Improve Rule Retrieval and Reasoning with Self-Induction and Relevance ReEstimate

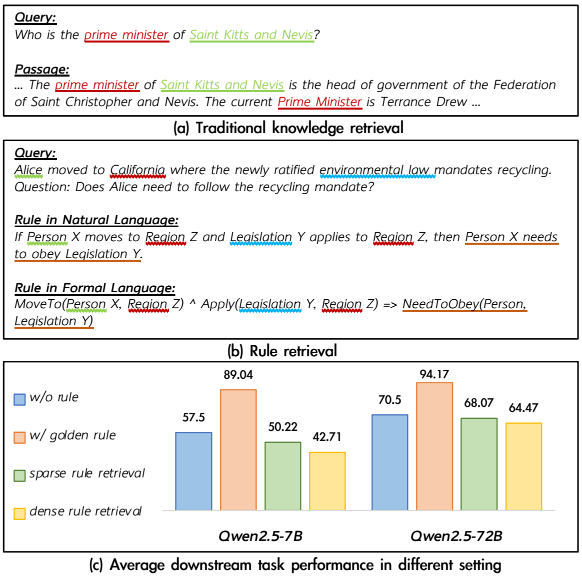
Ziyang Huang, Wangtao Sun, Jun Zhao, Kang Liu[*]
The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
huangziyang2023@ia.ac.cn, {jzhao, kliu}@nlpr.ia.ac.cn

Abstract

This paper systematically addresses the challenges of rule retrieval, a crucial yet underexplored area. Vanilla retrieval methods using sparse or dense retrievers to directly search for relevant rules to support downstream reasoning, often suffer from low accuracy. This is primarily due to a significant semantic gap between the instantiated facts in the queries and the abstract representations of the rules. Such misalignment results in suboptimal retrieval quality, which in turn negatively impacts reasoning performance. To overcome these challenges, we propose **Self-Induction Augmented Retrieval (SIAR)**, a novel approach that utilizes Large Language Models (LLMs) to induce potential inferential rules that might offer benefits for reasoning by abstracting the underlying knowledge and logical structure in queries. These induced rules are then used for query augmentation to improve retrieval effectiveness. Additionally, we introduce **Rule Relevance ReEstimate (R^3)**, a method that re-estimates the relevance of retrieved rules by assessing whether the abstract knowledge they contain can be instantiated to align with the facts in the queries and the helpfulness for reasoning. Extensive experiments across various settings demonstrate the effectiveness and versatility of our proposed methods.

1 Introduction

With the advancement of pre-training (Zhou et al., 2023) and prompting techniques (Schulhoff et al., 2024; Dong et al., 2024), Large Language Models (LLMs) (Zhao et al., 2024b; Dubey et al., 2024; Yang et al., 2024a; Abdin et al., 2024) have made significant progress in their understanding, reasoning, and decision-making capabilities (Wang et al., 2024a; Huang et al., 2025a). Rules can generate new knowledge from existing information (or make decisions based on observed situations), which can enhance these abilities further (Zhu et al., 2024; Wang et al., 2024b).



Query:
Who is the prime minister of Saint Kitts and Nevis?

Passage:
... The prime minister of Saint Kitts and Nevis is the head of government of the Federation of Saint Christopher and Nevis. The current Prime Minister is Terrance Drew ...

(a) Traditional knowledge retrieval

Query:
Alice moved to California where the newly ratified environmental law mandates recycling. Question: Does Alice need to follow the recycling mandate?

Rule in Natural Language:
If Person X moves to Region Z and Legislation Y applies to Region Z, then Person X needs to obey Legislation Y.

Rule in Formal Language:
MoveTo(Person X, Region Z) ^ Apply(Legislation Y, Region Z) => NeedToObey(Person, Legislation Y)

(b) Rule retrieval

(c) Average downstream task performance in different setting

Figure 1: (a) and (b) show the different characteristics of traditional knowledge retrieval and rule retrieval. (c) illustrates that the golden rule can significantly improve reasoning performance, while existing rule retrieval methods typically lead to a decline in reasoning performance due to suboptimal recall.

The de facto approach of rule-based reasoning typically involves summarizing generalized rules from past experiences by LLMs, then retrieving the relevant rules based on the descriptions of downstream tasks or feedback from the observed environment, and finally using the retrieved rules to assist in reasoning or decision-making (Yang et al., 2023; Sun et al., 2023a; Zhang et al., 2024; Zhao et al., 2024a; Huang et al., 2025b). Unfortunately, existing research has primarily focused on rule generation (Sivasothy et al., 2024; Wang et al., 2024c) and application (Wang et al., 2024d), neglecting the development of the rule retrieval techniques. Furthermore, rule retrieval plays a crucial role in real-world scenarios. For example, in legal scenar-

[*] Kang is the corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 5473–5488
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ios, one must retrieve the relevant laws based on the crime to make a judgment (Xiao et al., 2018), and in medical settings, domain-specific rules must be retrieved based on symptoms to assist in diagnosis (Wang et al., 2024e). The above highlights the urgency of exploring rule retrieval area.

The rules discussed in this paper are referential rules (Sun et al., 2024b; Wang et al., 2024c), which typically manifest as the derivation of one set of facts from another. In natural language, they are usually expressed in the form "if `Premise`, then `Conclusion`," whereas in formal language, they are often represented as "`Premise` ⇒ `Conclusion`." As illustrated in Figure 1 (a), the query and the corresponding golden passage typically share some keywords or similar semantics in traditional knowledge retrieval scenarios (Bajaj et al., 2016; Karpukhin et al., 2020), making direct matching between the query and passage feasible. However, in rule retrieval scenarios, the following characteristics of the query and golden rule present significant challenges: (1) The facts in the query are instantiated and specific. (2) The facts contained in the rule are composed of variables and predicates, where the variables typically have an abstract, conceptual type, and the predicates represent relationships between different variables, which might not be expressed in the query explicitly. (3) The gap between the query (concrete, implicit) and the rule (abstract, explicit) leads to a semantic misalignment. (4) There is an explicit derivation in the rule, but not in the query. For example, as shown in Figure 1 (b), "environmental law in California mandates recycling" is the fact in the query. In the golden rule, the entity "California" corresponds to "Region Z", "environmental law" corresponds to "Legislation Y", and this fact implies "Legislation Y applies to Region Z". Furthermore, the rule incorporates the inferred conclusion "Person X needs to obey Legislation Y".

Existing methods typically overlook the aforementioned characteristics and apply traditional retrieval techniques directly during the rule retrieval phase. As demonstrated in Figure 1 (c), whether using sparse retrieval or dense retrieval, relying on vanilla retrieval to assist in reasoning often results in varying degrees of performance degradation compared to reasoning without rules. This demonstrates the inadequacy of traditional retrieval methods in rule-based scenarios. In fact, if the retrieved rules are irrelevant or contain noise, the reasoning will be distracted. As the size of the rule base increases, rule retrieval will become the bottleneck for downstream task performance. In contrast, when the golden rule is directly provided to aid the LLM in its reasoning process, performance enhancement of 31.54% and 23.67% can be witnessed in the 7B and 72B models respectively. The performance gap highlights the importance of rules in supporting reasoning and the necessity of accurate rule retrieval.

To this end, this paper proposes **Self-Induction Augmented Retrieval (SIAR)**. SIAR leverages self-induction to summarize and abstract the facts presented in the query and hypothesize potential inferential relationships to generate a potential rule. This newly generated rule is then used as the new query, or combined with the original query to form a new query for retrieval. Specifically, we utilize few-shot in-context learning to prompt the LLM to produce a self-induced rule. Our theoretical insight is as follows: if we consider the query set and the rule set as belonging to different semantic subspaces, where the former is characterized by instantiated, concrete facts and the latter by abstract, conceptual knowledge. We hypothesize that these two subspaces are nearly non-overlapping. The role of self-induction is to project the query as much as possible into the rule subspace, enabling the query to better match rules that share similar underlying logic during retrieval.

Although SIAR can improve the ranking of the golden label in the retrieved rule list, the limited inductive capabilities of LLMs still make it challenging to handle more difficult queries. Moreover, the retriever can only evaluate the semantic similarity instead of the helpfulness of the rule for the query. Therefore, building on SIAR, we propose **Rule Relevance ReEstimate ($\mathbf{R^3}$)**, which utilizes the LLMs to estimate the relevance of the retrieved rule list. R^3 evaluates whether each rule can be applied to the current query for better reasoning and reranks the list based on the relevance estimation.

We conduct experiments on two synthetic datasets as well as one real-world dataset. Compared to direct retrieval, SIAR achieves significant improvements in both retrieval and reasoning performance, demonstrating its effectiveness in extracting and summarizing the knowledge and logic embedded in queries to assist retrieval. Building on SIAR, R^3 further enhances both retrieval and reasoning performance, proving that LLMs can reliably assess the relevance between queries and rules, thereby improving the quality of rule retrieval. Moreover, SIAR and R^3 consistently improve performance across different settings, including varying rule formats (natural and formal language), different types of retrievers (sparse and dense retrievers), and LLMs of different parameter scales. This demonstrates the generalizability of our proposed methods.

The contributions of this paper are as follows:

- We systematically introduce the problem of rule retrieval and provide a detailed analysis of the semantic misalignment challenges faced in rule retrieval.
- We propose SIAR and R^3 to address the issues in rule retrieval: SIAR induces and abstracts the knowledge and logic embedded in the query to map it into the rule space for more effective retrieval, while R^3 enhances retrieval quality by assessing the relevance of retrieved rules to the original query.
- Extensive experiments demonstrate that SIAR and R^3 achieve performance improvements across various datasets and settings. Furthermore analysis offers more insights for future research on rule retrieval.

2 Preliminary

In the rule reasoning scenario, we have a rule library $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{R}|}$, which incorporates inferential rules offering benefits for new knowledge induction and decision-making. We use the query q to retrieve relevant rules from this library, and the retrieved rules are concatenated with the query as context. This combined input is then fed into the LLM to perform reasoning. We name this workflow as `retrieve-then-reason`.

In the retrieval phase, we employ either sparse retrieval or dense retrieval. The former uses BM25 to compute the similarity between the query and the rules, while the latter leverages a pre-trained encoder to map both the query and the rules into a shared vector space, where their similarity is computed using the cosine function. The top-k rules are then returned based on the similarity ranking. To improve retrieval speed, we pre-build and cache the index of $\mathcal{R}$.

3 Method

In the aforementioned `retrieve-then-reason` paradigm, our method primarily focuses on improving retrieval quality by aligning the semantics between the query and the rule and re-assessing the relevance of retrieved rules. The former technique is inserted before the retrieval stage and the latter one is inserted between retrieval and reasoning. Furthermore, we aim to positively impact the overall reasoning performance. Our method is based on prompting the LLMs without any training, which is versatile and cost-efficient.

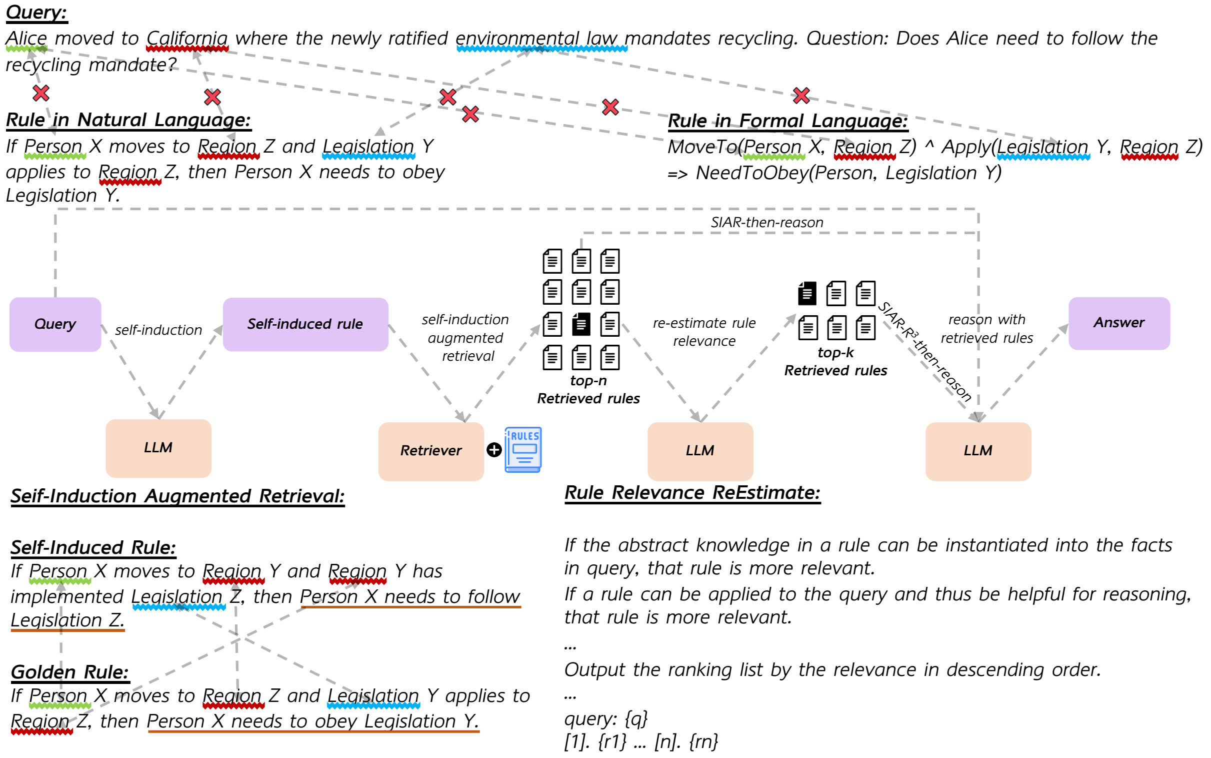
3.1 SIAR: Self-Induction Augmented Retrieval

As illustrated in Figure 2, before performing rule retrieval, we employ a self-induction process where the LLM generates a potentially useful rule to aid reasoning. This process highly relies on the inductive capability of LLM (Wang et al., 2024b; Zhu et al., 2024; Bowen et al., 2024; Cheng et al., 2024). We refer to this generated rule as the self-induced rule (SI). The key to self-induction lies in summarizing and abstracting the facts embedded in the query and hypothesizing potential inferential relationships. Due to the different characteristics of query and rule, the primary target of self-induction is to project the query to rule space. Specifically, we utilize few-shot prompting to guide the LLM for self-induction, with the corresponding instruction template shown in Appendix A. We show one self-induction example in Figure 2.

After generating the SI, we have two options for utilizing it in the retrieval process: we can either treat the SI as the new search query, or we can concatenate the SI with the original query to form a new combined search query. The former approach is referred to as SIAR (w/ SI), while the latter is referred to as SIAR (w/ SI + input). These two designs result from the different natures of sparse retrieval and dense retrieval, and we talk about the impact in Section 4.

3.2 R^3: Rule Relevance ReEstimate

The principle of retrieval is to match two different strings based on keyword or semantic similarity. Therefore, if the inductive capability of the LLM is not strong enough, the generated rule might still not align well with the golden rule. As a result, the retrieval list produced by SIAR may still have suboptimal ranking quality. Moreover, the retriever cannot determine whether a rule can aid the LLM in reasoning, nor can it specifically assess the relationship between the two. To address these, we propose to rerank the top-n rules from the previous



Alice moved to California where the newly ratified environmental law mandates recycling. Question: Does Alice need to follow the recycling mandate?

If Person X moves to Region Z and Legislation Y applies to Region Z, then Person X needs to obey Legislation Y.

MoveTo(Person X, Region Z) ^ Apply(Legislation Y, Region Z) => NeedToObey(Person, Legislation Y)

If Person X moves to Region Y and Region Y has implemented Legislation Z, then Person X needs to follow Legislation Z.

If Person X moves to Region Z and Legislation Y applies to Region Z, then Person X needs to obey Legislation Y.

If the abstract knowledge in a rule can be instantiated into the facts in query, that rule is more relevant.
If a rule can be applied to the query and thus be helpful for reasoning, that rule is more relevant.
...
Output the ranking list by the relevance in descending order.
...
query: {q}
[1]. {r1} ... [n]. {rn}

Figure 2: The workflow of `retrieve-then-reason` augmented with our method is shown in the middle of the Figure. To address the semantic misalignment issues, self-induction is first utilized to generate the hypothesized rule for query augmentation. Then, the new query is used for rule retrieval. And the retrieved rules are concatenated with the original query for reasoning. Building on this, we can reestimate the relevance of the rules with the query and improve the retrieval quality for better reasoning. The left bottom of the Figure shows the example of the self-induced rule. And the right bottom of the Figure shows the simplified reestimation prompt.

stage by evaluating the relevance of the retrieved rules to the original query, as shown in Figure 2. The key to R^3 is determining whether the abstract knowledge in a rule can be instantiated into the facts contained in the query, and whether the rule can assist the LLM in reasoning.

Inspired by RankGPT (Sun et al., 2023b), we prompt the LLM to directly output a ranked list of rules, which can reduce the prompting times compared to pair-wise estimation and thus accelerate the R^3 process. And we select the top-k rules from this reranked list as the final retrieval result. The corresponding instruction template is shown in Appendix A. This prompt encourages the LLM to assess both the relevance and utility of each rule, ensuring a more accurate final retrieval. Based on the query used in SIAR, R^3 also has two versions: R^3 (w/ SI) and R^3 (w/ SI + input).

4 Experiment

We select two synthetic datasets, Clutrr (Sinha et al., 2019) and ULogic (Wang et al., 2024c), as well as a real-world dataset, CAIL2018 (Xiao et al., 2018) from RuleBench for our evaluation. We report Recall@1, Recall@5, Recall@10 for retrieval results and Match for reasoning results. We use gpt-4o (OpenAI, 2024) and Qwen2.5 (Team, 2024) series (7B and 72B) as the tested LLMs. Due to space limitation, we put the entire experiment setting in Appendix B.

4.1 Retrieval Results and Discussion

As shown in Table 1 and Table 4, we report the retrieval performance using the different rule libraries (natural vs. formal). In each table, we present the performance of different retrievers (sparse vs. dense), LLMs with varying architectures (openai-gpt vs. Qwen) and parameter scales (72B vs. 7B), and different forms of queries (w/ SI vs. w/ SI + input). Due to space limitation, we put the formal language results in the Table 4 of the Appendix C.

Open-source models have comparable performance with closed-source models. In most settings, Qwen2.5-72B-Instruct demonstrates performance similar to GPT-4o, and in certain configurations, such as SIAR-R3 (w/ SI) + CAIL2018, it

	CLUTRR			ULogic			CAIL2018		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
sparse retrieval (BM25)									
`vanilla retrieval`	6.67	16.60	24.52	68.91	85.42	92.29	25.30	49.40	59.04
`gpt-4o`									
+ SIAR (w/ SI)	7.16	15.55	22.71	58.43	81.08	87.35	68.07	87.34	93.98
+ SIAR (w/ SI + input)	10.78	23.00	30.73	75.78	91.93	96.63	61.45	84.33	89.16
+ SIAR-R^3 (w/ SI)	11.45	19.75	24.62	87.83	92.65	92.65	83.13	**93.38**	**93.98**
+ SIAR-R^3 (w/ SI + input)	**16.32**	**26.81**	**32.92**	**92.65**	**97.71**	**97.83**	**83.73**	92.17	92.17
`Qwen2.5-72B-Instruct`									
+ SIAR (w/ SI)	8.11	17.36	26.04	57.22	79.76	86.87	78.91	88.55	90.96
+ SIAR (w/ SI + input)	11.06	23.66	**32.44**	74.82	90.48	94.94	74.70	86.14	89.76
+ SIAR-R^3 (w/ SI)	13.36	22.42	28.24	87.47	93.25	93.37	**86.75**	**93.98**	**93.98**
+ SIAR-R^3 (w/ SI + input)	**14.31**	**25.38**	31.58	**92.17**	**96.39**	**96.75**	86.14	92.17	92.17
`Qwen2.5-7B-Instruct`									
+ SIAR (w/ SI)	2.29	8.30	11.93	60.48	83.86	90.00	62.05	71.08	77.11
+ SIAR (w/ SI + input)	**7.06**	**16.70**	**22.81**	76.14	90.24	95.18	57.83	74.10	78.31
+ SIAR-R^3 (w/ SI)	2.00	6.39	10.02	84.81	93.13	93.37	72.89	80.12	80.12
+ SIAR-R^3 (w/ SI + input)	4.58	12.5	19.27	**88.67**	**96.14**	**96.50**	**75.30**	**83.13**	**84.33**
dense retrieval (bge)									
`vanilla retrieval`	2.10	7.73	12.02	30.36	58.43	71.45	9.04	15.66	21.69
`gpt-4o`									
+ SIAR (w/ SI)	12.31	22.61	28.91	65.18	87.11	91.33	56.02	74.70	81.33
+ SIAR (w/ SI + input)	5.25	12.60	19.37	43.61	73.13	83.61	21.68	40.96	51.81
+ SIAR-R^3 (w/ SI)	**16.51**	**25.38**	30.34	**86.62**	**94.58**	**94.70**	**78.31**	**84.94**	**84.94**
+ SIAR-R^3 (w/ SI + input)	10.21	18.80	**31.58**	80.72	88.92	89.52	55.42	60.84	60.84
`Qwen2.5-72B-Instruct`									
+ SIAR (w/ SI)	11.74	**24.71**	**31.58**	64.82	86.74	92.65	76.51	84.34	85.54
+ SIAR (w/ SI + input)	4.68	12.98	19.37	41.80	71.80	83.25	23.49	49.40	57.23
+ SIAR-R^3 (w/ SI)	**14.03**	23.09	30.25	**88.19**	**94.70**	**95.06**	**81.32**	**88.55**	**89.15**
+ SIAR-R^3 (w/ SI + input)	10.31	16.22	20.32	83.86	89.76	90.00	66.27	69.28	69.28
`Qwen2.5-7B-Instruct`									
+ SIAR (w/ SI)	**5.53**	**12.21**	**16.13**	71.57	90.96	95.54	59.64	68.07	69.88
+ SIAR (w/ SI + input)	2.96	10.88	16.13	42.89	71.93	83.01	23.49	43.98	51.81
+ SIAR-R^3 (w/ SI)	2.39	7.73	11.45	**87.59**	**96.39**	**97.11**	**70.48**	**74.10**	**74.70**
+ SIAR-R^3 (w/ SI + input)	2.29	7.54	11.64	76.75	87.95	88.92	56.02	59.64	60.24

Table 1: Performance of different methods with rule library in *Natural Language*. We use Recall@1, Recall@5 and Recall@10 as the retrieval metrics.

even outperforms GPT-4o. Moreover, compared to the baseline, they all achieve better performance. Therefore, we believe that open-source models have reached a level of rule induction and ranking capability comparable to that of the most advanced closed-source models. For fair comparison, the following analysis uses 72B and 7B Qwen models within the same family.

SIAR can consistently improve performance compared to vanilla retrieval. Under various combinations of retrievers, rule formats, and query formats, SIAR consistently outperforms direct retrieval. In different scenarios, SIAR achieves improvements of up to 9.64 (natural, dense, 72B, w/ SI), 60.12 (formal, dense, 72B, w/ SI), and 67.47 (natural, dense, 72B, w/ SI) in Recall@1 on Clutrr, ULogic, and CAIL2018, respectively. These results highlight the self-induction capabilities of LLMs, enabling them to effectively project queries into the rule space and reduce semantic misalignment between queries and rules. Additionally, we observe that models with 72B parameters tend to exhibit greater performance gains compared to 7B models, suggesting that inductive abilities improve with larger model scales.

SIAR-R^3 can usually improve performance compared to SIAR. On ULogic and CAIL2018, R^3 significantly boosts the performance of SIAR across all setup combinations. Notably, SIAR-R^3 achieves maximum improvements in Recall@1 of 43.25 (formal, dense, 72B, w/SI + input) and 42.78 (natural, dense, 72B, w/SI + input). These results indicate that R^3 effectively reevaluates and reranks the relevance of rules retrieved by SIAR. By directly assessing the relevance between the query and the rule, R^3 overcomes the limitations of retrievers that rely solely on keyword or semantic similarity, thus enhancing retrieval quality. Ad-

	gpt-4o			Qwen2.5-72B-Instruct			Qwen2.5-7B-Instruct		
	CLUTRR	ULogic	CAIL2018	CLUTRR	ULogic	CAIL2018	CLUTRR	ULogic	CAIL2018
w/o retrieval									
Direct	42.65	92.28	76.47	38.36	93.01	80.12	25.34	87.47	61.45
Golden rule	93.51	89.40	98.67	89.03	94.58	98.90	82.06	88.67	96.39
CoT	51.34	93.61	77.85	49.43	90.12	83.13	17.84	88.07	69.88
Self-Induction	50.76	87.83	82.98	49.62	91.69	84.94	31.58	88.43	64.46
w/ sparse retrieval									
vanilla	37.69	89.04	74.36	37.60	93.13	73.49	26.81	87.11	36.75
SIAR	46.09	87.71	80.77	49.14	94.21	86.14	33.87	88.92	59.64
SIAR-R^3	49.33	**89.64**	**85.71**	**51.71**	**95.90**	**86.75**	**33.97**	91.33	**73.49**
w/ dense retrieval									
vanilla	34.06	88.07	80.65	30.53	90.00	72.89	25.00	83.25	19.88
SIAR	52.19	86.39	80.75	49.81	95.06	86.75	34.73	89.64	60.24
SIAR-R^3	**54.20**	89.28	83.54	51.05	95.78	84.94	33.59	**91.81**	68.07

Table 2: Downstream reasoning performance. We use Match as the metric.

ditionally, on the CLUTRR dataset, performance gains were only observed in models with 72B parameters, and the improvements from the formal rule base were smaller than those from the natural language rule base. This suggests that on more complex datasets, models with smaller parameter scales lack the capacity to effectively rerank rules, limiting their ability to drive performance improvements.

The performance difference of sparse retrieval and dense retrieval depends on the format of rule and dataset. On the Clutrr and CAIL2018 datasets, sparse retrieval generally outperforms dense retrieval. However, on the ULogic dataset, performance varies depending on the rule base used. With the natural language rule base, sparse retrieval achieves a higher accuracy (92.17) compared to dense retrieval (88.19). Conversely, with the formal rule base, dense retrieval (89.75) surpasses sparse retrieval (80.60). This suggests that retrieval performance is highly dependent on the dataset and the linguistic form of the rule base. Despite these variations, we believe that in most cases, sparse retrieval will outperform dense retrieval. This is because, in rule-based scenarios, many concepts may not be well-represented in dense vector spaces. In contrast, sparse retrieval, which relies on keyword matching, may offer a more precise alignment between the query and the corresponding rules.

We add more analysis in Appendix D.

4.2 Reasoning Results and Discussion

Baselines (1) Direct: answer the question directly. This is set as the bottom of the performance. (2) Golden rule: answer the question with the golden rule. This is set as the ceil of performance. (3) CoT (Wei et al., 2022): reason step by step and then produce the answer. (4) Self-Induction: answer the question with the self-induced rule. (5) vanilla retrieval: use the original query to retrieve the rule and then answer the question.

Based on the conclusion from the previous section, for sparse retrieval, we use SI+input as the query, while for dense retrieval, we use SI as the query for retrieval. And we use the rule library in natural language. We report downstream reasoning performance in Table 2.

We use the average performance enhancement over three different datasets to analyze and get the following conclusions. Similarly as the Section 4.1, GPT-4o and Qwen2.5-72B have comparable performance, so we use Qwen-72B and Qwen-7B for further analysis.

An exception occurs with the ULogic, where gpt-4o outperforms the golden rule even without utilizing the rules. Based on our observations, gpt-4o has already achieved a relatively saturated performance (>90%) on this dataset, and additional rule knowledge may not bring further performance improvements on this dataset. Apart from this, the conclusions drawn from the analysis are reflected on the other two datasets.

Rules can effectively assist LLMs in reasoning, while directly retrieving rules for reasoning may lead to a decline in performance. Incorporating the Golden Rule as an aid in reasoning, rather than directly answering questions, has significantly improved performance across various models. For instance, the Qwen2.5-7B-Instruct model saw an

	Qwen-2.5-7B-Instruct			Llama-3.1-8B-Instruct			Yi-1.5-6B-Chat		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
w/ sparse retrieval									
vanilla retrieval	68.91	85.42	92.29	68.91	85.42	92.29	68.91	85.42	92.29
vanilla retrieval + R^3	86.99	93.98	94.70	56.71	92.77	93.98	69.88	86.14	91.08
SIAR (w/ SI)	71.57	90.96	95.54	54.94	80.12	86.87	59.76	82.65	88.31
SIAR (w/ SI + input)	42.89	71.93	83.01	**76.87**	**91.69**	**96.27**	**74.58**	**90.10**	**95.06**
SIAR-R^3 (w/ SI)	84.81	93.13	93.37	61.08	89.04	90.36	61.08	83.73	87.83
SIAR-R^3 (w/ SI + input)	**88.67**	**96.14**	**96.50**	58.31	94.94	96.62	73.25	90.00	94.22
w/ dense retrieval									
vanilla retrieval	30.36	58.43	71.45	30.36	58.43	71.45	30.36	58.43	71.45
vanilla retrieval + R^3	70.12	79.88	80.48	59.15	78.43	79.64	39.52	61.93	71.33
SIAR (w/ SI)	60.48	83.86	90.00	59.76	85.54	90.36	**69.28**	**87.95**	**92.05**
SIAR (w/ SI + input)	76.14	90.24	95.18	42.29	72.29	82.53	41.45	70.60	82.05
SIAR-R^3 (w/ SI)	**87.59**	**96.39**	**97.11**	**66.75**	**92.17**	**93.01**	68.19	87.59	91.32
SIAR-R^3 (w/ SI + input)	76.75	87.95	88.92	63.86	87.35	89.40	47.71	71.80	81.69

Table 3: Retrieval performance with different types of models.

average improvement of 31.54, while the Qwen2.5-72B-Instruct model showed a gain of 23.67. These substantial improvements suggest that the Golden Rule effectively enhances the ability of LLMs to infer from existing information, generate new knowledge, and make more reasonable decisions. In contrast, when relying on vanilla retrieval, performance decreases by 7.28 on the Qwen2.5-7B-Instruct model and by 2.43 on the Qwen2.5-72B-Instruct model. Vanilla dense retrieval leads to even larger drops, with declines of 14.79 and 6.03, respectively. These findings indicate that reasoning without accurate rule-based assistance, such as the golden rule, is less effective when based solely on vanilla retrieved results. It is noteworthy that using the question directly as a query for rule retrieval often produces low-quality, noisy results. The noise negatively impacts the reasoning process and degrades model performance. This phenomenon underscores a key challenge faced by current retrieval systems: semantic misalignment between queries and rules. Existing retrieval techniques struggle to accurately compute the similarity between the two, resulting in difficulty retrieving truly relevant rules, which ultimately hampers reasoning performance.

SIAR and R^3 can boost the performance significantly. The SIAR method significantly enhances model performance compared to direct retrieval. In scenarios utilizing sparse retrieval, performance increased by 31.76 and 25.27 for the 7B and 72B models, respectively. The improvements are even more pronounced with dense retrieval, where performance gains reached 56.48 and 38.20 for the 7B and 72B models. These results demonstrate that SIAR provides substantial performance boosts across models of varying sizes. When the R^3 mechanism was introduced, performance improved further. In sparse retrieval, the 7B model gains an additional 16.36, while the 72B model sees an increase of 4.87 points. For dense retrieval, the 7B model achieves an extra gain of 8.86, and the 72B model improves by 0.15. These findings validate the effectiveness of SIAR in enhancing retrieval quality, allowing for better alignment between queries and relevant rules, which in turn strengthens the reasoning process. SIAR addresses the semantic mismatch inherent in traditional retrieval methods by self-induction to map queries into the rule space. The R^3 mechanism further refines the retrieval by reassessing the relevance and applicability of each rule to the current query, overcoming the limitations of traditional retrievers that struggle to evaluate rules effectively. Compared to other baselines that do not utilize retrieval, our method demonstrates significant superiority. These results highlight the critical role of high-quality rule retrieval in reasoning tasks, showing that accurate retrieval is essential for improving reasoning performance.

5 Ablation Study

We perform more ablation experiments to explore more influencing factors. We use the ULogic dataset and test six methods: vanilla prompt, vanilla prompt + R^3, SIAR (w/ SI), SIAR (w/ SI + input), SIAR-R^3 (w/ SI), and SIAR-R^3 (w/ SI + input). Among them, "vanilla prompt + R^3" refers to retrieving using the original query and then directly performing R^3. Due to space limitations, this method was not presented in the previous section. Moreover, we put the result table of Section 5.2 and Section 5.3 in Appendix E.

5.1 The effects of different models

Different LLMs have different model architectures and use different training data. To demonstrate the generalizability of our method, we conducted experiments on a wider range of model types (Dubey et al., 2024; AI et al., 2025), as shown in the table 3. The results show that our method achieves a significant improvement over the baseline across different models.

5.2 The effects of different retrievers

Different types of retrievers have different characteristics. To validate the generalizability of our method, we compared the performance of three different types of retrievers: sparse retriever (bm25), dense retriever (bge), and LLM retriever (bge-gemma2 (Chen et al., 2024a)). For comparison, the dense retriever has only 110M parameters, and the LLM retriever has 9B parameters. As shown in the table 5, the results demonstrate that our method performs well across different types of retrievers. Even with large retrieval model, our method is still able to provide further enhancement, which strongly demonstrates the generalizability of our approach.

5.3 The effects of the number of rules

To validate the robustness of our method, we added the counterfactual rule set from the ulogic dataset (constructed by the original RuleBench (Sun et al., 2024b)) to the original rule set and re-tested the performance of our method, as shown in Table 3. In this setup, the number of rules doubles compared to the original. As the number of irrelevant rules increases, the performance of retrieval will continuously decline. So the number of rules is a very important influencing factor. However, our method still demonstrates a significant improvement compared to the baseline.

6 Related Work

6.1 LLM and rule

As the inductive (Yang et al., 2024b; Wang et al., 2024c) and deductive (Saparov et al., 2023) capabilities of LLMs continue to advance, they are increasingly being employed to summarize latent transformation patterns from sets of inputs and outputs (Sun et al., 2024a; Qiu et al., 2024). These patterns are then formalized as executable rules, stored, and used to support reasoning in downstream tasks (Yang et al., 2023; Sun et al., 2023a; Zhu et al., 2024; Wang et al., 2024b,d).

More specifically, They learn rules from input-output pairs to represent relationships between inputs and outputs, then use these rules for reasoning and quality verification. High-quality rules are stored in a library (Zhu et al., 2024; Wang et al., 2024b). Previous research on rule retrieval includes two methods: one concatenates all rules with the input for inference, requiring a hierarchical storage structure (Zhu et al., 2024); the other uses vanilla retrieval (Sun et al., 2023a; Yang et al., 2023), which deteriorates as the rule set grows. This paper focuses on addressing semantic misalignment and relevance estimation issues in retrieval, proposing solutions to improve semantic matching and relevance evaluation for more accurate rule retrieval.

6.2 Generation Augmented Retrieval

The Generation Augmented Retrieval (GAR) (Mao et al., 2021) is a common approach that leverages the capabilities of language models to perform query decomposition (Chen et al., 2024b), query rewriting (Ma et al., 2023), and query expansion (Wang et al., 2023), helping to supplement missing background knowledge in queries to achieve higher retrieval quality. In addition to passage retrieval, GAR can also play a role in code retrieval (Li et al., 2024). Our SIAR can be seen as a type of GAR that utilizes the self-inductive abilities of large language models (LLMs).

7 Conclusion

This paper introduces Self-Induction Augmented Retrieval (SIAR) and Rule Relevance Re-Estimate (R^3) to address the challenges of rule retrieval in complex reasoning tasks. These techniques significantly enhance retrieval accuracy by LLMs to induce abstract inferential rules and assess the relevance of retrieved rules to queries. SIAR and R3 offer promising solutions for overcoming the semantic misalignment issues in traditional retrieval techniques, paving the way for more effective rule-based reasoning in real-world applications.

Limitations

Currently, the rule libraries we discussed remain quite limited in size, as seen in datasets like Clutrr, ULogic, and CAIL2018, which contain only 1,048, 830, and 166 rules, respectively. Compared to the vast number of articles in traditional passage retrieval, the rule bases we retrieved are still relatively small. However, even with these small datasets, traditional retrieval methods have shown a decline in reasoning performance, underscoring the need for deeper exploration in rule retrieval. The smaller number of rules reduces the difficulty of the benchmark. In future work, we aim to introduce more irrelevant rules to explore additional challenges in rule retrieval.

Acknowledgement

This work was supported by the Beijing Natural Science Foundation (L243006) and the National Natural Science Foundation of China (No.62376270).

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2025. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Chen Bowen, Rune Sætre, and Yusuke Miyao. 2024. A comprehensive evaluation of inductive reasoning capabilities and problem solving in large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 323–339, St. Julian's, Malta. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Xinran Chen, Xuanang Chen, Ben He, Tengfei Wen, and Le Sun. 2024b. Analyze, generate and refine: Query expansion with LLMs for zero-shot open-domain QA. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11908–11922, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Kewei Cheng, Jingfeng Yang, Haoming Jiang, Zhengyang Wang, Binxuan Huang, Ruirui Li, Shiyang Li, Zheng Li, Yifan Gao, Xian Li, et al. 2024. Inductive or deductive? rethinking the fundamental reasoning abilities of llms. *arXiv preprint arXiv:2408.00114*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ziyang Huang, Xiaowei Yuan, Yiming Ju, Jun Zhao, and Kang Liu. 2025a. Reinforced internal-external knowledge synergistic reasoning for efficient adaptive search agent. *Preprint*, arXiv:2505.07596.

Ziyang Huang, Jun Zhao, and Kang Liu. 2025b. Towards adaptive mechanism activation in language agent. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2867–2885, Abu Dhabi, UAE. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Haochen Li, Xin Zhou, and Zhiqi Shen. 2024. Rewriting the code: A simple method for large language model augmented code search. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1371–1389, Bangkok, Thailand. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

OpenAI. 2024. Hello gpt-4o. *https://openai.com/index/hello-gpt-4o/*.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2024. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *The Twelfth International Conference on Learning Representations*.

David Rau, Hervé Déjean, Nadezhda Chirkova, Thibault Formal, Shuai Wang, Vassilina Nikoulina, and Stéphane Clinchant. 2024. Bergen: A benchmarking library for retrieval-augmented generation. *Preprint*, arXiv:2407.01102.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using ood examples. *Preprint*, arXiv:2305.15269.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle, and Philip Resnik. 2024. The prompt report: A systematic survey of prompting techniques. *Preprint*, arXiv:2406.06608.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Shangeetha Sivasothy, Scott Barnett, Rena Logothetis, Mohamed Abdelrazek, Zafaryab Rasool, Srikanth Thudumu, and Zac Brannelly. 2024. Large language models for generating rules, yay or nay? *Preprint*, arXiv:2406.06835.

Wangtao Sun, Haotian Xu, Xuanqing Yu, Pei Chen, Shizhu He, Jun Zhao, and Kang Liu. 2024a. ItD: Large language models can teach themselves induction through deduction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2719–2731, Bangkok, Thailand. Association for Computational Linguistics.

Wangtao Sun, Xuanqing Yu, Shizhu He, Jun Zhao, and Kang Liu. 2023a. Expnote: Black-box large language models are better task solvers with experience notebook. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Wangtao Sun, Chenxiang Zhang, Xueyou Zhang, Ziyang Huang, Haotian Xu, Pei Chen, Shizhu He, Jun Zhao, and Kang Liu. 2024b. Beyond instruction following: Evaluating rule following of large language models. *arXiv preprint arXiv:2407.08440*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023b. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah Goodman. 2024b. Hypothesis search: Inductive reasoning with language models. In *The Twelfth International Conference on Learning Representations*.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024c. Can LLMs reason with rules? logic scaffolding for stress-testing and improving LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7523–7543, Bangkok, Thailand. Association for Computational Linguistics.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024d. Symbolic working memory enhances language models for complex rule application. *Preprint*, arXiv:2408.13654.

Xiaohan Wang, Xiaoyan Yang, Yuqi Zhu, Yue Shen, Jian Wang, Peng Wei, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. 2024e. Rulealign: Making large language models better physicians with diagnostic rule alignment. *arXiv preprint arXiv:2408.12579*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *Preprint*, arXiv:1807.02478.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zeyuan Yang, Peng Li, and Yang Liu. 2023. Failures pave the way: Enhancing large language models through tuning-free rule accumulation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1751–1777, Singapore. Association for Computational Linguistics.

Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. 2024b. Language models as inductive reasoners. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–225, St. Julian's, Malta. Association for Computational Linguistics.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through inconsistent solving perspectives. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3602–3622, Bangkok, Thailand. Association for Computational Linguistics.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024a. Expel: Llm agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024b. A survey of large language models. *Preprint*, arXiv:2303.18223.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *Preprint*, arXiv:2302.09419.

Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. 2024. Large language models can learn rules.

A Prompt Template

Self-Induction Prompt Template

You are given a Query. Please write the inferential rule may help answer the question. The rule should summarize and abstract the facts in the query and catch the underlying logic. I will give some examples. Just output the rule and do not output anything else.
query: $\{q_1\}$
rule: $\{r_1\}$
... more demonstrations

Rule Relevance ReEstimate Prompt Template

You are an intelligent assistant that can rank rules based on their relevancy to the query. If the abstract knowledge in a rule can be instantiated into the facts in query, that rule is more relevant. If a rule can be applied to the query and thus be helpful for reasoning, that rule is more relevant. I will provide you with {*num_rule*} rules, each indicated by a numerical identifier []. Rank the rules based on their relevance to the query: {*query*}.
[1] $\{\text{rule}_1\}$
[2] $\{\text{rule}_2\}$
...
Query: {*query*}.
Rank the {*num*} rules above based on their relevance to the query. All the rules should be included and listed using identifiers, in descending order of relevance. The output format should be [] > [], e.g., [2] > [1], Only respond with the ranking results, do not say any word or explain.

B Experiment setting

Test Benchmark RuleBench (Sun et al., 2024b) evaluates the reasoning capabilities of large language models (LLMs) under a given set of rules. Building on the foundation of RuleBench, we consolidate all rules within the entire test set into a comprehensive rule library and used the original questions as queries. We establish both a natural language-based and a formal language-based rule library to assess the impact of different rule formats on retrieval performance. We select two synthetic datasets, Clutrr (Sinha et al., 2019) and ULogic (Wang et al., 2024c), as well as a real-world dataset, CAIL2018 (Xiao et al., 2018) from RuleBench for our evaluation.

Metrics We employ Recall@1, Recall@5, and Recall@10 to evaluate retrieval performance, and use the Match metric (Rau et al., 2024) to assess reasoning performance. Specifically, if the golden answer appears in the final answer generated by the LLM, then it is considered correct.

Implementation Details We use Pyserini (Lin et al., 2021) to implement the BM25 retriever and employ bge-base-en (Xiao et al., 2023) as the dense encoder. For self-induction, rule relevance re-estimation, and final reasoning, we utilize the gpt-4o (OpenAI, 2024) and Qwen2.5 (Team, 2024) series (7B and 72B) as the tested LLMs. We leverage VLLM (Kwon et al., 2023) to accelerate inference. For SIAR, we get top-10 rules for retrieval performance evaluation and use the top-1 rule for reasoning evaluation. For SIAR-R^3, we get top-20 rules by SIAR, and use R^3 to get the top-10 relevant rules for retrieval evaluation and use the top-1 rule for reasoning evaluation.

C Retrieval performance with rule library in Formal Language

Due to the api cost, we do not test the gpt-4o performance on Formal Language rules. We show the results in Talbe 4.

D More analysis on retrieval results.

w/ SI is suitable for dense retrieval, while w/ SI + input is suitable for sparse retrieval. Under the same conditions, we observe that when using SI as the query, dense retrieval typically outperforms sparse retrieval. Conversely, when using SI+input as the query, sparse retrieval tends to perform better than dense retrieval. This difference can be attributed to the nature of the two retrieval methods. Dense retrievers map both the query and the rule into a unified vector space to measure semantic similarity, whereas sparse retrievers rely on keyword matching. When SI+input is used as the query, it can disrupt the semantic coherence of the SI, while the rules in the library remain intact, resulting in a decrease in similarity within the vector space. As a result, dense retrieval is more effective when SI alone is used as the query. In contrast, for sparse retrieval, if the query contains keywords from the target rule, it can augment the SI, thus increasing the BM25 score between the SI and the rule. This makes sparse retrieval more suitable when SI+input is used as the query.

Rule library suits more in the format of *Natural Language*. By comparing Table 1 and Table 4, we observe that SIAR and SIAR-R^3 perform better when retrieving from the natural language rule base. Rules expressed in formal language are more abstract and harder to interpret, making it more challenging for the LLM to perform self-induction and assess relevance. Poor self-induction and relevance reestimation by the LLM can therefore degrade retrieval quality.

E Ablation results on types of retrievers and the number of rules.

We show the ablation results in Table 5 and Table 6.

	CLUTRR			ULogic			CAIL2018		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
sparse retrieval (BM25)									
`vanilla retrieval`	6.58	16.60	24.43	22.41	44.10	51.57	22.89	42.77	53.01
`Qwen2.5-72B-Instruct`									
+ SIAR (w/ SI)	6.58	18.89	27.67	45.90	68.67	75.30	79.52	93.37	95.18
+ SIAR (w/ SI + input)	10.01	**23.76**	**31.97**	51.57	73.49	80.24	59.04	83.13	89.15
+ SIAR-R^3 (w/ SI)	10.21	18.80	26.81	75.54	80.12	80.36	**83.13**	**95.78**	**95.78**
+ SIAR-R^3 (w/ SI + input)	**11.07**	23.00	30.25	**80.60**	**85.42**	**85.66**	83.13	93.37	93.37
`Qwen2.5-7B-Instruct`									
+ SIAR (w/ SI)	2.39	6.34	10.21	47.47	68.67	76.27	69.88	88.55	93.37
+ SIAR (w/ SI + input)	**7.16**	**16.89**	**23.57**	50.48	73.49	81.08	46.99	72.89	82.53
+ SIAR-R^3 (w/ SI)	1.34	5.25	9.73	72.17	81.33	82.17	**78.31**	**92.17**	**93.37**
+ SIAR-R^3 (w/ SI + input)	2.67	11.07	17.84	**75.90**	**85.66**	**86.02**	72.29	87.35	89.76
dense retrieval (bge)									
`vanilla retrieval`	2.86	8.59	12.79	18.31	43.98	55.66	1.81	7.83	14.46
`Qwen2.5-72B-Instruct`									
+ SIAR (w/ SI)	8.30	**20.90**	**27.10**	76.50	90.72	94.46	40.96	60.24	64.46
+ SIAR (w/ SI + input)	4.29	10.59	17.37	30.0	60.24	69.88	6.02	19.28	33.13
+ SIAR-R^3 (w/ SI)	**8.87**	18.70	25.48	**89.75**	**95.90**	**96.02**	**62.65**	**71.69**	**71.69**
+ SIAR-R^3 (w/ SI + input)	6.97	14.98	19.47	73.25	80.36	80.60	37.35	42.17	42.77
`Qwen2.5-7B-Instruct`									
+ SIAR (w/ SI)	2.96	8.59	11.93	78.43	92.17	95.54	24.70	44.58	54.22
+ SIAR (w/ SI + input)	**3.81**	**9.64**	**15.08**	31.08	60.36	69.76	8.43	19.88	29.52
+ SIAR-R^3 (w/ SI)	1.43	6.58	10.30	**88.31**	**96.98**	**97.23**	**55.42**	**64.46**	**64.46**
+ SIAR-R^3 (w/ SI + input)	3.05	7.35	11.93	69.40	80.24	80.48	33.73	40.96	41.57

Table 4: Performance of different methods with rule library in *Formal Language*. We use Recall@1, Recall@5 and Recall@10 as the retrieval metrics.

	`Sparse Retriever`			`Dense Retriever`			`LLM retriever`		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
w/ `Qwen2.5-7B-Instruct`									
vanilla retrieval	68.91	85.42	92.29	30.36	58.43	71.45	74.94	95.18	97.83
vanilla retrieval + R^3	86.99	93.98	94.70	70.12	79.88	80.48	86.99	97.83	98.67
SIAR (w/ SI)	60.48	83.86	90.00	71.57	90.96	95.54	78.07	95.06	97.47
SIAR (w/ SI + input)	76.14	90.24	95.18	42.89	71.93	83.01	86.39	98.67	99.88
SIAR-R^3 (w/ SI)	84.81	93.13	93.37	**87.59**	**96.39**	**97.11**	88.91	98.19	98.67
SIAR-R^3 (w/ SI + input)	**88.67**	**96.14**	**96.50**	76.75	87.95	88.92	**90.84**	**98.91**	**99.28**

Table 5: Retrieval performance with different types of retrievers.

	`Original`	`Original + Counterfactual`
w/ sparse retrieval		
vanilla retrieval	68.91/ 85.42/ 92.29	49.04/ 79.40/ 85.30
vanilla retrieval + R^3	86.99/ 93.98/ 94.70	79.64/ 90.48/ 91.69
SIAR (w/ SI)	60.48/ 83.86/ 90.00	59.16/ 77.11/ 84.34
SIAR (w/ SI + input)	76.14/ 90.24/ 95.18	72.53/ 87.71/ 91.08
SIAR-R^3 (w/ SI)	84.81/ 93.13/ 93.37	81.20/ 90.00/ 90.84
SIAR-R^3 (w/ SI + input)	**88.67 / 96.14 / 96.50**	**86.99 / 94.10 / 94.94**
w/ sparse retrieval		
vanilla retrieval	30.36/ 58.43/ 71.45	20.48/ 49.52/ 60.60
vanilla retrieval + R^3	70.12/ 79.88/ 80.48	56.02/ 69.64/ 70.84
SIAR (w/ SI)	71.57/ 90.96/ 95.54	67.83/ 87.47/ 92.53
SIAR (w/ SI + input)	42.89/ 71.93/ 83.01	32.53/ 62.77/ 73..49
SIAR-R^3 (w/ SI)	**87.59 / 96.39 / 97.11**	**84.33 / 94.57 / 95.42**
SIAR-R^3 (w/ SI + input)	76.75/ 87.95/ 88.92	66.87/ 80.48/ 82.65

Table 6: Retrieval performance with different number of rules. Original represents the rule set used in the previous section. Counterfactual represents the addtional rules we select from the RuleBench.