Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3086

MQPF: A MULTI-DIMENSIONAL QUALITY-AWARE PATH FUSION FRAMEWORK FOR QUESTION ANSWERING

XinYi Wang[1], Bo Liu[2*]
[1]*National University of Defense Technology, Changsha 410073, Hunan, China.*
[2]*Academy of Military Sciences, Beijing 100091, China.*
Corresponding Author: Bo Liu, Email: m19105020705@163.com

Abstract: In multi-hop question answering (MHQA) tasks, existing methods typically integrate multiple reasoning paths from knowledge graphs (KGs) and chains of thought (CoTs). Early KG-enhanced methods primarily focus on obtaining relevant knowledge but fail to consider the multi-dimensional quality of reasoning paths. Subsequent works filter paths but treat all retained paths as equally important without further differentiation. Although some recent works attempt to rank paths by quality, they only provide a relative order without quantifying the actual quality differences between paths. To address these limitations, we propose a Multi-dimensional Quality-aware Path Fusion (MQPF) framework. MQPF introduces a multi-dimensional evaluation mechanism that quantifies path quality from semantic, structural, and outcome-based dimensions. Based on the overall scores, MQPF first filters out low-quality paths to reduce noise and then assigns adaptive weights to the remaining paths according to their scores. This method effectively removes unreliable information and enhances the utilization of trustworthy information during reasoning. Experiments show that MQPF performs comparably to baselines on multiple datasets. Moreover, as a model-agnostic module, it can be used as a plug-and-play module to enhance the performance of existing multi-path reasoning methods.
Keywords: Question answering; Large language model; Knowledge graph

1 INTRODUCTION

Effective multi-hop question answering (MHQA) requires combining different reasoning paths to produce accurate answers [1-3]. This involves two key components: knowledge graph (KG) subgraphs that provide structured knowledge and chains of thought (CoTs) that support step-by-step logical reasoning. By integrating both types of reasoning paths, MHQA systems can achieve broader coverage of relevant information while maintaining robust reasoning capabilities [4,5].

Early KG-enhanced large language model (LLM) reasoning methods construct reasoning paths by retrieving or traversing KGs to improve reasoning accuracy and interpretability [3,6-8]. However, these methods only focus on obtaining relevant knowledge, overlooking the importance of evaluating the quality of reasoning paths across multiple dimensions. As a result, low-quality paths that appear highly relevant may introduce noise and biases into the subsequent reasoning steps. Subsequent works try to filter paths by majority voting or top-n sampling [9-12]. However, when generating final answers, they treat all retained paths as equally important, overlooking the quality differences between them. This equal treatment makes it hard for the LLM to effectively leverage the most critical information, especially when paths contain different answers. Some later works attempt to rank retained paths by quality [13-16], but they only establish a relative ordering. These methods cannot quantify the actual quality differences between paths, lacking a mechanism to reflect how much better one path is than another. As a result, two key challenges remain: (1) how to measure multiple quality aspects of reasoning paths, and (2) how to combine these evaluations to best guide path integration and final answer generation.

To address these challenges, we propose a Multi-dimensional Quality-aware Path Fusion (MQPF) framework. Our method begins with a multi-dimensional evaluation framework with three dimensions: Semantic Quality (S LLM): A powerful LLM is prompted as a reasoning quality evaluator to score paths based on logical coherence and factual correctness; Structural Quality (SS truct): Evaluates the structural effectiveness of paths using subgraph density (for KG paths), reasoning path length (for CoT paths), and question-path relevance; Result Quality (SRM): A reward model fine-tuned on human preferences evaluates the reliability of answers from each path. Each path receives a composite score based on these three dimensions. Based on this score, MQPF first filters out low-quality paths by applying a threshold. Next, it assigns adaptive weights to valid paths, making higher-quality paths have greater influence for answer generation. Finally, it makes the LLM prioritize high-weight paths by promoting. This integrated method provides multi-dimensional quality evaluation for hybrid reasoning paths and employs a quality-adaptive fusion strategy that translates the scores into influence weights during answer generation. In summary, our contributions are:

- We propose a comprehensive multi-dimensional quality evaluation framework for evaluating KG subgraphs and CoT reasoning paths in MHQA.
- We introduce a quality-driven fusion mechanism that assigns weights based on path quality to guide path fusion and answer generation.
- Method Experiments show that our method outperforms all baselines and can serve as a plug-and-play module to enhance existing multi-path reasoning methods.

2 RELATED WORKS

Early methods that enhance LLM reasoning with knowledge graphs (KGs) typically retrieve or traverse KGs to obtain reasoning paths or subgraphs. RoG proposes a planning-search-reasoning framework that retrieves paths from KGs to guide LLM reasoning[3]. GNN-RAG retrieves candidate answers from KG subgraphs and extracts the shortest paths between question entities and answer candidates as reasoning paths[6]. SubgraphRAG retrieves relevant triples as subgraphs to generate accurate and interpretable answers[7]. GraphReader uses an LLM to explore the KG and dynamically updates a notebook to record relevant information[8]. However, these methods only focus on retrieving relevant knowledge like reasoning paths and KG subgraphs without considering their quality across multiple dimensions. As a result, low-quality paths that appear highly relevant may lead to wrong final answers.

Subsequent works try to filter paths by majority voting or top-n sampling. Self-consistency CoT first generates multiple reasoning paths and answers and then selects the most consistent one via majority voting[9]. RoK uses CoT to expand the query and find more related entities[10]. It then builds a subgraph of the KG by matching paths between these entities and finally filters the top-n reasoning paths from this subgraph. Forest-of-Thought maintains multiple reasoning trees for parallel reasoning and applies consensus voting to determine the final answer[11]. ARG-KBQA performs multi-hop beam search over the KGs and converts the top-n reasoning paths into logical forms to enhance LLM reasoning[12]. However, these methods treat all retained paths as equally important during answer generation without further differentiation. This uniform treatment makes it difficult for the LLM to effectively leverage the most reliable information.

Some later works attempt to rank the retained paths by quality. AdaPCR uses a dense retriever to fetch candidate passages and performs context-aware reranking by concatenating each passage with the question to form a new query[13]. PromptRank uses an LLM to compute the generation probability of a question given a path as a relevance score, which is then used to rank the paths[14]. RAGtifier employs a Pinecone retriever and a BGE reranker to improve retrieval quality[15]. AttnRank finds that LLMs pay more attention to the documents at the beginning and the end[16], so it places the most relevant documents in these high-attention positions. While these methods order paths by relevance, they do not quantify the actual quality differences between paths, making it difficult to assess how much better one path is compared to another.

3 METHOD

To address the issue that multi-path information is not well utilized in multi-hop QA tasks, we propose a multi-dimensional quality assessment framework and a path filtering and weighted fusion mechanism. Specifically, we first generate multiple reasoning paths based on the question (including structured KG paths and textual CoTs), and then evaluate these paths from semantic, structural, and outcome-based dimensions. Based on the overall score of the path, we filter out low-quality paths and assign weights to the remaining valid paths to guide the LLM in generating the final answer. The overall framework is shown in Figure 1.

3.1 Multi-Dimensional Quality Assessment Framework

To effectively capture the quality differences among various reasoning paths, we propose a multi-dimensional quality assessment framework that quantifies the quality of reasoning paths from three perspectives: semantic, structural, and outcome-based dimensions. The goal of this framework is to assess the overall quality of each reasoning path and assign it an overall score. These scores are then used in the next step to weight and merge the reasoning paths.

3.1.1 Semantic quality

Semantic quality assessment measures whether a reasoning path is logically consistent and factually accurate. We use GPT-4, a large language model with strong reasoning and prompting capabilities, to evaluate each reasoning path through structured prompts. The prompt template is shown in Figure 2.

To improve scoring consistency, we evaluate each reasoning path three separate times and take the average as the semantic quality score (S_{LLM}). The result is considered valid only when the standard deviation of the three scores is below 0.5.

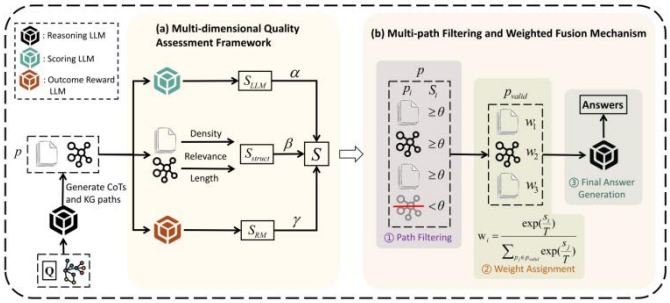

Figure 1 The Overview of MQPF. (a) Multi-dimensional Quality Assessment Framework quantifies the quality of reasoning paths from semantic, structural, and outcome-based dimensions. (b) Multi-path Filtering and Weighted Fusion Mechanism contains 3 parts: ① Path Filtering, ② Weight Assignment, and ③ Final Answer Generation.

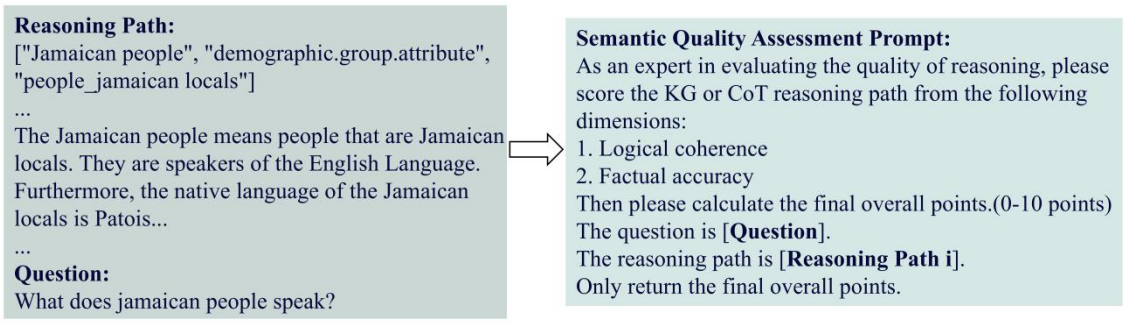

Figure 2 Prompt Template for Semantic Quality Assessment

3.1.2 Structural quality

Structural quality assessment focuses on evaluating the effectiveness of a reasoning path's structure. Paths that are more compact and contain higher information density receive higher scores. Considering the structural differences among path types, we designed separate evaluation strategies for structured paths (KG subgraphs) and textual paths (CoTs).

Structured Paths The structural quality of this type of path consists of two components: (1) Graph Density, which measures the connectivity and informational richness between nodes to avoid isolated or sparse paths; (2) Semantic Relevance between the path and the question, where higher relevance leads to a higher score. For the former, we use the number of relations and entities in the structured path to calculate it; for the latter, we convert the structured path into a sequence of triples and encode it into vectors. We then measure its relevance to the question embedding by calculating the cosine similarity between them. For a structured path $p_i(r, e)$ and question Q, the structural quality score S_{struct} is:

$$S_{struct} = \frac{num(r)}{num(e)} \cdot correlation(Q, p_i) \quad (1)$$

where r are the relationship edges and e are the entity nodes in the path. The correlation function calculates the relevance between question Q and reasoning path p_i with cosine similarity.

Textual Paths The scoring criteria for this type of path include length and relevance to the question. Although techniques like CoT improve the interpretability and correctness of the reasoning process, they often lead LLMs to produce unnecessarily long and redundant steps [17,18]. Therefore, we link the structural score of textual paths to their length. Paths that are more concise and information-dense receive higher scores. For a textual reasoning path p_i, the structural quality score is:

$$S_{struct} = e^{-\eta \cdot len(p_i)} \cdot correlation(Q, p_i) \quad (2)$$

where η is a hyperparameter that controls how quickly the score decreases as the path gets longer. The correlation function measures the relevance between question Q and reasoning path p_i with cosine similarity.

3.1.3 Outcome-based quality

Outcome-based quality evaluates whether the final answer from a reasoning path aligns with human preferences and factual standards. We use an outcome reward model (ORM) named SkyworkRM-Llama3.1-8B, which has been fine-tuned on human preference data, to assess each reasoning path. Since this reward model can only evaluate textual inputs, we first convert structured paths into a textual sequence of connected triplets. The textual path is then fed into the ORM to obtain an outcome-based quality score S_{RM}.

3.1.4 Overall score

The overall score S_i for each reasoning path p_i is computed by a weighted combination of scores from semantic (S_{LLM}), structural (S_{struct}), and outcome-based dimensions (S_{RM}).

$$S_i = \alpha \cdot S_{LLM} + \beta \cdot S_{struct} + \gamma \cdot S_{RM} \quad (3)$$

where α, β, and γ are the weight coefficients assigned to the semantic, structural, and outcome-based dimensions. They are used to keep each type of score within the same range, so the scores from each dimension can be considered equally and comprehensively when calculating the overall score. This overall score reflects the comprehensive quality of the path across multiple aspects and serves as the basis for weighted integration in the subsequent path fusion stage.

3.2 Multi-path Filtering and Weighted Fusion Mechanism

After obtaining overall scores for multiple reasoning paths, we propose a score-based multi-path filtering and weighted fusion mechanism to enhance the accuracy of the final answer. This mechanism consists of three parts: path filtering, weight assignment, and final answer generation. It is designed to select high-quality reasoning paths, dynamically assign fusion weights based on their scores, and ultimately guide LLMs to generate the final answer through prompts.

3.2.1 Path filtering

To reduce the influence of redundant and erroneous information in the subsequent fusion stage, we apply a threshold-based filtering strategy to remove low-quality reasoning paths. Specifically, a threshold θ is set, and all paths satisfying $S \geq \theta$ are retained to form a valid path set p_{valid}.

$$p_{valid} = \{p_i | S_i \geq \theta\} \quad (4)$$

The set of valid paths p_{valid} will then be assigned weights based on their scores in the next step.

3.2.2 Weight assignment

To better distinguish the quality differences among paths in the valid set p_{valid}, we assign each path a weight based on its score S_i. Paths with higher scores receive correspondingly greater weights, and larger score gaps result in more distinct differences in weight allocation. Specifically, we use a Softmax function with a temperature parameter T to compute the weights. For a path p_i in p_{valid}, its weight w_i is:

$$w_i = \frac{e^{\frac{S_i}{T}}}{\sum_{p_j \in p_{valid}} e^{\frac{S_j}{T}}} \tag{5}$$

where the temperature parameter T is used to control the concentration of the weight distribution, making the weighting more sensitive to variations in the scores. The weights w_i will be used in the subsequent fusion of reasoning paths and final answer generation.

3.2.3 Final answer generation

To effectively incorporate the weighted multi-path information into the LLMs for generating the final answer, we design a prompt template. This template explicitly indicates each reasoning path along with its assigned weight, making the LLMs prioritize information from higher-weighted paths. The detailed prompt template is shown in Figure 3. By explicitly embedding the paths and their weights into the input context, the template makes the LLMs recognize quality differences among reasoning paths to achieve weight-based integration and reasoning. This method enhances the accuracy and reliability of the final answer.

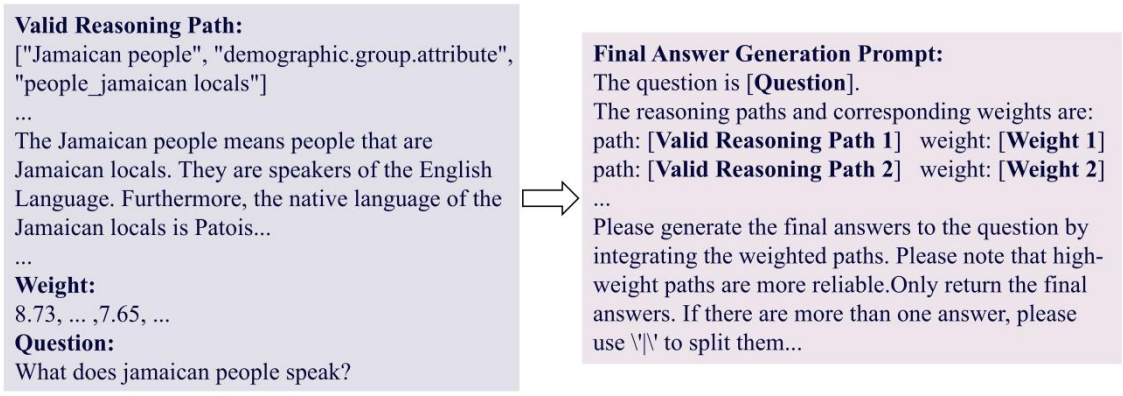

Figure 3 Prompt Template for Final Answer Generation

4 EXPERIMENTS

4.1 Experiment Setup

4.1.1 Datasets

Following previous works [3,19,20], we conduct experiments on two datasets, WebQuestionSP (WebQSP) and Complex WebQuestions (CWQ) [21, 22]. The statistics of the datasets are given in Table 1. We follow previous works (Luo et al., 2024a) to use the same train and test splits for fair comparison. The questions in WebQSP are 1-hop or 2-hop, and the questions in CWQ are 2-4 hops. The two datasets test the model's ability to understand and answer questions with multiple facts and reasoning steps. The KG for both datasets is Freebase [23].

Table 1 Statistics of Datasets

Dataset	#Train	#Test	Max #hop
WebQSP	2826	1628	2
CWQ	27639	3531	4

4.1.2 Baselines

We compare MQPF with 13 baselines grouped into 3 categories: (1) LLM only, (2) Multi-path LLM, (3) KG LLM.

(1) LLM-only methods use only LLMs for reasoning without other enhancement methods.

Qwen2-7B is one of a series of LLMs developed by the Alibaba Cloud Tongyi Qianwen team, with a parameter size of 7 billion[24].

Llama-2-7B is one of the Llama 2 series of LLMs developed by Meta AI, with a parameter size of 7 billion[25].

Llama-3.1-8B is one of the Llama 3 series of LLMs developed by Meta AI, with a parameter size of 8 billion[26].

(2) Multi-path+LLM methods prompt LLMs to generate multiple KG paths and CoTs and then generate the final answers. The reasoning LLMs for this kind of baselines are Llama-2-7B and Llama-3.1-8B.

(3) KG+LLM methods use KGs to enhance LLM reasoning.

G-Retriever retrieves the relevant nodes and edges, then constructs the relevant subgraph using the bonus Steiner tree method[27].

GRAG retrieves text subgraphs and performs soft pruning to identify relevant subgraph structures effectively, and proposes a new cue strategy[28].

SubgraphRAG generates accurate and explainable answers by efficiently retrieving relevant subgraphs from KGs and leveraging LLMs for reasoning[7].
RoG proposes a planning-search-reasoning framework, which retrieves reasoning paths from KGs to guide LLMs in reasoning[3].
GNN-RAG integrates graph neural networks (GNNs) as retrieval mechanisms to extract structured knowledge paths from KGs, which are then verbalized and fed into LLMs for answer generation[6].

4.1.3 Evaluation metrics

Following previous works [3,20,28], we use Hit@1 and the F1 score as evaluation metrics. Hit@1 checks if the ground truth exists in the generated answers. The F1 score is a harmonic average of accuracy and recall, providing a metric that balances false positives and false negatives.

4.1.4 Implementations

We choose Llama-3.1-8B-Instruct and Llama-2-7B-Chat as the reasoning LLMs. The number of CoT paths and KG paths is both set to 4 to get multiple reasoning paths. The hyperparameter η for S_{struct} is set to 1×10^{-4}, and the threshold θ for filtering is set to 2. Both of the parameters are selected based on the experimental results. We set the weight coefficients α, β and γ to ensure that the scores from each dimension are considered equally when calculating the overall score and to restrict the overall score to a range of $(0-10)$. Considering that S_{LLM} is in the range of $(0-10)$ and S_{struct} is in the range of $(0-1)$, we set $\alpha = \frac{1}{3}$ and $\beta = \frac{10}{3}$. Because S_{RM} varies with path quality, we dynamically adjust γ based on its maximum and minimum values to map the score to a range of $(0-\frac{10}{3})$.

4.2 Main Results

We compare our method, MQPF, to other baselines on the datasets. As Table 2 shows, MQPF performs best on CWQ and is comparable to other baselines on WebQSP. On CWQ, the F1 score and Hit@1 are 2.6% and 6.1% above the best baseline. On WebQSP, although MQPF itself performs slightly worse than RoG and SubgraphRAG, it can improve the performance of the two baselines when combined with them. The MQPF+baselines increase the F1 score of the corresponding baseline by up to 1.4% and Hit@1 by up to 1.0% on WebQSP. It shows that MQPF can effectively enhance the multi-path reasoning of the LLMs as a plug-and-play module.
It is found that the performance of Multi-path+LLM baselines is better than that of LLM only baselines. The F1 score is improved by up to 74.8% and Hit@1 by up to 92.9%. It indicates that multiple reasoning paths can enhance the coverage and accuracy of the final answers. Also, the overall performance of KG-enhanced methods (Multi-path+LLM and KG+LLM baselines) is better than that of LLM only baselines, indicating that KGs are important in MHQA tasks.
We also observe that larger LLMs do not always perform better than smaller LLMs. For the performance of Llama-2-7B and Llama-3.1-8B, Llama-2-7B has a higher F1 and Hit@1 on the WebQSP dataset, by 4.9% and 1.6%. In the CWQ dataset, its Hit@1 is still higher. This suggests that increasing the parameters does not inherently enhance the graph reasoning ability of LLMs.

Table 2 Model Performance on Two Datasets Comparing Three Catagories of Methods

Category	Method	WebQSP		CWQ	
		F1 Score	Hit@1	F1 Score	Hit@1
LLM only	Qwen2-7B [24]	0.3550	0.5080	0.2160	0.2530
	Llama-2-7B [25]	0.3650	0.5640	0.2140	0.2840
	Llama-3.1-8B [26]	0.3480	0.5550	0.2240	0.2810
Multi-path+LLM	Llama-2-7B [25]	0.4625	0.7168	0.3740	0.5411
	Llama-3.1-8B [26]	0.4601	0.7137	0.3810	0.5421
KG+LLM	G-Retriever [27]	0.4674	0.6808	0.3396	0.4721
	GRAG [28]	0.5022	0.7236	0.3649	0.5018
	SubgraphRAG [7]	0.7057	<u>0.8661</u>	0.4716	0.5698
	RoG [3]	0.7080	0.8570	0.5620	0.6260
	GNN-RAG [6]	0.7130	0.8060	0.5940	0.6170
Our method	MQPF+Llama-3.1-8B	0.6431	0.8337	**0.6094**	**0.6640**
	MQPF+Llama-2-7B	0.6427	0.8340	<u>0.5990</u>	<u>0.6639</u>
	MQPF+RoG	<u>0.7145</u>	0.8636	0.5762	0.6370
	MQPF+SubgraphRAG	**0.7154**	**0.8718**	0.5027	0.5892

Note: The best results are **bolded**, and the second best results are underlined.

4.3 Ablation Study

We conduct a series of evaluations of MQPF to see which component plays a key role in the overall score, including removing S_{LLM}, removing S_{struct}, and removing S_{RM}. We can see that the performances of variables all decrease, as shown in Table 3. It indicates that every component is indispensable. Among them, removing S_{struct} drops model performance the most. On the CWQ dataset, the F1 score decreased by 4.1% and Hit@1 decreased by 3.1%. The situation is similar on WebQSP. This suggests that S_{struct} plays a more central role in the overall score. It evaluates the structural validity and information density of reasoning paths to select more coherent and concise ones, which effectively improves the accuracy of the final answers.

Table 3 Performances of Three Model Variables

Method	WebQSP		CWQ	
	F1 Score	Hit@1	F1 Score	Hit@1
MQPF	**0.6431**	**0.8337**	**0.6094**	**0.6640**
w/o S_{LLM}	0.6271	0.8188	0.5955	0.6593
w/o S_{struct}	0.6196	0.8139	0.5846	0.6432
w/o S_{RM}	0.6293	0.8237	0.5915	0.6511

Note: The best results are **bolded**.

4.4 Analytical Experiments

4.4.1 Structural hyperparameter analysis

For the hyperparameter for S_{struct} for textual paths η, we conduct multiple experiments with different values of it. To make the distinction between different reasoning paths greater, we set η while keeping the score differences of the reasoning paths relatively large. As shown in Figure 4, the maximum score difference continues to improve as η decreases. However, as η decreases, S_{struct} will increasingly ignore the path length and rely more on the correlation. After comprehensive consideration, we set η to 1×10^{-4}.

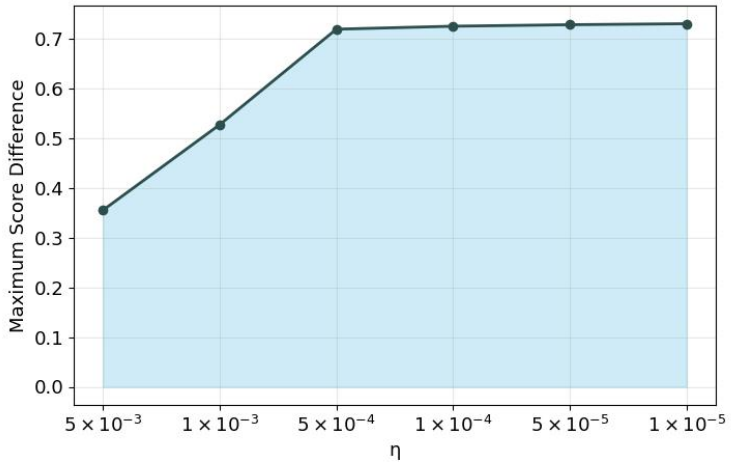

Figure 4 Performances on Different Values of η

4.4.2 Threshold analysis

For the threshold θ for path filtering, we conduct multiple experiments with different values of it. As shown in Figure 5, when $\theta = 2$, Hit@1 is the highest. Therefore, the threshold θ is set to 2.

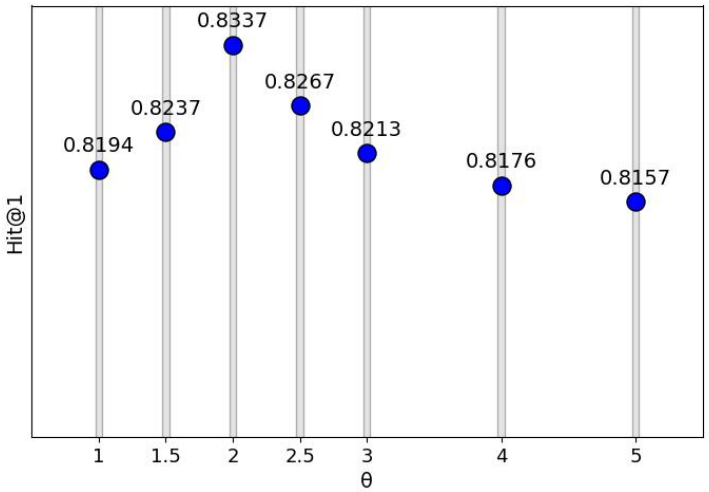

Figure 5 Performances on Different Values of θ

5 CONCLUSION

In this paper, we propose a Multi-dimensional Quality-aware Path Fusion framework (MQPF) for path quality assessment and multi-path fusion in MHQA tasks. It introduces a multi-dimensional evaluation mechanism that quantifies path quality from semantic, structural, and outcome-based dimensions. Based on the overall scores, it filters

out low-quality paths to reduce noise and then assigns adaptive weights. Experiments show that MQPF performs comparably to baselines and can be used as a plug-and-play module to enhance the performance of other multi-path reasoning methods.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCE

[1] Bi Z, Hajialigol D, Sun Z, et al. Stoc-tot: Stochastic tree-of-thought with constrained decoding for complex reasoning in multi-hop question answering. 2024. DOI: https://doi.org/10.48550/arXiv.2407.03687.

[2] Lee S, Shin J, Ahn Y, et al. Zero-shot multi-hop question answering via monte-carlo tree search with large language models. 2024. DOI: https://doi.org/10.48550/arXiv.2409.19382.

[3] Luo L, Li Y F, Haffari G, et al. Reasoning on graphs: Faithful and interpretable large language model reasoning. International Conference on Learning Representations. 2024.

[4] Park J, Patel A, Khan O Z, et al. Graph-guided reasoning for multi-hop question answering in large language models. 2023. DOI: https://doi.org/10.48550/arXiv.2311.09762.

[5] Chen L Y, Tong P R, Jin Z M, et al. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. Advances in Neural Information Processing Systems (NeurIPS) 37, 2024. DOI: https://doi.org/10.48550/arXiv.2410.23875.

[6] Mayromatis C, Karypis G. Gnn-rag: Graph neural retrieval for large language model reasoning. 2024. DOI: https://doi.org/10.48550/arXiv.2405.20139.

[7] Li M, Miao S, Li P. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. 2025. DOI: https://doi.org/10.48550/arXiv.2410.20724.

[8] Li S, He Y, Guo H, et al. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. 2024. DOI: https://doi.org/10.48550/arXiv.2406.14550.

[9] Wang X, Wei J, Schuurmans D, et al. Self-consistency improves chain of thought reasoning in language models. 2023. DOI: https://doi.org/10.48550/arXiv.2203.11171.

[10] Wang Y, Jiang B, Luo Y, et al. Reasoning on efficient knowledge paths: knowledge graph guides large language model for domain question answering. 2024. DOI: https://doi.org/10.48550/arXiv.2404.10384.

[11] Bi Z, Han K, Liu C, et al. 2025. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning. 2025. DOI: https://doi.org/10.48550/arXiv.2412.09078.

[12] Tian Y, Song D, Wu Z, et al. Augmenting reasoning capabilities of llms with graph structures in knowledge base question answering. Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, 11967-11977.

[13] Ko T W, Jiang J Y, Cheng P J. Beyond independent passages: Adaptive passage combination retrieval for retrieval augmented open-domain question answering. 2025. DOI: https://doi.org/10.48550/arXiv.2507.04069.

[14] Khalifa M, Logeswaran L, Lee M, et al. Few-shot reranking for multi-hop qa via language model prompting. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, 15882-15897. DOI: 10.18653/v1/2023.acl-long.885.

[15] Cofala T, Astappiev O, Xion W, et al. Ragtifier: Evaluating rag generation approaches of state-of-the-art rag systems for the si-gir liverag competition. 2025. DOI: https://doi.org/10.48550/arXiv.2506.14412.

[16] Yi Z, Zeng D, Ling Z, et al. Attention basin: Why contextual position matters in large language models. 2025. DOI: https://doi.org/10.48550/arXiv.2508.05128.

[17] Chiang C H, Lee H Y. Over-reasoning and redundant calculation of large language models. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, St. Julian's, Malta, 2024, 161-169.

[18] Nayab S, Rossolini G, Simoni M, et al. Concise thoughts: Impact of output length on llm reasoning and cost. 2024. DOI: https://doi.org/10.48550/arXiv.2407.19825.

[19] Wang K, Duan F, Wang S, et al. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. 2023. DOI: https://doi.org/10.48550/arXiv.2308.13259.

[20] Luo L, Zhao Z, Haffari G, et al. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. 2024. DOI: https://doi.org/10.48550/arXiv.2410.13080.

[21] Yih W t, Richardson M, Meek C, et al. The value of semantic parse labeling for knowledge base question answering. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, 201-206.

[22] Talmor A, Berant J. The web as a knowledge-base for answering complex questions. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, 641-651.

[23] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge. Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08). Association for Computing Machinery, New York, NY, USA, 2008, 1247-1250. DOI: https://doi.org/10.1145/1376616.1376746.
[24] Yang A, Yang B, Hui B, et al. Qwen2 technical report. 2024. DOI: https://doi.org/10.48550/arXiv.2407.10671.
[25] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models. 2023. DOI: https://doi.org/10.48550/arXiv.2307.09288.
[26] Grattafiori A, Dubey A, Jauhri A, et al. The llama 3 herd of models. 2024. DOI: https://doi.org/10.48550/arXiv.2407.21783.
[27] He X, Tian Y, Sun Y, et al. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. Advances in Neural Information Processing Systems, 2025(37): 132876-132907.
[28] Hu Y, Lei Z, Zhang Z, et al. Grag: Graph retrieval-augmented generation. 2025. DOI: https://doi.org/10.48550/arXiv.2405.16506.