

Search-o1：赋予推理模型主动搜索的能力

原创 卡卡卡卡比 NLP工作站 2025年01月10日 16:42 江苏

今天给大家带来一篇好友知乎@卡卡卡卡比 的文章，Search-o1，通过自主检索来补足O1推理模式中知识不足的问题。

知乎: <https://zhuanlan.zhihu.com/p/17527068532>

Paper: <https://arxiv.org/abs/2501.05366>

Github: <https://github.com/sunnynexus/Search-o1>

近年来，大型推理模型（LRMs）如OpenAI-o1、Qwen-QwQ和DeepSeek-R1通过大规模强化学习展现了卓越的多步推理能力，能够有效解决复杂的推理问题。这些模型通过生成思维链条，模仿人类的解决问题方式，增强了推理的逻辑性和可解释性。然而，长时间的推理过程也带来了**知识不足**的问题，任何知识缺口都可能导致错误传播，破坏整个推理链条。

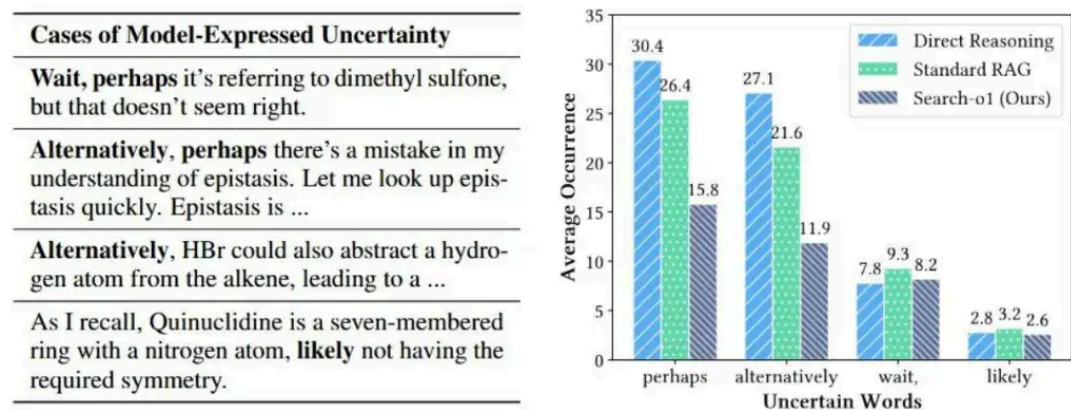
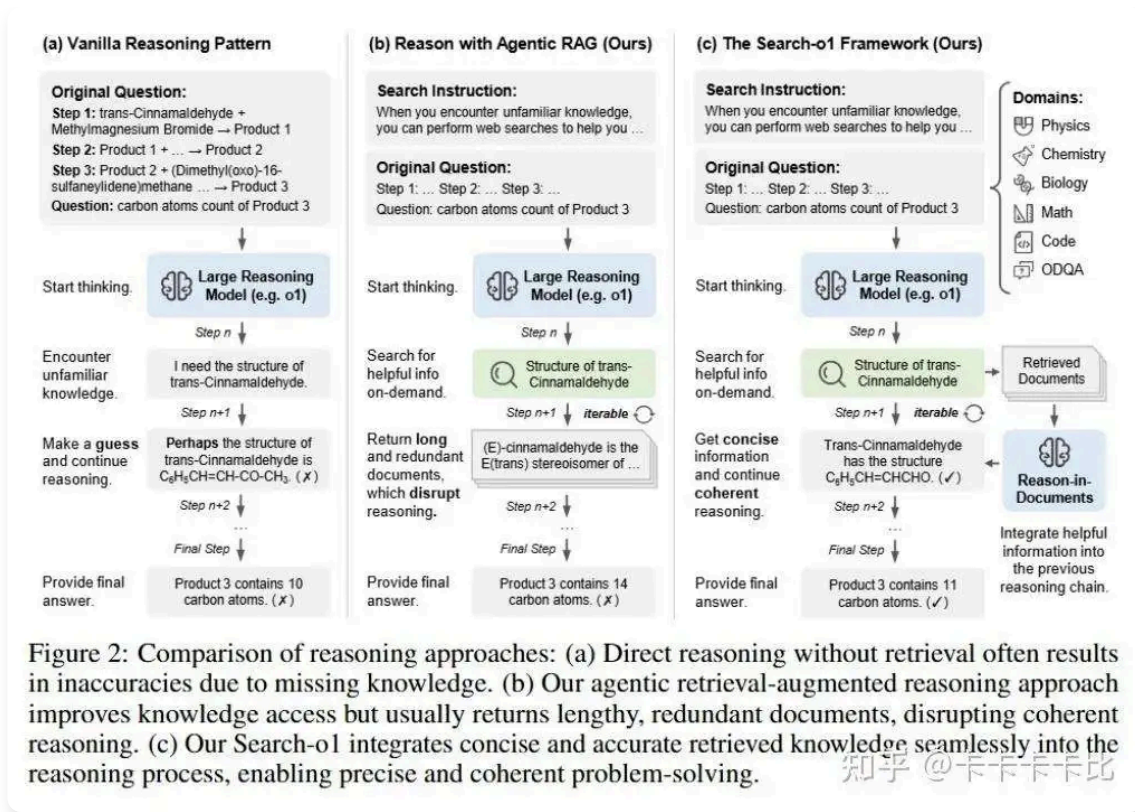


Figure 1: Analysis of reasoning uncertainty with QwQ-32B-Preview. **Left:** Examples of uncertain words identified during the reasoning process. **Right:** Average occurrence of high-frequency uncertain words per output in the GPQA diamond set.

初步实验表明，LRMs在处理复杂推理任务时频繁表达不确定性，表明其在知识覆盖方面存在不足。为了解决这一问题，本文提出了**Search-o1**，通过集成自主检索增强生成（RAG）机制和文档内推理模块，使LRMs能够在推理过程中自主检索和整合外部知识，弥补知识不足，保持推理的连贯性和准确性。

Search-o1：自主知识检索增强的推理框架

为了应对上述问题，本文提出了**Search-o1**框架。该框架融合了**自主检索增强生成（Agentic Retrieval-Augmented Generation）**机制和**文档内推理模块（Reason-in-Documents）**，从而实现了在推理过程中动态获取并整合外部知识的能力。



1. Search-o1框架概述

Search-o1框架通过两个核心组件增强LRMs的推理能力：

- **自主检索增强生成 (RAG) 机制：** 允许模型在推理过程中自主生成检索查询，动态获取所需的外部知识。
- **文档内推理模块：** 对检索到的文档进行精炼，提取相关且简明的信息，确保这些信息能够无缝整合到推理链条中，而不引入冗余或噪音。

2. 自主检索增强生成机制

在推理过程中，模型可能会生成被特殊符号包围的检索查询。一旦检测到检索查询，模型会暂停当前推理，利用查询从外部知识库中检索相关文档。检索到的文档随后被注入到推理链条中，供模型继续推理使用。这一机制使模型能够在需要时动态获取和整合外部知识，弥补内部知识的不足。

3. 知识精炼通过文档内推理

直接插入检索到的文档可能会引入冗余信息，扰乱推理的连贯性。文档内推理模块通过独立的生成过程，对检索到的文档进行深度分析，提炼出与当前推理相关的关键信息。这些精炼后的信息随后被整合回推理链条中，确保推理过程的逻辑性和连贯性。

4. Search-o1推理过程

Search-o1采用批量推理机制，能够高效处理多个问题。对于每个问题，初始化推理序列并生成推理步骤。当检测到检索查询时，批量处理这些查询，检索相关文档，并通过文档内推理模块进行精炼。精炼后

的知识被插入回各自的推理链条，继续推理，直至生成最终答案。

Algorithm 1 Search-o1 Inference

Require: Reasoning Model \mathcal{M} , Search function Search

```
1: Input: Questions  $\mathcal{Q}$ , Task instruction  $I$ , Reason-in-documents instruction  $I_{\text{docs}}$ 
2: Initialize set of unfinished sequences  $\mathcal{S} \leftarrow \{I \oplus q \mid q \in \mathcal{Q}\}$ 
3: Initialize set of finished sequences  $\mathcal{F} \leftarrow \{\}$ 
4: while  $\mathcal{S} \neq \emptyset$  do
5:   Generate next tokens for all sequences in  $\mathcal{S}$ :  $\mathcal{T} \leftarrow \mathcal{M}(\mathcal{S})$  ▷ Batch Generate
6:   Initialize empty set  $\mathcal{S}_r \leftarrow \{\}$  ▷ Reason-in-documents Inputs
7:   for each sequence  $\text{Seq} \in \mathcal{T}$  do
8:     if  $\text{Seq}$  ends with <end_search_query> then
9:       Pause generation for  $\text{Seq}$  ▷ Pause Generation
10:      Extract search query:  $q_{\text{search}} \leftarrow \text{Extract}(\text{Seq}, \text{<begin\_search\_query>}, \text{<end\_search\_query>})$ 
11:      Retrieve documents:  $\mathcal{D} \leftarrow \text{Search}(q_{\text{search}})$  ▷ Retrieval
12:      Construct input for Reason-in-documents:  $I_{\mathcal{D}} \leftarrow I_{\text{docs}} \oplus q_{\text{search}} \oplus \text{Seq}$ 
13:      Append the tuple  $(I_{\mathcal{D}}, \text{Seq})$  to  $\mathcal{S}_r$ 
14:     else if  $\text{Seq}$  ends with EOS then
15:       Remove  $\text{Seq}$  from  $\mathcal{S}$ , add  $\text{Seq}$  to  $\mathcal{F}$  ▷ Sequence Finished
16:   if  $\mathcal{S}_r \neq \emptyset$  then
17:     Prepare batch inputs:  $\mathcal{I}_r \leftarrow \{I_{\mathcal{D}} \mid (I_{\mathcal{D}}, \text{Seq}) \in \mathcal{S}_r\}$ 
18:     Reason-in-documents:  $\mathcal{T}_r \leftarrow \mathcal{M}(\mathcal{I}_r)$  ▷ Batch Generate
19:     for  $i \leftarrow \{1, \dots, |\mathcal{T}_r|\}$  do
20:       Let  $r \leftarrow \mathcal{T}_r[i]$ ,  $\text{Seq} \leftarrow \mathcal{S}_r[i].\text{Seq}$ 
21:       Extract knowledge-injected reasoning step:  $r_{\text{final}} \leftarrow \text{Extract}(r)$ 
22:       Update sequence in  $\mathcal{S}$ :  $\text{Seq} \leftarrow \text{Insert}(\text{<begin\_search\_result>}, r_{\text{final}}, \text{<end\_search\_result>})$ 
23: Output: Finished Sequences  $\mathcal{F}$ 
```

知乎 @卡卡卡卡比

实验

1. 任务与数据集

◦ 复杂推理任务:

- **GPQA**: 博士级科学多选问答，涵盖物理、化学和生物学。
- **数学基准**: 包括MATH500、AMC2023和AIME2024，覆盖算术、代数、几何等。
- **LiveCodeBench**: 评估编程能力，包含不同难度级别的问题。



◦ 开放领域问答任务:

- **单跳问答**: 包括Natural Questions (NQ) 和TriviaQA。
- **多跳问答**: 包括HotpotQA、2WikiMultihopQA (2WIKI)、MuSiQue和Bamboogle。

2. 基线方法

- **直接推理**: 不使用检索，依赖模型内部知识（如Qwen2.5、QwQ-32B、GPT-4o）。
- **检索增强推理**:
 - **标准RAG**: 对整个问题进行一次性文档检索。
 - **RAgent**: 允许模型在推理过程中生成多个检索查询。

Search-o1实验效果

1. 复杂推理任务

主要发现：

- **QwQ-32B**在无需检索的直接推理中表现优于更大的模型，显示出o1-like长链思维方法在复杂推理中的有效性。
- **RAgent-QwQ-32B**通过自主检索机制，在大多数任务上超越了标准RAG和直接推理方法。
- **Search-o1**通过文档内推理模块进一步提升了性能，在大多数任务上取得了最佳表现，显示出其在知识整合和推理连贯性方面的优势。

Table 1: Main results on challenging reasoning tasks, including PhD-level science QA, math and code benchmarks. We report Pass@1 metric for all tasks. For models with 32B parameters, the best results are in **bold** and the second-best are underlined. Results from larger or non-proprietary models are in gray color for reference. Symbol “†” indicates results from their official releases.

Method	GPQA (PhD-Level Science QA)				Math Benchmarks			LiveCodeBench			
	Physics	Chemistry	Biology	Overall	MATH500	AMC23	AIME24	Easy	Medium	Hard	Overall
<i>Direct Reasoning (w/o Retrieval)</i>											
Qwen2.5-32B	57.0	33.3	52.6	45.5	75.8	57.5	23.3	42.3	18.9	14.3	22.3
Qwen2.5-Coder-32B	37.2	25.8	57.9	33.8	71.2	67.5	20.0	<u>61.5</u>	16.2	12.2	25.0
QwQ-32B	75.6	39.8	68.4	58.1	83.2	<u>82.5</u>	<u>53.3</u>	<u>61.5</u>	<u>29.7</u>	20.4	33.0
Qwen2.5-72B	57.0	37.6	68.4	49.0	79.4	67.5	20.0	53.8	29.7	24.5	33.0
Llama3.3-70B	54.7	31.2	52.6	43.4	70.8	47.5	36.7	57.7	32.4	24.5	34.8
DeepSeek-R1-Lite†	-	-	-	58.5	91.6	-	52.5	-	-	-	51.6
GPT-4o†	59.5	40.2	61.6	50.6	60.3	-	9.3	-	-	-	33.4
o1-preview†	89.4	59.9	65.9	73.3	85.5	-	44.6	-	-	-	53.6
<i>Retrieval-augmented Reasoning</i>											
RAG-Qwen2.5-32B	57.0	37.6	52.6	47.5	82.6	72.5	30.0	<u>61.5</u>	24.3	8.2	25.9
RAG-QwQ-32B	<u>76.7</u>	38.7	<u>73.7</u>	58.6	84.8	<u>82.5</u>	50.0	57.7	16.2	12.2	24.1
RAgent-Qwen2.5-32B	58.1	33.3	63.2	47.0	74.8	65.0	20.0	57.7	24.3	6.1	24.1
RAgent-QwQ-32B	<u>76.7</u>	<u>46.2</u>	68.4	<u>61.6</u>	<u>85.0</u>	85.0	56.7	65.4	18.9	12.2	<u>26.8</u>
<i>Retrieval-augmented Reasoning with Reason-in-Documents</i>											
Search-o1 (Ours)	77.9	47.3	78.9	63.6	86.4	85.0	56.7	65.7	32.7	20.5	33.0

- 1. 大型推理模型的优势：**即使在没有检索增强的情况下，QwQ-32B-Preview模型在多个任务上也表现优异，甚至超过了一些更大规模的模型，如Qwen2.5-72B和Llama3.3-70B。这展示了大型推理模型在推理任务中的强大能力。
- 2. 自主检索增强的效果：**使用自主RAG机制的RAgent-QwQ-32B在大多数任务上超越了标准RAG和直接推理的QwQ-32B，表明自主检索能够有效提升推理模型的知识获取能力。
- 3. Search-o1的卓越表现：**进一步引入文档内推理模块后的Search-o1，在大多数任务上超越了RAgent-QwQ-32B，尤其在GPQA、数学和编码任务上取得了显著的性能提升。
- **扩展分析：**增加检索文档的数量进一步提升了Search-o1的性能，证明了动态检索和精炼机制的有效性。



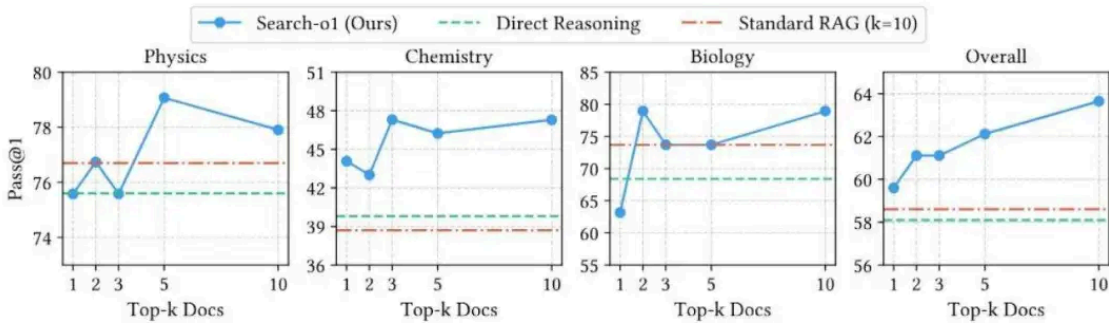


Figure 3: Scaling analysis of top-k retrieved documents utilized in reasoning. All results are based on QwQ-32B-Preview model.

研究发现，Search-o1能够有效利用增加的检索文档数量，进一步提升复杂推理任务的处理能力。即使只检索一篇文档，Search-o1也能够超过直接推理和标准RAG模型，显示出自主检索和文档精炼策略的高效性。

2. 开放领域问答任务

主要发现：

- 直接推理的LRMs在开放领域问答任务上的表现与非推理模型相似，表明存在知识不足。
- 检索增强方法显著提升了模型在多跳问答任务上的表现，尤其是RAgent-QwQ-32B在多跳问答上的表现优于标准RAG。
- Search-o1**在多跳问答任务上进一步超越所有基线方法，证明了文档内推理模块在复杂问答中的重要性 和有效性。

Table 3: Performance comparison on open-domain QA tasks, including single-hop QA and multi-hop QA datasets. For models with 32B parameters, the best results are in **bold** and the second-best are underlined. Results from larger models are in gray color for reference.

Method	Single-hop QA				Multi-hop QA							
	NQ		TriviaQA		HotpotQA		2WIKI		MuSiQue		Bamboogle	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
<i>Direct Reasoning (w/o Retrieval)</i>												
Qwen2.5-32B	22.8	33.9	52.0	60.3	25.4	34.7	29.8	36.3	8.4	18.0	49.6	63.2
QwQ-32B	23.0	33.1	53.8	60.7	25.4	33.3	34.4	40.9	9.0	18.9	38.4	53.7
Qwen2.5-72B	27.6	41.2	56.8	65.8	29.2	38.8	34.4	42.7	11.4	20.4	47.2	61.7
Llama3.3-70B	36.0	48.7	68.8	76.8	37.8	49.1	46.0	54.2	14.8	23.6	54.4	67.8
<i>Retrieval-augmented Reasoning</i>												
RAG-Qwen2.5-32B	33.4	<u>49.3</u>	65.8	79.2	38.6	50.4	31.6	40.6	10.4	19.8	52.0	66.0
RAG-QwQ-32B	29.6	44.4	<u>65.6</u>	<u>77.6</u>	34.2	46.4	35.6	46.2	10.6	20.2	<u>55.2</u>	<u>67.4</u>
RAgent-Qwen2.5-32B	32.4	47.8	63.0	72.6	<u>44.6</u>	<u>56.8</u>	55.4	69.7	13.0	25.4	54.4	66.4
RAgent-QwQ-32B	<u>33.6</u>	48.4	62.0	74.0	43.0	55.2	58.4	<u>71.2</u>	<u>13.6</u>	<u>25.5</u>	52.0	64.7
<i>Retrieval-augmented Reasoning with Reason-in-Documents</i>												
Search-o1 (Ours)	34.0	49.7	63.4	74.1	45.2	57.3	<u>58.0</u>	71.4	<u>15.1</u>	<u>28.2</u>	59.0	67.8

总结

本文提出的**Search-o1**框架通过集成自主检索增强生成机制和文档内推理模块，成功解决了大型推理模型在长时间推理过程中面临的知识不足问题。Search-o1能够在推理过程中动态检索和精炼外部知识，提升了推理的准确性和连贯性。大量实验结果表明，Search-o1在科学、数学、编程等复杂推理任务以及多个开放领域问答基准上均优于现有的检索增强和直接推理方法，部分领域的表现甚至超过了人类专家。这

表明Search-o1在提升LRMs的可信度和适用性方面具有显著潜力，为开发更可靠和能够利用更多工具的慢思考系统奠定了基础。

PS：看到这里，如果觉得不错，可以来个[点赞](#)、[在看](#)、[关注](#)。给公众号添加【星标★】不迷路！您的支持是我坚持的最大动力！

欢迎多多关注公众号「NLP工作站」，[加入交流群](#)，交个朋友吧，一起学习，一起进步！



NLP工作站

日常分享AIGC前沿知识&落地经验总结，也欢迎关注《ChatGPT原理与实战》、《大型语...
190篇原创内容

公众号

LLMs 132

LLMs · 目录

上一篇 · 从ROPE到Yarn, 一条通用公式速通长文本大模型中的位置编码

阅读原文

