MRM: Improving Multihop Reasoning Capability for LLMs With Multistage QA Retrieval

Wei Li, *Member, IEEE*, Yanqing Zhang, Guifang Sun, Chao Ma, *Member, IEEE*, Jipeng Qiang, *Senior Member, IEEE*, and Jiaxing Shen

***Abstract*—Large Language Models (LLMs) continue to face significant challenges in improving their reasoning capabilities, particularly for knowledge-intensive multi-hop questions. Existing approaches attempt to address this hurdle through generating reasoning procedures or retrieving additional knowledge, while encountering limitations. Conventional prompting methods often produce undesired reasoning traces, while retrieving pertinent knowledge from external data sources remains problematic. In this paper, we introduce a novel approach to enhance LLM reasoning capabilities: the Multistage Retrieval Method (MRM). It comprises two key components: plan design and implementation with retrieval, utilizing four specialized stages: Plan Stage, Extraction Stage, Reasoning Stage, and Output Stage. The MRM process begins by constructing a prompt with demonstration examples and an instruction, guiding to design a logically coherent multi-step plan in a standardized format in Plan Stage. To acquire the necessary knowledge for reasoning, we employ an innovative question-answer strategy by generating a question at each step, followed by document retrieval using vector-based methods. The retrieved documents and the generated questions are then fed into the Extraction Stage to extract relevant knowledge, which are utilized to execute the next step with Reasoning Stage. It helps mitigate the risk of acquiring irrelevant information by performing retrieval prior. We repeat this process until the final step of the plan, whereupon we input all extracted knowledge and the original question into the Output Stage to generate the final answer. Experimental results demonstrate the effectiveness of MRM over SOTA methods, achieving the highest accuracy scores across both common-sense and symbolic reasoning tasks in four public datasets.**

***Index Terms*—LLM, reasoning capability, plan, multistage retrieval.**

Received 24 August 2025; accepted 14 November 2025. This work was supported in part by Lingnan University (LU) under Grant SDS24A17 and in part by by the National Natural Science Foundation of China Grant 52571073. *(Corresponding authors: Jipeng Qiang; Jiaxing Shen.)*

Wei Li and Yanqing Zhang are with the School of Artificial Intelligence and Computer Science & Engineering Research Center of Intelligent Technology for Healthcare, Ministry of Education, Jiangnan University, Wuxi 214122, China (e-mail: cs_weili@jiangnan.edu.cn; 6223110058@stu.jiangnan.edu.cn).

Guifang Sun is with the School of Mechanical Engineering, Southeast University, Nanjing 210094, China (e-mail: gfsun@seu.edu.cn).

Chao Ma is with the The School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: chaoma@whu.edu.cn).

Jipeng Qiang is with the Department of Computer Science, Yangzhou University, Yangzhou 225009, China (e-mail: jpqiang@yzu.edu.cn).

Jiaxing Shen is with the School of Data Science, Lingnan University, Hong Kong SAR, China (e-mail: jiaxingshen@ln.edu.hk).

The source code is available at https://github.com/foreverlove944/mrm.

Recommended for acceptance by H. Cai.

Digital Object Identifier 10.1109/TETCI.2025.3642285

I. Introduction

LARGE Language Models (LLMs) have revolutionized Natural Language Processing (NLP) by leveraging the transformer architecture to learn statistical relationships from vast amounts of text data [1]. LLMs have demonstrated exceptional capabilities in text generation, particularly in applications such as code generation [2], where they have substantially enhanced productivity. Beyond their generative prowess, these advanced models have also shown promising reasoning capabilities, enabling their effective deployment in complex tasks including sentiment analysis [3] and text classification [4], [5], with performance that surpasses expectations. Despite their impressive language processing abilities, LLMs face a persistent challenge in reasoning capabilities [6], often causing suboptimal outputs, especially for multi-hop questions. The definition of multi-hop question is that it requires inference combination of multiple related sub questions or information fragments to obtain the final answer, and it involves the association of multiple objects, relationships, or contexts, requiring LLM to have cross segment logical reasoning capabilities [7].

A. Problem Statement and Motivation

The reasoning capability heavily relies on the learned knowledge when LLM facing multi-hop questions. However, the knowledge could be fixed after training. Fine-tuning increases the training cost and bringing disaster forgetting issue, so the intuitive method is to supplement the extra knowledge. Yet, the supplemented knowledge should be dynamic because different reasoning stages have different objects or relationships. Moreover, existing methods have no reasoning trace, which reduces the quality of produced answers. Therefore, this paper focuses on the problem of supplementing extra knowledge with high quality for answering multi-hop questions, with clearly reasoning trace demonstration.

Motivated by greedy algorithm [8], which decomposes a complicated problem into multiple sub-problems. A greedy algorithm makes a locally optimal choice at each stage with the hope of finding a global optimum. Analogously, we explicitly decomposes the multi-hop question into multiple sub-questions and each sub-question solely holds one subject and retrieve the knowledge required for the current subject and its contexts to reason the temporary answer. After that, we reason the answer for the next object, based on the previously reasoned temporary

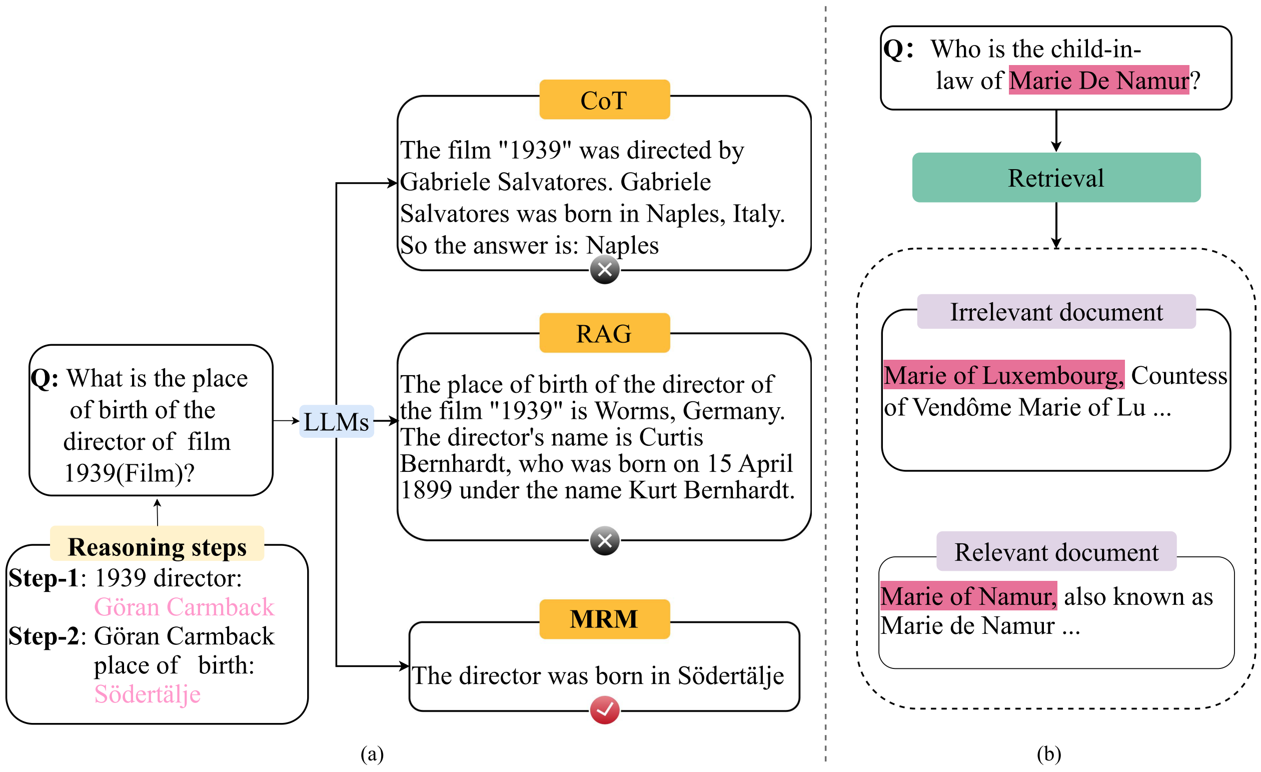
CoT
The film "1939" was directed by Gabriele Salvatores. Gabriele Salvatores was born in Naples, Italy. So the answer is: Naples
RAG
The place of birth of the director of the film "1939" is Worms, Germany. The director's name is Curtis Bernhardt, who was born on 15 April 1899 under the name Kurt Bernhardt.
MRM
The director was born in Södertälje
Q: What is the place of birth of the director of film 1939(Film)?
LLMs
Reasoning steps
Step-1: 1939 director: Göran Carmback
Step-2: Göran Carmback place of birth: Södertälje
(a)
Q: Who is the child-in-law of Marie De Namur?
Retrieval
Irrelevant document
Marie of Luxembourg, Countess of Vendôme Marie of Lu ...
Relevant document
Marie of Namur, also known as Marie de Namur ...
(b)

Fig. 1. (a) Final answers from CoT, RAG, and our **MRM** for a three-hop question (**Q**). (b) Illustration of irrelevant and relevant documents.

answers, until all subjects are processed. With this, we obtain the final answer.

B. Limitations of Prior Art

Existing research has primarily focused on two approaches to enhance LLM reasoning: prompting techniques and knowledge augmentation. Chain-of-Thought (CoT) [9] and its variants represent prominent prompting methods. CoT employs a single invocation to generate both the final answer and reasoning traces, ignoring the implicit linguistic logic. However, this approach often leads to undesired outcomes (Fig. 1(a)) as the reasoning process is not iterative or truly reasoned [9]. Variants such as Auto-CoT [10] and zero-shot CoT [11] face similar limitations. Knowledge augmentation methods, exemplified by Retrieval-Augmented Generation (RAG) [12] and its variants, attempt to supplement LLMs with external knowledge. RAG retrieves relevant documents based on the input question and incorporates them into the LLM's reasoning process. However, this approach often retrieves irrelevant information due to its single retrieval step before reasoning and there is no logic to guide retrieval, leading to suboptimal results, especially for multi-hop questions requiring diverse knowledge at different reasoning stages (Fig. 1(a)). Other RAG variants, such as rewrite-retrieve-read [13] and Internet-Augmented [14], encounter similar challenges. While some methods like Zero-shot Plan-and-Solve Prompting [15] attempt to improve reasoning through plan design, they essentially function as CoT variants and inherit their limitations. Considering the significance of challenges of the aforementioned methods and the potential benefits of overcoming them, it is worth developing a new approach to improve the reasoning capability.

C. Our Approach

To address these challenges, we propose the Multistage Retrieval Method (**MRM**), a new approach to enhance LLM reasoning capabilities. **MRM** comprises two main components: 1) plan design, and 2) implementation with retrieval, utilizing four specialized stages: **Plan Stage**, **Extraction Stage**, **Reasoning Stage**, and **Output Stage**. Specifically, it constructs a prompt with demonstration examples and an instruction to guide the **Plan Stage** in designing a logically coherent multi-step plan. In general, a multi-hop question includes several general subjects and a specific subject. **MRM** focuses on a subject from the input question in each step in the plan, implicitly indicating the desired knowledge for that specific subject and then rendering each general subject to be a specific one in the rest steps. The instruction emphasizes attention to nouns (e.g., people, places) as key subjects for knowledge acquisition. To retrieve relevant knowledge for reasoning, we introduce a novel question-answer strategy. This approach generates a unique question at each step and retrieves documents using vector-based methods. To mitigate the risk of retrieving irrelevant information (Fig. 1(b)), we employ the **Extraction Stage** to extract knowledge from retrieved documents based on the generated question and subject conditions. The extracted knowledge is then used by the **Reasoning Stage** to execute the next step of the plan through a new prompting technique. This process is iterated until the final step, where all extracted knowledge and the original question are input to the **Output Stage** to generate the final answer.

D. *Novelty and Contributions*

Different from existing works, which they either employ a single invocation without caring the linguistic logic or retrieve irrelevant information thereby generating undesired final answers, we innovatively design a plan with four stages to show the reasoning trajectory, and propose a new retrieval strategy and a question-answer paradigm to obtain knowledge required by reasoning to produce the final answer.

Our contributions can be summarized as follows:

- We introduce **MRM**, a new method to enhance LLM reasoning capabilities via multi-step planning and targeted retrieval, extracting the knowledge of subject by subject in linguistic logical manner in the reasoning process.
- We propose an innovative retrieval strategy based on a question-answer paradigm, generating step-specific questions to retrieve relevant documents and extract pertinent knowledge.
- Extensive experiments demonstrate the efficacy of **MRM** in both common-sense and symbolic reasoning scenarios, with significant improvements in accuracy across multiple datasets compared to state-of-the-art methods.

II. Related Work

In this section, we first introduce the LLM, and then discuss the two categories of improving reasoning capability studies, according to their adopted strategies.

Large Language Models: Large Language Models (LLMs) employ the transformer architecture and they are trained on vast amounts of text data to achieve large-scale text generation. The training process of LLM generally includes four stages: pre-training, supervised fine-tuning (SFT), reward, and reinforcement learning. Pre-training focuses on learning the knowledge and in-context relationships among sentences with the cross-entropy loss. After obtaining the pre-trained model, SFT uses prompts (e.g., questions or instructions) to serve as the input and then fine-tunes the model with the same loss, rendering model to generate outputs with instructions [16]. However, the generated answers from SFT might not be suitable for the human preferences. With this, both reward and reinforcement learning are employed to evaluate the generated answers. It maximizes the distance between the evaluated scores to train the reward model. Then, it utilizes the reinforcement learning and the trained reward model to tune LLM to improve the generation quality. However, the persisting challenge in LLM is the low reasoning capability, given that the output is reasoned by LLM. The traditional strategy is to further increase the number of parameters of LLM (e.g., 7 billion parameters to 130 billion parameters), because more number of parameters implicitly indicate more reasoning capability [1], [17]. Yet, it is difficult to infinitely increase the parameters in machine with limited physical resources. Hence, researchers develop different methods to attempt to improve the reasoning capability discussed as below.

Prompting: Since vast amounts of text training data hold many subjects and contents, prompting LLM with demonstration examples and instructions is in the ascendant [18]. Chain-of-Thought (CoT) [9] randomly selects questions from dataset to manually craft demonstration examples in prompting to try to improve the reasoning capability. Obviously, randomly selecting questions might not be related to the subjects appeared in posed question, and manually crafted examples have subjectivity and their quality vary. Moreover, although CoT can generate the reasoning procedure with those examples, both the final answer and generated steps are simultaneously output, because it adopts one invocation to directly generate final answer. In other words, there is unclear reason behind the generated steps, so the final answer is usually undesired. Zero-shot CoT [11] designs a new fixed format instruction "Let's think step by step", while abandons the demonstration examples in the prompting. It reduces the influence of crafted examples on reasoning capability in LLM. Yet, the absence of demonstration example causes that the generated chains might include reasoning mistakes. Besides, Zero-shot CoT has the same issue as the vanilla CoT, that is the unclear reason behind the generated steps, generating undesired final answer. Auto-CoT [10] groups the training data into several clusters with clustering method, and selects a representative question from each cluster to construct inference chains with Zero-shot CoT to improve the quality of demonstration examples. However, it still holds the same issue as the vanilla CoT.

Knowledge Augmentation: The another direction of improving reasoning capability in LLM is to supplement more knowledge obtained from the extra data sources, and it usually relies on the retrieval method. The representative methods are Retrieval-Augmented Generation (RAG) [12] and its variants [13], [19]. RAG follows *retrieve-and-read* paradigm. Specially, it utilizes the neural network to construct an embedding model to vectorize all documents and the posed question, and then employs the **Euclidean** metric to measure such a distance between vectorized documents and the question. RAG retrieves documents with the measured distance values. The smaller the measured value is, the more related to posed question the retrieved documents are. Last, the retrieved documents are fed into LLM to generate the final answer. However, RAG and its variants perform the retrieval only once before reasoning. In other words, they solely retrieve documents at the first step, and abandon retrieval in the rest of reasoning. Since each reasoning step needs different knowledge, they might retrieve many irrelevant documents to LLM to reason answer [20], thus generating the undesired answer. It becomes more challenge when facing the multi-hop question. To retrieve more desired documents, some researchers decomposes a complicated question into multiple simple sub-questions. Rewrite-retrieve-read [13] rewrites the original posed question to multiple sub-questions by using a well-trained small LLM, which helps retrieve more documents from extra data source; Least-to-Most [21] decomposes the posed question to multiple sub-questions with prompting, and retrieves documents from extra data source, according to each sub-question. It is simpler than Rewrite-retrieve-read method. Take-A-Step-Back [22] abstracts the posed question to form a semantic concept, which helps reduce the complexity of the

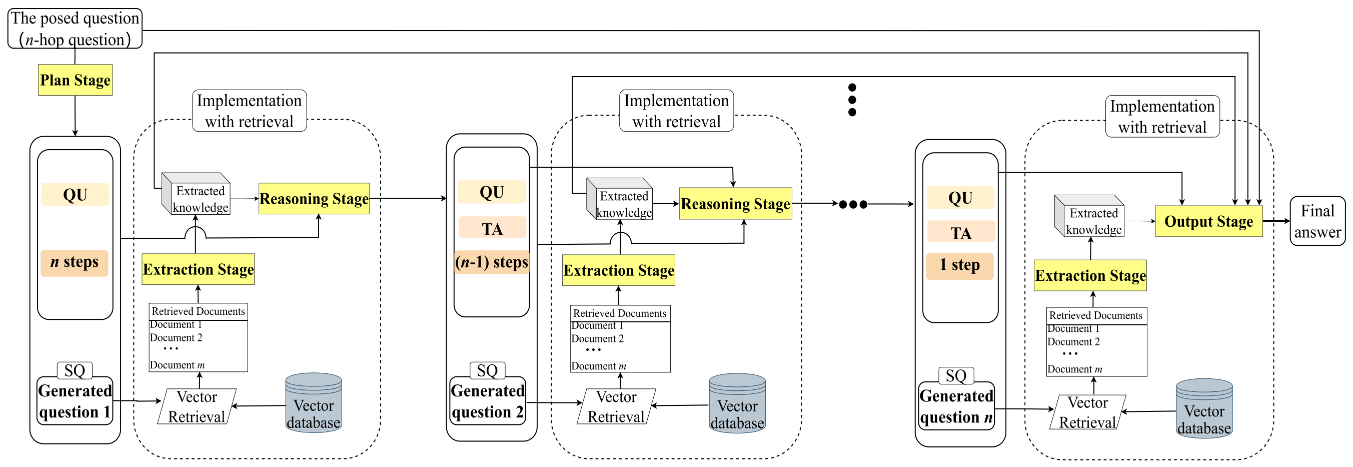

Fig. 2. The reasoning procedure on a three-hop question example. QU indicates the "*Question understanding*"; SQ denotes the "*Subject question*"; TA means the "*Temporary answer*".

original question, thus enabling to retrieve more documents. Yet, those methods solely perform one retrieval before reasoning, and do not retrieve any documents in the rest of reasoning as the vanilla RAG, thus generating the final answer with low quality.

Designing Plan with Prompting: Given the challenges appeared in CoT and RAG and their variants [9], [12], researchers attempt to design a plan to improve the reasoning capability, and the representative method is the Zero-shot Plan-and-Solve Prompting (Zero-shot PS) [15], a zero-shot CoT variant in practice. It sets one instruction to design plan and adopts the similar strategy (i.e., one invocation) to prompt LLM to perform plan. Therefore, it also holds the same issues as the vanilla CoT. Although the plan demonstrates the reasoning trace, it does not utilize this trace to improve the common-sense reasoning capability, so its performance is, in turn, lower than vanilla CoT [15]. Inspired by the plan method, this paper proposes a new retrieval strategy to improve the reasoning capability along with the plan implementation.

III. MRM

In this section, we propose Multistage Retrieval Method (**MRM**) to improve the reasoning capability as shown in Fig. 2. **MRM** consists of two components: 1) plan design; and 2) implementation with retrieval, with **Plan Stage**, **Extraction Stage**, **Reasoning Stage**, and **Output Stage**. The **Plan Stage** focuses on designing a logically coherent plan; the **Extraction Stage** extracts the desired knowledge from the retrieved documents; the **Stage** concentrates on reasoning the temporary answers, and the **Output Stage** generates the final answer.

The plan design component only includes the **Plan Stage**, which is to make a logically coherent plan to reason the final answer for the posed question, in which each step explicitly indicates the subjects included in posed question and the knowledge about subjects. The plan implementation with retrieval component has other three stages, which implements the designed plan and outputs the final answer. It performs one step after extracting the desired knowledge from the retrieved documents with a new question-answer manner. Specifically, each step within plan generates a question, and we retrieve documents about subjects, according to the generated question. Considering the subjects of documents might not be related to generated question, we view the subjects as the conditions to extract knowledge from the retrieved documents with **Extraction Stage**, and then utilize the **Reasoning Stage** to obtain a temporary answer, according to the extracted knowledge. After that, we input the temporary answer into the next step. We repeat this process until the final answer is obtained, with **Output Stage**. The plan demonstrates the reasoning procedure, while our retrieval method can avoid the influence of irrelevant documents on reasoning the final answer, because we retrieve documents at each step and we perform knowledge extraction before running this step.

A. *Plan Design*

The goal of designing plan is that each step within plan loyally reflects each reasoning step, and different reasoning steps focuses on different subjects when facing complicated questions, especially for the multi-hop questions. Different from the traditional plan method (e.g., Zero-shot PS [15] solely utilizes one instruction to design plan), we design plan with a prompting, which includes both demonstration examples and an instruction. Considering the demonstration examples describe the details of plan, they help demonstrate the reasoning procedure. Note that the demonstration examples include 2-hop, 3-hop, and 4-hop examples, which helps LLM set the number of steps (i.e., n *steps* in Fig. 2). Our plan follows the criteria shown as below:

- Avoiding to misunderstand the potential meaning of posed multi-hop question, because it holds multiple subjects and each subject might confuse the reasoning;
- Clearly indicating a subject involved in the posed question and the desired knowledge about this subject at each step, which helps demonstrate the reasoning procedure;
- Maintaining logical coherence between different steps within plan, which helps reason the final answer with high quality.

For the first criteria, we design demonstration examples in the constructed prompting. It includes the *Question understanding* to indicate the potential meaning of the given question, which is achieved by **Plan Stage**. The potential meaning is stored in **QU** shown in Fig. 2. Note that **Plan Stage** also designs the multi-step plan during understanding the posed question in our **MRM**. Hence, the output from **Plan Stage** is a triplet, which consists of the **Question understanding** with "*The question is about...*" format, the multi-step plan with "*I need to know [the knowledge of subject] by [subject]*" format, and the **Subject question**. Take "*What genre is the record label of the performer of So Long, See You Tomorrow associated with?*" as the example. Here, the subject "*So Long, See You Tomorrow*" is specific, while other subjects (e.g., "*genre*", "*the record label*", and "*the performer*") are general. Our **MRM** helps LLM avoid the misunderstanding by rendering each general subject to be a specific one.

For the second criteria, we design a new fixed format (i.e., "*I need to know [knowledge of subject] by [subject]*") for each step within plan to demonstrate the desired knowledge. Here, "*[subject]*" is involved in the posed question, and different steps focus on different subjects; the "*[knowledge of subject]*" denotes that we should obtain the knowledge in this step for reasoning. To figure out the subjects, we set the instruction (i.e., "*Pay attention to the nouns (e.g., people, and places) that appear in the question as they represent the subjects of the knowledge you desire to obtain*"), and then add it into the prompting to guide **Plan Stage** to focus on different subjects, with the demonstration examples. The demonstration examples include the posed question and multiple steps, in which each step focuses on one subject and the desired knowledge about this subject in **MRM**. Once we obtain the desired knowledge at a certain step, we can utilize those knowledge to reason a temporary answer at this step. Since different steps have different knowledge about different subjects, we reason different temporary answers at different steps to gradually approach to the final answer, with the plan implementation. Such a procedure demonstrates each reasoning step, thus forming the whole reasoning procedure when obtaining the final answer.

For the third criteria, we utilize a temporary answer from each **Reasoning Stage** to take in the output from the last step and feed this temporary answer into next step (See Fig. 2), which guarantees the logic coherence for the plan, with such a format: "*I need to know the [knowledge of the subject involved in last step's knowledge] by [subject involved in last step's knowledge]*". With this in place, we finish the plan designing.

Take a three-hop question as the example, which is "*What genre is the record label of the performer of So Long, See You Tomorrow associated with?*". To follow the first criteria, the **Plan Stage** figures out the potential meaning of this question, which is "*The question is about genre*" denoted by QU. Moreover, **MRM** observes that this question contains three subjects, which are "record label", "the performer", and the "So Long, See You Tomorrow". With this, we design a plan with three steps (i.e., $n = 3$). Each step demonstrates the desired knowledge about one subject for following the second criteria. For instance, the first step demonstrates "*I need to know the performer of So Long, See You Tomorrow by So Long, See You Tomorrow*", and the extracted knowledge (i.e., "*So Long, See You Tomorrow*") can answer this by reasoning, which is "*The performer is Bombay Bicycle Club*" and we view it as the temporary answer denoted by TA, because it is not the final answer but it is related to the posed question; the second step demonstrates "*I need to know the record label of Bombay Bicycle Club by Bombay Bicycle Club*" after feeding TA to this step, and the extracted "*Bombay Bicycle Club*" knowledge can answer this question by reasoning, which is "*The record label is Island Records*" and we still view it as another TA. With the plan implementation, we gradually approach to the final answer for following third criteria. Such an approaching implicitly demonstrates the whole reasoning procedure. Note that there are two steps when facing the two-hop questions, and we set four steps when facing four-hop questions. Others are unchanged.

Note that **MRM** is difficult to un-decompose the multi-hop question. The plan guarantees logical coherence through a special structured format, in which each step of this plan is designed: "I need to know [the knowledge of subject] by [subject]", and the subsequent steps build on the previous ones to maintain the logical coherence: "I need to know the [knowledge of the subject involved in last step's knowledge] by [subject involved in last step's knowledge]. Such a logical coherence guarantees that the model decomposes one subject by one subject, avoiding decomposition failing. Take the sentence of *"What genre is the record label of the performer of So Long, See You Tomorrow associated with?"* as the example. Since **MRM** is guided to identify the specific subjects within question as the starting point for reasoning, the first step of plan is: "I need to know the performer of So Long, See You Tomorrow by So Long, See You Tomorrow", given that "So Long, See You Tomorrow" is the specific subject. The second step is: "I need to know the record label of the performer by the performer", which naturally follows the first step and maintains logical consistency to avoid decomposition failing. Besides, since the quality of SQ heavily relies on the decomposition and it is impossible to fail to decompose the multi-hop question, the generated SQ is robustness.

B. Plan Implementation With New Retrieval Method

Demonstrating the reasoning trace heavily relies on effectively implementing the designed plan. The traditional method (e.g., Zero-shot PS [15]) adopts one invocation to perform plan as the vanilla CoT, so it might demonstrate the in-correct reasoning procedure. Different from it, our **MRM** performs the designed plan with retrieval method. We extract the desired knowledge from the retrieved documents, which helps reason the answers in each step in a new question-answer (QA) manner. In this way, out method consists of *retrieval* and the *planning implementation* parts. The *retrieval* part includes a language model, denoted as **Extraction Stage** shown in Fig. 2, and a vector database that stores all documents; the *planning implementation* part includes a language model, denoted as **Reasoning Stage**, a prompting that includes demonstration examples describing the implementation details and an instruction. The *retrieval* appears at each step within plan (See Fig. 2), and it is targeted to retrieve documents, which helps extract the desired knowledge to obtain answers. Next, we demonstrate the retrieval process in our QA manner.

Each stage (**Plan Stage**, **Extraction Stage**, and **Reasoning Stage**) outputs a generated question, which is about one subject, named **Subject question** (i.e., SQ). We input the generated question into the *retrieval* part to retrieve documents from the vector database, with vector retrieval method. Then, we view subject in this step as a condition, and feed both retrieved documents and the generated question into **Extraction Stage** to extract desired knowledge, according to this condition. The process of new retrieval is formalized as follows. Let a multi-hop question be Q, which contains a subject set, $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ where S_i denotes the reasoned subject at the i-th step (e.g., *So Long, See You Tomorrow*). With this, we have:

$$\mathcal{D}_i = \left\{ \frac{\mathbf{q}_i \cdot Doc}{\|\mathbf{q}_i\| \|Doc\|} \geq \tau \right\}, \quad \mathbf{q}_i = \Phi(S_i) \tag{1}$$

where q_i denotes the generated question appeared at the i-th step $(1 \leq i \leq n)$. Doc indicates all documents saved in the vector database. $\Phi(S_i)$ represents the constraints of a subject, S_i, which means that q_i is related to S_i. τ is a threshold, and we utilize it to filter out the un-relevant documents. We traverse all documents within Doc by calculating the similarity between each document and q_i. Only the similarity value that is larger than τ can be filtered out. $\mathcal{D}_i$ indicates the retrieved documents at i-th step, which are regarding to q_i and satisfy $\Phi(S_i)$.

After that, we utilize the extracted knowledge to perform next step of plan with **Reasoning Stage** via a new prompting. The new prompting also includes both demonstration examples and an instruction. Different from the demonstration examples appeared in **Plan Design**, such demonstration examples describe the implementation details for each step of plan. There are other retrieval methods for LLM (e.g., Rewrite-retrieve-read [13] or Take-A-Step-Back [22]), and they either directly retrieve documents with the posed question across all documents (e.g., Rewrite-retrieve-read [13]) or directly provide the retrieved documents to LLM after retrieval (e.g., Take-A-Step-Back [22]). However, those methods solely perform one retrieval before reasoning, and they might retrieve many irrelevant documents. Moreover, each reasoning needs different knowledge, there are no more knowledge to be provided LLM to reason answers, thus degenerating the quality of final answer. Our method utilizes the subject to filter out irrelevant documents, and perform the retrieval at each step. Therefore, we can provide more knowledge to LLM to reason the desired final answer.

We still take the same question, "*What genre is the record label of the performer of So Long, See You Tomorrow associated with?*", as the example to further demonstrate the technical details. Different from existing methods (e.g., Few-shot CoT and its variants), a logically coherent plan is designed to decompose this question with a fixed format: "I need to know [the knowledge of subject] by [subject]". For this example, we have such a demonstration example in the first step of plan: *"I need to know the performer of So Long, See You Tomorrow by So Long, See You Tomorrow"*. With this, the SQ in the first step is: {*"subject":"So Long, See You Tomorrow", "question":"Who performed So Long, See You Tomorrow?*}. The generated question is used to retrieve documents through semantic similarity and the subject serves as a constraint to filter out the irrelevant documents. After inputting the retrieved documents and the generated question into the **Extraction Stage**, we obtain the extracted knowledge (e.g., *"The performer is Bombay Bicycle Club"*), without requiring extra instructions or demonstration examples. We then feed the knowledge into the **Reasoning Stage**, and it views the extracted knowledge as the temporary answer. For the second step, the demonstration example with fixed format is: *"I need to know the record label of the performer by the performer"*. However, after inputting the temporary answer appeared in the first step into the second step, we have a new demonstration example: *"I need to know the record label of Bombay Bicycle Club"*. With this in place, a new SQ is generated, based on the new demonstration example. We repeat this process until reaching the last step of the plan, in which we feed the last step, the posed question, and all temporary answers into the **Output Stage** to produce the final answer. Our **MRM** is formalized as follows. Assuming Q depends on a set of extracted knowledge at each step, $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$ where k_i denotes the extracted knowledge (e.g., "the performer is Bombay Bicycle Club") by using retrieved documents with q_i (e.g., "Who performed So Long, See You Tomorrow?") at the i-th step $(1 \leq i \leq n)$. k_i is utilized to execute the i-th step with **Reasoning Stage**. Hence, **MRM** can be modeled as a conditional probability maximization problem:

$$P(A \mid Q, \mathcal{K}) = \prod_{t=1}^{T} P(w_t \mid w_{<t}, Q, \mathcal{K}), \mathcal{K} = \bigcup_{i=1}^{n} \text{Extract}(q_i, \mathcal{D}_i). \tag{2}$$

where $A = \{w_1, w_2, \ldots, w_T\}$ denotes the set of generated temporary answers, given that producing the final answer requires all generated temporary answers. w_i indicates a specific generated temporary answer at the i-th step, according to q_i.

C. Analysis of MRM

LLM usually relies on reasoning to obtain the final answer. However, different reasoning methods have different performance. Besides, the multi-hop question implicitly includes linguistic logic. Take a three-hop format of "**A** of **B** of **C** of **D**" as the example (e.g., "*What genre is the record label of the performer of So Long, See You Tomorrow associated with?*"). According to the rule of linguistic logic, we first need to reason the answer of who performing the song of "*So Long, See You Tomorrow*", and then reason the answer of which company "*the performer*" belongs to. Last, we reason the answer of what kind of "*genre*" the record is, based on the previous three reasoned answers. In other words, we need to reason the answer of **C** from **D**, and reason the answer of **B** from both **C** and **D**, and then reason the answer of **A** from **B**, **C**, and **D**. Only this we can obtain the correct final answer.

CoT and its variants [9], [10], [11] solely rely on the learned knowledge to reason the final answer, with the crafted demonstration examples. Since they adopt one invocation to generate the final answer and there is no any strategy to pay their attention to the linguistic logic, they might directly reason the answer of **A** from **D**, resulting in the incorrect final answer.

RAG and its variants [7], [12], [13], [19] usually decouple the posed question into sub-questions, and then retrieve the documents, based on the decoupled sub-questions. Similar to CoT and its variants, RAG and its variants ignore the linguistic logic. They target to retrieve documents about all sub-questions. However, they might retrieve the irrelevant documents regarding to subjects, because they perform retrieval before reasoning and there is no logic to guide retrieval. Besides, the decoupled method also affects the retrieving performance, and it might also retrieve the irrelevant documents to LLM, still resulting in the incorrect final answer.

Different from CoT and RAG and their variants, our **MRM** designs a logically coherent plan to render LLM to focus on the implicit linguistic logic behind the posed question. **MRM** utilizes the instruction to guide LLM to capture different subjects (i.e., **A**, **B**, **C**, and **D**), with the demonstration examples. Then, **MRM** generates the questions about the subjects. In the first step within plan, **MRM** generates question about **D** subject, and retrieves the documents to answer the generated question. After that, **MRM** extracts the desired knowledge, according to the retrieved documents and the generated question in the question-answer manner. The answer of **D** subject is included by the desired knowledge. Last, **MRM** utilizes the extracted knowledge to perform next step of plan (i.e., reasoning the answer of **C**, **B**, and **A** subjects one step by one step). Since **MRM** notices the linguistic logic of posed question and targeted retrieve documents regarding to the subjects, **MRM** outputs the correct final answer.

IV. Experiments

For the experiments, five public datasets are studied, which are Musique [23], HotpotQA [24], 2WikiMultihopQA [25], Coin Flip and Last Letter Concatenation [9]. The first three datasets are multi-hop datasets, and they are widely used to reflect the common-sense reasoning capability; while the last two datasets are utilized to reflect the symbolic reasoning capability. Moreover, we randomly sample 2000 questions from Musique, HotpotQA, and 2WikiMultihopQA datasets, separately. To retrieve documents, we view the "contexts" as the open-domain settings for the three datasets. For the last two datasets, we randomly sample 500 questions from each one, given that the two datasets have small data. Since symbolic reasoning is the closed-domain QA, it does not have retrieval.

Seven baseline methods are employed, which are **Direct prompting**, **TDB** [22], **Few-shot CoT** [9], **RAG** [26], **IR-CoT** [27], **EfficientRAG** [28] and **Iter-RetGen** [29]. For **Direct prompting**, it solely uses "Q: <question> \nA:" as our prompting to generate the final answer. Besides, we employ the answering accuracy as the metric to reflect the performance of each method. TDB and Few-shot cot are the representatives on improving the reasoning capability for LLM. **TDB** directly appends the posed question to the constructed prompting (e.g., "Take a deep breath and work on this problem step-by-step."), and then utilizes the prompting to reason the final answer. It is a variant of Zero-shot CoT [22]. **RAG** follows the *Retrieve-and-Read* paradigm. For **Few-shot CoT**, the prompting includes 10 demonstration examples and each demonstration example holds a pair of question-answer [9]. It vectorizes all documents and the posed question to retrieve documents and subsequently provides the retrieved documents to LLM to produce the final answer. IR-Cot [27] and Iter-RetGen [29] assume that the content generated by the model contains contextual information, and thus leverage this content to retrieve documents (Generation-Augmented Retrieval). However, the contextual information might not be useful. Moreover, they do not retrieve documents, according to knowledge required for reasoning. EfficientRAG [28] trains a small language model (e.g., DeBERTa-v3 with 330M parameters) to generate final answer, and only invoke the LLM at the final answer-generation stage, which results in lower accuracy. These methods are designed specifically for knowledge-intensive multi-hop questions and are not suitable for symbolic reasoning tasks. Besides, we set $\tau = 0.2$ in Eq. (1). To make a fair comparison, we select the first five retrieved documents for **MRM**, RAG, IRCoT, EfficientRAG, and Iter-RetGen, because those methods adopt retrieval strategy and the number of documents that are related to the posed question is less than five [23], [24], [25].

Since GPT-3.5-turbo [1] is the most widely utilized LLM backbone in practice, we employ it to perform all methods for a fair comparison. Moreover, both GLM-4 [30] and Qwen-turbo [31] are also widely used in practice, we employ them as the backbones for all methods.

A. Experimental Results on Common-Sense Reasoning

Musique: We first deploy all methods to the three backbone LLMs, and quantitatively validate their answering accuracies on Musique dataset, which are shown in Table I. Specifically, we count the answering accuracy scores in the randomly sampled 2000 questions for each method, in which there are 1700 two-hop questions, 216 three-hop questions, and 84 four-hop questions. From Table I, we can observe that our **MRM** achieves the highest accuracy score over all baseline methods in all cases, including 2-hop, 3-hop, and 4-hop cases. TDB and few-shot CoT have the similar answering results as the Direct prompting (i.e., benchmark), which indicates that both TDB and Few-shot CoT have the similar performance as the benchmark. In the three backbones, Qwen-turbo shows the worst performance, and the overall answering accuracy score is only 29.20% for the RAG. For the recent LLM methods, IRCoT achieves the highest answering accuracy scores among all baseline methods, while other two LLM methods (i.e., EfficientRAG and Iter-RetGen) achieve the similar answering performance. However, our **MRM** obtains the highest answering accuracy scores among all methods, achieving the accuracy improvement by 3.65% (9.1% and 14.55) in GPT-3.5-turbo (Qwen-turbo and GLM-4), compared with the recent IRCoT. The experiments demonstrate the effectiveness of **MRM** on reasoning the high quality final answers.

HotpotQA: We continue to validate all methods on HotpotQA dataset, and the quantitative results are shown in Table II.

TABLE I
THE ANSWERING ACCURACY SCORES FROM ALL METHODS ON MUSIQUE DATASET FOR MULTI-HOP QUESTIONS

Method		GPT-3.5-turbo	Qwen-turbo	GLM-4
Direct prompting	2-hop	23.76%	7.82%	20.24%
	3-hop	16.20%	4.63%	10.19%
	4-hop	14.29%	7.14%	9.52%
	overall	22.55%	7.45%	18.70%
TDB	2-hop	23.76%	6.82%	20.24%
	3-hop	14.35%	3.24%	11.57%
	4-hop	13.10%	9.52%	9.52%
	overall	22.30%	6.55%	18.85%
Few-shot CoT	2-hop	25.94%	9.47%	23.88%
	3-hop	14.81%	6.94%	13.43%
	4-hop	29.76%	9.52%	14.29%
	overall	24.90%	9.20%	22.35%
RAG	2-hop	45.00%	31.18%	41.06%
	3-hop	28.24%	17.13%	25.46%
	4-hop	10.71%	20.24%	16.67%
	overall	41.75%	29.20%	38.35%
IRCoT	2-hop	50.47%	45.29%	45.89%
	3-hop	28.70%	32.41%	33.27%
	4-hop	14.29%	11.90%	23.81%
	overall	46.60%	42.50%	43.60%
EfficientRAG	2-hop	47.70%	42.06%	43.59%
	3-hop	19.91%	15.74%	24.07%
	4-hop	14.29%	9.52%	17.86%
	overall	43.30%	37.85%	40.40%
Iter-RetGen iter4	2-hop	49.00%	43.53%	45.35%
	3-hop	21.76%	25.93%	25.46%
	4-hop	16.67%	11.90%	21.43%
	overall	44.70%	39.30%	42.20%
MRM	2-hop	**53.76%**	**54.53%**	**60.24%**
	3-hop	**31.48%**	**33.33%**	**48.61%**
	4-hop	**27.38%**	**39.29%**	**40.48%**
	overall	**50.25%**	**51.60%**	**58.15%**

TABLE II
THE ANSWERING ACCURACY SCORES FROM ALL METHODS ON HOTPOTQA DATASET FOR MULTI-HOP QUESTIONS

Method		GPT-3.5-turbo	Qwen-turbo	GLM-4
Direct prompting	2-hop	52.15%	27.26%	47.33%
	3-hop	38.77%	16.30%	27.90%
	overall	50.30%	25.75%	44.65%
TDB	2-hop	51.04%	23.83%	42.63%
	3-hop	31.88%	20.23%	30.43%
	overall	48.40%	23.35%	40.95%
Few-shot CoT	2-hop	54.41%	30.40%	51.91%
	3-hop	37.68%	27.90%	41.30%
	overall	52.10%	30.05%	50.45%
RAG	2-hop	64.85%	59.74%	70.30%
	3-hop	62.32%	52.17%	58.70%
	overall	64.50%	58.70%	68.70%
IRCoT	2-hop	72.74%	68.44%	76.62%
	3-hop	49.64%	28.10%	53.62%
	overall	69.55%	62.85%	73.45%
EfficientRAG	2-hop	70.24%	61.89%	72.16%
	3-hop	46.01%	34.42%	53.26%
	overall	66.90%	58.10%	69.55%
Iter-RetGen iter4	2-hop	71.46%	63.46%	74.13%
	3-hop	49.28%	42.03%	45.65%
	overall	68.40%	60.50%	70.02%
MRM	2-hop	**73.90%**	**65.95%**	**79.76%**
	3-hop	**70.65%**	**64.13%**	**75.72%**
	overall	**73.45%**	**65.70%**	**79.20%**

Obviously, our **MRM** still achieves the highest accuracy scores over all methods in those three backbones. For HotpotQA dataset, we also count the answering accuracy in the randomly sampled 2000 questions for each method, in which there are 1724 two-hop questions, 276 three-hop questions. From Table II, we can observe that TDB holds the worst performance, because its answering accuracy scores are the lowest among all methods. It is observed that TDB, few-shot CoT, and Direct prompting have the similar performance, as the **Musique** dataset. Moreover, we found that Qwen-turbo still shows the worst performance among the three backbones, and the overall answering accuracy score is 62.85% for the recent IRCoT. Although its score is higher than other baseline methods, while it is still lower than our **MRM** (i.e., 65.7%), achieving the accuracy improvement by 2.85% . For the other two backbones (i.e., GPT-3.5-turbo and GLM-4), **MRM** improves the overall accuracy scores by 3.9% and 5.75% when comparing with IRCoT, given that it also achieves the highest scores among all baseline methods. The experimental results further show the effectiveness of **MRM**.

TABLE III
THE ANSWERING ACCURACY SCORES FROM ALL METHODS ON 2WIKIMULTIHOPQA DATASET FOR MULTI-HOP QUESTIONS

Method		GPT-3.5-turbo	Qwen-turbo	GLM-4
Direct prompting	2-hop	14.92%	14.03%	31.06%
	3-hop	50.00%	16.67%	50.00%
	4-hop	38.33%	8.33%	28.33%
	overall	33.00%	12.85%	30.55%
TDB	2-hop	29.27%	12.24%	37.82%
	3-hop	41.67%	25.00%	25.00%
	4-hop	38.10%	12.14%	26.67%
	overall	31.20%	12.30%	35.25%
Few-shot CoT	2-hop	37.30%	15.60%	32.40%
	3-hop	41.66%	41.67%	50.00%
	4-hop	69.52%	22.86%	35.23%
	overall	44.10%	17.35%	33.10%
RAG	2-hop	44.13%	26.34%	44.90%
	3-hop	50.00%	25.00%	33.33%
	4-hop	28.81%	21.51%	26.43%
	overall	40.95%	25.30%	40.45%
IRCoT	2-hop	60.14%	54.34%	62.31%
	3-hop	50.00%	25.00%	41.67%
	4-hop	45.48%	31.67%	69.05%
	overall	57.00%	49.40%	63.60%
EfficientRAG	2-hop	53.51%	50.19%	58.04%
	3-hop	33.33%	33.33%	28.57%
	4-hop	47.86%	37.38%	63.81%
	overall	52.15%	47.35%	59.10%
Iter-RetGen iter4	2-hop	55.80%	52.10%	61.10%
	3-hop	50.00%	33.33%	41.67%
	4-hop	58.33%	51.90%	67.14%
	overall	56.30%	51.90%	62.25%
MRM	2-hop	**63.71%**	**60.59%**	**77.10%**
	3-hop	**66.67%**	**50.00%**	**66.67%**
	4-hop	**71.43%**	**57.14%**	**82.00%**
	overall	**65.35%**	**59.80%**	**78.05%**

2WikiMultihopQA: Last, we validate all methods on the 2WikiMultihopQA dataset. As the above-mentioned two datasets, we count the answering accuracy scores in the randomly sampled 2000 questions for each method. In those questions, there are 1568 two-hop questions, 12 three-hop questions, and 420 four-hop questions. Table III shows the quantitative results from all methods. Obviously, our **MRM** still achieves the highest accuracy scores among all methods. From Table III, we can observe that only TDB holds the similar performance as the benchmark, given that its overall answering accuracy scores are similar to Direct prompting in the three backbones; while Few-shot CoT shows the similar performance as the RAG. Moreover, we still found that Qwen-turbo shows the worst performance among the three backbones. Besides, the recent

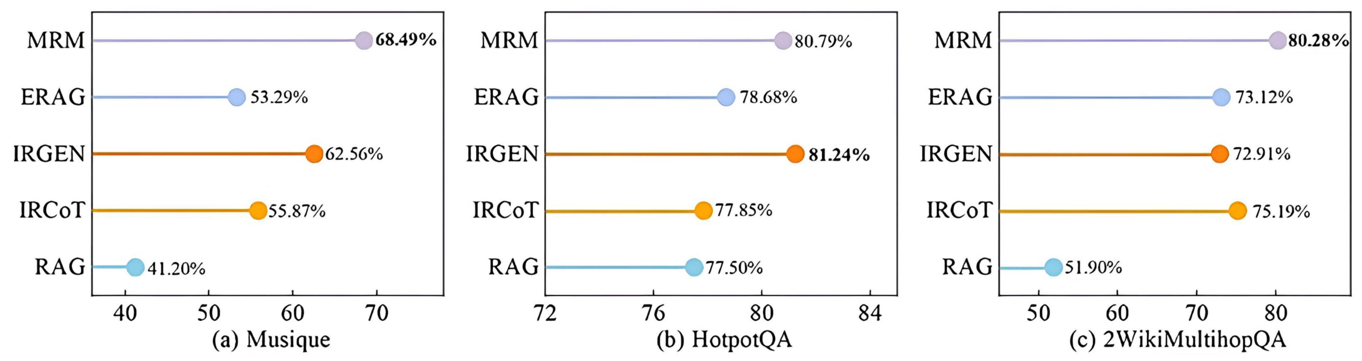

Fig. 3. ERAG indicates the EfficientRAG, and IRGEN means the Iter-RetGen. We abbreviate their names for saving space.

IRCoT obtains the highest score in both GPT-3.5.turbo (57.00%) and GLM-4 (63.60%) backbones, while Iter-RetGen achieves the highest score in Qwen-turbo with 51.9% in all baseline methods. However, **MRM** still holds larger score then them, achieving the improvements by 8.35% for GPT-3.5-turbo, 7.9% for Qwen-turbo, and 14.45% for GLM-4. The experimental results also demonstrate the effectiveness of our **MRM**.

B. Hit Rate

Since our **MRM** and the RAG, IRCoT, EfficientRAG, Iter-RetGen have retrieval strategy, and the retrieved documents have a significant influence on the reasoning capability, it is necessary to compare **MRM** with those methods on retrieving the desired documents. With this, we employ the hit rate metric to loyally reflect the retrieval performance, with hit rate $= \frac{\sum_{i=1}^{k} \frac{n_i}{m_i}}{k}$ where k represents the number of questions. m_i indicates the quantity of documents about the i-th question and those documents are stored in the vector database; n_i denotes the number of documents retrieved from m_i. For example, assuming the vector database has 10 documents regarding to i-th question, we have m_i=10. n_i=2 if we retrieve 2 relevant documents from the 10 documents. The more desired documents the method retrieves, the higher the hit rate score is. For those methods, we utilize the posed question as the query to perform the retrieval, and take GPT-3.5-turbo as the backbone to compare **MRM** with baseline methods on the three datasets, given that GPT-3.5-turbo is widely used in practice. We set the same experimental settings as the previous ones, and the hit rate scores of those methods are shown in Fig. 3. From Fig. 3, it is obviously to observe that our **MRM** achieves the higher hit rate scores than other methods on different datasets. It illustrates the retrieval performance of our **MRM** which is better than RAG, IRCoT, EfficientRAG, and Iter-RetGen, further demonstrating that the reasoning capability of **MRM** is better than RAG.

C. Experimental Results on Symbolic Reasoning

To highlight the significance and the generalization of our **MRM**, we also compare **MRM** with baseline methods on symbolic reasoning, and the studied datasets are: Coin Flip and Last Letter Concatenation [9]. Since symbolic reasoning does not hold the retrieval, we remove the retrieval from **MRM**, but solely input the generated questions to **Extraction Stage** in **MRM**. To make a fair comparison, we also remove the retrieval from RAG. Fig. 4 illustrates the demonstration examples on the two datasets, and Table IV reports the symbolic reasoning accuracy scores. Since the symbolic reasoning strategy is different from the common-sense reasoning, there are other studies that mainly focus on the symbolic reasoning, and the representative methods are: Zero-shot PS+ [15], Zero-short CoT [11], and Self-consistency [32]. Here, we also view those studies as baseline methods in symbolic reasoning scenario. From Table IV, we can observe that RAG without retrieval degenerates to the benchmark. Zero-shot PS+, Zero-shot CoT, Few-shot CoT, and Self-consistency show the similar performance on Coin Flip dataset, while Zero-shot PS+, Few-shot CoT, Self-consistency, and TDB exhibit the similar performance on Last Letter Concatenation dataset. However, our **MRM** still outperforms all methods on both Coin Flip and Last letters Concatenation datasets, which also demonstrate the effectiveness of **MRM** on symbolic reasoning.

TABLE IV
THE REASONING RESULTS ON COIN FLIP AND LAST LETTERS CONCATENATION

Method	Coin Flip	Last Letter Concatenation
Direct prompting	77.8%	53.8%
Zero-shot PS+	99.2%	75.0%
Zero-shot CoT	96.8%	65.0%
Few-shot CoT	99.2%	70.8%
Self-consistency	98.4%	73.4%
TDB	84.2%	74.8%
RAG without retrieval	77.8%	53.8%
MRM	**99.6%**	**78.8%**

D. Ablation Studies

Since our **MRM** utilizes **Extraction Stage**, retrieval, and the instruction components to reason the answers, we conduct a series of ablation experiments to show the necessity of each one by only removing the **Extraction Stage** or only removing the retrieval or only removing the instruction from **MRM** on the three datasets. We deploy them to GPT-3.5-turbo backbone, and sample 150 questions from each dataset. Table V shows the ablation experimental results. From Table V, we can observe

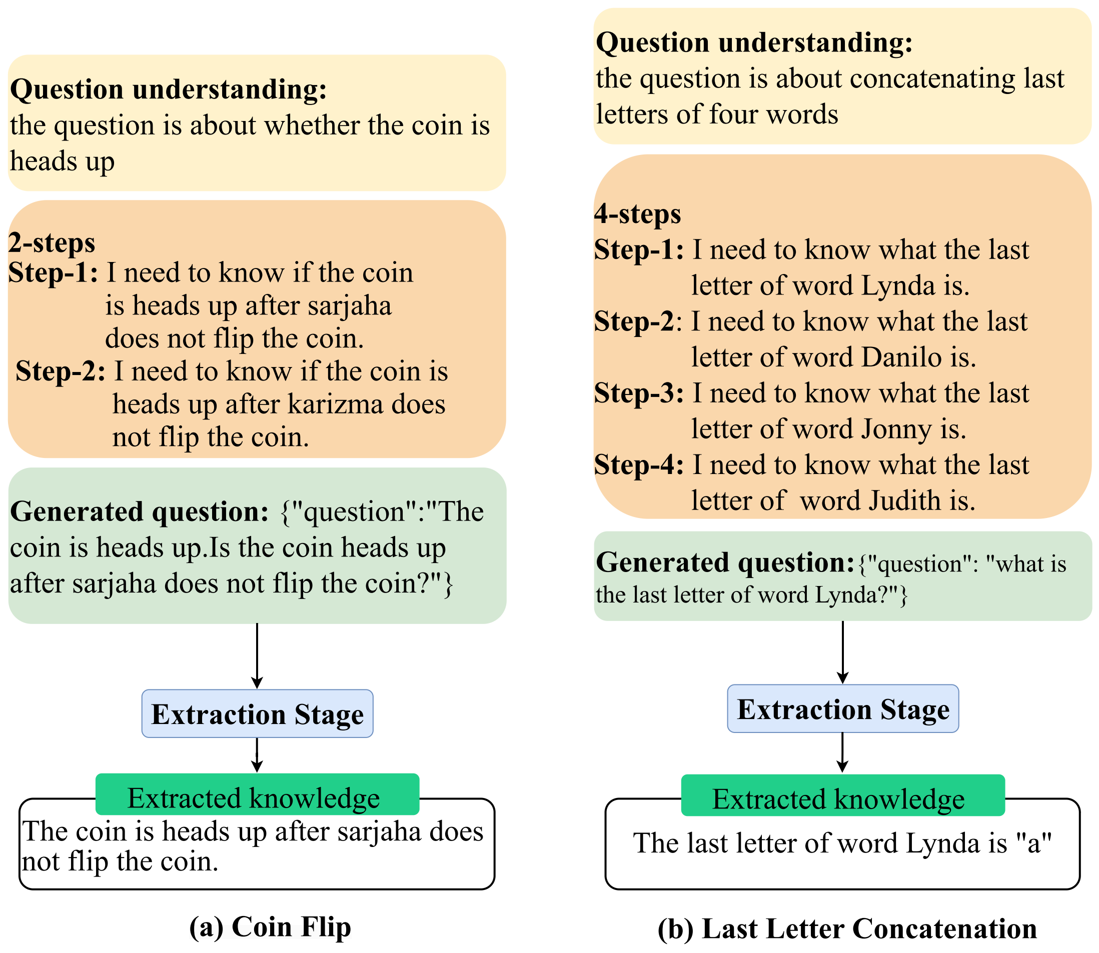
Question understanding: the question is about whether the coin is heads up
2-steps
Step-1: I need to know if the coin is heads up after sarjaha does not flip the coin.
Step-2: I need to know if the coin is heads up after karizma does not flip the coin.
Generated question: {"question":"The coin is heads up.Is the coin heads up after sarjaha does not flip the coin?"}
Extraction Stage
Extracted knowledge
The coin is heads up after sarjaha does not flip the coin.
(a) Coin Flip
Question understanding: the question is about concatenating last letters of four words
4-steps
Step-1: I need to know what the last letter of word Lynda is.
Step-2: I need to know what the last letter of word Danilo is.
Step-3: I need to know what the last letter of word Jonny is.
Step-4: I need to know what the last letter of word Judith is.
Generated question: {"question": "what is the last letter of word Lynda?"}
Extraction Stage
Extracted knowledge
The last letter of word Lynda is "a"
(b) Last Letter Concatenation

Fig. 4. Sub-figure (a) shows a demonstration example on the Coin Flip dataset, with each step of the plan indicating the knowledge. Sub-figure (b) shows a demonstration example on the Last letter Concatenation dataset, with each step of the plan indicating the knowledge.

TABLE V
THE ABLATION STUDIES ON THE THREE DATASETS

Dataset	Removing Extraction Stage	Removing Retrieval	Removing Instruction	**MRM**
Musique	52.00%	28.67%	46.67%	**54.67%**
HotpotQA	66.67%	46.67%	76.67%	**81.33%**
2WikiMultihopQA	73.33%	36.67%	72.00%	**76.67%**

that the answering accuracy scores significantly decrease when removing any one component, in which only removing retrieval from **MRM** holds the worst performance in the three datasets, which demonstrates the significance of retrieval; **Extraction Stage** can help improve the reasoning capability by extracting knowledge from the retrieved documents, so its final answers are higher than other two cases (i.e., only removing retrieval or only removing instruction). Removing the instruction also decreases the answering accuracy, which demonstrates the importance on implementation plan in an order. We do not show such results when removing both retrieval and the instruction, because **MRM** degenerates to the benchmark. The detailed QA examples are shown in Table VI. It is obviously observed that only **MRM** provides the correct answers, removing any component results in incorrect answers. The experimental results demonstrate the necessity of each component in **MRM**.

E. Discussion

In our study, we select three large language models (i.e., GPT-3.5-turbo, Qwen-turbo, and GLM-4) as the backbones, because they are widely used in real-world NLP tasks [33] and many studies utilize them to serve as the backbones such as Few-shot CoT [9] and TDB [34]. The three models have their own advantages, respectively. GPT-3.5-turbo holds the shortest response time among GPT-3 variants [1]; Qwen-turbo adopts the auto-regressive architecture and it directly predicts a next word in text as GPT-3.5-turbo [31]; while GLM-4 is constructed by both auto-regressive and auto-encoding, and it continuously predicts several next words at random positions within text with auto-regressive blank infilling method.

Baseline models like Few-shot CoT usually make the task-specific demonstration examples, and they also heavily rely

TABLE VI
THE ABLATION QA EXAMPLES ON THE THREE DATASETS

Dataset	Question	Removing Extraction	Removing Retrieval	Removing Instruction	**MRM**
Musique	When did Swedish become the official language of the country where the singers of love me love me say that you love me are from?	Sweden has been the official language of Sweden for centuries. Jessie Ware and Ellie Goulding are from the United Kingdom, where English is the official language.	Swedish has never been the official language of the United States or the United Kingdom.	Standard Swedish prevails throughout the country.	**20th century**
HotpotQA	Who is the father of a Belizean rapper featured on the mixtape California Republic?	The question cannot be answered as it refers to a Belizean rapper featured on the mixtape "California Republic," but no specific rapper is mentioned.	I cannot provide the answer without more specific information about the Belizean rapper	This question is asking for the name of the father of a specific Belizean rapper	**Dean Barrow.**
2WikiMultihopQA	Where was the place of death of Nina Caroline Studley-Herbert, 12Th Countess Of Seafield's father?	The place of death for James Ogilvie-Grant, 11th Earl of Seafield, is not provided in the given information.	Banff, Scotland	The father of Nina Caroline Studley-Herbert, was James Ogilvie-Grant, ... However, the information of ... is not provided.	**Flanders, Belgium.**

on the quality, the quantity, and the diversity of those examples. However, **MRM** does not rely on these characteristics, and it solely makes two simple demonstration examples for all datasets, which implicitly demonstrates its insensitivity to the quality and the quantity. Considering the reasoning capability in LLM is usually reflected by validating methods on multi-hop datasets [35], [36], we choose three widely used public multi-hop datasets: Musique, HotpotQA, and 2WikiMultihopQA to compare our **MRM** with baseline methods, with judging whether the output answers equal to the standard answers from datasets. Usually, the number of hops is less than or equal four hops [37], and the widely used datasets (e.g., Musiue, HotpotQA, and 2WikiMultihopQA) have a maximum four hop questions. Therefore, multi-hop question covers up to 4-hop questions in our study. With this, the answering accuracy is employed [22]. The higher the accuracy score is, the more the reasoning capability is. From Tables I–III, we can observe that our **MRM** significantly improves the reasoning capability, because the results from **MRM** show the highest accuracy scores among all methods. It not only demonstrates its effectiveness on two-hop and three-hop questions, but show its significance on the uncommon four-hop questions. Moreover, since the retrieved documents have an influence on improving the reasoning capability [38], we compare our **MRM** with the recent study RAG, which also adopts retrieval method to reason answer, with hit rate metric. Fig. 3 illustrates the effectiveness of **MRM** on retrieving documents regarding to the posed question. Note that although **MRM** targets to reduce irrelevant information, it inevitably retrieves irrelevant information in each step, which might generate the final answer with low quality. From Tables I–IV, we can observe that the accuracy scores from **MRM** are not 100% , even though it achieves the highest accuracy scores among all baseline models in all datasets. However, **MRM** holds the unique mechanism to find out the irrelevant information, which could show the entire reasoning trajectories from the first step to the last one within plan to help discover the irrelevant information. The baseline models (e.g., Few-Shot CoT, IRCoT, TDB, and RAG) do not have such mechanism.

MRM can still handle such a scenario where there are very nuanced subject relationships. Take such a question as the example: "Which country is the largest trading partner of the largest trading partner of the largest trading partner of the United States?". This is a three-hop question, so the plan is designed to hold three steps. According to "I need to know [the knowledge of subject] by [subject]", we have such a plan: The first step: "I need to know the largest trading partner of the United States by the United States", and then extract the relevant knowledge, based on it. The second step: "I need to know the largest trading partner of the country by the country.", in which the country is extracted in the first step. The last step: "I need to know the largest trading partner of country by country.", in which the country is extracted in the second step. Therefore, our **MRM** can handle the nuanced subject relationships.

Besides, we also validate all methods in symbolic reasoning on both Coin Flip and Last letters Concatenation datasets, because the logical reasoning problems (e.g., math [39]) is transferred to the symbolic reasoning [40]. Table 4 shows that our **MRM** still outperforms the baseline methods in symbolic reasoning. Moreover, both Table V and Table VI demonstrate that each component in **MRM** is necessity, and removing any one component would cause the answering accuracy decreasing.

Since **MRM** includes plan design, SQ, extraction, and reasoning stages, one may wonder its computation overhead. Note that the main stream of LLM is to improve the answering accuracy [27], and our **MRM** achieves the best answering performance. Take Musique dataset as the example, we randomly select 100 questions and test the average response times and accuracy scores of all models. The results are shown in Table VII. Although the earliest studies (e.g., Direct prompting, RAG, TDB, and Few-shot CoT) have the faster response time, their answering accuracy scores are lower; the recent studies (e.g., Iter-RetGen and IRCoT) hold the larger response times (e.g., Iter-RetGen for 13.48 s and IRCoT for 14.32 s), while they significantly improve the answering accuracy scores, in which IRCoT only improves by 1.4% compared with Iter-RetGen but

TABLE VII
THE AVERAGE RESPONSE TIMES AND ANSWERING ACCURACY SCORES OF ALL MODELS ON MUSIQUE DATASET

Method	Response time(s)	Accuracy
Direct prompting	1.71	16.00%
TDB	1.64	17.00%
Few-shot CoT	1.72	20.00%
RAG	4.83	36.00%
EfficientRAG	5.18	45.00%
IRCoT	14.32	48.00%
Iter-RetGen iter4	13.48	43.00%
MRM	**15.43**	**52.00%**

they have similar response time. Nevertheless, **MRM** significantly improves by 14.6% compared with IRCoT, which holds the largest accuracy score among all baseline models, and they also have the similar response time.

V. CONCLUSION

To improve the reasoning capability, we propose a new method, named **MRM**, which includes Plan Design and Implementation with a new Retrieval method. The contributions of our **MRM** are as follows: 1) improving the LLM's reasoning capability and showing the reasoning traces in each step within plan; 2) presenting a new retrieval method to retrieve the desired knowledge to LLM; 3) the experimental results demonstrate that our **MRM** significantly outperforms SOTA baseline methods; 4) providing new insights into the understanding of LLM variants.

REFERENCES

[1] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[2] G. S. Black, B. P. Rimal, and V. M. Vaidyan, "Balancing security and correctness in code generation: An empirical study on commercial large language models," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 1, pp. 419–430, Feb. 2025.

[3] Z. Zhang, L. Peng, T. Pang, J. Han, H. Zhao, and B. W. Schuller, "Refashioning emotion recognition modeling: The advent of generalized large models," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 5, pp. 6690–6704, Oct. 2024.

[4] T. Zhou, H. Ishibuchi, and S. Wang, "Stacked-structure-based hierarchical takagi-sugeno-kang fuzzy classification through feature augmentation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 6, pp. 421–436, Dec. 2017.

[5] E. De Santis, A. Martino, F. Ronci, and A. Rizzi, "From bag-of-words to transformers: A comparative study for text classification in healthcare discussions in social media," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 1, pp. 1063–107, Feb. 2025.

[6] Z. Li et al., "Llms for relational reasoning: How far are we?," in *Proc. 1st Int. Workshop Large Lang. Models Code*, 2024, pp. 119–126.

[7] G. Huang, Y. Long, C. Luo, J. Shen, and X. Sun, "Prompting explicit and implicit knowledge for multi-hop question answering based on human reading process," in *Proc. Joint Int. Conf. Comput. Linguist., Lang. Resour. Eval.*, 2024, pp. 13179–13189.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2022.

[9] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 24824–24837.

[10] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *The Eleventh Int. Conf. Learn. Representations*, 2022.

[11] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 22199–22213.

[12] O. Ram et al., "In-context retrieval-augmented language models," *Trans. Assoc. Comput. Linguistics*, vol. 11, pp. 1316–1331, 2023.

[13] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting for retrieval-augmented large language models," 2023.

[14] M. Komeili, K. Shuster, and J. Weston, "Internet-augmented dialogue generation," in *Proc. 60th ann. meeting Assoc. Comput. Linguistics*, vol. 1, 2022.

[15] L. Wang et al., "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," *Long Papers. Assoc. Comput. Linguistics (ACL)*, pp. 2609–2634, 2023.

[16] S. Mishra, D. Khashabi, C. Baral, Y. Choi, and H. Hajishirzi, "Reframing instructional prompts to GPTk's language," *Findings assoc. computat. linguistics: ACL*, 2022.

[17] J. Kaplan et al., "Scaling laws for neural language models," 2020, *arXiv:2001.08361*.

[18] Y. Zheng, Z. Tan, P. Li, and Y. Liu, "Black-box prompt tuning with subspace learning," *IEEE/ACM Trans. Audio, Speech Lang. Proc.*, vol. 32, pp. 3002–3013., 2024.

[19] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist.*, A. J. Rogers Boyd-Graber and N. Okazaki, Eds., Toronto, ON, Canada, Jul. 2023, pp. 1762–1777. [Online]. Available: https://aclanthology.org/2023.acl-long.99

[20] N. F. Liu et al., "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 157–173, 2024.

[21] D. Zhou et al., "Least-to-most prompting enables complex reasoning in large language models," ICLR, 2023.

[22] H. S. Zheng et al., "Take a step back: Evoking reasoning via abstraction in large language models," CoRR abs/2310.06117, 2023.

[23] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Musique: Multihop questions via single-hop question composition," *Trans. Assoc. Comput. Linguistics*, vol. 10, pp. 539–554, 2022.

[24] Z. Yang et al., "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proc. 2018 conf. empirical methods natural lang. process. (C)*, 2018.

[25] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-HOP QA dataset for comprehensive evaluation of reasoning steps," in *Proc. 28th Int. Conf. Comput. Linguistics (C)*, 2020.

[26] X. Wang et al., "KnowledGPT: Enhancing large language models with retrieval and storage access on knowledge bases," 2023, *arXiv:2308.11761*.

[27] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 10014–10037.

[28] Z. Zhuang et al., "Efficientrag: Efficient retriever for multi-hop question answering," in *Proc. Empirical Methods Natural Lang. Process.*, 2024, pp. 3392–3411.

[29] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," in *Proc. Findings Assoc. Comput. Linguistics: EMNLP 2023*, 2023, pp. 9248–9274.

[30] T. U. Zhipu AI, "ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools," CoRR, 2024.

[31] J. Bai et al., "Qwen technical report [J].," 2023, *arXiv:2309.16609*.

[32] X. Wang et al., "Self-consistency improves chain of thought reasoning in language models [J].," 2022, *arXiv:2203.11171*.

[33] A. Chowdhery et al., "Palm: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 240, pp. 1–113, 2023.

[34] C. Yang et al., "Large language models as optimizers," 2023.

[35] T. Khot et al., "Decomposed prompting: A modular approach for solving complex tasks," 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID252715485

[36] F. Yu, H. Zhang, P. Tiwari, and B. Wang, "Natural language reasoning, A survey," *ACM Comput. Surv.*, vol. 56, no. 12, pp. 1–39, Dec. 2024, doi: 10.1145/3664194.

[37] V. Mavi, A. Jangra, and A. Jatowt, "Multi-hop question answering," *Found. Trends Inf. Retrieval*, vol. 17, no. 5, pp. 457–586, 2024.

[38] P. Zhao et al., "Retrieval-augmented generation for AI-generated content: A survey [J].," 2024, *arXiv:2402.19473*.

[39] P. Lu et al., "Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2021.

[40] L. Pan, A. Albalak, X. Wang, and W. Wang, "Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning," in *Proc. Findings Assoc. Comput. Linguist.*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3806–3824. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.248

Wei Li (Member, IEEE) received the Undergraduate degree in information and computing science in 2008, the master's degree in agricultural engineering from South China Agricultural University, Guangzhou, China, in 2012, and the Ph.D. degree in software engineering from Wuhan University, Wuhan, China, in 2019. He was the Visiting Student with the University of Massachusetts Boston, Boston, MA, USA, and had visited the The Hong Kong Polytechnic University, Hong Kong, as Research Assistant. He is currently an Associate Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. His research interests include data mining, artificial intelligence, and federated learning.

Yanqing Zhang received the B.S. degree in electrical engineering and automation from the Henan University of Technology, Zhengzhou, China, in 2021. He is currently working toward the M.S. degree with the School of Artifcial Intelligence and Computer Science, Jiangnan University, Wuxi, China. His research interests mainly include nature language processing and large language model.

Guifang Sun received the Ph.D. degree in materials science from Northestern University, Shenyang, China, in July 2009. She has also studied with the University of Michigan, Ann Arbor, MI, USA, in 2007, 2008, and 2010, respectively. She is currently a Full Professor with the School of Mechanical Engineering, Southeast University, Nanjing, China. Her research interests include powder-based underwater laser additive manufacturing and its applicatioin in marine engineering.

Chao Ma (Member, IEEE) is currently an Assistant Professor with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. He has authored or coauthored more than 30 academic papers in major international journals and conference proceedings. His research interests include time series analytics, representation learning, deep learning, explainable AI, and Big Data analytics. He is also a Professional Member of CCF.

Jipeng Qiang (Senior Member, IEEE) received the Ph.D. degree from the Hefei University of Technology, Hefei, China, in 2016, under the supervision of Professor Wu Xindong. He is currently a Professor and Ph.D. Supervisor with the School of Information and Artificial Intelligence, Yangzhou University, Yangzhou, China. He has authored or coauthored more than 100 papers in international journals and conferences, including AIJ, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, TACL, IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, ACL, CVPR, AAAI, and EMNLP. He has also authored two monographs and two textbooks published by Science Press of China. His research interests include artificial intelligence and large models. He has presided over various national and provincial-level projects, such as the General Program and Youth Program of the National Natural Science Foundation of China, the Key Project of the National Language Commission, and the Youth Program of the Natural Science Foundation of Jiangsu Province. During his Doctoral studies, he was sponsored to study with the University of Massachusetts Boston, Boston, MA, USA, under the mentorship of Professor Ding Wei. He is also an Associate Editor for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE journal. He is an Outstanding Young Backbone Teacher of Jiangsu Province's "Blue and Blue Project". He is a CCF Senior Member.

Jiaxing Shen received the B.E. degree in software engineering from Jilin University, Changchun, China, in 2014, and the Ph.D. degree in computer science from Hong Kong Polytechnic University, Hong Kong, in 2019. In 2017, he was a Visiting Scholar with Media Lab, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently an Assistant Professor with the School of Data Science, Lingnan University, Hong Kong. His research has been authored or coauthored in top-tier journals such as IEEE TRANSACTIONS ON MOBILE COMPUTING, ACM TOIS, ACM IMWUT, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. His research interests include human-centric computing, mobile computing, and Generative AI. He was recipient of the several conference Best Paper including INFOCOM 2020 and ICNP 2025.