Optimizing Question Semantic Space for Dynamic Retrieval-Augmented Multi-hop Question Answering

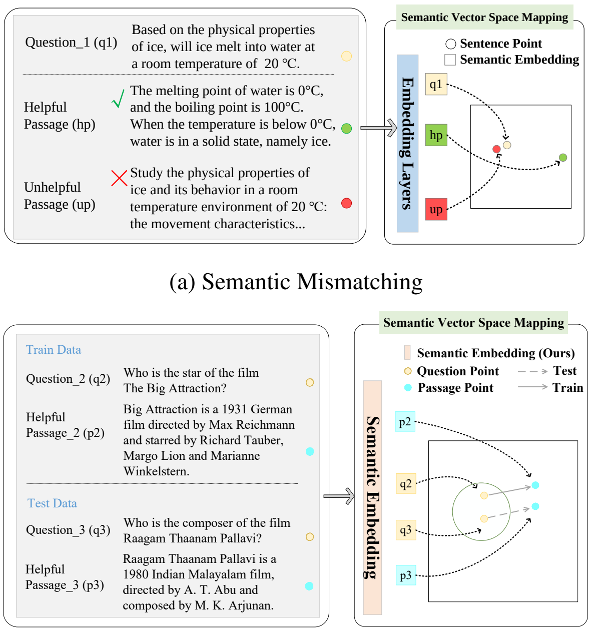
Linhao Ye, Lang Yu, Zhikai Lei, Qin Chen*, Jie Zhou*, and Liang He
School of Computer Science and Technology, East China Normal University
{lhye,lyu,kausal}@stu.ecnu.edu.cn {qchen, jzhou, lhe}@cs.ecnu.edu.cn

Abstract

Retrieval-augmented generation (RAG) is usually integrated into large language models (LLMs) to mitigate hallucinations and knowledge obsolescence. Whereas, conventional one-step retrieve-and-read methods are insufficient for multi-hop question answering, facing challenges of retrieval semantic mismatching and the high cost in handling interdependent subquestions. In this paper, we propose Optimizing Question Semantic Space for Dynamic Retrieval-Augmented Multi-hop Question Answering (`Q-DREAM`). `Q-DREAM` consists of three key modules: (1) the Question Decomposition Module (*QDM*), which decomposes multi-hop questions into fine-grained subquestions; (2) the Subquestion Dependency Optimizer Module (*SDOM*), which models the interdependent relations of subquestions for better understanding; and (3) the Dynamic Passage Retrieval Module (*DPRM*), which aligns subquestions with relevant passages by optimizing the semantic embeddings. Experimental results across various benchmarks demonstrate that `Q-DREAM` significantly outperforms existing RAG methods, achieving state-of-the-art performance in both in-domain and out-of-domain settings. Notably, `Q-DREAM` also improves retrieval efficiency while maintaining high accuracy compared with recent baselines.

1 Introduction

Recently, the advent of Large Language Models (LLMs), such as GPT (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and Mistral (Jiang et al., 2023), has significantly expanded the boundaries of machine language understanding and generation, enhancing the performance of a wide range of NLP tasks (Bang et al., 2023; Ouyang et al., 2022). However, they also tend to exhibit an inclination to generate hallucinations (Bang et al., 2023; Guerreiro et al., 2023; Chen et al., 2024). In addition, LLMs inherently suffer from knowledge obsolescence, as they are trained on static datasets collected at a fixed point in time (Dhingra et al., 2022; Huang et al., 2020). Consequently, the responses do not incorporate real-time updates or newly emerging information, which can be critical for many real-world applications (Zhang et al., 2024; Nguyen et al., 2024; Chen et al., 2025).



Question_1 (q1) Based on the physical properties of ice, will ice melt into water at a room temperature of 20 ℃.

Helpful Passage (hp) ✓ The melting point of water is 0°C, and the boiling point is 100°C. When the temperature is below 0°C, water is in a solid state, namely ice.

Unhelpful Passage (up) ✗ Study the physical properties of ice and its behavior in a room temperature environment of 20 ℃: the movement characteristics...

(a) Semantic Mismatching

Train Data

Question_2 (q2) Who is the star of the film The Big Attraction?

Helpful Passage_2 (p2) Big Attraction is a 1931 German film directed by Max Reichmann and starred by Richard Tauber, Margo Lion and Marianne Winkelstern.

Test Data

Question_3 (q3) Who is the composer of the film Raagam Thaanam Pallavi?

Helpful Passage_3 (p3) Raagam Thaanam Pallavi is a 1980 Indian Malayalam film, directed by A. T. Abu and composed by M. K. Arjunan.

(b) Semantic Space Optimization

Figure 1: (a) Semantic Mismatching: A semantic proximity gap leads to retrieval failures between question and helpful passage. (b) Semantic Space Optimization: Semantic embedding optimization aligns question and helpful passage by learning the latent semantic matching pattern.

One solution is to periodically retrain these models on updated corpora, while this approach is both computationally expensive and time-consuming (Jiang et al., 2024; Xia et al., 2024). A more efficient alternative is retrieval-augmented generation

*Corresponding Author.

(RAG), which integrates LLMs with information retrieved from external updated knowledge bases (Ram et al., 2023; Ye et al., 2024; Guu et al., 2020; Shi et al., 2023). Whereas, the RAG methods struggle with multi-hop QA tasks, as they usually fail to handle combinatorial questions requiring information from multiple passages. Recent prompting-based approaches (Press et al., 2023; Khot et al., 2022a; Kim et al., 2024) like IRCoT (Trivedi et al., 2023) attempt to address the multi-hop QA tasks by interleaving retrieval with chain-of-thought (CoT) reasoning. While IRCoT leverages retrieved results to refine reasoning, it heavily relies on the CoT capability of the model and demands multiple interactions between retriever and generator, incurring high computational costs.

Another challenge is that the existing RAG methods suffer from the semantic mismatching problem, which introduces the semantic similar but unhelpful passages for generation and negatively impacts the performance of multi-hop QA. In other words, even semantically similar passages may lack relevance to the question, leading models to prioritize the high similar but unhelpful content. As shown in Figure 1a, for the question "will ice melt into water at a root temperature at 20°C?", the method incorrectly retrieves a passage describing the movement characteristics of ice (high similar but unhelpful), while ignoring the low similar passage containing the relevant fact "The melting point of water is 0°C". This mismatching problem stems from the semantic proximity gap between the question and the truly helpful passages.

To address the above problems, we propose Optimizing **Q**uestion Semantic Space for **D**ynamic **Re**trieval-**A**ugmented **M**ulti-hop Question Answering (Q-DREAM), which enhances retrieval efficiency and resolves semantic mismatching problem with a three-module pipeline. The modules consist of the Question Decomposition Module (*QDM*), Subquestion Dependency Optimizer Module (*SDOM*), and Dynamic Passage Retrieval Module (*DPRM*). The *QDM* module first decomposes complex questions into fine-grained subquestions. Each subquestion is then processed separately. The independent subquestions are directly passed to *DPRM* for retrieval, while dependent subquestions are refined by *SDOM* before retrieval. The *DPRM* module integrates semantic alignment mechanism, which clusters subquestions and maps each cluster to a dedicated retrieval space for dynamic retrieval. As illustrated in Figure 1b, our method bridges the semantic gap between the question and the helpful passage. During training, the semantic embeddings of question q2 and helpful passage p2 are optimized to be closer in the dedicated retrieval space, which captures the latent matching pattern by associating "[Role] of [Film]" with the helpful content ("Film is..., Role by..."). At test time, the similar question q3 is mapped to the same cluster as q2. As both questions share the same semantic pattern "[Role] of [Film], q3 can align with the helpful passage p3 (Raagam Thaanam Pallavi is..., composed by...") by leveraging the learned matching patterns during training.

We perform extensive experiments on various datasets, and the results demonstrate that Q-DREAM significantly outperforms existing approaches in handling multi-hop questions, achieving superior performance across in-domain and out-of-domain settings. The main contributions can be summarized as follows:

- We propose a novel retrieval-augmented framework, namely Q-DREAM, which is model-agnostic and can be easily adapted to various LLMs to enhance the retrieval efficiency and effectiveness for retrieval-augmented multi-hop QA.

- Three modules as *QDM*, *SDOM* and *DPRM* are integrated into the framework, which work collaboratively to address the reconstruction of interdependent subquestions and resolve the semantic mismatching issues.

- We conduct elaborate analyses of the experimental results on three benchmark datasets, demonstrating the effectiveness of Q-DREAM under both in-domain and out-of-domain settings, and exhibiting scalability across various LLM backbones.

2 Related Work

2.1 Task Decomposition

Task decomposition is a crucial approach for addressing complex tasks, particularly in multi-turn and multi-hop question answering. Prior studies have explored various methods to break down complex questions into a series of simpler subquestions. Several works (Iyyer et al., 2017; Talmor and Berant, 2018; Rao and Daumé III, 2019; Wolfson et al., 2020; Khot et al., 2022b) propose models that decompose complex questions, yet these approaches do not leverage pre-trained language models (LMs). More recent methods, such as (Wang et al., 2022), utilize pre-trained models to generate contextual information for multiple-choice tasks. In addition, SEQZERO (Yang et al., 2022) introduces a few-shot semantic parsing technique that decomposes questions into structured subquestions aligned with a formal representation, enabling efficient reasoning through concise prompts. RA-ISF (Liu et al., 2024) further refines decomposition strategies by iteratively answering subquestions to minimize the impact of irrelevant text. However, despite these advancements, existing question decomposition techniques often fail to adequately handle interdependent subquestions in retrieval-augmented settings. Since the retrieval of each subquestion is performed independently, it may lead to incomplete or suboptimal retrieval results, reducing the accuracy of multi-hop question answering.

2.2 Retrieval-Augmented Language Models

Retrieval-augmented language models (LMs) enhance the reasoning and factual accuracy of LMs by incorporating externally retrieved information, thereby mitigating hallucination and factual inconsistency issues in open-domain question answering (ODQA) (Guu et al., 2020; Lewis et al., 2020; Lazaridou et al., 2022). In previous studies, such as REALM (Guu et al., 2020) jointly optimizes the retriever and language model to enhance retrieval-aware generation, RETRO (Borgeaud et al., 2022) introduces training language models on top of a frozen retriever, Atlas (Izacard et al., 2022) advances further by exploring dedicated loss functions for end-to-end training of both the retriever and the LM, demonstrating superior performance in few-shot learning tasks, RePlug (Shi et al., 2023) maintains a frozen black-box LM during the fine-tuning of retrieval modules.

However, these retrieval-augmented approaches struggle with multi-hop reasoning in QA. Recently, researchers have explored prompt-based methods to improve multi-hop QA. SelfAsk (Press et al., 2023) enhances retrieval by integrating structured prompting and search engines. DecomP (Khot et al., 2022a) decomposes tasks into modular sub-prompts tailored to specific reasoning steps. SURE (Kim et al., 2024) employs prompts to guide the LLMs in generating summaries for each answer candidate. All of these approaches do not utilize the Chain-of-Thought (CoT) reasoning of LLMs. ReAct (Yao et al., 2023) combines reasoning and action, prompting LLMs to generate task-related reasoning traces and actions in an interactive manner. IRCoT (Trivedi et al., 2023) integrates retrieval with CoT reasoning, using the reasoning to guide the retrieval and then leveraging the retrieval results to refine the reasoning process. However, such interactive approaches rely heavily on model performance and introduce a high computational cost.

3 Method

3.1 Overview

The overall architecture of our approach is shown in Figure 2, which consists of three modules: Question Decomposition Module (*QDM*), Subquestion Dependency Optimizer Module (*SDOM*) and Dynamic Passage Retrieval Module (*DPRM*). *QDM* first decomposes complex questions into fine-grained subquestions. These subquestions then undergo individual processing, where independent ones proceed directly to *DPRM* for retrieval while dependent ones are refined by *SDOM* before retrieval. *DPRM* incorporates a semantic alignment mechanism that clusters subquestions and maps each cluster to a dedicated retrieval space for dynamic retrieval.

Specifically, an origin question as Q_{ori} is input into the Q-DREAM framework to obtain the answer. The overall process is as follows: Firstly, we use the *QDM* to decompose Q_{ori} into subquestions $Q_{sub} = \{q_1, \ldots, q_n\}$. If a subquestion q_i does not depend on the answer to a previous subquestion, it is directly sent to the *DPRM* for passage retrieval. Otherwise, q_i is first optimized by the *SDOM* to generate a new subquestion $q_i\prime$, which is then sent to the *DPRM*.

In the *DPRM*, each subquestion ${q_i}^*$ (where ${q_i}^*$ is either q_i or $q_i\prime$) is first assigned to a semantic cluster based on its embedding. A corresponding LoRA block is then indexed and used for retrieval according to the cluster labels. Next, ${q_i}^*$ and the candidate retrieved passages $P = \{p_1, \ldots, p_m\}$ are encoded with the corresponding LoRA block. The embeddings $E_{{q_i}^*}$ and $E_P = \{E_{p_1}, \ldots, E_{p_m}\}$ are extracted from the last hidden state of the last layer of the model. We calculate the similarity between $E_{{q_i}^*}$ and each E_{P_v}, and the passage with the highest similarity score is selected as the retrieved result for ${q_i}^*$. Finally, the original question, the subquestions, and their corresponding retrieved passages are integrated into the answer generation

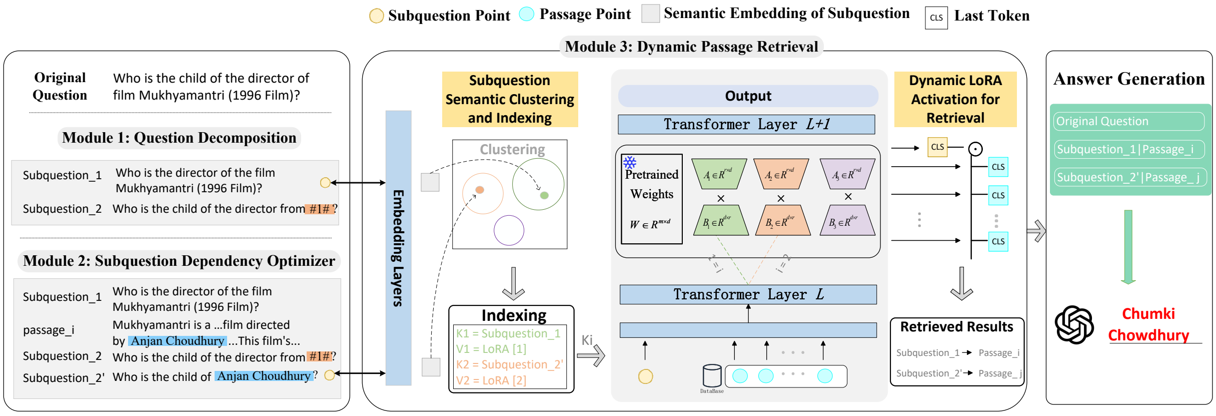

Subquestion Point
Passage Point
Semantic Embedding of Subquestion
CLS
Last Token
Original Question
Who is the child of the director of film Mukhyamantri (1996 Film)?
Module 1: Question Decomposition
Subquestion_1
Who is the director of the film Mukhyamantri (1996 Film)?
Subquestion_2
Who is the child of the director from #1#?
Module 2: Subquestion Dependency Optimizer
Subquestion_1
Who is the director of the film Mukhyamantri (1996 Film)?
passage_i
Mukhyamantri is a ...film directed by Anjan Choudhury ...This film's...
Subquestion_2
Who is the child of the director from #1#?
Subquestion_2'
Who is the child of Anjan Choudhury?
Module 3: Dynamic Passage Retrieval
Embedding Layers
Subquestion Semantic Clustering and Indexing
Clustering
Indexing
K1 = Subquestion_1
V1 = LoRA [1]
K2 = Subquestion_2'
V2 = LoRA [2]
Ki
Output
Transformer Layer L+1
Pretrained Weights
$W \in R^{m \times d}$
$A_1 \in R^{r \times d}$
$A_2 \in R^{r \times d}$
$A_3 \in R^{r \times d}$
×
$B_1 \in R^{d \times r}$
$B_2 \in R^{d \times r}$
$B_3 \in R^{d \times r}$
i = 1
i = 2
Transformer Layer L
DataBase
Dynamic LoRA Activation for Retrieval
CLS
Retrieved Results
Subquestion_1 → Passage_i
Subquestion_2' → Passage_ j
Answer Generation
Original Question
Subquestion_1|Passage_i
Subquestion_2'|Passage_ j
Chumki Chowdhury

Figure 2: Overall framework of Q-DREAM.

model to generate the final answer.

3.2 Q-DREAM based Training

In this section, we will introduce the training process for the three modules of Q-DREAM.

Question Decomposition Module (*QDM*) decomposes Q_{ori} into subquestions via a model M_{QDM}. For training the model, we optimize:

$$\max_{M_{QDM}} \log P(Q_{sub} \mid Q_{ori}; M_{QDM}) \tag{1}$$

where $Q_{sub} = \{q_1, \dots, q_n\}$, and q_i is a subquestion.

Subquestion Dependency Optimization Module *(SDOM)* aims at refining the subquestions that have dependencies on others. For a subquestion q_i that requires the answer to q_j, *SDOM* optimizes q_i by utilizing the retrieved passage p_{q_j} of q_j as:

$$\max_{M_{SDOM}} \log P(q_{i'} \mid q_j, p_{q_j}, q_i; M_{SDOM}) \tag{2}$$

Dynamic Passage Retrieval Module (*DPRM*) integrates semantic alignment mechanism to address the issue of semantic mismatching. We cluster subquestions into multiple categories and train cluster-specific embeddings to align questions and helpful passages in the semantic space. Given subquestions processed through *QDM* and *SDQM*, we partition them into k clusters $\{C_1, \dots, C_k\}$ using the k-means algorithm on their embeddings. For each cluster C_i, we fine-tune a LoRA block $M_{DPRM}^{C_i}$ to maximize the similarity between the subquestion $q \in C_i$ and the helpful passage p:

$$\max_{M_{DPRM}^{C_i}} \mathbb{E}_{(q,p)\in C_i} \log \left(\frac{\mathbf{E}_q \cdot \mathbf{E}_p}{\|\mathbf{E}_q\| \|\mathbf{E}_p\|} \right), \tag{3}$$

where E_q and E_p are the embeddings from the final hidden state of the last layer of $M_{DPRM}^{C_i}$.

3.3 Q-DREAM based Inference

In this section, we offer a comprehensive overview of how Q-DREAM framework processes and generates an answer for the original question Q_{ori}. Algorithm 1 presents the details of Q-DREAM. During the inference, we use all pre-trained models of the three modules.

Question Decomposition. In this process, we employ the M_{QDM} to decompose the original question Q_{ori} into multiple subquestions Q_{sub}, which is formulated as follows:

$$\arg\max_{Q_{sub}} P(Q_{sub} \mid Q_{ori}; M_{QDM}) \tag{4}$$

Next, we process the subquestions in Q_{sub} sequentially. If the subquestion q_i does not depend on the answer to previous subquestion, then directly input to the *DPRM* and retrieve the passage p_{q_i}. Otherwise, q_i is sent to the next step.

We determine whether a subquestion exhibits dependencies by checking if there is a "#number#" marker by itself. As shown in Figure 2, the decomposed Subquestion_2 "Who is the child of the director from #1#? " exhibits dependencies, with the marker "#1#" indicating that it depends on the answer to Subquestion_1.

Subquestion Dependency Optimizer. If subquestion q_i depends on the answer to subquestion #j#, it is incomplete and cannot effectively retrieve relevant content. Therefore, we need to utilize subquestion q_j and its retrieved passage p_{q_j} that contains useful information for answering q_j, to optimize q_i, which is formulated as:

$$\arg\max_{q_{i'}} P(q_{i'} \mid q_j, p_{q_j}, q_i; M_{SDOM}) \tag{5}$$

Algorithm 1 The inference process of Q-DREAM

```
Require: $M, M_{QDM}, M_{SDOM}, M_{DPRM}$,
         $Q_{ori}, Q_{sub}\prime \leftarrow \emptyset, P_{Q_{sub}\prime} \leftarrow \emptyset$
Ensure: $A_{ori}$
$Q_{sub} = \{q_1, \dots, q_n\} \leftarrow M_{QDM}(Q_{ori})$
for $i = 1$ to $n$ do
    if $q_i$ depend on the answer to $q_j$ then
        $q_i\prime = M_{SDOM}(q_j, p_{q_j}, q_i)$
        $q_i{}^* = q_i\prime$
    else
        $q_i{}^* = q_i$
    end if
    $Q_{sub}\prime \leftarrow Q_{sub}\prime \cup \{q_i{}^*\}$
    $c_{q_i{}^*} = cluster(q_i{}^*)$, Max_Score = -1
    $E_{q_i{}^*} = h_{Last}^{(M)} \leftarrow M_{DPRM}^{c_{q_i}{}^*}(q_i{}^*)$
    for $p_v$ to $P$ do
        $E_{p_v} = h_{Last}^{(M)} \leftarrow M_{DPRM}^{c_{q_i}{}^*}(p_v)$
        score $= cos(E_{q_i{}^*}, E_{p_v})$
        if score $>$ Max_Score then
            Max_Score = score
            $p_{q_i{}^*} = p_v$
        end if
    end for
    $P_{Q_{sub}\prime} \leftarrow P_{Q_{sub}\prime} \cup \{p_{q_i{}^*}\}$
end for
$A_{ori} = M(Q_{ori}, Q_{sub}\prime, P_{Q_{sub}\prime})$
```

where $q_i\prime$ is the reconstructed subquestion after optimization.

After optimization, reconstructed subquestion_2 "Who is the child of Anjan Choudhury?" in Figure 2 becomes more complete by explicitly identifying 'Anjan Choudhury' as the director during the retrieval process. Moreover, this explicit reconstruction process acts as the Chain-of-Thought, which helps the answer generation model to improve the reasoning performance for multi-hop QA.

Dynamic Passage Retrieval. In the Dynamic Passage Retrieval Module, each subquestion $q_i{}^*$ (where $q_i{}^*$ is either q_i or $q_i\prime$) is clustered based on its semantic embedding and the corresponding LoRA block is indexed. Subsequently, $q_i{}^*$ and candidate passages $P = \{p_1, \dots, p_m\}$ are input into $M_{DPRM}^{c_{q_i}{}^*}$ respectively. We obtain $E_{q_i{}^*}$ and $E_P = \{E_{p_1}, \dots, E_{p_m}\}$ form the last hidden state of the last layer of $M_{DPRM}^{c_{q_i}{}^*}$ and select the passage with the highest score according to the following formula:

$$p_{q_i{}^*} = \arg\max_{p_v \in P} f(E_{q_i{}^*}, E_{p_v}) \qquad (6)$$

where $f(\cdot)$ is a score function such as cosine similarity, and $p_{q_i{}^*}$ is the retrieved passage of $q_i{}^*$.

Finally, we replace the subquestions in Q_{sub} that exhibit dependencies with those reconstructed from *SDOM*, forming $Q_{sub}\prime$, and obtain the retrieved passages $P_{Q_{sub}\prime}$ with *DPRM*. Then the original question Q_{ori}, the subquestions $Q_{sub}\prime$, and the retrieved passages $P_{Q_{sub}\prime}$ are input into the answer generation model M to generate the answer, which is formulated as follows:

$$\arg\max_{A_{ori}} P(A_{ori} \mid Q_{ori}, Q_{sub}\prime, P_{Q_{sub}\prime}; M) \qquad (7)$$

where A_{ori} is the answer to the original question.

4 Experimental Setup

4.1 Datasets

We use the following three multi-hop QA datasets in the open-domain setting to evaluate Q-DREAM: **HotpotQA** (Yang et al., 2018), **2WikiMultihopQA** (2WikiMQA) (Ho et al., 2020), **IIRC** (Ferguson et al., 2020). For the above three datasets, we use the subsampled splits released by (Trivedi et al., 2023) as our test set. To evaluate the generalization of Q-DREAM, we only utilize 2WikiMQA (in-domain) for training, and HotpotQA and IIRC as the out-of-domain benchmarks for testing.

4.2 Baselines and Evaluation Metrics

We compare with the recent advanced baselines:

InstructRAG (Wei et al., 2025): allows LMs to denoise retrieved contents by generating rationales for better verifiability and trustworthiness.

ChatQA2 (Xu et al., 2024): bridge the gap between open-source LLMs and leading proprietary models in long context understanding and retrieval-augmented generation capabilities.

ChatGPT (Achiam et al., 2023): excels at multi-hop questions by leveraging its ability to understand and analyze questions. In the experiment setting, we evaluate ChatGPT in a one-shot prompting setting to guide its reasoning process.

SURE (Kim et al., 2024): enhances question-answering tasks by summarizing retrieved passages and selecting the most plausible answer from multiple candidates.

IRCoT (Trivedi et al., 2023): interleaves retrieval with CoT reasoning, dynamically refining

Table 1: EM / F1 scores for different methods on three datasets. **Bold** number indicates the best performance among all methods. * indicates the results from the original paper.

Methods/Datasets	In-domain	Out-of-domain		Average
	2WikiMQA	HotpotQA	IIRC	
Llama				
InstructRAG (Llama3-8B)	30.4/38.9	22.6/32.1	14.2/18.1	22.4/29.7
ChatQA2 (Llama3-8B)	29.0/35.2	32.6/42.5	21.4/25.7	27.7/34.5
Q-DREAM (Llama2-7B)	**32.4/39.7**	**36.0/46.1**	**23.8/27.2**	**30.7/37.7**
ChatGPT				
ChatGPT	23.6/29.2	24.2/32.5	11.4/13.8	19.7/25.2
SURE	32.8*/38.1*	33.2*/43.4*	20.6/25.5	28.9/35.7
IRCoT	41.9/55.2	25.5/38.1	21.0/31.0	29.5/41.4
Q-DREAM	**48.6/62.1**	**48.4/60.9**	**28.2/31.9**	**41.7/51.6**

the retrieval process based on intermediate reasoning steps.

We evaluate the QA performance using the standard exact match (EM) and F1 scores, which have been widely used in previous studies (Trivedi et al., 2023; Kim et al., 2024).

4.3 Implementation Details

Our framework consists of one answer generation model and three modules that serve as intermediate components. For the answer generation model, we experiment with open-source Llama2-7B (Touvron et al., 2023) and closed-source ChatGPT (GPT-3.5-turbo) (Achiam et al., 2023) through the API. As for the three modules, we fine-tune Mistral-7B (Jiang et al., 2023) for *QDM* and *SDOM*, and E5-Mistral-7B-Instruct (E5-Mistral) (Wang et al., 2023) for *DPRM*, which is pre-trained on a multilingual mixed data set.

We initially select 15,000 samples from the 2WikiMQA and generate training labels for *QDM* and *SDOM* using ChatGPT. After data cleaning and denoising, we ultimately obtain 13,363 samples as our training set. In the *DPRM*, we use the k-means (Krishna and Murty, 1999) algorithm for clustering. During training, we adopt Adam (Kingma and Ba, 2014) with a constant learning rate of 5e-5 and a dropout rate of 10%. we set the batch size to 16 and train the model with one A100. We use greedy decoding during the inference process across all experiments to ensure deterministic generation.

5 Results and Analyses

5.1 Main Results

We report the EM and F1 scores on three multi-hop QA datasets under in-domain and out-of-domain settings, and the results are shown in Table 1.

(1) In-Domain Performance with ChatGPT. Q-DREAM achieves state-of-the-art performance on 2WikiMQA with ChatGPT as the backbone model. Compared to ChatGPT without retrieval augmentation, our method exhibits absolute improvements of 25.0 and 32.9 regarding the EM and F1 scores respectively. Moreover, Q-DREAM outperforms the strongest retrieval-augmented baseline IRCoT with substantial improvements. These findings verify the effectiveness of our method in multi-hop question answering.

(2) Generalization in Out-of-Domain Settings. To evaluate the generalization of our method, we directly apply our method trained on 2WikiMQA to HotpotQA and IIRC. As shown in the "Out-of-domain" column of Table 1, Q-DREAM surpasses all baselines in out-of-domain datasets, achieving great improvements of 17.5 in terms of F1 compared with the second best model SURE in HotpotQA. On the whole, our method is superior to all the baselines in the average performance, demonstrating good generalization and robust cross-domain adaptability of our method.

(3) Performance with Smaller-Scale Model. To show the effectiveness of our method with smaller-scale models, we also experiment Q-DREAM with Llama2-7B as the backbone. The results show that Q-DREAM achieves an average performance of 30.7 and 37.7 in EM and F1 scores respectively across three datasets, surpassing baselines built on larger architectures like InstructRAG and ChatQA2. Moreover, Q-DREAM with Llama2-7B even outperforms SURE that uses larger-scale ChatGPT as the backbone, indicating the superiority of our RAG method especially for smaller LLMs.

5.2 Ablation Studies

We set up three variants on 2WikiMQA to investigate the effectiveness of each component of Q-DREAM:

- - *DPRM*: This variant refers to the subquestions generated by the *QDM* and *SDOM* being directly input to the E5-Mistral model for passage retrieval, without utilizing the *DPRM*.
- - *(SDOM, DPRM)*: After *QDM*, the question is decomposed into subquestions which are subsequently directly input to the E5-Mistral model for passage retrieval.
- - *All*: The question is directly input to the E5-Mistral model for passage retrieval. For a fair comparison, the number of passages retrieved in this variant is consistent with our method.

The results are shown in Table 2. We can see that removing *DPRM* substantially degrades performance, attributable to the absence of semantic matching optimization with dynamic dedicated retrieval. Concurrently removing *SDOM* and *DPRM* yields the most severe deterioration, as directly using the decomposed questions without optimization for conventional retrieval is prone to the semantic mismatching problem. Removing all components achieves higher performance than - *(SDOM, DPRM)* alone, underscoring the critical role of *SDOM*. Without *SDOM* optimization, the dependent subquestions are vague and lack the necessary contextual information for effective retrieval. Overall, each module within the framework plays an indispensable role in multi-hop QA.

Table 2: The results of ablation studies.

Methods	EM	F1
Q-DREAM	**48.6**	**62.1**
- *DPRM*	44.4	56.5
- *(SDOM, DPRM)*	32.4	38.1
- *All*	37.2	44.7

5.3 Further Analyses

Efficiency of Inference. The baseline methods require multiple model interactions during the reasoning process, resulting in substantial time consumption. In contrast, our approach utilizes pre-trained sub-modules to generate fine-grained subquestions and retrieve relevant passages, which are then directly processed by the final question generation model. Thus, our method is more efficient.

To quantify the efficiency of inference, we evaluate the inference time in the 2WikiMQA test set. We compare Q-DREAM with the strong baseline IRCoT on the same computing platform with uniform hardware configurations and operating systems. To ensure the reliability of the results, each method run three times independently under the same conditions. As shown in Table 3, Q-DREAM processes each sample in 4 seconds on average, which is 6× faster than IRCoT. Additionally, our method demonstrates a smaller variance in inference time across three experiments, highlighting its stronger stability compared to IRCoT. These characteristics are crucial for practical applications.

Performance of Retrieval. To further assess the effectiveness of our retrieval method, we compare with the classical and recent advanced information retrieval (IR) models. Three categories of baselines are used for comparison: BM25 (Jones et al., 2000) is a lexical-based retrieval model; Contriever (Izacard et al., 2021) is a dense neural retriever; and the retrieval methods integrated in InstructRAG and IRCoT. For the space of limit, we present the results of 2WikiMQA in Table 4.

It is observed that our retrieval method with *DPRM* obtains superior retrieval performance compared with the pure IR models and the IR methods used in recent RAG baselines. By removing *DPRM* and using the initial E5-Mistral model for retrieval, the retrieval performance significantly drops since all questions and passages share the same semantic encoder, which is prone to the semantic mismatching problem. By integrating our retrieval method as *DPRM*, the similar questions are assigned with a dedicated LoRA block for retrieval, which forces the embeddings of the helpful passages to be closer

Table 3: Efficiency of inference on 2WikiMQA. Avg(s)/Dataset represents the average time required to process the dataset. Avg(s)/Sample represents the average time required to process a single sample.

Methods	Inference Time (s)			Avg(s)/Dataset	Avg(s)/Sample
	First	Second	Third		
IRCoT	8887	16138	12485	12503	25
Q-DREAM	2009	2000	2025	2011	4

Table 4: Comparison of the retrieval performance. - *DPRM* refers to an ablated variant of Q-DREAM without its dynamic passage retrieval module.

Metric / Model	BM25	Contriever	InstructRAG	IRCoT	Q-DREAM (- *DPRM*)	Q-DREAM
Precision (%)	41.0	20.5	19.7	53.8	57.4	**81.8**
Recall (%)	68.0	33.0	40.7	68.9	62.0	**85.7**
F1 (%)	51.2	25.3	26.5	60.4	59.6	**83.7**

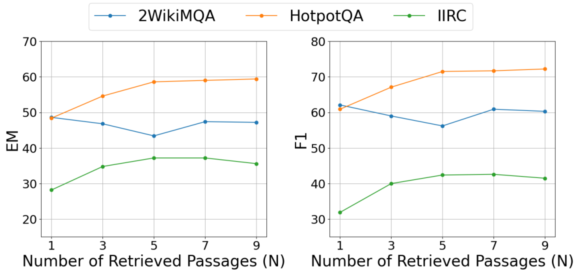


Figure 3: Impact of the number of retrieved passages.

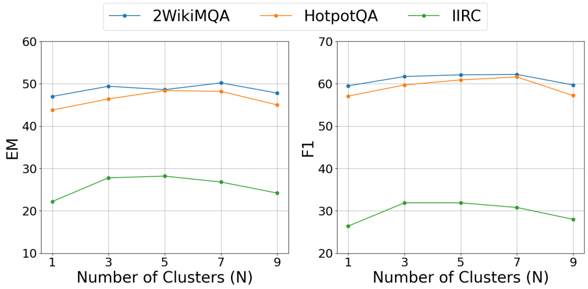


Figure 4: Impact of the number of clusters.

to the question in the question semantic space.

Impact of Retrieved Passages. We analyze how the number of retrieved passages (N) for each sub-question affects the QA performance. As illustrated in Figure 3, the sensitivity of N varies across various datasets. The results show that retrieving just a single passage in the in-domain dataset 2WikiMQA achieves optimal performance, which suggests that our retrieval method can effectively prioritizes relevant information above the irrelevant ones. Regarding to the out-of-domain datasets such as HotpotQA and IIRC, the performance improves with the increasing number of retrieved passages at first, and then tends to drop as more irrelevant information will be introduced by excessive retrieval. This indicates that retrieving additional appropriate information is usually beneficial to provide more clues for enhancing the QA performance in the out-of-domain scenarios.

Impact of Clusters. The impact of the number of clusters is shown in Figure 4. We observe that the performance first increases with the growing number of clusters, and then declines. Specifically, too few clusters result in coarse-grained retrieval, which reduces the model's ability to distinguish between different question spaces and relevant passages, and is prone to the semantic mismatching problem. Whereas, excessive clustering is also detrimental, as it will cause the semantically related questions to be unnecessarily separated into different spaces, which obstructs the learning of the common matching patterns and thereby diminishes the retrieval effectiveness. Overall, these findings indicate that an optimal cluster size balances the retrieval granularity and generalization. Too few clusters fail to capture fine-grained semantic distinctions, while too many clusters lead to semantic fragmentations, ultimately reducing the overall question answering performance.

6 Conclusions

In this paper, we propose a method to optimize the question semantic space for dynamic retrieval-augmented multi-hop question answering. By integrating three modules as *QDM*, *SDOM* and *DPRM*, our method well bridges the semantic gaps between the question and helpful passages. Extensive experiments verify the effectiveness of each module of our method. In particular, our method outperforms the state-of-the-art baselines with significant improvements in both in-domain and out-of-domain settings, indicating the good generalization in unknown scenarios. Moreover, our method improves the retrieval accuracy, while maintaining high efficiency compared with the recent advanced retrieval methods. In the future, we will explore to extend our method to multilingual or multimodal settings, and investigate the effectiveness on more out-of-domain datasets.

Limitations

Though Q-DREAM enhances the multi-hop question answering, its performance remains to be studied with other complex reasoning tasks. In addition, how to retrieve the long-tail knowledge for RAG remains to be studied.

Ethics Statements

Language models may generate incorrect or biased information, especially when handling sensitive topics. While retrieval-augmented methods can help mitigate this issue, they do not fully eliminate the risk of biased or inappropriate content. Therefore, caution is necessary when deploying such systems in user-facing applications.

This work utilizes publicly available datasets (HotpotQA, 2WikiMQA, IIRC) that comply with academic licenses and do not contain personal or sensitive information. All models (e.g., ChatGPT, Llama2-7B) and training data were used in accordance with their respective terms of service. Our study does not involve human subjects or private data collection, thus avoiding risks related to consent or privacy.

Acknowledgements

This research is funded by the National Nature Science Foundation of China (No. 62477010 and No.62307028), the Natural Science Foundation of Shanghai (No. 23ZR1441800), Shanghai Science and Technology Innovation Action Plan (No. 24YF2710100 and No.23YF1426100) and Shanghai Special Project to Promote High-quality Industrial Development (No. 2024-GZL-RGZN-02008).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Kedi Chen, Qin Chen, Jie Zhou, Xinqi Tao, Bowen Ding, Jingwen Xie, Mingchen Xie, Peilong Li, and Zheng Feng. 2025. Enhancing uncertainty modeling with semantic graph for hallucination detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23586–23594.

Kedi Chen, Qin Chen, Jie Zhou, He Yishen, and Liang He. 2024. Diahalu: A dialogue-level hallucination evaluation benchmark for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9057–9079.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. Iirc: A dataset of incomplete information reading comprehension questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1137–1147.

Nuno M Guerreiro, Duarte M Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. 2023. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831.

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*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ziheng Jiang, Haibin Lin, Yinmin Zhong, Qi Huang, Yangrui Chen, Zhi Zhang, Yanghua Peng, Xiang Li, Cong Xie, Shibiao Nong, et al. 2024. {MegaScale}: Scaling large language model training to more than 10,000 {GPUs}. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 745–760.

K Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management*, 36(6):809–840.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022a. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022b. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. *arXiv preprint arXiv:2404.13081*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

K Krishna and M Narasimha Murty. 1999. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

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.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. Ra-isf: Learning to answer and understand from retrieval augmentation via iterative self-feedback. *arXiv preprint arXiv:2403.06840*.

Xuan-Phi Nguyen, Shrey Pandit, Senthil Purushwalkam, Austin Xu, Hailin Chen, Yifei Ming, Zixuan Ke, Silvio Savarese, Caiming Xong, and Shafiq Joty. 2024. Sfr-rag: Towards contextually faithful llms. *arXiv preprint arXiv:2409.09916*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 143–155.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv e-prints*, pages arXiv–2307.

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.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Shepherd pre-trained language models to develop a train of thought: An iterative prompting approach. *arXiv preprint arXiv:2203.08383*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. InstructRAG: Instructing retrieval-augmented generation via self-synthesized rationales. In *The Thirteenth International Conference on Learning Representations*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Callie Hao, and Nishil Talati. 2024. Understanding the performance and estimating the cost of llm fine-tuning. In *2024 IEEE International Symposium on Workload Characterization (IISWC)*, pages 210–223. IEEE.

Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. *arXiv preprint arXiv:2407.14482*.

Jingfeng Yang, Haoming Jiang, Qingyu Yin, Danqing Zhang, Bing Yin, and Diyi Yang. 2022. Seqzero: Few-shot compositional semantic parsing with sequential prompts and zero-shot models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 49–60.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Linhao Ye, Zhikai Lei, Jianghao Yin, Qin Chen, Jie Zhou, and Liang He. 2024. Boosting conversational question answering with fine-grained retrieval-augmentation and self-check. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2301–2305.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. 2024. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*.