RAR-Agent: Retrieval Augmented Reflection Learning from Scratch for Reasoning

Shipeng Xie
Zhejiang Communications Investment Group Co.,Ltd
Hang Zhou, China
`jtgsxieshipeng@cncico.com`

Haichao Zhu
Tencent Media Lab
Tencent America
`lszhuhaichao@gmail.com`

Da Chen[*]
University of Bath
Bath, UK
`da.chen@bath.edu`

Abstract

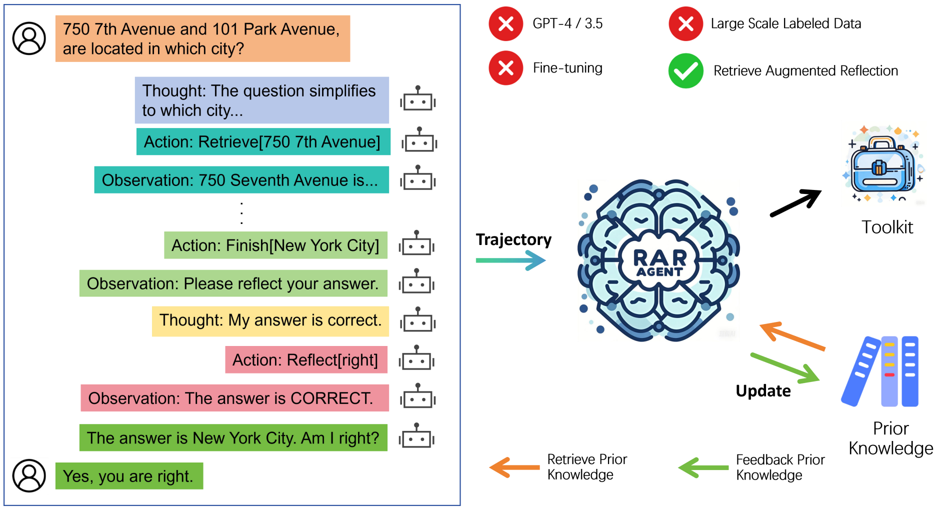
In various complex question-answering scenarios, large-scale model agents have achieved remarkable performance by leveraging external tools for reasoning and planning. Despite incessant exploration in this domain, current large-scale model agent systems still suffer from issues such as high costs, difficulties in relying on repeatable prior knowledge, and the challenge of enabling a single model to fulfill multiple functions in open-world environments. To address these issues, we propose RAR-Agent (Retrieval Augmented Reflection Agent), a framework that learns from scratch for reasoning and knowledge update through retrieval-augmented reflection, without relying on vast annotated data or requiring fine-tuning. Given limited prior knowledge data and a tool library, RAR-Agent first autonomously synthesizes trajectory data for reasoning decisions, bypassing the need for manual annotation or assistance from powerful closed-source models. Subsequently, RAR-Agent autonomously constructs a prior knowledge base and provides with task-specific prior knowledge through retrieval. Through interactive dialogue with users, RAR-Agent collects a small amount of human feedback and leverages a continuous learning mechanism to update its prior knowledge base. We conduct comprehensive experiments with diverse LLMs (Large Language Models), demonstrating that RAR-Agent can achieve better or comparable performance to many benchmarks, all with very little annotated data and no extra fine-tuning required.

1 Introduction

Agents based on Large Language Models (LLMs)[1, 2, 3], leveraging their powerful reasoning capabilities and interacting with executable tools, have emerged as an effective paradigm for designing AI agent systems to tackle complex tasks in open-world settings[4, 5, 6, 7]. However, concerns regarding the interleave reasoning and decision-making capabilities of such agents have garnered increasing attention. Within the reasoning and decision-making process, planning plays a pivotal role[8], decomposing complex tasks into simpler subtasks[9, 10, 11, 12], determining the invocation of external tools[13, 14, 15], and reflecting on past thought trajectories to ultimately accomplish tasks efficiently[16, 17]. Given the low cost and customizability of open-source LLMs, recent efforts[18, 19] have aimed to enhance their planning capabilities through fine-tuning[20, 21].

Preprint. Under review.

[*] Da Chen is the corresponding author



750 7th Avenue and 101 Park Avenue, are located in which city?

Thought: The question simplifies to which city...

Action: Retrieve[750 7th Avenue]

Observation: 750 Seventh Avenue is...

⋮

Action: Finish[New York City]

Observation: Please reflect your answer.

Thought: My answer is correct.

Action: Reflect[right]

Observation: The answer is CORRECT.

The answer is New York City. Am I right?

Yes, you are right.

Figure 1: **The basic framework of RAR-Agent.** Armed with just prior knowledge database and toolkit, the RAR-Agent can reason and reflect based on the retrieval augmented prior knowledge that can collaborate to complete the task. Different colors represent the outcomes of each reasoning step performed by the RAR-Agent. The reasoning at each step is achieved through retrieval-augmented reflection, taking into account both the current contextual situation and the prior knowledge.

Nevertheless, despite the successes achieved by existing fine-tuning-based approaches, they are still subject to limitations. On one hand, training open-source models necessitates a certain amount of manually annotated data. In open-world scenarios, such as personal assistant robots or traffic flow management on highways, satisfying these conditions can be extremely challenging. On the other hand, constructing agents based on fine-tuning essentially forces a single model to learn the planning capabilities for all scenario tasks[22]. While some work has achieved success through division-of-labor approaches[22, 23], they still require substantial computational resources for fine-tuning the models.

Methods based on Retrieval-Augmented Generation (RAG)[24, 25] have attempted to alleviate these issues by exploring collaborative RAG, leveraging retrieved information for long-horizon reasoning[25, 26]. The original intention of such approaches is to assist agents' intermediate reasoning processes with external knowledge bases. However, key questions remain regarding how to collaborate with RAG for reasoning and decision-making and how to continuously learn and update the knowledge base.

To this end, we propose RAR-Agent, a Retrieval-Augmented Reflection Agent learning framework, which does not rely on large-scale manually annotated data or closed-source models while augmented reasoning and continuous learning, as illustrated in Figure 1. Given a small set of example data from task scenarios, RAR-Agent leverages self-instruction[27, 23] to obtain a data-augmented scenario dataset and automatically generates planning trajectory data. Subsequently, we design an advanced RAG system, which serves as prior knowledge for reasoning and decision-making. The prior knowledge is the key to addressing the issue of low-cost scenario adaptation. Finally, we adopt a division-of-labor [23] strategy to enable RAR-Agent to progressively execute tasks. Utilizing task templates, LLMs integrate historical trajectories and prior knowledge to reason and make decision of the next step. Similar to human reasoning processes, this strategy leverages prior knowledge to iteratively reflect, reason, and adjust the process of solving complex long-horizon problems[25, 26]. Furthermore, we propose an interactive continuous learning mechanism that updates the prior knowledge base with minimal human feedback, addressing the issue of updating prior knowledge in RAR-Agent's collaborative reasoning. Table 1 summarizes the differences between RAR-Agent and previous work.

Experiments with various LLMs on complex question-answering and mathematical reasoning tasks demonstrate that RAR-Agent achieves better or comparable performance to many benchmarks. Specifically, we observe improved performance over fine-tuned models, with gains of 1.03% on

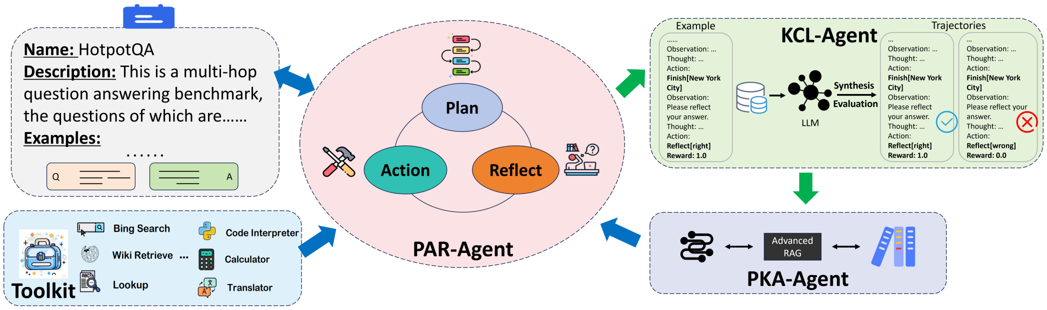


Figure 2: **The overview of our proposed framework RAR-Agent.** It comprises four integral components: the PAR-Agent (Plan-Action-Reflection Agent), responsible for reasoning, decision-making, and reflection; the PKA-Agent (Prior Knowledge Augmentation Agent), tasked with retrieving and augmenting prior knowledge; the KCL-Agent (Knowledge Continuous Learning Agent), designed to facilitate continuous learning through minimal human feedback; and the Toolkit, which supports the agent in accomplishing open-world tasks with exceptional performance. We initiate the task prior knowledge with self-instruct to extend few task examples from scratch.

HotpotQA and 0.9% on ScienceQA for complex question-answering tasks. For mathematical reasoning tasks, we achieve comparable results with gains of 2.32%. Extended experimental analyses validate the effectiveness of prior knowledge augmentation and the continuous learning strategy.

2 RAR-Agent

The RAR-Agent (Retrieval Augmented Reflection Agent) framework (see Figure 2) comprises four primary components: **the PAR-Agent (Plan-Action-Reflection Agent)** , **the PKA-Agent (Prior Knowledge Augmentation Agent)**, **the KCL-Agent (Knowledge Continuous Learning Agent)** and an additional **Toolkit**. In RAR-Agent, the sub-agents can be initialized with any kind of open-source model.

Objective Task Specification. The central emphasis of this research endeavor lies in the exploration of agent learning from scratch, signifying that the available task information is inherently constrained and primarily encompasses three pivotal aspects: task name T, task description D, task data examples E. Concretely, D represents a detailed description of the task's characteristics. $E = \{t_i, a_i\}_{i=1}^{|E|}$ indicates $|E|$ objective and outcome pairs of the task, where $|E|$ is very small which users can effortlessly provide (e.g., a few demonstrations or human feedback).

2.1 Toolkit

To excel in open-world tasks, RAR-Agent necessitates a comprehensive toolkit comprising essential tools tailored to specific scenarios, along with tool descriptions, invocation parameters, and formats. The tool library can be denoted as $A = \{a_i, f_i, u_i\}_{i=1}^{|A|}$, where a represents the tool name, f defines the tool functionality, u details the tool usage instruction, and $|A|$ stands for the tool amount of the library. Detailed descriptions can be found in the experimental design section.

2.2 KCL-Agent

The KCL-Agent (Knowledge Continuous Learning Agent) is crucial for continuous learning prior knowledge within the RAR-Agent framework, as shown in Figure 2. During system cold start, it initializes prior knowledge through data augmentation of a small set of manually annotated data E, expanding them with LLM for data augmentation E_x . Then, we execute reasoning tasks in E_x with the PAR-Agent (initially as a traditional agent without prior knowledge augmentation). In order to obtain high-quality synthesized trajectories, we filter out all the trajectories with $reward < 1$ and collect trajectories with exactly correct answers ($reward = 1$) as the prior knowledge, as depicted in Figure 2.

We propose updating prior knowledge through interactive human feedback. Specifically, the KCL-Agent samples human feedback during interactive task execution (e.g., satisfaction with answers in QA tasks in Figure 1). It treats this feedback as new prior knowledge, which is updated into the PKA-Agent (Figure 2). Through iterations, the KCL-Agent continuously learns from minimal human feedback to adapt to scene tasks.

2.3 PKA-Agent

The PKA-Agent (Prior Knowledge Augmentation Agent) significantly enhances the reasoning and decision-making process by leveraging prior knowledge. Inspired by [26], we propose a reasoning trajectory augmentation method based on the advanced Retrieval Augmented Generation (RAG), optimizing for reasoning trajectories through sparse and dense representations. We treat correctly executed reasoning trajectories as prior knowledge and embed them into vector database.

Specifically, we assume PKA-Agent as function $K = Y(query)$, where Y is the advanced RAG system, $query$ is the current trajectory of the PAR-Agent, and K is the prior knowledge retrieved from the vector database. Notably, K is empty before prior knowledge initialized. The KCL-Agent module oversees knowledge updates, as detailed in section KCL-Agent.

2.4 PAR-Agent

The PAR-Agent (Plan-Action-Reflection Agent) serves as the main agent within the RAR-Agent framework. Inspired by [23], we integrate plan, action and reflection for reasoning, decision-making, and reflection, as depicted in Figure 2.

The Plan phase R_{plan}, analyzes the context based on the task execution trajectory and augments this with retrieved prior knowledge. It perceives the current task status and anticipates future states, as illustrated by the light blue thought in Figure 1.

The Action phase R_{act}, utilizing the task execution trajectory as a query, retrieves and augments knowledge from the prior knowledge base. Based on the comprehensive context, it decides the next action and selects appropriate tools, as indicated by the cyan action and observation in Figure 1.

The Reflection phase $R_{reflect}$, occurs after the action is marked as finished. This phase revisits the reasoning trajectory to reflect the reasoning and decision-making process, as illustrated by the red color in Figure 1. The RAR-Agent framework boasts reflection capabilities to adjust reasoning trajectories through retrieved prior knowledge. Experimental results further substantiate the conclusion.

We assume that the planning loop at time t can be denoted as (p_t, a_t, o_t), where p denotes Thought, a signifies Action, and o represents Observation. a can be further expressed as $({}^n a_t, {}^s a_t)$, where ${}^n a_t$ is the name of the action, and ${}^s a_t$ is the parameters required to perform the action. Then the historical trajectory at time t can be signaled as:

$$H_t = (p_0, a_0, o_0, p_1, ..., p_{t-1}, a_{t-1}, o_{t-1}) \tag{1}$$

Eventually, supposing that PKA-Agent is initialized, the prior knowledge K_t at time t is:

$$K_t = Y(H_t) \tag{2}$$

Then, the responsibilities of each phase can be defined as:

$$p_t, {}^n a_t = R_{plan}(H_t, K_t) \tag{3}$$

$${}^s a_t = R_{action}(H_t, K_t, p_t, {}^n a_t) \tag{4}$$

$$p_t^r, a_t^r = R_{reflect}(H, K) \tag{5}$$

where p_t^r and a_t^r represent the thought and action of the reflection process, and H, K is the planning history and prior knowledge after finishing the answer.

Method	Data Acquisition	Trajectory Acquisition	Multi-Agent	Fine-Tuning	Generality	Reflection	Continual Learning
REACT [10]	User	Prompt	×	×	✓	×	×
Reflexion [14]	User	Prompt	×	×	✓	✓	×
Camel [18]	User	Prompt	✓	×	✓	×	×
Chameleon [14]	User	Prompt	×	×	✓	×	×
HuggingGPT [13]	User	Prompt	×	×	✓	×	×
AutoGPT [5]	User	Prompt	×	×	✓	✓	×
BOLAA [28]	User	Prompt	✓	×	✓	×	×
AgentVerse [29]	User	Prompt	✓	×	✓	×	×
AgentTuning [20]	Benchmark	GPT-4	×	✓	×	×	×
FIREACT [19]	Benchmark	GPT-4	×	✓	×	✓	×
Lumos [21]	Benchmark	Benchmark+GPT-4	✓	✓	×	×	×
AUTOACT [23]	User + Self-Instruct	Self-Planning	✓	✓	✓	✓	×
RAR-Agent (ours)	User + Self-Instruct	Self-Planning	✓	×	✓	✓	✓

Table 1: **Comparison to other baselines.** Data and Trajectory Acquisitions refer to the way for obtaining training data and trajectories. Multi-Agent indicates whether the framework contains multi-agent. Fine-Tuning stands for whether the method is a fine-tuning-based agent learning framework. Generality signifies whether the method is applicable to various tasks. Reflection denotes whether the planning process incorporates reflection. Continual Learning represents whether the method possesses the ability to continually learn through interactive means.

3 Experiments

We evaluate our proposed RAR-Agent approach on several distinct evaluation benchmarks, demonstrating its effectiveness in multi-step reasoning and long-horizon reasoning and decision-making. We cordially recommend readers to refer to Section 3.4 (Analysis) for a more detailed discussion.

3.1 Experimental Setups

We adopt three groups of benchmarks.

HotpotQA [30] is a multi-hop QA task challenging for rich background knowledge, the answer of which is usually a short entity or yes/no. Following [28], we randomly select 300 dev questions divided into three levels for evaluation, with 100 questions in each level.

ScienceQA [31] is a multi-modal QA task spanning various scientific topics. We also divide the test set into three levels based on the grade, with 120 randomly sampled data in each level. Note that due to the limitations of LMs in generating images, for ScienceQA, during the prior knowledge initiation, we directly generate captions for the images instead.

GSM8K and GSM-HARD [32, 33], which comprises thousands of multi-step mathematical problems. We conduct mathematical reasoning evaluation on both datasets.

Evaluation Metrics. For HotpotQA, the $reward \in [0, 1]$ is defined as the F1 score grading between the prediction and groundtruth answer. For ScienceQA, since it is a multi-choice task, the $reward \in \{0, 1\}$ is exactly the accuracy. For GSM8K and GSM-HARD, we compute accuracy to evaluate every question in mathematical reasoning tasks, aligning with the established metric for the GSM8K [26].

Baselines. We choose the open-source ChatGLM4-9B [34], Llama-2 models [35] as the backbones of our RAR-Agent and its sub-agents. The compared baselines include CoT [9], REACT [10], Chameleon [14], Reflexion [16], BOLAA [28], ReWOO [36], FIREACT [19], AutoAct [23], RAT [26]. To ensure fairness, we maintain an equal training trajectory volume of 200 for FIREACT and AUTOACT (200 synthesized data). We use 200 synthesized trajectory data as prior konwledge for RAR-Agent without fine-tuning. As Reflexion provides answer correctness labels during reflection but other methods including RAR-Agent do not, we test all the other methods twice and choose the correct one for evaluation. For all the prompt-based baselines, we uniformly provide two examples in the prompt. For mathematical reasoning, we choose the open-source ChatGLM4-9B [34] as base model for RAR-Agent and other baselines. Similar to [26], DIRECT is the original language models.

Other Setups. Despite no fine-tune for RAR-Agent, we fine-tune other baseline models [20, 19, 21, 23] with LoRA [37] in the format proposed in Alpaca [38]. All the training and inference experiments are conducted on 8 V100 GPUs within 16 hours.

Backbone	Method	HotpotQA				ScienceQA			
		Easy	Medium	Hard	All	G1-4	G5-8	G9-12	All
GPT-3.5	CoT	48.21	44.52	34.22	42.32	60.83	55.83	65.00	60.56
	Zero-Shot Plan	50.71	45.17	38.23	44.70	76.67	61.67	78.33	72.22
Llama-2 13B-chat	CoT	37.90	25.28	21.64	28.27	61.67	52.50	69.17	61.11
	ReAct	28.68	22.15	21.69	24.17	57.50	51.67	65.00	58.06
	Chameleon	40.01	25.39	22.82	29.41	69.17	60.83	73.33	67.78
	Reflexion	44.43	37.50	28.17	36.70	67.50	64.17	73.33	68.33
	BOLAA	33.23	25.46	25.23	27.97	60.00	54.17	65.83	60.00
	ReWOO	30.09	24.01	21.13	25.08	57.50	54.17	65.83	59.17
	FireAct	45.83	38.94	26.06	36.94	60.83	57.50	67.50	61.94
	AUTOACT	**47.29**	**41.27**	**32.92**	**40.49**	**70.83**	**66.67**	**76.67**	**71.39**
	RAR-Agent	42.39	38.28	21.34	34.00	62.59	52.55	66.49	60.54
ChatGLM4-9B	CoT	37.90	25.28	21.64	28.27	61.67	52.50	69.17	61.11
	ReAct	28.68	22.15	21.69	24.17	57.50	51.67	65.00	58.06
	Chameleon	40.01	25.39	22.82	29.41	69.17	60.83	73.33	67.78
	Reflexion	44.43	37.50	28.17	36.70	67.50	64.17	73.33	68.33
	BOLAA	33.23	25.46	25.23	27.97	60.00	54.17	65.83	60.00
	ReWOO	30.09	24.01	21.13	25.08	57.50	54.17	65.83	59.17
	FireAct	45.83	38.94	26.06	36.94	60.83	57.50	67.50	61.94
	AUTOACT	49.96	44.27	**35.92**	43.38	72.90	68.97	**79.07**	73.65
	RAR-Agent	**50.09**	**45.76**	35.01	**43.62**	**73.66**	**69.32**	78.23	**73.74**
Llama-2 70B-chat	CoT	45.37	36.33	32.27	37.99	74.17	64.17	75.83	71.39
	ReAct	39.70	37.19	33.62	36.83	64.17	60.00	72.50	65.56
	Chameleon	46.86	38.79	34.43	40.03	77.83	69.17	76.67	74.56
	Reflexion	48.01	46.35	35.64	43.33	75.83	67.50	78.33	73.89
	BOLAA	46.44	37.29	33.49	39.07	70.00	67.50	75.00	70.83
	ReWOO	42.00	39.58	35.32	38.96	65.00	61.67	76.67	67.78
	FireAct	50.82	41.43	35.86	42.70	72.50	68.33	75.00	71.94
	AUTOACT	56.94	50.12	38.35	48.47	82.50	72.50	80.83	78.61
	RAR-Agent	**58.29**	**51.36**	**38.84**	**49.50**	**83.84**	**73.49**	**81.19**	**79.51**

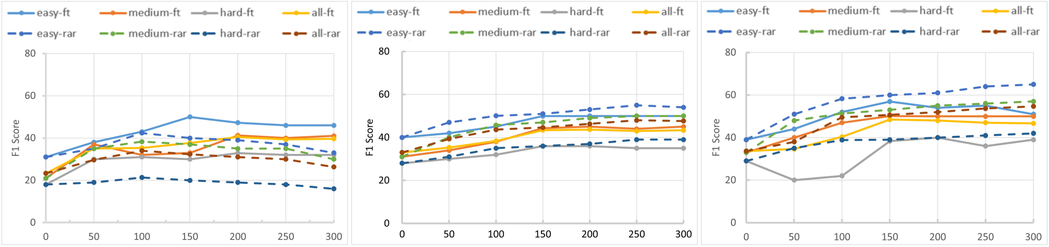
Table 2: **Main results of RAR-Agent compared to various baselines** on HotpotQA and ScienceQA. The FireAct, AUTOACT are fine-tuning-based agent learning, while other methods are prompt-based agent learning without fine-tuning. Refer to Table 1, methods are based on single-agent learning and symbolizes multi-agent learning. The best results of each model are marked in bold and the second-best results are marked with underline.

3.2 Results

Comparison with Template-based Agent Learning Baselines. As shown in Table 2 , the RAR-Agent, leveraging either the ChatGLM4-9B or the LLaMA-2-70B model, consistently outperforms various template-based baselines. Notably, the RAR-Agent based on LLaMA-70B even surpasses the performance of GPT-3.5-based agents, achieving an average improvement of 4.5% on HotpotQA and 7.21% on ScienceQA. However, RAR-Agents utilizing LLaMA-2-13B models exhibit slightly inferior results compared to some template-based baselines. This underscores the challenge of accurately customizing agent behavior through template-based methods, whether in single-agent or multi-agent frameworks. Nevertheless, the RAR-Agent's outstanding performance hinges critically on the robust reasoning and decision-making capabilities of the underlying LLMs.

Comparison with Fine-tuning-based Agent Learning Baselines. As evident in Table 2, AUTOACT decomposes the planning process into division-of-labor sub-agents, achieving better performance gains than FireAct, albeit at the cost of requiring fine-tuning the model. The RAR-Agent based on LLaMA-2-70B outperforms all fine-tuning based baselines, without the need for fine-tuning or extensive manually annotated data. This paves the way for continuous agent learning from scratch using open-source models. Similarly, we observe that the RAR-Agent's advantages diminish when the LLMs' reasoning capabilities are insufficient.

Comparison with Retrieval-Augmented Agent Baselines. For mathematical reasoning tasks in Table 3, the RAR-Agent demonstrates remarkable performance, achieving a 14.86% accuracy improvement on GSM8K and a 37.57% boost on GSMHard, reaching SOTA results comparable to RAT. Notably, the RAR-Agent, grounded in ChatGLM4-9B, attains performance parity with GPT-3.5. We speculate that this is attributed to the synergy of retrieval-augmented prior knowledge and the



(a) 13B-model 13B-data

(b) 9B-model 9B-data

(c) 70B-model 70B-data

Figure 3: **Performance of RAR-Agent on HotpotQA, comparing to fine-tune based method.** The (13,70)B represents Llama-2-(13,70)B-chat models and 9B indicates ChatGLM4-9B model respectively. (a-c) shows the results of the RAR-Agent augmented by prior knowledge and the fine-tune based models trained on synthesized trajectories. The dashed line is the results of RAR-Agent.

Method	Math Reasoning Accuracy		
	GSM8K	GSMHard	Average(△)
GPT-3.5 DIRECT	65.85%	51.26%	58.56%
DIRECT	60.85%	46.26%	53.56%
CoT	57.82%	39.72%	48.77(-8.94)%
RAG-1 shot	58.89%	48.26%	53.58(+0.01)%
RAG-5 shot	58.81%	52.78%	55.80(+4.17)%
RAT	66.56%	62.34%	64.45(+20.33)%
RAR-Agent	**69.89%**	**63.64%**	**66.77(+24.66)%**

Table 3: **Evaluation results on mathematical reasoning**. For mathematical reasoning, we use ChatGLM4-9B as base model of RAR-Agent and other baselines. △ represents the relative improvements than DIRECT.

division-of-labor sub-agents' capabilities. Furthermore, the RAR-Agent's continuous learning ability endows it with richer prior knowledge across diverse task scenarios, as detailed in our analysis and discussions in Section 3.4.

3.3 Ablation Study

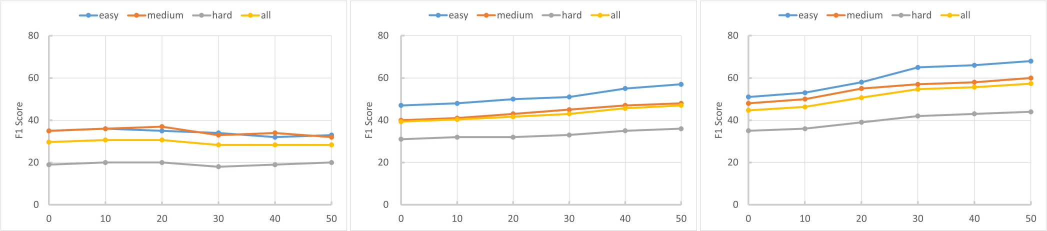
Table 4 presents the performance of RAR-Agent on the Llama-2-70B model after removing certain key processes.

Ablation Study on Prior Knowledge Retrieval Augmentation in RAR-Agent (- knowledge). As evident from Table 4, removing the augmentation of prior knowledge retrieval has a significant negative impact on the overall performance of the RAR-Agent. We hypothesize that this is due to two primary factors: firstly, the prior knowledge may contain relevant questions or content snippets that serve as valuable references; secondly, the execution traces of other specific tasks, as a form of prior knowledge, enable LLMs to draw upon similar experiences for reasoning and decision-making. Without this component, LLMs are solely reliant on historical trajectories for inference, devoid of the beneficial prior knowledge.

Ablation Study on Reflection in RAR-Agent (- reflection). Our observations indicate that ablating the reflection component has minimal impact on the overall performance of the RAR-Agent. In the zero-shot scenario, we find that the model tends to overconfidently make reasoning decisions, echoing similar conclusions reported in (Huang et al. 2024a; AutoAct 2024). Typically, it can only identify obvious formatting errors or repetitive patterns in reasoning decisions. However, in our experiments, when combined with retrieval-augmented prior knowledge, the RAR-Agent is capable of reflecting not only during each step of the reasoning process but also on the task execution trajectory and outcomes.

	HotpotQA	ScienceQA
RAR-Agent	49.50	79.51
- knowledge	38.36 ↓ 11.14	61.39 ↓ 18.12
- reflection	46.53 ↓ 2.97	75.94 ↓ 3.57
- learning	37.69 ↓ 11.81	60.12 ↓ 19.39

Table 4: **Approach ablations of RAR-Agent**. **- knowledge** denotes ablation study on prior knowledge retrieval augmentation in RAR-Agent. **- reflection** symbolizes removing the reflection componen in RAR-Agent. **- learning** indicates knowledge continual learning for updating prior knowledge defined in RAR-Agent.



(a) 13B-model 13B-data

(b) 9B-model 9B-data

(c) 70B-model 70B-data

Figure 4: Performance of RAR-Agent on HotpotQA based on different degrees of interactive human feedback. The (13,70)B represents Llama-2- (13,70)B-chat models and 9B indicates ChatGLM4-9B model respectively. (a-c) shows the results of the RAR-Agent augmented by prior knowledge. The horizontal axis represents different feedback iteration rounds to RAR-Agent.

Ablation Study on Knowledge Continual Learning in RAR-Agent (- learning). To underscore the necessity of continual learning for updating prior knowledge, we specifically removed the process of continual knowledge learning to test the RAR-Agent's effectiveness. We simulated varying degrees of continual learning by using different proportions of prior knowledge. Based on human feedback during task execution, it represents the extent to which prior knowledge is updated . As shown in Table 4, the prior knowledge accumulated through continual learning under task scenarios is crucial for the overall system, its importance being on par with that of prior knowledge retrieval augmentation.

3.4 Analysis

Here we employ the HotpotQA task within the complex question answering scenario for our analysis and discussion.

Fine-tuning based on scenario-specific tasks does not necessarily mean better choice. We evaluate the performance disparity between fine-tuning methods based on various foundation models and our RAR-Agent on HotpotQA, as illustrated in Figure 3(a-c). It can be observed that when the reasoning capacity of the foundation model is insufficient, the fine-tuning-based approach exhibits a notable advantage. As the foundation model becomes larger or its inherent reasoning capability strengthens, the superiority of RAR-Agent promptly manifests, swiftly outperforming the fine-tuning-based method. Furthermore, across different foundation models, it is discernible that the fine-tuning-based approach necessitates over 150 manually annotated data points to yield satisfactory results. Notably, once the number of manually annotated data exceeds 200, the overall performance of fine-tuning-based methods across models plateaus, failing to demonstrate further improvement. In contrast, RAR-Agent is capable of enhancing performance even with minimal annotated data, and its performance continues to improve as the data volume increases. This enables RAR-Agent-based intelligent systems to swiftly cold-start in novel task scenarios and adaptively align with the characteristics of tasks within those scenarios.

Only robust models are capable of acquiring high-quality prior knowledge. Similarly, on HotpotQA, we evaluate the quality of prior knowledge (reasoning trajectories) generated by models with varying levels of capabilities and its direct impact on the performance of RAR-Agent, as depicted in Figure 3(a-c). The prior knowledge generated from different models exhibits quality variations, with the 70B model yielding the highest quality data and thus the best performance. As the quality of prior knowledge generated by the models improves, the overall performance of RAR-Agent also enhances, as indicated by the dashed lines in Figure 3(a-c).

The advantage of RAR-Agent is unleashed by powerful models. From the experimental results in Table 2 and Figure 3, we observe that when the base LLM has a relatively small parameter size or insufficient reasoning ability, RAR-Agent's performance is moderate. However, as the size of the base LLM's parameters increases or its inherent reasoning ability strengthens, the quality of generated prior knowledge improves. Through retrieval-augmented reflection, RAR-Agent's reasoning capability is rapidly fortified, leading to a swift improvement in performance on the evaluation set. We can conclude that although RAR-Agent does not require fine-tuning, it heavily relies on the reasoning and decision-making capabilities of the base model itself.

The continually accumulated prior knowledge through learning can effectively and consistently enhance the overall performance. When RAR-Agent interacts with humans to execute tasks, we evaluate the impact of the volume of receiving human feedback on its continuous learning ability. As shown in Figure 4(a-c), once the reasoning capability of the base model reaches a certain level, as the number of human feedback interactions increases, RAR-Agent learns more prior knowledge, resulting in progressively better performance on HotpotQA. This continuous learning capability offers the potential for RAR-Agent to be extended and applied to other real-world scenarios in open environments.

4 Related work

LLM-Powered Agent Fine-Tuning. The rise of LLMs has positioned them as the most promising key to providing robust support for the development of LLM-centered AI agents [1, 39, 40, 41] . Despite the vast interest in LLM-powered agents, the construction of agents through fine-tuning has received limited attention. Recently, more works have emphasized endowing open-source LLMs with agent capabilities through fine-tuning [19, 20, 21, 42]. However, these works suffer from the need for a large amount of annotated data and trajectory annotation. Our approach enables the RAR-Agent to synthesize trajectories and continually learn knowledge from few human feedback, without relying on fine-tuning.

Retrieval-augmented Generation (RAG). RAG is a cost-effective way for LLMs to interact with the external world [43, 24] . RAG is widely applied to downstream tasks, such as code generation[44, 45, 46], question answering [47, 48], and creative writing [49, 50] .

RAG-enhanced Reasoning. Some recent works also leverage RAG [18] to enhance the performance of LLM-based reasoning. For example, IRGR [51] performs iteratively retrieval to search for suitable premises for multi-hop QA, GEEK [52] can choose to query external knowledge or perform a single logical reasoning step in long-horizon generation tasks, and ITRG [53] performs retrieval based on the last-step generation. However, these previous RAG methods simply adopt a single query to retrieve the knowledge for question-answering tasks [53, 33] , while our proposed RAR-Agent performs retrieval using reasoning trajectory in an autoregressive way, which significantly improves the performance of open-world reasoning and decision-making in various tasks as demonstrated in Figure 2.

5 Conclusion and Future Work

In this paper, we introduce RAR-Agent, a retrieval-augmented reflection agent learning framework that synergizes retrieval augmentation and continual learning, without relying on large-scale manually annotated data or closed-source model-generated planning trajectory data. We aim to tackle the challenges of long-horizon reasoning in open-world settings and cost-effective scenario task adaptation. Our key idea revolves around augmenting the reasoning process of Large Language Models (LLMs) through the incorporation of retrieved prior knowledge, while leveraging the inherent powerful reasoning capabilities of LLMs.

References

[1] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024.

[2] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges, 2024.

[3] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023.

[4] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

[5] Significant Gravitas. AutoGPT.

[6] Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning, 2024.

[7] Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luoxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, Leo Z. Liu, Yiheng Xu, Hongjin Su, Dongchan Shin, Caiming Xiong, and Tao Yu. Openagents: An open platform for language agents in the wild, 2023.

[8] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of llm agents: A survey, 2024.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[10] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[11] XAgent Team. Xagent: An autonomous agent for complex task solving, 2023.

[12] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development, 2024.

[13] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face, 2023.

[14] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models, 2023.

[15] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023.

[16] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

[17] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

[18] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society, 2023.

[19] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning, 2023.

[20] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms, 2023.

[21] Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Agent lumos: Unified and modular training for open-source language agents, 2024.

[22] Michael Mintrom. 12Herbert A. Simon, Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization. In *The Oxford Handbook of Classics in Public Policy and Administration*. Oxford University Press, 03 2015.

[23] Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. Autoact: Automatic agent learning from scratch for qa via self-planning, 2024.

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[25] Keith J. Holyoak and Robert G. Morrison. *The Oxford Handbook of Thinking and Reasoning*. Oxford University Press, 03 2012.

[26] Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation, 2024.

[27] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics.

[28] Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents, 2023.

[29] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors, 2023.

[30] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[31] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 2507–2521. Curran Associates, Inc., 2022.

[32] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[33] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023.

[34] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao

Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[36] Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. Rewoo: Decoupling reasoning from observations for efficient augmented language models, 2023.

[37] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[38] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[39] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey, 2023.

[40] Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models, 2023.

[41] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents, 2024.

[42] Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. Small llms are weak tool learners: A multi-llm agent, 2024.

[43] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. Search engine guided non-parametric neural machine translation, 2018.

[44] Shuai Lu, Nan Duan, Hojae Han, Daya Guo, Seung won Hwang, and Alexey Svyatkovskiy. Reacc: A retrieval-augmented code completion framework, 2022.

[45] Noor Nashid, Mifta Sintaha, and Ali Mesbah. Retrieval-based prompt selection for code-related few-shot learning. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 2450–2462, 2023.

[46] Shuyan Zhou, Uri Alon, Frank F. Xu, Zhiruo Wang, Zhengbao Jiang, and Graham Neubig. Docprompting: Generating code by retrieving the docs, 2023.

[47] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, 2023.

[48] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering, 2022.

[49] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

[50] Zhihua Wen, Zhiliang Tian, Wei Wu, Yuxin Yang, Yanqi Shi, Zhen Huang, and Dongsheng Li. GROVE: A retrieval-augmented complex story generation framework with a forest of evidence. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3980–3998, Singapore, December 2023. Association for Computational Linguistics.

[51] Danilo Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henry Zhu, Xinchi Chen, Zhiheng Huang, Peng Xu, Andrew Arnold, and Dan Roth. Entailment tree explanations via iterative retrieval-generation reasoner, 2022.

[52] Chang Liu, Xiaoguang Li, Lifeng Shang, Xin Jiang, Qun Liu, Edmund Lam, and Ngai Wong. Gradually excavating external knowledge for implicit complex question answering. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14405–14417, Singapore, December 2023. Association for Computational Linguistics.

[53] Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. Retrieval-generation synergy augmented large language models, 2023.