Graph Neural Network Enhanced Retrieval for Question Answering of Large Language Models

Zijian Li[1][2][*] **, Qingyan Guo**[2][3]**, Jiawei Shao**[1]**, Lei Song**[2]**, Jiang Bian**[2]**,**
Jun Zhang[1][†] **, Rui Wang**[2][†]
[1]HKUST [2]Microsoft Research [3]Tsinghua University
{zijian.li,jiawei.shao}@connect.ust.hk, gqy22@mails.tsinghua.edu.cn,
{lesong,jiabia,ruiwa}@microsoft.com, eejzhang@ust.hk

Abstract

Retrieval augmented generation has revolutionized large language model (LLM) outputs by providing factual supports. Nevertheless, it struggles to capture all the necessary knowledge for complex reasoning questions. Existing retrieval methods typically divide reference documents into passages, treating them in isolation. These passages, however, are often interrelated, such as passages that are contiguous or share the same keywords. Therefore, it is crucial to recognize such relatedness for enhancing the retrieval process. In this paper, we propose a novel retrieval method, called *GNN-Ret*, which leverages *graph neural networks* (GNNs) to enhance retrieval by exploiting the relatedness between passages. Specifically, we first construct a *graph of passages* by connecting passages that are structure-related or keyword-related. A *graph neural network* (GNN) is then leveraged to exploit the relationships between passages and improve the retrieval of supporting passages. Furthermore, we extend our method to handle multi-hop reasoning questions using a *recurrent graph neural network* (RGNN), named *RGNN-Ret*. At each step, *RGNN-Ret* integrates the graphs of passages from previous steps, thereby enhancing the retrieval of supporting passages. Extensive experiments on benchmark datasets demonstrate that *GNN-Ret* achieves higher accuracy for question answering with a single query of LLMs than strong baselines that require multiple queries, and *RGNN-Ret* further improves accuracy and achieves state-of-the-art performance, with up to 10.4% accuracy improvement on the 2WikiMQA dataset. The code is publicly available at https://github.com/zli999/GNN_Ret.

1 Introduction

Large language models (LLMs) continue to struggle with factual errors when encountering knowledge intensive questions (Huang et al., 2023; Mallen et al., 2022; Ji et al., 2023). Although retrieval-augmented LLMs (Lewis et al., 2020) have improved factuality and precision of question answering by including relevant passages, there remains a persistent challenge in accurately capturing all the supporting passages when encountering complex knowledge-intensive questions. This limitation can be attributed to the inherent information asymmetry in complex questions. In particular, the questions tend to consist of elaborated background details, leaving only a small portion dedicated to specific inquiries. As an example, in question: *'Why did crime rise on Mars after the Mafia's arrival?'*, the majority part delves into the background (i.e., *'crime rise ...... Mafia's arrival'*) with only a few words requesting the reason (i.e., '*Why*'), which consequently retrieves passages on details of crime rising instead of the reason for it.

This phenomenon also frequently arises in multi-hop reasoning questions. Considering the sample question '*Where was the performer of song Left & Right (Song) born?*' in Fig. 1, it becomes apparent that while we can retrieve the knowledge that *the performer of song Left & Right is D'Angelo*, his birthplace remains absent. Previous works (Trivedi et al., 2023; Press et al., 2022; Yao et al., 2022) have attempted to address this issue by incorporating multi-hop reasoning or question rewriting into retrieval processes, which enables them to retrieve information based on prior reasoning outcomes. LLMs, however, often generate plausible reasons and incorrect rewriting questions without accurate prior domain knowledge of the given question (Huang et al., 2023; Zhang et al., 2023), thus affecting the subsequent retrieval process.

One reason why existing methods struggle to

[*] Work done during an internship at Microsoft Research Asia.

[†] Corresponding Author.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6612–6633
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

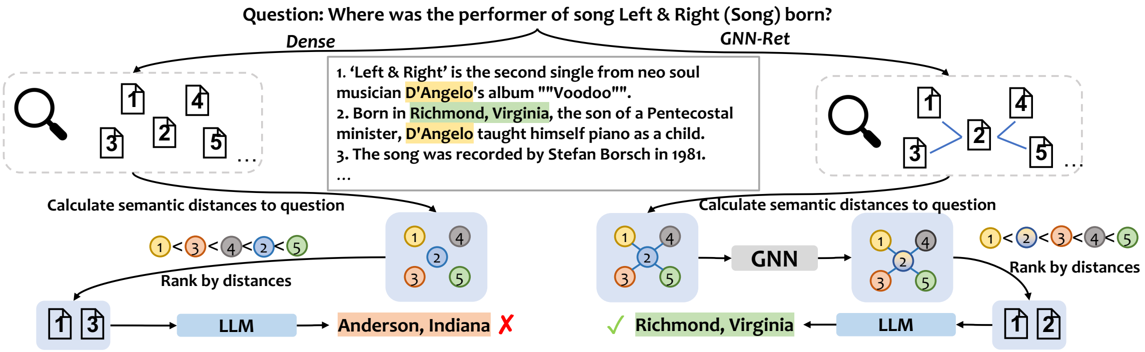

Question: Where was the performer of song Left & Right (Song) born?
Dense
GNN-Ret
1. 'Left & Right' is the second single from neo soul musician D'Angelo's album ""Voodoo"".
2. Born in Richmond, Virginia, the son of a Pentecostal minister, D'Angelo taught himself piano as a child.
3. The song was recorded by Stefan Borsch in 1981.
...
Calculate semantic distances to question
1<3<4<2<5
Rank by distances
LLM
Anderson, Indiana ✗
Calculate semantic distances to question
GNN
1<2<3<4<5
Rank by distances
✓ Richmond, Virginia
LLM

Figure 1: Overview of comparison between dense retrieval and *GNN-Ret*. The shared keywords and ground-truth answers are highlighted in yellow and green, respectively. By considering the relatedness between passages, *GNN-Ret* can retrieve all the supporting passages for QA.

handle the information asymmetry is their tendency to consider passages in isolation (Karpukhin et al., 2020a) and retrieve the supporting passages based mainly on semantic distances, making it difficult to retrieve all the supporting passages, especially those containing only few words for inquiry. However, the supporting passages of background and inquiry information are usually correlated. They can be *structure-related* when located in the same section or document, e.g., the happened event (i.e., *crime rises on Mars*) and its corresponding reason are located in the same section but different passages. Also, they can be *keyword-related* by sharing the same keywords or entities, e.g., the passages of background information (*performer of song Left & Right*) and inquiry information (*birthplace of D'Angelo*) in Fig. 1 share the same keyword: *D'Angelo*. By considering the relatedness between passages, it is possible to retrieve all the supporting passages even when they have significant semantic differences from the query. In this work, we aim to enhance retrieval by taking the relatedness between passages into account.

Contributions. To establish the relatedness between passages for retrieval purposes, this study initially constructs a *graph of passages* by connecting individual passages based on both *structural information* and *shared keywords*, with each passage as a node in this graph. The key challenge lies in how to effectively leverage the passage of graphs to enhance retrieval coverage. Graph neural networks (GNNs) are neural networks tailored to analyze graph data and adeptly grasp the relationships between nodes and edges (Scarselli et al., 2008). Thus, we propose to leverage a GNN to enhance the retrieval process by effectively capturing the relatedness between passages and name this method *GNN-Ret*. The GNN facilitates the integration of semantic distances between related passages, thus enabling the semantic distances of passages containing background information to impact the retrieval of passages relevant to the inquiry. To address multi-hop reasoning questions, we further propose a retrieval method, named *RGNN-Ret*, which leverages a *recurrent graph neural network* (RGNN) to enhance the retrieval process at each step by integrating the retrievals from previous steps through the interconnected graphs of passages. This boosts the retrieval coverage for supporting passages over steps even when LLMs generate incorrect reasons or subquestions for retrieval.

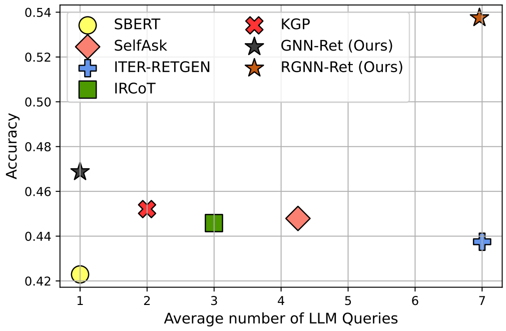


Figure 2: Accuracy and average number of LLM queries for our proposed methods and baselines on 2WikiMQA.

Through the experiments on four benchmark datasets, we demonstrate the effectiveness of GNNs in enhancing retrieval of supporting passages and thus improving accuracy for QA. For example, as shown in Fig. 2, our proposed *GNN-Ret* with a single query of LLMs significantly outperforms baselines in accuracy on 2WikiMQA (Ho et al., 2020), including those methods that require multiple queries of LLMs. Moreover, by extending GNN to multi-hop reasoning questions, our proposed *RGNN-Ret* achieves state-of-the-art accuracy.

2 Related Works

Retrieval-augmented LLM. QA (Voorhees et al., 1999; Wang et al., 2024a) is a task that often requires external and up-to-date knowledge sources to answer factoid-based questions. Retrieve-and-read is the basic framework for these questions (Gao et al., 2023; Zhu et al., 2023). For retrieval, sparse retrievers, e.g., TF-IDF and BM25 (Robertson et al., 2009), or dense retrievers, e.g., DPR (Karpukhin et al., 2020b) and Contriever (Izacard et al., 2021), are applied to compute lexical distances or semantic distances between passages and the question. The passages with smaller distances are retrieved, which are then prefixed with the question for factual answering (Lewis et al., 2020). However, this framework treats passages as isolated units, making it difficult to retrieve all the supporting passages for the question in a single step. To address this limitation, many approaches have been proposed to integrate the retrieval and reasoning processes to improve the retrieval of supporting passages (Gao et al., 2023; Shao et al., 2023; Feng et al., 2024; Trivedi et al., 2023; Jiang et al., 2023b; Yao et al., 2022; Press et al., 2022). They first prompt LLMs to generate the next-step reason or subquestion and then use it to guide retrieval for QA. However, a challenge in these methodologies pertains to the potential generation of hallucinated reasons or erroneous subquestions by LLMs without accurate domain knowledge of this question, leading to degraded subsequent retrievals (Huang et al., 2023).

Graph-enhanced LLM. As structured, explicit, and editable representations of knowledge, knowledge graphs (KGs) have been effectively used for boosting retrieval coverage (Min et al., 2019) and enhancing reasoning capabilities of LLMs (Li et al., 2023a; Xie et al., 2022; Baek et al., 2023; Wang et al., 2023; Park et al., 2023; Ding et al., 2024; Jin et al., 2023; Li et al., 2023b; Xiong et al., 2024; He et al., 2024). Instead of naively combining KGs with LLMs, many works directly prompt LLMs to perform reasoning on KGs (Sun et al., 2024; Jiang et al., 2023a; LUO et al., 2024). They first identified the entities on the knowledge graph and then search reasoning paths accordingly (Sun et al., 2024; LUO et al., 2024; Jiang et al., 2023a). However, KG-based methods encounter a limitation that they can only handle questions that are effectively represented as KGs. They may struggle with questions that necessitate a comprehensive understanding of lengthy contextual information, such as the mentioned example: the cause of crime increasing. Rather than KGs, a recent study (Jin et al., 2024) attempts to construct a graph of passages and perform reasoning on this graph. It connects documents according to some specific words and performs reasoning on this graph for QA. In contrast, we utilize the graph of passages to enhance the retrieval process. Another related work, KGP (Wang et al., 2024b), performs retrieval on the graph and search related passages. However, this approach does not consider the integration between related passages and may not effectively enhance retrieval for the supporting passages with poor semantic similarity.

3 Graph Neural Network Enhanced Retrieval for QA

The QA system typically adopts a *retrieve-and-read* routine (Lewis et al., 2020): the retriever computes semantic distances between passages and the question in the embedding space, finds out relevant passages from the whole corpus, and then the *reader* (LLM) generates the answer based on them. Considering the information asymmetry of complex questions, directly using the initial semantic distances of passages for retrieval is difficult to retrieve all the supporting passages. Therefore, we propose to process by taking the relatedness between passages into account. Specifically, we construct a *graph of passages* (GoPs) by connecting related passages (Section 3.1). To exploit the relatedness between passages to enhance retrieval, we utilize a GNN to enable integration between related passages (Section 3.2). For multi-hop questions, we further leverage an RGNN to establish relationship between graphs over steps to enhance retrieval over answering processes (Section 3.3).

3.1 Graph of Passages (GoPs)

Establish relationships between related passages. Existing retrieval methods rely on semantic distances between passages and the question (Gao et al., 2023; Zhu et al., 2023), which may fail to retrieve some supporting passages with large semantic distances, but related to those with small semantic distances to the question, due to the information asymmetry. These passages can be *structural-related*, e.g., the happened event (*'crime rising'*) and its corresponding reasons are located in the same section. As shown in Fig. 1, they can also be *keyword-related*, e.g., the supporting passages for background information ('*performer of song Left & Right*') and inquiry information ('*the birthplace of D'Angelo*') share the same keyword ('*D'Angelo*'). Therefore, we propose to establish relationship between passages using structural information and shared keywords to construct the graph of passages. Specifically, when chunking the documents, we record the order of passages and connect passages that are physically next to each other in documents. In addition, we extract keywords from passages by prompting LLMs and connect passages containing the same keywords (Min et al., 2019; Wang et al., 2024b). As such, the related passages are connected, having the potential to facilitate retrieval.

3.2 GNN Enhanced Retrieval on GoPs

After establishing connections between related passages, we obtain a GoPs, with each passage as a node. When conducting the retrieval process, we first compute the semantic distances between passages and the question q (Gao et al., 2023; Zhu et al., 2023). The semantic distances of related passages are connected according to the GoPs. To take relatedness between passages into account, we utilize a GNN to process semantic distances based on GoPs and obtain the *integrated semantic distances*, which benefits the retrieval of supporting passages.

Graph neural network. A graph is defined as an ordered pair $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes v_i and $\mathcal{E}$ is the set of edges e_{ij} connecting two nodes v_i and v_j. We define h_i^l as the hidden state (integrated semantic distance) of the $i-$th node for $i \in \{1, \cdots, |\mathcal{V}|\}$ at layer l. We input semantic distances of passages into GNN as h_i^0. The set of nodes $\mathcal{N}_{v_i}$ stands for the neighbors of node v_i. For an $L-$layer GNN (Scarselli et al., 2008), the hidden state of each node $v_i \in \mathcal{V}$ is updated through iterative interactions with neighbors. Specifically, at layer l, the received messages m_i^l of node v_i are calculated using the hidden states of its neighbors. Then the received messages m_i^l and the previous-layer hidden state h_i^{l-1} are utilized to compute the hidden state h_i^l at layer l. By processing hidden states for all the passages layer by layer, we obtain the integrated semantic distance h_i^L for each node $v_i \in \mathcal{V}$. We elaborately introduce GNNs for retrieval as below.

Minimum semantic distance as message. Since each passage maintains and shares many keywords with other passages, each passage has a large number of neighbors. Some of them are not relevant to the question, contributing to large semantic distances and misleading information. Therefore, each node v_i only receives the message from the neighbor that has the minimum (integrated) semantic distance to the question at layer l. The received message of node v_i is thus formulated as $m_i^l = \min_{j \in \mathcal{N}_{v_i}} h_j^{l-1}$.

Relevant nodes sampling. Given the large scale of GoPs with plenty of nodes, it is computationally inefficient to propagate information across all of them. Besides, allowing irrelevant nodes to pass messages to their neighbors may affect the retrieval process. Therefore, we only sample the relevant node set $\mathcal{T}_K^l$ with the top K smallest (integrated) semantic distances at layer l. Only neighbors of relevant nodes receive messages from them and update hidden states. We define $\mathcal{S}_K^l$ as the neighbor set of the relevant node set $\mathcal{T}_K^l$. For each node $v_i \in \mathcal{S}_K^l$, the message passed from neighbors that are relevant nodes is reformulated as follows:

$$m_i^l = \min_{j \in \mathcal{N}_{v_i}, v_j \in \mathcal{T}_K^l} h_j^{l-1}. \quad (1)$$

By integrating the message from neighbors, the hidden state of each node $v_i \in \mathcal{S}_K^l$ at layer l is computed using the parameter α^l as follows:

$$h_i^l = \alpha^l h_i^{l-1} + (1 - \alpha^l) m_i^l. \quad (2)$$

The nodes without received messages maintain their hidden states of the previous layer. For the GNN with L layers, we compute the hidden state layer by layer and then obtain the integrated semantic distance h_i^L for each node v_i. With GNN, the supporting passages for inquiry information integrate with the small semantic distance from those for background information and thereby obtain smaller integrated semantic distances, promoting them to be retrieved.

Hinge objective for GNN. To retrieve all the supporting passages, we aim to reduce their integrated semantic distances more than those of non-target nodes. With the ground-truth set of supporting passages $\mathcal{S}_Y$, we first define the average integrated semantic distances of supporting passages and non-target nodes as $\bar{d}_Y^L = \frac{1}{|\mathcal{S}_Y|} \sum_{j \in \mathcal{S}_Y} h_j^L$ and $\bar{d}_{O/Y}^L = \frac{1}{|\mathcal{S}_{O/Y}|} \sum_{o \in \mathcal{S}_{O/Y}} h_o^L$, respectively, where S_O is the competitive node set with the top O smallest semantic distances, and $S_{O/Y}$ is the non-target node set by removing the target node set S_Y from S_O. We only consider the competitive node set since the average integrated semantic distance of all the nodes is significantly large and loses effectiveness in training. With the output average semantic distances of supporting passages and non-target

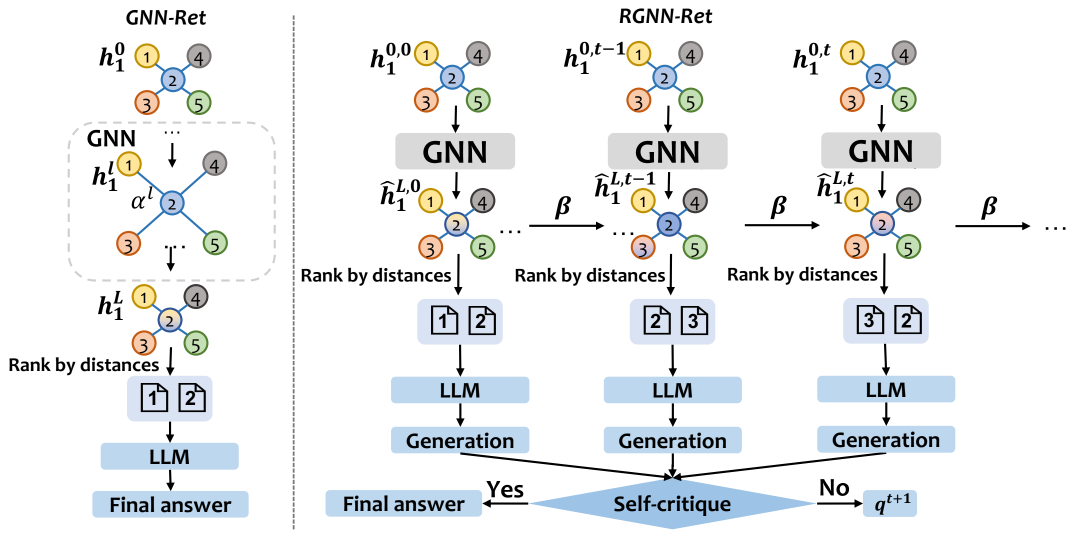


Figure 3: Illustration of *GNN-Ret* and *RGNN-Ret*.

nodes, we formulate the hinge objective function with threshold r as follows:

$$\ell = \max(0, r + \bar{d}_Y^L - \bar{d}_{O/Y}^L). \quad (3)$$

We update the GNN using the gradient descent, and training details are elaborated in Appendix A.1.

3.3 RGNN Enhanced Retrieval for Multi-hop Reasoning Questions

The information asymmetry phenomenon also frequently arises in complex multi-hop reasoning questions, as analyzed in the Introduction section. Although recent works have incorporated multi-hop reasoning or question rewriting into retrieval (Yao et al., 2022; Trivedi et al., 2023; Shao et al., 2023), LLMs may generate hallucinated reasons or incorrect subquestions due to the absence of prior knowledge to the question (Huang et al., 2023). These hallucinated reasons or incorrect subquestions fail to retrieve supporting passages when only considering the retrieval process in isolation for each step. Additionally, they also encounter another challenge: whether to continue the next-step answering process or output the final answer. To overcome these challenges, we first propose a *self-critique* technique by prompting LLMs to determine the termination of the answering process. To enhance retrieval of supporting passages over steps, we further utilize an RGNN to enable the integration between graphs of passages in different steps.

Self-critique of LLMs. To answer multi-hop reasoning questions, *self-critique* involves prompting the LLM to generate subquestions and answer them using retrieved passages in a step-by-step manner. As depicted in Fig. 3, at each step, after the LLM generates a subanswer to the subquestion, the question and all the generated subanswers are input into the LLM and critiqued to determine whether the generated subanswers are evident to generate the final answer to the question. If the output is 'Yes', the LLM will be prompted to generate the final answer. Otherwise, it will be requested to generate the next-step subquestion. Compared with *SelfAsk* (Press et al., 2022), we omit the subquestions and focus on whether existing evidences are satisfactory for answering the question when determining the termination. This *self-critique* technique decomposes the question over steps and enables a more accurate answer to the question.

Recurrent graph neural network. To enhance retrievals over steps, as shown in Fig. 3, we utilize an RGNN to establish relationships between the semantic distances of subquestions. Specifically, with the subquestion q^t at step t, we first compute $h_i^{L,t}$ for each node $v_i \in \mathcal{V}$ using the aforementioned L-layer GNN. To consider the effect of semantic distances from previous subquestions, we integrate $h_i^{L,t}$ with the previous-step integrated semantic distance to compute the integrated semantic distance $\hat{h}_i^{L,t}$ at step t using the parameter β. This can be formulated as follows:

$$\hat{h}_i^{L,t} = \beta h_i^{L,t} + (1-\beta)\hat{h}_i^{L,t-1}, \quad (t > 1), \quad (4)$$

where $\hat{h}_i^{L,t-1}$ is the integrated semantic distance from the previous step $t-1$ and $\hat{h}_i^{L,1} = h_i^{L,1}$. The integrated semantic distance $\hat{h}_i^{L,t}$ will be used to integrate with $h_i^{L,t+1}$ of node v_i at the next step $t+1$ recurrently until the last step of the answering process T. GNN facilitates the integration of semantic distances between related passages at each step, and the recurrent parameter β enables the integration of semantic distances of different subquestions across steps, which effectively mitigates the impacts of incorrect subquestions and enhances the retrieval of supporting passages for them.

Hinge objective for RGNN. We adopt a hinge objective for RGNN by considering all the T answering steps for question q. With the subquestion q^t at step t, the primary objective of RGNN is to reduce the semantic distance of its corresponding supporting passage and make it retrieved. Furthermore, to enhance the retrieval process for subsequent answering steps, we also reduce the semantic distances of supporting passages that should appear at subsequent steps to make them retrieved subsequently. Specifically, at step t, we adopt a hinge objective that encourages the average integrated semantic distance of supporting passages that should appear at the current and subsequent steps $t+ = \{t, \cdots, T\}$ to be lower than that of the non-target node set. We first define the average integrated semantic distances of supporting passages and non-target nodes at step t as $\bar{d}^{L,t}_{Y^{t+}} = \frac{1}{|\mathcal{S}^t_{Y^{t+}}|}\sum_{j\in\mathcal{S}^t_{Y^{t+}}} \hat{h}^{L,t}_j$ and $\bar{d}^{L,t}_{O/Y^{t+}} = \frac{1}{|\mathcal{S}^t_{O/Y^{t+}}|}\sum_{o\in\mathcal{S}^t_{O/Y^{t+}}} \hat{h}^{L,t}_o$, respectively, where $\mathcal{S}^t_O$ is the competitive node set with top O smallest semantic distances, and $\mathcal{S}^t_{O/Y^{t+}}$ is the non-target node set by removing the target node set $\mathcal{S}^t_{Y^{t+}}$ from $\mathcal{S}^t_O$. The hinge objective function with threshold r for the RGNN is formulated as follows:

$$\ell = \frac{1}{T}\sum_{t\in[T]} \max(0, r + \bar{d}^{L,t}_{Y^{t+}} - \bar{d}^{L,t}_{O/Y^{t+}}). \quad (5)$$

By employing this hinge objective, the RGNN takes into account the retrieval of subsequent answering steps and proactively reduces the semantic distances associated with them. These small semantic distances are then transferred to the next subquestion and enhance the retrieval of the corresponding supporting passages. We update the RGNN (parameters α^l and β) using the gradient descend, which are elaborated in Appendix A.2.

4 Experiments

In this section, we conduct experiments to demonstrate the effectiveness of *GNN-Ret* and *RGNN-Ret*, including **1)** superiority of *GNN-Ret* and *RGNN-Ret* on various QA datasets (Table 1), **2)** ablation studies to verify the effectiveness of components in GoPs, *GNN-Ret*, and *RGNN-Ret* (Table 2 and 3), and **3)** superiority of *GNN-Ret* on open-sourced LMs (Table 4). Additional analysis on hyperparameter selection, statistics of retrieval accuracy, token number for retrieval, multi-layer *GNN-Ret*, extension of *RGNN-Ret* to other multi-hop answering methods, and qualitative results are discussed in Appendix C, D, E, F, G, H, respectively.

4.1 Experimental Setups

Evaluation datasets. We measure all the methods on four different QA datasets, including multi-hop Wikipedia reasoning datasets: (1) MuSiQue (Trivedi et al., 2022), (2) IIRC (Ferguson et al., 2020), (3) 2WikiMQA (Ho et al., 2020), and a single-hop multi-choice QA dataset: (4) Quality (Pang et al., 2021). As evaluation metrics, we calculate the F1 score, exact match (EM) and accuracy (Acc) for multi-hop reasoning datasets. We use the ChatGPT (`gpt-3.5-turbo-2023-06-01-preview`) to evaluate if the prediction matches with the gold answer, following (Shao et al., 2023; Wang et al., 2024b). Since Quality is not a multi-hop dataset, we only validate accuracy performance of *GNN-Ret* on it.

Baselines. We compare *GNN-Ret* and *RGNN-Ret* with the following baselines: (1) *Direct* answers questions without retrieved passages. (2)We use retrievers *bm25* (Robertson et al., 2009), *DPR* (Karpukhin et al., 2020a), and *SentenceBert*[1] (*SBERT*) (Reimers and Gurevych, 2019) to retrieve passages without considering relatedness between passages for QA. (3) *SelfAsk* (Press et al., 2022) prompts LLMs to generate the follow-up question and answers it with retrieved passages until generating the final answer. (4) *ITER-RETGEN* (Shao et al., 2023) iteratively answers questions with retrieved passages and uses the generated answer for the next-step retrieval until the generation of final answer. (5) *IRCoT* (Trivedi et al., 2023) iteratively prompts LLMs to generate chains of thoughts with retrieved passages and retrieves with the generated reasons until reaching the maximum token number. All the retrieved passages are then used to generate the final answer. (6) *KGP* (Wang et al., 2024b) first searches seed passages using the question and then prompts LLMs to generate the needed evidence to retrieve the other relevant passages among neighbors of seed nodes using semantic distances.

Implementation details. We use ChatGPT (`gpt-3.5-turbo-2023-06-01-preview`) as the LLM backbone for experiments and adopt a temperature of 0 to remove the effect of random sam-

[1] `https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1`

Methods	MuSiQue			IIRC			2WikiMQA			Quality		
	F1	EM	Acc	F1	EM	Acc	F1	EM	Acc	F1	EM	Acc
No Retrieval												
Direct	14.1	4.0	9.6	15.8	9.6	16.0	24.4	17.3	24.8	-	-	39.5
One-hop Retrieval												
bm25	22.2	10.2	18.8	27.3	15.4	31.7	33.8	23.8	34.6	-	-	47.0
DPR	23.8	10.8	17.5	29.9	16.3	35.6	36.2	24.4	36.3	-	-	52.5
SBERT	28.6	13.5	23.8	31.6	17.5	40.4	43.4	31.0	42.3	-	-	53.4
GNN-Ret (K=5)	**31.6**	16.5	27.1	32.2	17.6	44.0	47.7	32.7	44.8	-	-	55.5
GNN-Ret (K=10)	30.0	14.4	24.4	35.2	19.6	44.0	47.7	32.9	46.3	-	-	**58.2**
GNN-Ret (w. train)	31.1	**17.5**	**27.3**	**36.1**	**21.9**	**44.4**	**48.1**	**33.5**	**46.9**	-	-	-
Multi-hop Retrieval												
SelfAsk	28.0	18.3	22.9	37.1	**29.6**	36.9	51.2	41.5	44.8	-	-	-
ITER-RETGEN	30.0	16.7	27.1	32.6	20.0	39.6	45.6	33.3	43.8	-	-	-
IRCoT	29.9	12.5	27.5	32.5	18.8	41.0	46.0	30.8	44.6	-	-	-
KGP	30.0	15.4	24.0	33.9	18.5	42.1	45.5	31.9	45.2	-	-	-
RGNN-Ret	32.9	20.8	**31.3**	**43.4**	28.3	48.1	59.7	43.3	**55.8**	-	-	-
RGNN-Ret (w. train)	**34.8**	**21.9**	**31.3**	42.8	27.6	**48.4**	**61.4**	**45.2**	55.6	-	-	-

Table 1: F1/EM/Acc for different QA methods with ChatGPT on four QA datasets. The best and second best scores are highlighted in **bold** and underline, respectively

pling. We employ SBERT as the embedding model for all our methods and baselines. The semantic distance is derived by (1 − cosine similarity) following (Sarthi et al., 2024). We set up a maximum token number of 3500 for retrieval to leave the space to instruction and demonstrations, and all the methods retrieve the semantically similar passages until reaching the maximum token number. We adopt a one-layer GNN for our proposed methods *GNN-Ret* and *RGNN-Ret*. We set up $\alpha^1 = 0.5$ for *GNN-Ret* and $\alpha^1 = 0.5$ and $\beta = 0.9$ for *RGNN-Ret* when there is no training. When training the RGNN, we set up $r = 0.01$, $K = 5$, and $O = 25$ for *GNN-Ret* (w. train) and $r = 0.01$, $K = 5$, and $O = 10$ for *RGNN-Ret* (w. train), respectively, which achieve the best performance in grid search experiments in Appendix C. We do not train the parameters for Quality dataset without the ground-truth labels of supporting passages. Since training the RGNN requires ground-truth subquestions, we manually generate subquestions for 5 questions sampled from the preset training data. More implementation details are presented in Appendix B.

4.2 Main Results

Table 1 summarizes the results for our proposed methods and baselines on four QA datasets using ChatGPT. Results show that *GNN-Ret* significantly outperforms baselines in terms of F1, EM, and accuracy on all these four QA datasets with a single retrieval. For instance, compared with *SBERT*, *GNN-Ret* improves EM by 4.4 and accuracy by 4% on the IIRC dataset. Additionally, *GNN-Ret* maintains its superiority on the Quality dataset, which requires a comprehensive understanding of the context of an entire story. This can be attributed to the ability of *GNN-Ret* to enhance the retrieval prioritization of supporting passages through the incorporation of structural information. Surprisingly, our proposed one-hop *GNN-Ret* even slightly outperforms baselines with multi-hop retrieval processes and queries of LLMs in accuracy, highlighting significant potentials of leveraging passage relatedness to enhance the retrieval process for answering multi-hop reasoning questions.

Furthermore, results in Table 1 demonstrate that our proposed *RGNN-Ret* achieves state-of-the-art performance in terms of F1, EM, and accuracy on the three multi-hop reasoning datasets. *RGNN-Ret* outperforms the best baseline, *KGP*, by more than 10% in accuracy on 2WikiMQA. This performance improvement can be attributed to our proposed *self-critique* technique and RGNN. The *self-critique* technique enables more accurate judgement on the ending of the answering process and the generation of final answers. The incorporation of RGNN further enhances retrieval for each step of answering.

4.3 Additional Analysis

Effectiveness of components in *GNN-Ret*. To assess the individual contributions of each proposed component, we conduct ablation studies and present the results in Table 2. We explore an alternative approach by utilizing the mean semantic distance (*GNN-Ret (mean)*) instead of the minimum semantic distance as the message in equation (1)

Methods	MuSiQue	2WikiMQA
SBERT	23.8	42.3
GNN-Ret (mean)	26.3 (+2.5)	44.6 (+1.7)
GNN-Ret	**27.3** (+3.5)	**46.9** (+4.6)
Self-critique + SBERT	27.9	52.9
Self-critique + recurrent	28.1 (+0.2)	54.0 (+1.1)
Self-critique + GNN	30.8 (+2.9)	55.2 (+2.3)
RGNN-Ret	**31.3** (+3.4)	**55.6** (+2.7)
RGNN-Ret (Y)	27.1	53.3
RGNN-Ret (Y^t)	30.2 (+3.1)	54.6 (+1.3)
RGNN-Ret (Y^{t+})	**31.3** (+4.2)	**55.6** (+2.3)

Table 2: Ablation studies of components of GNN-Ret and RGNN-Ret in accuracy with ChatGPT on MuSiQue and 2WikiMQA.

for the GNN. This modification results in a lower accuracy compared to the use of the minimum semantic distance as the message, which suggests that employing the minimum semantic distance as the message effectively filters out interfering messages from irrelevant neighbors and preserves the most relevant message for the GNN.

Effectiveness of components in *RGNN-Ret*. We conduct ablation studies to evaluate the effectiveness of each component in *RGNN-Ret* (i.e., recurrent, GNN, and RGNN). Results in Table 2 show that both the recurrent part (parameter β) and GNN improve accuracy compared with setting with SBERT on MuSiQue and 2WikiMQA datasets. By combining the recurrent part and GNN, our proposed method *RGNN-Ret* further improves accuracy up to 31.3% and 55.6% on MuSiQue and 2WikiMQA datasets, respectively, which demonstrates the effectiveness of *RGNN-Ret* in enhancing the retrieval by interconnecting with the GoPs from previous steps. This interconnection enhances the retrieval of supporting passages across multiple steps, leading to more accurate answers in the multi-hop reasoning process.

We also explore different settings for the target node sets during the training of RGNN and present results in Table 2. When using the entire node set of ground-truth supporting passages $\mathcal{S}_Y$ as the retrieval labels for each subquestion, denoted as *RGNN-Ret* (Y), the accuracy decreases to 27.1% and 53.3% on MuSiQue and 2WikiMQA datasets, respectively. This is because the supporting passages for initial steps are irrelevant to the subquestions at subsequent steps. Including all of them as labels for each subquestion disrupts the training of RGNN. Instead, we employ the corresponding label for each subquestion, denoted as *RGNN-Ret* (Y^t). This approach effectively improves the accuracy to 30.2% and 54.6% on the MuSiQue and 2WikiMQA datasets, respectively. Futhremore, our proposed *RGNN-Ret* method includes label set of supporting passages not only for the current subquestion but also for subsequent subquestions. This comprehensive approach further improves the accuracy to 31.3% and 55.6% on MuSiQue and 2WikiMQA datasets, respectively, which verifies the effectiveness of *RGNN-Ret* in enhancing the retrieval process for subsequent subquestions.

Methods	2WikiMQA	Quality
SBERT (w.o. graph)	42.3	53.4
graph w. SI	42.7 (+0.4)	55.5 (+2.1)
graph w. SK	45.4 (+3.1)	54.8 (+1.4)
graph w. SI and SK	**46.9** (+4.6)	**58.2** (+4.8)

Table 3: Ablation studies of components on graph construction in accuracy with ChatGPT on 2WikiMQA and Quality. SI and SK represent structural information and shared keywords, respectively.

Effectiveness of GoPs. We explore the effectiveness of components (structural information (SI) and shared keywords (SK)) for graph construction. Results in Table 3 show that both structural information and shared keywords are able to improve accuracy for the baseline without GoPs. Notably, in the context-specific Quality story dataset, which demands a comprehensive grasp of the entire narrative for accurate retrieval, the graph that incorporates structural information markedly outperforms the setting that relies solely on shared keywords. While for 2WikiMQA that requires knowledge of multiple documents, the graph with shared keywords achieves a better accuracy compared with that with structural information. By combining them into graph construction, we achieve the best accuracy in these two datasets.

	MuSiQue			2WikiMQA		
Qwen2-7B-Instruct	F1	EM	Acc	F1	EM	Acc
SBERT	27.8	16.9	23.3	43.3	34.0	40.8
GNN-Ret	**32.3**	**20.8**	**26.7**	**50.5**	**37.3**	**45.8**
gemma-2-9b-it	F1	EM	Acc	F1	EM	Acc
SBERT	32.8	18.9	26.9	46.4	28.8	43.8
GNN-Ret	**34.6**	**20.8**	**29.0**	**51.6**	**30.8**	**45.8**

Table 4: F1/EM/Acc of SBERT and GNN-Ret with open-sourced LMs on MuSiQue and 2WikiMQA.

Performance of *GNN-Ret* with open-sourced LMs. Our proposed *GNN-Ret* utilizes the relatedness of passages to enhance retrieval with the meticulously designed GNNs, which is generalized and robust to all the LM architectures. To substantiate this, we supplement the experiments for *GNN-Ret* and SBERT with two well-behaved open-sourced LMs (Qwen/Qwen2-7B-Instruct (Yang et al., 2024) and google/gemma-2-9b-it (Team, 2024)) from Open LLM Leaderboard (Fourrier et al., 2024). Results in Table 4 show that *GNN-Ret* consistently outperforms SBERT across varying open-sourced LMs, demonstrating the generalization of *GNN-Ret* in enhancing retrieval coverage with varying LMs.

5 Conclusion

In this paper, we proposed *GNN-Ret*, an effective method to enhance retrieval for QA of LLMs by exploiting the inherent relatedness between passages on a graph of passages. By extending *GNN-Ret* to multi-hop reasoning questions, we proposed *RGNN-Ret*, which enhances the retrieval for subquestions through the interconnection between graphs of passages across steps. The experiments clearly demonstrated the superiority of both *GNN-Ret* and *RGNN-Ret* over baselines, highlighting the effectiveness of leveraging relatedness between passages to enhance retrieval. From these advantages, we believe that the incorporation of graph-based representations and the exploitation of passage relatedness can open up new avenues of research in the field of LLMs in understanding structural documents and answering complex questions.

6 Acknowledgements

This work was supported by the Hong Kong Research Grants Council under the NSFC/RGC Collaborative Research Scheme grant CRS_HKUST603/22.

7 Limitations

Costs of graph construction. While our proposed *GNN-Ret* demonstrates impressive accuracy in QA tasks with a single query of LLMs, it still relies on multiple LLM queries to extract keywords from passages and construct the graph of passages. Fortunately, there are alternative fine-tuned language models specific for keyword extraction[2][3]. These models are more compact and significantly accelerate the graph construction process. Moreover, the flexibility of the graph of passages allows for dynamic modifications, such as adding new passage nodes or removing outdated ones. By maintaining a domain-specific graph of passages, we can effectively address various questions for this domain without the need for reconstructing the graph.

Costs of message passing. We first analyze the statistics of the graph of passages in experiments and detail results in Table 5. In larger datasets, each node typically exhibits a higher number of neighbors, attributed to an increased likelihood of keyword sharing among more nodes. We calculate the density of the graph of passages, denoted as $D = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$, for each dataset in our experiments, based on the established definition (Pieterse and Black, 2005), where $\mathcal{E}$ and $\mathcal{V}$ represent the edges and vertices of the graph, respectively. The densities of all graphs are significantly less than 1, underscoring the substantial sparsity observed in these graphs across our experimental datasets.

Datasets	Number of nodes ($\|\mathcal{V}\|$)	Average number of edges ($\frac{\|\mathcal{E}\|}{\|\mathcal{V}\|}$)	Density
2WikiMQA	9815	91.38	0.018
IIRC	21866	259.22	0.024
MuSiQue	20071	335.77	0.033
Quality	1104	26.00	0.047

Table 5: Statistics of graph of passages in experiments.

We adopt relevant nodes sampling (Section 3.2) to allow the relevant node set with only top K smallest semantic distances to pass the information to their neighbors during message passing. Assuming an average number of neighbors $\mathcal{V}_K$ among these K relevant nodes, the time complexity for message passing can be derived as $O(K|\mathcal{V}_K|)$, considering a single-layer GNN. With K set to 5 or 10 in our experiments, and given the small empirical average number of edges per node $\frac{|\mathcal{E}|}{|\mathcal{V}|}$ reported in Table 5, the time complexity of message passing is neglectable compared with LLM queries.

Undirected Graph. In this study, our focus lies solely on improving the retrieval process through an undirected graph of passages, with no additional information associated with the edges. While we employ a sampling technique to select relevant nodes and utilize the minimum semantic distance as the message in the GNN, we acknowledge that there is still a possibility of irrelevant passages receiving messages, potentially impacting the retrieval performance. Consequently, the task of selecting an appropriate path on the graph of passages that aligns with the given question remains an area that is yet to be thoroughly explored. In addition, we only consider the structural-related passages that are in the same section. However, some questions may require more complex structural information. For example, when the question is about comparison of two entities in different sections, more complex structural information is needed to connect these corresponding passages to enhance retrieval. Therefore, a more delicate graph of passages for more complex tasks warrant further

[2] https://huggingface.co/ml6team/keyphrase-extraction-distilbert-inspec

[3] https://github.com/MaartenGr/KeyBERT

investigation in future studies.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106.

Wentao Ding, Jinmao Li, Liangchuan Luo, and Yuzhong Qu. 2024. Enhancing complex question answering over knowledge graphs through evidence pattern retrieval. *arXiv preprint arXiv:2402.02175*.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2024. Retrieval-generation synergy augmented large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11661–11665. IEEE.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. Iirc: A dataset of incomplete information reading comprehension questions. *arXiv preprint arXiv:2011.07127*.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

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*, 33:9459–9474.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. 2023a. Chain of knowledge: A framework for grounding large language models with structured knowledge bases. *arXiv preprint arXiv:2305.13269*.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2023b. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*.

LINHAO LUO, Yuan-Fang Li, Reza Haf, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2021. Quality: Question answering with long input texts, yes! *arXiv preprint arXiv:2112.08608*.

Jinyoung Park, Ameen Patel, Omar Zia Khan, Hyunwoo J Kim, and Joo-Kyung Kim. 2023. Graph-guided reasoning for multi-hop question answering in large language models. *arXiv preprint arXiv:2311.09762*.

V Pieterse and PE Black. 2005. Dictionary of algorithms and data structures. *URL: http://www. nist. gov/dads/HTML/greedyalgo. html*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Gemma Team. 2024. Gemma.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. 2023. Boosting language models reasoning with chain-of-knowledge prompting. *arXiv preprint arXiv:2306.06427*.

Shu Wang, Lei Ji, Renxi Wang, Wenxiao Zhao, Haokun Liu, Yifan Hou, and Ying Nian Wu. 2024a. Explore the reasoning capability of llms in the chess testbed. *arXiv preprint arXiv:2411.06655*.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024b. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19206–19214.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631.

Siheng Xiong, Yuan Yang, Ali Payani, James C Kerce, and Faramarz Fekri. 2024. Teilp: Time prediction over knowledge graphs via logical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16112–16119.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

A Training Details

In this section, we supplement the training details of GNN and RGNN as below.

A.1 Training of GNN

Hinge objective for GNN. Recall that the objective of GNN is to reduce the semantic distances of supporting passages more than those of non-target nodes. With the ground-truth set of supporting passages $\mathcal{S}_Y$, we first define the average integrated semantic distances of supporting passages $\bar{d}_Y^L$ and non-target nodes $\bar{d}_{O/Y}^L$ as below:

$$\bar{d}_Y^L = \frac{1}{|\mathcal{S}_Y|} \sum_{j \in \mathcal{S}_Y} h_j^L, \tag{6}$$

$$\bar{d}_{O/Y}^L = \frac{1}{|\mathcal{S}_{O/Y}|} \sum_{o \in \mathcal{S}_{O/Y}} h_o^L, \tag{7}$$

where S_O is the competitive node set with the top O smallest semantic distances and $S_{O/Y}$ is the non-target node set by removing the target node set S_Y from S_O, respectively. We only consider the competitive node set since the average integrated semantic distance of all the nodes is significantly large and loses effectiveness in training. With the output average semantic distances of supporting passages and non-target nodes, we formulate the hinge objective function with threshold r as follows:

$$\ell = \max(0, r + \bar{d}_Y^L - \bar{d}_{O/Y}^L). \tag{8}$$

We update the parameters of GNN using gradient descent, and the backward process is presented as below. With only partial nodes accepting messages from neighbors, we first define the received node set at each layer l as follows:

$$\mathcal{S}_R^l = \begin{cases} \mathcal{S}_O \cap \mathcal{S}_K^L, & l = L \\ \mathcal{S}_K^l, & \text{else} \end{cases}. \tag{9}$$

For each node $i \in \mathcal{S}_R^L$ at the last layer L, the gradient of ℓ w.r.t. h_i^L is:

$$\frac{\partial \ell}{\partial h_i^L} = \begin{cases} 1.0, & i \in \mathcal{S}_Y \\ -1.0, & \text{else} \end{cases}. \tag{10}$$

For the layer $l < L$, the gradient of l w.r.t. α^l is presented as follows:

$$\frac{\partial \ell}{\partial \alpha^l} = \sum_{j \in \mathcal{S}_R^l} \frac{\partial \ell}{\partial h_i^l} (h_i^{l-1} - m_i^l). \tag{11}$$

Assuming that node $i \in \mathcal{T}_k^{l-1}$ at layer $l-1$ transfers the message to the node set $\mathcal{S}_{K,i}^l$ at layer l (one node can transfers messages to multiple neighbors), we update the gradient of α^l w.r.t. h_i^{l-1} at the previous layer $l-1$ as follows:

$$\frac{\partial \ell}{\partial h_i^{l-1}} = \sum_{j \in \mathcal{S}_{K,i}^l} -\alpha^l \frac{\partial \ell}{\partial h_j^l} + \mathbb{1}(i \in \mathcal{S}_{K,i}^l) \alpha^l \frac{\partial \ell}{\partial h_j^l}. \tag{12}$$

By propagating the gradients backward to the first layer, we can obtain the gradients for each layer of α.

A.2 Training of RGNN

Hinge objective for RGNN. For each subquestion q^t at step t, we input its semantic distances into GNN and compute hidden state $h_j^{l,t}$ and message $m_j^{l,t}$ of the $j-$th node for $j \in \{1, \cdots, |\mathcal{V}|\}$ at layer l. Recall that we adopt a hinge objective that encourages the average integrated semantic distance of supporting passages that should appear at the current and subsequent steps $t+ = \{t, \cdots, T\}$ to be lower than that of the non-target node set for subquestion q^t at step t. We first define the average semantic distances of supporting passages $\bar{d}_{Y^{t+}}^{L,t}$ and non-target nodes $\bar{d}_{O/Y^{t+}}^{L,t}$ at step t as below:

$$\bar{d}_{Y^{t+}}^{L,t} = \frac{1}{|\mathcal{S}_{Y^{t+}}^t|} \sum_{j \in \mathcal{S}_{Y^{t+}}^t} \hat{h}_j^{L,t}, \tag{13}$$

$$\bar{d}_{O/Y^{t+}}^{L,t} = \frac{1}{|\mathcal{S}_{O/Y^{t+}}^t|} \sum_{o \in \mathcal{S}_{O/Y^{t+}}^t} \hat{h}_o^{L,t}, \tag{14}$$

where $\mathcal{S}_O^t$ is the competitive node set with top O smallest semantic distances at step t and $\mathcal{S}_{O/Y^{t+}}^t$ is the non-target node set by removing the target node set $\mathcal{S}_{Y^{t+}}^t$ from $\mathcal{S}_O^t$, respectively. We formulate the hinge objective function with threshold r for the RGNN as follows:

$$\begin{aligned} \ell &= \frac{1}{T} \sum_{t \in [T]} \ell^t \\ &= \frac{1}{T} \sum_{t \in [T]} \max(0, r + \bar{d}_{Y^{t+}}^{L,t} - \bar{d}_{O/Y^{t+}}^{L,t}). \end{aligned} \tag{15}$$

We update the parameters of RGNN using gradient descent, and the backward process is presented as below. With only partial nodes accepting messages from neighbors, we first define the recieved node set $\mathcal{S}_R^{l,t}$ at layer l at step t as follows:

$$\mathcal{S}_R^{l,t} = \begin{cases} \mathcal{S}_O^t \cap \mathcal{S}_K^{L,t}, & l = L \\ \mathcal{S}_K^{l,t}, & \text{else} \end{cases}. \tag{16}$$

For each node $i \in \mathcal{S}_R^{L,t}$ at the last layer L, the gradient of ℓ^t w.r.t. $\hat{h}_i^{L,t}$ is:

$$\frac{\partial \ell^t}{\partial \hat{h}_i^{L,t}} = \begin{cases} 1.0, & i \in \mathcal{S}_{Y^t}^t \\ -1.0, & \text{else} \end{cases}. \tag{17}$$

Therefore, the gradient of ℓ w.r.t. $\hat{h}_i^{L,t}$ is:

$$\frac{\partial \ell}{\partial \hat{h}_i^{L,t}} = \sum_{\tau=t}^{T} \frac{\partial \ell^\tau}{\partial \hat{h}_i^{L,t}} = \frac{\partial \ell^t}{\partial \hat{h}_i^{L,t}} + \frac{\partial \ell}{\partial \hat{h}_i^{L,t+1}}(1-\beta), \tag{18}$$

where

$$\frac{\partial \ell}{\partial \hat{h}_i^{L,T}} = \frac{\partial \ell^T}{\partial \hat{h}_i^{L,T}}. \tag{19}$$

Each node $j \in \mathcal{V}$ at step t transfers the message to the same node at the next step through β. Therefore, the gradient of loss ℓ with respect to β is computed by:

$$\frac{\partial \ell}{\partial \beta} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|\mathcal{V}|} \sum_{j \in \mathcal{V}} \frac{\partial \ell}{\partial \hat{h}_j^{L,t}} (h_j^{L,t} - \hat{h}_j^{L,t-1}). \tag{20}$$

Recall that $\hat{h}_i^{L,t} = \beta h_i^{L,t} + (1-\beta) h_i^{L,t-1}$ for each node $i \in \mathcal{V}$. The gradient of loss ℓ with respect to $h_i^{L,t}$ is computed by:

$$\frac{\partial \ell}{\partial h_i^{L,t}} = \beta \frac{\partial \ell}{\partial \hat{h}_i^{L,t}}. \tag{21}$$

With $\frac{\partial \ell}{\partial h_i^{L,t}}$, we propagate the gradient backward to the GNN and update the parameter α^l layer by layer using (11) and (12).

B Additional Experimental Details

Datasets processing. We measure all the methods on four different QA datasets, including multi-hop Wikipedia reasoning datasets: (1) MuSiQue (Trivedi et al., 2022), (2) IIRC (Ferguson et al., 2020), (3) 2WikiMQA (Ho et al., 2020), and a single-hop multi-choice QA dataset: (4) Quality (Pang et al., 2021). For each multi-hop dataset, we randomly sample 500 questions from the development set, which provides the ground-truth supporting passages. We use 20 of them to train the GNN and select the rest as test data. We use all the Wikipedia documents of these 500 questions as the retrieval corpus, which is significantly larger than that provided by datasets[4]. For Quality, we randomly sample 30 articles with overall corresponding 561 questions and use these articles as the retrieval corpus. For each document or article, we split it into multiple passages with a maximum token length of 200 (Sarthi et al., 2024).

Graph of passages construction. We constructe a large graph of passages to serve as the retrieval corpus for each evaluation dataset. For the Quality dataset, we randomly sample 30 articles with a diverse range of topics as the retrieval corpus. For multi-hop reasoning datasets, we first collect the Wikipedia documents using the provided titles required for answering these 500 questions. For the multi-hop reasoning datasets, we collect Wikipedia documents based on the provided titles required to answer the 500 questions. We chunk the documents into smaller passages and record their sequential order. Passages that are physically adjacent to each other are regarded as structure-related and connected together in the graph. Besides, to find out the keyword-related passages, we also extract the keywords for each passage and then connect those that share the same keywords. Specifically, we prompt ChatGPT (`gpt-3.5-turbo-2023-06-01-preview`) to extract the Wikipedia keywords from the passages and generate their corresponding links. The links are used to ensure the consistency of the Wikipedia document that use different keywords in passages. The prompt for keyword extraction is given as follows:

> Prompt for extracting keywords
>
> Instruction: Extract the entities exist in this text and then provide the wikipedia links for the entities exist in this text. Output the entities and their wikipedia links in the list format, e.g., [['entity1', 'link1'], ['entity2', 'link2']].
>
> <Passage>

[4] Prior works often use 10–20 paragraphs or documents as the retrieval corpus for each question (Shao et al., 2023; Wang et al., 2024b).

The passages share the same links of keywords are considered as keyword-related and connected together in the graph.

Implemental details of training. We initialize $\alpha = 0.1$ for GNN-Ret and $\alpha^1 = 1.0$ and $\beta = 1.0$ for RGNN-Ret, respectively. We then use gradient descent to update the GNN and RGNN. For each iteration, we compute the average gradient of all the training samples and update the parameters with a learning rate of 1.0. The training will stop when the absolute value of gradient is smaller than 0.001 or the loss keeps increasing for five consecutive iterations. As there is often the whole label set of supporting passages for the question but not the individual index for each subquestion, we select the one with the lowest semantic distance as the label for each subquestion and then remove this index label from the label set for subsequent subquestions. Since training the RGNN requires ground-truth subquestions, we manually generate subquestions for 5 questions sampled from the preset training data. The training samples of MuSiQue, IIRC, and 2WikiMQA datasets for the RGNN are displayed in Appendix J. We also display the instructions and prompt templates of all the methods in Appendix I.

C Selection of K and O.

In order to effectively train the GNN and RGNN models, we conduct grid search for hyperparameters K and O, which determine the number of relevant nodes to be sampled and the size of the competitive node set, respectively. To assess the impact of different values of K and O, we evaluate the accuracy of *GNN-Ret* and *RGNN-Ret* on the MuSiQue and IIRC datasets with varying K and O. The average accuracy across these two datasets is represented by the green points in Fig. 4. Results demonstrate that both *GNN-Ret* and *RGNN-Ret* consistently achieve stable accuracy performance across different settings of K and O. Upon analyzing the results, we observe that *GNN-Ret* achieves the highest accuracy when $K = 5$ and $O = 25$, while *RGNN-Ret* performs best with $K = 5$ and $O = 10$. *GNN-Ret* adopts a larger value of O since it considers the whole label set during training, while *RGNN-Ret* only considers the labels for the current step and subsequent steps. Therefore, we select these settings of K and O to train the GNN and RGNN models for experiments reported in Table 1.

D Statistics of retrieval accuracy.

We explore the EM between the retrieved passages and the ground-truth passages and collect the exact-match number of test samples with varying numbers of supporting passages required in questions on MuSiQue and 2WikiMQA datasets. We do not evaluate the statistics of retrieval accuracy for *RGNN-Ret* without the specific ground-truth supporting passages for each subquestion. Results in Fig. 5 show that our proposed *GNN-Ret* achieves higher retrieval accuracy compared with *SBERT* on MuSiQue and 2WikiMQA datasets. For instance, *GNN-Ret* outperforms *SBERT* by 29 and 43 exact-match test samples for the questions that require 2 supporting passages. This demonstrates that the GNN indeed improves retrieval coverage of supporting passages and the retrieval prioritization of supporting passages for subsequent reasoning.

Token number for retrieval	1k	3k	5k	10k	20k
SBERT	18.3	23.8	22.5	22.5	27.9
GNN-Ret	**20.4**	**25.8**	**25.8**	**26.7**	**31.9**

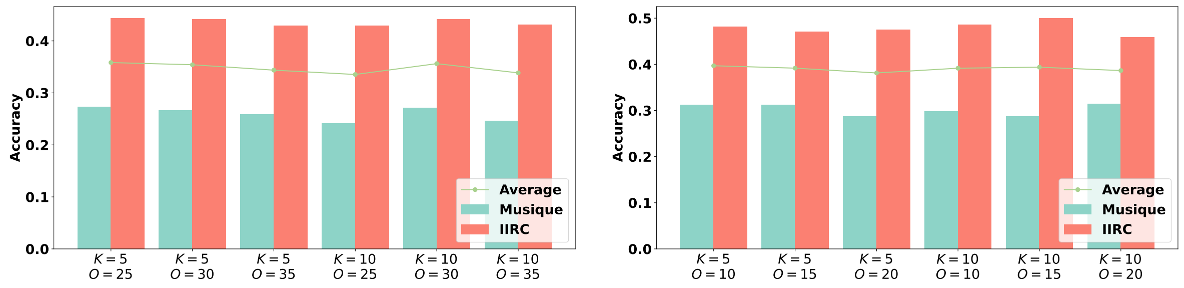
Table 6: Accuracy of GNN-Ret and SBERT with varying token numbers for retrieval using Qwen/Qwen2-7B-Instruct on MuSiQue dataset.

E Ablation Studies of Token Number for Retrieval

We conduct experiments with varying numbers of tokens for retrieval on *SBERT* and *GNN-Ret* with the long-context model Qwen/Qwen2-7B-Instruct (Yang et al., 2024), which can handle more than 100k tokens. Results in Table 6 show that our proposed *GNN-Ret* consistently outperforms *SBERT* with varying numbers of tokens for retrieval, demonstrating the robustness of our method for long contexts.

F Extending GNN-Ret to Multiple Layers

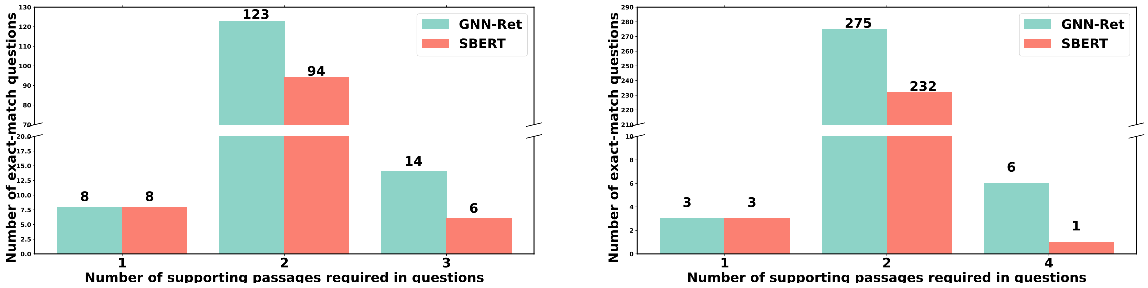
Our proposed GNNs are designed to be extensible across multiple layers. To explore this capability, we conducted additional experiments by increasing the number of GNN layers. The results, along with the final learned values of α, are presented in Table 7. The results indicate that *GNN-Ret* maintains consistent performance across different layer configurations. This observation could be attributed to the fact that many questions in datasets require only two-hop supporting passages, suggesting that



(a) GNN-Ret

(b) RGNN-Ret

Figure 4: Accuracy of *GNN-Ret* and *RGNN-Ret* with various K and O in Musique and IIRC datasets. The average accuracy of two datasets are displayed in green points.



(a) MuSiQue

(b) 2WikiMQA

Figure 5: Exact-match number of test samples with varying numbers of supporting passages required for QA on MuSiQue and 2WikiMQA datasets.

a single-layer GNN is adequately equipped to address these queries.

G Effectiveness of RGNN-Ret for other multi-hop answering methods

To further demonstrate the effectiveness of *RGNN-Ret*, we supplement experiments by employing it with other multi-hop baselines *IRCoT* and *ITER-RETGEN*. These methods utilize the generated reasons for retrieval, followed by the generation of the next-step reason or the final answer. Results in Table 8 show that *RGNN-Ret* consistently and significantly improves the accuracy performance for other multi-hop baselines, which demonstrates the effectiveness and adaptability of *RGNN-Ret* in enhancing the retrieval performance for multi-hop answering processes.

H Qualitative Results

We analyze the qualitative results in experiments and demonstrate the effectiveness of *GNN-Ret* and *RGNN-Ret* in improving accuracy for QA in this section.

***GNN-Ret* improves the retrieval coverage of supporting passages.** We compare the retrieval process between *SBERT* and *GNN-Ret* and display the results on Table 9. *GNN-Ret* can retrieve all of the supporting passages while *SBERT* fails in both of these two cases. For the first question, *SBERT* can retrieve the knowledge that '*the director of file Hotel By The Hour is Rolf Olsen.*'. Since it considers passages in isolation during retrieval, the supporting passage for the inquiry information ('*which country is Rolf Olsen from*') cannot be retrieved with a poor semantic distance. In contrast, *GNN-Ret* is able to retrieve both of the supporting passages about 'the director of file Hotel By The Hour is Rolf Olsen' and 'Rolf Olsen was an Austrian actor'. Consequently, it can output the correct final answer. This is attributed to the fact that *GNN-Ret* takes relatedness between passages into account and thus allows the supporting passages for inquiry information to accept the effect of semantic distances from those for background information, thereby improving the retrieval coverage of supporting passages.

***RGNN-Ret* better determines the terminal of answer process compared with *SelfAsk*.** We analyze the qualitative results for *RGNN-Ret* and *SelfAsk* since they have the similar answering procedures by generating subquestions and use them for retrieval. The qualitative results of them are shown in Table 10. For the first question '*Which award the performer of song One More Time (Joe Jackson Song) earned?*', *SelfAsk* generates the first-step subquestion and also answer it correctly. It obtains the knowledge that 'the performer of the song One More Time (Joe Jackson Song) is Joe

GNN-Ret	Learned values of α	MuSiQue			Learned values of α	2WikiMQA		
		F1	EM	Acc		F1	EM	Acc
1 layer	$\alpha = [0.326]$	31.1	17.5	27.3	$\alpha = [0.277]$	48.1	33.5	46.9
2 layers	$\alpha = [0.564, 0.821]$	31.5	16.3	27.3	$\alpha = [0.582, 0.578]$	47.0	33.3	46.5
3 layers	$\alpha = [0.506, 0.511, 0.944]$	30.8	16.3	26.7	$\alpha = [0.504, 0.597, 0.802]$	48.6	34.4	47.5

Table 7: F1, EM, and Accuracy of GNN-Ret with varying layers on MuSiQue and 2WikiMQA datasets.

Method	MuSiQue	2WikiMQA
IRCoT	27.5	44.6
IRCoT + RGNN-Ret	30.4 (+2.9)	47.5 (+2.9)
ITER-RETGEN	27.1	43.8
ITER-RETGEN + RGNN-Ret	29.8 (+2.7)	47.5 (+3.7)

Table 8: Accuracy of IRCoT (w. RGNN-Ret) and ITER-RETGEN (w. RGNN-Ret) on MuSiQue and 2WikiMQA datasets.

Jackson'. However, it terminates the answering process at this step and thus outputs the incorrect final answer. In contrast, *RGNN-Ret* understands that the generated intermediate answers are not sufficient to output the final answer and thus continues the next-step answering. Consequently, it can output the final answer '*Grammy*'. This qualitative comparison between *RGNN-Ret* and *SelfAsk* demonstrates the effectiveness of our proposed *self-critique* technique in determining the termination of the answering process and improving the accuracy for QA.

***RGNN-Ret* enhances the retrieval process for subquestions.** For the second question '*What other notable work did the creator of Shrek make?*' in Table 10, both *SelfAsk* and *RGNN-Ret* are able to correctly answer the first subquestion and generate the second-step subquestion 'What other notable works did William Steig create?'. However, *SelfAsk* cannot retrieve the knowledge about 'other work of William Steig', which locates in another passage about the book 'Doctor De Soto'. Using SBERT for retrieval fails to retrieve this passage for QA. In contrast, *RGNN-Ret* retrieve this passage since it enhances the semantic distance by integrating with the small semantic distances from the previous step. Consequently, *RGNN-Ret* outputs the correct answer. This qualitative example demonstrates that the RGNN can enhance the retrieval over steps for subquestions and thus improve the accuracy for QA.

I Prompts for Experiments

We display prompt templates of all the methods in this section.

I.1 Prompt template for Direct.

The prompt template for *Direct* is shown as follows:

> **Prompt template for Direct**
>
> Instruction: Given the following question, create a final answer to the question. Please answer less than 6 words.
>
> `<Question>`

where `<Question>` indicates the question.

I.2 Prompt templates for dense retrievers

The prompt template for retrievers (bm25, DPR, and SBERT) is shown as follows:

> **Prompt template for retrievers**
>
> Instruction: Given the following question, create a final answer to the question. Please answer less than 6 words.
>
> Context:
> `<Context>`
>
> Question:
> `<Question>`

where `<Context>` indicates the retrieved passages.

We repeat the question before the length retrieved passages when answering the questions on IIRC and 2WikiMQA datasets for better performance.

I.3 Prompt templates for IRCoT and KGP

IRCoT and KGP employ different retrieval methods but the same prompt templates for QA. The prompt template is shown as follows:

Question: Which country the director of film Hotel By The Hour is from? **Ground-truth answer:** Austria
SBERT
Retrieved passages: (1) Hotel by the Hour (German title:) is a 1970 West German crime film directed by Rolf Olsen and starring Curd Jürgens, Andrea Rau and Corny Collins. (2) ... **Final answer:** Rolf Olsen is from Germany.
GNN-Ret
Retrieved passages: (1) Hotel by the Hour (German title:) is a 1970 West German crime film directed by Rolf Olsen and starring Curd Jürgens, Andrea Rau and Corny Collins. (2) Rolf Olsen (26 December 1919 – 3 April 1998) was an Austrian actor. **Final answer:** Rolf Olsen is from Austria.

Question: What is the date of death of the performer of song Goodbye Pork Pie Hat? **Ground-truth answer:** January 5, 1979
SBERT
Retrieved passages: (1) 'Goodbye Pork Pie Hat' is a jazz standard composed by Charles Mingus, originally recorded by his sextet in 1959 as listed below, and released on his album. (2) ... **Final Answer:** February 17, 2017
GNN-Ret
Retrieved passages: (1)'Goodbye Pork Pie Hat' is a jazz standard composed by Charles Mingus, originally recorded by his sextet in 1959 as listed below, and released on his album. (2) Charles Mingus Jr. (April 22, 1922 – January 5, 1979) was an American jazz double bassist, pianist, composer and bandleader. **Final answer:** January 5, 1979

Table 9: Two examples demonstrating effectiveness of *GNN-Ret*. We highlight the shared keywords, correct answers, and incorrect answers in violet, green, and red.

Prompt template for IRCoT / KGP

Instruction: Given the following question, create a final answer to the question. Please answer less than 6 words.

Context:
`<Context>`

Question:
`<Question>`

where `<Context>` indicates the retrieved passages.

We repeat the question before the length retrieved passages when answering the questions on IIRC and 2WikiMQA datasets for better performance.

Question: Which award the performer of song One More Time (Joe Jackson Song) earned? **Ground-truth answer:** Grammy
SelfAsk
Generation: Here is the question: Question: Which award the performer of song One More Time (Joe Jackson Song) earned? Are follow up questions needed here: Yes. Follow up: Who is the performer of the song One More Time (Joe Jackson Song)? The performer of the song ""One More Time"" (Joe Jackson Song) is Joe Jackson. So the final answer to the Question is: Joe Jackson
RGNN-Ret
Generation: Question: Which award the performer of song One More Time (Joe Jackson Song) earned? Next step: What is the name of the performer of the song ""One More Time (Joe Jackson Song)""? Intermediate Answer: The performer of the song ""One More Time"" is Joe Jackson. Next step: What award did Joe Jackson earn for the song ""One More Time""? Intermediate Answer: Joe Jackson earned a Grammy for Best Pop Instrumental Album in 2001 for his album Symphony No. 1. Final answer: Grammy

Question: What other notable work did the creator of Shrek make? **Ground-truth answer:** Doctor De Soto
SelfAsk
Generation: Question: What other notable work did the creator of Shrek make? Are follow up questions needed here: Yes. Follow up: Who is the creator of Shrek? The creator of Shrek is William Steig. Are follow up questions needed here: Yes. Follow up: What other notable work did William Steig make? William Steig also wrote "Sylvester and the Magic Pebble." So the final answer is: Sylvester and the Magic Pebble
RGNN-Ret
Generation: Question: What other notable work did the creator of Shrek make? Next step: Who is the creator of Shrek? Intermediate Answer: The creator of Shrek is American author William Steig. Next step: What other notable works did William Steig create besides Shrek? Intermediate Answer: William Steig, the creator of Shrek, also wrote and illustrated the children's picture book "Doctor De Soto" which was first published in 1982... Final answer: Doctor De Soto

Table 10: Two generation examples of comparison between *RGNN-Ret* and *SelfAsk*. We highlight the correct answers and incorrect answers in green and red. We omit the retrieved passages in the table.

I.4 Prompt template for SelfAsk

SelfAsk prompts LLMs to generate the follow-up question and answers it with retrieved passages until generating the final answer. The retrieved passages are included into the prompt when the LLMs are prompted to answer the follow-up question.

Prompt template for SelfAsk

Instruction: Your goal is to answer the question step by step following procedures of examples. If no follow up questions are necessary, answer the question in five words directly in format: So the final answer is: xxx.

Here are the examples of how to answer the questions:
`<Examples>`

Context (Optional):
`<Context>`

Question:
`<Question>`

We repeat the question before the length retrieved passages when answering the follow-up questions on IIRC and 2WikiMQA datasets for better performance.

I.5 Prompt template for ITER-RETGEN

ITER-RETGEN iteratively answers questions with retrieved passages and uses the generated answer for the next-step retrieval, which continues until the generation of final answer. The prompt template is shown as follows:

Prompt template for ITER-RETGEN

Instruction: Given the following question, create a final answer to the question. Please answer less than 6 words.

Here are the examples of how to answer the questions:
`<Examples>`

Context:
`<Context>`

Question:
`<Question>`

Let's think step by step.

I.6 Prompt template for RGNN-Ret

RGNN-Ret iteratively generates the next-step subquestion, answers the subquestion, and performs self-critique until the generation of final answer. The prompt template of these procedures are shown as follows:

Prompt template for generating next-step subquestions

Instruction: Your goal is to ask the next step question logically.

Here are the examples of how to answer the questions:
`<Examples>`

Question:
`<Question>`

Prompt template for generating subanswers

Instruction: Your goal is to answer the next step question. I will provide you some wikipedia snippets, and you need to answer the next step question by considering the wikipedia snippets.

Context:
`<Context>`

Question:
`<Question>`

We repeat the question before the length retrieved passages when answering the next-step subanswer on IIRC and 2WikiMQA datasets for better performance.

Prompt template of self-critique

Instruction: You are a wikipedia QA expert. Your goal is to critique whether the intermediate answers are enough to generate the final answer to the question.First analyze if the intermediate answers is enough to generate the final answer step by step logically. Then, if it is enough, output 'Critique: yes'. If not, output 'Critique: no'.

Question:
`<Question>`

"Analyze if the intermediate answers is enough to generate the final answer to the question step by step logically. Then, if it is enough, output 'Critique: yes'. If not, output 'Critique: no'.

The demonstrations of *RGNN-Ret* for asking the next-step subquestion on MuSiQue, IIRC, and 2WikiMQA datasets are shown as follows:

Demonstrations of *RGNN-Ret* for asking next-step subquestions on MuSiQue dataset

Question: Who lived longer, Muhammad Ali or Alan Turing?
Next step: How old was Muhammad Ali when he died?
Intermediate answer: Muhammad Ali was 74 years old when he died.
Next step: How old was Alan Turing when he died?

Question: When was the founder of craigslist born?
Next step: Who was the founder of craigslist?
Intermediate answer: Craigslist was founded by Craig Newmark.
Next step: When was Craig Newmark born?

Question: Who was the maternal grandfather of George Washington?
Next step: Who was the mother of George Washington?
Intermediate answer: The mother of George Washington was Mary Ball Washington.
Next step: Who was the father of Mary Ball Washington?

Question: Are both the directors of Jaws and Casino Royale from the same country?
Next step: Who is the director of Jaws?
Intermediate Answer: The director of Jaws is Steven Spielberg.
Next step: Where is Steven Spielberg from?
Intermediate Answer: The United States.
Next step: Who is the director of Casino Royale?
Intermediate Answer: The director of Casino Royale is Martin Campbell.
Next step: Where is Martin Campbell from?

Demonstrations of *RGNN-Ret* for asking next-step subquestions on IIRC dataset

Question: Who lived longer, Muhammad Ali or Alan Turing?
Next step: How old was Muhammad Ali when he died?
Intermediate answer: Muhammad Ali was 74 years old when he died.
Next step: How old was Alan Turing when he died?

Question: When was the founder of craigslist born?
Next step: Who was the founder of craigslist?
Intermediate answer: Craigslist was founded by Craig Newmark.
Next step: When was Craig Newmark born?

Question: Was the city where Eva was born the capital of its country?
Next step: Where was Eva born?
Intermediate answer: Eva was born in Berlin, Germany.
Next step: Is Berlin the capital of Germany?

Question: Was Ryuji Yamakawa a good solo wrestler?
Next step: Who is Ryuji Yamakawa?
Intermediate answer: Ryuji Yamakawa is a Japanese professional wrestler.
Next step: Was Ryuji Yamakawa a good solo wrestler?

Demonstrations of *RGNN-Ret* for asking next-step subquestions on 2WikiMQA dataset

Question: Who lived longer, Muhammad Ali or Alan Turing?
Next step: How long did Muhammad Ali live?
Intermediate answer: Muhammad Ali was 74 years old when he died.
Next step: How long did Alan Turing live?

Question: Who was the paternal grandfather of Princess Alexandrine Of Prussia (1842-1906)?
Next step: Who was the father of Princess Alexandrine Of Prussia (1842-1906)?
Intermediate answer: Princess Alexandrine Of Prussia (1842-1906) was the daughter of Prince Albert of Prussia.
Next step: Who was the father of Prince Albert of Prussia?

Question: Who is the mother of the composer of film 404 (Film)?
Next step: Who is the composer of film 404 (Film)?
Intermediate answer: The composer of film 404 (Film) is Ilayaraja.
Next step: Who is the mother of Ilayaraja?

Question: Do both films Across The Badlands and A Gutter Magdalene have the directors that share the same nationality?
Next step: What is the nationality of the director of Across The Badlands?
Intermediate answer: The director of Across The Badlands is American.
Next step: What is the nationality of the director of A Gutter Magdalene?
Intermediate answer: The director of A Gutter Magdalene is American.
Next step: Do both films Across The Badlands and A Gutter Magdalene have the directors that share the same nationality?

Question: What is the home stadium of the team that Asprey hit a hat trick against on 16 January 1961?
Next step: Which team did Asprey hit a hat trick against on 16 January 1961?
Intermediate answer: Asprey hit a hat trick against Charlton Athletic on 16 January 1961.
Next step: What is the home stadium of Charlton Athletic?

J Training data for RGNN

We manually generate subquestions for 5 questions sampled from the preset training data for MuSiQue, IIRC, and 2WikiMQA datasets. The training samples are shown as follows:

Training data of MuSiQue for RGNN

1. Question: Why did Roncali leave the place of death of the creator of Malchiostro Annunciation?

Subquestions:
Who is the creator of Malchiostro Annunciation?
Where did Titian die?
Why did Roncali leave Venice?

2. Question: Where did who argued that the country of citizenship of Victor Denisov had itself beome an imperialist power declare that he would intervene in the Korean conflict?

Subquestions:
What is the country of citizenship of Victor Denisov?
Who argued that Russia had itself become an imperialist power?
Where did Mao Zedong declare that he would intervene in the Korean conflict?

3. Question: What military overran much of Erich Zakowski's place of birth?

Subquestions:
What is the place of birth of Erich Zakowski?
What military overran East Prussia?

4. Question: How many people were in British Colonies where does the london broil cut come from ?

Subquestions:
Where does the london broil cut come from?
How many people were in North American?

5. Question: When was the country established that lies immediately north of the Persian Gulf and the region where the country containing Urim is located?

Subquestions:
What is the region containing Urim?
Where is the region that Iraq is located?
What is the country that lies immediately north of the Persian Gulf and the Middle East?
When was Iran established?

Training data of IIRC for RGNN

1. Question: Was the city where Eva was born the capital of its country?

Subquestions:
Where was Eva born?
Is Berlin the capital of its country?

2. Question: In what state did Galambos attend high school?

Subquestions:
What high school did Galambos attend?
In what state is Athens High School located?

3. Question: What was the previous name of the team that Feng started playing with in 1999?

Subquestions:
What team did Feng start playing with in 1999?
What was the previous name of Chongqing Longxin?

4. Question: How many years after the first Marvel Cinematic Universe film came out was Black Panther released?

Subquestions:
When was the first Marvel Cinematic Universe film released?
When was Black Panther released?
How many years after Iron Man came out was Black Panther released?

5. Question: Who was the first draft pick the year Damarius Bilbo went undrafted?

Subquestions:
What year did Damarius Bilbo go undrafted?
Who was the first draft pick in 2006?

Training data of 2WikiMQA for RGNN

1. Question: Which film came out earlier, Subliminal Seduction or Australia Marches With Britain?

Subquestions:
When did Subliminal Seduction come out?
When did Australia Marches With Britain come out?

2. Question: Who is the father-in-law of Deuteria?

Subquestions:
Who is the husband of Deuteria?
Who is the father of Eusebio?

3. Question: Who is the mother of the composer of film 404 (Film)?

Subquestions:
Who is the composer of film 404 (Film)?
Who is the mother of Ilayaraja?

4. Question: Which country the composer of film Sergeant Hassan is from?

Subquestions:
Who is the composer of film Sergeant Hassan?
Which country is Tamer Karaoğlu from?

5. Question: Do both films Across The Badlands and A Gutter Magdalene have the directors that share the same nationality?

Subquestions:
What is the director of Across The Badlands?
Where is Fred F. Sears from?
What is the director of A Gutter Magdalene?
Where is George Melford from?