Towards Verifiable Text Generation with Generative Agent

Bin Ji[1*], Huijun Liu[1*], Mingzhe Du[2], Shasha Li[1†], Xiaodong Liu[1†], Jun Ma[1†], Jie Yu[1†], See-Kiong Ng[3]

[1]College of Computer Science and Technology, National University of Defense Technology
[2]Nanyang Technological University
[3] National University of Singapore
{jibin, liuhuijun, shashali, liuxiaodong, majun, yj}@nudt.edu.cn, {mingzhe, seekiong}@nus.edu.sg

Abstract

Text generation with citations makes it easy to verify the factuality of Large Language Models' (LLMs) generations. Existing one-step generation studies expose distinct shortages in answer refinement and in-context demonstration matching. In light of these challenges, we propose R^2-MGA, a Retrieval and Reflection Memory-augmented Generative Agent. Specifically, it first retrieves the memory bank to obtain the best-matched memory snippet, then reflects the retrieved snippet as a reasoning rationale, next combines the snippet and the rationale as the best-matched in-context demonstration. Additionally, it is capable of in-depth answer refinement with two specifically designed modules. We evaluate R^2-MGA across five LLMs on the ALCE benchmark. The results reveal R^2-MGA' exceptional capabilities in text generation with citations. In particular, compared to the selected baselines, it delivers up to +58.8% and +154.7% relative performance gains on answer correctness and citation quality, respectively. Extensive analyses strongly support the motivations of R^2-MGA.

Introduction

With the advent of Large Language Models (LLMs), they have shown exceptional capabilities in completing various complex tasks like text generation (Wang et al. 2023b). Although these generations are usually coherent, they may be unfaithful because of LLMs' tendency to hallucinate. To authenticate the factuality, a novel research topic of text generation with citations is brought to the public (Gao et al. 2023; Funkquist et al. 2023), which focuses on instructing LLMs to provide citations to support their generations.

Research on text generation with citations begins with commercial search engines such as Bing Chat[1], which integrate with LLMs like GPT models (OpenAI 2023). In particular, LLMs generate answers to given questions by synthesizing and citing searched results. However, these scenarios fall short of automatically evaluating the generation quality, which call for high-cost and low-efficiency human evaluations instead (Liu, Zhang, and Liang 2023). In light of this challenge, Gao et al. (2023) propose ALCE, an automatic evaluation benchmark for text generation with citations. Extensive efforts have been taken centering around ALCE. The one-step generation study is a representative line of work, which directly prompts LLMs to generate answers to questions by synthesizing retrieved documents, as the example in Figure 1(b) shows. Gao et al. (2023) explore a series of LLMs in the one-step generation study and propose several off-the-shelf document retrieval settings. Ji et al. (2024) incorporate Chain-of-Thought into the one-step generation study. Li et al. (2023) and Li et al. (2024) investigate advanced techniques to improve document retrieval accuracy in the one-step generation study.

Despite the great success achieved by existing one-step generation studies, they expose two distinct shortages: (1) They fail to provide matched in-context demonstrations. To be more precise, they typically use the same demonstrations across all cases. However, the question type in the demon-

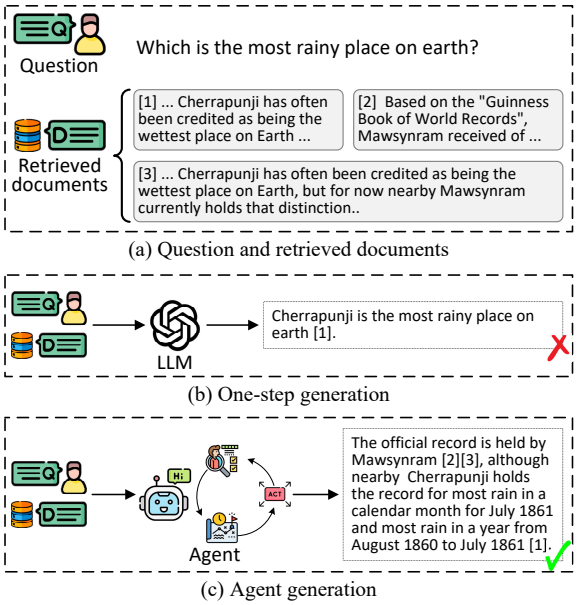

Figure 1: Illustrations of different types of text generation with citations. ✓ and ✗ denote correct and wrong generation cases, respectively.

*These authors contributed equally.
†Corresponding author.

[1]https://www.bing.com/new

strations may mismatch the type of the to-answer question, which is likely to confuse LLMs and mislead them to generate inferior answers. (2) They cannot further refine their generations. To be specific, like humans, LLMs do not always output the best generation on their first try (Madaan et al. 2024), indicating the great potential in improving the generation quality.

To tackle the above challenges, we propose an LLM-powered agent, named Retrieval and Reflection Memory-augmented Generative Agent (R^2-MGA), which consists of four modules, *i.e.*, Memory, Initialization, Assessment, Planning & Action. R^2-MGA is designed to generate high-quality and verifiable answers to questions by using the best-matched demonstration for in-context learning supervision and further refining the answers to improve their quality. Specifically, the Memory module contains a memory bank that stores three types of R^2-MGA's histories, namely full-generation history, assessment history, and planning history. Given a question and retrieved documents, the Initialization module first generates an initial answer; then, the Assessment module assesses the answer quality from multiple aspects and returns feedback; finally, the Planning & Action module first plans future actions based on the feedback and then executes concrete actions to refine the initial answer, as shown in Figure 1(c). Each module necessitates a 1-shot demonstration for in-context learning supervision. To obtain the best-matched one, we first retrieve the best-matched memory snippet from the memory bank; then, we reflect the retrieved snippet by generating a high-level reasoning rationale; finally, we combine the memory snippet and the rationale as the best-matched in-context demonstration.

We evaluate R^2-MGA on five state-of-the-art LLMs. Experimental results on the ALCE benchmark demonstrate that R^2-MGA significantly improves the generation quality, including answer correctness, citation recall and precision. Specifically, (1) compared to one-step generation baselines, R^2-MGA delivers up to +58.8%, +139.3%, and +154.7% relative performance gains on the Correctness, Citation Recall and Precision, respectively; (2) the LLaMA-70B-powered R^2-MGA consistently outperforms the ChatGPT-based one-step generation baseline, and even achieves competitive performance compared to the GPT-4-based baseline. Extensive analyses and ablation studies further firmly verify the effectiveness of R^2-MGA.

Related Work

Text Generation with Citations. This research topic is brought to the public due to LLMs' tendency to hallucinate. In practical scenarios, existing efforts incorporate LLMs into commercial search engines like Bing Chat, Perplexity[2], and Coral[3]. These systems directly generate answers to questions with verifiable links to web pages. However, it's hard to automatically evaluate the answer correctness with these links. In academic research scenarios, several benchmarks are specifically designed to evaluate text generation with citations automatically, *e.g.*, ALCE (Gao et al. 2023) and CiteBench (Funkquist et al. 2023). Centering around these benchmarks, Li et al. (2023), Ji et al. (2024), Li et al. (2024), Wei, Chen, and Meng (2024), and Xia et al. (2024) explore the one-step generation approaches. Ye et al. (2024) and Huang et al. (2024) fine-tune LLMs for the task. However, existing studies use the same in-context demonstrations across all cases, which may mismatch with the to-answer question, decreasing the utility of in-context learning. In contrast, R^2-MGA retrieves its memory and generates reasoning reflections to obtain the best-matched demonstration.

LLM-powered Agents. AI agents have long been regarded as a feasible way to achieve artificial general intelligence (Wang et al. 2024). Compared to LLMs, agents should learn and complete tasks in dynamic environments and borrow experience from their memory to facilitate coherent and believable actions (Park et al. 2023). LLM-powered agents take LLMs as the backbone and have been comprehensively explored, such as general agent (Wei et al. 2022; Yao et al. 2024; Qiao et al. 2024), simulation agent (Park et al. 2023), tool agent (Qin et al. 2024; Gou et al. 2024), embodied agent (Cho, Yoon, and Ahn 2024; Choi et al. 2024; Zheng et al. 2024; Zhang et al. 2024), conversational agent (Deng et al. 2024a,b), and game agent (Wang et al. 2023a). As for the text generation with citations, Lee et al. (2024) and Sun et al. (2023) propose agent-like studies to refine the initial answers. However, these studies neither borrow experience from agents' histories nor provide matched demonstrations. In contrast, R^2-MGA tackles these challenges.

Task Formalization

For fair comparisons, we follow the task formalization presented in ALCE (Gao et al. 2023), as shown below: Given a question $\mathcal{Q}$ and retrieved documents (*i.e.*, $\mathcal{D} = \{d_1, d_2, \cdots, d_m\}$) that contain the knowledge to answer $\mathcal{Q}$, a generation system is required to generate an answer $\mathcal{A}$ by synthesizing $d_i \in \mathcal{D}$. $\mathcal{A}$ is composed of multiple statements, *i.e.*, $\mathcal{A} = s_1 s_2 \cdots s_n$, where each s_i is a factual statement summarized from a set of relevant documents, *i.e.*, $\mathcal{D}_i = \{d_{i,1}, d_{i,2}, \cdots\}$, and s_i should cite each document included in $\mathcal{D}_i$ using the format of `[1][2][3]`.

Note that ALCE regards each sentence of $\mathcal{A}$ as a statement, which can be distinguished by the full stop symbol. Additionally, it allows for at least one and at most three citations for each statement.

Method

We introduce the R^2-MGA framework, a generative agent augmented by memory retrieval and reflection. It is expected to generate answers to given questions by synthesizing retrieved documents and properly citing them. The overview of R^2-MGA is presented in Figure 2, consisting of four modules: Memory, Initialization, Assessment, and Planning & Action, with Memory being the core module.

Memory Module

Memory Snippets. The Memory module contains a memory bank, which stores R^2-MGA's three types of histories,

[2]https://www.perplexity.ai/

[3]https://coral.cohere.com/

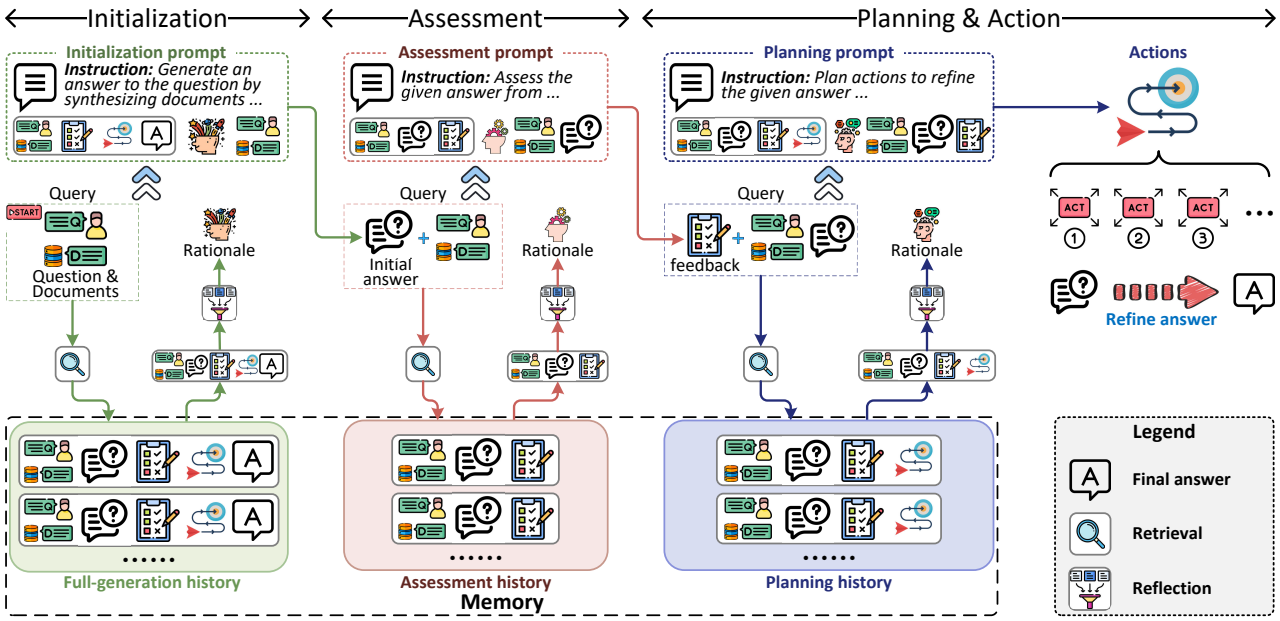

Figure 2: Overview of R^2-MGA. Given a question $\mathcal{Q}$ and retrieved documents $\mathcal{D}$, the Initialization module first generates an initial answer $\mathcal{A}$; then, the Assessment module assesses $\mathcal{A}$ from multiple aspects and returns feedback $\mathcal{F}$; finally, the Planning & Action module first plans actions $\mathcal{T}$ based on $\mathcal{F}$, and then executes concrete actions to refine $\mathcal{A}$ and outputs the final answer $\mathcal{A}^*$. In each module, we retrieve the memory and reflect the retrieved snippet to obtain the best-matched in-context demonstration.

i.e., full-generation history, assessment history, and planning history. We first illustrate three symbols in the following:

- $\mathcal{F}$, the answer assessment feedback returned by the Assessment module.
- $\mathcal{T}$, the future actions planned by the Planning & Action module.
- $\mathcal{A}^*$, the final answer obtained by refining the initial answer $\mathcal{A}$.

The full-generation history is constructed by answer generation steps. Each memory snippet can be represented by $\mathcal{M}_i = \{\mathcal{Q}, \mathcal{D}, \mathcal{A}, \mathcal{F}, \mathcal{T}, \mathcal{A}^*\}$, including all generation steps in the Initialization, Assessment, and Planning & Action modules.

The assessment history is built with the feedback generation steps, which assess the initial answer $\mathcal{A}$ from multiple aspects and return the feedback $\mathcal{F}$. Each memory snippet can be denoted as $\mathcal{M}_j = \{\mathcal{Q}, \mathcal{D}, \mathcal{A}, \mathcal{F}\}$.

The planning history is built with the action planning steps, which plan future actions based on the feedback $\mathcal{F}$. Each memory snippet can be expressed as $\mathcal{M}_k = \{\mathcal{Q}, \mathcal{D}, \mathcal{A}, \mathcal{F}, \mathcal{T}\}$.

We set a specific indicator for each type of memory in the memory bank to distinguish them in the memory bank.

Memory Retrieval and Reflection. An extensive token budget is required to inject each memory snippet into the agent's current running memory, which will be limited by LLMs' context window. Meanwhile, some memory snippets, due to the inconsistency of their question types to the to-answer question type, reduce the utility of in-context learning supervision. As such, we introduce a best-matching approach to retrieve the best-matched snippets within the memory bank to maximize the utility of in-context learning supervision.

Formally, in the Initialization, given a to-answer question Q and retrieved documents $\mathcal{D}$, we take $<Q, \mathcal{D}>$ as the query to retrieve $\mathcal{M}_i$ type memory snippets. Similarly, in the Assessment, we take $<Q, \mathcal{D}, \mathcal{A}>$ as the query to retrieve $\mathcal{M}_j$ type memory snippets; in the Planning & Action, we take $<Q, \mathcal{D}, \mathcal{A}, \mathcal{F}>$ as the query to retrieve $\mathcal{M}_k$ type memory snippets. Figure 2 also presents the query constitutions.

Following Park et al. (2023) and Deng et al. (2024b), we use the language model to generate an embedding vector of the query and each memory snippet. Then, we compute the cosine similarity in the embedding space to retrieve the best-matched memory snippet.

Memory reflection enables agents to independently summarize and infer more abstract, complex, high-level information (Wang et al. 2023b). To enrich memory snippet information, we leverage LLMs' exceptional reasoning capability to generate intermediate reasoning rationale as a supervised signal. In particular, for the retrieved memory snippet $\mathcal{M}$, we prompt OpenAI's `gpt-3.5-turbo-0125` to generate an in-depth rationale r explaining the reason for the decision-making process included in $\mathcal{M}$.

We combine the retrieved memory $\mathcal{M}$ with the reasoning rationale r as the best-matched demonstration in the Initialization, Assessment, and Planning & Action modules.

Initialization, Assessment, and Planning & Action Modules

Given $\mathcal{Q}$ and $\mathcal{D}$, the three modules generate the final answer through the following three steps.

(1) **Initialization**. This module generates an initial answer for $\mathcal{Q}$. To be specific, first retrieve the best-matched memory snippet $\mathcal{M}_i$ and obtain the reasoning rationale r_i; then, build the initialization prompt, which can be represented by $\mathcal{P}_I = \{\mathcal{I}, \mathcal{M}_i, r_i, \mathcal{Q}, \mathcal{D}\}$, where $\mathcal{I}$ is the instruction; at last, instruct LLMs to generate an initial answer $\mathcal{A}$ and pass it to the Assessment module.

(2) **Assessment**. This module assesses the quality of $\mathcal{A}$ from multiple aspects, including answer completeness, answer correctness, and citation correctness. To be precise, first retrieve the best-matched memory snippet $\mathcal{M}_j$ and obtain the reasoning rationale r_j; then, build the assessment prompt, which can be represented by $\mathcal{P}_A = \{\mathcal{I}, \mathcal{M}_j, r_j, \mathcal{A}\}$; at last, instruct LLMs to generate the feedback (*i.e.*, $\mathcal{F}$) of $\mathcal{A}$ and pass it to the Planning & Action module.

(3) **Planning & Action**. This module first plans answer refinement actions based on $\mathcal{F}$, and then executes concrete actions to refine $\mathcal{A}$. In particular, first retrieve the best-matched memory snippet $\mathcal{M}_k$ and obtain the reasoning rationale r_k; then, build the planning prompt, which can be represented by $\mathcal{P}_P = \{\mathcal{I}, \mathcal{M}_k, r_k, \mathcal{Q}, \mathcal{D}, \mathcal{A}, \mathcal{F}\}$; next, instruct LLMs to generate future answer refinement actions (*i.e.*, $\mathcal{T}$); at last, instruct LLMs to refine $\mathcal{A}$ based on $\mathcal{T}$, obtaining the final answer $\mathcal{A}^*$. There are two paradigms to execute actions: 1) one-time execution and 2) step-by-step execution.

We update the memory bank according to the above three new experiences. To avoid updating the memory with bad experiences, we first prompt an LLM to re-assess $\mathcal{A}^*$, and then update the memory if and only if $\mathcal{A}^*$ is assessed to be correct and have proper citations.

Discussions

Sun et al. (2023) propose a framework (VTG) to first generate an initial answer, and then verify and simplify the citations. Madaan et al. (2024) and Lee et al. (2024) propose two similar frameworks (SELF-REFINE and A^{2}R), which prompt LLMs to generate, assess, and refine their outputs. We demonstrate that our R^2-MGA significantly differs from the above studies. In particular, R^2-MGA manages a memory bank, enabling R^2-MGA to borrow its successful experiences by retrieving the best-matched in-context demonstration for new generation cases, which ensures consistent agent behaviours and improves the generation quality. In contrast, the above studies don't retain such a memory bank and always use the same demonstrations across all cases.

We claim that the best-matched demonstration can maximize the utility of in-context learning supervision. For example, given a to-answer question "*Who has the highest goals all-time in men's football?*", the demonstration regarding "*Who has the highest goals in men's world international football?*" is more informative than the demonstration regarding "*When was the first apple i phone made?*", since the former is more matched with the to-answer question than the latter one.

Research on retrieving high-quality in-context demonstrations has been widely explored. For example, Wang, Yang, and Wei (2024) train dense retrievers to identify suitable in-context examples. Shum, Diao, and Zhang (2023) propose a variance-reduced policy gradient strategy to measure the importance of candidate in-context examples. Yu, He, and Ying (2024) propose analogical prompting, which instructs LLMs to self-generate relevant in-context examples. Different from these sophisticated approaches, we first compute the cosine similarity between the query and memory snippets to retrieve the best-matched memory snippet, then combine the snippet with a reasoning rationale summarized from the snippet as the best-matched in-context demonstration. We conduct investigations in the Analyses section.

Experiments

Benchmark, LLMs, and Implementation Details

The ALCE benchmark collects three datasets, *i.e.*, ASQA (Stelmakh et al. 2022), QAMPARI (Rubin et al. 2022), and ELI5 (Fan et al. 2019) and pre-defines automatic evaluation metrics. In particular, it uses Correctness (Correct.), Citation Recall (Rec.) and Precision (Prec.) to evaluate ASQA and ELI5, and uses Correctness Recall-5 (Rec.-5) and Precision (Prec.), Citation Recall and Precision to evaluate QAMPARI. We report more benchmark details in Appendix A.

We build R^2-MGA upon five LLMs including closed-source ChatGPT (`gpt-3.5-turbo-0301`) and GPT-4 (`gpt-4-0613`), and open-source LLaMA-2-70B-Chat ("LLaMA-70B" for short), Vicuna-13B, and LLaMA-2-7B-Chat ("LLaMA-7B" for short).

We use four NVIDIA A100 40GB GPUs to run R^2-MGA. For R^2-MGA built upon open-source LLMs, we evaluate it by setting the LLMs' temperature to 0.001, 0.1, 0.3, 0.5, 0.7, 0.9, and 1, respectively, and report the averaged performance. Limited by the API costs of ChatGPT and GPT-4, we solely set the temperature value to 1 for them. We report more implementation details in Appendix B.

Baselines

For fair comparisons, we use the one-step generation approaches proposed by Gao et al. (2023) and Ji et al. (2024) as the baselines, which are denoted as ALCE-base and ALCE-CoT, respectively. In particular, the ALCE-base is the benchmark baseline included in ALCE; and the ALCE-CoT integrates the Chain-of-Thought into the ALCE-base, empowering LLMs to mimic the human thought process. We report more baseline details in Appendix C.

There exist some other studies, including fine-tuning task-specific LLMs (Huang et al. 2024; Lee et al. 2024) and one-step generation with answer refinement (Sun et al. 2023). We demonstrate that these studies adopt quite different LLMs and experimental settings. Despite these differences, we make attempts to compare our R^2-MGA with them fairly. Appendix D reports the comparison results.

Main Results

We use the one-time paradigm to execute the answer refinement actions and report performance comparisons between our R^2-MGA and the baselines in Table 1. We have the following observations:

Strategy	Approach	ASQA			QAMPARI				ELI5		
		Correct.	Citation		Correct.		Citation		Correct.	Citation	
			Rec.	Prec.	Rec.-5	Prec.	Rec.	Prec.		Rec.	Prec.
GPT-4											
VANILLA	ALCE-base	41.3	68.5	75.6	22.2	25.0	25.9	27.0	14.2	44.0	50.1
	R^2-MGA (Ours)	**42.6**	**84.7**	**85.2**	**25.3**	**27.6**	**32.4**	**33.9**	**18.2**	**69.8**	**70.5**
ChatGPT											
VANILLA	ALCE-base	40.4	73.6	72.5	20.8	20.8	20.5	20.9	12.0	51.1	50.0
	R^2-MGA (Ours)	**41.2**	**84.4**	**84.3**	**24.2**	**23.0**	**29.8**	**28.1**	**16.4**	**74.5**	**75.1**
SUMM	ALCE-base	43.3	68.8	61.8	23.6	21.2	23.6	25.7	12.5	51.5	48.2
	R^2-MGA (Ours)	**44.6**	**81.4**	**80.2**	**25.2**	**24.1**	**32.9**	**33.4**	**16.9**	**72.4**	**74.3**
SNIPPET	ALCE-base	41.4	65.3	57.4	24.5	21.5	22.9	24.9	14.3	50.4	45.0
	R^2-MGA (Ours)	**43.0**	**77.4**	**75.1**	**25.1**	**24.0**	**34.1**	**33.6**	**15.8**	**73.2**	**73.8**
ORACLE	ALCE-base	48.9	74.5	72.7	37.0	36.9	24.1	24.6	21.3	57.8	56.0
	R^2-MGA (Ours)	**51.1**	**89.6**	**86.9**	**42.1**	**41.4**	**34.4**	**34.7**	**27.2**	**77.2**	**79.2**
LLaMA-70B											
VANILLA	ALCE-base	41.5	62.9	61.3	21.8	18.4	15.1	15.6	12.8	38.3	37.9
	ALCE-CoT	43.9	70.2	69.8	23.2	21.7	22.3	24.0	13.9	51.1	49.1
	R^2-MGA (Ours)	**44.7**	**77.6**	**73.9**	**23.7**	**23.2**	**24.2**	**25.8**	**18.1**	**57.4**	**57.7**
SUMM	ALCE-CoT	44.6	71.4	68.0	23.6	22.0	23.1	23.7	12.7	49.7	48.4
	R^2-MGA (Ours)	**45.1**	**79.0**	**74.2**	**24.2**	**23.7**	**23.9**	**25.8**	**17.4**	**56.8**	**57.2**
SNIPPET	ALCE-CoT	42.9	70.4	69.4	22.9	23.1	22.3	23.8	13.9	49.6	45.2
	R^2-MGA (Ours)	**43.4**	**79.3**	**75.6**	**24.9**	**24.2**	**23.8**	**25.2**	**18.6**	**56.6**	**55.4**
ORACLE	ALCE-CoT	46.2	73.5	72.9	36.1	36.7	24.4	24.4	20.8	56.7	55.3
	R^2-MGA (Ours)	**50.0**	**87.5**	**83.8**	**40.8**	**41.1**	**33.7**	**32.0**	**26.8**	**76.4**	**78.2**
Vicuna-13B											
VANILLA	ALCE-base	31.9	51.1	50.1	14.0	15.9	12.5	13.4	10.0	15.6	19.6
	ALCE-CoT	36.4	56.6	56.4	18.1	17.4	11.6	15.4	13.3	21.3	24.1
	R^2-MGA (Ours)	**37.1**	**73.5**	**67.9**	**20.1**	**19.8**	**23.9**	**23.4**	**14.3**	**38.9**	**40.4**
SUMM	ALCE-base	43.2	52.7	50.0	21.1	17.1	15.7	17.8	4.9	9.7	12.2
	ALCE-CoT	43.2	56.9	56.8	25.6	19.8	17.2	19.3	10.2	11.7	15.8
	R^2-MGA (Ours)	**43.7**	**71.2**	**69.3**	**25.8**	**20.6**	**27.4**	**26.9**	**16.2**	**28.0**	**28.7**
SNIPPET	ALCE-base	42.1	53.4	48.7	21.9	18.2	16.8	19.7	11.2	27.2	27.9
	ALCE-CoT	42.9	55.6	52.3	25.6	21.3	19.4	24.4	13.6	29.3	33.6
	R^2-MGA (Ours)	**43.1**	**70.1**	**68.2**	**27.0**	**23.4**	**26.4**	**27.3**	**19.7**	**38.9**	**40.4**
ORACLE	ALCE-base	42.5	52.2	50.7	25.9	28.4	15.8	16.8	17.1	20.2	26.5
	ALCE-CoT	42.7	56.2	54.8	27.8	31.2	20.2	19.7	22.1	24.3	31.1
	R^2-MGA (Ours)	**44.0**	**74.1**	**69.4**	**30.8**	**33.9**	**25.5**	**26.5**	**24.7**	**42.7**	**43.6**
LLaMA-7B											
VANILLA	ALCE-base	33.9	50.9	47.5	16.2	15.3	10.6	10.9	10.9	19.8	15.0
	R^2-MGA (Ours)	**39.6**	**71.8**	**66.4**	**21.3**	**20.0**	**21.3**	**21.7**	**15.2**	**37.4**	**38.2**

Table 1: Performance comparisons between our R^2-MGA and one-step generation baselines (ALCE-base and ALCE-CoT) on the ALCE datasets. Bolded values denote the best performance scores under each <Strategy, LLM> setting.

(1) From the holistic perspective, R^2-MGA consistently improves the answer correctness and citation quality by large margins.

(2) On ASQA, R^2-MGA delivers up to +16.8% (VANILLA, LLaMA-7B), +41.1% (VANILLA, LLaMA-7B), and +39.8% (VANILLA, LLaMA-7B) relative gains on Correctness, Citation Recall and Precision, respectively. On average, it brings +3.7%, +21.6%, and +20.5% relative gains on the three metrics.

(3) On QAMPARI, R^2-MGA delivers up to +31.5% (VANILLA, LLaMA-7B), +30.7% (VANILLA, LLaMA-7B), +106.0% (VANILLA, Vicuna-13B), and +99.1% (VANILLA, LLaMA-7B) relative gains on Correctness Recall-5 and Precision, Citation Recall and Precision, respectively. On average, it brings +9.6%, +11.2%, 41.9%, and +32.8% relative gains on the four metrics.

(4) On ELI5, R^2-MGA delivers up to +58.8% (SUMM, Vicuna-13B), +139.3% (SUMM, Vicuna-13B), and +154.7% (VANILLA, LLaMA-7B) relative gains on Correctness, Citation Recall and Precision, respectively. On average, it brings +30.8%, +51.3%, and +51.0% relative gains on the three metrics.

We attribute these performance gains in answer correctness and citation quality to the memory-augmented retrieval and reflection strategy and the answer refinement strategy, where the former provides the best-matched demonstration to facilitate in-context learning supervision, and the latter further improves the answer quality.

R^2-MGA built upon LLaMA-70B consistently outperforms the ChatGPT-powered ALCE-base across the three benchmark datasets and the four prompting strategies, and it even outperforms the GPT-4-based baseline in 6 out of 10 evaluation cases under the VANILLA setting, indicating R^2-MGA as a strong baseline to facilitate future study.

Take inspiration from SELF-REFINE (Madaan et al. 2024), we investigate iteratively executing Assessment and

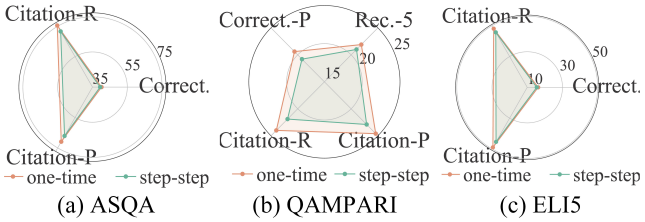

Figure 3: Comparisons between the one-time and step-by-step action executions.

Planning & Action to continually refine the answer. More investigation details can be found in Appendix E.

Analyses

Investigation on Action Executions

We pre-define the one-time and step-by-step action execution paradigms, and use the one-time paradigm to report the main experimental results. In this section, we compare the one-time paradigm with the step-by-step paradigm under the <VANILLA, Vicuna-13B> setting. Figure 3 presents the comparisons, from which we can observe that the one-time paradigm consistently outperforms the step-by-step one across all datasets and all metrics. A convincing reason is that the step-by-step paradigm may cause new answer flaws that can be solved in previous actions but cannot in subsequent actions. In contrast, the one-time paradigm refines the answers holistically, avoiding the above problems.

Motivated by this investigation, we use the one-time paradigm in the other experiments.

Investigation on Memory Bank

The memory bank records R^2-MGA's experiences (*i.e.,* memory snippets) and helps to obtain the best-matched demonstration for in-context learning supervision. The memory snippets accumulate along with R^2-MGA consistently answering questions. Theoretically, the more snippets there are, the better the best-matched in-context demonstration is since more snippets provide more retrieval cases.

We conduct detailed analyses to explore this assumption. In particular, we first split the 1,000 dataset entries into five buckets, *i.e.*, 1-200, 201-400, 401-600, 601-800, and 801-1000, where R^2-MGA processes them from 1 to 1000 and consistently accumulates memory snippets;[4] then, we calculate the evaluation metrics of each bucket. In particular, we consider all LLMs for each dataset and reported the average results, as presented in Figure 4. We observe that the answer quality consistently improves along with the memory snippets increase, validating the goal of the memory design and our assumption. In addition, we conduct investigations on memory snippets and report the results in Appendix F.

Investigation on the Best-matched Demonstration

In this section, we investigate the effectiveness of the best-matched in-context demonstration under the <VANILLA, Vicuna-13B, ASQA> setting. In particular, we conduct another four groups of experiments, which replace the best-matched demonstration with the second-, the third-, the fourth-, and the fifth-matched demonstration, respectively. For scenarios in which the memory doesn't have enough memory snippets, we use the last-matched memory snippet as an alternative. For example, if the memory solely has three snippets, we use the third-matched snippet to construct the fourth- and the fifth-matched in-context demonstration.

Figure 5 shows the investigation results. We observe that: (1) the best-matched in-context demonstration consistently outperforms the other matched demonstrations across all three datasets; (2) the performance scores generally drop as the demonstration's matched-degree decreases, *i.e.*, from the best-matched to the fifth-matched. These results verify our motivation for the best-matched in-context demonstration.

[4]Each of ASQA, QAMPARI, and ELI5 has 1,000 data entries.

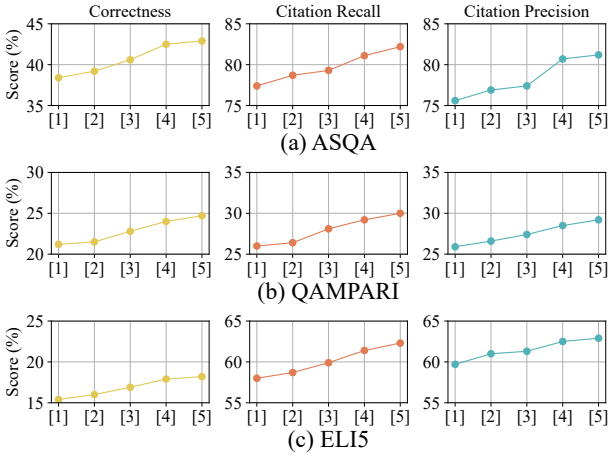

Figure 4: Changes of answer quality along with memory snippets increasing. We use '[1]', '[2]', '[3]', '[4]', and '[5]' to represent the five buckets for short. For QAMPARI, we average the Correctness Recall-5 and Precision as the overall Correctness score.

Investigation on Computation Costs

Compared to one-step generation baselines, R^2-MGA needs additional LLMs' operations in answer assessment, action

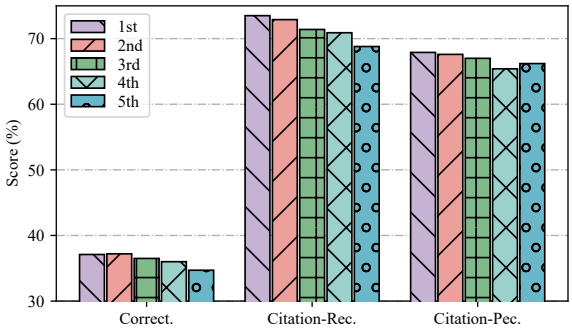

Figure 5: Comparisons of different best-matched in-context demonstrations.

	ASQA			QAMPARI				ELI5		
	Correct.	Citation		Correct.		Citation		Correct.	Citation	
		Rec.	Prec.	Rec.-5	Prec.	Rec.	Prec.		Rec.	Prec.
R^2-MGA	37.1	73.5	67.9	20.1	19.8	23.9	23.4	14.3	38.9	40.4
-best-matched demo	36.4 (-0.7)	68.7 (-4.8)	63.1 (-4.8)	18.9 (-1.2)	19.1 (-0.7)	17.6 (-6.3)	18.2 (-5.2)	13.5 (-0.8)	28.2 (-10.7)	29.1 (-11.3)
-answer refinement	34.2 (-2.9)	58.4 (-5.1)	58.6 (-9.3)	16.3 (-3.8)	16.6 (-3.2)	15.4 (-8.5)	14.9 (-8.5)	11.7 (-2.6)	22.3 (-16.6)	25.1 (-15.3)
-rationale	36.7 (-0.4)	70.6 (-2.9)	65.2 (-2.7)	18.5 (-1.6)	18.2 (-1.6)	21.0 (-2.9)	22.6 (-0.8)	13.6 (-0.7)	35.9 (-3.0)	37.7 (-2.7)

Table 2: Results of ablation studies. We obtain the results using the <VANILLA, Vicuna-13B> setting.

Approach	LLM	ASQA	QAMPARI	ELI5
ALCE-base	LLaMA-70B	1	1	1
w/RERANK		4	4	4
R^2-MGA		2.42	2.87	2.56
ALCE-base	Vicuna-13B	1	1	1
w/RERANK		4	4	4
R^2-MGA		2.66	3.04	2.89
ALCE-base	LLaMA-7B	1	1	1
w/RERANK		4	4	4
R^2-MGA		2.72	2.90	2.65

Table 3: Comparisons of computation costs between R^2-MGA and ALCE-base. For each <LLM, Dataset> setting, we use the counts of LLM's input and output tokens across all dataset entries as the computation cost. For simplicity, we always take ALCE-base's computation cost as the unit and the scores calculated by comparing the costs of R^2-MGA to ALCE-base as R^2-MGA's costs.

planning, action executing, and rationale generating, which call for computation resources. To reduce the costs, R^2-MGA adopts a 1-shot in-context demonstration when generating the initial answer $\mathcal{A}$ and other generations, which dramatically reduces LLMs' input tokens compared to the 2-shot in-context demonstration used in one-step generation baselines like ALCE-base. Additionally, we store the generated rationales for memory snippets to avoid repeated generations, which also reduces the costs.

We compare the computation costs between R^2-MGA and ALCE-base with the three open-source LLMs and under the VANILLA setting. Additionally, we also consider the RERANK prompting strategy of ALCE-base, which instructs LLMs to repeatedly generate four answers for each question and select the best answer using the citation recall metric. Table 3 reports the comparison results. We can observe that (1) compared to ALCE-base, R^2-MGA consumes additional 1.42× to 2.04× computation resources, but it delivers significant performance gains (See Table 1); (2) compared to ALCE-base w/RERANK, R^2-MGA consumes much less computation resources. It's worth noting that R^2-MGA consistently outperforms ALCE-base w/RERANK. Appendix G reports detailed comparison results.

Ablation Study

We conduct ablation studies on R^2-MGA across all three datasets under the <VANILLA, Vicuna-13B> setting. Specifically, we investigate the effectiveness of the best-matched in-context demonstration, the answer refinement strategy, and the rationale by ablating them individually. Table 2 reports the ablation results, where "-best-matched demo" denotes always using the same demonstration for the in-context learning supervision; "-answer refinement" denotes removing the Assessment and the Planning & Action modules; "-rationale" denotes not providing reasoning rationales in the best-matched in-context demonstration. We have the following observations:

(1) "-best-matched demo" decreases the Correctness scores ranging from -0.7% to -1.2%, and the Citation Recall scores ranging from -4.8% to -10.7%, and the Citation Precision scores ranging from -4.8% to -11.3%. These performance drops indicate that the best-matched demonstration is more informative and provides better in-context learning supervision.

(2) "-answer refinement" dramatically decreases the Correctness scores ranging from -2.6% to -3.8%, and the Citation Recall scores ranging from -5.1% to -16.6%, and the Citation Precision scores ranging from -8.5% to -15.3%. These decreased scores prove the claim that LLMs do not always generate the best output on their first try (Madaan et al. 2024). They also strongly support our motivation for further answer refinement.

(3) "-rationale" showcases obvious negative impacts on the three metrics. Specifically, it brings performance drops on the Correctness scores ranging from -0.4% to -1.6%, the Citation Recall scores ranging from -2.9% to -3.0%, and the Citation Precision scores ranging from -0.8% to -2.7%. These results firmly support our motivation to use reasoning rationales in in-context demonstrations, enhancing the in-context learning supervision.

Conclusion

In this paper, we present a novel framework named Retrieval and Reflection Memory-augmented Generative Agent (R^2-MGA), which leverages the memory retrieval and reflection techniques to provide the best-matched in-context demonstration, as well as an answer refinement strategy to first assess answers and further refine them accordingly. We design a rational memory architecture and a succinct but reliable memory updating mechanism to maintain this core module. Experimental results on the ALCE benchmark indicate R^2-MGA dramatically boosting LLMs' capabilities in text generation with citations, outperforming those one-step generation baselines by large margins. Extensive analyses firmly verify the various motivations of R^2-MGA.

Acknowledgments

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002).

References

Cho, J.; Yoon, J.; and Ahn, S. 2024. Spatially-Aware Transformers for Embodied Agents. In *The Twelfth International Conference on Learning Representations*.

Choi, J.-W.; Yoon, Y.; Ong, H.; Kim, J.; and Jang, M. 2024. LoTa-Bench: Benchmarking Language-oriented Task Planners for Embodied Agents. In *The Twelfth International Conference on Learning Representations*.

Deng, Y.; Liao, L.; Zheng, Z.; Yang, G. H.; and Chua, T.-S. 2024a. Towards Human-centered Proactive Conversational Agents. arXiv:2404.12670.

Deng, Y.; Zhang, X.; Zhang, W.; Yuan, Y.; Ng, S.-K.; and Chua, T.-S. 2024b. On the Multi-turn Instruction Following for Conversational Web Agents. arXiv:2402.15057.

Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; and Auli, M. 2019. ELI5: Long Form Question Answering. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3558–3567. Florence, Italy: Association for Computational Linguistics.

Funkquist, M.; Kuznetsov, I.; Hou, Y.; and Gurevych, I. 2023. CiteBench: A Benchmark for Scientific Citation Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7337–7353.

Gao, T.; Yen, H.; Yu, J.; and Chen, D. 2023. Enabling Large Language Models to Generate Text with Citations. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 6465–6488. Singapore: Association for Computational Linguistics.

Gou, Z.; Shao, Z.; Gong, Y.; yelong shen; Yang, Y.; Huang, M.; Duan, N.; and Chen, W. 2024. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. In *The Twelfth International Conference on Learning Representations*.

Huang, C.; Wu, Z.; Hu, Y.; and Wang, W. 2024. Training Language Models to Generate Text with Citations via Fine-grained Rewards. *arXiv preprint arXiv:2402.04315*.

Ji, B.; Liu, H.; Du, M.; and Ng, S.-K. 2024. Chain-of-Thought Improves Text Generation with Citations in Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18345–18353.

Lee, D.; Park, E.; Lee, H.; and Lim, H.-S. 2024. Ask, Assess, and Refine: Rectifying Factual Consistency and Hallucination in LLMs with Metric-Guided Feedback Learning. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2422–2433.

Li, X.; Jin, J.; Zhou, Y.; Zhang, Y.; Zhang, P.; Zhu, Y.; and Dou, Z. 2024. From Matching to Generation: A Survey on Generative Information Retrieval. *arXiv preprint arXiv:2404.14851*.

Li, X.; Zhu, C.; Li, L.; Yin, Z.; Sun, T.; and Qiu, X. 2023. Llatrieval: Llm-verified retrieval for verifiable generation. *arXiv preprint arXiv:2311.07838*.

Liu, N.; Zhang, T.; and Liang, P. 2023. Evaluating Verifiability in Generative Search Engines. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 7001–7025. Singapore: Association for Computational Linguistics.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.

Qiao, S.; Zhang, N.; Fang, R.; Luo, Y.; Zhou, W.; Jiang, Y. E.; Lv, C.; and Chen, H. 2024. Autoact: Automatic agent learning from scratch via self-planning. *arXiv preprint arXiv:2401.05268*.

Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; Zhao, S.; Hong, L.; Tian, R.; Xie, R.; Zhou, J.; Gerstein, M.; dahai li; Liu, Z.; and Sun, M. 2024. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In *The Twelfth International Conference on Learning Representations*.

Rubin, S. J. A. O.; Yoran, O.; Wolfson, T.; Herzig, J.; and Berant, J. 2022. QAMPARI:: An Open-domain Question Answering Benchmark for Questions with Many Answers from Multiple Paragraphs. *arXiv preprint arXiv:2205.12665*.

Shum, K.; Diao, S.; and Zhang, T. 2023. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 12113–12139.

Stelmakh, I.; Luan, Y.; Dhingra, B.; and Chang, M.-W. 2022. ASQA: Factoid Questions Meet Long-Form Answers. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 8273–8288. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Sun, H.; Cai, H.; Wang, B.; Hou, Y.; Wei, X.; Wang, S.; Zhang, Y.; and Yin, D. 2023. Towards verifiable text generation with evolving memory and self-reflection. *arXiv preprint arXiv:2312.09075*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023a. Voyager: An Open-Ended Embodied Agent with Large Language Models. In *Intrinsically-Motivated and Open-Ended Learning Workshop@ NeurIPS2023*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2023b. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wang, L.; Yang, N.; and Wei, F. 2024. Learning to Retrieve In-Context Examples for Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1752–1767.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E. H.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*.

Wei, Z.; Chen, W.-L.; and Meng, Y. 2024. InstructRAG: Instructing Retrieval-Augmented Generation with Explicit Denoising. *arXiv preprint arXiv:2406.13629*.

Xia, S.; Wang, X.; Liang, J.; Zhang, Y.; Zhou, W.; Deng, J.; Yu, F.; and Xiao, Y. 2024. Ground Every Sentence: Improving Retrieval-Augmented LLMs with Interleaved Reference-Claim Generation. *arXiv preprint arXiv:2407.01796*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Ye, X.; Sun, R.; Arik, S. Ö.; and Pfister, T. 2024. Effective large language model adaptation for improved grounding.

Yu, J.; He, R.; and Ying, Z. 2024. Thought Propagation: An Analogical Approach to Complex Reasoning with Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Zhang, H.; Du, W.; Shan, J.; Zhou, Q.; Du, Y.; Tenenbaum, J. B.; Shu, T.; and Gan, C. 2024. Building Cooperative Embodied Agents Modularly with Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Zheng, S.; jiazheng liu; Feng, Y.; and Lu, Z. 2024. Steve-Eye: Equipping LLM-based Embodied Agents with Visual Perception in Open Worlds. In *The Twelfth International Conference on Learning Representations*.