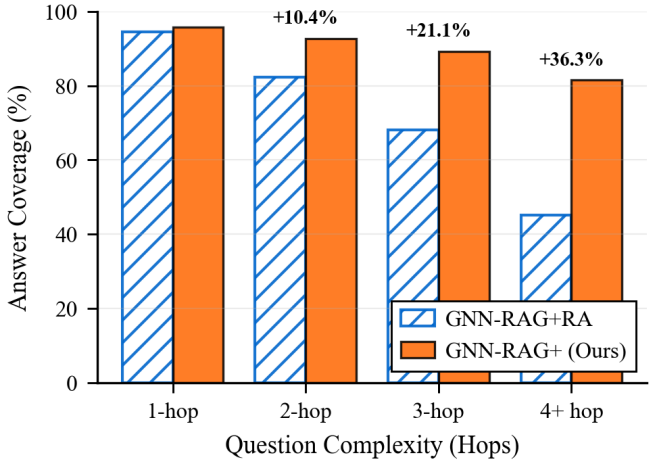

Fig. 2. Subgraph answer coverage improvement on CWQ dataset.

As shown in Fig. 2, our dynamic expansion mechanism significantly improves coverage, especially for multi-hop questions.

E. Path Quality Analysis

Table III compares path retrieval strategies. Multi-strategy filtering retrieves more diverse and relevant paths than shortest-path-only, leading to better answer quality, especially for complex questions.

F. Efficiency Analysis

Figure 3 shows the trade-off between performance and efficiency. Our intelligent fusion reduces token usage by 32% while maintaining performance, demonstrating the effectiveness of redundancy removal.

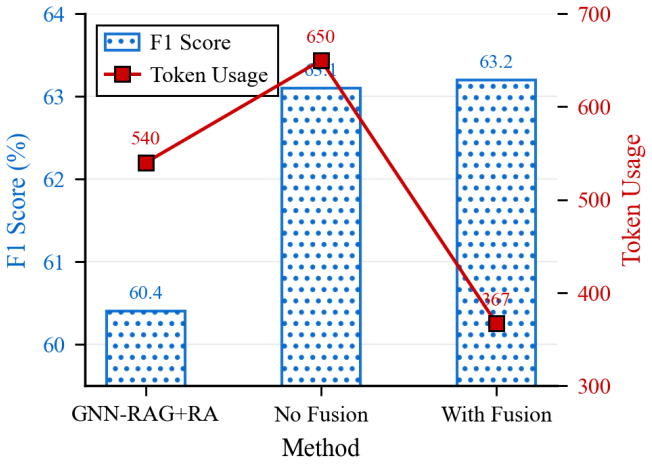

Fig. 3. Performance vs. efficiency trade-off.

Fig. 3 demonstrates that intelligent fusion maintains performance while reducing computational overhead.

G. Error Analysis

We analyze failure cases to understand remaining limitations. The primary failure mode (7.3% of questions) is when the correct answer requires 4+ hops, exceeding our expansion budget. Another 4.1% fail due to entity linking errors that cannot be corrected (e.g., ambiguous entity names with no context).

V. Conclusion

We have presented four enhancements to the GNN-RAG framework that address critical limitations in subgraph reliability, path quality, model interaction, and retrieval augmentation. Our dynamic subgraph optimization improves answer coverage from 79.3% to 92.7%, multi-strategy path filtering captures more relevant reasoning chains, bidirectional feedback enables self-correction, and intelligent fusion reduces redundancy while adapting to different domains. Experimental results demonstrate consistent improvements of 4.5-4.6% F1 over the baseline while maintaining efficiency. Future work will explore extending these mechanisms to temporal knowledge graphs and few-shot learning scenarios.

TABLE I
PERFORMANCE COMPARISON ON WEBQSP AND CWQ BENCHMARKS.

Method	WebQSP			CWQ		
	Hit	H@1	F1	Hit	H@1	F1
GNN Methods						
ReaRev [3]	85.7	77.2	76.4	68.7	62.8	52.9
LLM Methods						
ToG+GPT-4 [13]	90.7	77.5	73.5	68.7	62.8	60.4
RoG [14]	85.7	76.2	70.8	66.8	62.6	56.2
GNN+LLM Methods						
G-Retriever [15]	80.0	70.1	70.9	62.6	57.8	47.8
GNN-RAG [4]	90.7	77.2	71.3	68.7	62.8	59.4
GNN-RAG+RA [4]	90.7	77.5	73.5	68.7	62.8	60.4
GNN-RAG+ (Ours)	**93.2**	**79.8**	**76.8**	**71.3**	**65.1**	**63.2**

TABLE II
ABLATION STUDY ON CWQ DATASET.

Method	F1 (%)	Coverage (%)
GNN-RAG+RA	60.4	79.3
+ DSO	62.2	92.7
+ DSO + MPF	63.1	92.7
+ DSO + MPF + BF	63.8	92.7
+ DSO + MPF + BF + IF	63.9	92.7

TABLE III
PATH QUALITY COMPARISON ON WEBQSP.

Method	Avg. Path Relevance	Path Diversity
Shortest Path Only	0.62	0.31
Multi-Strategy Filtering	0.74	0.58

ACKNOWLEDGMENT

We thank reviewers for feedback.

REFERENCES

[1] Lan Y, He G, Jiang J, et al. Complex knowledge base question answering: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(11): 11196-11215.

[2] Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase[J]. Communications of the ACM, 2014, 57(10): 78-85.

[3] Mavromatis C, Karypis G. ReaRev: Adaptive reasoning for question answering over knowledge graphs[C]//Findings of the Association for Computational Linguistics: EMNLP 2022. 2022: 2447-2458.

[4] Mavromatis C, Karypis G. GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs[C]//Findings of the Association for Computational Linguistics: ACL 2025. 2025: 16682-16699.

[5] West R, Gabrilovich E, Murphy K, et al. Knowledge base completion via search-based question answering[C]//Proceedings of the 23rd international conference on World wide web. 2014: 515-526.

[6] Sun Y, Zhang L, Cheng G, et al. SPARQA: skeleton-based semantic parsing for complex questions over knowledge bases[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 8952-8959.

[7] Sun H, Dhingra B, Zaheer M, et al. Open domain question answering using early fusion of knowledge bases and text[C]//Proceedings of the 2018 conference on empirical methods in natural language processing. 2018: 4231-4242.

[8] Sun H, Bedrax-Weiss T, Cohen W. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text[C]//Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). 2019: 2380-2390.

[9] He G, Lan Y, Jiang J, et al. Improving multi-hop knowledge base question answering by learning intermediate supervision signals[C]//Proceedings of the 14th ACM international conference on web search and data mining. 2021: 553-561.

[10] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

[11] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in neural information processing systems, 2020, 33: 9459-9474.

[12] Jiang J, Zhou K, Dong Z, et al. Structgpt: A general framework for large language model to reason over structured data[C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023: 9237-9251.

[13] Sun J, Ju C, Tang L. Think-on-graph: deep and responsible reasoning of large language model with knowledge graph. arXiv[J]. arXiv preprint arXiv:2307.07697, 2023.

[14] Luo L, Li Y F, Haffari G, et al. Reasoning on graphs: Faithful and interpretable large language model reasoning[J]. arXiv preprint arXiv:2310.01061, 2023.

[15] He X, Tian Y, Sun Y, et al. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering[J]. Advances in Neural Information Processing Systems, 2024, 37: 132876-132907.

[16] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C]//International conference on machine learning. PMLR, 2016: 2071-2080.

[17] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[C]//Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). 2019: 3982-3992.

[18] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.

[19] Yih W, Chang M W, He X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 1321-1331.

[20] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint arXiv:2307.09288, 2023.

[21] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[22] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[23] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C]//European semantic web conference. Cham: Springer International Publishing, 2018: 593-607.